

Cover-free families on graphs and hypergraphs

Prangya Parida

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirements for the degree of
Doctorate in Philosophy Mathematics and Statistics*

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

© Prangya Parida, Ottawa, Canada, 2026

*The Ph.D. program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Abstract

A family of subsets of a t -set is a d -cover-free family or d -CFF if no subset in the family is contained in the union of any d other subsets. Let $t(d, n)$ denote the minimum t for which there exists a d -CFF on a t -set with n subsets. Since a 1-CFF is the same as a Sperner family, using Sperner's theorem, we get $t(1, n) \sim \log_2(n)$ as n grows. Erdős, Frankl, and Füredi (JCTA 1982) proved that $3.106 \log_2(n) < t(2, n) < 5.512 \log_2(n)$. This thesis focuses on generalizing d -CFF using a graph and hypergraph where vertices correspond to subsets in the set system. The main contributions of this thesis are in three main topics.

First, we focus on generalizing 1-CFF and 2-CFF using a graph $G = ([1, n], E)$ where a subset in the family corresponds to a vertex of G . A G -Sperner(t, n) is a family of subsets of a t -set such that each edge of G specifies a pair of subsets that must not be contained in each other, while a G -CFF(t, n) is a family of subsets of a t -set such that it is G -Sperner and the union of each pair of subsets corresponding to an edge of G does not contain any other subset in the family. Let $t_s(G)$ and $t(G)$ denote the minimum t for which there exist a G -Sperner(t, n) and a G -CFF(t, n), respectively. In this way, $t_s(K_n) = t(1, n)$ and $t(K_n) = t(2, n)$. Firstly, we prove $t_s(G) = t(1, \chi(G))$ for any simple graph G with no isolated vertices, and provide various upper and lower bounds for $t(G)$. The *trivial bound*, $t(1, n) \leq t(G) \leq t(2, n)$ holds for any simple graph G with no isolated vertex, with the lower bound tight for an infinite family of star graphs and the upper bound tight for complete graphs. We study when these bounds can be improved and give better constructive upper bounds for families of graphs such as stars, paths, cycles, wheels, and windmill graphs. In particular, a construction based on mixed-radix Gray codes yields $\log_2(n) \leq t(P_n) \leq t(C_n) \leq 1.893 \log_2(n) + \mathcal{O}(1)$ where P_n and C_n are paths and cycles with n vertices.

Second, we study a generalization of d -CFF for $d \geq 2$ using a hypergraph H of rank d and n vertices. An H -CFF(t, n) is a family of n subsets of a t -set such that the union of the subsets corresponding to the vertices in a subset of a hyperedge does not contain any other subset in the family. Let $t(H)$ denote the minimum t for which there exists an H -CFF(t, n). In this way, $t(H) = t(d, n)$ if H is a complete d -uniform hypergraph on n vertices, but can be much smaller for hypergraphs in general. We study the connection between H -CFFs and other generalized set systems used in group testing, such as H -separable set systems, investigate their characterizations through directed hypergraph homomorphisms, and explore bounds on $t(H)$ using tools from hypergraph theory. Furthermore, we study CFFs on graph and hypergraph products. We provide an upper bound for CFFs on the Cartesian product of (hyper)graphs and investigate families of graphs that attain the upper bound, as well as families for which the upper bound is close to the trivial lower bound. Moreover, we

generalize some well-known constructions of d -CFFs through the lens of CFFs on the strong product of (hyper)graphs.

Finally, we focus on the classical mixed-radix *reflected* and *modular* Gray codes by providing their recursive constructions, which are necessary for proving a result concerning cover-free families on paths and cycles. Their respective loopless algorithms can be found as Algorithm H and Exercise 77 in Section 7.2.1.1 of The Art of Computer Programming Vol. 4A by Knuth. Furthermore, we study their generation through the lens of their change sequences. We observe that both of these Gray codes, built on a mixed-radix base, are guided by the same change sequence, called the *ruler sequence*. We show how the generation of the reflected and modular Gray codes can be fully parallelized, generating each new codeword in constant time. We also show that modular Gray codes can be generated using a greedy cyclic increment approach. Furthermore, we present a new family of modular Gray codes starting from any tuple w that can be constructed using the same greedy cyclic increment approach to generate the next word that is lexicographically greater than or equal to w . We show that although this order is not a suffix of the original modular Gray code, its change sequence is a suffix of the latter.

Dedications

To all those I hold dear.

Acknowledgements

I would like to take this opportunity to thank all those who have been very supportive throughout this journey.

I owe my deepest academic gratitude to my supervisor, Prof. Lucia Moura, without whom this thesis simply would not have been possible. I truly admire her patience, cheerfulness, and, above all, her expertise. Her ability to raise the right questions during every discussion has constantly pushed me to think more carefully and deeply about my work. What I appreciate most is what she taught me about turning an initial idea into a solid piece of research: how to remain focused, how to write with attention to detail, and, during a literature review, how not to overlook even the smallest but important details. More than anything, thank you for training me to become a researcher. Thank you for everything.

I am thankful to Prof. Andrea Burgess, Prof. Mike Newman, Prof. Brett Stevens, and Prof. Mateja Šajna for serving as the examiners of my thesis and for their valuable feedback. I am particularly grateful to Andrea, Brett, and Mateja for raising thoughtful questions and suggesting ideas that greatly improved the content of the thesis, and especially to Mateja for comments that significantly improved its writing.

A special thanks to Prof. Brett Stevens as well. I am grateful for the many invitations to restaurants and for the wonderful conversations we have had. His infectious enthusiasm for so many things – culture, movies, travel, food, and much more – has always made our discussions enjoyable. These are conversations and memories that I will cherish for a long time.

It has been a privilege to work with Prof. Aaron Williams, whose expertise and insights into Gray codes and combinatorial generation made Chapter 6 of this thesis possible. Collaborating with him has been a constant learning experience, for which I will always be grateful. I will especially remember his encouragement throughout the time we worked on our paper and the many things I learned from our discussions.

I am thankful to Prof. Daniel Panario for teaching me post-quantum cryptography. I thoroughly enjoyed taking his class, and our conversations beyond the classroom have been equally enjoyable. I am also thankful to Prof. Benoit Dionne for always remembering us during Holi and Diwali, and taking the time to send his wishes. It has always meant a lot to know that we were in his thoughts, and I am grateful for his constant support.

I am very thankful to Felipe for always being kind and helpful whenever I had questions about departmental matters.

I have also been privileged to have friends who happen to be experienced researchers: Tattwamasi Amrutam, Lord Kavi, Lokanath Patra, Debasish Pattanayak, and Cameron Shock. Thank you for always being just one call away. I have learned so much from our conversations, both mathematical and otherwise, and I am grateful to have had you alongside me during this journey.

To my friends – Archi, Behnaz, César, Daniel, Ekta, Farzane, Homa, Khalil, Kianoosh, Maruša, Mary Rose, Masoomah, Mico, Mihir, Mitra, Serine, Shivam, Sophie, Utkarsh, Xiao – thank you for making my time in Ottawa so much more memorable and for making Ottawa feel a little more like home.

Finally, to my parents, who have been the emotional driving force behind me, I am deeply grateful. Thank you for making this journey seem less daunting by always providing me with the emotional support I needed.

Contents

List of Figures	xi
List of Tables	xii
1 Introduction	1
1.1 Contributions and thesis structure	3
2 Preliminaries	7
2.1 Two theorems from extremal set theory	7
2.1.1 Sperner’s theorem	7
2.1.2 Dilworth’s theorem	8
2.2 Graphs and hypergraphs	8
2.2.1 Basic graph theory	9
2.2.2 Hypergraph theory	10
2.2.3 Graph products	11
2.3 Optimal integer partitions for product	12
3 Cover-free families	15
3.1 Cover-free families	15
3.1.1 CFFs and combinatorial group testing	16
3.2 Bounds for cover-free families	18
3.2.1 Optimal 1-CFFs	18
3.2.2 Some explicit constructions of CFFs	22
3.2.3 Probabilistic and algorithmic constructions of CFFs	25
3.3 Cover-free families on hypergraphs	25
3.3.1 Structure-aware cover-free families	26
3.3.2 Some bounds for $(H, 1)$ -CFFs	27

4	Cover-free families on graphs	30
4.1	G -Sperner set system	34
4.2	Cover-free families on graphs	37
4.2.1	Establishing that $\mathbf{t}(\mathbf{G}) \sim \mathbf{t}_e(\mathbf{G})$	41
4.3	Tightness of various lower and upper bounds	43
4.4	Cover-free families on paths and cycles	48
4.4.1	Mixed-radix Gray code	48
4.4.2	Constructions of cover-free families on paths and cycles using Gray codes	50
4.5	Generalizing the construction of CFFs on star graphs	55
4.6	Values of $t(P_n)$, $t(C_n)$, and $t(W_n)$ for small n	57
4.7	Conclusion and further directions	64
5	Cover-free families on hypergraphs	65
5.1	Preliminaries	67
5.2	Connections between separable families and CFFs on hypergraphs	71
5.3	Characterizing H -CFFs through directed hypergraph homomorphisms	76
5.3.1	Directed hypergraphs	76
5.3.2	Characterization of H -CFFs	78
5.4	Constructions of H -CFFs and bounds on $t(H)$	79
5.4.1	Using transversal	81
5.4.2	Using strong vertex coloring	83
5.5	Cover-free families on products of hypergraphs	85
5.5.1	Hypergraph products	85
5.5.2	Cover-free families on the Cartesian product of hypergraphs	88
5.5.3	Cover-free families on the strong product of hypergraphs	94
5.6	Future Work	99
6	Gray codes and their generalizations	101
6.1	Preliminaries	102
6.2	Lexicographic order and Gray codes	103
6.2.1	Reflected Gray codes and the greedy increment/decrement algorithm	105
6.2.2	Modular Gray codes	108

6.3	Change sequences of reflected and modular Gray codes and their loopless generations	112
6.3.1	Loopless algorithm for binary reflected Gray codes	112
6.3.2	Loopless algorithm for mixed-radix Gray codes	114
6.4	Ranking and unranking algorithms	119
6.4.1	Ranking and unranking mixed-radix codewords	120
6.4.2	Unranking Data Structures	122
6.4.3	Loopless parallel generation of R_n^b and M_n^b	125
6.5	The ‘modified’ modular Gray code	126
6.5.1	Reinterpretation of modular Gray codes and further generalization	126
6.5.2	Loopless generation of our modified modular Gray code	128
6.6	Future work	128
Bibliography		135
Index		135

List of Figures

2.1 A hypergraph (left) and its 2-section (right)	11
3.1 Doubling construction of 2-CFFs	24
3.2 A k -strong vertex coloring construction of an $(H, 1)$ -CFF.	28
3.3 Construction of an $(H, 1)$ -CFF as described in Construction 3.3.10.	29
4.1 $t(L_{12}) = t(1, 12) = 6$, $6 \leq t(C_{12}) \leq 7$, and $t(K_{12}) = t(2, 12) = 9$	31
4.2 Sperner graph of order 3	36
4.3 A P_4 -disjunct matrix	38
4.4 An optimal $\overline{S}_9$ -disjunct matrix	45
4.5 A $K_{11} \cup K_2$ (left) and a $K_{10} \dot{\cup} K_2$ (right)	48
4.6 C_8 -CFF(6, 8) constructed from a binary reflected Gray code of length 3 (left) and the corresponding 6×8 $\overline{C}_8$ -disjunct matrix (right)	51
4.7 Examples of a C_{36} -CFF and a C_{54} -CFF	53
4.8 Examples of a P_{27} -CFF and a C_{27} -CFF	53
4.9 A C_{27} -CFF(9, 27) and a C_{19} -CFF(9, 19)	54
4.10 Examples of a friendship graph and a windmill graph	56
4.11 A $\overline{\text{Wd}}(k, n)$ -disjunct	56
4.12 Types of pair (B_u, B_v) with $\{u, v\} \in E(H)$ as described in Remark 4.6.2	58
4.13 Case 2.1 of Lemma 4.6.4	59
4.14 Case 2.2 of Lemma 4.6.4	60
4.15 Case 2.3 of Lemma 4.6.4	61
4.16 Type of the three subsets removed from 1-CFF(5, 10) as discussed in Case 4 of Lemma 4.6.4	61
4.17 Maximal graph H in Case 4.4 of Lemma 4.6.4	62

5.1 Examples of an H -disjunct, H -In-disjunct, and $\overline{H}$ -disjunct matrix for the hypergraph H provided in Example 5.1.3	68
5.2 Relationships among various types of separable and disjunct matrices in [19]	71
5.3 An H' -disjunct matrix, but not H -disjunct in Example 5.2.9	73
5.4 A 4×4 H -disjunct matrix (left) and therefore H -separable, but the matrix obtained by deleting its first row (right) is not H -separable.	75
5.5 Relationships among cover-free families and separable families on hypergraphs	76
5.6 Examples of a flower hypergraph and its 2-section	81
5.7 An $\overline{F(r, m)}$ -disjunct	82
5.8 An $\overline{(S_n \odot S_n)}$ -disjunct matrix	91
5.9 An attempt to construct a $(t(1, n - 1) + 2) \times 2n$ $\overline{(S_n \odot S_n)}$ -disjunct matrix in Corollary 5.5.17.	93
5.10 A $\overline{(G \boxtimes K_2)}$ -disjunct matrix	98
6.1 A simulation of Algorithm 1 for $n = 3$ that generates $\mathbf{R}$ corresponding to z	115

List of Tables

1.1 A 2-CFF(9, 12)	2
3.1 Incidence matrix of the 1-CFF(6, 13) in Example 3.1.3	16
3.2 Group testing using a 1-CFF(6, 13) with $t(1, 13) = 6$	18
3.3 A 4×6 1-disjunct matrix	19
3.4 Upper bounds of $t(2, n)$ for small values of n	25
4.1 Lower bounds for $t(G)$	32
4.2 Upper bounds for $t(G)$	33
4.3 Bounds for $t(G)$ for specific families of graphs G	33
4.4 Upper bounds for $t(P_n)$, $t(C_n)$, $t(W_n)$, and $t(K_n)$, where the bold entries indicate exact values.	34
6.1 The G_3 corresponds to a minimal change ordering of the subsets of $\{1, 2, 3\}$	103
6.2 L_3^b for $b = (2, 3, 3)$	104
6.3 The greedy interpretation of $R_4^{(2,3,2,2)}$	106
6.4 A simulation of Algorithm 2 for $b = (2, 3, 3)$	117
6.5 A simulation of Algorithm 3 for $b = (2, 3, 3)$	117

Chapter 1

Introduction

Cover-free families (CFFs) were first introduced by Kautz and Singleton [42] as *superimposed codes*. Since then, they have been actively studied under different names, such as *families of finite sets in which no set is covered by the union of d others* by Erdős, Frankl, and Füredi [21, 22], and *disjunct matrices* by Du and Hwang [19]. Cover-free families have several applications to combinatorial group testing (first studied by Hwang and Sós [36]), cryptographic problems, and digital communications. Also, there have been various generalizations of CFFs, for these applications [12, 26, 38, 40, 61, 69, 70, 72]. For more information, we refer the reader to the surveys [39, 72].

A d -CFF(t, n) is a set system $(\mathcal{X}, \mathcal{B})$, where $\mathcal{X}$ is a set of t points and $\mathcal{B}$ is a set of n subsets of $\mathcal{X}$ such that no subset in $\mathcal{B}$ is contained in the union of d other subsets. If $d = 1$, the set system $(\mathcal{X}, \mathcal{B})$ has the property that no subset is contained in any other, which is equivalent to a Sperner family [68]. The incidence matrix of a d -CFF(t, n) is a $t \times n$ binary matrix where the matrix restricted to any $d + 1$ columns contains a permutation submatrix of order $d + 1$ and such a matrix is also called a d -disjunct matrix. In other words, if we choose any $d + 1$ columns, the presence of a row with 0's in any d columns with a 1 in the remaining one is guaranteed by the fact that the subset corresponding to the latter has an element that does not belong to the union of the subsets corresponding to d other columns. Du and Hwang [19] study disjunct matrices and their applications to combinatorial group testing.

Combinatorial group testing is an efficient method of identifying d defective items from a set of n items with t tests where $d < t \leq n$ and this is accomplished by strategically grouping the items into t groups and testing the combined group instead. A negative test result indicates all items in the group are non-defective, while a positive result indicates that there is at least one defective item in the group. Eventually, based on the test results of t groups, it is possible to identify up to d defective items. Thus, in a traditional group testing problem GT($n, d; t$), given n and d , the goal is to find the minimum possible t to save time and resources.

A d -CFF(t, n) is used in group testing where the columns of its incidence matrix correspond to n items to be tested and the rows correspond to t tests/groups and its defining property ensures an efficient identification of up to d infected items in $\mathcal{O}(tn)$ time. This

motivates the study of finding the minimum such t , given n and d , for the existence of a d -CFF(t, n), which is denoted by $t(d, n)$. We provide further details about the application of CFFs in group testing in Section 3.1.1. For $d = 1$, the exact value of $t(1, n)$ is determined due to Sperner's theorem [68] and it is well known that $t(1, n) \sim \log_2(n)$ as $n \rightarrow \infty$ (a proof is provided in Theorem 3.2.4). For $d \geq 2$, there are lower and upper bounds on $t(d, n)$, with more details given in Section 3.2. We provide an example of a 2-CFF(9, 12) where $\mathcal{X} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and

$$\mathcal{B} = \left\{ \begin{array}{lll} \{1, 2, 3\}, & \{4, 5, 6\}, & \{7, 8, 9\}, \\ \{1, 4, 7\}, & \{2, 5, 8\}, & \{3, 6, 9\}, \\ \{1, 5, 9\}, & \{2, 6, 7\}, & \{3, 4, 8\}, \\ \{1, 6, 8\}, & \{2, 4, 9\}, & \{3, 4, 7\} \end{array} \right\}.$$

with $|\mathcal{B}| = 12$. Note that such a set system is also a Steiner triple system of order 9. The incidence matrix corresponding to the 2-CFF(9, 12) presented above is given in Table 1.1. It is proven that $t(2, 12) = 9$ [46]. Furthermore, it is easy to verify that no subset in $\mathcal{B}$ is contained in any other, which implies that this set system is also 1-CFF.

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	0	0	1	0	0	1	0	0	1	0	0
2	1	0	0	0	1	0	0	1	0	0	1	0
3	1	0	0	0	0	1	0	0	1	0	0	1
4	0	1	0	1	0	0	0	0	1	0	1	0
5	0	1	0	0	1	0	1	0	0	0	0	1
6	0	1	0	0	0	1	0	1	0	1	0	0
7	0	0	1	1	0	0	0	1	0	0	0	1
8	0	0	1	0	1	0	0	0	1	1	0	0
9	0	0	1	0	0	1	1	0	0	0	1	0

Table 1.1: A 2-CFF(9, 12)

Motivated by applications of CFFs in combinatorial group testing, Idalino and Moura [38, 40] study a generalization of d -CFFs. To demonstrate the practical relevance of these combinatorial objects, we provide an example of a real-world scenario such as the outbreak of a contagious disease in a community (for example a school, university, office). Such a community can be modeled in terms of a hypergraph where the vertices are associated with individuals (such as community members or students) and hyperedges with households or classrooms. Then, the generalized cover-free family [38] would help identify infected individuals, as long as they are contained in the union of a small number r of hyperedges. This motivates the introduction of this generalized cover-free family for group testing using a hypergraph H , denoted by (H, r) -CFF. This thesis focuses on the case of $r = 1$, that is, the case where all infected individuals are contained in an edge.

In this thesis, we study cover-free families based on graphs and hypergraphs from a perspective where each subset of the set system corresponds to a vertex, and the hyperedges specify the subsets whose union must not contain any other subset. Let $H = (V, E)$ be a

hypergraph with $|V| = n$. An H -ECFF(t_1, n) is a set system $\mathcal{F}_1 = ([1, t_1], \mathcal{B}_1)$ with $|\mathcal{B}_1| = n$ where the union of the subsets corresponding to the vertices within a hyperedge of H do not cover any other subset in $\mathcal{B}_1$. An H -ECFF allows us to determine an edge $e \in E(H)$ that contains all defective items. H -ECFFs, however, may not always allow us to identify which items within an edge are defective. An H -InCFF(t_2, n) is a set system $\mathcal{F}_2 = ([1, t_2], \mathcal{B}_2)$ with $|\mathcal{B}_2| = n$, which has the property that the union of subsets corresponding to the vertices of any non-empty subset e' of a hyperedge e does not contain any other subset in $\mathcal{B}_2$ corresponding to a vertex in $e \setminus e'$. An H -InCFF allows us to determine which items are defective, within the vertices of an edge that contains all defective items. An H -CFF is a family of subsets that is both an H -ECFF and H -InCFF. Let $\mathcal{B}'_2 = \{\{b + t_1 : b \in B\} : B \in \mathcal{B}_2\}$. Then, the set system $\mathcal{F} = ([1, t_1 + t_2], \mathcal{B}_1 \cup \mathcal{B}'_2)$ is an H -CFF($t_1 + t_2, n$). In this sense, an H -CFF(t, n) coincides with a d -CFF(t, n) when $H = K_n^d = ([1, n], \binom{[1, n]}{d})$. Such extension of cover-free families are a special case of (H, r) -ECFF, (H, r) -InCFF, and (H, r) -CFF, studied by Idalino and Moura where $r = 1$. We denote by $t_e(H)$, $t_{in}(H)$, and $t(H)$ the minimum t required for the existence of an H -ECFF(t, n), an H -InCFF(t, n), and an H -CFF(t, n), respectively. Clearly, $t(H) \leq t_e(H) + t_{in}(H)$.

Idalino and Moura provided bounds on (H, r) -CFFs which immediately provide a bound on $t(H)$. Some of their constructions are of interest in this thesis. Using a construction involving vertex coloring, Idalino and Moura [40] showed that for a hypergraph $H = ([1, n], E)$ with strong chromatic number $\chi(H)$, we have $t(H) \leq \chi(H) \log(n)$. In particular, this gives an upper bound for paths (P_n), even cycles (C_n), and trees (T_n) with $t(P_n), t(C_n), t(T_n) \leq 2 \log_2(n)$, which is interestingly smaller than the lower bound of $t(2, n)$ provided by Erdős, Frankl, and Füredi [21], namely, $3.106 \log_2(n) < t(2, n) < 5.512 \log_2(n)$. Let $H = (V, E)$ be a hypergraph with its hyperedge set E consisting of m pairwise disjoint hyperedges such that the size of each hyperedge is bounded by k . Idalino and Moura [40] showed that $t_e(H) = t(1, m)$ and $t(H) \leq t(1, m) + k$. When such H is a graph, denoted by $\mathcal{E}_{2m}$, this result implies $t_e(\mathcal{E}_{2m}) = t(1, m)$ and $t(\mathcal{E}_{2m}) \leq t(1, m) + 2$.

This thesis explores cover-free families on graphs and hypergraphs. By using tools from graph and hypergraph theory, extremal set theory, and coding theory, we provide constructive solutions to build such families for different families of (hyper)graphs and explore bounds on $t_e(H)$, $t_{in}(H)$, and $t(H)$.

1.1 Contributions and thesis structure

We summarize the main results in this thesis, which can be found in Chapters 4, 5, and 6.

1. In Chapter 4, we conduct an in-depth study of cover-free families on graphs for simple graphs G with no isolated vertices. The results discussed in this chapter are contained in a recent preprint [56]. First, we provide the exact value of $t_{in}(G)$, that is, $t_{in}(G) = t(1, \chi(G))$ where $\chi(G)$ is the chromatic number of G . As the traditional 2-CFF(t, n) is a special case of a G -CFF when $G = K_n$, such extension of 2-CFFs to graphs results in extremal families which lie between 1-CFF and 2-CFF in terms of bounds. For a graph G with n vertices, we prove $t(1, n) \leq t(G) \leq t(2, n)$. Furthermore, for a graph G

with minimum degree at least 2, we show that $t_e(G) = t(G)$, while for a graph G with minimum degree 1 and no single-edge component, we have $t(G) \leq t_e(G) + 1$. Similarly, for a graph G with minimum degree 1 and with at least one component K_2 , we obtain $t(G) \leq t_e(G) + 2$. In this way, the upper bound of $t(G)$ when $G = \mathcal{E}_{2m}$ coincides with the one provided by Idalino and Moura. In addition, we provide two infinite subfamilies of $\mathcal{E}_{2m}$ meeting the upper bound of $t(G) \leq t_e(G) + 2$. Similarly, a star graph S_n with n vertices is proven to be an infinite family of graphs meeting the upper bound of $t(G) \leq t_e(G) + 1$ as we prove $t_e(S_n) = t(1, n-1)$ and $t(S_n) = t(1, n-1) + 1 = t_e(S_n) + 1$. Furthermore, S_n with $n = \binom{x}{\lfloor \frac{x}{2} \rfloor} + 1$ is an infinite family of graphs meeting the trivial lower bound, that is, $t(S_n) = t(1, n)$. For a graph $G + \{x\}$, which is constructed by adding a universal vertex x to G , we prove $t(G + \{x\}) \leq t(G) + 1$, generalizing a well-known bound for d -CFFs, which, by plugging in $d = 2$, gives $t(2, n+1) \leq t(2, n) + 1$. S_n is also proven to be an infinite family of graphs meeting this upper bound, that is, $t(S_n + \{x\}) = t(S_n) + 1$. We further generalize our construction for CFFs on star graphs to produce CFFs on other families of graphs, in particular, windmill graphs ($\text{Wd}(k, n)$), friendship graphs ($\text{Wd}(3, n)$), and wheel graphs (W_n) and provide their respective lower and upper bounds. Moreover, we determine the exact value of $t(\text{Wd}(3, n))$ for an infinite family of friendship graphs.

Furthermore, in this chapter, a correspondence between a mixed-radix Gray code and a C_n -CFF(t, n) (also a P_n -CFF) is shown and such construction requires us to invoke a very well-known problem in combinatorial optimization (provided along with a proof in Theorem 2.3.2). This results in obtaining an asymptotic upper bound for $t(P_n)$ and $t(C_n)$ which improves the upper bound of $t(P_n) \leq t(C_n) \leq 2 \log_2(n)$ and is closer to the trivial lower bound, that is, $\log_2(n) \leq t(P_n) \leq t(C_n) \leq 1.893 \log_2(n)$. Finally, we examine the obtained upper bounds of $t(P_n)$, $t(C_n)$, and $t(W_n)$ for some small values of $n \leq 10$ and show that all of them give exact values, except possibly for $t(C_n)$ when $n = 10$, which is left as an open problem.

2. In Chapter 5, we study cover-free families on hypergraphs. We start with studying the relationship among H -CFF, H -ECFF, and generalized separable families, which we call H -separable and $\overline{H}$ -separable set systems. Du and Hwang [19] have studied separable matrices (families) for their applications in group testing and they provide relationships among disjoint and separable matrices (as displayed in Figure 5.2), which we extend to hypergraphs (Figure 5.5). Furthermore, for a hypergraph H with the size of each hyperedge bounded by r and strong chromatic number $\chi(H)$, we provide a construction of H -CFF that improves the upper bound of $t(H)$ when $r = o\left(\sqrt{\frac{\chi(H)}{\log_2(\chi(H))}}\right)$. Generalizing the construction provided for CFFs on windmill graphs in Chapter 4, a construction of cover-free families on flower hypergraphs is provided, which leads us to find an exact bound on $t(H)$ for an infinite family of flower hypergraphs H . We further explore cover-free families on graph and hypergraph products and study their bounds. In particular, for hypergraphs G and H , we have $t(G \square H) \leq t(G) + t(H)$. If $H = K_2$, then $t(G \square K_2) \leq t(G) + 2$. Interestingly, a star graph S_n with $n = \binom{x}{\lfloor \frac{x}{2} \rfloor} + 1$ is proven to be an infinite family of graphs meeting the upper bound of $t(S_n \square K_2) \leq t(S_n) + 2$. Moreover, proving this result leads us to identifying a family of sparse graphs $G \subseteq S_n \square K_2$

with minimum degree 1 and no K_2 component that also meets the upper bound of $t(G) \leq t_e(G) + 1$.

Furthermore, two well-known recursive constructions of d -CFFs involving Kronecker products (discussed in Theorems 3.2.12 and 3.2.15) are generalized to the setting of CFFs on the strong product of hypergraphs. The doubling construction of 2-CFF($t', 2n$) from a known 2-CFF(t, n) (discussed in Theorem 3.2.16) is also generalized to produce a $(G \boxtimes K_2)$ -CFF from a known G -CFF, obtaining a bound on $t(G \boxtimes K_2)$ analogous to the corresponding bound on $t(2, 2n)$.

3. In Chapter 6, we initially provide an overview on the classical binary reflected Gray codes R_n^2 , and their generalization to mixed-radix Gray codes. R_n^2 can be generated using a greedy approach (introduced by Williams [73]) and it is also well known that the position of the bit that changes at each step in the generation of R_n^2 is given by the ruler sequence [66]. This observation further leads to a loopless generation of R_n^2 (due to Bitner, Ehrlich, and Reingold [7]), where each successive tuple can be obtained in $\mathcal{O}(1)$ -time. Mixed-radix reflected Gray codes, R_n^b and mixed-radix modular Gray codes, M_n^b , are natural extensions of R_n^2 , where we use $b = (b_1, b_2, \dots, b_n)$ such that the base b_i of each digit is not necessarily binary. R_n^b and M_n^b are extensively discussed in The Art of Computer Programming, Volume 4A by Knuth [43]; their respective loopless algorithms can be found as Algorithm H and Exercise 77 in Section 7.2.1.1. We redefine R_n^b and M_n^b by providing a recursive construction for each of these objects and analyze their structure, which is crucial for proving the main result on CFF on paths and cycles (Theorem 4.4.11), provided in Chapter 4, and further provide alternative proofs for the conditions of R_n^b and M_n^b being “strongly cyclic”.

From this point onward, all presented results are from the article [51], which has recently been published in the proceedings of the 37th International Workshop in Combinatorial algorithms (IWOCa 2026), and an extended version is being prepared for submission to a journal. We discuss that both R_n^b and M_n^b built on a mixed-radix base b are guided by the same change sequence, and show that the generation of R_n^b and M_n^b can be fully parallelized with each new codeword generated in $\mathcal{O}(1)$ -time, which requires the ranking and unranking of the internal state of their corresponding loopless algorithms. We show that M_n^b can also be generated using a greedy cyclic increment approach, following the greedy approach by Williams. Furthermore, we present a new family of modular Gray codes, $M_n^b(w)$, that can be constructed starting from any tuple w , and then generating the next word using the greedy cyclic increment approach that is lexicographically greater than or equal to w . We show that although this order is not a suffix of the M_n^b , its change sequence is a suffix of the change sequence of M_n^b .

The structure of the thesis is as follows:

- Chapter 2 gives the necessary background on extremal set theory, basic graph and hypergraph theory, and the proof of Theorem 2.3.2 required to prove Theorem 4.4.11.
- Chapter 3 provides an overview on cover-free families and their applications on group testing, and bounds on $t(d, n)$. We prove some auxiliary results on $t(1, n)$ that are

crucial in the proof of some results in Chapters 4 and 5. We end this chapter by discussing (H, r) -CFFs and some of their bounds.

- Chapter 4 provides a systematic study of cover-free families on graphs and contains the materials of a recent preprint [56].
- Chapter 5 approaches cover-free families on hypergraphs from a broad perspective, with a lot of potential for future research. It contains the results discussed in item 2 of the main results listed above.
- Chapter 6 includes a reinterpretation of the classical reflected and modular Gray codes [43], and a discussion of the results from our article [51].

Each of Chapters 4, 5, and 6 starts with an introduction to the results proven in the chapter and concludes with a discussion of current ongoing work and future directions.

Chapter 2

Preliminaries

2.1 Two theorems from extremal set theory

The main objective of the problems in extremal set theory is to find an extremal set of subsets satisfying certain restricted conditions. *Sperner's theorem* [68] is one of the classic results in extremal set theory which determines the maximum size of a *Sperner family* of subsets of an n -set, which we discuss in Section 2.1.1.

For $a \in \mathbb{Z}$ and $b > a$, $[a, b]$ denotes the set $\{a, a+1, \dots, b\}$. For a set S , $\mathcal{P}(S)$ represents the *power set* of S , which is the set of all subsets of S .

We begin by defining a family of subsets and its incidence matrix.

Definition 2.1.1. A *family of subsets* (also called a *set system*) is a pair $(\mathcal{X}, \mathcal{B})$ where:

- $\mathcal{X}$ is a set, called the *ground set*, and
- $\mathcal{B} \subseteq \mathcal{P}(\mathcal{X})$ is a collection of some subsets of $\mathcal{X}$.

Definition 2.1.2. The *incidence matrix* $\mathcal{M}$ of a set system $(\mathcal{X}, \mathcal{B})$ is a binary matrix with rows corresponding to $\mathcal{X}$ and columns corresponding to $\mathcal{B}$, such that $\mathcal{M}_{ij} = 1$ if and only if $x_i \in B_j$, for all $x_i \in \mathcal{X}$ and $B_j \in \mathcal{B}$.

2.1.1 Sperner's theorem

Definition 2.1.3. A family $\mathcal{F}$ of subsets of $[1, t]$ has the *Sperner property* if no subset in $\mathcal{F}$ is contained in any other subset in $\mathcal{F}$. Such a family is called a *Sperner family* or an *antichain*.

The upper bound on the maximum size of a Sperner family on a given t -set was proved by Sperner [68]. An elegant proof of Sperner's theorem, due to Lubell, can also be found in [47].

Theorem 2.1.4. [68] If $\mathcal{F}$ is a Sperner family on $[1, t]$, then $|\mathcal{F}| \leq \binom{t}{\lfloor \frac{t}{2} \rfloor}$. The upper bound is only achieved by the following two families: the family of all $\lfloor \frac{t}{2} \rfloor$ -subsets and the family of all $\lceil \frac{t}{2} \rceil$ -subsets of the t -set.

Cover-free families (discussed in Chapter 3) generalize the concept of Sperner families.

Notation 2.1.5. We denote by $SF(t)$ the maximum size of a Sperner family on $[1, t]$, that is,

$$SF(t) = \max\{n : \exists \text{ a Sperner family } ([1, t], \mathcal{B}) \text{ with } |\mathcal{B}| = n\}.$$

By Theorem 2.1.4, $SF(t) = \binom{t}{\lfloor \frac{t}{2} \rfloor}$. Sperner's theorem has many applications to other problems in extremal set theory, poset theory, and combinatorial optimization, including constructing optimal 1-cover-free families (See Section 3.2.1) and binary covering arrays. *Dilworth's Theorem* is one of the classical results in combinatorics [17, 18], and it can be interpreted as a dual to Sperner's Theorem.

2.1.2 Dilworth's theorem

A *comparable pair* in a set $\mathcal{B}$ of subsets of $[1, t]$ is a pair $(A, B) \in \mathcal{B} \times \mathcal{B}$ such that either $A \subseteq B$ or $B \subseteq A$, that is, one set is contained in the other. A *chain* in the partially ordered set of all subsets of $[1, t]$ ordered by inclusion (the Boolean lattice $\mathcal{P}([1, t])$) is a collection of subsets in which every pair of subsets is comparable. A *chain decomposition* of the poset of $[1, t]$ is a partition of $\mathcal{P}([1, t])$ into disjoint chains.

Theorem 2.1.6 (Dilworth's Theorem). [17, 18] In any finite partially ordered set, the size of the largest antichain equals the minimum number of chains in any chain decomposition of the poset.

Thus, Sperner's theorem imposes a lower bound on the number of chains in any chain decomposition of $\mathcal{P}([1, t])$ and moreover, Sperner's theorem and Dilworth's Theorem imply the following corollary.

Corollary 2.1.7. The poset of subsets of $[1, t]$ can be decomposed into exactly $\binom{t}{\lfloor \frac{t}{2} \rfloor}$ disjoint chains.

Since each chain consists of subsets ordered by inclusion, with every pair being comparable, it follows that each chain contains exactly one subset of size $\lfloor \frac{t}{2} \rfloor$.

2.2 Graphs and hypergraphs

For a set V , we denote by $\binom{V}{m}$ the set of all subsets of size m of V .

2.2.1 Basic graph theory

A graph $G = (V(G), E(G))$ consists of a non-empty finite set $V(G)$ of elements called *vertices* and a set $E(G)$ of subsets of $V(G)$, each having size 1 or 2, called *edges*. An edge e with $|e| = 1$ is called a *loop* and an edge e with $|e| = 2$ is called a *proper edge*. An edge e is *incident* to a vertex v if $v \in e$. If G has no loops, then G is a *simple graph*. A graph G is *complete* if G is simple and $|E(G)| = \binom{|V(G)|}{2}$. A complete graph with n vertices is denoted by K_n .

A graph $G' = (V', E')$ is a *subgraph* of a graph $G = (V, E)$ if $V' \subseteq V$, and $E' \subseteq E \cap \mathcal{P}(V')$. We write $G' \subseteq G$. The subgraph of G *induced* by V' , denoted by $G[V']$, is the graph (V', E') such that $E' = E \cap \mathcal{P}(V')$. For some $I \subset V$, we denote by $G - I$ the subgraph induced by $V(G) \setminus I$, that is, $G - I = G[V \setminus I]$. A graph is *bipartite* if its vertex set V can be partitioned into two non-empty subsets U and W such that every edge has a vertex in U and a vertex in W , and it is a *complete bipartite* graph if, in addition, for every vertex x in U and y in W , there exists an edge e such that $e = \{x, y\}$. A graph with n vertices is a *path*, denoted by P_n , if its vertices can be ordered $(v_1, v_2, \dots, v_n)$ where $E(P_n) = \{\{v_i, v_{i+1}\} : i \in [1, n-1]\}$. A graph with n vertices, denoted by C_n , is a *cycle* if its vertices can be ordered $(v_1, v_2, \dots, v_n)$ where $E(C_n) = \{\{v_i, v_{i+1}\} : i \in [1, n-1]\} \cup \{\{v_1, v_n\}\}$. A graph G is *Hamiltonian* if it has a cycle subgraph C_n such that $V(G) = V(C_n)$. A graph is a *tree* if between any pair of distinct vertices there exists a unique path.

The *degree* of a vertex v in a graph G , denoted by $\deg(v)$, is the number of edges incident to v . The *minimum degree* of G , denoted by $\delta(G)$, is the smallest degree of a vertex in G , and the *maximum degree*, denoted by $\Delta(G)$, is the largest degree of a vertex in G . A vertex $v \in V(G)$ is *isolated* if $\deg(v) = 0$, otherwise it is *non-isolated*.

Definition 2.2.1. A *graph homomorphism* from G to H is a mapping $\phi : V(G) \rightarrow V(H)$ such that for every edge $\{u, v\} \in E(G)$, we have $\{\phi(u), \phi(v)\} \in E(H)$. If there exists a homomorphism from G to H , we write it as $G \rightarrow H$. If ϕ is a bijective homomorphism with $|E(G)| = |E(H)|$, then ϕ is called a *graph isomorphism* and in this case, we write $G \cong H$.

A *k-vertex coloring* is a function $f : V(G) \rightarrow \mathbb{Z}_k$ such that for every edge $\{a, b\}$, we have $f(a) \neq f(b)$. The *chromatic number* of a graph G , denoted by $\chi(G)$, is the smallest integer k such that a k -vertex coloring of G exists. A *clique* in a graph G is a subset of vertices $C \subseteq V(G)$ such that $G[C] = K_{|C|}$; a *maximum clique* is a largest such clique and its size is called the *clique number*, denoted by $\omega(G)$. A graph is related via homomorphism to the following complete graphs.

Theorem 2.2.2. [33] $K_{\omega(G)} \rightarrow G \rightarrow K_{\chi(G)}$.

A *k-edge coloring* of a graph is a mapping $f : E \rightarrow \mathbb{Z}_k$ such that for any two distinct edges e_1, e_2 incident to the same vertex, we have $f(e_1) \neq f(e_2)$. The *chromatic index* or *edge chromatic number* of G , denoted by $\chi'(G)$, is the smallest integer k such that a k -edge coloring of G exists. The following theorem due to Vizing [71] is well-known.

Theorem 2.2.3. [71] For any simple graph G , we have $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$.

A graph G is *maximal* with respect to some property X if G satisfies X and for every graph H satisfying X with $V(H) = V(G)$ and $G \subseteq H$, we have $G = H$.

2.2.2 Hypergraph theory

In this section, we discuss *hypergraphs*, which are a natural generalization of graphs where the edges do not necessarily consist of one or two vertices. A comprehensive introduction to hypergraph theory can be found in [8].

Definition 2.2.4. A *hypergraph* H is a pair $(V(H), E(H))$ where $V(H)$ is a set of elements, called *vertices*, and $E(H)$ is a set of non-empty subsets of $V(H)$, called *hyperedges* or *edges*.

A hypergraph is said to be *k-uniform* if all edges in $E(H)$ have size k . A 2-uniform hypergraph is a graph with no loops. A *complete d-uniform* hypergraph with n vertices, denoted K_n^d , is the hypergraph with n vertices and edge set consisting of all possible edges of size d , or mathematically, $E(K_n^d) = \binom{V(K_n^d)}{d}$. The *rank* of a hypergraph H , denoted by $r(H)$, is the size of the largest edge in H . A hypergraph of rank 2 is called a graph. A vertex v is said to be *incident* to a hyperedge e if $v \in e$. Two vertices u and v are said to be *adjacent* if there exists a hyperedge $e \in E$ such that $\{u, v\} \subseteq e$.

The following definitions are adapted from [3, 16].

Definition 2.2.5. Let $H = (V, E)$ be a hypergraph.

- A *subhypergraph* G of H is a pair $(V(G), E(G))$ such that $V(G) \subseteq V$ and $E(G) = \{I \cap V(G) : I \in E'', I \cap V(G) \neq \emptyset\}$ for some $E'' \subseteq E$.
- A *subhypergraph of H induced by $V' \subseteq V$* is the subhypergraph $H|_{V'} = (V', E')$ where $E' = \{I \cap V(G) : I \in E, I \cap V(G) \neq \emptyset\}$.
- A subhypergraph $G = (V', E')$ of H is called *strong* if $V' \subseteq V$ and $E' \subseteq E \cap \mathcal{P}(V')$. G is *strongly induced* if $E' = E \cap \mathcal{P}(V')$.

We are also interested in the 2-section of a hypergraph, which is a graph defined below.

Definition 2.2.6. Let $H = (V, E)$ be a hypergraph. The *2-section* of H is the simple graph $[H]_2 = (V, E')$ where $\{u, v\} \in E'$ if and only if there exists a hyperedge $e \in E$ such that $u, v \in e$ and $u \neq v$.

A hypergraph $H = (V, E)$ is *conformal* if every clique in $[H]_2$ is contained in some hyperedge of H . The hypergraph on the left of Figure 2.1 is conformal and the graph on the right of Figure 2.1 is its 2-section.

Next we present the definition of hypergraph homomorphism which is the natural extension of graph homomorphism. Some basic results presented below are proven in Raaphorst's PhD thesis [60].

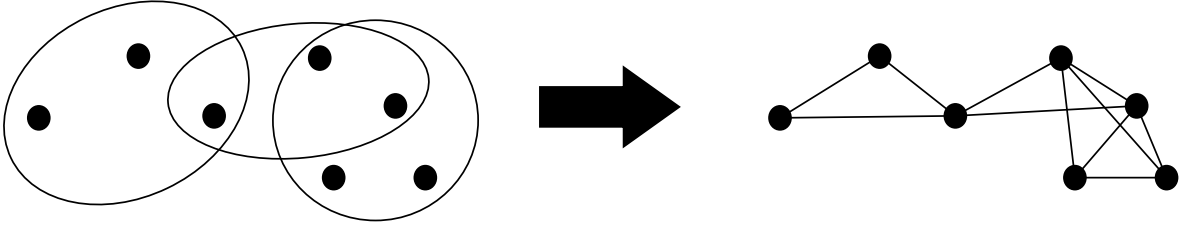


Figure 2.1: A hypergraph (left) and its 2-section (right)

Definition 2.2.7. For any two hypergraphs G and H , a *hypergraph homomorphism* ϕ is a mapping from $V(G)$ to $V(H)$ such that for any hyperedge $I = \{v_0, v_1, \dots, v_{n-1}\} \in E(G)$, we have $\phi(I) = \{\phi(v_0), \phi(v_1), \dots, \phi(v_{n-1})\} \in E(H)$ and $|I| = |\phi(I)|$. If such a homomorphism exists, then G is *homomorphic* to H , denoted by $G \rightarrow H$.

Definition 2.2.8. A *d-clique of size n* of a hypergraph $H = (V, E)$ is a set of vertices $C \subseteq V$ such that $K_n^d = (C, \binom{C}{d})$ is a strong subhypergraph of H . The size of the largest *d-clique* in H is called the *d-clique number*, denoted by $\omega_d(H)$.

Theorem 2.2.9. [60] If G and H are hypergraphs such that $G \rightarrow H$, then for any d , $\omega_d(G) \leq \omega_d(H)$.

Definition 2.2.10. A *strong c-vertex coloring* of a hypergraph H is a function $f : V(H) \rightarrow \mathbb{Z}_c$ such that for every $I \in E(H)$, we have $|f(I)| = |I|$.

The *strong chromatic number* of H is the smallest integer c such that there exists a strong c -vertex coloring for H , and it is denoted by $\chi(H)$.

Theorem 2.2.11. [60] A hypergraph H is r -uniform and has a strong c -coloring if and only if $H \rightarrow K_c^r$.

Definition 2.2.12. A *c-edge coloring* of a hypergraph $H = (V, E)$ is a mapping from E to $[1, c]$ such that no vertex is incident to more than one edge mapping to the same color.

The *edge chromatic number* is the smallest integer c such that there exists a c -edge coloring for H , and it is denoted by $\chi'(H)$.

2.2.3 Graph products

In this section, we present definitions of several types of graph products [32]. A graph product is a binary operation on the set of all finite graphs that can be constructed in several different ways. We focus on the following three well-studied standard graph products that have been extended to hypergraphs (discussed in Section 5.5.1):

1. the Cartesian product ($\square$)
2. the direct product ($\times$)

3. the strong product ($\boxtimes$)

We define the Cartesian product of graphs below.

Definition 2.2.13. (The Cartesian product of graphs) For graphs G_1 and G_2 , the Cartesian product of G_1 and G_2 (denoted by $G_1 \square G_2$) is a graph with $V(G_1 \square G_2) = V(G_1) \times V(G_2)$ and

$$E(G_1 \square G_2) = \left\{ \{(a, b_1), (a, b_2)\} : a \in V(G_1), \{b_1, b_2\} \in E(G_2) \right\} \\ \cup \left\{ \{(a_1, b), (a_2, b)\} : \{a_1, a_2\} \in E(G_1), b \in V(G_2) \right\}.$$

It is well-known that if G_1 and G_2 are Hamiltonian, then $G_1 \square G_2$ is also Hamiltonian [41]. Similarly, for graphs $G_1, G_2, \dots, G_k$, the graph $G_1 \square G_2 \square G_3 \square \dots \square G_k$ is defined by

$$V(G_1 \square G_2 \square G_3 \square \dots \square G_k) = V(G_1) \times V(G_2) \times V(G_3) \times \dots \times V(G_k)$$

and $\{(x_1, x_2, \dots, x_k), (y_1, y_2, \dots, y_k)\} \in E(G_1 \square G_2 \square G_3 \square \dots \square G_k)$ if and only if there exists $i \in [1, k]$ such that $x_i \neq y_i$ and $\{x_i, y_i\} \in E(G_i)$ and for $j \in [1, k] \setminus \{i\}$, $x_j = y_j$.

We define some families of graphs obtained by taking the Cartesian product of some well-known graphs.

Definition 2.2.14. The *Hamming graph* $H(d, n)$ is obtained by the Cartesian product of d complete graphs K_n . The *hypercube graph* Q_d is the Hamming graph $H(d, 2)$. The generalized Hamming graph $H_{a_1, a_2, \dots, a_d}$ is the graph $K_{a_1} \square K_{a_2} \square \dots \square K_{a_d}$.

Definition 2.2.15. (The direct product of graphs) For graphs G_1 and G_2 , the direct product of G_1 and G_2 (denoted by $G_1 \times G_2$) is a graph with $V(G_1 \times G_2) = V(G_1) \times V(G_2)$ and

$$E(G_1 \times G_2) = \{ \{(a_1, b_1), (a_2, b_2)\} : \{a_1, a_2\} \in E(G_1) \text{ and } \{b_1, b_2\} \in E(G_2) \}.$$

Definition 2.2.16. (The strong product of graphs) For graphs G_1 and G_2 , the strong product of G_1 and G_2 (denoted by $G_1 \boxtimes G_2$) is a graph with $V(G_1 \boxtimes G_2) = V(G_1) \times V(G_2)$ and

$$E(G_1 \boxtimes G_2) = E(G_1 \square G_2) \cup E(G_1 \times G_2).$$

It is well-known that $K_m \boxtimes K_n \cong K_{mn}$.

2.3 Optimal integer partitions for product

A *partition* of a positive integer n is a sequence of positive integers $a_1, a_2, \dots, a_l > 0$ such that $a_1 + a_2 + \dots + a_l = n$. We discuss a well-known combinatorial optimization problem: *How can a positive integer be partitioned to maximize the product of its parts?* This well-studied problem has been addressed in competitions [63, 64] and included in standard problem collections [29, 31, 45, 53] and featured in various magazines [5, 6].

Example 2.3.1. The integer 9 can be partitioned as $3 + 3 + 3$, yielding the maximum product of $3^3 = 27$. For 10, the maximum product is $3^2 \cdot 2^2 = 36$, achieved by the partition $10 = 3 + 3 + 2 + 2$. Similarly, 11 can be partitioned as $3 + 3 + 3 + 2$, giving the maximum product of $3^3 \cdot 2 = 54$.

The sequence of the maximum values of products of positive integers whose sum is equal to n appears as sequence A000792 in the online Encyclopedia of Integer Sequences [65]. We present the following theorem that answers the problem along with a proof.

Theorem 2.3.2. [65] Let $a(n)$ be the function that gives the maximum product of the parts of n , that is,

$$a(n) = \max \left\{ \prod_{i=1}^l a_i : a_1 + a_2 + \cdots + a_l = n, a_i \in \mathbb{Z}^+, l \in \mathbb{Z}^+ \right\}.$$

Then,

$$a(n) = \begin{cases} 3^k & \text{if } n = 3k, \\ 4 \cdot 3^{k-1} & \text{if } n = 3k + 1, \\ 2 \cdot 3^k & \text{if } n = 3k + 2. \end{cases}$$

Proof. Let $a_1, a_2, \dots, a_l \in \mathbb{Z}^+, l \in \mathbb{Z}^+$ such that $n = \sum_{i=1}^l a_i$ and $a(n) = \prod_{i=1}^l a_i$. Suppose, by contradiction, that $a_i \geq 5$, for some $i \in [1, l]$. Without loss of generality, suppose $a_l \geq 5$. We have

$$\begin{aligned} n &= a_1 + a_2 + \cdots + a_{l-1} + a_l \\ &= a_1 + a_2 + \cdots + a_{l-1} + (a_l - 3) + 3 \end{aligned}$$

Let $a'(n)$ be the product of the parts of n corresponding to $a_1, a_2, \dots, a_{l-1}, a_l - 3, 3$, that is,

$$a'(n) = \left(\prod_{j=1}^{l-1} a_j \right) 3(a_l - 3).$$

Since $a_l \geq 5$, we have $3(a_l - 3) > a_l$, which implies that $a'(n) > a(n)$. This is a contradiction to the fact that $a(n)$ is the maximum such product. Thus, we conclude $a_i \leq 4$.

Now, suppose that $a_i = 1$ for some i . Without loss of generality, assume that $a_l = 1$, that is, $n = a_1 + a_2 + \cdots + a_{l-2} + a_{l-1} + 1$. Let us denote $a_{l-1} + 1$ by a'_{l-1} . Then, $n = a_1 + a_2 + \cdots + a_{l-2} + a'_{l-1}$. Let $a'(n)$ be the product of the parts of n pertaining to $a_1, a_2, \dots, a_{l-2}, a'_{l-1}$, that is,

$$a'(n) = \left(\prod_{j=1}^{l-2} a_j \right) a'_{l-1}.$$

Since $a_{l-1} + 1 > a_{l-1}$ and

$$a(n) = \prod_{j=1}^l a_j = \left(\prod_{j=1}^{l-2} a_j \right) a_{l-1} \cdot 1,$$

this implies that $a'(n) > a(n)$, which is a contradiction. Therefore, we must have $2 \leq a_i \leq 4$.

Suppose $a_i = 4$ for some $i \in [1, l]$. Without loss of generality, let $a_l = 4$. Then,

$$\begin{aligned} n &= a_1 + a_2 + \cdots + a_{l-1} + 4 \\ &= a_1 + a_2 + \cdots + a_{l-1} + 2 + 2 \end{aligned}$$

It is easy to see that substituting two 2's for 4 does not change $a(n)$. Thus, we conclude without loss of generality a maximum product occurs where $a_i \in \{2, 3\}$ for all $i \in [1, l]$.

Suppose there are more than two 2's, that is, without loss of generality $a_1 = a_2 = a_3 = 2$. Then, $n = 2+2+2+a_4+\cdots+a_l$. Replacing three 2's by two 3's, we have $n = 3+3+a_4+\cdots+a_l$. If $a'(n)$ is the product of the parts of n pertaining to 3, 3, $a_4, \dots, a_l$, then

$$a'(n) = 9 \prod_{j=4}^l a_j > 8 \prod_{j=4}^l a_j = a(n),$$

which is a contradiction. Hence, we conclude that at most two of the a_i 's are equal to 2, and all the remaining a_i 's are equal to 3.

In conclusion, the partition $n = a_1 + a_2 + \cdots + a_n$ is determined by the value of $n \bmod 3$. If $n = 3k$, then we have $l = k$, and $a_1 = a_2 = \cdots = a_k = 3$. If $n = 3k + 1$, then $l = k + 1$ and without loss of generality, we have $a_1 = a_2 = \cdots = a_{k-1} = 3$ and $a_k = a_{k+1} = 2$. If $n = 3k + 2$, then $l = k + 1$ and without loss of generality, we have $a_1 = a_2 = \cdots = a_k = 3$ and $a_{k+1} = 2$. Thus, we have the desired result. $\blacksquare$

Chapter 3

Cover-free families

This chapter consists of a brief overview of cover-free families, their generalizations, along with the bounds that are of interest in our study of cover-free families on graphs and hypergraphs. To demonstrate the relevance of these objects in real-world scenarios, we also provide an introduction to combinatorial group testing, along with the application of cover-free families in this area.

3.1 Cover-free families

Definition 3.1.1. For positive integers $d < t \leq n$, a set system $\mathcal{F} = (\mathcal{X}, \mathcal{B})$ with $|\mathcal{X}| = t$ and $|\mathcal{B}| = n$ is a *d-cover-free family*, denoted by $d\text{-CFF}(t, n)$, if for any $d + 1$ subsets $B_{i_0}, B_{i_1}, \dots, B_{i_d} \in \mathcal{B}$, we have

$$B_{i_0} \not\subseteq B_{i_1} \cup B_{i_2} \cup \dots \cup B_{i_d}. \quad (3.1.1)$$

Equation 3.1.1 is equivalent to requiring that the set difference of B_{i_0} with the union of d others will contain at least 1 element, that is,

$$|B_{i_0} \setminus (\cup_{j=1}^d B_{i_j})| \geq 1.$$

The following proposition provides an important observation about a $d\text{-CFF}(t, n)$.

Proposition 3.1.2. Let $d < t \leq n$ be positive integers, and let $\mathcal{M}$ be a $t \times n$ binary matrix. Then, $\mathcal{M}$ is the incidence matrix of a $d\text{-CFF}(t, n)$ if and only if any $t \times (d + 1)$ submatrix contains a permutation submatrix of order $d + 1$.

Proof. According to Definition 3.1.1, $\mathcal{M}$ is a $d\text{-CFF}(t, n)$ if and only if for any $d + 1$ subsets $B_{i_0}, B_{i_1}, \dots, B_{i_d} \in \mathcal{B}$, we have $B_{i_0} \not\subseteq B_{i_1} \cup B_{i_2} \cup \dots \cup B_{i_d}$. This is true if and only if for any $d + 1$ columns of its incidence matrix and any $l \in \{0, 1, \dots, d\}$, we have $B_{i_l} \not\subseteq \bigcup_{\substack{j=0 \\ j \neq l}}^d B_{i_j}$, which is equivalent to saying that any $t \times (d + 1)$ submatrix of its incidence matrix contains a permutation submatrix of order $d + 1$. ■

An example of a 1-CFF(6, 13) as the set system and the corresponding incidence matrix is given in Table 3.1.

Example 3.1.3.

$$\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$$

$$\mathcal{B} = \left\{ \begin{array}{llll} \{1, 2, 3\}, & \{1, 2, 4\}, & \{1, 2, 5\}, & \{1, 2, 6\}, \\ \{1, 3, 4\}, & \{1, 3, 5\}, & \{1, 3, 6\}, & \{2, 4, 6\}, \\ \{2, 5, 6\}, & \{3, 4, 5\}, & \{3, 4, 6\}, & \{3, 5, 6\}, \\ \{4, 5, 6\} \end{array} \right\}$$

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	1	1	1	1	1	1	0	0	0	0	0	0
2	1	1	1	1	0	0	0	1	1	0	0	0	0
3	1	0	0	0	1	1	1	0	0	1	1	1	0
4	0	1	0	0	1	0	0	1	0	1	1	0	1
5	0	0	1	0	0	1	0	0	1	1	0	1	1
6	0	0	0	1	0	0	1	1	1	0	1	1	1

Table 3.1: Incidence matrix of the 1-CFF(6, 13) in Example 3.1.3

Notation 3.1.4. We denote by $t(d, n)$ the minimum possible value t for which there exists a d -CFF(t, n), that is,

$$t(d, n) = \min\{t : \exists \text{ a } d\text{-CFF}(t, n)\}.$$

When $d = 1$, then a 1-CFF(t, n) is also called a *Sperner family*, which is a set system where sets do not contain one another.

We say that a d -CFF(t, n) with $t = t(d, n)$ is *optimal*. As a consequence of Sperner's Theorem, we get $t(1, n) = \min\left\{t : \binom{t}{\lfloor \frac{t}{2} \rfloor} \geq n\right\}$ and in Section 3.2.1 we provide a proof of the well-known fact (Theorem 3.2.4) that $t(1, n) \sim \log_2 n$ as $n \rightarrow \infty$.

3.1.1 CFFs and combinatorial group testing

Combinatorial group testing (CGT) is a tool in discrete mathematics for identifying defective items from a larger pool through tests conducted on groups of items. Traditionally, a group testing problem consists of a set of n items combined into t groups, which are tested together to identify a subset of d defective items with $d < n$. Each test among the t tests examines a subset of the n items and provides a negative result if and only if all items in the subset are non-defective. It gives a positive result if the corresponding subset contains at least one defective item. This can be represented using a $t \times n$ binary matrix M where each row i corresponds to a test t_i and each column corresponds to an item j to be tested, and $M[i, j] = 1$ if and only if $j \in t_i$. The main focus of CGT is to determine the minimum

number of tests (or the number of rows in M) t , for given n and d , while identifying all defective items.

Generally, group testing can be constructed in two ways, *adaptive* or *non-adaptive*. In adaptive CGT, each test is chosen based on the result of previous tests. In non-adaptive CGT, each test is fixed in advance and all the tests can be run in parallel based on a pre-determined strategy without knowing the outcomes of previously run tests. For a $t \times n$ binary matrix M to solve a group testing problem, M must be d -separable under the assumption of exactly d defective items, or $\bar{d}$ -separable under the assumption of at most d defective items.

The *Boolean sum* of two binary vectors is the resultant vector obtained by taking their coordinate-wise OR, that is, an entry in the resultant vector is 1 if and only if at least one of the corresponding entries in the two original vectors is 1. For two binary vectors A and B , we say that A *covers* B , denoted by $A \supseteq B$, if every position where B has a 1, A also has a 1. We provide the definition of d -separable and $\bar{d}$ -separable next.

Definition 3.1.5. A binary $t \times n$ matrix $\mathcal{M}$ is called *d -separable* if the Boolean sum of any d columns are all distinct. $\mathcal{M}$ is called *$\bar{d}$ -separable* if the Boolean sum of any up to d columns are all distinct.

CFFs are used for non-adaptive group testing where rows of the corresponding incidence matrix refer to the tests and the columns to the items to be tested. Since a d -CFF guarantees the appearance of a non-defective item in one of the negative tests (because every non-defective item must appear in a test that avoids up to d defective items), all defective items can be identified, after excluding all non-defective ones by using this property of the CFF. The goal of combinatorial group testing using a d -CFF(t, n) is to identify up to d defective items out of a set of n items, grouped into t tests such that t is minimum for any given n and d , that is, to minimize $t(d, n)$.

Example 3.1.6. Table 3.2 gives an example of group testing using a 1-CFF, where candidates numbered $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$ are tested. The test pools are as follows.

- Test 1 includes the candidates 1, 2, 3, 4, 5, 6, and 7.
- Test 2 includes 1, 2, 3, 4, 8, and 9.
- Test 3 includes 1, 5, 6, 7, 10, 11, and 12.
- Test 4 includes 2, 5, 8, 10, 11, and 13.
- Test 5 includes 3, 6, 9, 10, 12, and 13.
- Test 6 includes 4, 7, 8, 9, 11, 12, and 13.

By eliminating the items from the tests with negative results, we conclude that 7 is the defective item.

Test 1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	Positive
Test 2	1	1	1	1	0	0	0	1	1	0	0	0	0	0	Negative
Test 3	1	0	0	0	1	1	1	0	0	1	1	1	1	0	Positive
Test 4	0	1	0	0	1	0	0	1	0	1	1	0	1	1	Negative
Test 5	0	0	1	0	0	1	0	0	1	1	0	1	1	1	Negative
Test 6	0	0	0	1	0	0	1	1	1	0	1	1	1	1	Positive

Table 3.2: Group testing using a 1-CFF(6, 13) with $t(1, 13) = 6$.

Group testing has been extensively studied by Du and Hwang in [19], where they define *disjunct* matrices. A binary $t \times n$ matrix $\mathcal{M}$ is called a $\bar{d}$ -*disjunct* matrix if the Boolean sum of any up to d columns does not cover any other column. Similarly, a binary $t \times n$ matrix $\mathcal{M}$ is called a d -*disjunct* matrix if the Boolean sum of any d columns does not cover any other column. This definition also implies that the Boolean sum of up to any d columns does not cover any other column, which tells us that a matrix is d -disjunct if and only if it is $\bar{d}$ -disjunct. By definition and Proposition 3.1.2, a $t \times n$ d -disjunct matrix is equivalent to the incidence matrix of a d -CFF(t, n).

A discussion of the relationships among d -disjunct, $\bar{d}$ -disjunct, d -separable, and $\bar{d}$ -separable families, studied by Du and Hwang [19], is presented in Section 5.2.

3.2 Bounds for cover-free families

In this section, we discuss bounds for cover-free families that arise from both direct and algorithmic constructions. An optimal 1-CFF is determined by Sperner's theorem and we discuss more about it in Section 3.2.1. However, for $d \geq 2$, we rely on upper and lower bounds on $t(d, n)$. We discuss probabilistic bounds for CFFs in Section 3.2.3. There are several direct constructions of CFFs in the literature, particularly those based on finite fields, codes, and other combinatorial objects. We present a list of the constructions that are relevant to this thesis in Section 3.2.2. Some of these constructions are generalized in the next chapters to our setting of CFFs on graphs and hypergraphs. For further details on constructions of CFFs, the reader is referred to the surveys [39, 72].

3.2.1 Optimal 1-CFFs

Since a Sperner family is equivalent to a 1-CFF, by Notation 3.1.4,

$$\begin{aligned} t(1, n) &= \min\{t : \exists \text{ a 1-CFF}(t, n)\} \\ &= \min\{t : \exists \text{ a Sperner family } ([1, t], \mathcal{B}) \text{ with } |\mathcal{B}| = n\}. \end{aligned}$$

Thus, Sperner's theorem [68] gives an optimal construction for a 1-CFF which can be done by taking n distinct sets out of all $\lfloor \frac{t}{2} \rfloor$ -subsets (or all $\lceil \frac{t}{2} \rceil$ -subsets) of a t -set for the smallest possible t .

Theorem 3.2.1. [39] For any $n \geq 2$,

$$t(1, n) = \min \left\{ t : \binom{t}{\lfloor \frac{t}{2} \rfloor} \geq n \right\}. \quad (3.2.1)$$

Equation 3.2.1 further implies that $t(1, n) = k$ if and only if $n \in \left[\binom{k-1}{\lfloor \frac{k-1}{2} \rfloor} + 1, \binom{k}{\lfloor \frac{k}{2} \rfloor} \right]$.

Example 3.2.2. For $n = 6$ and $d = 1$, we have $t(1, 6) = t = 4$, and every subset has size $\lfloor \frac{t}{2} \rfloor = 2$. Hence, for $\mathcal{X} = \{1, 2, 3, 4\}$ and

$$\mathcal{B} = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\},$$

$\mathcal{B}$ forms a Sperner system with the maximum possible cardinality 6. Table 3.3 presents the incidence matrix of an optimal 4×6 1-disjunct matrix.

	1	2	3	4	5	6
1	1	1	1	0	0	0
2	1	0	0	1	1	0
3	0	1	0	1	0	1
4	0	0	1	0	1	1

Table 3.3: A 4×6 1-disjunct matrix

We denote the entry in the i^{th} row and j^{th} column of a matrix A by $A[i, j]$.

Proposition 3.2.3. For an odd t , let $\mathcal{F}_1 = ([1, t], \mathcal{B}_1)$ and $\mathcal{F}_2 = ([1, t], \mathcal{B}_2)$ be the 1-CFF(t, n) constructed by taking n distinct sets out of all $\lfloor \frac{t}{2} \rfloor$ -subsets and all $\lceil \frac{t}{2} \rceil$ -subsets of a t -set, respectively. Let A_1 and A_2 be incidence matrices of $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively. Then,

1. For any two columns $i, j \in [1, n]$, there exists a row $x \in [1, t]$ such that $A_1[x, i] = A_1[x, j] = 0$.
2. For any two columns $i, j \in [1, n]$, there exists a row $x \in [1, t]$ such that $A_2[x, i] = A_2[x, j] = 1$.

Proof.

1. For any $B_1, B_2 \in \mathcal{B}_1$, since $|B_1 \cup B_2| \leq |B_1| + |B_2| \leq \lfloor \frac{t}{2} \rfloor + \lfloor \frac{t}{2} \rfloor < t$, it follows that there exists an element $x \in [1, t]$ such that $x \notin B_1 \cup B_2$. This proves item 1.
2. Suppose that there exist two subsets $B_1, B_2 \in \mathcal{B}_2$ such that $B_1 \cap B_2 = \emptyset$. Then, $|B_1 \cup B_2| = |B_1| + |B_2| = \lceil \frac{t}{2} \rceil + \lceil \frac{t}{2} \rceil > t$, leading to a contradiction. Thus, every pair of sets in $\mathcal{B}_2$ intersect, implying item 2.

■

Theorem 3.2.4. [39] $t(1, n) \sim \log_2 n$.

Proof. Consider an optimal 1-CFF(t, n). By Theorem 3.2.1, t is the smallest integer such that $\binom{t}{\lfloor \frac{t}{2} \rfloor} \geq n$. This also implies that $\binom{t}{\lceil \frac{t}{2} \rceil} \geq n$. Therefore, we have

$$\binom{t-1}{\lceil \frac{t-1}{2} \rceil} < n \leq \binom{t}{\lceil \frac{t}{2} \rceil}. \quad (3.2.2)$$

Since $t \pm 1 \sim t$, as $t \rightarrow \infty$, we may assume without loss of generality that t is even. Thus, Inequality 3.2.2 can be re-written as

$$\binom{t-1}{\frac{t}{2}} < n \leq \binom{t}{\frac{t}{2}}.$$

Using $\binom{t}{\frac{t}{2}} = 2\binom{t-1}{\frac{t}{2}}$, we have

$$\frac{1}{2} \binom{t}{\frac{t}{2}} < n \leq \binom{t}{\frac{t}{2}}.$$

The lower and upper bounds on n are of the form

$$c_i \frac{t!}{\left(\left(\frac{t}{2}\right)!\right)^2},$$

where $c_1 = \frac{1}{2}$ and $c_2 = 1$, respectively. Then, for an even $t \rightarrow \infty$, using Stirling's approximation of factorials [23], we get:

$$c_i \frac{t!}{\left(\left(\frac{t}{2}\right)!\right)^2} \sim c_i \frac{\sqrt{2\pi t} \left(\frac{t}{e}\right)^t}{\pi t \left(\frac{t}{2e}\right)^t} = c_i \frac{2^t}{\sqrt{\frac{\pi t}{2}}}.$$

Taking logarithm (base 2) for the main inequalities, we obtain as $t \rightarrow \infty$:

$$t - \log_2 \sqrt{t} + \log_2 \frac{c_1}{\sqrt{\frac{\pi}{2}}} \leq \log_2 n \leq t - \log_2 \sqrt{t} + \log_2 \frac{c_2}{\sqrt{\frac{\pi}{2}}}.$$

Simplifying lower and upper bounds on $\log_2 n$, we get

$$t - \log_2 \sqrt{t} + \log_2 \frac{c_i}{\sqrt{\frac{\pi}{2}}} = t \left(1 - \frac{\log_2 \sqrt{t}}{t} + \frac{\log_2 \frac{c_i}{\sqrt{\frac{\pi}{2}}}}{t} \right) = t(1 + o(1)).$$

Hence, $\log_2 n \sim t$ as $t \rightarrow \infty$. Also, $n \rightarrow \infty$ as $t \rightarrow \infty$ since $\binom{t-1}{\frac{t}{2}} < n \leq \binom{t}{\frac{t}{2}}$. Thus, we conclude that $t \sim \log_2 n$ as $n \rightarrow \infty$. ■

The following technical lemmas will be used later to prove some of the results in Chapters 4 and 5.

Proposition 3.2.5. For $k \geq 2$, we have $t\left(1, \left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor\right) = 2k$.

Proof. Since $\binom{2k+1}{k} = \binom{2k+1}{k+1} = \frac{2k+1}{k+1} \binom{2k}{k} < 2 \binom{2k}{k}$, we have $\left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor \leq \frac{1}{2} \binom{2k+1}{k} < \binom{2k}{k}$. This implies $t\left(1, \left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor\right) \leq 2k$. It remains to show that $t\left(1, \left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor\right) > 2k - 1$. Since $\binom{2k+1}{k} = \binom{2k}{k} + \binom{2k}{k-1}$ and $\binom{2k}{k} = \frac{2k}{k} \binom{2k-1}{k-1} = 2 \binom{2k-1}{k-1}$, we have $\binom{2k+1}{k} = 2 \binom{2k-1}{k-1} + \binom{2k}{k-1}$. In addition, since $k \geq 2$, this implies $2k > k - 1$ and $\binom{2k}{k-1} > \binom{2k}{1} \geq 4$. Thus, $\left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor = \left\lfloor \frac{1}{2} \left(2 \binom{2k-1}{k-1} + \binom{2k}{k-1} \right) \right\rfloor = \left\lfloor \binom{2k-1}{k-1} + \frac{1}{2} \binom{2k}{k-1} \right\rfloor = \binom{2k-1}{k-1} + \left\lfloor \frac{1}{2} \binom{2k}{k-1} \right\rfloor$. Since $\frac{1}{2} \binom{2k}{k-1} \geq 2$, we have $\left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor \geq \binom{2k-1}{k-1} + 2$. Thus, we conclude that, by Equation 3.2.1, $t\left(1, \left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor\right) > 2k - 1$. ■

Lemma 3.2.6. If $t(1, n) = 2k + 1$ for some $k \geq 1$, then $t(1, 2n) = 2k + 2$.

Proof. If $t(1, n) = 2k + 1$, then by Equation 3.2.1, $\binom{2k}{k} < n \leq \binom{2k+1}{k}$. This implies $2n \leq 2 \binom{2k+1}{k} = \frac{2k+2}{k+1} \binom{2k+1}{k} = \binom{2k+2}{k+1}$. Since $n \geq \binom{2k}{k} + 1$, we have $2n \geq 2 \binom{2k}{k} + 2 > \binom{2k}{k} + \binom{2k}{k-1} + 2 = \binom{2k+1}{k} + 2$. Thus, $\binom{2k+1}{k} < 2n \leq \binom{2k+2}{k+1}$. This implies $t(1, 2n) = 2k + 2$. ■

Lemma 3.2.6 also implies that if $SF(k) = n$ with k odd, then $SF(k + 1) = 2n$.

Lemma 3.2.7. Suppose $t(1, n) = 2k$ for $k \geq 2$.

1. If $n \in \left[\binom{2k-1}{k} + 1, \left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor \right]$, then $t(1, 2n) = t(1, n) + 1$.
2. If $n \in \left[\left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor + 1, \binom{2k}{k} \right]$, then $t(1, 2n) = t(1, n) + 2$.

Proof. If $t(1, n) = 2k$, then by Equation 3.2.1, $n \in \left[\binom{2k-1}{k-1} + 1, \binom{2k}{k} \right]$. This implies $2 \binom{2k-1}{k-1} + 2 \leq 2n \leq 2 \binom{2k}{k}$. Thus, $2n \geq 2 \binom{2k-1}{k-1} + 2 = \frac{2k}{k} \binom{2k-1}{k-1} + 2 = \binom{2k}{k} + 2$. Hence, $t(1, 2n) > 2k$. Now we provide a proof for items 1 and 2.

1. It is enough to show that $t\left(1, 2 \left(\binom{2k-1}{k} + 1 \right)\right) = 2k + 1$ and $t\left(1, 2 \left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor\right) = 2k + 1$. First, $t\left(1, 2 \left(\binom{2k-1}{k} + 1 \right)\right) = 2k + 1$ since $2 \binom{2k-1}{k} + 2 > 2 \binom{2k-1}{k} + 1 = \binom{2k}{k} + 1$. Let $x = 1$ if $\binom{2k+1}{k}$ is odd and $x = 0$ if $\binom{2k+1}{k}$ is even. Then, $\left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor = \frac{\binom{2k+1}{k} - x}{2}$. Thus, $t\left(1, 2 \left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor\right) = t\left(1, \binom{2k+1}{k} - x\right)$ if $\binom{2k+1}{k}$ is odd and $t\left(1, 2 \left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor\right) = t\left(1, \binom{2k+1}{k}\right)$ if $\binom{2k+1}{k}$ is even. In both cases, it follows by Equation 3.2.1 that $t\left(1, 2 \left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor\right) = 2k + 1$.
2. It is enough to show that $2k + 2 \leq t\left(1, 2 \left(\left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor + 1 \right)\right) \leq t\left(1, 2 \binom{2k}{k}\right) = 2k + 2$. First, $\left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor + 1 = \frac{\binom{2k+1}{k} - 1}{2} + 1$ if $\binom{2k+1}{k}$ is odd and $\left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor + 1 = \frac{1}{2} \binom{2k+1}{k} + 1$ if $\binom{2k+1}{k}$ is even. Then, $t\left(1, 2 \left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor + 2\right) = t\left(1, \binom{2k+1}{k} + 1\right)$ if $\binom{2k+1}{k}$ is odd and $t\left(1, 2 \left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor + 2\right) = t\left(1, \binom{2k+1}{k} + 2\right)$ if $\binom{2k+1}{k}$ is even. In both cases, it follows by

Equation 3.2.1 that $t\left(1, 2\left\lfloor \frac{1}{2}\binom{2k+1}{k} \right\rfloor + 2\right) \geq 2k + 2$. Also, $t\left(1, 2\binom{2k}{k}\right) = 2k + 2$ as $\binom{2k+2}{k+1} = 2\binom{2k+1}{k} = 2\binom{2k+1}{k+1} > 2\binom{2k}{k} > \frac{2k+1}{k+1}\binom{2k}{k} = \binom{2k+1}{k+1}$.

■

Lemmas 3.2.6 and 3.2.7 can be summarized as follows.

Proposition 3.2.8. For $n \geq 2$, let $\bar{n} = \min\left\{\left(\left\lfloor \frac{x}{2} \right\rfloor\right) : x \in \mathbb{N}, \left(\left\lfloor \frac{x}{2} \right\rfloor\right) \geq n\right\}$. If $n \leq \left\lfloor \frac{1}{2}(\bar{n} + 1) \right\rfloor$, then $t(1, 2n) = t(1, n) + 1$. Otherwise, $t(1, 2n) = t(1, n) + 2$.

Proof. Let $x = t(1, n)$. Then, $n \in \left[\left(\left\lfloor \frac{x-1}{2} \right\rfloor\right) + 1, \left(\left\lfloor \frac{x}{2} \right\rfloor\right)\right]$, and $\bar{n} = \left(\left\lfloor \frac{x}{2} \right\rfloor\right)$. This implies $\bar{n} + 1 = \left(\left\lfloor \frac{x}{2} \right\rfloor\right) + 1$, so $\bar{n} + 1 = \left(\left\lfloor \frac{x+1}{2} \right\rfloor\right)$. We can assume $n \geq 3$, since for $n = 2$, the result follows as $t(1, 4) = 4$ and $t(1, 2) = 2$. We consider two different cases depending on the parity of x .

Case 1: $x = 2k + 1$ for some $k \geq 1$. In this case, $n \leq \left(\left\lfloor \frac{x}{2} \right\rfloor\right) = \binom{2k+1}{k} = \left\lfloor \frac{1}{2} \frac{2k+2}{k+1} \binom{2k+1}{k} \right\rfloor = \left\lfloor \frac{1}{2} \binom{2k+2}{k+1} \right\rfloor = \left\lfloor \frac{1}{2} \left(\left\lfloor \frac{x+1}{2} \right\rfloor\right) \right\rfloor = \left\lfloor \frac{1}{2}(\bar{n} + 1) \right\rfloor$. In addition, since $t(1, n) = 2k + 1$, by Lemma 3.2.6, we obtain $t(1, 2n) = t(1, n) + 1$.

Case 2: $x = 2k$ for some $k \geq 2$. In this case, $\left\lfloor \frac{1}{2}(\bar{n} + 1) \right\rfloor = \left\lfloor \frac{1}{2} \left(\left\lfloor \frac{x+1}{2} \right\rfloor\right) \right\rfloor = \left\lfloor \frac{1}{2} \left(\left\lfloor \frac{2k+1}{2} \right\rfloor\right) \right\rfloor = \left\lfloor \frac{1}{2} \binom{2k+1}{k} \right\rfloor$. The result then follows by Lemma 3.2.7.

■

3.2.2 Some explicit constructions of CFFs

Some cover-free families can be constructed recursively by building them from smaller CFFs. We start with a simple construction provided by Li et al. [46] that builds a CFF from two smaller ones, as described in the following lemma. By $\mathbf{1}_{m \times n}$ and $\mathbf{0}_{m \times n}$ we denote $m \times n$ matrices with all entries as 1 and 0, respectively.

Proposition 3.2.9. [46] Let A_1 and A_2 be a $t_1 \times n_1$ d -disjunct matrix and a $t_2 \times n_2$ d -disjunct matrix, respectively. Then

$$A = \begin{bmatrix} A_1 & \mathbf{0}_{t_1 \times n_2} \\ \mathbf{0}_{t_2 \times n_1} & A_2 \end{bmatrix}$$

is a $(t_1 + t_2, n_1 + n_2)$ d -disjunct matrix.

Proof. Let the rows of A_1 and A_2 be labeled a_i , for $i \in [1, t_1]$, and b_i , for $i \in [1, t_2]$, respectively. Then the rows of A are labeled $a_1, \dots, a_{t_1}, b_1, \dots, b_{t_2}$. Let the columns of A be labeled by the elements of $[1, n_1 + n_2]$ and $I = \{i_0, i_1, \dots, i_d\}$ be a set of distinct $d + 1$ columns. If $i_l \in [1, n_1]$ for all $l \in [0, d]$, there exists a permutation submatrix of order $d + 1$ in A_1 formed by these columns. Therefore, such a submatrix can also be identified within A . A similar argument can also be used for the case where $i_l \in [n_1 + 1, n_1 + n_2]$ for all $l \in [0, d]$. Now, consider the case where $I = I_1 \dot{\cup} I_2$ with $|I_1|, |I_2| < d + 1$ and $|I_1| + |I_2| = d + 1$. Since

A_1 and A_2 are $\bar{d}$ -disjunct, the submatrices of A_1 and A_2 restricted to I_1 and I_2 , respectively, contain permutation submatrices of orders $|I_1|$ and $|I_2|$, respectively. Let the corresponding row sets be R_1 and R_2 , respectively. Then, it is easy to see that the submatrix of A restricted to the rows in $R_1 \cup R_2$ and the columns in I is a permutation submatrix of order $d + 1$. ■

A proof similar to that of Proposition 3.2.9 gives Proposition 3.2.10.

Proposition 3.2.10. Let A be a $t \times n$ d -disjunct matrix. Then

$$B = \begin{bmatrix} A & \mathbf{0}_{t \times 1} \\ \mathbf{0}_{1 \times t} & \mathbf{1}_{1 \times 1} \end{bmatrix}$$

is a $(t + 1) \times (n + 1)$ d -disjunct matrix.

In general, we have:

$$t(d, n + 1) \leq t(d, n) + 1, \quad (3.2.3)$$

since from any d -CFF(t, n), $\mathcal{F} = ([1, t], \mathcal{B})$, we can build $\mathcal{F}' = ([1, t + 1], \mathcal{B} \cup \{\{t + 1\}\})$, which is a d -CFF($t + 1, n + 1$), with incidence matrix B given in Proposition 3.2.10.

Definition 3.2.11. (Kronecker product of matrices) Let A_k be an $m_k \times n_k$ binary matrix for $k = 1, 2$ and let $\mathbf{0}$ be the $m_2 \times n_2$ matrix of all zeroes. The Kronecker product $A = A_1 \otimes A_2$ is an $m_1 m_2 \times n_1 n_2$ binary matrix with blocks $A_{i,j}$ where

$$A_{i,j} = \begin{cases} A_2, & \text{if } A_1[i, j] = 1, \\ \mathbf{0}, & \text{otherwise.} \end{cases}$$

The following theorem shows that taking the Kronecker product of two disjunct matrices results in a larger disjunct matrix. The case for $d = 2$ was proved in [46], and the result was generalized for any d in [37].

Theorem 3.2.12. ([37, 46]) For $d \geq 2$, let A_1 be a $t_1 \times n_1$ d -disjunct matrix and A_2 be a $t_2 \times n_2$ d -disjunct matrix. Then $A = A_1 \otimes A_2$ is a $t_1 t_2 \times n_1 n_2$ d -disjunct matrix. Thus, $t(d, n_1 n_2) \leq t(d, n_1) \cdot t(d, n_2)$.

Corollary 3.2.13. Let A_1 be a $t_1 \times n_1$ d_1 -disjunct matrix and A_2 be a $t_2 \times n_2$ d_2 -disjunct matrix. Then $A = A_1 \otimes A_2$ is a $t_1 t_2 \times n_1 n_2$ d -disjunct matrix where $d = \min\{d_1, d_2\}$.

Proof. Since $d \leq d_1, d_2$, we know A_1 and A_2 are d -disjunct. Thus, by Theorem 3.2.12, $A_1 \otimes A_2$ is d -disjunct. ■

Next, we present a result arising from a recursive construction that provides an improved bound on $t(d, n_1 n_2)$ whenever $s < \frac{t_1 t_2 - t_2}{t_1}$. The case $d = 2$ was proved in [46], and the result was generalized to any d in [37].

Construction 3.2.14. Let A_1, A_2 , and B be $t_1 \times n_1, t_2 \times n_2$ and $s \times n_2$ binary matrices, respectively. We construct an $(st_1 + t_2) \times n_1 n_2$ binary matrix as follows:

1. We construct an $st_1 \times n_1n_2$ matrix by computing the Kronecker product of B and A_1 . This results in a matrix having n_2 blocks of columns with n_1 columns each.
2. Then, for each $i \in [1, n_1]$, we append the i^{th} column of A_2 to the bottom of every column in the i^{th} block.

Theorem 3.2.15. (Optimized Kronecker) ([37, 46]) For $d \geq 2$, let A_1, A_2 , and B be a $t_1 \times n_1$ d -disjunct matrix, a $t_2 \times n_2$ d -disjunct matrix and an $s \times n_2$ $(d-1)$ -disjunct matrix, respectively. Then construction 3.2.14 produces an $(st_1 + t_2) \times n_1n_2$ d -disjunct matrix. Thus, $t(d, n_1n_2) \leq t(d-1, n_2) \times t(d, n_1) + t(d, n_2)$.

Next, we discuss a recursive doubling construction, which is applicable when $d = 2$.

Theorem 3.2.16. [46] Given a 2-CFF($t(2, n), n$) and a 1-CFF($t(1, n), n$), there exists a 2-CFF($t(2, n) + t(1, n) + 2, 2n$). If $t(1, n)$ is odd, then there exists a 2-CFF($t(2, n) + t(1, n) + 1, 2n$). Thus,

$$t(2, 2n) \leq \begin{cases} t(2, n) + t(1, n) + 2, & \text{if } t(1, n) \text{ is even,} \\ t(2, n) + t(1, n) + 1, & \text{if } t(1, n) \text{ is odd.} \end{cases}$$

Proof. Let A_1 be a $t(2, n) \times n$ 2-disjunct matrix. Let A_2 be the incidence matrix of a 1-CFF($t(1, n), n$) that is constructed by taking n distinct sets out of all $\lceil \frac{t}{2} \rceil$ -subsets of $[1, t]$ whereas $\overline{A_2}$ is the complement of A_2 . If $t(1, n)$ is even, then construct a $(t(2, n) + t(1, n) + 2) \times 2n$ matrix M as shown in Figure 3.1a. Otherwise, construct a $(t(2, n) + t(1, n) + 1) \times 2n$ matrix N as shown in Figure 3.1b. It can be verified that M and N are 2-disjunct, using Proposition 3.2.3 for the case of $t(1, n)$ odd.

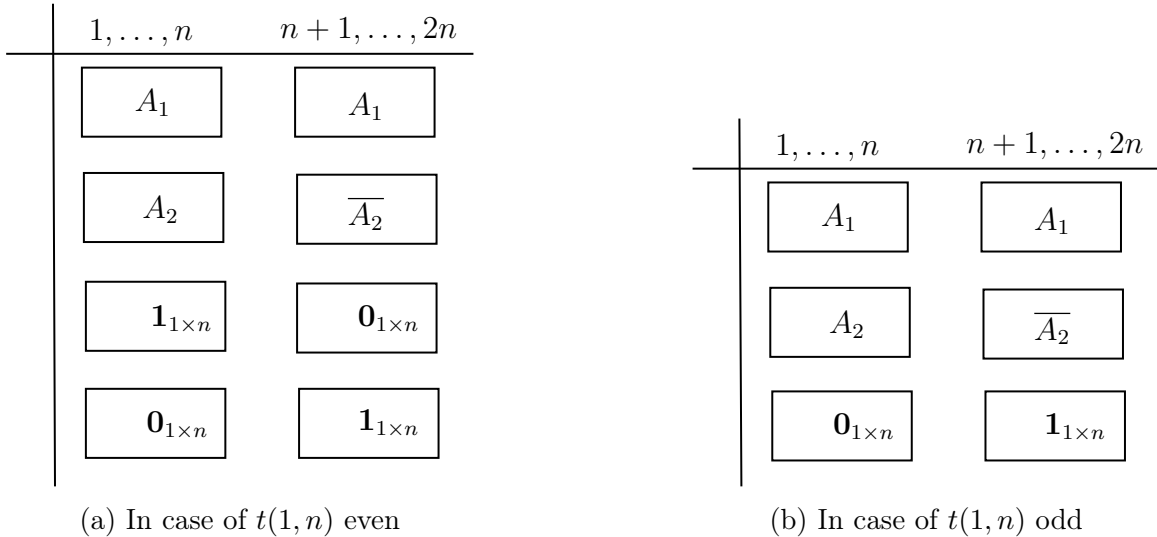


Figure 3.1: Doubling construction of 2-CFFs

■

Table 3.4 provides upper bounds on $t(2, n)$ for small values of n [39, 46] and the **bold** entries imply the bound is exact.

n	$3 \leq y \leq 8$	12	13	17	20	26	28	42	48	68	69	76	90	120	176	253
$t(2, n)$	y	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Table 3.4: Upper bounds of $t(2, n)$ for small values of n

3.2.3 Probabilistic and algorithmic constructions of CFFs

It is known that for $d \geq 2$,

$$c_1 \frac{d^2 \log_2 d}{\log_2 n} \leq t(d, n) \leq c_2 d^2 \log_2 n,$$

for some constants c_1 and c_2 . The value of this constant c_1 is proven to be approximately $\frac{1}{2}$, $\frac{1}{4}$ and $\frac{1}{8}$ in [20], [25], and [62] respectively. Furthermore, the existence of d -CFFs based on the probabilistic method produces the best known asymptotic upper bounds on $t(d, n)$, and derandomization techniques can be used to obtain polynomial time algorithms to construct a d -CFF (t, n) with $t = \Theta(d^2 \log_2 n)$ [9, 26, 58, 61]. Using a correspondence between a linear code and a CFF provided in [72], a Reed-Solomon code with an appropriate choice of parameters produces a d -CFF (t, n) with t in $O(d^2 (\frac{\log_2 n}{\log_2 d})^2)$, and its proof can be found in [4, 30].

Porat and Rothschild [58] gave the first polynomial time algorithm to construct a d -CFF (t, n) with $t = \Theta(d^2 \log_2 n)$ by showing the existence of linear codes meeting the Gilbert-Varshamov bound and then derandomizing it using the method of conditional probabilities [1] to obtain these codes. A detailed explanation of this algorithm has been provided in [14], which further shows that the constructions using Reed-Solomon codes produce d -CFFs with smaller t than those produced by Porat-Rothschild's algorithm for the values of n that are feasible in practice.

Using the constructive algorithm of the Lovász Local Lemma by Moser and Tardos [49, 50], Gargano, Rescigno, and Vaccaro [26] and Rescigno and Vaccaro [61] provided polynomial-time algorithms with $t = \Theta(d^2 \log_2 n)$. The use of the Lovász Local Lemma to find a bound for $t(d, n)$ is not new in the literature. Yeh [74] provided a bound on $t(d, n)$ while Deng et. al [15] also used Lovász local lemma to show a probabilistic existence of a generalization of CFFs.

Erdős, Frankl, and Füredi [21] provided the following lower and upper bounds on $t(2, n)$:

$$3.106 \log_2(n) < t(2, n) < 5.512 \log_2(n).$$

3.3 Cover-free families on hypergraphs

Motivated by several applications, further generalizations of cover-free families, which satisfy some additional properties, have also been studied. More details on cover-free families, their generalizations, and their practical applications can be found in the survey by Idalino and Moura [39]. In this thesis, we are interested in studying a generalization of cover-free families,

first introduced by Idalino and Moura [38, 40], which uses a hypergraph to indicate which subsets of the ground set must have the “cover-free” property.

3.3.1 Structure-aware cover-free families

Recently, a new version of the group testing problem has been considered in response to the COVID-19 pandemic, where the set of defective items tends to appear in clusters [13, 27, 54, 55]. This scenario is best described by assuming that the set of infected individuals, drawn from a pool of n items, appears in the union of at most r edges of a given hypergraph. Idalino and Moura [38, 40] studied this problem by defining *structure-aware cover-free families* (CFFs), which are applicable to contagious disease screening. In their model, edges (or hyperedges) represent groups of individuals with close contact, and the parameter r serves as an upper bound on the number of hyperedges whose union covers infected individuals.

The goal is to minimize the number t of tests given the hypergraph H and r , where the vertices of the hypergraph correspond to the columns of the incidence matrix of CFF, and hyperedges correspond to subsets of columns that are more likely to contain all infected individuals together. Idalino and Moura provided several constructions that are discussed in the next subsection, showing that the defective items can be efficiently identified using the structure of the underlying hypergraph. The definition of structure-aware CFFs is presented next.

Definition 3.3.1. [38, 40] For any positive integers n, t , let $H = ([1, n], E(H))$ be a hypergraph with n vertices and m edges. Let $\mathcal{F} = ([1, t], \mathcal{B})$ be a set system with $\mathcal{B} = \{B_1, B_2, \dots, B_n\}$ and $\mathcal{M}$ be the associated $t \times n$ binary matrix. For any $r \geq 0$, $\mathcal{F}$ is an (H, r) -CFF(t, n) if for any $l \leq r$, any l -set of hyperedges $\{I_1, \dots, I_l\} \subseteq E(H)$, any $S \subseteq \cup_{j=1}^l I_j$, and any $i_0 \in [1, n] \setminus S$, we have

$$\left| B_{i_0} \setminus \left(\bigcup_{i \in S} B_i \right) \right| \geq 1.$$

The main focus of studying structure-aware CFFs is to minimize t such that there exists an (H, r) -CFF, given $H = ([1, n], E)$ and r . Thus, we have the following notation:

$$t(r, H) = \min\{t : \exists \text{ an } (H, r)\text{-CFF}(t, n)\}.$$

Taking into account Definition 3.3.1, a d -CFF(t, n) can be written in the following ways:

1. A d -CFF(t, n) is an $(H, 1)$ -CFF(t, n) where $E(H) = \binom{[1, n]}{d}$, that is, $H = K_d^n$.
2. A d -CFF(t, n) is an (H, d) -CFF(t, n) where $E(H) = \{\{1\}, \{2\}, \dots, \{n\}\}$, that is, H is a graph with n loops.

Idalino and Moura [38, 40] also introduced a weaker version of structure-aware CFF, called *edge-identifying CFF* and denoted by (H, r) -ECFF, which can identify infected communities (or more formally, hyperedges) not necessarily being able to locate which individuals

are infected within the community. They also define an *intra-edge-screening CFF*, denoted by (H, r) -InCFF, whose property combined with (H, r) -ECFF produces an (H, r) -CFF. The definitions of (H, r) -ECFF and (H, r) -InCFF are presented below.

Definition 3.3.2. [40] Let $n, t, r, H, \mathcal{F}, \mathcal{B}$, and $\mathcal{M}$ be as in Definition 3.3.1.

1. $\mathcal{F}$ is an (H, r) -ECFF(t, n) if, for any $l \leq r$, any l -subset of hyperedges $\{I_1, \dots, I_l\} \subseteq E(H)$, and $S = \cup_{j=1}^l I_j$, and any $i_0 \in [1, n] \setminus S$, we have:

$$\left| B_{i_0} \setminus \left(\bigcup_{i \in S} B_i \right) \right| \geq 1.$$

2. $\mathcal{F}$ is an (H, r) -InCFF(t, n) if, for any r -subset of hyperedges $\{I_1, \dots, I_r\} \subseteq E(H)$, and $I \subset S = \cup_{j=1}^r I_j$, and any $i_0 \in S \setminus I$, we have:

$$\left| B_{i_0} \setminus \left(\bigcup_{i \in I} B_i \right) \right| \geq 1.$$

By Definitions 3.3.1 and 3.3.2, $\mathcal{F}$ is an (H, r) -CFF(t, n) if and only if $\mathcal{F}$ is an (H, r) -ECFF(t, n) and an (H, r) -InCFF(t, n). We use the following notation:

$$\begin{aligned} t_e(r, H) &= \min\{t : \exists \text{ a } (H, r)\text{-ECFF}(t, n)\}, \\ t_{in}(r, H) &= \min\{t : \exists \text{ a } (H, r)\text{-InCFF}(t, n)\} \end{aligned}$$

The following proposition is obtained directly, by observing that we can vertically concatenate the incidence matrices of an (H, r) -ECFF(t, n) and an (H, r) -InCFF(t, n) to obtain the incidence matrix of an (H, r) -CFF(t, n).

Proposition 3.3.3. [40] For a hypergraph H without isolated vertices,

$$t(r, H) \leq t_e(r, H) + t_{in}(r, H).$$

In this thesis, we study specifically the case of $r = 1$, that is, $(H, 1)$ -CFF(t, n), which we call CFF on hypergraphs. In Chapter 5, we focus on these objects and simplify the notation of $(H, 1)$ -CFF to H -CFF. We also write $t(H)$, $t_e(H)$, and $t_{in}(H)$ to denote $t(1, H)$, $t_e(1, H)$, and $t_{in}(1, H)$, respectively.

3.3.2 Some bounds for $(H, 1)$ -CFFs

Idalino and Moura [38, 40] provided general constructions of (H, r) -CFFs and specific constructions on some families of hypergraphs. We list their results that use edge coloring and strong vertex coloring restricted to $(H, 1)$ -CFFs, in Theorems 3.3.4 and 3.3.5, respectively.

Theorem 3.3.4. [38, 40] Let $H = (V, E)$ be a hypergraph with rank k . Then,

$$t(1, H) \leq \chi'(H) \cdot t\left(1, \left\lceil \frac{n}{k} \right\rceil\right) + k.$$

Theorem 3.3.5. [40] Let $H = (V, E)$ be a hypergraph and $\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_k$ be the sets of vertices in each color class of a k -strong vertex coloring of H . For each i , let $n_i = |\mathcal{N}_i|$ and $\sum_{i=1}^k n_i = |V| = n$. Suppose there exists a 1-CFF(t_i, n_i) for all $1 \leq i \leq k$. Then, there exists an $(H, 1)$ -CFF(t, n) with $t = \sum_{i=1}^k t_i$.

We show the construction used to prove Theorem 3.3.5 next.

Construction 3.3.6. Let $t_1, t_2, \dots, t_k, n_1, n_2, \dots, n_k$ be as in Theorem 3.3.5. Construct a matrix A consisting of blocks B_{ij} , $1 \leq i, j \leq k$, where block column j corresponds to vertices having color j . Each B_{ij} is a $t_i \times n_j$ matrix given as follows:

$$B_{i,j} = \begin{cases} 1\text{-CFF}(t_i, n_i), & \text{if } i = j \text{ and } n_i > 1 \\ \mathbf{1}_{t_i \times n_i} \text{ with } t_i = 1, & \text{if } i = j \text{ and } n_i = 1 \\ \mathbf{0}_{t_i \times n_j}, & \text{otherwise} \end{cases}$$

A diagram illustrating this construction is given in Figure 3.2.

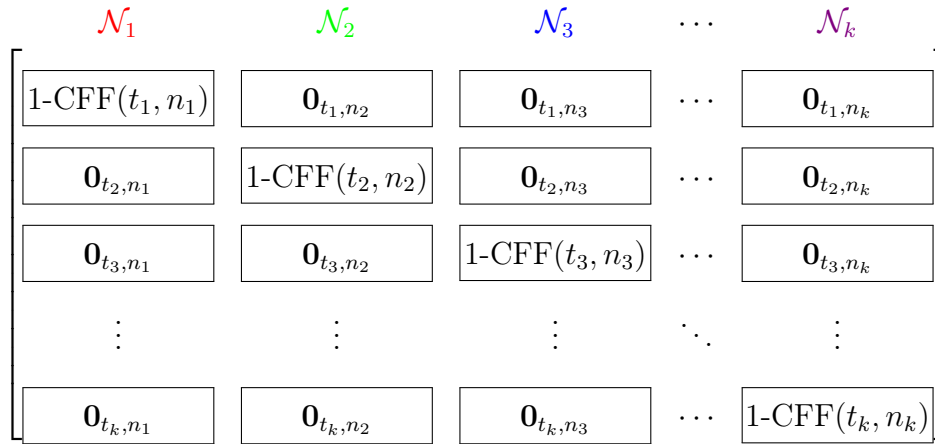


Figure 3.2: A k -strong vertex coloring construction of an $(H, 1)$ -CFF.

The following corollary is an immediate consequence of Theorem 3.3.5 when the color classes have the same size m .

Corollary 3.3.7. [40] Let $H = (V, E)$ be a hypergraph and $\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_k$ be the sets of vertices in each color class of a k -strong vertex coloring of H such that for all $i \in [1, k]$, $|\mathcal{N}_i| = m$. So, $|V| = n = mk$. Let B be a $t(1, m) \times m$ 1-disjunct matrix. Then, $I_k \otimes B$ is the incidence matrix of an $(H, 1)$ -CFF(t, n) with $t = k \cdot t(1, m)$.

Definition 3.3.8. A *matching hypergraph* $H = ([1, n], E)$ is a hypergraph without any isolated vertices in which any pair of hyperedges are disjoint. If H is 2-uniform, then H is called a *matching graph* and we denote it as $\mathcal{E}_n$, where n must be even.

The following proposition provides an exact bound on $t_e(1, H)$ and an upper bound on $t(1, H)$ for a matching hypergraph.

Proposition 3.3.9. [38, 40] Let $H = ([1, n], E)$ be a matching hypergraph with the cardinality of each hyperedge bounded by $k < n$. Then,

$$\begin{aligned} t_e(1, H) &= t(1, |E|), \\ t(1, H) &\leq t(1, |E|) + k. \end{aligned}$$

We present the construction that obtains the bounds provided in Proposition 3.3.9.

Construction 3.3.10. Let H be a hypergraph as defined in Proposition 3.3.9 with $E = \{e_1, e_2, \dots, e_m\}$. Let A be a $t(1, m) \times m$ $\bar{1}$ -disjunct matrix with the columns labeled as A^j for each $j \in [1, m]$. We construct a $t(1, m) \times n$ matrix C , and a $k \times n$ matrix D , where columns of C and D are labeled as C^i and D^i , respectively, corresponding to vertex i for each $i \in [1, n]$. Matrices C and D are constructed as follows: for a vertex $i \in [1, n]$ incident to an edge e_j with $j \in [1, m]$, we have $C^i = A^j$ and D^i is identical to the i^{th} column of the $k \times k$ identity matrix, I_k . It is easy to verify that matrix C is an $(H, 1)$ -ECFF($t(1, m), n$) and the resultant $(t(1, m) + k) \times n$ matrix by the vertical concatenation of C and D is an $(H, 1)$ -CFF($t(1, m) + k, n$).

A diagram illustrating this construction is given in Figure 3.3.

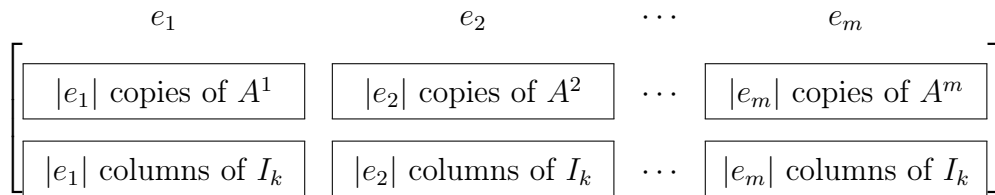


Figure 3.3: Construction of an $(H, 1)$ -CFF as described in Construction 3.3.10.

Chapter 4

Cover-free families on graphs

Before studying cover-free families on hypergraphs, we do a detailed study for the special case of graphs. This chapter consists of the contents from the article [56].

In this chapter, we study a generalized version of 1-CFFs and 2-CFFs that extends these definitions to include a graph G , where vertices of G correspond to sets in the set system. A G -Sperner(t, n) is a set system $([1, t], \mathcal{B})$ such that the sets corresponding to endpoints of an edge do not contain one another (Definition 4.1.1). In this sense, a K_n -Sperner(t, n) coincides with a 1-CFF(t, n) or a Sperner family. An *edge cover-free family* on a graph G , denoted by G -ECFF(t, n), is a set system $([1, t], \mathcal{B})$ such that for any $e \in E(G)$, the union of the sets corresponding to vertices in e does not contain any other set in the set system. A cover-free family on a graph G , denoted by G -CFF(t, n), is a set system $([1, t], \mathcal{B})$ such that for any $\emptyset \neq e' \subseteq e \in E(G)$, the union of the sets corresponding to vertices in e' does not contain any other set in the set system. Thus, a K_n -CFF(t, n) is equivalent to a 2-CFF(t, n). Furthermore, being G -CFF is equivalent to being both G -ECFF and G -Sperner. We denote by $t_s(G)$, $t_e(G)$, and $t(G)$ the minimum value of t for which there exists a G -Sperner(t, n), a G -ECFF(t, n), and a G -CFF(t, n), respectively, where $n = |V(G)|$. Clearly, $t_s(K_n) = t(1, n)$ and $t(K_n) = t(2, n)$. It is also known that $t_e(K_n) = t(K_n) = t(2, n)$ [38, 40], but in general, we may have $t_e(G) \neq t(G)$ for some G (for example, $t_e(S_n) = t(S_n) - 1$, for a star graph S_n [Corollary 4.3.4]). Let L_n be a graph with n vertices and the edge set consisting of loops on all vertices. Then, $t(L_n) = t_e(L_n) = t(1, n)$.

In this chapter, we first prove the exact value of $t_s(G)$ for a simple graph G , that is, $t_s(G) = t(1, \chi(G))$ where $\chi(G)$ is the chromatic number of G (Corollary 4.1.10). We achieve this by providing a characterization of G -Sperner families via graph homomorphisms to a special graph, which we call a *Sperner graph*.

We further show that a G -CFF(t, n) provides a richer combinatorial structure that lies between being a 1-CFF and a 2-CFF in terms of bounds, giving the trivial bounds: $t(1, n) \leq t(G) \leq t(2, n)$ (see Corollary 4.2.8), for any graph G with no isolated vertices. We exemplify this claim in Figure 4.1 for a cycle of length 12 (C_{12}), where the incidence matrix of a C_{12} -CFF(7, 12) is given. Using the known values of $t(1, 12) = 6$ and $t(2, 12) = 9$ [46], we obtain the trivial bounds $6 \leq t(C_{12}) \leq 9$, and the existence of a C_{12} -CFF(7, 12) shows that the trivial upper bound is not always tight, and that $6 \leq t(C_{12}) \leq 7$. This observation

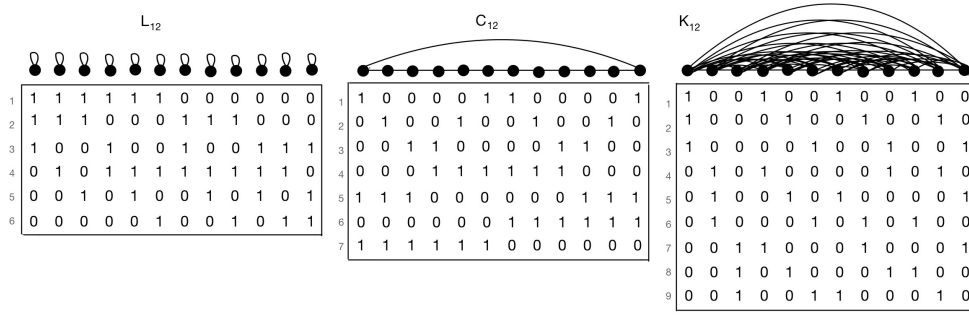


Figure 4.1: $t(L_{12}) = t(1, 12) = 6$, $6 \leq t(C_{12}) \leq 7$, and $t(K_{12}) = t(2, 12) = 9$

motivates us to study conditions under which the trivial upper bound can be improved upon and find which families of graphs meet either of these extreme bounds. We also find some families of graphs where $t(G)$ is close to the trivial lower bound.

From the above definitions and $t_s(G) = t(1, \chi(G))$, it is easy to see that $t_e(G) \leq t(G) \leq t_e(G) + t_s(G) = t_e(G) + t(1, \chi(G))$. Thus, we further investigate under which conditions $t(G)$ is close to $t_e(G)$, even when $\chi(G)$ may be large. Surprisingly, for any graph G with no isolated vertex, we show that $t_e(G) \leq t(G) \leq t_e(G) + 2$ (Theorem 4.2.11).

Tables 4.1 and 4.2 summarize various lower and upper bounds on $t(G)$ proven in this chapter, along with infinite families of graphs achieving these bounds. We assume that G is a graph with no isolated vertex and $|V(G)| = n \geq 3$. We denote by S_n the star graph with n vertices (see Definition 4.3.1), by $\mathcal{E}_n$ with n even, the graph with $\frac{n}{2}$ single edge components, by $G + \{x\}$ the graph G with the addition of a universal vertex x (see Notation 4.2.12), by $K_n \cup K_2$ the graph K_n with a pendant vertex attached, and by $K_n \dot{\cup} K_2$ the graph K_n with an edge component (see Notation 4.3.12).

Table 4.1 shows an infinite family of graphs that meets the trivial lower bound (first row) and an infinite family of graphs that does not meet the trivial lower bound (second row). It is also shown that $\delta(G) \geq 2$ is a sufficient condition for $t_e(G) = t(G)$ (third row) and that $\mathcal{E}_n$ is an infinite family of graphs not meeting the lower bound $t_e(G) \leq t(G)$ (fourth row).

Table 4.2 shows various upper bounds for $t(G)$. We show that there are two infinite subfamilies of $\mathcal{E}_n$, meeting the upper bound of $t(G) \leq t_e(G) + 2$. If $\delta(G) = 1$ and G has no connected component as K_2 , then we have a tighter upper bound $t(G) \leq t_e(G) + 1$ and S_n is an infinite family of graphs meeting this upper bound. Recall from Table 4.1 that $t(G) = t_e(G)$ whenever $\delta(G) \geq 2$. The bound $t(H + \{x\}) \leq t(H) + 1$ generalizes the bound in Equation 3.2.3 and is tight for any H with $t(H) = t(1, |V(H)|) + 1$, as well as for S_n .

Finally, we obtain bounds on $t(G)$ for specific classes of graphs G , as provided in Table 4.3. We denote by P_n , C_n , and W_n , the path, cycle, and wheel graph of n vertices, respectively. $\text{Wd}(k, n)$ refers to the windmill graph defined in Definition 4.5.1. For $G = S_n$ and $G = S_n + \{x\}$, the value of $t(G)$ is determined. Similarly, for $G = \mathcal{E}_{2m}$, $G = \text{Wd}(3, n)$, and $G = W_{n+1}$, the value of $t(G)$ is determined up to a difference of 1. For $G = \text{Wd}(k, n)$ with $k \geq 4$, upper and lower bounds are provided. For $G = C_n$, an upper bound is provided in

Lower bound for $t(G)$	Extra assumptions for G	Lower bound is tight infinitely often
$t(1, n) \leq t(G)$	—	1. S_n with $n = \binom{x}{\lfloor \frac{x}{2} \rfloor} + 1$ (Corollary 4.3.5) 2. $\mathcal{E}_{2m}$ with $m > \lfloor \frac{1}{2}(\overline{\overline{m}} + 1) \rfloor$ (Proposition 4.3.7)
$t(1, n) + 1 \leq t(G)$ (Corollary 4.2.9)	$n = \binom{x}{\lfloor \frac{x}{2} \rfloor}$	$\mathcal{E}_{\binom{x}{\lfloor \frac{x}{2} \rfloor}}$ with $\binom{x}{\lfloor \frac{x}{2} \rfloor}$ even (Theorem 4.3.8)
$t_e(G) \leq t(G)$	—	1. $\delta(G) \geq 2$ (Theorem 4.2.11 part 1) 2. $K_n \cup K_2$ with $t(2, n) = t(2, n + 1)$ (Corollary 4.3.14) 3. $K_n \dot{\cup} K_2$ with $t(2, n) = t(2, n + 2)$ (Corollary 4.3.14)
$t_e(G) + 1 \leq t(G)$ (Proposition 4.3.7)	$G = \mathcal{E}_n$ with n even	$\mathcal{E}_8^\dagger$ (Proposition 4.3.9)

[†] The bound is tight for $\mathcal{E}_8$, not known if infinitely often.

Table 4.1: Lower bounds for $t(G)$

Theorem 4.4.11 using a construction involving mixed-radix Gray codes [43]. This improves the upper bound $t(C_n) \leq 2 \log_2(n)$, which is a consequence of a construction by Idalino and Moura [40], to $t(C_n) \leq 1.893 \log_2(n) + \mathcal{O}(1)$ (see Corollary 4.4.12). It further results in having upper bounds for $t(P_n)$ and $t(C_n)$ that are close to the trivial lower bound, that is, we have $\log_2(n) \leq t(P_n) \leq t(C_n) \leq 1.893 \log_2(n) + \mathcal{O}(1)$ as n grows.

For small values of n , we are able to determine $t(P_n)$, $t(C_n)$, and $t(W_n)$, as given in Proposition 4.6.1, Theorem 4.6.5, and Corollary 4.6.8. In Table 4.4, we display the known values for $t(P_n)$, $t(C_n)$, and $t(W_n)$ or their upper bounds for $n \leq 12$. We also display the exact values of $t(K_n) = t(2, n)$, to show how far the previous values are from the trivial upper bound.

The structure of the chapter is as follows. Section 4.1 focuses on G -Sperner families and provides all the necessary technical results before proving its main result, that is, $t_s(G) = t(1, \chi(G))$. In Section 4.2, we provide formal definitions of cover-free families on graphs, along with lower and upper bounds on $t(G)$, as summarized in Tables 4.1 and 4.2. In Section 4.3, we present infinite families of graphs G for which $t(G)$ meets the extreme bounds proven in Section 4.2. In Section 4.4, we provide an introduction to mixed-radix Gray codes and

Upper bound for $t(G)$	Extra assumptions for G	Upper bound is tight infinitely often
$t(G) \leq t(2, n)$ (Corollary 4.2.8)	—	K_n
$t(G) \leq t_e(G) + 2$ (Theorem 4.2.11 part 3)	$\delta(G) = 1$	1. $\mathcal{E}_{2m}$ with $m > \lfloor \frac{1}{2}(\overline{m} + 1) \rfloor$ (Proposition 4.3.7) 2. $\mathcal{E}_n$ with $n = \binom{x}{\lfloor \frac{x}{2} \rfloor}$, n even (Theorem 4.3.8)
$t(G) \leq t_e(G) + 1$ (Theorem 4.2.11 part 2)	$\delta(G) = 1$ with no single edge components	S_n (Corollary 4.3.4)
$t(G) \leq t(H) + 1$ (Theorem 4.2.13)	$G = H + \{x\}$ with $\delta(H) \geq 1$	1. H with $t(H) = t(1, V(H)) + 1$ (Corollary 4.2.14) 2. $H = S_n$ (Corollary 4.3.6)

Table 4.2: Upper bounds for $t(G)$

G	$t(G)$
$S_n, n \geq 3$ (Corollary 4.3.4)	$t(S_n) = t(1, n - 1) + 1$
$S_n + \{x\}, n \geq 3$ (Corollary 4.2.14)	$t(S_n + \{x\}) = t(1, n - 1) + 2$
$\mathcal{E}_{2m}, m \geq 2$ (Proposition 4.3.7)	$t(1, m) + 1 \leq t(\mathcal{E}_{2m}) \leq t(1, m) + 2$ (the upper bound is tight for $m > \lfloor \frac{1}{2}(\overline{m} + 1) \rfloor$)
$P_n, C_n, n \geq 3$ (Theorem 4.4.11)	$t(P_n) \leq t(C_n) \leq \begin{cases} 3k & \text{if } n \in (2 \cdot 3^{k-1}, 3^k] \text{ for some } k \geq 1, \\ 3k + 1 & \text{if } n \in (3^k, 4 \cdot 3^{k-1}] \text{ for some } k \geq 1, \\ 3k + 2 & \text{if } n \in (4 \cdot 3^{k-1}, 2 \cdot 3^k] \text{ for some } k \geq 1. \end{cases}$
$\text{Wd}(k, n), n \geq 2, k \geq 4$ (Theorem 4.5.4)	$t(1, (k - 1)n) + 1 \leq t(\text{Wd}(k, n)) \leq t(1, n) + t(2, k - 1) + 1$
$\text{Wd}(3, n), n \geq 2$ (Corollary 4.5.5)	$t(1, n) + 2 \leq t(\text{Wd}(3, n)) \leq t(1, n) + 3$ (the upper bound is tight for $n > \lfloor \frac{1}{2}(\overline{n} + 1) \rfloor$)
$W_{n+1}, n \geq 3$ (Corollary 4.4.18)	$\max\{t(C_n), t(C_{n+1})\} \leq t(W_{n+1}) \leq t(C_n) + 1$

Table 4.3: Bounds for $t(G)$ for specific families of graphs G

describe a construction of CFFs on paths and cycles using such Gray codes, which yields the upper bound in $\log_2(n) \leq t(P_n) \leq t(C_n) \leq 1.893 \log_2(n) + \mathcal{O}(1)$. Section 4.5 generalizes the

n	$t(P_n)$	$t(C_n)$	$t(W_n)$	$t(K_n)[46]$
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	5	5	6	6
7	6	6	6	7
8	6	6	7	8
9	6	6	7	9
10	6	7	7	9
11	7	7	8	9
12	7	7	8	9

Table 4.4: Upper bounds for $t(P_n)$, $t(C_n)$, $t(W_n)$, and $t(K_n)$, where the bold entries indicate exact values.

construction of CFFs on star graphs provided in Section 4.3 to obtain bounds (tight bounds in some cases) for windmill graphs and friendship graphs. In Section 4.6, we provide the results for exact values of $t(P_n)$, $t(C_n)$, and $t(W_n)$ for small values of n , as provided in Table 4.4 (in **bold**). Conclusions and future work are provided in Section 4.7.

4.1 G -Sperner set system

Definition 4.1.1 is the definition of G -Sperner set systems for a graph G . Definition 4.1.2 is an equivalent one, using the incidence matrix of the set system.

Definition 4.1.1. (via set system) Let n, t be positive integers with $n \geq t$, and let $G = ([1, n], E(G))$ be a graph. Let $\mathcal{F} = ([1, t], \mathcal{B})$ be a set system with $\mathcal{B} = \{B_1, B_2, \dots, B_n\}$ and $|\mathcal{B}| = n$ where B_i corresponds to the vertex i . Then, $\mathcal{F}$ is *Sperner for a proper edge* $\{a, b\} \in E(G)$ if

$$|B_a \setminus B_b| \geq 1 \text{ and } |B_b \setminus B_a| \geq 1.$$

The set system $\mathcal{F}$ is *G -Sperner*, or a *G -Sperner(t, n)*, if it is Sperner for every proper edge in $E(G)$.

Definition 4.1.2. (via incidence matrix) Let $G = ([1, n], E(G))$ be a graph, and let A be a $t \times n$ binary matrix where each column corresponds to a vertex of G . We denote by $A[i, j]$ the entry of A in row i and the column corresponding to the vertex $j \in [1, n]$, and by A^j the column vector of A corresponding to the vertex j . Then, matrix A is *Sperner for a proper edge* $\{a, b\}$ if there exist row indices $l, m \in [1, t]$ such that $A[l, a] = 0$ and $A[l, b] = 1$, and $A[m, a] = 1$ and $A[m, b] = 0$. Matrix A is *G -Sperner* if it is Sperner for every proper edge in $E(G)$.

It is easy to verify the equivalence given in the following proposition.

Proposition 4.1.3. Let $G = ([1, n], E(G))$ be a graph and $\mathcal{F} = ([1, t], \mathcal{B})$ be a set system where $|\mathcal{B}| = n$. Then, $\mathcal{F}$ is a G -Sperner(t, n) if and only if its incidence matrix is G -Sperner.

We denote by $t_s(G)$ the minimum t such that there exists a G -Sperner($t, |V(G)|$) set system, that is,

$$t_s(G) = \min\{t : \exists \text{ a } G\text{-Sperner}(t, |V(G)|)\}.$$

Observe that a G -Sperner(t, n) is equivalent to a $(G, 1)$ -InCFF(t, n), given in Definition 3.3.2. Thus, $t_s(G) = t_{in}(1, G)$.

In this section, we show that $t_s(G)$ is determined from its chromatic number for any simple graph G . Our method is to characterize it using a special family of graphs, which we call a *Sperner graph*, defined in this section. We first prove a series of fundamental results before proving the main result.

For the rest of the chapter, we assume that G is a simple graph. Therefore, G has only proper edges.

Proposition 4.1.4. Let G and H be graphs such that $G \rightarrow H$. Then $t_s(G) \leq t_s(H)$.

Proof. Let $f : V(G) \rightarrow V(H)$ be a homomorphism from G to H , and A be a $t^* \times |V(H)|$ H -Sperner matrix with $t^* = t_s(H)$. Now, from A we construct a $t^* \times |V(G)|$ matrix M with every column corresponding to a vertex in G such that for every $v \in V(G)$, we set the column M^v identical to $A^{f(v)}$. Let $\{x, y\} \in E(G)$. Since homomorphisms preserve edges, $\{f(x), f(y)\} \in E(H)$. Since A is Sperner for the edge $\{f(x), f(y)\}$, then M is also Sperner for $\{x, y\}$. Thus, M is a G -Sperner matrix with t^* rows and this implies $t_s(G) \leq t^* = t_s(H)$. ■

Proposition 4.1.5. Let $n \geq 2$. Then, $t_s(K_n) = t(1, n)$.

Proof. It is easy to see that a set system being K_n -Sperner is equivalent to being Sperner, which is equivalent to being a 1-CFF. ■

Definition 4.1.6. The *Sperner graph of order z* , denoted $\mathcal{S}(z)$ is a graph (V, E) with

- vertex set $V = \{0, 1\}^z = \mathbb{Z}_2^z$, and
- edge set E such that $\{x, y\} \in E$ if and only if there exist $j, k \in [1, z]$ such that $x_j = 0, y_j = 1$ and $x_k = 1, y_k = 0$.

where x_j represents the j^{th} position of the binary z -tuple x .

From the definition above, it is also clear that the vertices of a Sperner graph of order z correspond to the elements in the power set $\mathcal{P}([1, z])$ and there is an edge between $x, y \subseteq [1, z]$ if and only if $x \not\subseteq y$ and $y \not\subseteq x$. Figure 4.2 shows the Sperner graph $\mathcal{S}(3)$.

A *chain* in the partially ordered set of all subsets of $[1, n]$ ordered by inclusion is a set of subsets in which for every pair (A, B) , either $A \subseteq B$ or $B \subseteq A$. A *chain decomposition* is a partition of $\mathcal{P}([1, n])$ into disjoint chains. The following theorem provides the chromatic number and the clique number of a Sperner graph of order z . The proof of this theorem is inspired by work on qualitative independent graphs by Meagher and Stevens [48].

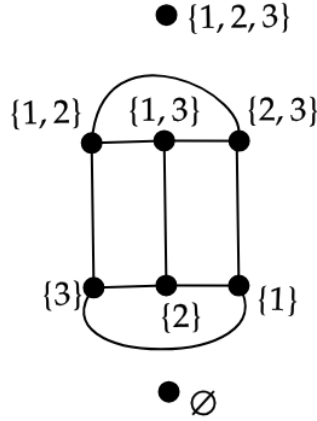


Figure 4.2: Sperner graph of order 3

Theorem 4.1.7. $\omega(\mathcal{S}(z)) = \chi(\mathcal{S}(z)) = \binom{z}{\lfloor \frac{z}{2} \rfloor}$.

Proof. By definition of $\mathcal{S}(z)$, only Sperner families on $[1, z]$ correspond precisely to the cliques in $\mathcal{S}(z)$. Hence, by Sperner's theorem, $\omega(\mathcal{S}(z)) = \binom{z}{\lfloor \frac{z}{2} \rfloor}$.

Clearly, $\chi(\mathcal{S}(z)) \geq \omega(\mathcal{S}(z)) = \binom{z}{\lfloor \frac{z}{2} \rfloor}$. By Corollary 2.1.7, we have that the poset of subsets of $[1, z]$ ordered by inclusion can be decomposed into $\binom{z}{\lfloor \frac{z}{2} \rfloor}$ disjoint chains such that each of the chains contains exactly one set of size $\lfloor \frac{z}{2} \rfloor$. Let $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{\binom{z}{\lfloor \frac{z}{2} \rfloor}}$ be such chains. By the definition of Sperner's graph, it is clear that there cannot be an edge in $\mathcal{S}(z)$ between vertices corresponding to any two subsets in a chain. Thus, we can achieve a $\binom{z}{\lfloor \frac{z}{2} \rfloor}$ -coloring of $\mathcal{S}(z)$ by assigning color i to all vertices in chain $\mathcal{C}_i$, for $1 \leq i \leq \binom{z}{\lfloor \frac{z}{2} \rfloor}$. Thus, $\chi(\mathcal{S}(z)) \leq \binom{z}{\lfloor \frac{z}{2} \rfloor}$. Hence, $\chi(\mathcal{S}(z)) = \binom{z}{\lfloor \frac{z}{2} \rfloor}$. ■

Corollary 4.1.8. $t_s(\mathcal{S}(z)) = z$.

Proof. Since $K_{\omega(\mathcal{S}(z))} \rightarrow \mathcal{S}(z) \rightarrow K_{\chi(\mathcal{S}(z))}$, by Proposition 4.1.4, we have $t_s(K_{\omega(\mathcal{S}(z))}) \leq t_s(\mathcal{S}(z)) \leq t_s(K_{\chi(\mathcal{S}(z))})$. Then, by Theorem 4.1.7, Proposition 4.1.5, and Equation 3.2.1, we have that $t_s(\mathcal{S}(z)) = t\left(1, \binom{z}{\lfloor \frac{z}{2} \rfloor}\right) = z$. ■

In the next proposition, we give a characterization of $t_s(G)$ via a homomorphism from the graph G to a Sperner graph.

Proposition 4.1.9. $t_s(G) = \min_{l \in \mathbb{N}} \{l : G \rightarrow \mathcal{S}(l)\}$.

Proof. We claim that there exists an $l \times |V(G)|$ G -Sperner matrix A if and only if there exists a homomorphism f from G to $\mathcal{S}(l)$. Indeed, if there exists such A , then $f : V(G) \rightarrow \mathbb{Z}_2^l$ with $f(v) = A^v$ is a homomorphism because for any edge $\{a, b\} \in E(G)$, there exist $j, k \in [1, l]$ such that $A[j, a] = 0, A[j, b] = 1$ and $A[k, a] = 1, A[k, b] = 0$, and thus $\{A^a, A^b\} \in E(\mathcal{S}(l))$. Now, for the converse, if there exists a homomorphism f from G to $\mathcal{S}(l)$, then for every edge

$\{a, b\} \in E(G)$, we must have $\{f(a), f(b)\} \in E(\mathcal{S}(l))$. Let A be an $l \times |V(G)|$ matrix where $A^v = f(v)$. Then, if $\{f(a), f(b)\} \in E(\mathcal{S}(l))$, there exist $j, k \in [1, l]$ such that $(f(a))_j = 0$, $(f(b))_j = 1$, $(f(a))_k = 1$, and $(f(b))_k = 0$. This also means that in rows j and k of A , we have $A[j, a] = 0$, $A[j, b] = 1$, $A[k, a] = 1$, and $A[k, b] = 0$. This claim implies $t_s(G) = \min_{l \in \mathbb{N}} \{l : \exists \text{ a } G\text{-Sperner } (l, |V(G)|)\} = \min_{l \in \mathbb{N}} \{l : G \rightarrow \mathcal{S}(l)\}$. ■

We have now provided all the foundational results needed to prove the main result of this section.

Corollary 4.1.10. Let $G = (V, E)$ be a graph without any isolated vertex. Then, $t_s(G) = t(1, \chi(G))$.

Proof. Since $G \rightarrow K_{\chi(G)}$, from Propositions 4.1.4 and 4.1.5, we obtain $t_s(G) \leq t_s(K_{\chi(G)})$ and $t_s(K_{\chi(G)}) = t(1, \chi(G))$. Thus, $t_s(G) \leq t(1, \chi(G))$.

Assume $t(1, \chi(G)) = k$ for some $k \in \mathbb{N}$. Suppose for the sake of contradiction that $t_s(G) < t(1, \chi(G))$, that is, $t_s(G) \leq k - 1$. Then, by Proposition 4.1.9, $G \rightarrow \mathcal{S}(k - 1)$. By Theorem 4.1.7, $\chi(\mathcal{S}(k - 1)) = \binom{k-1}{\lfloor \frac{k-1}{2} \rfloor}$. Since we know that $t(1, \chi(G)) = \min\{l : \binom{l}{\lfloor \frac{l}{2} \rfloor} \geq \chi(G)\}$ and $\chi(G) \leq \chi(\mathcal{S}(k - 1)) = \binom{k-1}{\lfloor \frac{k-1}{2} \rfloor}$, the minimum l occurs for some $l \leq k - 1$. Thus, $t(1, \chi(G)) \leq k - 1 < t(1, \chi(G))$, which is a contradiction. So, $t_s(G) = t(1, \chi(G))$. ■

4.2 Cover-free families on graphs

We begin this section by giving two equivalent definitions of cover-free families on graphs. Definition 4.2.1 uses the set system and is a specialization to graphs from the more general cover-free families on hypergraphs defined in [40]. Definition 4.2.2 is an equivalent one using the incidence matrix of the set system.

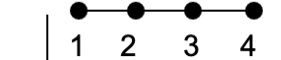
Definition 4.2.1. (via set system) Let n, t be positive integers with $n \geq t$, and let $G = ([1, n], E(G))$ be a graph. Let $\mathcal{F} = ([1, t], \mathcal{B})$ be a set system with $\mathcal{B} = \{B_1, B_2, \dots, B_n\}$ and $|\mathcal{B}| = n$ where B_i corresponds to vertex i . The set system $\mathcal{F}$ is *cover-free for an edge* $\{a, b\} \in E(G)$, if for any $i_0 \in [1, n] \setminus \{a, b\}$, we have

$$|B_{i_0} \setminus (B_a \cup B_b)| \geq 1.$$

Then,

1. $\mathcal{F}$ is a G -ECFF(t, n) if $\mathcal{F}$ is cover-free for all edges in $E(G)$.
2. $\mathcal{F}$ is a G -CFF(t, n) if $\mathcal{F}$ is both G -ECFF(t, n) and G -Sperner(t, n).

We now present equivalent definitions of the above definitions via incidence matrices.



	1	2	3	4
0	0	0	1	1
1	1	0	0	1
1	1	1	0	0

Figure 4.3: A P_4 -disjunct matrix

Definition 4.2.2. (via incidence matrix) Let $G = ([1, n], E(G))$ be a graph, and let A be a $t \times n$ binary matrix where each column corresponds to a vertex of G . We denote by $A[i, j]$ the entry of A in row i and the column corresponding to the vertex $j \in [1, n]$, and by A^j the column vector of A corresponding to the vertex j . A is *cover-free for an edge* $\{a, b\} \in E(G)$, if for any $v_0 \in [1, n] \setminus \{a, b\}$, there exists a row index $l \in [1, t]$ such that $A[l, a] = 0$, $A[l, b] = 0$, and $A[l, v_0] = 1$. Then,

1. Matrix A is G -disjunct if it is cover-free for every edge in $E(G)$.
2. Matrix A is $\overline{G}$ -disjunct if it is both G -disjunct and G -Sperner.

We remark that the notation $\overline{G}$ used here should not be confused with the standard notation for the complement of G . It is easy to prove the equivalences given in the following proposition.

Proposition 4.2.3. Let $G = ([1, n], E(G))$ be a graph and $\mathcal{F}$ be a set system. Then,

1. $\mathcal{F}$ is a G -ECFF(t, n) if and only if its incidence matrix is G -disjunct.
2. $\mathcal{F}$ is a G -CFF(t, n) if and only if its incidence matrix is $\overline{G}$ -disjunct.

Since a 2-disjunct matrix is also a $\overline{2}$ -disjunct matrix, being a 2-CFF(t, n) is equivalent to being a K_n -ECFF(t, n) or being a K_n -CFF(t, n). By Definition 4.2.1, a G -CFF (or a $\overline{G}$ -disjunct matrix) is a G -ECFF (or a G -disjunct matrix). However, in general, being a G -ECFF (or a G -disjunct matrix) is not equivalent to being a G -CFF (or a $\overline{G}$ -disjunct matrix). Figure 4.3 shows a G -disjunct matrix that is not $\overline{G}$ -disjunct, since the matrix is not Sperner for edges $\{1, 2\}$ and $\{3, 4\}$. We denote by $t_e(G)$ and $t(G)$ the minimum t such that there exist a G -ECFF(t, n) and a G -CFF(t, n); respectively:

$$t_e(G) = \min\{t : \exists \text{ a } G\text{-ECFF}(t, |V(G)|)\},$$

$$t(G) = \min\{t : \exists \text{ a } G\text{-CFF}(t, |V(G)|)\}.$$

It is clear that a G -ECFF(t, n) and a G -CFF(t, n) are equivalent to a $(G, 1)$ -ECFF(t, n) and a $(G, 1)$ -CFF(t, n), respectively, which were defined in Section 3.3.1. Thus, $t_e(G) = t_e(1, G)$ and $t(G) = t(1, G)$. The following proposition is obtained directly from Definition 4.2.1 and is also an immediate consequence of Proposition 3.3.3.

Proposition 4.2.4. [40] For a simple graph G , $t_e(G) \leq t(G) \leq t_e(G) + t_s(G)$.

For the graph L_n with n loops, we have $t(L_n) = t_e(L_n) = t(1, n)$. If G is a graph with a set of isolated vertices I and at least 3 non-isolated vertices, then $t_e(G - I) = t_e(G)$ and $t(G - I) = t(G)$; indeed, since isolated vertices are not part of any edges in the graph, to obtain a G -disjunct matrix from a $(G - I)$ -disjunct matrix or a $\overline{G}$ -disjunct matrix from a $(\overline{G - I})$ -disjunct matrix, it suffices to add columns of all 1's as the columns corresponding to the isolated vertices. Therefore, throughout the rest of the chapter, we assume that the graph has no isolated vertices.

The following corollary is a direct consequence of Theorem 3.3.5. Recall that Idalino and Moura [40] construct CFFs on hypergraphs using strong vertex coloring (Construction 3.3.6), which can be adapted to graphs as a construction using vertex coloring.

Corollary 4.2.5. Let P_n, C_n, T_n be respectively, a path, a cycle, a tree with $n \geq 3$ vertices, and let K_{n_1, n_2} be a complete bipartite graph with $n_1, n_2 \geq 2$. Then,

1.

$$t(P_n) \leq \begin{cases} 3 & \text{if } n = 3 \\ 2t(1, \frac{n}{2}) & \text{if } n \text{ is even, } n \geq 4, \\ t(1, \lfloor \frac{n}{2} \rfloor) + t(1, \lceil \frac{n}{2} \rceil) & \text{if } n \text{ is odd, } n \geq 5 \end{cases}$$

2.

$$t(C_n) \leq \begin{cases} 3 & \text{if } n = 3, \\ 2t(1, \frac{n}{2}) & \text{if } n \text{ is even, } n \geq 4, \\ 2t(1, \lfloor \frac{n}{2} \rfloor) + 1 & \text{if } n \text{ is odd, } n \geq 5. \end{cases}$$

3.

$$t(K_{n_1, n_2}) \leq t(1, n_1) + t(1, n_2).$$

4. $t(T_n) \leq t(1, n_1) + t(1, n_2)$ where n_1 and n_2 are the number of vertices in each of the two color classes for a 2-coloring of T_n .

Proof. The result follows from Theorem 3.3.5 and the fact that P_n, T_n are 2-colorable and C_n is 2-colorable for n even and 3-colorable for n odd. The size n_i of each color class for a 2-coloring is computed in the case of each type of graph:

1. For $P_n, n_1 = \lfloor \frac{n}{2} \rfloor$ and $n_2 = \lceil \frac{n}{2} \rceil$, for $n > 3$. For P_3 , see Proposition 4.6.1.
2. For $C_n, n_1 = \frac{n}{2}$ and $n_2 = \frac{n}{2}$ if n is even. If $n \geq 5$ is odd, construct a P_n -disjunct matrix by Theorem 3.3.5 and add a row with 0's in the columns corresponding to its endpoints and 1's for the remaining vertices. See Proposition 4.6.1, for C_3 .
3. K_{n_1, n_2} is 2-colorable with color classes of size n_1 and n_2 .
4. The tree T_n is 2-colorable with color classes of size n_1 and n_2 , as given.

■

For all cases of graph G_n with n vertices given in Corollary 4.2.5 above, $t(G_n) \leq 2\log_2 n + 1$ as n grows. However, in Section 4.4 we improve this bound for $t(P_n)$ and $t(C_n)$ to $\frac{3}{\log_2 3} \log_2 n + \mathcal{O}(1) \approx 1.893 \log_2(n) + \mathcal{O}(1)$.

Proposition 4.2.6. If G is a subgraph of H , then $t(G) \leq t(H)$ and $t_e(G) \leq t_e(H)$.

Proof. Let A be the incidence matrix of an H -CFF($t, |V(H)|$). Let M be the $t \times |V(G)|$ matrix where each column corresponding to a vertex $v_i \in V(G)$ is identical to the column corresponding to v_i in A . It is easy to see that M is $\overline{H[V(G)]}$ -disjunct since deletion of vertices from H to obtain $H[V(G)]$ does not affect the property that M is cover-free and Sperner for all edges in $H[V(G)]$. Similarly, removing edges from $H[V(G)]$ to obtain the graph G does not affect the property that M is cover-free and Sperner for all edges in G . Thus, M is $\overline{G}$ -disjunct and $t(G) \leq t(H[V(G)]) \leq t(H)$. A similar argument yields $t_e(G) \leq t_e(H[V(G)]) \leq t_e(H)$. ■

Theorem 4.2.7. Let $G = (V(G), E(G))$ be a graph with no isolated vertices. Then any G -CFF($t, |V(G)|$) is a 1-CFF($t, |V(G)|$).

Proof. Let $n = |V(G)|$ and A be a $\overline{G}$ -disjunct matrix. We want to show that A is 1-disjunct. Let $u, v \in V(G)$. First, consider the case where $\{u, v\} \in E(G)$. Since A is a G -Sperner matrix, there exists a row j_1 such that $A[j_1, u] = 0$ and $A[j_1, v] = 1$, and a row j_2 such that $A[j_2, u] = 1$ and $A[j_2, v] = 0$. Second, consider the case where $\{u, v\} \notin E(G)$. Let $u', v' \in V(G)$ such that $\{u, u'\} \in E(G)$ and $\{v, v'\} \in E(G)$, which must exist since G has no isolated vertex. Thus, since A is a $\overline{G}$ -disjunct matrix, then there must exist a row i_1 with $A[i_1, u] = A[i_1, u'] = 0$ and $A[i_1, v] = 1$ and a row i_2 with $A[i_2, v] = A[i_2, v'] = 0$ and $A[i_2, u] = 1$. From both cases considered, we conclude that for any $u, v \in V(G)$, there exists a row in A that has a 0 and a 1 in the columns corresponding to vertices u and v , respectively, and a row that has a 1 and a 0 in the columns corresponding to vertices u and v , respectively. This implies A is a 1-CFF. ■

Corollary 4.2.8. Let $G = (V(G), E(G))$ be a graph on n vertices, with no isolated vertices. Then,

$$t(1, n) \leq t(G) \leq t(2, n). \quad (4.2.1)$$

Proof. Since G is a subgraph of K_n , by Proposition 4.2.6, $t(G) \leq t(K_n) = t(2, n)$. Since, by Theorem 4.2.7, a G -CFF(t, n) is a 1-CFF(t, n), we have $t(1, n) \leq t(G)$. ■

Corollary 4.2.9. Let G be a graph with no isolated vertices and let $n = |V(G)|$ with $n = \binom{x}{\lfloor \frac{x}{2} \rfloor}$, $x \geq 3$. Then,

$$t(G) \geq x + 1 = t(1, n) + 1. \quad (4.2.2)$$

Proof. By Corollary 4.2.8, $t(G) \geq t(1, n)$. Then, by Equation 3.2.1,

$$t(1, n) = t\left(1, \binom{x}{\lfloor \frac{x}{2} \rfloor}\right) = x.$$

Suppose there exists a G -CFF $\left(x, \left(\lfloor \frac{x}{2} \rfloor\right)\right)$. Since by Theorem 4.2.7, a G -CFF is also a Sperner family with $\left(\lfloor \frac{x}{2} \rfloor\right)$ subsets; by Theorem 2.1.4, this family must consist of every subset of size $\lfloor \frac{x}{2} \rfloor$ or every subset of size $\lceil \frac{x}{2} \rceil$. Thus, the union of any two subsets in the family must contain at least one other subset in the family, a contradiction. Hence, it is impossible to construct a G -CFF $\left(x, \left(\lfloor \frac{x}{2} \rfloor\right)\right)$ on a ground set of size x , so $t(G) \geq x + 1 = t(1, n) + 1$. ■

4.2.1 Establishing that $t(G) \sim t_e(G)$

From Definition 4.2.1, $t_e(G) \leq t(G)$. In this section, we show that, in many cases, $t(G) = t_e(G)$ and in all cases, their difference is at most 2. First, we provide a necessary and sufficient condition for a graph G such that any G -ECFF must be a G -CFF.

Proposition 4.2.10. For a graph $G = ([1, n], E(G))$, any G -disjunct matrix is $\overline{G}$ -disjunct if and only if $\delta(G) \geq 2$.

Proof. First, suppose $\delta(G) \geq 2$. We need to show that every G -disjunct matrix is also G -Sperner. Let A be a G -disjunct matrix. For an edge $\{u, v\}$, let x and y be neighbors of u and v , respectively, such that $x, y \notin \{u, v\}$. Then, for the edge $\{u, x\}$ and the vertex v , there must exist a row i in A such that $A[i, u] = 0$, $A[i, x] = 0$, and $A[i, v] = 1$. Similarly, for the edge $\{v, y\}$ and the vertex u , there must exist a row j in A such that $A[j, v] = 0$, $A[j, y] = 0$, and $A[j, u] = 1$. Thus, we have $A[i, u] = 0$, $A[i, v] = 1$, $A[j, v] = 0$, and $A[j, u] = 1$, which proves that A is Sperner for the edge $\{u, v\}$. Thus, A is G -Sperner, which implies A is $\overline{G}$ -disjunct.

Now, we show by contraposition that if $\delta(G) < 2$, there exists a G -disjunct matrix that is not $\overline{G}$ -disjunct. Let $\{u, v\}$ be an edge with $\deg(u) = 1$. Let A be a $t \times (n - 1)$ $(G - \{u\})$ -disjunct matrix. Let B be a $t \times n$ matrix constructed by adding a column to A , corresponding to u , which is equal to A^v . It is easy to see that B is G -disjunct since using that A is 1-disjunct, we conclude that B is cover-free for $\{u, v\}$. Note that B is not $\overline{G}$ -disjunct because B is not Sperner for $\{u, v\}$. Hence, B is not G -Sperner which means B is not $\overline{G}$ -disjunct. ■

Theorem 4.2.11. Let G be a graph.

1. If $\delta(G) \geq 2$, then

$$t(G) = t_e(G). \quad (4.2.3)$$

2. If $\delta(G) = 1$ and G has no connected component that is a single edge, then

$$t_e(G) \leq t(G) \leq t_e(G) + 1. \quad (4.2.4)$$

3. If $\delta(G) = 1$, then

$$t_e(G) \leq t(G) \leq t_e(G) + 2. \quad (4.2.5)$$

Proof. By Proposition 4.2.4, we already know that $t_e(G) \leq t(G)$. Let A be a $t_e(G) \times |V(G)|$ G -disjunct matrix. In the proof of Proposition 4.2.10, we showed that A is Sperner for an edge $\{u, v\}$ if $\deg(u), \deg(v) \geq 2$. Now, we prove each of the statements:

1. By Proposition 4.2.10, if $\delta(G) \geq 2$, every G -ECFF is also a G -CFF. This implies $t(G) \leq t_e(G)$, which concludes this case.
2. Let A' be a $(t_e(G) + 1) \times |V(G)|$ matrix created by appending a new row to A with 0's in the columns corresponding to vertices of degree 1, and 1's in the remaining columns. It is enough to show that A' is Sperner for any edge with an endpoint of degree 1. Let $\{u, v\}$ be an edge such that $\deg(v) = 1$, and let w be another neighbor of u . Such w exists because no connected component of G is a single edge. Then, for the edge $\{u, w\}$ and the vertex v , there must exist a row i in A such that $A[i, u] = 0, A[i, w] = 0$, and $A[i, v] = 1$. The last row of A' contains a 0 in the column corresponding to v and a 1 in the column corresponding to u . Thus, A' is Sperner for $\{u, v\}$. This yields the bound $t(G) \leq t_e(G) + 1$.
3. The only case we need to consider is when $\delta(G) = 1$ and G has at least one connected component that is a single edge. Let A'' be a $(t_e(G) + 2) \times |V(G)|$ matrix obtained by appending two new rows to A for dealing with vertices of degree 1 as follows. First, copy an identity matrix I_2 under the columns corresponding to the vertices of each single-edge component. Then, for the vertices that are not in a single-edge component, add an extra row to the corresponding columns of A , in the same way as done for A' in Case 2, and then add an arbitrary row to complete $t_e(G) + 2$ rows. Thus, A'' is G -Sperner, and since A is G -disjunct, then A'' is $\overline{G}$ -disjunct. This yields the bound $t(G) \leq t_e(G) + 2$.

■

Notation 4.2.12. Let G be a graph and let $x \notin V(G)$. Then, $G + \{x\}$ is a graph where $V(G + \{x\}) = V(G) \cup \{x\}$ and $E(G + \{x\}) = E(G) \cup \{\{v, x\} : v \in V(G)\}$.

In other words, $G + \{x\}$ adds a universal vertex x to G . A *wheel* graph W_n is a graph constructed by joining a single universal vertex to all vertices of a cycle of length $n - 1$. Thus, $W_n = C_{n-1} + \{n\}$ and $K_{n+1} = K_n + \{n + 1\}$.

Theorem 4.2.13. Let $n \geq 3$. Let $G = ([1, n], E(G))$ be a graph with no isolated vertices. Then,

$$\max\{t(1, n) + 1, t(G)\} \leq t(G + \{0\}) \leq t(G) + 1. \quad (4.2.6)$$

Proof. Let A be a $t \times n$ $\overline{G}$ -disjunct matrix with $t = t(G)$. Let M be a $(t + 1) \times (n + 1)$ matrix of the form $\begin{pmatrix} \mathbf{0}_{t \times 1} & A \\ \mathbf{1}_{1 \times 1} & \mathbf{0}_{1 \times n} \end{pmatrix}$, with columns labeled, in order, by $0, 1, \dots, n$; and rows labeled as elements in $[1, t + 1]$. We first show that M is $\overline{G + \{0\}}$ -disjunct. By Theorem 4.2.7, A is 1-disjunct. Thus, for any vertices $y, z \in [1, n]$, there exists a row $r \in [1, t]$ of A such that $A[r, y] = 0$ and $A[r, z] = 1$. Thus, for $\{0, y\} \in E(G + \{0\})$, we have $M[r, 0] = 0, M[r, y] = 0$,

and $M[r, z] = 1$. So, M is cover-free for $\{0, y\}$. Furthermore, since A is also G -disjunct, for each edge $\{y, w\} \in E(G)$ and any vertex $v \in [1, n] \setminus \{y, w\}$, there exists a row $r \in [1, t]$ of A such that $A[r, y] = 0$, $A[r, w] = 0$, and $A[r, v] = 1$. So, $M[r, y] = 0$, $M[r, w] = 0$, and $M[r, v] = 1$. Hence, M is $(G + \{0\})$ -disjunct. Thus, $t_e(G + \{0\}) \leq t(G) + 1$. Since $\delta(G) \geq 1$, $\delta(G + \{0\}) \geq 2$. Then, by Theorem 4.2.11 part 1, $\overline{t(G + \{0\})} = t_e(G + \{0\}) \leq t(G) + 1$ (Using Proposition 4.2.10, we also obtain that M is $\overline{(G + \{0\})}$ -disjunct).

Let M be a $\overline{(G + \{0\})}$ -disjunct matrix. Let I be the set of rows in M such that $M[i, 0] = 0$ for all $i \in I$. Then, the submatrix M' of M restricted to the columns $M^1, M^2, \dots, M^n$ and rows in I must be a $|I| \times n$ 1-disjunct matrix. Therefore, $|I| \geq t(1, n)$. Since there must be at least one row r such that $A[r, 0] = 1$ and $A[r, y] = 0$ for $y \in [1, n]$, then $t(G + \{0\}) \geq t(1, n) + 1$. Furthermore, since $G \subseteq G + \{0\}$; by Proposition 4.2.6, $t(G) \leq t(G + \{0\})$. Thus, we have the desired lower bound. ■

Theorem 4.2.13 generalizes the bound $t(2, n) \leq t(2, n - 1) + 1$ in Equation 3.2.3, which can also be written as $t(K_n) \leq t(K_{n-1}) + 1$. In addition, if $c = t(G) - t(1, n)$, then from Theorem 4.2.13, it follows that $t(1, n) + c \leq t(G + \{0\}) \leq t(G) + 1 = t(1, n) + c + 1$. For the case of $c = 1$, the next corollary provides an exact bound of $t(G + \{0\})$ in terms of $t(G)$.

Corollary 4.2.14. Let $G = ([1, n], E)$ be a graph without isolated vertices. If $t(G) - t(1, n) = 1$, then $t(G + \{0\}) = t(G) + 1$.

Proof. Since $t(G) = t(1, n) + 1$, by Theorem 4.2.13 we have $t(1, n) + 1 \leq t(G + \{0\}) \leq t(1, n) + 2$. Suppose, for the sake of contradiction, that M is a $(t(1, n) + 1) \times (n + 1)$ $\overline{(G + \{0\})}$ -disjunct matrix. Let I be the set of rows in M such that $M[i, 0] = 0$ for all $i \in I$. Then, the submatrix M' of M restricted to the columns $M^1, M^2, \dots, M^n$ and rows in I must be a $|I| \times n$ 1-disjunct matrix. Therefore, $|I| \geq t(1, n)$. Since M has $t(1, n) + 1$ rows, we have $|I| = t(1, n)$ and the remaining row $r \notin I$ must be the only row such that $M[r, 0] = 1$ and $M[r, y] = 0$ for all $y \in [1, n]$. This structure also implies M' must be $\overline{G}$ -disjunct, but $t(G) = t(1, n) + 1$ and M' has $t(1, n)$ rows, which leads to a contradiction. Thus, $t(G + \{0\}) = t(1, n) + 2 = t(G) + 1$. ■

4.3 Tightness of various lower and upper bounds

In this section, we provide families of graphs G for which $t(G)$ achieves some of the extreme bounds presented in Section 4.2. In particular, we show that for a star graph S_n , $t(S_n)$ meets the upper bound of Equation 4.2.4 in Theorem 4.2.11, and exhibit a subfamily of star graphs where $t(S_n)$ meets the lower bound of Equation 4.2.1 in Corollary 4.2.8. Furthermore, S_n also achieves the upper bound of Equation 4.2.6 in Theorem 4.2.13. We also provide infinite families of graphs achieving the upper bound of Equation 4.2.5 in Theorem 4.2.11 and one of them meets the lower bound of Equation 4.2.1 in Corollary 4.2.8.

Definition 4.3.1. A star S_n is the complete bipartite graph $K_{1, n-1}$.

Thus, a star graph S_n is $G + \{x\}$ where G is a graph with $n - 1$ vertices and no edges.

Construction 4.3.2. Let A be a $t \times (n - 1)$ binary matrix. Let M_1 be a matrix with columns labeled as $0, 1, \dots, n - 1$ such that $M_1^0 = \mathbf{0}_{t \times 1}$ and $M_1^i = A^i$ for all $i \in [1, n - 1]$. Let M_2 be the matrix obtained by appending a row to M_1 such that $M_2[t + 1, 0] = 1$ and $M_2[t + 1, i] = 0$ for all $i \in [1, n - 1]$.

Theorem 4.3.3. Let $n \geq 3$ and $G = S_n$ be a star graph on n vertices. Then $t_e(S_n) = t(1, n - 1)$.

Proof. We give label 0 to the universal vertex and labels $1, \dots, n - 1$ to the other vertices. We then create a $t \times n$ matrix M_1 by following Construction 4.3.2 with A being a $t \times (n - 1)$ 1-disjunct matrix, with $t = t(1, n - 1)$. For every edge $e = \{0, i\}$ with $i \in [1, n - 1]$ and any vertex $j \in [1, n - 1] \setminus \{i\}$, there exists a row $l \in [1, t(1, n - 1)]$ such that $M_1[l, 0] = 0$, $M_1[l, i] = 0$ and $M_1[l, j] = 1$ since M_1^i and M_1^j are the i^{th} and j^{th} columns of A . Thus, M_1 is an S_n -disjunct matrix, which implies $t_e(S_n) \leq t(1, n - 1)$.

Now, let M be a $t^* \times n$ S_n -disjunct matrix. Let $i, j \in [1, n - 1], i \neq j$. Then since $\{0, i\}$ is an edge, there must be a row $l \in [1, t^*]$ such that $M[l, 0] = 0$, $M[l, i] = 0$, and $M[l, j] = 1$. Similarly, since $\{0, j\}$ is an edge, there must be a row $k \in [1, t^*]$ such that $M[k, 0] = 0$, $M[k, j] = 0$, and $M[k, i] = 1$. Therefore, the submatrix $[M^1 \ M^2 \ M^3 \ \dots \ M^{n-1}]$ is 1-disjunct, which implies $t^* \geq t(1, n - 1)$. Taking $t^* = t_e(S_n)$ implies $t_e(S_n) = t(1, n - 1)$. ■

We show that $t(S_n)$ meets the upper bound of Equation 4.2.4 in Theorem 4.2.11.

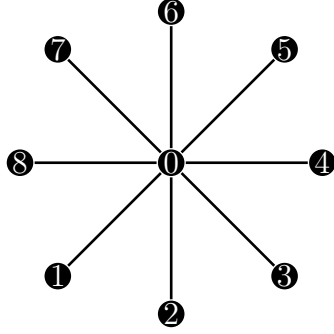
Corollary 4.3.4. Let $n \geq 3$ and $G = S_n$ be a star graph on n vertices. Then $t(S_n) = t_e(S_n) + 1 = t(1, n - 1) + 1$.

Proof. We construct the matrix M_2 using Construction 4.3.2, where A is a $t \times (n - 1)$ 1-disjunct matrix where $t = t(1, n - 1)$. In the proof of Theorem 4.3.3, we obtain that M_1 is S_n -disjunct. It guarantees the existence of a row l such that $M_2[l, 0] = 0$ and $M_2[l, i] = 1$ for any $i \in [1, n - 1]$. For the last row l' , we have $M_2[l', 0] = 1$ and $M_2[l', i] = 0$ for all $i \in [1, n - 1]$. This implies M_2 is S_n -Sperner. Thus, M_2 is an $\overline{S_n}$ -disjunct matrix, which implies $t(S_n) \leq t(1, n - 1) + 1$.

Let M be a $t^* \times n$ $\overline{S_n}$ -disjunct matrix with M^0 having weight x . Let M' be the submatrix of M corresponding to the rows of M that contain a zero in column M^0 and to the columns of vertices $1, 2, \dots, n - 1$. We claim M' is 1-disjunct. Indeed, for any $i, j \in [1, n - 1], i \neq j$, since $\{0, i\}$ and $\{0, j\}$ are edges and M is $\overline{S_n}$ -disjunct, there must be row indices k, l such that $M[k, 0] = 0, M[k, i] = 0, M[k, j] = 1$ and $M[l, 0] = 0, M[l, j] = 0, M[l, i] = 1$. Thus, $M'[k, i] = 0, M'[k, j] = 1$ and $M'[l, j] = 0, M'[l, i] = 1$. This implies that M' is 1-disjunct. So, $t^* - x \geq t(1, n - 1)$. Also, $x \geq 1$ since there must exist a row that has a 0 in the column $M^i, i \in [1, n - 1]$ and a 1 in the column M^0 . Hence, $t^* = t^* - x + x \geq t(1, n - 1) + 1$. Thus, $t(S_n) = t(1, n - 1) + 1$ and by Theorem 4.3.3, $t(1, n - 1) = t_e(S_n)$. ■

We provide an example of an optimal $\overline{S_n}$ -disjunct matrix for $n = 9$ in Figure 4.4.

The next corollary provides an infinite family of star graphs S_n such that $t(S_n) = t(1, n)$, showing that the lower bound in Equation 4.2.1 is tight.

(a) Star graph S_9

$\{0\}$	$V(S_9) \setminus \{0\}$
$\mathbf{0}_{5 \times 1}$	$\mathbf{1\text{-CFF}(5, 8)}$
$\mathbf{1}_{1 \times 1}$	$\mathbf{0}_{1 \times 8}$

(b) A 6×9 $\overline{S_9}$ -disjunct matrixFigure 4.4: An optimal $\overline{S_9}$ -disjunct matrix

Corollary 4.3.5. If $n = \binom{x}{\lfloor \frac{x}{2} \rfloor} + 1$ for some $x \in \mathbb{N}$, then $t(S_n) = t(1, n)$.

Proof. If $n = \binom{x}{\lfloor \frac{x}{2} \rfloor} + 1$, by Equation 3.2.1, $t(1, n-1) = t\left(1, \binom{x}{\lfloor \frac{x}{2} \rfloor}\right) = x$. Then, by Corollary 4.3.4, $t(S_n) = t(1, n-1) + 1 = x + 1 = t(1, n)$. ■

We show next that S_n is an infinite family of graphs satisfying the upper bound of Equation 4.2.6.

Corollary 4.3.6. For $n \geq 3$, $t(S_n + \{x\}) = t(S_n) + 1$.

Proof. By Theorem 4.2.13, $t(1, n) + 1 \leq t(S_n + \{x\}) \leq t(S_n) + 1$. By Corollary 4.3.4, $t(S_n) = t(1, n-1) + 1$. If $n = \binom{x}{\lfloor \frac{x}{2} \rfloor} + 1$ for some $x \in \mathbb{N}$, then by Corollary 4.3.5, $t(S_n) = t(1, n)$ and it implies $t(S_n + \{x\}) = t(S_n) + 1$. For other values of n , we have $t(1, n) = t(1, n-1)$. It further implies $t(S_n) - t(1, n) = 1$. Then by Corollary 4.2.14, $t(S_n + \{x\}) = t(S_n) + 1$. ■

We now show that there exists an infinite family of graphs meeting the upper bound of Equation 4.2.5 in Theorem 4.2.11, which also meets the lower bound of Equation 4.2.1 in Corollary 4.2.8.

Proposition 4.3.7. If $m \leq \lfloor \frac{1}{2}(\overline{\overline{m}} + 1) \rfloor$, then $t_e(\mathcal{E}_{2m}) + 1 \leq t(\mathcal{E}_{2m}) \leq t_e(\mathcal{E}_{2m}) + 2$. Otherwise, $t(\mathcal{E}_{2m}) = t_e(\mathcal{E}_{2m}) + 2 = t(1, 2m)$.

Proof. By Proposition 3.3.9, $t_e(\mathcal{E}_{2m}) = t(1, m)$ and by Theorem 4.2.11 item 3, $t(\mathcal{E}_{2m}) \leq t_e(\mathcal{E}_{2m}) + 2 = t(1, m) + 2$. By Theorem 4.2.8, it follows that $t(1, 2m) \leq t(\mathcal{E}_{2m}) \leq t(1, m) + 2$. If $m \leq \lfloor \frac{1}{2}(\overline{\overline{m}} + 1) \rfloor$, then by Proposition 3.2.8, $t(1, 2m) = t(1, m) + 1$ and it yields $t_e(\mathcal{E}_{2m}) + 1 \leq t(\mathcal{E}_{2m}) \leq t_e(\mathcal{E}_{2m}) + 2$. Otherwise, $t(1, 2m) = t(1, m) + 2$ and we obtain $t(\mathcal{E}_{2m}) = t(1, m) + 2 = t(1, 2m)$. ■

Notice that Proposition 4.3.7 provides an infinite family of $\mathcal{E}_n$ which meets the upper bound of $t(G)$ given in Equation 4.2.5 of Theorem 4.2.11, that is, if $n = 2m$ with $m > \lfloor \frac{1}{2}(\overline{\overline{m}} + 1) \rfloor$, then $t(\mathcal{E}_{2m}) = t_e(\mathcal{E}_{2m}) + 2$. The next theorem provides another infinite family

of $\mathcal{E}_n$ which also meets the upper bound of $t(G)$ given in Equation 4.2.5 of Theorem 4.2.11, as well as, the lower bound of $t(G)$ given in Equation 4.2.2 of Corollary 4.2.9.

Theorem 4.3.8. Let $x \geq 4$ be an integer such that $\binom{x}{\lfloor \frac{x}{2} \rfloor}$ is even. Then,

$$t\left(\mathcal{E}_{\left(\lfloor \frac{x}{2} \rfloor\right)}\right) = x + 1 = t_e\left(\mathcal{E}_{\left(\lfloor \frac{x}{2} \rfloor\right)}\right) + 2.$$

Proof. By Proposition 4.3.7, $t\left(\mathcal{E}_{\left(\lfloor \frac{x}{2} \rfloor\right)}\right) \leq t_e\left(\mathcal{E}_{\left(\lfloor \frac{x}{2} \rfloor\right)}\right) + 2$ and by Proposition 3.3.9, we have $t_e\left(\mathcal{E}_{\left(\lfloor \frac{x}{2} \rfloor\right)}\right) = t\left(1, \frac{1}{2}\binom{x}{\lfloor \frac{x}{2} \rfloor}\right)$. Since $\left|V\left(\mathcal{E}_{\left(\lfloor \frac{x}{2} \rfloor\right)}\right)\right| = \binom{x}{\lfloor \frac{x}{2} \rfloor}$, by Corollary 4.2.9, $t\left(\mathcal{E}_{\left(\lfloor \frac{x}{2} \rfloor\right)}\right) \geq x + 1$. Therefore, $x + 1 \leq t\left(\mathcal{E}_{\left(\lfloor \frac{x}{2} \rfloor\right)}\right) \leq t_e\left(\mathcal{E}_{\left(\lfloor \frac{x}{2} \rfloor\right)}\right) + 2 = t\left(1, \frac{1}{2}\binom{x}{\lfloor \frac{x}{2} \rfloor}\right) + 2$. To complete the proof, it is enough to show that $t\left(1, \frac{1}{2}\binom{x}{\lfloor \frac{x}{2} \rfloor}\right) = x - 1$. To prove this, we consider two cases:

Case 1: $x = 2k + 1$ for some $k \geq 2$. Then, $t\left(1, \frac{1}{2}\binom{x}{\lfloor \frac{x}{2} \rfloor}\right) = t\left(1, \frac{1}{2}\binom{2k+1}{k}\right)$ and by Proposition 3.2.5, we have $t\left(1, \frac{1}{2}\binom{2k+1}{k}\right) = 2k$, which is equal to $x - 1$.

Case 2: $x = 2k$ for some $k \geq 2$. Then, $t\left(1, \frac{1}{2}\binom{x}{\lfloor \frac{x}{2} \rfloor}\right) = t\left(1, \frac{1}{2}\binom{2k}{k}\right) = t\left(1, \frac{2k}{2k}\binom{2k-1}{k-1}\right) = t\left(1, \binom{2k-1}{k-1}\right) = 2k - 1 = x - 1$.

$$\text{Thus, } t\left(\mathcal{E}_{\left(\lfloor \frac{x}{2} \rfloor\right)}\right) = x + 1 = t_e\left(\mathcal{E}_{\left(\lfloor \frac{x}{2} \rfloor\right)}\right) + 2. \quad \blacksquare$$

Proposition 4.3.9. $t(\mathcal{E}_8) = 5 = t_e(\mathcal{E}_8) + 1$.

Proof. Since $4 \leq \overline{\overline{\lfloor \frac{1}{2}(4+1) \rfloor}} = 5$, by Proposition 4.3.7, we have $5 \leq t(\mathcal{E}_8) \leq 6$. Let $V(\mathcal{E}_8) = \{v_i : i \in [1, 8]\}$ and

$$E(\mathcal{E}_8) = \{\{v_1, v_2\}, \{v_3, v_4\}, \{v_5, v_6\}, \{v_7, v_8\}\}.$$

With the ground set $[1, 5]$, let $\mathcal{F} = \{B_{v_i} : i \in [1, 8]\}$ be a set system where

$$\begin{aligned} B_{v_1} &= \{1, 2\} & B_{v_3} &= \{1, 4\} & B_{v_5} &= \{2, 4\} & B_{v_7} &= \{3, 4\} \\ B_{v_2} &= \{1, 3\} & B_{v_4} &= \{1, 5\} & B_{v_6} &= \{2, 5\} & B_{v_8} &= \{3, 5\} \end{aligned}$$

It is easy to verify that $\mathcal{F}$ is an $\mathcal{E}_8$ -CFF implying $t(\mathcal{E}_8) = 5$. Also, by Proposition 3.3.9, $t_e(\mathcal{E}_8) = t(1, 4) = 4$. $\blacksquare$

Propositions 4.3.7 and 4.3.9 lead us to the following open problems, which are worth investigating.

Open Problem 4.3.10. Is there an infinite family of $\mathcal{E}_{2m}$, where $t(\mathcal{E}_{2m}) = t_e(\mathcal{E}_{2m}) + 1$?

Open Problem 4.3.11. Determine the values of m such that $t(\mathcal{E}_{2m}) = t_e(\mathcal{E}_{2m}) + 1$ and $t(\mathcal{E}_{2m}) = t_e(\mathcal{E}_{2m}) + 2$.

Now, we show that there are infinite families of graphs G with $\deg(G) = 1$ meeting the lower bounds provided in Equations 4.2.4 and 4.2.5.

Notation 4.3.12. • A *complete graph K_n with a pendant vertex attached* is a graph G obtained from K_n by adding a new vertex x and an edge joining x to exactly one vertex of K_n . In other words, $V(G) = V(K_n) \dot{\cup} \{x\}$ and $E(G) = E(K_n) \cup \{\{x, v\}\}$ for some $v \in V(K_n)$. We denote this graph by $K_n \cup K_2$.

- A *complete graph K_n with an edge component* is a graph G obtained by adding two new vertices x, y to K_n and an edge $\{x, y\}$. In other words, $V(G) = V(K_n) \dot{\cup} \{x, y\}$ and $E(G) = E(K_n) \cup \{\{x, y\}\}$. We denote this graph by $K_n \dot{\cup} K_2$.

Proposition 4.3.13. 1. Let J_{n+1} be a simple graph with $n + 1$ vertices such that $K_n \subset J_{n+1}$. If $t(2, n + 1) = t(2, n)$, then $t(J_{n+1}) = t_e(J_{n+1})$.

2. Let Y_{n+2} be a simple graph with $n + 2$ vertices such that $K_n \subset Y_{n+2}$. If $t(2, n + 2) = t(2, n)$, then $t(Y_{n+2}) = t_e(Y_{n+2})$.

Proof.

1. Since $K_n \subseteq J_{n+1} \subseteq K_{n+1}$, by Propositions 4.2.4 and 4.2.6, we have $t(2, n) = t_e(K_n) \leq t_e(J_{n+1}) \leq t(J_{n+1}) \leq t(K_{n+1}) = t(2, n + 1)$. Since $t(2, n) = t(2, n + 1)$, we have $t(J_{n+1}) = t_e(J_{n+1})$.
2. Since $K_n \subseteq Y_{n+2} \subseteq K_{n+2}$, by Propositions 4.2.4 and 4.2.6, we have $t(2, n) = t_e(K_n) \leq t_e(Y_{n+2}) \leq t(Y_{n+2}) \leq t(K_{n+2}) = t(2, n + 2)$. Since $t(2, n) = t(2, n + 2)$, we have $t(Y_{n+2}) = t_e(Y_{n+2})$.

■

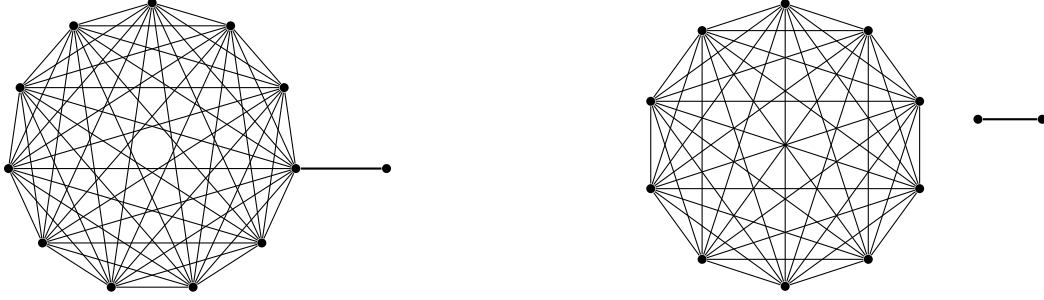
Corollary 4.3.14. 1. If $t(2, n + 1) = t(2, n)$, then $t(K_n \cup K_2) = t_e(K_n \cup K_2)$.

2. If $t(2, n + 2) = t(2, n)$, then $t(K_n \dot{\cup} K_2) = t_e(K_n \dot{\cup} K_2)$.

Proof. We get the desired results by applying Proposition 4.3.13 with $J_{n+1} = K_n \cup K_2$ and $Y_{n+2} = K_n \dot{\cup} K_2$. ■

Remark 4.3.15. Since $t(2, n) \leq 5.512 \log_2(n)$ [21], there must exist infinite families of n_i and n'_i with $t(2, n_i) = t(2, n_i + 1)$ and $t(2, n'_i) = t(2, n'_i + 2)$, respectively.

Example 4.3.16. We provide an example of $K_{11} \cup K_2$ and a $K_{10} \dot{\cup} K_2$ in Figure 4.5. From Table 3.4, we have $t(2, 10) = t(2, 11) = t(2, 12) = 9$. By Proposition 4.3.13, in both cases, we have $t(G) = t_e(G)$.

Figure 4.5: A $K_{11} \cup K_2$ (left) and a $K_{10} \dot{\cup} K_2$ (right)

4.4 Cover-free families on paths and cycles

By Corollary 4.2.5 and Proposition 4.2.6, $t(P_n) \leq t(C_n) \leq 2t(1, \lfloor n/2 \rfloor) + (n \bmod 2)$. Using a mixed-radix Gray code, we provide a construction that improves this upper bound to $1.893 \log_2 n + \mathcal{O}(1)$. Section 4.4.1 provides an introduction to Gray codes and their generalizations. Further details, together with proofs of certain propositions stated in this section, are presented in Chapter 6.

4.4.1 Mixed-radix Gray code

The *Hamming distance* between two n -tuples is the number of positions at which the corresponding co-ordinates differ. A *binary Gray code* G_n is an ordering of the 2^n binary n -tuples such that any two consecutive tuples have Hamming distance exactly one. Analogously, a mixed-radix Gray code $G_n^{(m_1, m_2, \dots, m_n)}$ is an ordering of the n -tuples $(a_1, a_2, \dots, a_n) \in \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \dots \times \mathbb{Z}_{m_n}^1$ such that any two consecutive tuples have Hamming distance exactly one. Hence, a binary Gray code G_n is $G_n^{(2, 2, \dots, 2)}$. A Gray code is *cyclic* if the Hamming distance between the first and last codewords is 1. Knuth discusses *mixed-radix reflected Gray codes* and *mixed-radix modular Gray codes* in [43], whose definitions and constructions are discussed next. Throughout the rest of this chapter, the adjective “mixed-radix” will be dropped from the mixed-radix Gray code, and we will simply refer to the Gray code $G_n^{(m_1, m_2, \dots, m_n)}$ and simply specify when it is binary.

Notation 4.4.1. For a Gray code $\mathcal{D} = [D_0, D_1, \dots, D_{m-1}, D_m]$, the reflection of $\mathcal{D}$ is denoted $\mathcal{D}^R = [D_m, D_{m-1}, \dots, D_1, D_0]$. For an n -tuple $D = (a_1, a_2, \dots, a_n)$, $x \cdot D$ denotes $(x, a_1, a_2, \dots, a_n)$ and $x \cdot \mathcal{D}$ denotes $[x \cdot D_0, x \cdot D_1, \dots, x \cdot D_{m-1}, x \cdot D_m]$. Similarly, $D \cdot x$ denotes $(a_1, a_2, \dots, a_n, x)$ and $\mathcal{D} \cdot x$ denotes $[D_0 \cdot x, D_1 \cdot x, \dots, D_{m-1} \cdot x, D_m \cdot x]$.

The *reflected Gray code* $R_n^{(m_1, m_2, \dots, m_n)}$ is defined recursively in Construction 6.2.5.

Construction 4.4.2. (Reflected Gray code)

$$R_1^{(m_n)} = \mathcal{R}_1 := [(0), (1), (2), \dots, (m_n - 1)]. \text{ For } i \geq 2,$$

¹We use $\mathbb{Z}_{m_1} \times \dots \times \mathbb{Z}_{m_n}$ to denote the Cartesian product of the sets $\mathbb{Z}_{m_1}, \dots, \mathbb{Z}_{m_n}$; not the direct product.

$$R_{i+1}^{(m_{n-i}, m_{n-i+1}, \dots, m_n)} = \mathcal{R}_{i+1} := \begin{cases} [0 \cdot \mathcal{R}_i, 1 \cdot \mathcal{R}_i^R, 2 \cdot \mathcal{R}_i, \dots, m_{n-i} - 1 \cdot \mathcal{R}_i^R], & \text{if } m_{n-i} \text{ is even} \\ [0 \cdot \mathcal{R}_i, 1 \cdot \mathcal{R}_i^R, 2 \cdot \mathcal{R}_i, \dots, m_{n-i} - 1 \cdot \mathcal{R}_i], & \text{if } m_{n-i} \text{ is odd} \end{cases}$$

Examples of reflected Gray codes are provided in Examples 6.1.1, 6.1.3, and 6.2.6–6.2.8. The proof of the next proposition can be found as the proof of Proposition 6.2.9.

Proposition 4.4.3. $R_n^{(m_1, m_2, \dots, m_n)}$ is cyclic if and only if m_1 is even.

Modular Gray codes $M_n^{(m_1, m_2, \dots, m_n)}$ [43] are, however, different from reflected ones in that each successor is obtained from its predecessor by incrementing exactly one coordinate (say at the i^{th} position) modulo m_i . The term ‘modular’ indicates that the ordering is generated by ‘cyclically wrapping around’. In general, transitions for reflected Gray code are $0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow m_i - 2 \rightarrow m_i - 1 \rightarrow m_i - 2 \rightarrow \dots \rightarrow 2 \rightarrow 1 \rightarrow 0$ while transitions for modular Gray code are $0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow m_i - 2 \rightarrow m_i - 1 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow \dots$.

Let M_n^q be an ordered list of tuples of length n where $m_1 = m_2 = \dots = m_n = q$. For $n \geq 2$, we express a construction of M_n^q recursively by referring to the individual codewords in M_{n-1}^q .

Notation 4.4.4. For $D \in \mathbb{Z}_q^{n-1}$ and $j \in \mathbb{Z}_q$, denote $D \cdot j^+ = [D \cdot (j \bmod q), D \cdot ((j+1) \bmod q), \dots, D \cdot ((j+q-1) \bmod q)]$.

Construction 4.4.5. (Modular Gray code) Let $q > 1$, $n \geq 1$. Define $M_1^q = [(0), (1), \dots, (q-1)]$, and if $n \geq 2$ and $M_{n-1}^q = [D_0, D_1, \dots, D_{q^{n-1}-1}]$; then

$$M_n^q = [D_0 \cdot 0^+, D_1 \cdot (q-1)^+, D_2 \cdot (q-2)^+, \dots, D_i \cdot (q-i)^+, \dots, D_{q^{n-1}-1} \cdot 1^+].$$

Examples of modular Gray codes are provided in Examples 6.2.14 and 6.2.15.

Proposition 4.4.6. Let $q > 1$ and $n \geq 1$ be integers. The modular Gray code M_n^q given in Construction 4.4.5 is cyclic.

Proof. The proof is by induction on n . The result can be easily verified for $n = 1$.

Induction Hypothesis: Assume that M_{n-1}^q is a cyclic modular Gray code for some integer $n \geq 2$. We will prove that M_n^q is a cyclic modular Gray code.

It is clear that M_n^q contains all the q^n n -tuples. For the order of the codewords listed in $D_i \cdot (q-i \bmod q)^+$ for all $0 \leq i \leq q^{n-1}-1$, it is clear that the rightmost digit of the codewords changes following the general transitions $0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow q-1 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow \dots$.

Now we consider two consecutive codewords, where one of them is the last codeword of $D_i \cdot l^+$ and the other is the first codeword of $D_{i+1} \cdot m^+$, with $0 \leq i \leq q^{n-1}-2$, $l = q-i \bmod q$, and $m = q-i-1 \bmod q$. Then the last codeword of $D_i \cdot l^+$ is $D_i \cdot (2q-i-1 \bmod q)$, which is $D_i \cdot (-i-1 \bmod q)$, and the first codeword of $D_{i+1} \cdot m^+$ is $D_{i+1} \cdot (q-i-1 \bmod q)$, which is $D_{i+1} \cdot (-i-1 \bmod q)$. Since, by the Induction Hypothesis, D_i and D_{i+1} are two consecutive codewords in M_{n-1}^q , which is cyclic and modular, $D_i \cdot (-i-1 \bmod q)$ and $D_{i+1} \cdot (-i-1 \bmod q)$ are also codewords following the general transition of a cyclic modular Gray code.

The above claim also holds for the first and last codewords in M_n^q , which are 0^n and $(q-1)0^{n-1}$. Therefore, M_n^q is a cyclic modular Gray code. ■

4.4.2 Constructions of cover-free families on paths and cycles using Gray codes

Let $n = \prod_{i=1}^k m_i$, $t = \sum_{i=1}^k m_i$, and $f(i, d) = \sum_{l=1}^{i-1} m_l + d + 1$ where $m_i, d \in \mathbb{Z}^+$. Let $\mathcal{P}_i = \{f(i, 0), f(i, 1), \dots, f(i, m_i - 1)\}$. Thus, $[1, t] = \mathcal{P}_1 \cup \mathcal{P}_2 \cup \dots \cup \mathcal{P}_k$. We define a bijection F between tuples $D = (d_1, d_2, \dots, d_k)$ in $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \dots \times \mathbb{Z}_{m_k}$ and k -subsets of $[1, t]$ that are transversals of $\{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_k\}$:

$$F(D) = \{f(1, d_1), f(2, d_2), \dots, f(k, d_k)\}. \quad (4.4.1)$$

Construction 4.4.7. Let $G_k^{(m_1, m_2, \dots, m_k)} = [D_0, D_1, D_2, \dots, D_{n-1}]$ where $t = \sum_{i=1}^k m_i$, $n = \prod_{i=1}^k m_i$ and $\mathcal{B}(G_k^{(m_1, m_2, \dots, m_k)}) = \{F(D_0), F(D_1), \dots, F(D_{n-1})\}$ with $B_i = F(D_i)$ for $i \in [0, n-1]$ and take

$$\mathcal{F}(G_k^{(m_1, m_2, \dots, m_k)}) = ([1, t], \mathcal{B}(G_k^{(m_1, m_2, \dots, m_k)})).$$

Given two subsets A and B , $A \Delta B$ denotes the symmetric difference of A and B . We prove the following lemma.

Lemma 4.4.8. Let F be the function defined in Equation 4.4.1. Let D_{z_1}, D_{z_2} , and D_{z_3} be distinct tuples in $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \dots \times \mathbb{Z}_{m_k}$. If the Hamming distance between D_{z_1} and D_{z_2} is 1, then $F(D_{z_3}) \not\subseteq F(D_{z_1}) \cup F(D_{z_2})$.

Proof. Assume D_{z_1} and D_{z_2} have Hamming distance 1. Let l be the unique coordinate in which D_{z_1} and D_{z_2} differ, that is, $D_{z_1} = (d_1, d_2, \dots, d_l, \dots, d_k)$ and $D_{z_2} = (d_1, d_2, \dots, d'_l, \dots, d_k)$ with $d_j \in \mathbb{Z}_{m_j}$, $d_l, d'_l \in \mathbb{Z}_{m_l}$, and $d_l \neq d'_l$. Let $A = \{f(1, d_1), \dots, f(l-1, d_{l-1}), f(l+1, d_{l+1}), \dots, f(k, d_k)\}$, $f_1 = f(l, d_l)$, and $f_2 = f(l, d'_l)$. Thus, $F(D_{z_1}) = A \cup \{f_1\}$ and $F(D_{z_2}) = A \cup \{f_2\}$. If $A \subseteq F(D_{z_3})$, then $F(D_{z_3}) = A \cup \{f_3\}$ where $f_3 \notin \{f_1, f_2\}$. So, $f_3 \notin F(D_{z_1}) \cup F(D_{z_2})$. If $A \not\subseteq F(D_{z_3})$, then there exists $x \in [1, k] \setminus \{l\}$ such that the x^{th} coordinate of D_{z_3} , $y_x \neq d_x$. In this case, $f(x, y_x) \in F(D_{z_3})$ and $f(x, y_x) \notin F(D_{z_1}) \cup F(D_{z_2})$. Therefore, in both cases, $F(D_{z_3}) \not\subseteq F(D_{z_1}) \cup F(D_{z_2})$. ■

Theorem 4.4.9. Let $n = \prod_{i=1}^k m_i$ and $t = \sum_{i=1}^k m_i$. Let $G_k = G_k^{(m_1, m_2, \dots, m_k)}$ be a Gray code and define $\mathcal{F}(G_k)$ as in Construction 4.4.7. Then, $\mathcal{F}(G_k)$ is a P_n -CFF(t, n). If G_k is cyclic, then $\mathcal{F}(G_k)$ is a C_n -CFF(t, n).

Proof. Since the sets in $\mathcal{B}(G_k)$ are pairwise distinct and of size k , $\mathcal{F}(G_k)$ is a Sperner family. Since any two consecutive codewords D_i and D_{i+1} with $i \in [0, n-2]$ have Hamming distance 1, by Lemma 4.4.8, $F(D_i) \cup F(D_{i+1})$ does not contain any other subset of $\mathcal{F}(G_k)$. Thus, $\mathcal{F}(G_k)$ is a P_n -CFF(t, n).

If G_k is cyclic, then the first and last codeword D_0 and D_{n-1} have Hamming distance 1. Thus, by Lemma 4.4.8, $F(D_0) \cup F(D_{n-1})$ does not contain any other subset of $\mathcal{F}(G_k)$. Thus, $\mathcal{F}(G_k)$ is a C_n -CFF(t, n). ■

$$[1, 6] = \{1, 2\} \cup \{3, 4\} \cup \{5, 6\}.$$

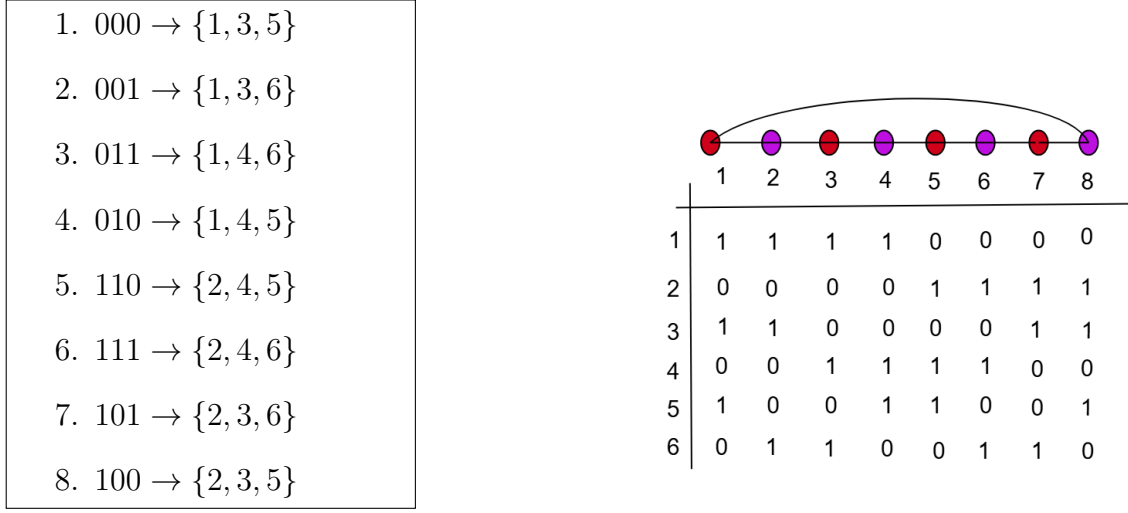


Figure 4.6: C_8 -CFF(6, 8) constructed from a binary reflected Gray code of length 3 (left) and the corresponding 6×8 C_8 -disjunct matrix (right)

Example 4.4.10. We present an example of the binary reflected Gray code of length 3 and construct the corresponding set system as described in Construction 4.4.7. By Theorem 4.4.9 and Proposition 6.2.9, $\mathcal{F}$ is a C_8 -CFF(6, 8). See Figure 4.6.

The ability to construct C_n -CFFs out of Gray codes ultimately leads us to the problem of how a set of size t should be partitioned into distinct parts such that the product n of the sizes of the parts is maximum, whose solution is provided in Theorem 2.3.2.

Theorem 4.4.11.

$$t(P_n) \leq t(C_n) \leq \begin{cases} 3k & \text{if } n \in (2 \cdot 3^{k-1}, 3^k] \text{ for some } k \geq 1, \\ 3k + 1 & \text{if } n \in (3^k, 4 \cdot 3^{k-1}] \text{ for some } k \geq 1, \\ 3k + 2 & \text{if } n \in (4 \cdot 3^{k-1}, 2 \cdot 3^k] \text{ for some } k \geq 1. \end{cases}$$

Proof. We split the proof into three different cases depending on the parity of n modulo 3.

Case 1: $n \in (2 \cdot 3^{k-1}, 3^k]$ for some $k \geq 1$.

For $k = 1$, use the identity matrix of order 3 as P_3 -CFF or C_3 -CFF. For $k \geq 2$, use Construction 4.4.7 applied to the Gray code M_k^3 to obtain $\mathcal{F}(M_k^3)$. By Proposition 4.4.6, M_k^3 is cyclic. By Theorem 4.4.9, $\mathcal{F}(M_k^3)$ is a C_{3^k} -CFF($3k, 3^k$). So, it remains to address the case $2 \cdot 3^{k-1} < n < 3^k$. By Construction 4.4.5, we observe that each triplet of codewords of M_k^3 appears in the form $D \cdot i, D \cdot (i + 1), D \cdot (i + 2)$ with $i \in \mathbb{Z}_3$, where D is a codeword of M_{k-1}^3 and there are 3^{k-1} such triplets. We delete $a = 3^k - n$ codewords D_j with $j \equiv 1 \pmod{3}$, where $1 \leq a \leq 3^{k-1} - 1$ to obtain the resultant ordering of the remaining codewords $(M_k^3)^a$, where consecutive codewords also have Hamming distance 1. By similar argument as in the proof of Lemma 4.4.8 and Theorem 4.4.9, $\mathcal{F}((M_k^3)^a)$ is a C_n -CFF($3k, n$).

Case 2: $n \in (3^k, 4 \cdot 3^{k-1}]$ for some $k \geq 1$.

Use Construction 6.2.5 to build the reflected Gray code $R_{k+1} = R_{k+1}^{(2,2,3,3,\dots,3)}$ and by Proposition 6.2.9, R_{k+1} is cyclic. By Theorem 4.4.9, $\mathcal{F}(R_{k+1})$ is a $C_{4 \cdot 3^{k-1}}\text{-CFF}(3k+1, 4 \cdot 3^{k-1})$, which solves the case $n = 4 \cdot 3^{k-1}$. For $3^k < n < 4 \cdot 3^{k-1}$, we employ a similar technique as in Case 1. Note that each triplet of codewords of $R_{k+1}^{(2,2,3,3,\dots,3)}$ appears in the form $D \cdot 0, D \cdot 1, D \cdot 2$ or $D \cdot 2, D \cdot 1, D \cdot 0$, where D is a codeword of $R_k^{(2,2,3,3,\dots,3)}$. Remove $a = 4 \cdot 3^{k-1} - n$ codewords of type $D \cdot 1$. This is always possible since $1 \leq a \leq 3^{k-1} - 1$ and there are $4 \cdot 3^{k-2} > 3^{k-1}$ such codewords. Thus, we obtain a $C_n\text{-CFF}(3k+1, n)$.

Case 3: $n \in (4 \cdot 3^{k-1}, 2 \cdot 3^k]$ for some $k \geq 1$.

Use Construction 6.2.5 to build the reflected Gray code $R_{k+1} = R_{k+1}^{(2,3,3,\dots,3)}$ and by Proposition 6.2.9, R_{k+1} is cyclic. By Theorem 4.4.9, $\mathcal{F}(R_{k+1})$ is a $C_{2 \cdot 3^k}\text{-CFF}(3k+2, 2 \cdot 3^k)$, which solves the case $n = 2 \cdot 3^k$. For $4 \cdot 3^{k-1} < n < 2 \cdot 3^k$, we employ a similar technique as in Case 2, and remove $a = 2 \cdot 3^k - n$ codewords of type $D \cdot 1$. This is always possible since $1 \leq a \leq 2 \cdot 3^{k-1} - 1$ and there are $2 \cdot 3^{k-1}$ such codewords. Thus, we obtain a $C_n\text{-CFF}(3k+2, n)$. ■

The following corollary gives asymptotic bounds for $t(P_n)$ and $t(C_n)$.

Corollary 4.4.12. For $n \geq 4$, we have $t(C_n) \leq 3 \log_3 n + 2 \leq 1.893 \log_2 n + \mathcal{O}(1)$.

Proof. By Theorem 4.4.11, $t(C_n) \leq 3k$ when $2 \cdot 3^{k-1} < n \leq 3^k$. Taking logarithms base 3 and multiplying through by 3 from the inequality $2 \cdot 3^{k-1} < n$, we get $3k < 3 \log_3 n + 3 - 3 \log_3 2$. Theorem 4.4.11 implies $t(C_n) \leq 3k + 1$ for the case $n \in (3^k, 4 \cdot 3^{k-1}]$ for some $k \geq 1$. Going through similar calculations as above, from $3^k < n$ we obtain $3k + 1 < 3 \log_3 n + 1$. Similarly, in the case $t(C_n) \leq 3k + 2$, when $n \in (4 \cdot 3^{k-1}, 2 \cdot 3^k]$ for some $k \geq 1$, we obtain $3k + 2 < 3 \log_3 n + 5 - 3 \log_3 4$. Hence, from the above three cases, we have $t(C_n) \leq 3 \log_3 n + 2$. Since $3 \log_3 n = \frac{3}{\log_2 3} \log_2 n \leq 1.893 \log_2(n)$, we get the desired result. ■

Example 4.4.13. An example of a $C_{36}\text{-CFF}$ and a $C_{54}\text{-CFF}$ built on the ground sets $[1, 10]$ and $[1, 11]$, respectively, is given in Figure 4.7. They are also $P_{36}\text{-CFF}$ and $P_{54}\text{-CFF}$, respectively. We also provide the reflected Gray code $R_4^{(2,2,3,3)}$ used to construct the $C_{36}\text{-CFF}$ in Figure 4.7a. In contrast, we present the $C_{54}\text{-CFF}(11, 54)$ in Figure 4.7 without including the corresponding reflected Gray code $R_4^{(2,3,3,3)}$. We also provide an example of a $P_{27}\text{-CFF}$ (using $R_3^{(3,3,3)}$) and a $C_{27}\text{-CFF}$ (using M_3^3) built on $[1, 9]$ in Figure 4.8. Figure 4.9 provides an example of a $C_{19}\text{-CFF}(9, 19)$ by shortening M_3^3 .

We further show that the set system produced in Construction 4.4.7 is also a CFF on a generalized Hamming graph. Let $H = K_{a_1} \square K_{a_2} \square \dots \square K_{a_d}$ with $V(K_{a_l}) = \{0, 1, \dots, a_l - 1\}$ for $l \in [1, d]$. Then, $V(H)$ is a set of all d -tuples in $V(K_{a_1}) \times V(K_{a_2}) \times \dots \times V(K_{a_d})$ and, for a pair of vertices $(x_1, x_2, \dots, x_d), (y_1, y_2, \dots, y_d) \in V(H)$, we have that $\{(x_1, x_2, \dots, x_d), (y_1, y_2, \dots, y_d)\} \in E(H)$ if and only if for some $i \in [1, d]$, we have $x_i \neq y_i$ with $\{x_i, y_i\} \in E(K_{a_i})$, and for each $j \in [1, d] \setminus \{i\}$, we have $x_j = y_j$. In other words, $K_{a_1} \square K_{a_2} \square \dots \square K_{a_d}$ has the vertex set consisting of codewords of $G_d^{(a_1, a_2, \dots, a_d)}$ and $\{(x_1, x_2, \dots, x_d), (y_1, y_2, \dots, y_d)\} \in E(K_{a_1} \square K_{a_2} \square \dots \square K_{a_d})$ if and only if the Hamming distance between $(x_1, x_2, \dots, x_d)$ and $(y_1, y_2, \dots, y_d)$ is 1.

$$[1, 10] = \{1, 2\} \cup \{3, 4\} \cup \{5, 6, 7\} \cup \{8, 9, 10\}.$$

$$[1, 11] = \{1, 2\} \cup \{3, 4, 5\} \cup \{6, 7, 8\} \cup \{9, 10, 11\}.$$

(0) 0000	(0) {1, 3, 5, 8}
(1) 0001	(1) {1, 3, 5, 9}
(2) 0002	(2) {1, 3, 5, 10}
(3) 0012	(3) {1, 3, 6, 10}
(4) 0011	(4) {1, 3, 6, 9}
(5) 0010	(5) {1, 3, 6, 8}
(6) 0020	(6) {1, 3, 7, 8}
(7) 0021	(7) {1, 3, 7, 9}
(8) 0022	(8) {1, 3, 7, 10}
(9) 0122	(9) {1, 4, 7, 10}
(10) 0121	(10) {1, 4, 7, 9}
(11) 0120	(11) {1, 4, 7, 8}
(12) 0110	(12) {1, 4, 6, 8}
(13) 0111	(13) {1, 4, 6, 9}
(14) 0112	(14) {1, 4, 6, 10}
(15) 0102	(15) {1, 4, 5, 10}
(16) 0101	(16) {1, 4, 5, 9}
(17) 0100	(17) {1, 4, 5, 8}
(18) 1100	(18) {2, 4, 5, 8}
(19) 1101	(19) {2, 4, 5, 9}
(20) 1102	(20) {2, 4, 5, 10}
(21) 1112	(21) {2, 4, 6, 10}
(22) 1111	(22) {2, 4, 6, 9}
(23) 1110	(23) {2, 4, 6, 8}
(24) 1120	(24) {2, 4, 7, 8}
(25) 1121	(25) {2, 4, 7, 9}
(26) 1122	(26) {2, 4, 7, 10}
(27) 1022	(27) {2, 3, 7, 10}
(28) 1021	(28) {2, 3, 7, 9}
(29) 1020	(29) {2, 3, 7, 8}
(30) 1010	(30) {2, 3, 6, 8}
(31) 1011	(31) {2, 3, 6, 9}
(32) 1012	(32) {2, 3, 6, 10}
(33) 1002	(33) {2, 3, 5, 10}
(34) 1001	(34) {2, 3, 5, 9}
(35) 1000	(35) {2, 3, 5, 8}

(0) {1, 3, 6, 9}	(27) {2, 5, 8, 11}
(1) {1, 3, 6, 10}	(28) {2, 5, 8, 10}
(2) {1, 3, 6, 11}	(29) {2, 5, 8, 9}
(3) {1, 3, 7, 11}	(30) {2, 5, 7, 9}
(4) {1, 3, 7, 10}	(31) {2, 5, 7, 10}
(5) {1, 3, 7, 9}	(32) {2, 5, 7, 11}
(6) {1, 3, 8, 9}	(33) {2, 5, 6, 11}
(7) {1, 3, 8, 10}	(34) {2, 5, 6, 10}
(8) {1, 3, 8, 11}	(35) {2, 5, 6, 9}
(9) {1, 4, 8, 11}	(36) {2, 4, 6, 9}
(10) {1, 4, 8, 10}	(37) {2, 4, 6, 10}
(11) {1, 4, 8, 9}	(38) {2, 4, 6, 11}
(12) {1, 4, 7, 9}	(39) {2, 4, 7, 11}
(13) {1, 4, 7, 10}	(40) {2, 4, 7, 10}
(14) {1, 4, 7, 11}	(41) {2, 4, 7, 9}
(15) {1, 4, 6, 11}	(42) {2, 4, 8, 9}
(16) {1, 4, 6, 10}	(43) {2, 4, 8, 10}
(17) {1, 4, 6, 9}	(44) {2, 4, 8, 11}
(18) {1, 5, 6, 9}	(45) {2, 3, 8, 11}
(19) {1, 5, 6, 10}	(46) {2, 3, 8, 10}
(20) {1, 5, 6, 11}	(47) {2, 3, 8, 9}
(21) {1, 5, 7, 11}	(48) {2, 3, 7, 9}
(22) {1, 5, 7, 10}	(49) {2, 3, 7, 10}
(23) {1, 5, 7, 9}	(50) {2, 3, 7, 11}
(24) {1, 5, 8, 9}	(51) {2, 3, 6, 11}
(25) {1, 5, 8, 10}	(52) {2, 3, 6, 10}
(26) {1, 5, 8, 11}	(53) {2, 3, 6, 9}

(a) $R_4^{(2,2,3,3)}$ and the corresponding C_{36} -CFF(10, 36) built on $[1, 10]$ (b) A C_{54} -CFF(11, 54) built on $[1, 11]$ using $R_4^{(2,3,3,3)}$

Figure 4.7: Examples of a C_{36} -CFF and a C_{54} -CFF

Definition 4.4.14. Let $\mathcal{F}$ be a set system consisting of $n \geq 3$ subsets of a t -set. A graph G is the *CFF-maximal graph* for $\mathcal{F}$ if it is the maximal graph G on n vertices such that $\mathcal{F}$ is a G -CFF(t, n). More specifically, $G = ([1, n], E)$ where

$$E = \left\{ e \in \binom{[1, n]}{2} : \mathcal{F} \text{ is cover-free and in-cover-free for } e \right\}.$$

Proposition 4.4.15. Let $G_k^{(m_1, m_2, \dots, m_k)}$ be a Gray code and let t, n , and $\mathcal{F} \left(G_k^{(m_1, m_2, \dots, m_k)} \right)$ be defined as in Construction 4.4.7. The CFF-maximal graph $\mathcal{H}$ for $\mathcal{F} \left(G_k^{(m_1, m_2, \dots, m_k)} \right)$ is the Hamming graph $\mathcal{H} = H_{m_1, m_2, \dots, m_k}$. In addition, $t(C_n) \leq t(H_{m_1, m_2, \dots, m_k}) \leq \sum_{i=1}^k m_i$.

$$[1, 9] = \{1, 2, 3\} \cup \{4, 5, 6\} \cup \{7, 8, 9\}.$$

(0) {1, 4, 7}	(9) {2, 6, 9}	(18) {3, 4, 7}
(1) {1, 4, 8}	(10) {2, 6, 8}	(19) {3, 4, 8}
(2) {1, 4, 9}	(11) {2, 6, 7}	(20) {3, 4, 9}
(3) {1, 5, 9}	(12) {2, 5, 7}	(21) {3, 5, 9}
(4) {1, 5, 8}	(13) {2, 5, 8}	(22) {3, 5, 8}
(5) {1, 5, 7}	(14) {2, 5, 9}	(23) {3, 5, 7}
(6) {1, 6, 7}	(15) {2, 4, 9}	(24) {3, 6, 7}
(7) {1, 6, 8}	(16) {2, 4, 8}	(25) {3, 6, 8}
(8) {1, 6, 9}	(17) {2, 4, 7}	(26) {3, 6, 9}

(0) {1, 4, 7}	(9) {2, 6, 7}	(18) {3, 5, 7}
(1) {1, 4, 8}	(10) {2, 6, 8}	(19) {3, 5, 8}
(2) {1, 4, 9}	(11) {2, 6, 9}	(20) {3, 5, 9}
(3) {1, 5, 9}	(12) {2, 4, 9}	(21) {3, 6, 9}
(4) {1, 5, 7}	(13) {2, 4, 7}	(22) {3, 6, 7}
(5) {1, 5, 8}	(14) {2, 4, 8}	(23) {3, 6, 8}
(6) {1, 6, 8}	(15) {2, 5, 8}	(24) {3, 4, 8}
(7) {1, 6, 9}	(16) {2, 5, 9}	(25) {3, 4, 9}
(8) {1, 6, 7}	(17) {2, 5, 7}	(26) {3, 4, 7}

(a) A P_{27} -CFF(9, 27) built on $[1, 9]$ using $R_3^{(3,3,3)}$

(b) A C_{27} -CFF(9, 27) and a P_{27} -CFF(9, 27) built on $[1, 9]$ using M_3^3

Figure 4.8: Examples of a P_{27} -CFF and a C_{27} -CFF

$$[1, 9] = \{1, 2, 3\} \cup \{4, 5, 6\} \cup \{7, 8, 9\}.$$

(0)	{1, 4, 7}	(0)	{1, 4, 7}
(1)	{1, 4, 8}	(2)	{1, 4, 9}
(2)	{1, 4, 9}	(3)	{1, 5, 9}
(3)	{1, 5, 9}	(5)	{1, 5, 8}
(4)	{1, 5, 7}	(6)	{1, 6, 8}
(5)	{1, 5, 8}	(8)	{1, 6, 7}
(6)	{1, 6, 8}	(9)	{2, 6, 7}
(7)	{1, 6, 9}	(10)	{2, 6, 8}
(8)	{1, 6, 7}	(11)	{2, 6, 9}
(9)	{2, 6, 7}	(12)	{2, 4, 9}
(10)	{2, 6, 8}	(13)	{2, 4, 7}
(11)	{2, 6, 9}	(14)	{2, 4, 8}
(12)	{2, 4, 9}	(15)	{2, 5, 8}
(13)	{2, 4, 7}	(16)	{2, 5, 9}
(14)	{2, 4, 8}	(17)	{2, 5, 7}
(15)	{2, 5, 8}	(18)	{3, 5, 7}
(16)	{2, 5, 9}	(19)	{3, 5, 8}
(17)	{2, 5, 7}	(20)	{3, 5, 9}
(18)	{3, 5, 7}	(21)	{3, 6, 9}
(19)	{3, 5, 8}	(22)	{3, 6, 7}
(20)	{3, 5, 9}	(23)	{3, 6, 8}
(21)	{3, 6, 9}	(24)	{3, 4, 8}
(22)	{3, 6, 7}	(25)	{3, 4, 9}
(23)	{3, 6, 8}	(26)	{3, 4, 7}
(24)	{3, 4, 8}		
(25)	{3, 4, 9}		
(26)	{3, 4, 7}		

(a) A C_{27} -CFF(9, 27) built on $[1, 9]$ using M_3^3 (b) A C_{19} -CFF(9, 19) built on $[1, 9]$ using $(M_3^3)^8$. $(M_3^3)^8$ is constructed by deleting the first 8 highlighted codewords in 4.9a.Figure 4.9: A C_{27} -CFF(9, 27) and a C_{19} -CFF(9, 19)

Proof. Let $\mathcal{H}$ be the Hamming graph $H_{m_1, m_2, \dots, m_k}$ with $V(H_{m_1, m_2, \dots, m_k}) = \mathbb{Z}_{m_1} \times \dots \times \mathbb{Z}_{m_k}$. Let $e = \binom{[1, n]}{2}$, where $n = \prod_{i=1}^k m_i$. If $e = \{D_i, D_j\} \in E(H_{m_1, m_2, \dots, m_k})$, then the Hamming distance between D_i and D_j is exactly 1, by Lemma 4.4.8, for any $D_l \neq D_i, D_j$, we have $F(D_l) \not\subseteq F(D_i) \cup F(D_j)$. Also, $F(D_i) \not\subseteq F(D_j)$ and $F(D_j) \not\subseteq F(D_i)$ because $|F(D_i)| = |F(D_j)|$. Thus, $\mathcal{F}$ is cover-free for e and in-cover-free for e . On the other hand, if $e' = \{D_i, D_j\} \notin E(H_{m_1, m_2, \dots, m_k})$, then the Hamming distance between D_i and D_j is strictly greater than 1. Without loss of generality, let us assume that D_i and D_j differ in the 1st and 2nd co-ordinates. Thus, D_i and D_j must be of the forms $D_i = (d_1, d_2, d_3, \dots, d_k)$ and $D_j = (d'_1, d'_2, d'_3, \dots, d'_k)$ with $d_l, d'_l \in \mathbb{Z}_{m_l}, l \in [1, k]$, and $d_1 \neq d'_1, d_2 \neq d'_2$. Let $D_y = (d_1, d'_2, d_3, \dots, d_k)$. Then, $F(D_y) \subseteq F(D_i) \cup F(D_j)$, which implies that $\mathcal{F}$ is not cover-free for e' . This shows that $H_{m_1, m_2, \dots, m_k}$ is the CFF-maximal graph for $\mathcal{F}$, and that $t(H_{m_1, m_2, \dots, m_k}) \leq \sum_{i=1}^k m_i$. Since $H_{m_1, m_2, \dots, m_k}$ is Hamiltonian, by Proposition 4.2.6, $t(C_n) \leq t(H_{m_1, m_2, \dots, m_k})$. ■

Next, we present Construction 4.4.16, which provides the bounds in Proposition 4.4.17.

Construction 4.4.16. Let A be a $t_1 \times n$ binary matrix and I_2 be the 2×2 identity matrix.

Let B be a matrix with columns of A in the reverse order. Using A and I_2 , we create a $(t_1 + 2) \times 2n$ binary matrix C by where columns are labeled by $1, 2, \dots, n, n+1, n+2, \dots, 2n$. The construction is as follows. We copy the matrices A and B under vertices $1, \dots, n$ and $n+1, \dots, 2n$, respectively and call the resultant matrix C_1 . We copy the first column and second column of I_2 under vertices labeled $1, \dots, n$ and $n+1, \dots, 2n$, respectively and call the resultant matrix C_2 . Let C be the vertical concatenation of C_1 and C_2 . We label the rows of C as the elements of $[1, t_1 + 2]$ with rows labeled $[1, t_1]$ corresponding to the rows of C_1 and rows labeled $[t_1 + 1, t_1 + 2]$ corresponding to the rows of C_2 .

Proposition 4.4.17. $t(C_{2n}) \leq t(C_n) + 2$ and $t(P_{2n}) \leq t(P_n) + 2$.

Proof. Let A be a $t_1 \times n$ $\overline{C_n}$ -disjunct matrix and B be the matrix with columns of A in the reverse order. We claim that the matrix C , obtained using Construction 4.4.16, is a $(t_1 + 2) \times 2n$ $\overline{C_{2n}}$ -disjunct matrix. Indeed, since A is cover-free for an edge $\{i, i+1\}$ with $i \in [1, n-1]$ and furthermore, $C[t_1 + 2, i] = C[t_1 + 2, i+1] = 0$ and $C[t_1 + 2, j] = 1$ for all $j \in [n+1, 2n]$, this implies that C is cover-free for $\{i, i+1\}$. Using a similar argument, we can conclude that C is cover-free for the edge $\{j, j+1\}$ for all $j \in [n+1, 2n-1]$. Since by Theorem 4.2.7, A is $\overline{1}$ -disjunct and for the edge $\{n, n+1\}$, we have $C_1^n = C_1^{n+1} = A^n$. Then for any $j \in [1, 2n] \setminus \{n, n+1\}$, there exists a row $l_1 \in [1, t_1]$ in C_1 such that $C_1[l_1, n] = C_1[l_1, n+1] = 0$ and $C_1[l_1, j] = 1$. Thus, C is cover-free for the edge $\{n, n+1\}$. Using a similar argument, we can also show that C is cover-free for the edge $\{1, 2n\}$. Thus, C is $\overline{C_{2n}}$ -disjunct. Since $\delta(C_{2n}) \geq 2$, by Proposition 4.2.10, C is $\overline{C_{2n}}$ -disjunct. Thus, $t(C_{2n}) \leq t(C_n) + 2$. A similar construction and proof can be applied to a $t_1 \times n$ $\overline{P_n}$ -disjunct matrix A to obtain $t(P_n) \leq t(P_n) + 2$. ■

Next, using $t(C_n)$, we provide an upper and lower bound of $t(W_n)$.

Corollary 4.4.18. For $n \geq 3$, $\max\{t(1, n) + 1, t(C_n), t(C_{n+1})\} \leq t(W_{n+1}) \leq t(C_n) + 1$.

Proof. Since W_{n+1} is hamiltonian, by Proposition 4.2.6, $t(C_{n+1}) \leq t(W_{n+1})$, and since $W_{n+1} = C_n + \{0\}$, by Theorem 4.2.13, $t(1, n) + 1 \leq t(W_{n+1}) \leq t(C_n) + 1$. Since $C_n \subseteq W_{n+1}$, by Proposition 4.2.6, we have $t(C_n) \leq t(W_{n+1})$. Thus, we have the desired bound. ■

4.5 Generalizing the construction of CFFs on star graphs

We generalize the construction of CFFs on star graphs to obtain CFFs for other families of graphs, such as windmill graphs, friendship graphs, and wheel graphs.

Definition 4.5.1. Let $k, n \geq 2$. The *windmill graph* $\text{Wd}(k, n)$ is an undirected graph constructed by joining n copies of the complete graph K_k at a shared universal vertex. Thus, $\text{Wd}(k, n)$ has $n(k-1) + 1$ vertices. The *friendship graph* is $\text{Wd}(3, n)$, which has $2n + 1$ vertices and is denoted by F_{2n+1} .

Thus, the star graph S_{n+1} is $\text{Wd}(2, n)$.

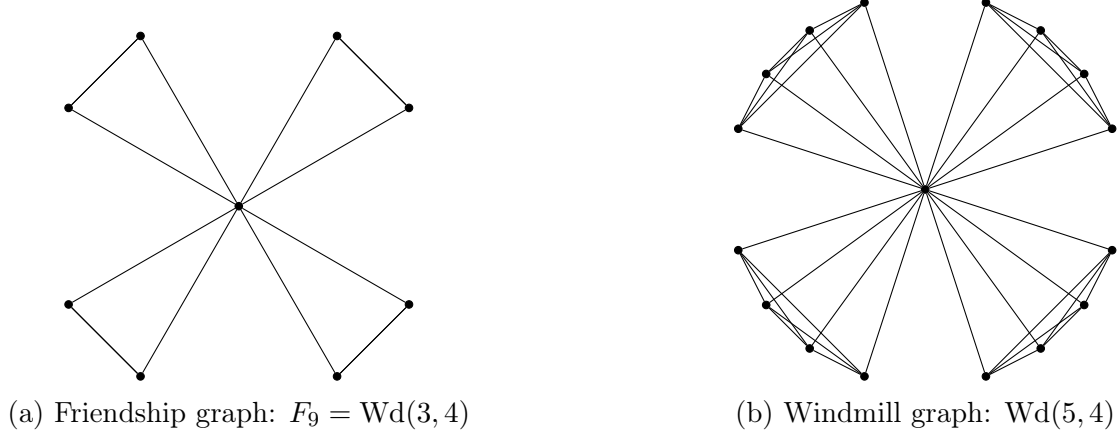


Figure 4.10: Examples of a friendship graph and a windmill graph

	v_0	$v_{1,1} \ v_{1,2} \ \dots \ v_{1,k-1}$	$v_{2,1} \ v_{2,2} \ \dots \ v_{2,k-1}$	$\dots$	$v_{n,1} \ v_{n,2} \ \dots \ v_{n,k-1}$
$t(1, n)$	0	<i>Column 1 of A</i>	<i>Column 2 of A</i>	$\dots$	<i>Column n of A</i>
+					
t_2	0	M	M	$\dots$	M
+					
1	1	0 0 $\dots$ 0 0	0 0 $\dots$ 0 0	$\dots$	0 0 $\dots$ 0 0

 Figure 4.11: A $\overline{\text{Wd}(k, n)}$ -disjunct

Example 4.5.2. We give an example of a friendship graph and a windmill graph in Figures 4.10a and 4.10b.

Below we give a construction for a $\overline{\text{Wd}(k, n)}$ -disjunct matrix.

Construction 4.5.3. Let $\text{Wd}(k, n)$ be a windmill graph with $k \geq 3$ and $n \geq 2$. Let v_0 be the universal vertex and for $1 \leq i \leq n$, let the remaining vertices of the i^{th} subgraph K_{k-1} be $v_{i,1}, v_{i,2}, \dots, v_{i,k-1}$. Let A be a $t(1, n) \times n$ 1-disjunct matrix and let M be a $t(2, k-1) \times (k-1)$ 2-disjunct matrix if $k \geq 4$, or a 2×2 identity matrix if $k = 3$. Let t_2 be $t(2, k-1)$ if $k \geq 4$, and 2 if $k = 3$.

Let $\mathcal{C}_1, \mathcal{C}_2$, and $\mathcal{C}_3$ be matrices where each column is labeled with the vertices of the graph $\text{Wd}(k, n)$ as described next. In $\mathcal{C}_1$, $\mathcal{C}_1^{v_0} = \mathbf{0}_{t(1,n) \times 1}$ and $\mathcal{C}_1^{v_{i,j}} = A^i$ for all $j \in [1, k-1]$. In $\mathcal{C}_2$, $\mathcal{C}_2^{v_0} = \mathbf{0}_{t_2 \times 1}$ and for all $i \in [1, n]$, $\mathcal{C}_2^{v_{i,j}} = M^j$ with $j \in [1, k-1]$. $\mathcal{C}_3$ is the row matrix such that $\mathcal{C}_3[1, v_0] = 1$ and $\mathcal{C}_3[1, v_{i,j}] = 0$ for all $i \in [1, n]$ and $j \in [1, k-1]$. Let $\mathcal{C}$ be the vertical concatenation of $\mathcal{C}_1, \mathcal{C}_2$, and $\mathcal{C}_3$, as shown in Figure 4.11.

Theorem 4.5.4. Let $n \geq 2$. For $k \geq 4$,

$$t(1, (k-1)n) + 1 \leq t(\text{Wd}(k, n)) \leq t(1, n) + t(2, k-1) + 1.$$

For $k = 3$,

$$t(1, 2n) + 1 \leq t(\text{Wd}(3, n)) \leq t(1, n) + 3.$$

Proof. We show that the matrix $\mathcal{C}$ created in Construction 4.5.3 is $\overline{\text{Wd}(k, n)}$ -disjunct. Since $\delta(\text{Wd}(k, n)) \geq 2$, by Theorem 4.2.11 part 1, it is enough to prove that $\mathcal{C}$ is $\text{Wd}(k, n)$ -disjunct. Let $\{u, v\}$ be an edge in $\text{Wd}(k, n)$ for $k \geq 3$. We need to show that for any other vertex w , there exists a row r with $\mathcal{C}[r, u] = \mathcal{C}[r, v] = 0$ and $\mathcal{C}[r, w] = 1$.

Case 1: $\{u, v\} = \{v_0, v_{i,j}\}$ for $i \in [1, n]$ and $j \in [1, k-1]$,

Case 1.1: $w = v_{l,m}$, where $l \in [1, n], m \in [1, k-1]$ and $l \neq i$. According to Construction 4.5.3, $\mathcal{C}_1^{v_0} = \mathbf{0}_{t(1,n) \times 1}$, $\mathcal{C}_1^{v_{i,j}} = A^i$, and $\mathcal{C}_1^{v_{l,m}} = A^l$. Since A is 1-disjunct, there is a row r in $\mathcal{C}_1$ (thus, in $\mathcal{C}$) such that $\mathcal{C}_1[r, v_0] = \mathcal{C}_1[r, v_{i,j}] = 0$ and $\mathcal{C}_1[r, v_{l,m}] = 1$.

Case 1.2: $w = v_{i,m}$, where $m \in [1, k-1]$ and $m \neq j$. According to Construction 4.5.3, $\mathcal{C}_2^{v_0} = \mathbf{0}_{t_2 \times 1}$, $\mathcal{C}_2^{v_{i,j}} = M^j$, and $\mathcal{C}_2^{v_{i,m}} = M^m$. Since M is also 1-disjunct, there is a row r in $\mathcal{C}_2$ such that $\mathcal{C}_2[r, v_0] = \mathcal{C}_2[r, v_{i,j}] = 0$ and $\mathcal{C}_2[r, v_{i,m}] = 1$.

Case 2: $\{u, v\} = \{v_{i,j}, v_{i,m}\}$ for $i \in [1, n]$ and $j, m \in [1, k-1]$ with $j \neq m$.

Case 2.1: $w = v_0$. It is easy to see that $\mathcal{C}_3[1, v_{i,j}] = \mathcal{C}_3[1, v_{i,m}] = 0$ and $\mathcal{C}_3[1, v_0] = 1$.

Case 2.2: $w = v_{i,p}$ with $p \in [1, k-1]$ and $p \neq j, m$. Note that in this case $k \geq 4$. Since $\mathcal{C}_2^{v_{i,j}} = M^j$, $\mathcal{C}_2^{v_{i,m}} = M^m$, and $\mathcal{C}_2^{v_{i,p}} = M^p$ and M is 2-disjunct, there exists a row r in $\mathcal{C}_2$ such that $\mathcal{C}_2[r, v_{i,j}] = \mathcal{C}_2[r, v_{i,m}] = 0$ and $\mathcal{C}_2[r, v_{i,p}] = 1$.

Case 2.3: $w = v_{l,p}$ for $l \in [1, n], p \in [1, k-1]$ with $l \neq i$. Since $\mathcal{C}_1^{v_{i,j}} = \mathcal{C}_1^{v_{i,m}} = A^i$, and $\mathcal{C}_1^{v_{l,p}} = A^l$ and A is 1-disjunct, there is a row r in $\mathcal{C}_1$ (hence, in $\mathcal{C}$) such that $\mathcal{C}_1[r, v_{i,j}] = \mathcal{C}_1[r, v_{i,m}] = 0$ and $\mathcal{C}_1[r, v_{l,p}] = 1$.

Thus, we have $t(\text{Wd}(k, n)) \leq t(1, n) + t(2, k-1) + 1$. Since $S_{(k-1)n+1} \subseteq \text{Wd}(k, n)$, by Proposition 4.2.6 and $t(S_{(k-1)n+1}) = t(1, (k-1)n) + 1$, by Corollary 4.3.4, we have $t(1, (k-1)n) + 1 \leq t(\text{Wd}(k, n))$. ■

Corollary 4.5.5. If $n > \lfloor \frac{1}{2}(\overline{n} + 1) \rfloor$, then $t(F_{2n+1}) = t(1, n) + 3$. Otherwise, $t(1, n) + 2 \leq t(F_{2n+1}) \leq t(1, n) + 3$.

Proof. Since F_{2n+1} is a $\text{Wd}(3, n)$, by Theorem 4.5.4, we obtain $t(1, 2n) + 1 \leq t(F_{2n+1}) \leq t(1, n) + 3$. If $n > \lfloor \frac{1}{2}(\overline{n} + 1) \rfloor$, then by Proposition 3.2.8, $t(1, 2n) = t(1, n) + 2$ and we obtain $t(F_{2n+1}) = t(1, n) + 3$. If $n \leq \lfloor \frac{1}{2}(\overline{n} + 1) \rfloor$, then by Proposition 3.2.8, $t(1, 2n) = t(1, n) + 1$ and we obtain $t(1, n) + 2 \leq t(F_{2n+1}) \leq t(1, n) + 3$. ■

4.6 Values of $t(P_n), t(C_n)$, and $t(W_n)$ for small n

In this section, we determine $t(P_n)$, $t(C_n)$, and $t(W_n)$ for some small values of n .

Proposition 4.6.1. If $n = 3, 4$; then $t(P_n) = t(C_n) = n$.

Proof. For $n = 3, 4$; we have $t(1, n) = n$, and also, $t(2, n) = n$ by [46]. Hence, by Theorem 4.2.8, $t(P_n) = n = t(C_n)$. ■

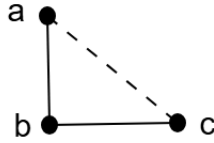
To obtain the exact values of $t(P_n)$ and $t(C_n)$ for some larger values of n , we provide Lemmas 4.6.3 and 4.6.4. We prove these two lemmas by case analysis. In the case where all subsets in a set system have size 2, we associate a graph with the set system, as explained in the next remark.

Remark 4.6.2. Let $\mathcal{F} = (\mathcal{X}, \mathcal{B})$ be a set system where $|\mathcal{X}| = t$ and for every $B \in \mathcal{B}$, we have $|B| = 2$. Let $\mathcal{M}$ be the incidence matrix associated with $\mathcal{F}$. Such $\mathcal{F}$ can be depicted as a graph G where $V(G) = \mathcal{X}$ and $E(G) = \mathcal{B}$.

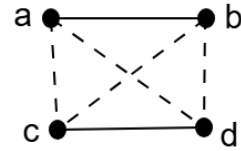
For a graph H , let $\mathcal{F}$ be H -CFF. For $\{u, v\} \in E(H)$, let $B_u = \{a, b\}$ and $B_v = \{c, d\}$. For $\{x, y\} \subseteq B_u \cup B_v$ and $\{x, y\} \neq B_u, B_v$, $\{x, y\}$ is said to be *covered by sets B_u and B_v* . If $\{x, y\}$ is covered by B_u and B_v , then by definition of H -CFF, $\{x, y\} \notin \mathcal{B}$. Hence, $\{x, y\} \notin E(G)$. The pair (B_u, B_v) can be one of two possible types:

Case 1: (Type 1 pair) $b = d$ and $a \neq b \neq c$. Then, $\{a, c\} \notin E(G)$. Figure 4.12a shows the type 1 pair $\{a, b\}$ and $\{b, c\}$ as black edges and the covered set $\{a, c\}$ as a non-edge, since $\{a, c\}$ is prohibited in $\mathcal{B}$.

Case 2: (Type 2 pair) a, b, c, d are pairwise distinct. Then, none of the sets $\{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}$ belong to $E(G)$. Figure 4.12b shows the type 2 pair $(\{a, b\}, \{c, d\})$ as black edges, and the covered subsets that are prohibited in $\mathcal{B}$ as the non-edges.



(a) Type 1 pair: $\{a, b\}$ and $\{b, c\}$



(b) Type 2 pair: $\{a, b\}$ and $\{c, d\}$

Figure 4.12: Types of pair (B_u, B_v) with $\{u, v\} \in E(H)$ as described in Remark 4.6.2

Lemma 4.6.3. Let $\mathcal{F} = (\mathcal{X}, \mathcal{B})$ be a P_n -CFF with $|\mathcal{X}| = 4$ and $n \geq 3$. Then $n \leq 4$.

Proof. Let $\mathcal{X} = \{1, 2, 3, 4\}$ and let $V(P_n) = \{v_i : i \in [1, n]\}$ and $E(P_n) = \{\{v_i, v_{i+1}\} : 1 \leq i \leq n-1\}$. We split the proof into the following three cases:

Case 1: $\mathcal{B}$ has a set B with $|B| = 3$. Without loss of generality, let $B = \{1, 2, 3\}$ and let v_i be the vertex such that $B = B_{v_i}$ for some $i \in [1, n]$. Since there is an edge $\{v_i, v_j\} \in E(P_n)$ for some $j \in \{i-1, i+1\}$, we have $4 \in B_{v_j}$. Thus, $B_{v_i} \cup B_{v_j} = \mathcal{X}$, which contains any other B_{v_l} with $l \in [1, n] \setminus \{i, j\}$ leading to a contradiction.

Case 2: $\mathcal{B}$ has a set B with $|B| = 1$. Without loss of generality, let $B = \{1\}$. By Theorem 4.2.7, $\mathcal{F}$ is a 1-CFF. Since 1 cannot be an element of other subsets in $\mathcal{F}$, $(\mathcal{X} \setminus \{1\}, \mathcal{B} \setminus \{\{1\}\})$ is also a 1-CFF. Let $\mathcal{X}' = \mathcal{X} \setminus \{1\}$ and $\mathcal{B}' = \mathcal{B} \setminus \{\{1\}\}$. Since $|\mathcal{X}'| = 3$, then by Theorem 2.1.4, $|\mathcal{B}'| \leq 3$. This implies $n = |\mathcal{B}| \leq 4$.

Case 3: $|B| = 2$ for all $B \in \mathcal{B}$. Then $\mathcal{B} \subseteq \binom{X}{2}$. Suppose, for the sake of contradiction, that $n \geq 5$. Let $\mathcal{F}' = (\mathcal{X}, \mathcal{B}')$ where $\mathcal{B}' = \{B_{v_1}, B_{v_2}, B_{v_3}, B_{v_4}, B_{v_5}\} \subseteq \mathcal{B}$. Then, $\mathcal{F}'$ is P_5 -CFF. Suppose, without loss of generality, that $\{1, 2\} \in \binom{X}{2} \setminus \mathcal{B}'$ and so, $\mathcal{B}' = \binom{X}{2} \setminus \{\{1, 2\}\}$. Let $B_{v_i} = \{3, 4\}$ for some $i \in [1, 5]$. It is easy to verify that for any $B \in \mathcal{B}' \setminus \{B_{v_i}\}$, there exists $B' \in \mathcal{B} \setminus \{B, B_{v_i}\}$ such that $B' \subseteq B_{v_i} \cup B$. Then by Remark 4.6.2, we have $B' \notin \mathcal{B}'$, which further implies that $n < 5$, a contradiction. ■

Lemma 4.6.4. Let $\mathcal{F} = (\mathcal{X}, \mathcal{B})$ be a P_n -CFF with $|\mathcal{X}| = 5$ and $n \geq 3$. Then $n \leq 6$.

Proof. Suppose, by contradiction, $\mathcal{F}$ is a P_7 -CFF(5, 7) and let A be the 5×7 $\overline{P_7}$ -disjunct matrix corresponding to $\mathcal{F}$. Let $V(P_7) = \{v_i : i \in [1, 7]\}$ and $E(P_7) = \{\{v_i, v_{i+1}\} : 1 \leq i \leq 6\}^2$. We split the proof into the following four cases:

Case 1: $\mathcal{B}$ has a set B , with $|B| = 4$. Without loss of generality, let $B = \{1, 2, 3, 4\} = B_{v_i}$ for some $i \in [1, n]$. Then, $5 \in B_{v_j}$ for $j \in \{i-1, i+1\}$. Thus, $B_{v_i} \cup B_{v_j} = [1, 5]$, which contains any other B_{v_l} with $l \in [1, n] \setminus \{i, j\}$ leading to a contradiction.

Case 2: $\mathcal{B}$ has a set B , with $|B| = 3$ and no other subset $B' \in \mathcal{B}$ exists such that $|B'| > 3$. Without loss of generality, let $B = \{1, 2, 3\}$. This case is further split into 4 subcases where $B_{v_i} = \{1, 2, 3\}$ for $i = 1, 2, 3, 4$ since the cases for $i = 5, 6, 7$ would be symmetric for P_7 .

Case 2.1: $B_{v_1} = \{1, 2, 3\}$ or $B_{v_2} = \{1, 2, 3\}$. First, assume $B_{v_1} = \{1, 2, 3\}$. Since $\{v_1, v_2\} \in E(P_7)$, then $B_{v_2} \not\subseteq B_{v_1}$ which implies that there is a row r with $A[r, v_1] = 0$ and $A[r, v_2] = 1$. Furthermore, for each $l \in \{v_3, v_4, v_5, v_6, v_7\}$ there exists a row r' with $A[r', v_1] = 0$, $A[r', v_2] = 0$, and $A[r', l] = 1$. Since the column A^{v_1} has two 0's, such row r' is exactly one row with $A[r', v_1] = 0$, $A[r', v_2] = 0$, and $A[r', l] = 1$ for all $l \in \{v_3, v_4, v_5, v_6, v_7\}$; see Figure 4.13a. Similarly, if $B_{v_2} = \{1, 2, 3\}$, following the same argument, we conclude that there exists a row r with $A[r, v_2] = 0$ and $A[r, v_1] = 1$, as well as exactly one row r' with $A[r', v_1] = 0$, $A[r', v_2] = 0$, and $A[r', l] = 1$ for all $l \in \{v_3, v_4, v_5, v_6, v_7\}$; see Figure 4.13b. Now in both cases, consider the submatrix A' obtained by restricting A to the columns $A^{v_3}, A^{v_4}, A^{v_5}, A^{v_6}$, and A^{v_7} and to the rows 1, 2, 3, and r . A' must be a 4×5 $\overline{P_5}$ -disjunct matrix, which is a contradiction by Lemma 4.6.3.

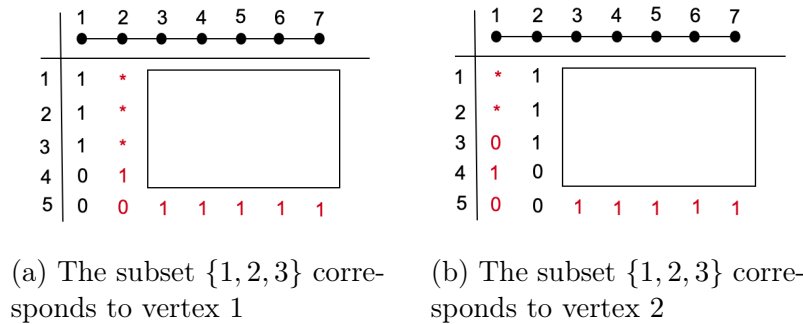


Figure 4.13: Case 2.1 of Lemma 4.6.4

²In all the figures in Lemma 4.6.4, the vertices are labeled i instead of v_i for $i \in [1, 7]$, for convenience and to save space in the figures.

Case 2.2: $B_{v_3} = \{1, 2, 3\}$. Since $\{v_2, v_3\} \in E(P_7)$; by the same argument as in Case 2.1, there exist a row $r \in \{4, 5\}$ with $A[r, v_3] = 0$ and $A[r, v_2] = 1$, and a row $r' \in \{4, 5\} \setminus \{r\}$ with $A[r', v_2] = 0$, $A[r', v_3] = 0$, and $A[r', l] = 1$ for all $l \in \{v_1, v_4, v_5, v_6, v_7\}$. Such entries are marked in red in Figures 4.14a and 4.14b where $r = 4$ and $r' = 5$. Since A is $\overline{P_7}$ -disjunct and $\{v_3, v_4\} \in E(P_7)$, the row r is the only one such that $A[r, v_3] = 0$, $A[r, v_4] = 0$, and $A[r, l'] = 1$ for all $l' \in \{v_1, v_2, v_5, v_6, v_7\}$. These entries are marked in blue in Figures 4.14a and 4.14b. Since $\{v_1, v_2\} \in E(P_7)$, for each $l'' \in \{v_3, v_4, v_5, v_6, v_7\}$, there exists a row $m \in \{1, 2, 3\}$ with $A[m, v_1] = A[m, v_2] = 0$, $A[m, l''] = 1$. We split this into two possible cases:

Case 2.2.1: Suppose there exists exactly one row m such that $A[m, v_1] = A[m, v_2] = 0$, $A[m, l''] = 1$ for all $l'' \in \{v_3, v_4, v_5, v_6, v_7\}$ (See the entries in green in Figure 4.14a where $m = 3$). The submatrix obtained by restricting the columns to A^{v_5} , A^{v_6} , and A^{v_7} and to the rows $m'' \in \{1, 2, 3\} \setminus \{m\}$ must form a 2×3 $\overline{P_3}$ -disjunct matrix, which is a contradiction by Proposition 4.6.1.

Case 2.2.2: Assume there are two such rows $m_1, m_2 \in \{1, 2, 3\}$ within which the condition $A[m_i, v_1] = A[m_i, v_2] = 0$, $A[m_i, l''] = 1$ is satisfied for $l'' \in \{v_3, v_4, v_5, v_6, v_7\}$ and $i \in \{1, 2\}$. Then, the only remaining row m_3 must satisfy the condition $A[m_3, v_1] = 0$ and $A[m_3, v_2] = 1$ (See the green entries in Figure 4.14b). However, $\{v_6, v_7\} \in E(P_7)$ and $B_{v_1} \subseteq B_{v_6} \cup B_{v_7}$, which is a contradiction.

	1	2	3	4	5	6	7
1	•	•	•	•	•	•	•
2							
3							
4							
5							

	1	2	3	4	5	6	7
1	•	•	•	•	•	•	•
2							
3							
4							
5							

	1	2	3	4	5	6	7
1							
2							
3	0	0	1	1	1	1	1
4	1	1	0	0	1	1	1
5	1	0	0	1	1	1	1

	1	2	3	4	5	6	7
1	•	•	•	•	•	•	•
2							
3							
4							
5							

	1	2	3	4	5	6	7
1	0	0	1				
2	0	1	1				
3	0	0	1				
4	1	1	0	0	1	1	1
5	1	0	0	1	1	1	1

(a) Case 2.2.1 of Lemma 4.6.4 (b) Case 2.2.2 of Lemma 4.6.4

Figure 4.14: Case 2.2 of Lemma 4.6.4

Case 2.3: Let $B_{v_4} = \{1, 2, 3\}$. Since $\{v_3, v_4\} \in E(P_7)$, there exist a row $r \in \{4, 5\}$ with $A[r, v_4] = 0$ and $A[r, v_3] = 1$, and a row $r' \in \{4, 5\} \setminus \{r\}$ with $A[r', v_3] = 0$, $A[r', v_4] = 0$, and $A[r', l] = 1$ for all $l \in \{v_1, v_2, v_5, v_6, v_7\}$ (notice the red entries in Figure 4.15 where $r = 4$ and $r' = 5$). Furthermore, since $\{v_4, v_5\} \in E(P_7)$, the row r is the one satisfying $A[r, v_4] = 0$, $A[r, v_5] = 0$, and $A[r, l'] = 1$ for all $l' \in \{v_1, v_2, v_3, v_6, v_7\}$. These entries are marked in blue. Since $\{v_6, v_7\} \in E(P_7)$, there exist rows $r_1, r_2 \in \{1, 2, 3\}$ such that $A[r_1, v_6] = 0$, $A[r_1, v_7] = 1$ and $A[r_2, v_6] = 1$, $A[r_2, v_7] = 0$; as well as exactly one row $r_3 \in \{1, 2, 3\} \setminus \{r_1, r_2\}$ with $A[r_3, v_6] = A[r_3, v_7] = 0$, and $A[r_3, l''] = 1$ for all $l'' \in \{v_1, v_2, v_3, v_4, v_5\}$ (these entries are specified in purple). The row r_1 must satisfy the condition $A[r_1, v_5] = A[r_1, v_6] = 0$ and $A[r_1, l'''] = 1$ for all $l''' \in \{v_1, v_2, v_3, v_4, v_7\}$ since $\{v_5, v_6\} \in E(P_7)$ (notice the green entries). Since $|B_{v_1}| \geq 4 > 3$, this is a contradiction to our initial assumption that no subset in $\mathcal{B}$ other than B_{v_4} has size > 3 .

Case 3: $\mathcal{B}$ has a set B with $|B| = 1$ and no other subset $B' \in \mathcal{B}$ exists such that $|B'| \geq 3$. Without loss of generality, let $B = \{1\}$ and $B = B_{v_i}$ for some $i \in [1, 7]$. Since $\overline{P_7}$ -disjunct

	1	2	3	4	5	6	7
1				1		1	0
2	1	1	1	1	0	0	1
3	1	1	1	1	1	0	0
4	1	1	1	0	0	1	1
5	1	1	0	0	1	1	1

Figure 4.15: Case 2.3 of Lemma 4.6.4

is also 1-disjunct, $1 \notin B_{v_j}$ for all $j \in [1, 7] \setminus \{i\}$. Now, consider the submatrix A' formed by the rows 2, 3, 4, 5 of A and the columns $\{A^{v_j} : j \in [1, 7] \setminus \{i\}\}$. Then the 4×6 matrix A' is also 1-disjunct, which implies that $\{B_{v_j} : j \in [1, 7] \setminus \{i\}\} = \binom{[2, 3, 4, 5]}{2}$. It is easy to verify by Remark 4.6.2 that for any $B_j, B_k \in \mathcal{B} \setminus \{B_i\}$, there exists $B_l \in \mathcal{B} \setminus \{B_i\}$ such that $B_l \subseteq B_j \cup B_k$, which is a contradiction.

Case 4: $|B| = 2$ for all $B \in \mathcal{B}$. Let A be the 5×7 $\overline{P}_7$ -disjunct matrix obtained by restricting the incidence matrix of a 1-CFF(5, 10) to the first 7 columns. In other words, three sets must be removed from the 1-CFF(5, 10). The structure of $\mathcal{F}$ is then divided into four subcases, according to the type of sets removed, as provided in Figure 4.16.



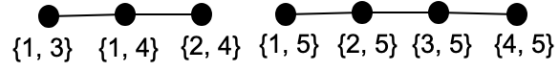
Figure 4.16: Type of the three subsets removed from 1-CFF(5, 10) as discussed in Case 4 of Lemma 4.6.4

Case 4.1: Without loss of generality, the subsets removed are $\{1, 2\}$, $\{1, 3\}$, and $\{1, 4\}$. The subsets in $\mathcal{F}$ are $\{1, 5\}$, $\{2, 3\}$, $\{2, 4\}$, $\{2, 5\}$, $\{3, 4\}$, $\{3, 5\}$, and $\{4, 5\}$. Let $B_{v_1} = \{2, 3\}$. For any $B \in \mathcal{B} \setminus \{B_{v_1}\}$, if $B_{v_2} = B$; then it is easy to see that there exists $B' \in \mathcal{B}$ such that $B' \subseteq B_{v_1} \cup B_{v_2}$, which is a contradiction by Remark 4.6.2.

Case 4.2: Without loss of generality, the subsets removed are $\{1, 2\}$, $\{2, 3\}$, and $\{1, 3\}$. The remaining subsets are $\{1, 4\}$, $\{1, 5\}$, $\{2, 4\}$, $\{2, 5\}$, $\{3, 4\}$, $\{3, 5\}$, and $\{4, 5\}$. Let $B_{v_1} = \{4, 5\}$. For any $B \in \mathcal{B} \setminus \{B_{v_1}\}$ with $B_{v_2} = B$; there exists $B' \in \mathcal{B}$ such that $B' \subseteq B_{v_1} \cup B_{v_2}$, which is a contradiction by Remark 4.6.2.

Case 4.3: Without loss of generality, the subsets removed are $\{1, 2\}$, $\{1, 3\}$, and $\{4, 5\}$. The remaining subsets are $\{1, 4\}$, $\{1, 5\}$, $\{2, 3\}$, $\{2, 4\}$, $\{2, 5\}$, $\{3, 4\}$, and $\{3, 5\}$. Let $B_{v_1} = \{2, 3\}$. For any choice of $B \in \mathcal{B} \setminus \{B_{v_1}\}$ for B_{v_2} , there exists a subset $B' \in \mathcal{B}$ covered by B_{v_1} and B . This is a contradiction by Remark 4.6.2.

Case 4.4: Without loss of generality, the subsets removed are $\{1, 2\}$, $\{2, 3\}$, and $\{3, 4\}$. The remaining subsets are $\{1, 3\}$, $\{1, 4\}$, $\{1, 5\}$, $\{2, 4\}$, $\{2, 5\}$, $\{3, 5\}$, and $\{4, 5\}$. This case, however, is a little different from the above cases. Notice that that for every $B \in \mathcal{B}$, there

Figure 4.17: Maximal graph H in Case 4.4 of Lemma 4.6.4

exists a $B' \in \mathcal{B}$ such that there exists a subset $\{x, y\} \subseteq B \cup B'$ with $\{x, y\} \notin \mathcal{B}$. We list all pairs (B, B') below:

$$\bullet (\{1, 3\}, \{1, 4\}) \quad \bullet (\{1, 4\}, \{2, 4\}) \quad \bullet (\{1, 5\}, \{2, 5\}) \quad \bullet (\{2, 5\}, \{3, 5\}) \quad \bullet (\{3, 5\}, \{4, 5\})$$

Hence, Figure 4.17 shows the maximal graph $H = (V, E)$ such that $\mathcal{F}$ is H -CFF with each vertex v_i corresponding to a subset B_{v_i} with $i \in [1, 7]$ in $\mathcal{F}$. This is a contradiction since $H \subseteq P_7$ and $H \neq P_7$.

Hence, in all the cases considered, we show that such A does not exist. Hence, $n < 7$. ■

Lemmas 4.6.3 and 4.6.4 are important tools used to obtain the exact values of $t(P_n)$ and $t(C_n)$ for some small values of n , as shown in the following theorem.

Theorem 4.6.5. 1. $t(P_5) = t(C_5) = t(P_6) = t(C_6) = 5$.

2. $t(P_7) = t(C_7) = t(P_8) = t(C_8) = t(P_9) = t(C_9) = t(P_{10}) = 6$.

Proof. We provide a proof for each item.

1. By Proposition 4.2.6, Corollary 4.2.8, and $t(2, 5) = 5$ (see Table 3.4), we have that $4 \leq t(P_5) \leq t(C_5) \leq 5$. However, in Lemma 4.6.3 we proved that with a ground set of size 4, a CFF on a path has at most 4 columns. Hence, $t(P_5) = t(C_5) = 5$. By Proposition 4.2.6 and Theorem 4.4.11, we have $t(P_5) \leq t(P_6) \leq t(C_6) \leq 5$. Since $t(P_5) = 5$, we have $t(P_6) = t(C_6) = 5$.

2. By Proposition 4.2.6, Theorem 4.4.11, and the above, we have $5 = t(P_6) \leq t(P_7) \leq t(C_7) \leq 6$. Since by Lemma 4.6.4, we have that with a ground set of size 5, a CFF on a path has at most 6 columns. Thus, $t(P_7) = t(C_7) = 6$. By Proposition 4.2.6 and Theorem 4.4.11, we have $t(P_7) \leq t(P_8) \leq t(C_8) \leq 6$. Since $t(P_7) = 6$, thus, $t(P_8) = t(C_8) = 6$. Again, by Proposition 4.2.6 and Theorem 4.4.11, $6 = t(P_8) \leq t(P_9) \leq t(C_9) \leq 6$. Hence, $t(P_9) = t(C_9) = 6$.

Let $V(P_{10}) = \{v_i : i \in [1, 10]\}$ and $E(P_{10}) = \{\{v_i, v_{i+1}\} : i \in [1, 9]\}$. With the ground set $[1, 6]$, let $\mathcal{F} = \{B_{v_i} : i \in [1, 10]\}$ be the set system where

$$\begin{aligned} B_{v_1} &= \{1, 2, 3\} & B_{v_3} &= \{1, 2, 5\} & B_{v_5} &= \{1, 3, 5\} & B_{v_7} &= \{2, 3, 5\} & B_{v_9} &= \{2, 4, 6\} \\ B_{v_2} &= \{1, 2, 4\} & B_{v_4} &= \{1, 5, 6\} & B_{v_6} &= \{3, 4, 5\} & B_{v_8} &= \{2, 3, 6\} & B_{v_{10}} &= \{4, 5, 6\} \end{aligned}$$

It is easy to check that $\mathcal{F}$ is a P_{10} -CFF. Since $6 = t(P_9) \leq t(P_{10}) \leq 6$, we have $t(P_{10}) = 6$.

■

Open Problem 4.6.6. Just as in the case of the P_{10} -CFF built with $[1, 6]$ by carefully eliminating subsets from its largest Sperner family, it would be interesting to investigate whether this technique can be generalized for large n , to see whether this can improve the bound presented in Theorem 4.4.11.

The fact that $t(P_5) = 5$ and $t(P_{10}) = 6$ is an example of the relationship $t(P_{2n}) = t(P_n) + 1$. Similarly, $t(C_4) = 4$ and $t(C_8) = 6$ is an example of the relationship $t(C_{2n}) = t(C_n) + 2$. It would be interesting to investigate when $t(C_{2n}) = t(C_n) + 1$ and when $t(C_{2n}) = t(C_n) + 2$.

Corollary 4.6.7. Let Q_n be the Hamming graph $H(n, 2)$. Then, $t(Q_2) = 4$ and $t(Q_3) = 6$.

Proof. Since $H(n, 2) = \underbrace{K_2 \square \cdots \square K_2}_{n \text{ times}}$ and it is Hamiltonian, by Propositions 4.2.6 and 4.4.15, we have $t(C_{2n}) \leq t(Q_n) \leq 2n$. For $n = 2$ and 3, since $t(C_4) = 4$ (by Proposition 4.6.1) and $t(C_8) = 6$ (by Proposition 4.6.5), we have the desired results. ■

We now determine $t(W_n)$ for small values of n .

Corollary 4.6.8. 1. $t(W_5) = 5$.

2. $t(W_6) = t(W_7) = 6$.

3. $t(W_8) = t(W_9) = t(W_{10}) = 7$.

Proof. In W_n , we label the universal vertex by 0, and label the other vertices forming a C_{n-1} as $[1, n-1]$.

1. By Corollary 4.4.18, $\max\{t(C_4), t(C_5), t(1, 4) + 1\} \leq t(W_5) \leq t(C_4) + 1$. By Equation 3.2.1, $t(1, 4) = 4$, by Proposition 4.6.1, $t(C_4) = 4$ and by Theorem 4.6.5, $t(C_5) = 5$. Thus, $t(W_5) = 5$.
2. By Corollary 4.4.18, $t(C_6) \leq t(W_6) \leq t(C_5) + 1$. By Theorem 4.6.5, $t(C_5) = 5$ and $t(C_6) = 5$. Thus, we obtain $5 \leq t(W_6) \leq 6$. Since $t(1, 5) = 4$ (by Equation 3.2.1) and $t(C_5) = t(1, 5) + 1$, by Corollary 4.2.14 we have $t(W_6) = 6$. By Corollary 4.4.18, $t(C_7) \leq t(W_7) \leq t(C_6) + 1$. By Theorem 4.6.5, $6 \leq t(W_7) \leq 6$. Thus, $t(W_7) = 6$.
3. Since $t(1, 7) = 5$ and by Theorem 4.6.5, $t(C_7) = 6 = t(C_8)$; by Corollary 4.4.18 we have $6 \leq t(W_8) \leq 7$. Since $t(C_7) = t(1, 7) + 1$, by Corollary 4.2.14 we have $t(W_8) = 7$. Since $t(C_8) = 6$ and $t(C_9) = 6$ (by Theorem 4.6.5) and $t(1, 8) = 5$; by Corollary 4.4.18 we obtain $6 \leq t(W_9) \leq 7$. Since $t(C_8) = t(1, 8) + 1$, by Corollary 4.2.14 we have $t(W_9) = 7$. Since $P_{10} \subseteq W_{10}$, by Proposition 4.2.6 $t(P_{10}) \leq t(W_{10})$. Since by Theorem 4.6.5, $t(P_{10}) = 6$, we have $t(W_{10}) \geq 6$. By Corollary 4.4.18, $t(W_{10}) \leq t(C_9) + 1$ and by Theorem 4.6.5, $t(C_9) = 6$. Thus, we obtain $6 \leq t(W_{10}) \leq 7$. Since $t(1, 9) = 5$ and $t(C_9) = t(1, 9) + 1$, by Corollary 4.2.14 we have $t(W_{10}) = 7$.

■

4.7 Conclusion and further directions

This chapter continues the work done in [38, 40] and is the first to systematically explore CFFs based on a graph structure. We provided several bounds on $t(G)$, along with families of graphs achieving the extreme bounds, as well as tight and non-trivial asymptotic bounds for CFFs on several families of graphs. Along with some open problems mentioned earlier (Open Problems 4.3.10, 4.3.11, and 4.6.6), we ask the following questions that are also worth investigating:

Open Problem 4.7.1. Since $t(K_n) = t(2, n)$, it is also worth noting that some dense graphs $G \subseteq K_n$ may have $t(G) = t(2, n)$. Our question is: what is the maximum number of edges that can be deleted from K_n such that the resultant graph G still has $t(G) = t(2, n)$? A related question is: what is the minimum number of edges that need to be deleted from K_n such that the resultant graph G has $t(G) < t(2, n)$?

Open Problem 4.7.2. Does there exist a family of sparse graphs G (possibly highly structured) such that $t(G) = t(2, |V(G)|)$?

In addition, finding exact values of or bounds on $t(G)$ for several other classes of graphs would be another interesting direction of research.

Chapter 5

Cover-free families on hypergraphs

The purpose of this chapter is to study a generalized version of d -CFFs by extending these definitions to include a hypergraph of rank $d > 1$, where vertices of H correspond to sets in the set system. Following the same reasoning from Chapter 4, we consider hypergraphs with no isolated vertices and no hyperedge of size 1 throughout the rest of this chapter. An H -ECFF(t, n) is a set system $([1, t], \mathcal{B})$ such that for any $e \in E(H)$, the union of the sets corresponding to vertices in a hyperedge does not contain any other set in the set system, whereas, an H -CFF(t, n) is a set system $([1, t], \mathcal{B})$ such that for any $e' \subseteq e \in E(H)$ with $e' \neq \emptyset$, the union of the sets corresponding to vertices in e' does not contain any other set in the system. In this sense, cover-free families on graphs studied in Chapter 4 are a special case where $d = 2$. Furthermore, the incidence matrices of an H -ECFF(t, n) and an H -CFF(t, n) for $H = K_n^d$ coincide with a $t \times n$ d -disjunct matrix and a $t \times n$ $\bar{d}$ -disjunct matrix, respectively. While a $t \times n$ d -disjunct matrix is also $\bar{d}$ -disjunct, it is not always the case that an H -ECFF(t, n) is also H -CFF(t, n); such example is already provided in Figure 4.3 for $H = P_4$. Note that an H -CFF(t, n) and an H -ECFF(t, n) are special cases of (H, r) -CFF(t, n) and (H, r) -ECFF(t, n) with $r = 1$. An H -InCFF(t, n) is a set system $([1, t], \mathcal{B})$ such that for any $e' \subsetneq e \in E(H)$ with $e' \neq \emptyset$, the union of the sets corresponding to vertices in e' does not contain any other set corresponding to a vertex in $e \setminus e'$. Thus, the property of a K_n^d -InCFF(t, n) is equivalent to a d -disjunct matrix with the property of being k -disjunct for $k < d$. Similarly, an H -InCFF(t, n) is also a special case of (H, r) -InCFF(t, n) with $r = 1$. Hence, being an H -CFF(t, n) is equivalent to being H -InCFF and H -ECFF.

In this chapter, we explore and further develop the theory of cover-free families on hypergraphs from multiple directions. Although we have discussed applications of disjunct matrices in combinatorial group testing, such matrices are also separable matrices, as studied by Du and Hwang [19]. They also provide a relationship between d -disjunct, $\bar{d}$ -disjunct, d -separable, and $\bar{d}$ -separable matrices, as depicted in Figure 5.2. This motivates us to study the separable matrices extended to a hypergraph (such as H -separable(t, n) and $\bar{H}$ -separable(t, n) defined in Definition 5.2.4). Such extensions have been studied by Gonen et. al. [27]. We further study the relationships among H -separable(t, n), $\bar{H}$ -separable(t, n), H -CFF(t, n) and H -ECFF(t, n), as provided in Figure 5.5. The most important part of this work is that we provide a sufficient condition for a hypergraph H such that any H -ECFF

must be an H -CFF. Moreover, we also prove that if H is a clutter (see Definition 5.2.7), this sufficient condition is also necessary (Theorem 5.2.11), which generalizes the result in Proposition 4.2.10.

We are also motivated to study the characterization of H -CFFs for a hypergraph H from the lens of *directed hypergraph homomorphism*. We define the *underlying cover-free directed hypergraph* based on H , denoted by $\vec{H}_{\text{CF}}$, and a *cover-free directed hypergraph*, denoted by $\vec{CF}^r(t)$. We further show that there exists an H -CFF(t, n) if and only if there exists a *directed hypergraph homomorphism* from $\vec{H}_{\text{CF}}$ to $\vec{CF}^{r+1}(t)$. While the above-mentioned result is an initial contribution to this topic, we outline several future directions of our interest in Section 5.6.

Motivated by the optimal construction of CFFs on star graphs provided in Chapter 4, we provide an upper bound for CFF on hypergraphs with transversal number k , whose upper bound is met by the star graph (Corollary 4.3.4). We further provide an upper bound and a lower bound on CFFs on flower hypergraphs (defined in Definition 5.4.4), which leads us to obtain an exact bound for an infinite family of flower hypergraphs of rank 3 (as provided in Corollary 5.4.9). We give a construction for CFFs on hypergraphs with rank r and strong chromatic number $\chi(H) = k$ using strong k -vertex coloring. For the case when $r = o\left(\sqrt{\frac{k}{\log_2 k}}\right)$, this improves upon the upper bound on $t(H)$ previously obtained from the strong vertex coloring construction by Idalino and Moura [40] (See Theorem 3.3.5).

We also study cover-free families on some hypergraph products that have been studied in the literature [34]. In particular, we focus on the Cartesian product ($\square$) of hypergraphs and the “strong” product ($\boxtimes$) of hypergraphs, which is also a generalization of the strong product of graphs in a certain way. We prove that for two hypergraphs G and H , $\max\{t(G), t(H)\} \leq t(G \square H) \leq t(G) + t(H)$ (Corollary 5.5.9 resulting from Theorem 5.5.8). This immediately gives us an upper bound for $G \square H$ with $G \in \{P_m, C_m\}$, $H \in \{P_n, C_n\}$, and $m, n \geq 3$, namely, the upper bound $t(G \square H) \leq 1.893 \log_2(mn) + \mathcal{O}(1)$ (Corollary 5.5.11), which is close to the trivial lower bound established in Corollary 4.2.8. The construction corresponding to $t(G \square H) \leq t(G) + t(H)$ also provides the result that for a hypergraph G , $t(G \square K_2) \leq t(G) + 2$. Interestingly, we show that S_n with $n = \left\lfloor \frac{x}{2} \right\rfloor + 1$ for some $x \in \mathbb{N}$ is an infinite family of graphs G meeting the upper bound for $t(G \square K_2)$ (see Corollary 5.5.18). We achieve this by identifying a sparse subgraph in $S_n \square K_2$, which we denote by $S_n \odot S_n$ (defined in Notation 5.5.13) and by proving that the exact value of $t(S_n \odot S_n)$ is equal to $t(S_n) + 2$ (Corollary 5.5.17). Recall that such a family of star graphs also meets the trivial lower bound of $t(G)$ and the upper bound for $t(G)$ in Theorem 4.2.11 item 2. Moreover, it turns out that $S_n \odot S_n$ with $n = \left\lfloor \frac{x}{2} \right\rfloor + 1$ is also another example of an infinite family of graphs meeting the upper bound for $t(G)$ in Theorem 4.2.11 item 2. Next, we study cover-free families on the strong product of hypergraphs by providing reinterpretation of some known constructions and bounds for d -CFFs from the perspective of the strong product of hypergraphs. In particular, we provide a generalization of Theorems 3.2.12 and 3.2.15 in Proposition 5.5.19 and Theorem 5.5.20, respectively. Furthermore, a generalization of the doubling construction of 2-CFFs to cover-free families on strong graph products is provided in Theorem 5.5.22.

The structure of the chapter is as follows. In Section 5.1, we present the definition of cover-free families on hypergraphs. In Section 5.2, we extend the definition of separable matrices to hypergraphs, and present a relationship between cover-free families and separable families on hypergraphs analogous to Figure 5.2. In Section 5.3, we first provide a background on directed hypergraphs and directed hypergraph homomorphism along with some auxiliary results followed by a proof of the existence of d -CFFs corresponding to the existence of a directed hypergraph homomorphism from $\vec{H}_{\text{CF}}$ to $\vec{CF}^{r+1}(t)$. In Section 5.4, we provide constructions and bounds for cover-free families on hypergraphs that are generalized from some constructions provided in Section 3.2 and Chapter 4. Section 5.5 provides a brief overview on the Cartesian product and the strong product of hypergraphs in the literature, and further explores cover-free families on the earlier mentioned hypergraph products. Finally, there is a lot of potential for interesting directions for further research, some of which are discussed in Section 5.6.

5.1 Preliminaries

We present the definition of cover-free families on hypergraphs, analogous to Definitions 4.2.1 and 4.2.2 for graphs. Definition 5.1.1 uses a set system and is a special case of the more general notion of cover-free families on hypergraphs defined in Definitions 3.3.1 and 3.3.2. Definition 5.1.2 is the reinterpretation of Definition 5.1.1 using the incidence matrix of the set system.

Definition 5.1.1. (via set system) Let n, t be positive integers with $n \geq t$, and let $H = ([1, n], E(H))$ be a hypergraph. Let $\mathcal{F} = ([1, t], \mathcal{B})$ be a set system with $\mathcal{B} = \{B_1, B_2, \dots, B_n\}$ and $|\mathcal{B}| = n$ where B_i corresponds to vertex i . $\mathcal{F}$ is *cover-free for a hyperedge* $I \in E(H)$, if for any $i_0 \in [1, n] \setminus I$, we have

$$|B_{i_0} \setminus (\cup_{i \in I} B_i)| \geq 1.$$

$\mathcal{F}$ is *in-cover-free for a hyperedge* $I \in E(H)$ if for any $I' \subset I \in E(H)$, and for any $i_0 \in I \setminus I'$, we have

$$|B_{i_0} \setminus (\cup_{i \in I'} B_i)| \geq 1.$$

Finally, we have:

1. $\mathcal{F}$ is an H -ECFF(t, n) if $\mathcal{F}$ is cover-free for all hyperedges in $E(H)$;
2. $\mathcal{F}$ is an H -InCFF(t, n) if $\mathcal{F}$ is in-cover-free for all hyperedges in $E(H)$;
3. $\mathcal{F}$ is an H -CFF(t, n) if $\mathcal{F}$ is both H -ECFF(t, n) and H -InCFF(t, n).

It is immediate from Definition 5.1.1 that $\mathcal{F}$ is an H -CFF(t, n) if and only if for any $I' \subseteq I \in E(H)$, and for any $i_0 \in [1, n] \setminus I'$, we have

$$|B_{i_0} \setminus (\cup_{i \in I'} B_i)| \geq 1.$$

When H is a graph, being an H -InCFF set system is equivalent to being H -Sperner. We present the definitions of H -CFF(t, n), H -InCFF(t, n) and H -ECFF(t, n) via incidence matrices.

	1	2	3	4	5
1	0	0	0	1	1
2	1	0	0	0	1
3	1	1	1	0	0

H -disjunct

	1	2	3	4	5
1	0	0	1	0	1
2	1	0	0	1	0
3	0	1	0	0	1

H -In-disjunct

	1	2	3	4	5
1	0	0	0	1	1
2	1	0	0	0	1
3	1	1	1	0	0
4	0	0	1	0	1
5	1	0	0	1	0
6	0	1	0	0	1

$\overline{H}$ -disjunct

Figure 5.1: Examples of an H -disjunct, H -In-disjunct, and $\overline{H}$ -disjunct matrix for the hypergraph H provided in Example 5.1.3

Definition 5.1.2. (via matrix) Let $H = ([1, n], E(H))$ be a hypergraph, and A be a $t \times n$ binary matrix where each column corresponds to a vertex of H . Let $A[i, j]$ be the entry of A in row i and the column corresponding to the vertex $j \in [1, n]$, and A^j be the column vector of A corresponding to vertex j . A is *cover-free* for a hyperedge $I \in E(H)$ if for every $j \in [1, n] \setminus I$, there exists a row index $l \in [1, t]$ such that $A[l, i] = 0$ for all $i \in I$ and $A[l, j] = 1$. A is *in-cover-free* for a hyperedge $I \in E(H)$ if for every $I' \subset I \in E(H)$ and for every $j \in I \setminus I'$, there exists a row index $l \in [1, t]$ such that $A[l, i] = 0$ for all $i \in I'$ and $A[l, j] = 1$. Finally, we have:

1. Matrix A is H -disjunct if it is cover-free for every hyperedge $I \in E(H)$;
2. Matrix A is H -In-disjunct if it is in-cover-free for every hyperedge $I \in E(H)$;
3. Matrix A is $\overline{H}$ -disjunct if A is both H -disjunct and H -In-disjunct.

Example 5.1.3. Let $H = ([1, 5], E)$ be a hypergraph where $E = \{\{1, 2, 3\}, \{2, 3, 4\}, \{4, 5\}\}$. We provide an example of an H -disjunct matrix, an H -In-disjunct matrix, and an $\overline{H}$ -disjunct matrix formed by vertical concatenation of the former two matrices in Figure 5.1.

Thus, matrix A is $\overline{H}$ -disjunct if for every $I' \subseteq I \in E(H)$ and for every $j \in [1, n] \setminus I'$, there exists a row index $l \in [1, t]$ such that $A[l, i] = 0$ for all $i \in I'$ and $A[l, j] = 1$. From Definitions 3.3.1, 3.3.2, and 5.1.1, it is clear that an H -ECFF(t, n) is equivalent to $(H, 1)$ -ECFF(t, n) and an H -CFF(t, n) is equivalent to $(H, 1)$ -CFF(t, n). Similar to Propositions 4.1.3 and 4.2.3, the following statements follow immediately from Definitions 5.1.1 and 5.1.2. $\mathcal{F}$ is H -ECFF(t, n) if and only if its incidence matrix is H -disjunct. Similarly, $\mathcal{F}$ is H -InCFF(t, n) if and only if its incidence matrix is H -In-disjunct. Moreover, $\mathcal{F}$ is H -CFF(t, n) if and only if its incidence matrix is $\overline{H}$ -disjunct. We denote by $t_e(H)$, $t_{in}(H)$, and $t(H)$ the minimum t such that there exist a H -ECFF, H -InCFF, and H -CFF, respectively:

$$t_e(H) = t_e(1, H) = \min\{t : \exists \text{ a } H\text{-ECFF}(t, |V(H)|)\}.$$

$$t_{in}(H) = t_{in}(1, H) = \min\{t : \exists \text{ a } H\text{-InCFF}(t, |V(H)|)\}.$$

$$t(H) = t(1, H) = \min\{t : \exists \text{ a } H\text{-CFF}(t, |V(H)|)\}.$$

By Proposition 3.3.3, $t(H) \leq t_e(H) + t_{in}(H)$. Note that if H is a simple graph, $t_{in}(H) = t_s(H)$. From Definitions 3.1.1, 3.3.1, and 5.1.1, it follows that an (H, r) -CFF(t, n) is equivalent to an H' -CFF(t, n) if $H = ([1, n], E(H))$ and $H' = ([1, n], E(H'))$ with $E(H') = \{\cup_{i=1}^r E_i : E_1, \dots, E_r \in E(H)\}$. Also, a d -CFF(t, n) is equivalent to a K_n^d -CFF(t, n) where K_n^d is the complete d -uniform hypergraph on n vertices. Thus, for $d < n$, $t(d, n) = t(K_n^d)$.

Lemma 5.1.4. Let $\phi : V(H_1) \rightarrow V(H_2)$ be an injective homomorphism from hypergraph $H_1 = (V_1, E_1)$ to $H_2 = (V_2, E_2)$. Then $t(H_1) \leq t(H_2)$.

Proof. Let C be a $t \times |V_2|$ $\overline{H_2}$ -disjunct matrix. We build a $t \times |V_1|$ matrix A such that for any $v_1 \in V_1$ and $v_2 \in V_2$ such that $\phi(v_1) = v_2$, we take $A^{v_1} = C^{v_2}$. We show that A is cover-free and in-cover-free for every hyperedge $e \in E(H_1)$. Let $e = \{v_1, \dots, v_r\}$. Then, $\phi(e) = \{\phi(v_1), \dots, \phi(v_r)\} \in E(H_2)$ and $|\phi(e)| = r$. Since C is in-cover-free for $\phi(e)$, the submatrix of C restricted to columns $C^{\phi(v_1)}, \dots, C^{\phi(v_r)}$ contains a permutation submatrix of order r . This implies the submatrix of A restricted to columns $A^{v_1}, \dots, A^{v_r}$ contains a permutation submatrix of order r . Hence, A is in-cover-free for e . Now, it remains to show that A is cover-free for e , that is, for any $v \in V(H_1) \setminus e$, there exists a row z such that $A[z, v_i] = 0$ for all $i \in [1, r]$ and $A[z, v] = 1$. Indeed, such a row exists since C is cover-free for $\phi(e)$. That is, there exists a row z such that $C[z, \phi(v_i)] = 0$ for all $i \in [1, r]$ and $C[z, \phi(v)] = 1$. The desired result thus follows since $C^{\phi(v_i)} = A^{v_i}$ for all $i \in [1, r]$ and $C^{\phi(v)} = A^v$. Thus, we establish that A is $\overline{H_1}$ -disjunct, and so $t(H_1) \leq t(H_2)$. ■

Using Lemma 5.1.4, it follows that if G is a strong subhypergraph of H , then $t(G) \leq t(H)$. Using the same argument in its proof, it can also be shown that $t_e(G) \leq t_e(H)$ and $t_{in}(G) \leq t_{in}(H)$. The next lemma states that this is also true for subhypergraphs that are not strong subhypergraphs.

Lemma 5.1.5. If H_1 is a subhypergraph of H_2 , then $t_e(H_1) \leq t_e(H_2)$, $t_{in}(H_1) \leq t_{in}(H_2)$, and $t(H_1) \leq t(H_2)$.

Proof. Let A and B be an H_2 -In-disjunct matrix and an H_2 -disjunct matrix, respectively. Let A' and B' be the submatrices of A and B , respectively, obtained by restricting to the columns corresponding to the vertices in H_1 . We show that A' and B' are H_1 -In-disjunct and H_1 -disjunct, respectively. Let $e \in E(H_1)$ where $e = e' \cap V_1$ for some $e' \in E(H_2)$. Since A is in-cover-free for e' in H_2 , A' is in-cover-free for e in H_1 . This implies A' is H_1 -In-disjunct and $t_{in}(H_1) \leq t_{in}(H_2)$. Let $i_0 \in V(H_1) \setminus e$. Then, $i_0 \notin e' \setminus e$. Otherwise, $e' \cap V_1 \supseteq e \cup \{i_0\}$, a contradiction. Since B is cover-free for e' , there exists a row j in B such that $B[j, v] = 0$ for all $v \in e'$ and $B[j, i_0] = 1$. This implies that $B'[j, v] = 0$ for all $v \in e$ and $B'[j, i_0] = 1$. Thus, we establish that B' is cover-free for e in H_1 . Hence, B' is H_1 -disjunct and $t_e(H_1) \leq t_e(H_2)$.

Let C be an $\overline{H_2}$ -disjunct matrix and C' be the submatrix of C , obtained by restricting to the columns C^v for all $v \in V(H_1)$. Since C is both H_2 -disjunct and H_2 -In-disjunct, using

the previous arguments in this proof, we conclude C' is both H_1 -disjunct and H_1 -In-disjunct. Hence, C' is $\overline{H_1}$ -disjunct, which implies $t(H_1) \leq t(H_2)$. ■

Proposition 5.1.6. For a hypergraph H , $t([H]_2) \leq t(H)$.

Proof. Let A be an $\overline{H}$ -disjunct matrix. We show that A is $\overline{[H]_2}$ -disjunct. Let $e = \{a, b\} \in E([H]_2)$. Then, there exists some $e' \in E(H)$ such that $e \subseteq e'$. Since A is in-cover-free for e' , the submatrix of A restricted to the columns corresponding to the vertices in e' contains a permutation submatrix of order $|e'| \geq 2$. This further implies that the submatrix corresponding to columns A^a and A^b contains a permutation submatrix of order 2. Thus, A is Sperner for e .

Let $v_0 \in V([H]_2) \setminus e$. If $v_0 \in V(H) \setminus e'$, since A is cover-free for e' , there exists a row r such that $A[r, v] = 0$ for all $v \in e'$ and $A[r, v_0] = 1$. So, we have $A[r, a] = A[r, b] = 0$ and $A[r, v_0] = 1$. If $v_0 \in e' \setminus \{a, b\}$, since A is in-cover-free for e' , there exists a row r such that $A[r, a] = A[r, b] = 0$ and $A[r, v_0] = 1$. Thus, in both cases, A is cover-free for e in $[H]_2$.

Hence, we conclude A is $\overline{[H]_2}$ -disjunct and $t([H]_2) \leq t(H)$. ■

Proposition 5.1.7. For positive integers d, n with $d < n$, $t_{in}(K_n^d) = t(d-1, n)$.

Proof. From Definition 5.1.1, it is easy to see that being K_n^d -InCFF(t, n) is equivalent to being $(d-1)$ -CFF(t, n). Thus, we have the desired result. ■

Lemma 5.1.8. Let $\phi : V(H_1) \rightarrow V(H_2)$ be a homomorphism from hypergraph $H_1 = (V_1, E_1)$ to $H_2 = (V_2, E_2)$. Then $t_{in}(H_1) \leq t_{in}(H_2)$.

Proof. Let A be a $t^* \times |V(H_2)|$ H_2 -In-disjunct matrix with $t^* = t_{in}(H_2)$. Now, from A we construct a $t^* \times |V(H_1)|$ matrix M with every column corresponding to a vertex in H_1 such that for every $v \in V(H_1)$, we set the column M^v identical to $A^{\phi(v)}$. Let $e \in E(H_1)$. Since homomorphisms preserve edges, we have $\phi(e) = \{\phi(v) : v \in e\} \in E(H_2)$. Since A is in-cover-free for the edge $\phi(e)$, then M is also in-cover-free for e . Thus, M is an H_1 -In-disjunct matrix with t^* rows and this implies $t_{in}(H_1) \leq t^* = t_{in}(H_2)$. ■

Proposition 5.1.9. Let $H = (V, E)$ be a d -uniform hypergraph. Then, $t_{in}(H) \leq t(d-1, \chi(H))$.

Proof. By Theorem 2.2.11, $H \rightarrow K_{\chi(H)}^d$. Hence, by Proposition 5.1.7 and Lemma 5.1.8, $t_{in}(H) \leq t_{in}(K_{\chi(H)}^d) = t(d-1, \chi(H))$. ■

Note that if H is a simple graph, $d = 2$ and $t_{in}(G) = t_s(G)$. Thus, by Proposition 4.1.10, $t_{in}(G) = t(1, \chi(G))$, achieving the upper bound in $t_{in}(H) \leq t(d-1, \chi(H))$.

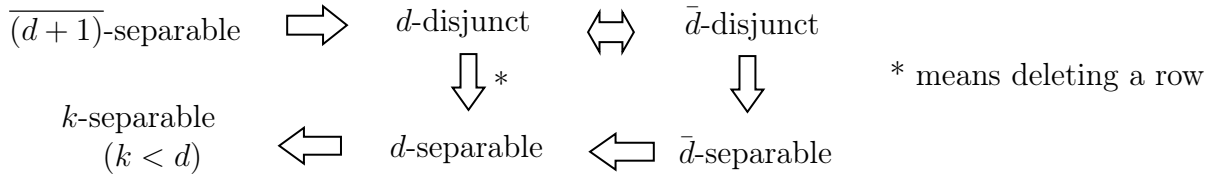


Figure 5.2: Relationships among various types of separable and disjunct matrices in [19]

5.2 Connections between separable families and CFFs on hypergraphs

Along with disjunct matrices, Du and Hwang [19] also study separable matrices and their applications in group testing. We provide the definition of separable matrices in terms of set systems as follows.

Definition 5.2.1. Let n, d and t be positive integers. Let $\mathcal{F} = (\mathcal{X}, \mathcal{B})$ be a set system where $\mathcal{X} = [1, t]$ and $\mathcal{B} = \{B_1, B_2, \dots, B_n\}$.

1. $\mathcal{F}$ is a d -separable(t, n) if for any distinct $J, J' \subseteq [1, n]$ with $|J| = |J'| = d$, we have

$$\cup_{i \in J} B_i \neq \cup_{i \in J'} B_i.$$

2. $\mathcal{F}$ is a $\bar{d}$ -separable(t, n) if for any distinct $J, J' \subseteq [1, n]$ with $|J| = l, |J'| = l'$ and $l, l' \leq d$, we have

$$\cup_{i \in J} B_i \neq \cup_{i \in J'} B_i.$$

Du and Hwang [19] have shown the relations between various types of separable and disjunct matrices as shown in Figure 5.2 where an arrow from a matrix of type T_1 to a matrix of type T_2 indicates that every matrix of type T_1 is also of type T_2 . Chen and Hwang [10] further provided a link from d -separable to k -disjunct or $\bar{k}$ -separable for some $k < d$, which is not depicted in Figure 5.2. They prove the following results.

Theorem 5.2.2. [10]

1. Let M be a d -separable matrix. Then, M is $\bar{k}$ -separable, with $2 \leq k \leq d$, if and only if M is $(k-1)$ -disjunct.
2. Let M be $2d$ -separable. Then, there exists a d -disjunct matrix formed by adding at most one row to M .

Corollary 5.2.3 is a direct consequence of the above theorem.

Corollary 5.2.3. Let M be $2d$ -separable. Then there exists a $\overline{d+1}$ -separable matrix formed by adding at most one row to M .

Similarly to extending a d -CFF(t, n) and a $\bar{d}$ -CFF(t, n) to an H -ECFF(t, n) and H -CFF(t, n), respectively, it is possible to extend a d -separable(t, n) and a $\bar{d}$ -separable(t, n) to an H -separable(t, n) and an $\bar{H}$ -separable(t, n), respectively, as defined below.

Definition 5.2.4. Let $n, t, \mathcal{F}, \mathcal{B}$ be defined as in Definition 5.1.1, and let $H = ([1, n], E(H))$ be a hypergraph. Then,

1. $\mathcal{F}$ is an H -separable(t, n) if for any $I, I' \in E(H)$ with $I \neq I'$,

$$\cup_{i \in I} B_i \neq \cup_{i \in I'} B_i.$$

2. $\mathcal{F}$ is an $\bar{H}$ -separable(t, n) if for any $I, I' \in E(H), I_1 \subseteq I, I_2 \subseteq I'$ and $I_1 \neq I_2$,

$$\cup_{i \in I_1} B_i \neq \cup_{i \in I_2} B_i.$$

From Definitions 5.2.1 and 5.2.4, it follows that a d -separable(t, n) is equivalent to a K_n^d -separable(t, n) and a $\bar{d}$ -separable(t, n) is equivalent to a $(K_n^d)^*$ -separable(t, n) where $E(K_n^d)^* = \{A \subseteq [1, n] : |A| \leq d\}$. Motivated by the extensions of d -disjunct and d -separable set systems in Definitions 5.1.1 and 5.2.4, we aim to generalize the results in Figure 5.2 to establish an analogous relationship between H -separable, $\bar{H}$ -separable, H -CFF ($\bar{H}$ -disjunct) and H -ECFF (H -disjunct) set systems, as shown in Figure 5.5. We prove the result corresponding to each implication in Figure 5.5 in the remainder of this section, where $n, t, H, E(H)$, and $\mathcal{F}$ are set up as in Definition 5.1.1, unless specified.

Proposition 5.2.5. If $\mathcal{F}$ is an $\bar{H}$ -separable(t, n), then it is also an H -separable(t, n).

Proof. The result is immediate from Definition 5.2.4, since the requirements for being $\bar{H}$ -separable is stronger than the requirements for being H -separable. ■

Definition 5.2.6. The *reduction* of a hypergraph H is the hypergraph H' obtained from H by removing any edge that is contained in another edge.

Definition 5.2.7. Let H be a hypergraph and H' be its reduction. H is *reduced* if $H' = H$. A reduced hypergraph is also called a *clutter*.

Hence, a uniform hypergraph is always a clutter.

Proposition 5.2.8. Let H be a hypergraph and H' be the reduction of H . Then,

- Any H' -InCFF must be an H -InCFF.
- Any H' -CFF must be an H -CFF.

Proof. Let $\mathcal{F}$ be an H' -InCFF set system. Since $\mathcal{F}$ is in-cover-free for any $e' \in E(H')$, $\mathcal{F}$ is also in-cover-free for any $e \in E(H)$ with $e \subseteq e'$. So, $\mathcal{F}$ is H -InCFF.

Let $e' \in E(H')$ and $e \in E(H)$ with $e \subsetneq e'$. For $i_0 \in V(H) \setminus e'$, the subset corresponding to i_0 is not contained in the union of the subsets corresponding to vertices in e since $\mathcal{F}$ is

	1	2	3	4	5
0	0	0	0	1	1
1	1	0	0	0	1
1	1	1	1	0	0

 Figure 5.3: An H' -disjunct matrix, but not H -disjunct in Example 5.2.9

cover-free for e' . Furthermore, for $i_0 \in e' \setminus e$, the subset corresponding to i_0 is not contained in the union of the subsets corresponding to vertices in e since $\mathcal{F}$ is in-cover-free for e' . Thus, $\mathcal{F}$ is cover-free for the edge e , which implies $\mathcal{F}$ is H -ECFF, and since $\mathcal{F}$ is H -InCFF, then $\mathcal{F}$ is H -CFF. ■

It is not true that any H' -ECFF must be H -ECFF. We provide a counterexample in Example 5.2.9.

Example 5.2.9. Let $H = ([1, 5], E)$ such that $E = \{\{1, 2\}, \{1, 2, 3\}, \{2, 3, 4\}, \{4, 5\}\}$. Let $H' = ([1, 5], E')$ be the reduction of H , that is, $E' = \{\{1, 2, 3\}, \{2, 3, 4\}, \{4, 5\}\}$. The matrix in Figure 5.3 is H' -disjunct, while it is not H -disjunct because it is not cover-free for $\{1, 2\}$.

Definition 5.2.10. Let $H = (V, E)$ be a hypergraph and let $H' = (V, E')$ be the reduction of H . An *excluded set* of H is a set $S = \{e \setminus \{v\} : e \in E' \text{ and } v \in e\}$. The *excluded degree* of $s \in S$, denoted by $\text{ExDeg}(s)$, is the number of edges in E' containing s . The *minimum excluded degree*, denoted by $\delta_{ex}(H)$, is the minimum $\text{ExDeg}(s)$ among all $s \in S$.

Theorem 5.2.11. Let $H = (V, E)$ be a hypergraph.

- If $\delta_{ex}(H) \geq 2$, then any H -ECFF set system is also H -CFF.
- If H is a clutter, we have:
any H -ECFF set system is also H -CFF if and only if $\delta_{ex}(H) \geq 2$.

Proof. Let $\delta_{ex}(H) \geq 2$ and let $\mathcal{F}$ be an H -ECFF. We need to show that $\mathcal{F}$ is H -InCFF. By Proposition 5.2.8, it is enough to show that $\mathcal{F}$ is H' -InCFF, where $H' = (V, E')$ is the reduction of H . Note that $\delta_{ex}(H') = \delta_{ex}(H) \geq 2$. Next, for any $I \in E'$, we show that $\mathcal{F}$ is in-cover-free for I . Let $i_0 \in I$. Since $\text{ExDeg}(I \setminus \{i_0\}) \geq 2$, there exists $I^* \in E'$ with $I^* \neq I$ such that $I^* = (I \setminus \{i_0\}) \cup X$ for some $X \subseteq V$ with $|X| \geq 1$. Clearly, $i_0 \notin X$, otherwise $I \subsetneq I^*$, contradicting to the fact that H' is reduced. Since $\mathcal{F}$ is H -ECFF, we have $|B_{i_0} \setminus (\cup_{i \in I^*} B_i)| \geq 1$, which implies $|B_{i_0} \setminus (\cup_{i \in I \setminus \{i_0\}} B_i)| \geq 1$. Thus, $\mathcal{F}$ is H' -InCFF, and hence, H -CFF.

Now, we show by contraposition that if H is a clutter with $\delta_{ex}(H) < 2$, there exists an H -ECFF(t, n) that is not H -CFF. Let A be an $|E| \times |V|$ matrix where each row r_e corresponds to an edge e with $A[r_e, v] = 0$ if $v \in e$, and $A[r_e, v] = 1$ if otherwise. Clearly, A is H -disjunct, so it is the incidence matrix of an H -ECFF, $\mathcal{F}$. Let I be an edge with $i_0 \in I$ such that for any edge $I^* \neq I$, we have $I \setminus \{i_0\} \not\subseteq I^*$. Such an edge exists since $\delta_{ex}(H) = 1$. However, by construction, no row r in A exists such that $A[r, v] = 0$ for all $v \in I \setminus \{i_0\}$ and

$A[r, i_0] = 1$. Thus, A is not in-cover-free for I which means A is not $\overline{H}$ -disjunct, so $\mathcal{F}$ is not H -CFF. ■

Observe that for a simple graph $G = (V, E)$, $\delta_{ex}(G) \geq 2$ if and only if $\delta(G) \geq 2$. Since G is a clutter, Theorem 5.2.11 thus generalizes the result for graphs in Proposition 4.2.10.

Proposition 5.2.12. If $\mathcal{F}$ is an H -CFF(t, n), then it is also $\overline{H}$ -separable.

Proof. (By contraposition) Suppose $\mathcal{F}$ is not $\overline{H}$ -separable. Hence, there exist hyperedges $I, I' \in E(H)$ and $I_1 \subseteq I, I_2 \subseteq I'$, with $I_1 \neq I_2$ such that $\cup_{i \in I_1} B_i = \cup_{i \in I_2} B_i$. Since $I_1 \neq I_2$, without loss of generality, there exists at least one vertex i_0 such that $i_0 \in I_1$, but $i_0 \notin I_2$. Then, $B_{i_0} \subseteq \cup_{i \in I_1} B_i$, which implies $B_{i_0} \subseteq \cup_{i \in I_2} B_i$ and it follows that $\mathcal{F}$ is not cover-free for I_2 and so $\mathcal{F}$ is not H -CFF. ■

Proposition 5.2.13. If $\mathcal{F}$ is an H -ECFF(t, n), then it is also an H -separable(t, n).

Proof. (By contraposition) Suppose $\mathcal{F}$ is not an H -separable(t, n). Then there exist $I_1, I_2 \in E(H)$ with $I_1 \neq I_2$ such that $\cup_{i \in I_1} B_i = \cup_{i \in I_2} B_i$. Since $I_1 \neq I_2$, without loss of generality, there exists some $j \in I_1$ such that $j \notin I_2$. Then, $B_j \subseteq \cup_{i \in I_1} B_i$, and by assumption $B_j \subseteq \cup_{i \in I_2} B_i$. Thus, $\mathcal{F}$ is not cover-free for I_2 and therefore $\mathcal{F}$ is not H -ECFF. ■

Notation 5.2.14. Let $\mathcal{F} = ([1, t], \mathcal{B})$ be a set system and $x \in [1, t]$. We denote by $\mathcal{F} \setminus \{x\}$ the set system $(\mathcal{X}', \mathcal{B}')$ where $\mathcal{X}' = [1, t] \setminus \{x\}$ and $\mathcal{B}' = \{B \setminus \{x\} : B \in \mathcal{B}\}$.

The next proposition gives a stronger result, in the case of a clutter.

Proposition 5.2.15. Let H be a clutter. Let $\mathcal{F} = ([1, t], \mathcal{B})$ be an H -ECFF(t, n) and $x \in [1, t]$. Then, $\mathcal{F} \setminus \{x\}$ is an H -separable($t - 1, n$).

Proof. Let $\mathcal{F} = (\mathcal{X}, \mathcal{B})$ where $\mathcal{B} = \{B_1, \dots, B_n\}$ and $\mathcal{F} \setminus \{x\} = (\mathcal{X} \setminus \{x\}, \mathcal{B}')$ where $\mathcal{B}' = \{B'_i = B_i \setminus \{x\} : 1 \leq i \leq n\}$. Let us consider $I_1, I_2 \in E(H)$ such that $I_1 \neq I_2$. We want to show that $\cup_{i \in I_1} B'_i \neq \cup_{i \in I_2} B'_i$. We prove this inequality using two cases, depending on the relationship of x with I_1 and I_2 .

Case 1: $x \in (\cup_{i \in I_1} B_i) \cap (\cup_{i \in I_2} B_i)$ or $x \notin (\cup_{i \in I_1} B_i) \cup (\cup_{i \in I_2} B_i)$. Since $I_1 \neq I_2$, without loss of generality, there exists $y \in I_1 \setminus I_2$. Then $B_y \not\subseteq \cup_{i \in I_2} B_i$ since $\mathcal{F}$ is H -ECFF and $B_y \subseteq \cup_{i \in I_1} B_i$. Since in both cases, we have $B_y \setminus \{x\} \not\subseteq \cup_{i \in I_2} (B_i \setminus \{x\})$ and $B_y \setminus \{x\} \subseteq \cup_{i \in I_1} (B_i \setminus \{x\})$, this implies that $B'_y \not\subseteq \cup_{i \in I_2} B'_i$ and $B'_y \subseteq \cup_{i \in I_1} B'_i$. Hence, there exists some $w \in B'_y$ such that $w \notin \cup_{i \in I_2} B'_i$, but $w \in \cup_{i \in I_1} B'_i$. Thus, we have $\cup_{i \in I_2} B'_i \neq \cup_{i \in I_1} B'_i$.

Case 2: $x \in \cup_{i \in I_1} B_i$ and $x \notin \cup_{i \in I_2} B_i$. Since H is a clutter and $I_1 \neq I_2$, there exists $y \in I_1 \setminus I_2$ and $z \in I_2 \setminus I_1$. Then, $B_z \subseteq \cup_{i \in I_2} B_i$ and since $\mathcal{F}$ is H -ECFF, we have $B_z \not\subseteq \cup_{i \in I_1} B_i$. Then, there exists $w \in B_z$ such that $w \notin \cup_{i \in I_1} B_i$, so $w \neq x$. So, $w \in B_z \subseteq \cup_{i \in I_2} B_i$, $w \notin \cup_{i \in I_1} B_i$, and $w \neq x$. Thus, $w \notin \cup_{i \in I_1} B'_i$, but $w \in \cup_{i \in I_2} B'_i$. Therefore, we have $\cup_{i \in I_2} B'_i \neq \cup_{i \in I_1} B'_i$. ■

The statement of Proposition 5.2.15, however, does not hold for an arbitrary hypergraph H . We provide an example below.

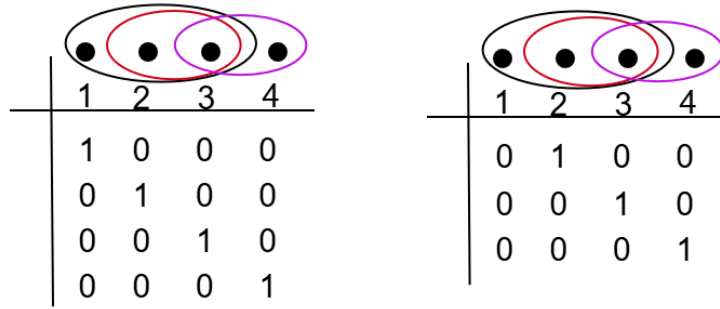


Figure 5.4: A 4×4 H -disjunct matrix (left) and therefore H -separable, but the matrix obtained by deleting its first row (right) is not H -separable.

Example 5.2.16. Let $H = ([1, 4], E)$ such that $E = \{\{1, 2, 3\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}$. H is not a clutter since $\{2, 3\} \subset \{1, 2, 3\}$. Figure 5.4 shows a 4×4 H -disjunct matrix C (left) and the matrix C_* obtained by deleting the first row of C (right). We note that C_* is not H -separable since the Boolean sum of C_*^1, C_*^2 , and C_*^3 is identical to the Boolean sum of C_*^2 and C_*^3 .

Notation 5.2.17. For a hypergraph $H = ([1, n], E)$ with rank $r < n$, let $H^\dagger = ([1, n], E(H^\dagger))$ be a hypergraph with $E(H^\dagger) = \{I \cup \{x\} : I \in E(H), x \in [1, n] \setminus I\}$. We denote the hypergraph with vertex set $[1, n]$ and hyperedge set $E(H) \cup E(H^\dagger)$ as $H \cup H^\dagger$.

Proposition 5.2.18. If $\mathcal{F}$ is an $\overline{H}^\dagger$ -separable(t, n), then it is also an H -CFF(t, n).

Proof. (By contraposition) Suppose $\mathcal{F}$ is not H -CFF. Then, there exists some $I \in E(H)$ with $I' \subseteq I$ and $i_0 \in [1, n] \setminus I'$ such that $B_{i_0} \subseteq \cup_{j \in I'} B_j$. This implies $\cup_{j \in I' \cup \{i_0\}} B_j = \cup_{j \in I'} B_j$.

If $i_0 \in I \setminus I'$, then there exists a vertex $x \in [1, n] \setminus I$ such that $I \cup \{x\} \in E(H^\dagger)$ and we have $I' \subseteq I' \cup \{i_0\} \subseteq I \cup \{x\}$. Such x exists because $|I| < n$. On the other hand, if $i_0 \in [1, n] \setminus I$, then $I \cup \{i_0\} \in E(H^\dagger)$ and clearly, $I' \subseteq I' \cup \{i_0\} \subseteq I \cup \{i_0\}$. Hence, note that regardless of i_0 belonging or not to I , there exists an edge $e \in E(H^\dagger)$ such that $I' \subseteq I' \cup \{i_0\} \subseteq e$. Since $I' \neq I' \cup \{i_0\}$ and we have $\cup_{j \in I' \cup \{i_0\}} B_j = \cup_{j \in I'} B_j$, we conclude that $\mathcal{F}$ is not $\overline{H}^\dagger$ -separable. ■

Proposition 5.2.19. If $\mathcal{F}$ is an $(H \cup H^\dagger)$ -separable(t, n), then it is also an H -ECFF(t, n).

Proof. (By contraposition) Suppose $\mathcal{F}$ is not H -ECFF. Then, there exists $I \in E(H)$ and $i_0 \in [1, n] \setminus I$, such that $B_{i_0} \subseteq \cup_{j \in I} B_j$. This implies $\cup_{j \in I \cup \{i_0\}} B_j = \cup_{j \in I} B_j$ and $\{I, I \cup \{i_0\}\} \subseteq E(H \cup H^\dagger)$. So, we conclude $\mathcal{F}$ is $(H \cup H^\dagger)$ -separable. ■

Thus, the results of this section establish the H -analogues of the relationships in Figure 5.2, connecting H -separable, $\overline{H}$ -separable, H -CFF ($\overline{H}$ -disjunct), and H -ECFF (H -disjunct) set systems, as summarized in Figure 5.5.

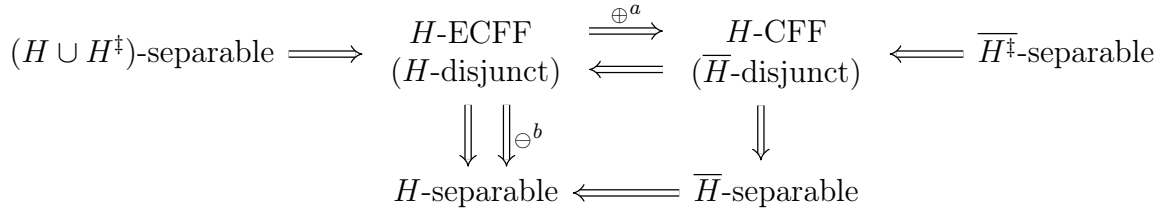


Figure 5.5: Relationships among cover-free families and separable families on hypergraphs

^a H is a hypergraph with $\delta_{ex}(H) \geq 2$

^bIf H is a clutter, this holds even after deleting any point from the base set of an H -ECFF (or any row of its incidence matrix).

5.3 Characterizing H -CFFs through directed hypergraph homomorphisms

In this section, we study the characterization of H -CFFs via the existence of a directed hypergraph homomorphism from the underlying directed hypergraph of H to a special directed hypergraph. These concepts are discussed in Section 5.3.2, following a brief background on directed hypergraphs in Section 5.3.1.

5.3.1 Directed hypergraphs

A *directed hypergraph* is a natural generalization of a *directed graph* or *digraph*. A *directed graph* is a graph where edges have directions, with each directed edge called an *arc*.

Definition 5.3.1. [2] A *directed hypergraph* $\vec{H}$ is a pair $(V(\vec{H}), \mathcal{N}(\vec{H}))$ where $V(\vec{H})$ is a set of vertices and $\mathcal{N}(\vec{H})$ is a set of *hyperarcs* with each *hyperarc* being a pair (I, v) such that I is a non-empty subset of $V(\vec{H})$, called the *tail* of the hyperarc, and $v \in V(\vec{H}) \setminus I$ is the *head* of the hyperarc. For some $r < |V(\vec{H})|$, if $|I| = r$ for all $(I, v) \in \mathcal{N}(\vec{H})$, then $\vec{H}$ is called $(r+1)$ -uniform.

Notation 5.3.2. We denote by $\vec{K}_n^r$ the r -uniform complete directed hypergraph on n vertices where $|V(\vec{K}_n^r)| = n$ and $\mathcal{N}(\vec{K}_n^r) = \left\{ (I, v) : I \in \binom{V(\vec{K}_n^r)}{r-1} \text{ and } v \in V(\vec{K}_n^r) \setminus I \right\}$.

Definition 5.3.3. • A *directed subhypergraph* $\vec{G}$ of $\vec{H}$ has $V(\vec{G}) \subseteq V(\vec{H})$ and $\mathcal{N}(\vec{G}) = \left\{ (I \cap V(\vec{G}), v) : (I, v) \in \mathcal{N}'(\vec{H}), v \in V(\vec{G}), \text{ and } I \cap V(\vec{G}) \neq \emptyset \right\}$ for some $\mathcal{N}'(\vec{H}) \subseteq \mathcal{N}(\vec{H})$.

- A directed subhypergraph $\vec{G}$ of $\vec{H}$ is *induced by* $V(\vec{G})$ if $V(\vec{G}) \subseteq V(\vec{H})$ and $\mathcal{N}(\vec{G}) = \{(I \cap V(\vec{G}), v) : (I, v) \in \mathcal{N}(\vec{H}), v \in V(\vec{G}), \text{ and } I \cap V(\vec{G}) \neq \emptyset\}$.
- A directed subhypergraph $\vec{G}$ of $\vec{H}$ is *strong* if $V(\vec{G}) \subseteq V(\vec{H})$ and $\mathcal{N}(\vec{G}) \subseteq \{(I, v) \in \mathcal{N}(\vec{H}) : I \subseteq V(\vec{G}) \text{ and } v \in V(\vec{G})\}$. A directed subhypergraph $\vec{G}$ of $\vec{H}$ is *strongly induced* if $V(\vec{G}) \subseteq V(\vec{H})$ and $\mathcal{N}(\vec{G}) = \{(I, v) \in \mathcal{N}(\vec{H}) : I \subseteq V(\vec{G}) \text{ and } v \in V(\vec{G})\}$.

Definition 5.3.4. A subset C of the vertex set of a directed hypergraph $\vec{H}$ is a t -clique if $\vec{K}_{|C|}^t$ is a strong directed subhypergraph of $\vec{H}$. The size of the largest t -clique in $\vec{H}$ is called the t -clique number, denoted by $\omega_t(\vec{H})$.

Definition 5.3.5. A *strong c -coloring* of $\vec{H}$ is a function $f : V(\vec{H}) \rightarrow \mathbb{Z}_c$ such that for every hyperarc $(I, v) \in \mathcal{N}(\vec{H})$ with $I = \{v_0, \dots, v_{|I|-1}\} \subseteq V(\vec{H})$ and $v \in V(\vec{H}) \setminus I$, $|\{f(v), f(v_0), \dots, f(v_{|I|-1})\}| = |I| + 1$. The *strong chromatic number* $\chi(\vec{H})$ is the smallest integer c for which there exists a strong c -coloring of $\vec{H}$.

Definition 5.3.6. Let $\vec{G} = (V, \mathcal{N})$ and $\vec{H} = (V', \mathcal{N}')$ be two directed hypergraphs. A *directed hypergraph homomorphism* from $\vec{G}$ to $\vec{H}$, denoted by $\vec{G} \rightarrow \vec{H}$, is a mapping $f : V \rightarrow V'$ such that if $J = (\{v_1, v_2, \dots, v_t\}, v) \in \mathcal{N}$, then $f(J) = (\{f(v_1), f(v_2), \dots, f(v_t)\}, f(v)) \in \mathcal{N}'$, and $|\{v, v_1, v_2, \dots, v_t\}| = |\{f(v), f(v_1), f(v_2), \dots, f(v_t)\}| = t + 1$.

Theorem 5.3.7. Let $\vec{G} = (V, \mathcal{N})$ and $\vec{H} = (V', \mathcal{N}')$ be two directed hypergraphs and $\vec{G} \rightarrow \vec{H}$. Then for any t , we have $\omega_t(\vec{G}) \leq \omega_t(\vec{H})$.

Proof. Let $\omega_t(\vec{G}) = k$ and let $V_1 \subseteq V$ be a t -clique of size k in $\vec{G}$. Let $f : V \rightarrow V'$ be the directed hypergraph homomorphism. For any $J = \{v_1, v_2, \dots, v_{t-1}, v_t\} \subseteq V_1$ of size t , there must exist a hyperarc $N = (I, v) \in \mathcal{N}$ such that $I = \{v_{i_1}, v_{i_2}, \dots, v_{i_{t-1}}\} \subseteq J$ and $v = v_{i_t} \in J \setminus I$. Since f is a directed hypergraph homomorphism, then $(\{f(v_{i_1}), \dots, f(v_{i_{t-1}})\}, f(v_{i_t})) \in \mathcal{N}'$ and $|\{f(v_1), f(v_2), \dots, f(v_{t-1}), f(v_t)\}| = t$. Since this is true for every $J \subseteq V_1$, we must have $f(v) \neq f(v')$ for every $v, v' \in V_1$ with $v \neq v'$. Thus, we conclude that $\{f(v) : v \in V_1\}$ has size k and forms a t -clique of size k in $\vec{H}$. Hence, $\omega_t(\vec{H}) \geq k = \omega_t(\vec{G})$. ■

Theorem 5.3.8. Let $\vec{H} = (V, \mathcal{N}(\vec{H}))$ be an r -uniform directed hypergraph. Then $\vec{H}$ has a strong c -coloring if and only if $\vec{H} \rightarrow \vec{K}_c^r$.

Proof. Assume $\vec{H}$ has a strong c -coloring $f : V \rightarrow \mathbb{Z}_c$ and assume without loss of generality that $V(\vec{K}_c^r) = \mathbb{Z}_c$. Then for any $(I, v) \in \mathcal{N}(\vec{H})$, since $\vec{H}$ is r -uniform, we have $|I| = r - 1$, and we write $I = \{v_1, \dots, v_{r-1}\}$. Since f is a strong c -coloring, we have $|\{f(v_1), \dots, f(v_{r-1}), f(v)\}| = r$. This further implies that $(\{f(v_1), \dots, f(v_{r-1})\}, f(v)) \in \mathcal{N}(\vec{K}_c^r)$. Then, by Definition 5.3.6, we have that $f : V \rightarrow V(\vec{K}_c^r)$ is a homomorphism.

Let $f : \vec{H} \rightarrow \vec{K}_c^r$ be a homomorphism. Then, for any $(\{v_1, \dots, v_{r-1}\}, v) \in \mathcal{N}(\vec{H})$, we have $(\{f(v_1), \dots, f(v_{r-1})\}, f(v)) \in \mathcal{N}(\vec{K}_c^r)$. Since $\vec{H}$ is r -uniform, then $|\{v_1, \dots, v_{r-1}, v\}| = |\{f(v_1), \dots, f(v_{r-1}), f(v)\}| = r$. Hence, f induces a strong c -coloring of $\vec{H}$. ■

Corollary 5.3.9. Let $\vec{G}$ and $\vec{H}$ be r -uniform directed hypergraphs, and $\vec{G} \rightarrow \vec{H}$. Then $\chi(\vec{G}) \leq \chi(\vec{H})$.

Proof. Let $f : V(\vec{G}) \rightarrow V(\vec{H})$ be a homomorphism from $\vec{G}$ to $\vec{H}$. Let us assume $\chi(\vec{H}) = c$. Then, there exists a homomorphism $g : V(\vec{H}) \rightarrow V(\vec{K}_c^r)$ from $\vec{H}$ to $\vec{K}_c^r$. Hence, $f \circ g : V(\vec{G}) \rightarrow V(\vec{K}_c^r)$ induces a strong c -coloring of $\vec{G}$. Hence, $\chi(\vec{G}) \leq c = \chi(\vec{H})$. ■

5.3.2 Characterization of H -CFFs

In this section, we study the characterization of H -CFFs for a hypergraph H . For this purpose, we define two special directed hypergraphs, namely, the *underlying cover-free directed hypergraph* of H , denoted by $\vec{H}_{\text{CF}}$, and the *cover-free directed hypergraph*, denoted by $\vec{CF}^r(t)$.

Definition 5.3.10. For any hypergraph $H = (V, E)$, let $\vec{H}_{\text{CF}} = (V, \mathcal{N})$ be a directed hypergraph such that $\mathcal{N} = \{(I', v_0) : \emptyset \neq I' \subseteq I \in E \text{ and } v_0 \in V \setminus I'\}$. We call $\vec{H}_{\text{CF}}$ as the *underlying cover-free directed hypergraph* of H .

Proposition 5.3.11. Let H be a hypergraph with $|V(H)| = v$ with rank $r < v$. Then $\omega_2(\vec{H}_{\text{CF}}) = \chi(\vec{H}_{\text{CF}}) = v$.

Proof. By Definition 5.3.10 and recalling our general assumption of no isolated vertices, $\vec{K}_v^2$ is a strong directed subhypergraph of $\vec{H}_{\text{CF}}$. The desired result thus follows. ■

Definition 5.3.12. Let $r, t > 0$. For any $v \in \mathbb{Z}_2^t$, we denote by $v[j]$ the entry in its j^{th} position. A *cover-free directed hypergraph* denoted by $\vec{CF}^r(t)$ is a directed hypergraph $(V, \mathcal{N})$ with:

- $V = \mathbb{Z}_2^t$.
- For any $s > 1$ with $s \leq r$ and any s distinct vertices $v_0, \dots, v_{s-1}$, we have $(\{v_1, v_2, \dots, v_{s-1}\}, v_0) \in \mathcal{N}$ if and only if there exists $j \in [0, t-1]$ such that $v_1[j] = v_2[j] = \dots = v_{s-1}[j] = 0$ and $v_0[j] = 1$.

Theorem 5.3.13. Let $n, t, H, E(H)$ be defined as Definition 5.1.1 and r be the rank of H . There exists an H -CFF(t, n) if and only if there exists a directed hypergraph homomorphism $\vec{H}_{\text{CF}} \rightarrow \vec{CF}^{r+1}(t)$.

Proof. Let $\mathcal{F}$ be an H -CFF(t, n), and A be its $t \times n$ incidence matrix. Consider the mapping $f : V(\vec{H}_{\text{CF}}) \rightarrow V(\vec{CF}^{r+1}(t))$ where $f(v_i) = A^i$. For any hyperarc $(I, v_0) \in \mathcal{N}(\vec{H}_{\text{CF}})$ with $I' = \{v_1, \dots, v_{|I'|}\} \subseteq I \in E(H)$, $|I'| \leq |I| \leq r$ and $v_0 \in V(H) \setminus I'$, then for $(f(I'), f(v_0))$ where $f(I') = \{f(v_1), \dots, f(v_{|I'|})\}$, there exists $j \in [0, t-1]$, $f(v_1)[j] = \dots = f(v_{|I'|})[j] = 0$, $f(v_0)[j] = 1$, and $|I'| = |f(I')| \leq r$. Hence, $(f(I'), f(v_0))$ is a hyperarc of $\vec{CF}^{r+1}(t)$. Thus, f is a directed hypergraph homomorphism from $\vec{H}_{\text{CF}}$ to $\vec{CF}^{r+1}(t)$.

Assume that there exists a directed hypergraph homomorphism f from $\vec{H}_{\text{CF}}$ to $\vec{CF}^{r+1}(t)$. Consider the array $A = [f(v_0), f(v_1), \dots, f(v_{n-1})]$. Since $f(v_i) \in V(\vec{CF}^{r+1}(t)) = \mathbb{Z}_2^t$, then for every $(I, v_0) \in \mathcal{N}(\vec{H}_{\text{CF}})$ with $I = \{v_1, \dots, v_{|I|}\} \subseteq I' \in E(H)$, $|I| \leq r$ and $v_0 \in V(H) \setminus I$, there exists a $j \in [0, t-1]$ such that $f(v_1)[j] = \dots = f(v_{|I|})[j] = 0$, $f(v_0)[j] = 1$, and $|I| = |f(I)| \leq r$. Hence, this guarantees that for any $I \subseteq I' \in E(H)$, the subset corresponding to the column $v_0 \in V(H) \setminus I$ is not contained in the union of the subsets corresponding to columns indexed by I . Thus, A is $\vec{H}$ -disjunct, and there exists an H -CFF(t, n). ■

Notation 5.3.14. Let $r \geq 2$. We denote by $\vec{K}_n^{\vec{r}}$ the directed hypergraph such that

$$\mathcal{N}(\vec{K}_n^{\vec{r}}) = \bigcup_{2 \leq r' \leq r} \mathcal{N}(\vec{K}_n^{r'}).$$

Corollary 5.3.15. There exists a d -CFF(t, n) if and only if there exists a directed hypergraph homomorphism $\vec{K}_n^{\vec{d+1}} \rightarrow \vec{CF}^{d+1}(t)$.

Proof. The proof immediately follows from Theorem 5.3.13 as $(\vec{K}_n^d)_{\text{CF}} = \vec{K}_n^{\vec{d+1}}$. ■

Proposition 5.3.16. Let G and H be two hypergraphs such that there exists a directed hypergraph homomorphism $\vec{G}_{\text{CF}} \rightarrow \vec{H}_{\text{CF}}$. Then $t(G) \leq t(H)$.

Proof. Let A be $t^* \times |V(H)|$ $\vec{H}$ -disjunct matrix with $t^* = t(H)$ and r be the rank of H . Then, by Theorem 5.3.13, there exists a directed hypergraph homomorphism $\vec{H}_{\text{CF}} \rightarrow \vec{CF}^{r+1}(t^*)$. Since $\vec{G}_{\text{CF}} \rightarrow \vec{H}_{\text{CF}}$ is a directed hypergraph homomorphism, there exists a directed hypergraph homomorphism from $\vec{G}_{\text{CF}}$ to $\vec{CF}^{r+1}(t^*)$. Thus, there exists a $t^* \times |V(G)|$ $\vec{G}$ -disjunct matrix, and $t(G) \leq t^* = t(H)$. ■

5.4 Constructions of H -CFFs and bounds on $t(H)$

We begin with a simple construction that builds a CFF on a hypergraph from CFFs on smaller ones. Proposition 5.4.2, thus, generalizes the construction presented in Proposition 3.2.9.

Theorem 5.4.1. Let $H_i = (V_i, E_i)$ be a hypergraph with no isolated vertices and $|V_i| = n_i \geq 1$, for $i = 1, 2$ with $|V_1 \cap V_2| = 0$. Let $H = (V_1 \dot{\cup} V_2, E)$ be a hypergraph where

$$E = E_1 \cup E_2 \cup \{I_1 \cup I_2 : \emptyset \neq I_i \subseteq e_i \text{ for } i = 1, 2 \text{ and } (e_1, e_2) \in E_1 \times E_2\}.$$

Then, there exists a $t \times (n_1 + n_2)$ $\overline{H}$ -disjunct matrix where $t = t(H_1) + t(H_2)$. If H_2 is a hypergraph with a single hyperedge, then there exists a $t \times (n_1 + n_2)$ $\overline{H}$ -disjunct matrix with $t = t(H_1) + n_2$.

Proof. Let A and B be $t_1 \times n_1$ $\overline{H}_1$ -disjunct and $t_2 \times n_2$ $\overline{H}_2$ -disjunct matrices, respectively. If $|E_2| = 1$ with $n_2 \geq 1$, then B is the identity matrix of order n_2 . Let $\mathcal{C}$ be a matrix of the form $\begin{pmatrix} A & \mathbf{0}_{t_1 \times n_2} \\ \mathbf{0}_{t_2 \times n_1} & B \end{pmatrix}$ with the first and second column blocks corresponding to vertices in V_1 and V_2 , respectively and rows in the first and second row blocks labeled as $(1, i)$ and $(2, j)$ for $i \in [1, t_1]$ and $j \in [1, t_2]$, respectively. We now prove that $\mathcal{C}$ is $\overline{H}$ -disjunct. We show that $\mathcal{C}$ is cover-free and in-cover-free for every hyperedge $I \in E$. We consider two different cases depending on the choice of I :

Case 1: $I \in E_1$ or $I \in E_2$. Suppose $I \in E_1$. Let $v_0 \in V \setminus E_1$. For the case of $v_0 \in V_2$, there exists a row j in B with $B[j, v_0] = 1$. This implies $\mathcal{C}[(2, j), v] = 0$ for all $v \in I$ and $\mathcal{C}[(2, j), v_0] = 1$. For the case of $v_0 \in V_1 \setminus E_1$, since A is cover-free for I , there exists a row j in A such that $A[j, v] = 0$ for all $v \in I$ and $A[j, v_0] = 1$. This further implies that $\mathcal{C}[(1, j), v] = 0$ for all $v \in I$ and $\mathcal{C}[(1, j), v_0] = 1$. Hence, in both cases, we see that $\mathcal{C}$ is cover-free for I . Furthermore, since A is in-cover-free for I , then $\mathcal{C}$ is in-cover-free for I . The argument for $I \in E_2$ follows similarly.

Case 2: $I = I_1 \cup I_2$ with $\emptyset \neq I_i \subseteq e_i \in E_i$ for $i = 1, 2$. Without loss of generality, let $v_0 \in V_1 \setminus I_1$. If $v_0 \in V_1 \setminus e_1$, since A is cover-free for e_1 in H_1 , there exists a row j in A such that $A[j, v] = 0$ for all $v \in I_1 \subseteq e_1$ and $A[j, v_0] = 1$. Then, we have $\mathcal{C}[(1, j), v] = 0$ for all $v \in I_1 \cup I_2$ and $\mathcal{C}[(1, j), v_0] = 1$. On the other hand, if $v_0 \in e_1 \setminus I_1$, since A is in-cover-free for e_1 in H_1 , there exists a row j in A such that $A[j, v] = 0$ for all $v \in I_1$ and $A[j, v_0] = 1$. Then, we have $\mathcal{C}[(1, j), v] = 0$ for all $v \in I_1 \cup I_2$ and $\mathcal{C}[(1, j), v_0] = 1$. Hence, $\mathcal{C}$ is cover-free for I .

Let $v_0 \in I$. Then, $v_0 \in I_1$ or $v_0 \in I_2$. Without loss of generality, assume $v_0 \in I_1$. Since A is in-cover-free for e_1 in H_1 and $I_1 \setminus \{v_0\} \subseteq I_1 \subseteq e_1$, there exists a row j in A such that $A[j, v] = 0$ for all $v \in I_1 \setminus \{v_0\}$ and $A[j, v_0] = 1$. Then, we have $\mathcal{C}[(1, j), v] = 0$ for all $v \in (I_1 \cup I_2) \setminus \{v_0\}$ and $\mathcal{C}[(1, j), v_0] = 1$. Hence, $\mathcal{C}$ is in-cover-free for I .

Thus, we conclude that $\mathcal{C}$ is $\overline{H}$ -disjunct with $t_1 + t_2$ rows. ■

The following corollary is a consequence of Theorem 5.4.1.

Corollary 5.4.2. Let $H = (V, E)$ be a hypergraph with $|V| = n$. Let V be partitioned into two nonempty subsets V_1 and V_2 and $H_i = H|_{V_i}$ be a hypergraph for $i = 1, 2$. Then, there exists a $t \times n$ $\overline{H}$ -disjunct matrix where $t = t(H_1) + t(H_2)$. If H_2 is a hypergraph with a single hyperedge, then there exists a $t \times (n_1 + n_2)$ $\overline{H}$ -disjunct matrix with $t = t(H_1) + n_2$.

Proof. Let $E_i = E(H_i)$ for $i \in \{1, 2\}$. Then, H is a subhypergraph of $H' = (V_1 \dot{\cup} V_2, E)$ where E is defined as in Theorem 5.4.1. Then, by applying Theorem 5.4.1 Lemma 5.1.5, we conclude there exists a $(t(H_1) + t(H_2)) \times n$ $\overline{H}$ -disjunct matrix. If $|E_2| = 1$, there exists a $(t(H_1) + n_2) \times n$ $\overline{H}$ -disjunct matrix. ■

5.4.1 Using transversal

In this section, we study bounds on $t(H)$ using a subset of $V(H)$ called a transversal, if one exists.

Definition 5.4.3. Let $H = (V, E)$ be a hypergraph without an isolated vertex. A set $T \subset V$ is a *transversal* if every hyperedge contains exactly one point from the set T , that is, $|T \cap V(I)| = 1$, for every $I \in E$.

The minimum cardinality of a transversal is called the *transversal number*, denoted by $\tau(H)$. By convention, if a transversal does not exist, then we define $\tau(H) = \infty$. If $\tau(H) = 1$, the only vertex in a minimum transversal is called a universal vertex. Note that, for a windmill graph $\text{Wd}(k, m)$ with $k \geq 2$, $t(\text{Wd}(k, m)) = \infty$ as a transversal does not exist.

Definition 5.4.4. A *flower hypergraph* $F(r, m)$ is a hypergraph (V, E) such that $|V| = (r - 1)m + 1$, $|E| = m$, and there exists a universal vertex v_0 such that for any hyperedges $e_1, e_2 \in E$, we have $e_1 \cap e_2 = \{v_0\}$.

Each hyperedge in $F(r, m)$ is also called a *petal*. It is clear that $\tau(F(r, m)) = 1$. $F(2, n)$ is the star graph S_{n+1} . Also, note that $[F(r, m)]_2 = \text{Wd}(r, m)$, and $F(r, m)$ is a conformal hypergraph.

Example 5.4.5. We give an example of a flower hypergraph and the corresponding 2-section in Figures 5.6a and 5.6b.

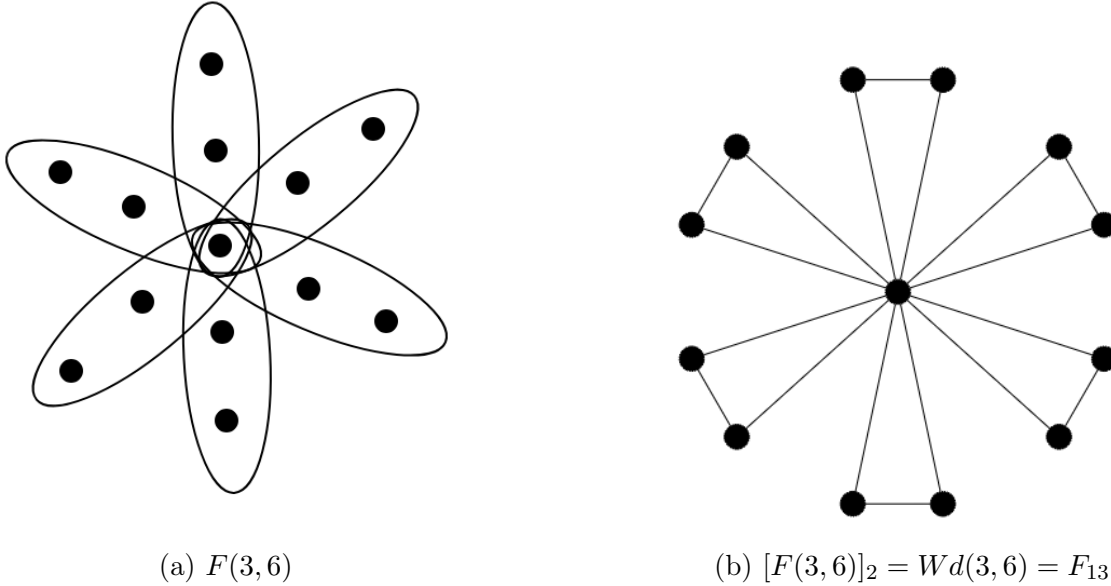


Figure 5.6: Examples of a flower hypergraph and its 2-section

Corollary 5.4.6. Let $H = (V, E)$ be a hypergraph with $|V| = n$, rank r , and let T be a transversal of H with $|T| = k$. Let $V' = V \setminus T$. Then,

$$t(H) \leq t(H|_{V'}) + t(1, k), \quad \text{if } k \geq 2 \quad (5.4.1)$$

$$t(H) \leq t(H|_{V'}) + 1, \quad \text{if } k = 1. \quad (5.4.2)$$

	v_0	$v_{1,1} \ v_{1,2} \ \dots \ v_{1,r-1}$	$v_{2,1} \ v_{2,2} \ \dots \ v_{2,r-1}$	$\dots$	$v_{m,1} \ v_{m,2} \ \dots \ v_{m,r-1}$
$t(1, m)$	$\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$	Column 1 of A	Column 2 of A	$\dots$	Column m of A
$+$					
$r-1$	$\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$	I_{r-1}	I_{r-1}	$\dots$	I_{r-1}
$+$					
1	$\begin{bmatrix} 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & \dots & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & \dots & 0 & 0 \end{bmatrix}$	$\dots$	$\begin{bmatrix} 0 & 0 & \dots & 0 & 0 \end{bmatrix}$

Figure 5.7: An $\overline{F(r, m)}$ -disjunct

Proof. Let T be a transversal of H size k and L_k be a graph with k vertices and the edge set consisting of loops on all k vertices. Then, $H|_T = L_k$ and we know that $t(L_n) = t(1, n)$ for $n \geq 2$ and $t(L_1) = 1$. Let A be a $t_1 \times (n - k)$ $\overline{H|_{V'}}$ -disjunct matrix and B be a $t_2 \times k$ $\overline{1}$ -disjunct with $k \geq 2$. If $k = 1$, then $B = \mathbf{1}_{1 \times 1}$ (that is, $t_2 = 1$). Then, let C be the matrix of the form $\begin{pmatrix} A & \mathbf{0}_{t_1 \times k} \\ \mathbf{0}_{t_2 \times n-k} & B \end{pmatrix}$ with the first and second column blocks corresponding to vertices in V' and T , respectively and rows in the first and second row blocks labeled as $(1, i)$ and $(2, j)$ for $i \in [1, t_1]$ and $j \in [1, t_2]$. By Corollary 5.4.2, C is $\overline{H}$ -disjunct. Thus, $t(H) \leq t(H|_{V'}) + t(L_k)$ and the desired result follows. ■

In case of a star graph S_n , since $\tau(S_n) = 1$ and $H|_{V'} = L_{n-1}$, by Corollary 5.4.6, we have $t(S_n) \leq t(1, n-1) + 1$. However, in Corollary 4.3.4, we showed that $t(S_n) = t(1, n-1) + 1$, achieving the upper bound in Equation 5.4.2.

Construction 5.4.7. Consider a flower hypergraph $F(r, m)$ where the universal vertex is labeled as v_0 . For $1 \leq i \leq m$, we label the non-universal vertices of the i^{th} petal as $v_{i,1}, v_{i,2}, \dots, v_{i,r-1}$. Let A be a $t(1, m) \times m$ $\overline{1}$ -disjunct matrix. Let $\mathcal{C}_1, \mathcal{C}_2$, and $\mathcal{C}_3$ be the matrices where each column is labeled with the vertices of the graph $F(r, m)$. For convenience, we arrange the vertices of the i^{th} petal as consecutive columns in each matrix. Let $\mathcal{C}_1$ be the matrix where $\mathcal{C}_1^{v_0} = \mathbf{0}_{t(1, m) \times 1}$ and $\mathcal{C}_1^{v_{i,j}}$ with $j \in [1, r-1]$ be identical to the i^{th} column of matrix A for each $i \in [1, m]$. Let $\mathcal{C}_2$ be the matrix where $\mathcal{C}_2^{v_0} = \mathbf{0}_{(r-1) \times 1}$ and $\mathcal{C}_2^{v_{i,j}}$ be identical to the j^{th} column of the identity matrix I_{r-1} for all $i \in [1, m]$. Let $\mathcal{C}_3$ be a row matrix with $\mathcal{C}_3[1, v_0] = 1$ and $\mathcal{C}_3[1, v_{i,j}] = 0$ for all $i \in [1, m]$ and $j \in [1, r-1]$. Let $\mathcal{C}$ be the vertical concatenation of $\mathcal{C}_1, \mathcal{C}_2$, and $\mathcal{C}_3$, as shown in Figure 5.7.

Corollary 5.4.8. For $r \geq 3$ and $m \geq 2$,

$$t_e(F(r, m)) = t(1, m), \quad (5.4.3)$$

$$t(\text{Wd}(r, m)) \leq t(F(r, m)) \leq t(1, m) + r. \quad (5.4.4)$$

Proof. We show that $\mathcal{C}_1$ constructed in Construction 5.4.7 is $F(r, m)$ -disjunct, that is, $\mathcal{C}_1$ is cover-free for every petal i . Let $v_{j,l}$ be a vertex with $j \in [1, m] \setminus \{i\}$ and $l \in [1, r-1]$. According to Construction 5.4.7, $\mathcal{C}_1^{v_{i,z}} = A^i$ for all $z \in [1, r-1]$ and $\mathcal{C}_1^{v_{j,l}} = A^j$. Since A is $\bar{1}$ -disjunct and $\mathcal{C}_1^{v_0} = \mathbf{0}_{t(1,m) \times 1}$, there exists a row with 0's in $\mathcal{C}_1^{v_0}, \mathcal{C}_1^{v_{i,z}}$, for all $z \in [1, r-1]$, and 1 in $\mathcal{C}_1^{v_{j,l}}$. Thus, $t_e(F(r, m)) \leq t(1, m)$.

Now, let $V' = V(F(r, m)) \setminus \{v_0\}$. Then, $F(r, m)|_{V'}$ is a matching hypergraph with m disjoint hyperedges and by Proposition 3.3.9, we have $t_e(F(r, m)|_{V'}) = t(1, m)$. In addition, since $F(r, m)|_{V'}$ is a subhypergraph of $F(r, m)$, by Lemma 5.1.5, $t_e(F(r, m)|_{V'}) \leq t_e(F(r, m))$. Thus, we have $t(1, m) \leq t_e(F(r, m))$. Hence, $t(1, m) = t_e(F(r, m))$.

By Corollary 5.4.6, we have $t(F(r, m)) \leq t(F(r, m)|_{V'}) + 1$ since $T = \{v_0\}$ is a transversal and $|T| = 1$. Since $F(r, m)|_{V'}$ is a matching hypergraph with each hyperedge of size $r-1$, by Proposition 3.3.9, we have $t(F(r, m)|_{V'}) \leq t(1, m) + r - 1$. Thus, we have $t(F(r, m)) \leq t(1, m) + r$. Note that $\mathcal{C}$ in Construction 5.4.7 is $\bar{F}(r, m)$ -disjunct with $t(1, m) + r$ rows. Since $\text{Wd}(r, m)$ is the 2-section of $F(r, m)$, by Proposition 5.1.6, $t(\text{Wd}(r, m)) \leq t(F(r, m))$. ■

Note that for $r = 2$, we also have $t_e(F(2, m)) = t(1, m) = t_e(S_{m+1})$ by Theorem 4.3.3.

Corollary 5.4.9. If $m > \lfloor \frac{1}{2}(\overline{\overline{m}} + 1) \rfloor$, then $t(F(3, m)) = t(1, m) + 3$. Otherwise, we have $t(1, m) + 2 \leq t(F(3, m)) \leq t(1, m) + 3$.

Proof. Since $\text{Wd}(3, m) = F_{2m+1}$, by Corollary 5.4.8, we obtain $t(F_{2m+1}) \leq t(F(3, m)) \leq t(1, m) + 3$. If $m > \lfloor \frac{1}{2}(\overline{\overline{m}} + 1) \rfloor$, by Corollary 4.5.5, we have $t(F_{2m+1}) = t(1, m) + 3$ and we conclude that $t(F(3, m)) = t(1, m) + 3$. If $m \leq \lfloor \frac{1}{2}(\overline{\overline{m}} + 1) \rfloor$, then by Corollary 4.5.5, we have $t(1, m) + 2 \leq t(F_{2m+1}) \leq t(1, m) + 3$, which implies that $t(1, m) + 2 \leq t(F(3, m)) \leq t(1, m) + 3$. ■

5.4.2 Using strong vertex coloring

We present a construction of H -CFFs using a technique involving strong vertex coloring. Note that Theorem 3.3.5 works in general and we know that the rank r of any hypergraph is less than or equal to the number of colors k in a strong vertex coloring. Theorem 5.4.10 works in the case $r < k$.

Theorem 5.4.10. Let $H = (V, E)$ be a hypergraph of rank r and $\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_k$ be the sets of vertices in each color class of a strong k -vertex-coloring of H . Assume that $|\mathcal{N}_1| = |\mathcal{N}_2| = \dots = |\mathcal{N}_k| = m > 1$ and $|V| = mk$. If $r < k$, then if there exists an r -CFF(t_1, k) and a 1-CFF(t_2, m), then there exists an H -CFF(t, n) where $t = t_1 \cdot t_2$.

Proof. Suppose the vertices in V are labeled as (x, y) where $1 \leq x \leq k$ and $1 \leq y \leq m$, and vertex (x, y) is in color class x . Let A and B be $t_1 \times k$ $\bar{r}$ -disjunct and $t_2 \times m$ $\bar{1}$ -disjunct matrices, respectively, where the columns of A and B are respectively labeled as $1, 2, \dots, k$ and $1, 2, \dots, m$. Let C be the matrix $A \otimes B$ where the columns correspond to the vertices in the order $(1, 1), \dots, (1, m), (2, 1), \dots, (2, m), \dots, (k, 1), \dots, (k, m)$ and the rows are labeled

as (i, j) where $1 \leq i \leq t_1$ and $1 \leq j \leq t_2$. In this way, the columns of A correspond to the color classes $\mathcal{N}_l$ with $l \in [1, k]$ and the columns $(l, 1), (l, 2), \dots, (l, m)$ correspond to the m vertices in $\mathcal{N}_l$. Now, we show that C is a $t_1 t_2 \times n$ $\bar{H}$ -disjunct matrix.

Let $I = \{(c_1, d_1), (c_2, d_2), \dots, (c_l, d_l)\} \in E$ with vertex $(c_z, d_z) \in \mathcal{N}_{e_z}$ for $z \in [1, l]$, $e_z \in [1, k]$, and $l \leq r$. We need to show that C is cover-free and in-cover-free for the edge I . For $I' \subseteq I$, we consider the following cases depending on the choice of $(c_{l'}, d_{l'}) \in V \setminus I'$:

Case 1: $I' \subset I$ and $(c_{l'}, d_{l'}) \in I \setminus I'$. Since $|I'| < r$ and A is $\bar{r}$ -disjunct, there exists a row $a \in [1, t_1]$ such that $A[a, c_z] = 0$ for all c_z such that $(c_z, d_z) \in I'$ and $A[a, c_{l'}] = 1$. Let $b \in [1, t_2]$ be a row such that $B[b, d_{l'}] = 1$. Thus, $C[(a, b), (c_z, d_z)] = 0$ for all $(c_z, d_z) \in I'$ and $C[(a, b), (c_{l'}, d_{l'})] = 1$. Hence, C is in-cover-free for the hyperedge I .

Case 2: $(c_{l'}, d_{l'}) \notin I$.

Case 2.1: $c_{l'} \neq c_z$ for all c_z such that $(c_z, d_z) \in I'$. Since $|I'| \leq r$ and A is $\bar{r}$ -disjunct, there exists a row a such that $A[a, c_z] = 0$ for all c_z such that $(c_z, d_z) \in I'$ and $A[a, c_{l'}] = 1$. Let b be a row in B such that $B[b, d_{l'}] = 1$. Thus, $C[(a, b), (c_z, d_z)] = 0$ for all $(c_z, d_z) \in I'$ and $C[(a, b), (c_{l'}, d_{l'})] = 1$.

Case 2.2: $c_{l'} = c_{z'}$ for some $c_{z'}$ such that $(c_{z'}, d_{z'}) \in I'$. Let $I'' = I' \setminus \{(c_{z'}, d_{z'})\}$. Since $|I''| < r$ and A is r -disjunct, there exists a row a such that $A[a, c_z] = 0$ for all c_z with $(c_z, d_z) \in I''$ and $A[a, c_{z'}] = 1$. For any $d_{l'} \neq d_{z'}$, let b be a row in B such that $B[b, d_{z'}] = 0$ and $B[b, d_{l'}] = 1$. Thus, $C[(a, b), (c_z, d_z)] = 0$ for all $(c_z, d_z) \in I'$ and $C[(a, b), (c_{z'}, d_{l'})] = 1$.

Since C is cover-free and in-cover-free for every edge, we conclude that C is $\bar{H}$ -disjunct. $\blacksquare$

Note that in the case covered by Corollary 3.3.7, that is, if $r = k$, then the matrix A corresponding to the r -CFF(t_1, k) in Theorem 5.4.10 is replaced by the $k \times k$ identity matrix I_k . However, we have for $r < k$, $t(r, k) \leq cr^2 \log(k)$ for some constant c . If $r = o\left(\sqrt{\frac{k}{\log_2 k}}\right)$, then $\mathcal{O}(r^2 \log_2 k)$ is asymptotically smaller than k . Thus, in this case Theorem 5.4.10 provides an asymptotically smaller bound on $t(H)$ than Corollary 3.3.7. Combining Corollary 3.3.7 and Theorem 5.4.10, we obtain the following result.

Theorem 5.4.11. Let $H = (V, E)$ be a hypergraph of rank r and $\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_k$ be the sets of vertices in each color class of a strong k -vertex-coloring of H . Assume that $|\mathcal{N}_1| = |\mathcal{N}_2| = \dots = |\mathcal{N}_k| = m > 1$ and $|V| = mk$. Then, $t(H) \leq \min\{t(r, k) \cdot t(1, m), k \cdot t(1, m)\}$.

Theorem 5.4.10 can also be extended to the case of color classes of different sizes and the proof is the same as in Theorem 5.4.10. We provide the construction and the corresponding theorem next.

Construction 5.4.12. Let $H = (V, E)$ be a hypergraph of rank r and $\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_k$ be the sets of vertices in each color class of a strong k -vertex-coloring of H . Assume that $|\mathcal{N}_i| = n_i$ and $|V| = n = \sum_{i=1}^k n_i$. Let A be a $t^* \times k$ r -disjunct matrix. For $i \in [1, k]$, let B_i be a $t_i \times n_i$ $\bar{1}$ -disjunct matrix with $n_i > 1$. If $n_i = 1$, then $B_i = \mathbf{1}_{1 \times 1}$. Construct a matrix C consisting of the blocks $C_{x,y}$, $1 \leq x \leq t^*$ and $1 \leq y \leq k$, where block column y corresponds to vertices of H having color y . For an $x \in [1, t^*]$, let $\mathcal{Y}$ be the set of columns with 1's in row x in A and

for each $i \in \mathcal{Y}$, let $t_x = \max\{t_i : i \in \mathcal{Y}\}$ and for B_i 's with $t_i < t_x$, append $t_x - t_i$ copies of $\mathbf{0}_{1 \times n_i}$ to it and call it B'_i .

$$C_{x,y} = \begin{cases} B_y \text{ with } t_x \text{ rows,} & \text{if } y \in \mathcal{Y} \text{ and } t_y = t_x \\ B'_y, & \text{if } y \in \mathcal{Y} \text{ and } t_y < t_x \\ \mathbf{0}_{t_x \times n_y}, & \text{otherwise} \end{cases}$$

The proof of the following theorem is similar to that of Theorem 5.4.10.

Theorem 5.4.13. Let $H = (V, E)$ be a hypergraph of rank r and $\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_k$ be the sets of vertices in each color class of a strong k -vertex-coloring of H . Assume that $|\mathcal{N}_i| = n_i$ and $|V| = n = \sum_{i=1}^k n_i$. If $r < k$, then the matrix C produced in Construction 5.4.12 is a $t \times n$ $\overline{H}$ -disjunct matrix where $t = \sum_{x=1}^{t^*} t_x$.

5.5 Cover-free families on products of hypergraphs

The main motivation behind studying CFFs on the product of hypergraphs is to provide constructions that produce cover-free families (with relatively small number of rows) on larger hypergraphs that can be obtained from the cover-free families on smaller hypergraphs. In this initial work, we consider the Cartesian product and the strong product of graphs and hypergraphs.

5.5.1 Hypergraph products

In this section, we present definitions of several hypergraph products [34]. A hypergraph product is a binary operation on finite hypergraphs, and several different notions of such a product have been defined.

Definition 5.5.1. (Cartesian Product of Hypergraphs) The *Cartesian Product* of two hypergraphs H_1 and H_2 , denoted by $H_1 \square H_2$, is the hypergraph with the set of vertices $V(H_1) \times V(H_2)$ and the set of hyperedges $E(H_1 \square H_2)$ where

$$E(H_1 \square H_2) = \{\{u\} \times e : u \in V(H_1) \text{ and } e \in E(H_2)\} \cup \{e \times \{u\} : e \in E(H_1) \text{ and } u \in V(H_2)\}.$$

In other words, for two hypergraphs H_1 and H_2 , if

$$e = \{(a_1, b_1), (a_2, b_2), \dots, (a_k, b_k)\} \in E(H_1 \square H_2),$$

then the hyperedge e is:

- either $\{(a, b_1), (a, b_2), \dots, (a, b_k)\}$ with $a \in V(H_1)$ and $\{b_1, \dots, b_k\} \in E(H_2)$
- or $\{(a_1, b), (a_2, b), \dots, (a_k, b)\}$ with $\{a_1, \dots, a_k\} \in E(H_1)$ and $b \in E(H_2)$.

The Cartesian product of graphs is a special case of $H_1 \square H_2$ when H_1 and H_2 are graphs. Similarly, the Cartesian product of hypergraphs $H_1, H_2, \dots, H_i$, denoted by

$$H_1 \square H_2 \square \dots \square H_i,$$

is the hypergraph with vertex sets $V(H_1) \times V(H_2) \times \dots \times V(H_i)$ and the set of hyperedges $E(H_1 \square H_2 \square \dots \square H_i)$ where two vertices $(u_1, u_2, \dots, u_i)$ and $(v_1, v_2, \dots, v_i)$ are adjacent if and only if $\{u_j, v_j\} \subseteq e \in E(H_j)$ for exactly one index $1 \leq j \leq i$ and $u_k = v_k$ for $k \neq j$.

Unlike Cartesian product, generalizing the direct product of graphs to hypergraphs is not so straightforward. While it is simpler to extend the definition of the direct graph product to r -uniform hypergraphs, generalizing it for non-uniform hypergraphs can be done in many possible ways. We first present the definition of the direct product of r -uniform hypergraphs.

Definition 5.5.2. (Direct product of r -uniform hypergraphs) For two r -uniform hypergraphs $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$, their *direct product* $H_1 \times_r H_2$ has vertex set $V_1 \times V_2$ and $e = \{(a_1, b_1), (a_2, b_2), \dots, (a_r, b_r)\}$ is an edge in $H_1 \times_r H_2$ if and only if

$$\{a_1, a_2, \dots, a_r\} \in E_1 \text{ and } \{b_1, b_2, \dots, b_r\} \in E_2.$$

If $r = 2$, Definition 5.5.2 is restricted to 2-uniform hypergraphs, and $\times_r$ coincides with the direct graph product of graphs without loops. In order to extend the definition of the direct product to non-uniform hypergraphs, the survey [34] discusses three possible extensions:

1. The minimal-rank preserving direct hypergraph product $(\widetilde{\times})$ (introduced by Sonntag in [67], see [34]),
2. The maximal-rank preserving direct hypergraph product $(\widehat{\times})$,
3. The direct hypergraph product preserving no rank $(\widetilde{\times})$.

Direct products $\widetilde{\times}$ and $\widehat{\times}$ coincide with $\times_r$, when restricted to r -uniform hypergraphs. For 2-uniform hypergraphs, direct products $\times$, $\widehat{\times}$ and $\widetilde{\times}$ coincide with the direct product of simple graphs. However, unlike $\widehat{\times}$, direct products $\times$ and $\widetilde{\times}$ are not applicable for direct graph product in the family of graphs with loops. Thus, throughout the thesis, we refer the maximal-rank preserving direct hypergraph product as the direct hypergraph product.

Definition 5.5.3. (Direct product of hypergraphs) For two hypergraphs $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$, their direct product $H_1 \widehat{\times} H_2$ has vertex set $V_1 \times V_2$ and edge set

$$E(H_1 \widehat{\times} H_2) = \{e \in \binom{V_1 \times V_2}{r} \mid e_i \in E_i \text{ and } p_i(e) = e_i, i = 1, 2\},$$

where $r = \max\{|e_1|, |e_2|\}$ and $p_i(e)$ is the projection of e onto H_i . In other words,

$$e = \{(a_1, b_1), (a_2, b_2), \dots, (a_r, b_r)\}$$

is an edge in $H_1 \widehat{\times} H_2$ if and only if

1. $\{a_1, a_2, \dots, a_r\} \in E_1$ and there exists an edge $e_2 \in E_2$ such that $\{b_1, b_2, \dots, b_r\}$ is a multiset of elements of e_2 , that is, $e_2 \subseteq \{b_1, b_2, \dots, b_r\}$, or
2. There exists an edge $e_1 \in E_1$ such that $\{a_1, a_2, \dots, a_r\}$ is a multiset of elements of e_1 , that is, $e_1 \subseteq \{a_1, a_2, \dots, a_r\}$, and $\{b_1, b_2, \dots, b_r\} \in E_2$.

The strong product of hypergraphs is the union of the cartesian product and the direct product.

Definition 5.5.4. (Strong product of hypergraphs) For any two hypergraphs $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$, their *strong product* $H_1 \widehat{\boxtimes} H_2$ has vertex set $V_1 \times V_2$ and

$$e = \{(a_1, b_1), (a_2, b_2), \dots, (a_r, b_r)\}$$

is an edge in $H_1 \widehat{\boxtimes} H_2$ if and only if

1. $e = \{(a, b_1), (a, b_2), \dots, (a, b_k)\}$ with $a \in V_1$ and $\{b_1, \dots, b_k\} \in E_2$, or
2. $e = \{(a_1, b), (a_2, b), \dots, (a_k, b)\}$ with $\{a_1, \dots, a_k\} \in E_1$ and $b \in V_2$, or
3. $\{a_1, a_2, \dots, a_r\} \in E_1$ and there exists an edge $e_2 \in E_2$ such that $\{b_1, b_2, \dots, b_r\}$ is a multiset of elements of e_2 , that is, $e_2 \subseteq \{b_1, b_2, \dots, b_r\}$, or
4. There exists an edge $e_1 \in E_1$ such that $\{a_1, a_2, \dots, a_r\}$ is a multiset of elements of e_1 , that is, $e_1 \subseteq \{a_1, a_2, \dots, a_r\}$, and $\{b_1, b_2, \dots, b_r\} \in E_2$.

Note that if $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$ are r -uniform hypergraphs, then $\widehat{\boxtimes}$ coincides with $\times_r$. In this case, we write $\boxtimes$ as $\boxtimes_r$.

We provide an example for each of the hypergraph products we discussed above.

Example 5.5.5. Let $H_1 = (\{a_1, a_2\}, \{\{a_1, a_2\}\})$ and $H_2 = (\{b_1, b_2, b_3\}, \{\{b_1, b_2, b_3\}\})$. Then, $V(H_1) \times V(H_2) = \{(a_i, b_j) : i \in [1, 2] \text{ and } j \in [1, 3]\}$.

1. $H_1 \square H_2 = ((V(H_1) \times V(H_2), E(H_1 \square H_2)))$ where

$$E(H_1 \square H_2) = \left\{ \{(a_1, b_1), (a_2, b_1)\}, \{(a_1, b_2), (a_2, b_2)\}, \{(a_1, b_3), (a_2, b_3)\}, \right. \\ \left. \{(a_1, b_1), (a_1, b_2), (a_1, b_3)\}, \{(a_2, b_1), (a_2, b_2), (a_2, b_3)\} \right\}.$$

2. $H_1 \widehat{\times} H_2 = ((V(H_1) \times V(H_2), E(H_1 \widehat{\times} H_2)))$, where

$$E(H_1 \widehat{\times} H_2) = \left\{ \{(a_1, b_1), (a_1, b_2), (a_2, b_3)\}, \{(a_1, b_1), (a_2, b_2), (a_1, b_3)\}, \right. \\ \{(a_2, b_1), (a_1, b_2), (a_1, b_3)\}, \{(a_2, b_1), (a_2, b_2), (a_1, b_3)\}, \\ \left. \{(a_2, b_1), (a_1, b_2), (a_2, b_3)\}, \{(a_1, b_1), (a_2, b_2), (a_2, b_3)\} \right\}.$$

Note that each hyperedge has size 3 and so $H_1 \widehat{\times} H_2$ is uniform. Furthermore, each hyperedge projects onto both factor hyperedges. In other words, for a hyperedge $e = \{(a_i, b_i) : i \in [1, |e|]\}$ of $H_1 \widehat{\times} H_2$, the projection of e with respect to the first coordinate is $p_1(e) = \{a_i : i \in [1, |e|]\} \in E(H_1)$. For example, for $e = \{(a_1, b_1), (a_1, b_2), (a_2, b_3)\}$, we have $p_1(e) = \{a_1, a_2\}$ and $p_2(e) = \{b_1, b_2, b_3\}$.

$$3. H_1 \boxtimes H_2 = \left((V(H_1) \times V(H_2), E(H_1 \boxtimes H_2)) \right) \text{ where } E(H_1 \boxtimes H_2) = E(H_1 \square H_2) \cup E(H_1 \widehat{\times} H_2).$$

While it is known that $K_{n_1}^d \boxtimes_d K_{n_2}^d \cong K_{n_1 n_2}^d$ when $d = 2$, this statement is not true for general d . Moreover, $K_{n_1}^d \boxtimes_d K_{n_2}^d$ is a strong subhypergraph of $K_{n_1 n_2}^d$. To illustrate this, we provide an example below.

Example 5.5.6. Let $H_1 = H_2 = K_3^3$ with $V(H_1) = \{a, b, c\}$, $V(H_2) = \{1, 2, 3\}$, $E(H_1) = \{\{a, b, c\}\}$, and $E(H_2) = \{\{1, 2, 3\}\}$. Let $V(H_1 \boxtimes_3 H_2) = \{(a, 1), (a, 2), (a, 3), (b, 1), (b, 2), (b, 3), (c, 1), (c, 2), (c, 3)\} = K_9^3$. Then $E(H_1 \boxtimes_3 H_2) = E(H_1 \square H_2) \cup E(H_1 \times_3 H_2)$ where

$$E(H_1 \square H_2) = \{\{(a, 1), (b, 1), (c, 1)\}, \{(a, 2), (b, 2), (c, 2)\}, \{(a, 3), (b, 3), (c, 3)\}, \\ \{(a, 1), (a, 2), (a, 3)\}, \{(b, 1), (b, 2), (b, 3)\}, \{(c, 1), (c, 2), (c, 3)\}\},$$

and

$$E(H_1 \times_3 H_2) = \{\{(a, 1), (b, 2), (c, 3)\}, \{(a, 1), (b, 3), (c, 2)\}, \{(a, 2), (b, 1), (c, 3)\}, \\ \{(a, 2), (b, 3), (c, 1)\}, \{(a, 3), (b, 1), (c, 2)\}, \{(a, 3), (b, 2), (c, 1)\}\}.$$

Let $e = \{(a, 1), (b, 1), (c, 2)\}$. Clearly, $e \notin E(H_1 \boxtimes_3 H_2)$ while $e \in E(K_9^3)$.

5.5.2 Cover-free families on the Cartesian product of hypergraphs

In this section, we provide a construction of an $(H_1 \square H_2)$ -CFF using an H_1 -CFF and an H_2 -CFF, and thus obtain an upper bound on $t(H_1 \square H_2)$.

Construction 5.5.7. Let $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$ be two hypergraphs. Let A and B be $t_1 \times |V_1|$ $\overline{H_1}$ -disjunct and $t_2 \times |V_2|$ $\overline{H_2}$ -disjunct matrices, respectively. If $H_2 = K_s^s$, then let B be the $s \times s$ identity matrix, I_s . Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be matrices, each with columns corresponding to vertices (a_i, b_j) for each $a_i \in V_1$ and each $b_j \in V_2$, such that $\mathcal{C}_1^{(a_i, b_j)} = B^{b_j}$ and $\mathcal{C}_2^{(a_i, b_j)} = A^{a_i}$. In this way, $\mathcal{C}_1$ and $\mathcal{C}_2$ are $t_2 \times |V_1 \times V_2|$ and $t_1 \times |V_1 \times V_2|$ matrices, respectively. Let $\mathcal{C}$ be the vertical concatenation of $\mathcal{C}_1$ and $\mathcal{C}_2$, and label the rows of $\mathcal{C}$ by elements in $[1, t_1 + t_2]$.

Theorem 5.5.8. Let $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$ be hypergraphs with $|V_1| = v_1$ and $|V_2| = v_2$. Let A and B be $t_1 \times v_1$ $\overline{H_1}$ -disjunct and $t_2 \times v_2$ $\overline{H_2}$ -disjunct matrices, respectively. Then, there exists a $t^* \times v_1 v_2$ $(\overline{H_1 \square H_2})$ -disjunct matrix where $t^* = t_1 + t_2$. In addition, if $H_2 = K_s^s$, then $t^* = t_1 + s$.

Proof. We show that matrix $\mathcal{C}$ in Construction 5.5.7 is $\overline{(H_1 \square H_2)}$ -disjunct. We need to show that $\mathcal{C}$ is cover-free and in-cover-free for any hyperedge $e = \{(a_1, b_1), (a_2, b_2), \dots, (a_r, b_r)\} \in E(H_1 \square H_2)$. If $\{(a_1, b_1), (a_2, b_2), \dots, (a_r, b_r)\} \in E(H_1 \square H_2)$, then by Definition 5.5.1, we have either $a_1 = \dots = a_r$ and $\{b_1, \dots, b_r\} \in E_2$ or $\{a_1, \dots, a_r\} \in E_1$ and $b_1 = \dots = b_r$. Without loss of generality, we consider $a_1 = \dots = a_r = a$ and $\{b_1, \dots, b_r\} \in E_2$.

For $e' \subseteq \{(a, b_1), (a, b_2), \dots, (a, b_r)\}$, let $e' = \{a\} \times I$ where $I \subseteq \{b_1, b_2, \dots, b_r\}$. Let $(a_{r'}, b_{r'}) \in (V_1 \times V_2) \setminus e'$. We consider the following two cases, depending on the choice of $a_{r'}$:

Case 1: $a_{r'} = a$. Since B is $\overline{H_2}$ -disjunct, for $I \subseteq \{b_1, \dots, b_r\}$ and any $b_{r'} \in V_2 \setminus I$, there exists a row $l \in [1, t_2]$ in B such that $B[l, v] = 0$ for all $v \in I$ and $B[l, b_{r'}] = 1$. Thus, there exists the row $l \in [1, t_2]$ in $\mathcal{C}_1$ such that $\mathcal{C}_1[l, (a, v)] = 0$ for each $v \in I$ and $\mathcal{C}_1[l, (a, b_{r'})] = 1$. A similar argument can also be applied for the case of $H_2 = K_s^s$.

Case 2: $a_{r'} \neq a$. Since A is $\overline{H_1}$ -disjunct, there exists a row l in A such that $A[l, a] = 0$ and $A[l, a_{r'}] = 1$. Thus, $t_2 + l$ is the row in $\mathcal{C}$ such that $\mathcal{C}[t_2 + l, (a, b)] = 0$ for each $b \in I'$ and $\mathcal{C}[t_2 + l, (a_{r'}, b_{r'})] = 1$.

Thus, we establish that $\mathcal{C}$ is $\overline{(H_1 \square H_2)}$ -disjunct with $t_1 + t_2$ rows. ■

Corollary 5.5.9. Let $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$ be two hypergraphs with $|V_1|, |V_2| \geq 3$. Then,

$$\max\{t(H_1), t(H_2)\} \leq t(H_1 \square H_2) \leq t(H_1) + t(H_2). \quad (5.5.1)$$

If $H_2 = K_s^s$ for some $s \geq 2$, then

$$t(H_1) \leq t(H_1 \square H_2) \leq t(H_1) + s. \quad (5.5.2)$$

In addition, for any hypergraph H with $|V(H)| \geq 3$,

$$t(H) \leq t(H \square H) \leq 2t(H). \quad (5.5.3)$$

Proof. The upper bound is achieved by Theorem 5.5.8. Consider an injective homomorphism $\phi_u : H_1 \rightarrow H_1 \square H_2$ such that $\phi_u(v) = (v, u)$ for a fixed $u \in V(H_2)$ and any $v \in V(H_1)$. Then, by Lemma 5.1.4, we have $t(H_1) \leq t(H_1 \square H_2)$. Similarly, $t(H_2) \leq t(H_1 \square H_2)$. Thus, we have the desired lower bound. ■

Corollary 5.5.10. Let $H = H_1 \square H_2 \square \dots \square H_k, k \geq 2$. Suppose for each i , there exists an H_i -CFF(t_i, v_i). Then, there exists an H -CFF($t^*, \prod_{i=1}^k v_i$) where $\max_{i=1}^k \{t_i\} \leq t^* \leq \sum_{i=1}^k t_i$. Hence, $\max_{i=1}^k \{t(H_i)\} \leq t(H) \leq \sum_{i=1}^k t(H_i)$.

Proof. For the upper bound, since the Cartesian product of hypergraphs is an associative operation, that is,

$$H = (\dots((H_1 \square H_2) \square H_3) \square \dots \square H_i) \dots \square H_k),$$

we can obtain an H -CFF($\sum_{i=1}^k t(H_i), \prod_{i=1}^k v_i$) by applying Theorem 5.5.8 $k - 1$ times. The lower bound can be achieved similarly to the proof of Corollary 5.5.9. ■

Since the Hamming graph $H_{a_1, a_2, \dots, a_m} = K_{a_1} \square K_{a_2} \square \dots \square K_{a_m}$, by Corollary 5.5.10 we have $t(H_{a_1, a_2, \dots, a_m}) \leq \sum_{i=1}^m t(K_{a_i}) = \sum_{i=1}^m t(2, a_i)$. Referring to Table 3.4, it is easy to note that for large values of a_i 's, this upper bound is better than the one given in Proposition 4.4.15. Now, using Corollary 5.5.9, we provide an upper bound on $t(G \square H)$ for some families of graphs G and H , which is closer to their trivial lower bounds, as provided in Equation 4.2.1.

Corollary 5.5.11. For $m, n \geq 3$, let $G \in \{P_m, C_m\}$ and $H \in \{P_n, C_n\}$. Then, $\log_2(mn) \leq t(G \square H) \leq 1.893 \log_2(mn) + \mathcal{O}(1)$.

Proof. For any combination of G and H , it is clear that $P_{mn} \subseteq G \square H$ and thus, by Proposition 4.2.6, $t(P_{mn}) \leq t(G \square H)$. Furthermore, by Corollary 5.5.9, $t(G \square H) \leq t(G) + t(H)$ and by Corollary 4.4.12, $t(G) \leq 1.893 \log_2(m) + \mathcal{O}(1)$ and $t(H) \leq 1.893 \log_2(n) + \mathcal{O}(1)$. Thus, $t(G \square H) \leq 1.893 \log_2(m) + 1.893 \log_2(n) + \mathcal{O}(1) = 1.893 \log_2(mn) + \mathcal{O}(1)$. By Corollary 4.2.8 and Theorem 3.2.4, we have $\log_2(mn) \leq t(G \square H)$. ■

Corollary 5.5.12. For $m_i \geq 3$, let $G_i \in \{P_{m_i}, C_{m_i}\}$ and let $G = G_1 \square G_2 \square \dots \square G_k, k \geq 2$. Then, $\log_2 \left(\prod_{i=1}^k m_i \right) \leq t(G) \leq 1.893 \log_2 \left(\prod_{i=1}^k m_i \right) + \mathcal{O}(1)$.

Proof. By Corollary 4.2.8 and Theorem 3.2.4, we have $\log_2 \left(\prod_{i=1}^k m_i \right) \leq t(G)$. Since the Cartesian product of hypergraphs is associative, the desired upper bound can be obtained by applying Corollary 5.5.11 $k - 1$ times. ■

In Equation 5.5.2, if $s = 2$, then $t(H_1) \leq t(H_1 \square K_2) \leq t(H_1) + 2$. In the remaining part of this subsection, we establish that there is an infinite family of star graphs S_n meeting the upper bound in Equation 5.5.2, that is, $t(S_n \square K_2) = t(S_n) + 2$.

Notation 5.5.13. By $S_n \odot S_n$, we denote the graph which is formed by taking two copies of S_n and joining their universal vertices by an edge.

We first provide the following construction to build an $\overline{(S_n \odot S_n)}$ -disjunct matrix.

Construction 5.5.14. Let G and H be two star graphs on n vertices with $V(G) = \{a_i : i \in [0, n - 1]\}$, $V(H) = \{b_i : i \in [0, n - 1]\}$, and a_0 and b_0 being the universal vertices in G and H , respectively. Let A be a $t(1, n - 1) \times (n - 1)$ $\bar{1}$ -disjunct matrix. We create a $t(1, n - 1) \times 2n$ matrix $\mathcal{C}_1$ and three row matrices $\mathcal{C}_2, \mathcal{C}_3$, and $\mathcal{C}_4$ where columns are labeled corresponding to the vertices of G and H in the following order: $a_0, a_1, a_2, \dots, a_{n-1}, b_0, b_1, b_2, \dots, b_{n-1}$. Let $\mathcal{C}_1$ be the matrix where $\mathcal{C}_1^{a_0} = \mathbf{0}_{t(1, n-1) \times 1} = \mathcal{C}_1^{b_0}$ and $\mathcal{C}_1^{a_i} = \mathcal{C}_1^{b_i} = i^{th}$ column of A . Let $\mathcal{C}_2$ and $\mathcal{C}_3$ be the row matrices such that $\mathcal{C}_2[1, a_i] = 1 = \mathcal{C}_3[1, b_i]$ and $\mathcal{C}_2[1, b_i] = 0 = \mathcal{C}_3[1, a_i]$ for each $i \in [0, n - 1]$. Let $\mathcal{C}_4$ be the row matrix such that $\mathcal{C}_4[1, a_0] = \mathcal{C}_4[1, b_0] = 1$ and $\mathcal{C}_4[1, a_i] = \mathcal{C}_4[1, b_i] = 0$ for each $i \in [1, n - 1]$. Let $\mathcal{C}$ be the matrix formed by the vertical concatenation of $\mathcal{C}_1, \mathcal{C}_2$, and $\mathcal{C}_3$ whereas $\mathcal{D}$ be the matrix formed by the vertical concatenation of $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ and $\mathcal{C}_4$, as provided in Figure 5.8.

Next, we show that $\mathcal{C}$ and $\mathcal{D}$ are $(S_n \odot S_n)$ -disjunct and $\overline{(S_n \odot S_n)}$ -disjunct, respectively. We further determine the exact values of $t_e(S_n \odot S_n)$ for all n , and of $t(S_n \odot S_n)$ for infinitely many n .

	a_0	a_1	a_2	$\cdots$	a_{n-1}	b_0	b_1	b_2	$\cdots$	b_{n-1}
$t(1, n-1)$	0	$1 - \text{disjunct}$					0	$1 - \text{disjunct}$		
$\vdots$	$\vdots$						$\vdots$			
0	0						0			
+										
1	1	1	1	$\cdots$	1	0	0	0	$\cdots$	0
+										
1	0	0	0	$\cdots$	0	1	1	1	$\cdots$	1
+										
1	1	0	0	$\cdots$	0	1	0	0	$\cdots$	0

Figure 5.8: An $(S_n \odot S_n)$ -disjunct matrix

Theorem 5.5.15. For $n \geq 3$, $t_e(S_n \odot S_n) = t(1, n-1) + 2$.

Proof. First, we show that $\mathcal{C}$ obtained in Construction 5.5.14 is $(S_n \odot S_n)$ -disjunct, that is, we show that $\mathcal{C}$ is cover-free for every edge. Taking any $e \in E(S_n \odot S_n)$, we split the proof into two cases depending on the type of the edge:

Case 1: $e \in E(S_n)$, for one of the copies of S_n . Without loss of generality, $e = \{a_0, a_i\}$ for $i \in [1, n-1]$. For a vertex a_j with $j \in [1, n-1] \setminus \{i\}$, there exists a row $z \in [1, t(1, n-1)]$ in $\mathcal{C}_1$ such that $\mathcal{C}_1[z, a_0] = \mathcal{C}_1[z, a_i] = 0$ and $\mathcal{C}_1[z, a_j] = 1$. Indeed, such a row exists because A is $\bar{1}$ -disjunct. Similarly, for a vertex b_j with $j \in [0, n-1]$, we have $\mathcal{C}_3[1, a_0] = \mathcal{C}_3[1, a_i] = 0$ and $\mathcal{C}_3[1, b_j] = 1$. Thus, $\mathcal{C}$ is cover-free for the edge $\{a_0, a_i\}$.

Case 2: $e = \{a_0, b_0\}$. Without loss of generality, let $v = a_i$ for some $i \in [1, n-1]$. Let z be a row in $\mathcal{C}_1$ such that $\mathcal{C}_1[z, v] = 1$. This implies that $\mathcal{C}_1[z, a_0] = \mathcal{C}_1[z, b_0] = 0$ and $\mathcal{C}_1[z, v] = 1$. Thus, $\mathcal{C}$ is cover-free for the edge $\{a_0, b_0\}$.

We conclude that $\mathcal{C}$ is $(S_n \odot S_n)$ -disjunct and $t_e(S_n \odot S_n) \leq t(1, n-1) + 2$. Now, we show that $t_e(S_n \odot S_n) \geq t(1, n-1) + 2$. Assume for the sake of contradiction that there exists a $(t(1, n-1) + 1) \times 2n$ $(S_n \odot S_n)$ -disjunct matrix $\mathcal{C}_*$. We consider two possible cases depending on the relationship between $t(1, n)$ and $t(1, n-1)$:

Case 1: $t(1, n) = t(1, n-1) + 1$. With a_0 as the universal vertex, there exists a maximal subgraph S_{n+1} with its edge set $\{\{a_0, a_i\} : i \in [1, n-1]\} \cup \{a_0, b_0\}$. Let I be the set of rows in $\mathcal{C}_*$ such that $\mathcal{C}_*[i, a_0] = 0$ for all $i \in I$ and let w be the weight of $\mathcal{C}_*^{a_0}$. Then, the submatrix $\mathcal{C}'_*$ of $\mathcal{C}_*$ restricted to the columns $\mathcal{C}_*^{a_1}, \mathcal{C}_*^{a_2}, \dots, \mathcal{C}_*^{a_{n-1}}, \mathcal{C}_*^{b_0}$ and rows in I must be a $|I| \times n$ $\bar{1}$ -disjunct matrix. Therefore, $|I| \geq t(1, n) = t(1, n-1) + 1$. However, $w \geq 1$ since there must exist a row that has a 0 in the columns $\mathcal{C}_*^{b_i}, i \in [1, n-1]$, and $\mathcal{C}_*^{b_0}$, and a 1 in the column $\mathcal{C}_*^{a_0}$ since $\{b_0, b_i\}$ is an edge. Thus, $\mathcal{C}_*$ has at least $t(1, n) + 1 = t(1, n-1) + 2$ rows, which is a contradiction to our initial assumption.

Case 2: $t(1, n) = t(1, n-1) = x$. Let G and H be the maximal star graphs S_{n+1} with universal vertices a_0 and b_0 , respectively. Then, $E(G) = \{\{a_0, a_i\} : i \in [1, n-1]\} \cup \{a_0, b_0\}$ and $E(H) = \{\{b_0, b_i\} : i \in [1, n-1]\} \cup \{a_0, b_0\}$. Let I and J be the set of rows in $\mathcal{C}_*$ such that $\mathcal{C}_*[i, a_0] = 0$ for all $i \in I$ and $\mathcal{C}_*[j, b_0] = 0$ for all $j \in J$, respectively. Let w_1

and w_2 be the weight of $\mathcal{C}_*^{a_0}$ and $\mathcal{C}_*^{b_0}$, respectively. Then, $|I| \geq t(1, n)$. Moreover, there must exist a row z_1 such that $\mathcal{C}_*[z_1, a_0] = 1, \mathcal{C}_*[z_1, b_0] = 0$, and $\mathcal{C}_*[z_1, b_i] = 0$ for some $i \in [1, n-1]$ as $\{b_0, b_i\}$ is an edge, obtaining $w_1 \geq 1$. Using a similar argument, we also obtain $|J| \geq t(1, n)$ and $w_2 \geq 1$ as $\{a_0, a_i\}$ is an edge and there must exist a row z_2 such that $\mathcal{C}_*[z_2, b_0] = 1, \mathcal{C}_*[z_2, a_0] = 0$, and $\mathcal{C}_*[z_2, a_i] = 0$ for some $i \in [1, n-1]$. Such rows z_1 and z_2 must be the only rows to satisfy $\mathcal{C}_*[z_1, a_0] = 1, \mathcal{C}_*[z_1, b_0] = 0, \mathcal{C}_*[z_1, b_i] = 0$ for all $i \in [1, n-1]$ and $\mathcal{C}_*[z_2, b_0] = 1, \mathcal{C}_*[z_2, a_0] = 0, \mathcal{C}_*[z_2, a_i] = 0$ for all $i \in [1, n-1]$. Since $\mathcal{C}_*$ has $t(1, n) + 1$ rows, we further observe that $z_1 \in J$ and $z_2 \in I$ with $I \triangle J = \{z_1, z_2\}$ and $|I \cap J| = t(1, n) - 1$ since $\mathcal{C}_*[z_1, a_0] = 1, \mathcal{C}_*[z_1, b_0] = 0, \mathcal{C}_*[z_2, a_0] = 0$, and $\mathcal{C}_*[z_2, b_0] = 1$. Since the submatrix restricted to rows in I and the columns in $\{\mathcal{C}_*^{a_1}, \mathcal{C}_*^{a_2}, \dots, \mathcal{C}_*^{a_{n-1}}, \mathcal{C}_*^{b_0}\}$ forms a $\bar{1}$ -disjunct matrix and $\mathcal{C}_*[z_2, a_i] = 0$ for all $i \in [1, n-1]$, the submatrix corresponding to rows in $I \setminus \{z_2\}$ and the columns in $\{\mathcal{C}_*^{a_1}, \mathcal{C}_*^{a_2}, \dots, \mathcal{C}_*^{a_{n-1}}\}$ forms a $\bar{1}$ -disjunct matrix with $t(1, n) - 1$ rows. This implies that $t(1, n-1) \leq t(1, n) - 1 = x - 1$, which is a contradiction.

Thus, we have $t_e(S_n) \geq t(1, n-1) + 2$, which completes the proof. $\blacksquare$

Corollary 5.5.16. For $n \geq 3$, $t(1, n-1) + 2 \leq t(S_n \odot S_n) \leq t(1, n-1) + 3$.

Proof. Since $t_e(S_n \odot S_n) \leq t(S_n \odot S_n)$, by Theorem 5.5.15, we obtain the desired lower bound. Now, we show that $\mathcal{D}$ obtained in Construction 5.5.14 is $(S_n \odot S_n)$ -disjunct. By Theorem 5.5.15, $\mathcal{D}$ is $(S_n \odot S_n)$ -disjunct. So, we only need to show that $\mathcal{D}$ is $(S_n \odot S_n)$ -Sperner. There are row matrices $\mathcal{C}_2$ and $\mathcal{C}_3$ such that $\mathcal{C}_3[1, a_0] = 0, \mathcal{C}_3[1, b_0] = 1, \mathcal{C}_2[1, a_0] = 1$, and $\mathcal{C}_3[1, b_0] = 0$, concluding $\mathcal{D}$ being Sperner for the edge $\{a_0, b_0\}$. For an edge $\{a_0, a_i\}, i \in [1, n-1]$, there exists a row z in $\mathcal{C}_1$ such that $\mathcal{C}_1[z, a_0] = 0, \mathcal{C}_1[z, a_i] = 1$ and we also have $\mathcal{C}_4[1, a_0] = 1, \mathcal{C}_4[1, a_i] = 0$. This means that $\mathcal{D}$ is Sperner for the edge $\{a_0, a_i\}$ and using a similar argument, it can be shown that $\mathcal{D}$ is Sperner for the edge $\{b_0, b_i\}$ for each $i \in [1, n-1]$. Thus, we obtain the upper bound $t(S_n \odot S_n) \leq t(1, n-1) + 3$. $\blacksquare$

Corollary 5.5.17. For $n \geq 3$, let $n = \lfloor \frac{x}{2} \rfloor + 1$ for some $x \in \mathbb{N}$. Then, $t(S_n \odot S_n) = t(1, n-1) + 3$.

Proof. By Corollary 5.5.16, we have $t(S_n \odot S_n) \leq t(1, n-1) + 3$. Now, we show that $t(S_n \odot S_n) \geq t(1, n-1) + 3$. Since $n = \lfloor \frac{x}{2} \rfloor + 1$, then $t(1, n) = t(1, n-1) + 1 = x + 1$. Assume for the sake of contradiction that there exists $\mathcal{D}$, a $(t(1, n-1) + 2) \times 2n$ $(S_n \odot S_n)$ -disjunct matrix. Figure 5.9 is provided to illustrate the main idea of the proof, where we repeat the column $\mathcal{D}^{a_0}$ for convenience.

Let G and H be the maximal star graphs S_{n+1} with universal vertices a_0 and b_0 , respectively. Then, $E(G) = \{\{a_0, a_i\} : i \in [1, n-1]\} \cup \{a_0, b_0\}$ and $E(H) = \{\{b_0, b_i\} : i \in [1, n-1]\} \cup \{a_0, b_0\}$. Let I and J be the set of rows in $\mathcal{D}$ such that $\mathcal{D}[i, a_0] = 0$ for all $i \in I$ and $\mathcal{D}[j, b_0] = 0$ for all $j \in J$, respectively. Let w_1 and w_2 be the weight of $\mathcal{D}^{a_0}$ and $\mathcal{D}^{b_0}$, respectively. Then, $|I| \geq t(1, n)$. Moreover, there must exist a row z_1 such that $\mathcal{D}[z_1, a_0] = 1, \mathcal{D}[z_1, b_0] = 0$, and $\mathcal{D}[z_1, b_i] = 0$ for some $i \in [1, n-1]$ as $\{b_0, b_i\}$ is an edge, obtaining $w_1 \geq 1$. Using a similar argument, we also obtain $|J| \geq t(1, n)$ and $w_2 \geq 1$ as $\{a_0, a_i\}$ is an edge and there must exist a row z_2 such that $\mathcal{D}[z_2, b_0] = 1, \mathcal{D}[z_2, a_0] = 0$, and $\mathcal{D}[z_2, a_i] = 0$ for some $i \in [1, n-1]$. Such rows z_1 and z_2

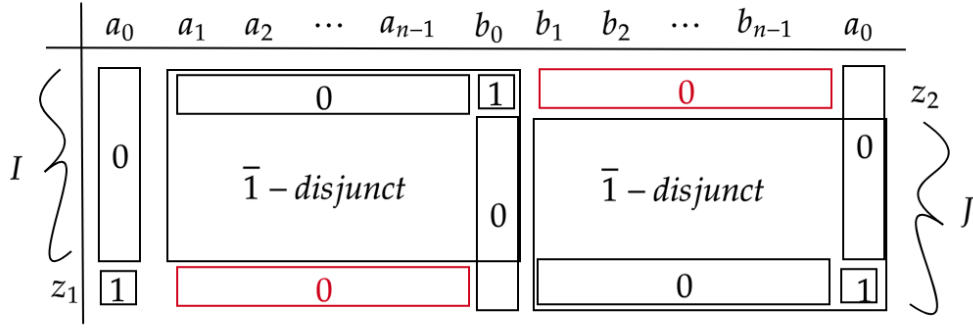


Figure 5.9: An attempt to construct a $(t(1, n - 1) + 2) \times 2n$ $(\overline{S_n \odot S_n})$ -disjunct matrix in Corollary 5.5.17.

must be the only rows to satisfy $\mathcal{D}[z_1, a_0] = 1, \mathcal{D}[z_1, b_0] = 0, \mathcal{D}[z_1, b_i] = 0$ for all $i \in [1, n - 1]$ and $\mathcal{D}[z_2, b_0] = 1, \mathcal{D}[z_2, a_0] = 0, \mathcal{D}[z_2, a_i] = 0$ for all $i \in [1, n - 1]$, respectively. Since $\mathcal{D}$ has $t(1, n) + 1 = t(1, n - 1) + 2$ rows, we further observe that $z_1 \in J$ and $z_2 \in I$ with $I \triangle J = \{z_1, z_2\}$ and $|I \cap J| = t(1, n) - 1 = t(1, n - 1)$ since $\mathcal{D}[z_1, a_0] = 1, \mathcal{D}[z_1, b_0] = 0, \mathcal{D}[z_2, a_0] = 0$, and $\mathcal{D}[z_2, b_0] = 1$ (see black entries in Figure 5.9). Furthermore, $\mathcal{D}$ must be Sperner for edges $\{a_0, a_i\}$ and $\{b_0, b_i\}$ for all $i \in [1, n - 1]$. Therefore, $\mathcal{D}[z_1, a_i] = 0$ and $\mathcal{D}[z_2, b_i] = 0$ for all $i \in [1, n - 1]$ (see red entries in Figure 5.9).

Since the submatrix restricted to rows in I and the columns in $\{\mathcal{D}^{a_1}, \mathcal{D}^{a_2}, \dots, \mathcal{D}^{a_{n-1}}, \mathcal{D}^{b_0}\}$ is a $\bar{1}$ -disjunct matrix and $\mathcal{D}[z_2, a_i] = 0$ for all $i \in [1, n - 1]$, the submatrix corresponding to rows in $I \setminus \{z_2\}$ and the columns in $\{\mathcal{D}^{a_1}, \mathcal{D}^{a_2}, \dots, \mathcal{D}^{a_{n-1}}\}$ is a $\bar{1}$ -disjunct matrix with $t(1, n) - 1 = x$ rows and $\binom{x}{\lfloor \frac{x}{2} \rfloor}$ columns. Then, by Theorem 2.1.4, the weight of $\mathcal{D}^{a_i}$ for all $i \in [1, n - 1]$ is either $\lfloor \frac{x}{2} \rfloor$ or $\lceil \frac{x}{2} \rceil$. Using a similar argument, we also have that the submatrix corresponding to rows in $J \setminus \{z_1\}$ and the columns in $\{\mathcal{D}^{b_1}, \mathcal{D}^{b_2}, \dots, \mathcal{D}^{b_{n-1}}\}$ forms a $\bar{1}$ -disjunct matrix with x rows and the weight of $\mathcal{D}^{b_i}$ for all $i \in [1, n - 1]$ is either $\lfloor \frac{x}{2} \rfloor$ or $\lceil \frac{x}{2} \rceil$. Furthermore, note that $I \setminus \{z_2\} = J \setminus \{z_1\}$. This leads us to two possibilities:

- x is odd and for all $i, j \in [1, n - 1]$, the weights of $\mathcal{D}^{a_i}$ and $\mathcal{D}^{b_j}$ are $\lfloor \frac{x}{2} \rfloor$ and $\lceil \frac{x}{2} \rceil$, respectively. Then, there exist two columns a_i, b_j for some $i, j \in [1, n - 1]$ such that $\mathcal{D}^{a_i} = [0, \underbrace{1, 1, \dots, 1}_{\lfloor \frac{x}{2} \rfloor \text{ times}}, 0, \dots, 0]$ and $\mathcal{D}^{b_j} = [0, \underbrace{1, 1, \dots, 1}_{\lceil \frac{x}{2} \rceil \text{ times}}, 1, 0, \dots, 0]$. This implies that $\mathcal{D}$ is not cover-free for $\{b_0, b_j\}$, as $\mathcal{D}^{b_0} \cup \mathcal{D}^{b_j} \supseteq \mathcal{D}^{a_i}$.
- Without loss of generality, for all $i, j \in [1, n - 1]$, the weight of each of $\mathcal{D}^{a_i}$ and $\mathcal{D}^{b_j}$ is $\lfloor \frac{x}{2} \rfloor$. Then, there exist two columns a_i, b_j for some $i, j \in [1, n - 1]$ such that $\mathcal{D}^{a_i} = \mathcal{D}^{b_j}$. This clearly implies that $\mathcal{D}$ is not cover-free for $\{a_0, a_j\}$ and $\{b_0, b_j\}$.

This leads to our desired contradiction. Thus, $\mathcal{D}$ must have at least $t(1, n - 1) + 3$ rows. ■

Now, we have all the ingredients to prove that S_n with $n = \binom{x}{\lfloor \frac{x}{2} \rfloor} + 1$ is an infinite family of graphs satisfying the upper bound in Equation 5.5.2.

Corollary 5.5.18. For $n \geq 3$, let $n = \binom{x}{\lfloor \frac{x}{2} \rfloor} + 1$ for some $x \in \mathbb{N}$. Then, $t(S_n \square K_2) = t(S_n) + 2$.

Proof. By Corollaries 5.5.9 and 4.3.4, we have $t(S_n \square K_2) \leq t(S_n) + 2 = t(1, n-1) + 3$. Furthermore, $S_n \odot S_n \subseteq S_n \square K_2$. Then, by Proposition 4.2.6 and Corollary 5.5.17, we have $t(1, n-1) + 3 = t(S_n \odot S_n) \leq t(S_n \square K_2)$. Thus, the desired result is obtained. ■

5.5.3 Cover-free families on the strong product of hypergraphs

In this section, we study bounds for cover-free families on the strong product of (hyper)graphs. All the results presented in this section generalize some well-known bounds on $t(d, n)$. In particular, Proposition 5.5.19, Theorem 5.5.20, and Theorem 5.5.22 generalize the bounds presented in Theorem 3.2.12, Theorem 3.2.15, and Theorem 3.2.16, respectively.

Despite having interesting properties, the direct product preserving maximal-rank, $\widehat{\times}$ is not very well-studied in the literature. However, Proposition 5.5.19 indeed provides a natural starting point for the study of cover-free families on the strong product of hypergraphs.

Proposition 5.5.19. Let $H_1 = (V_1 = [1, n_1], E_1)$ and $H_2 = (V_2 = [1, n_2], E_2)$ be two hypergraphs with ranks r_1 and r_2 , respectively. Let A_1 and A_2 be a $t_1 \times n_1$ $\overline{H_1}$ -disjunct matrix and a $t_2 \times n_2$ $\overline{H_2}$ -disjunct matrix, respectively. Then, $A = A_1 \otimes A_2$ is a $t_1 t_2 \times n_1 n_2$ $(H_1 \widehat{\boxtimes} H_2)$ -disjunct matrix. Thus, $t(H_1 \widehat{\boxtimes} H_2) \leq t(H_1) \cdot t(H_2)$.

Proof. Let the columns of A correspond to the vertices in the following order:

$$(1, 1), \dots, (1, n_2), (2, n_1), \dots, (2, n_2), \dots, (n_1, 1), \dots, (n_1, n_2)$$

and the rows of A be labeled as $(1, 1), \dots, (1, t_2), (2, 1), \dots, (2, t_2), \dots, (t_1, 1), \dots, (t_1, t_2)$. In this way, columns corresponding to vertices $(x, 1), (x, 2), \dots, (x, n_2)$ for $x \in [1, n_1]$ correspond to column block x and rows corresponding to $(y, 1), (y, 2), \dots, (y, t_2)$ for $y \in [1, t_1]$ correspond to a row block y . We wish to show that A is cover-free and in-cover-free for every hyperedge $e = \{(a_1, b_1), (a_2, b_2), \dots, (a_r, b_r)\} \in E(H_1 \widehat{\boxtimes} H_2)$. Since $E(H_1 \widehat{\boxtimes} H_2) = E(H_1 \square H_2) \cup E(H_1 \widehat{\times} H_2)$, we have four different possibilities, depending on the type of hyperedge. However, since there exist permutation matrices P and Q such that $A_2 \otimes A_1 = P(A_1 \otimes A_2)Q$ [35], it is enough to consider the following two cases:

Case 1: $e = \{(a, b_1), (a, b_2), \dots, (a, b_r)\}$ where $a \in V_1$ and $\{b_1, b_2, \dots, b_r\} \in E_2$ with $r \leq r_2$. In this case, the columns corresponding to $\{(a, b_1), (a, b_2), \dots, (a, b_r)\}$ are in one column block. For $e' \subseteq \{(a, b_1), (a, b_2), \dots, (a, b_r)\}$, denote $e' = \{a\} \times I$ for some $I \subseteq \{b_1, b_2, \dots, b_r\}$. We further consider two subcases, depending on the choice of $(a_{r'}, b_{r'}) \in V(H_1 \widehat{\boxtimes} H_2) \setminus e'$:

Case 1.1: $a_{r'} = a$. Since A_2 is $\overline{H_2}$ -disjunct, there exists a row $l \in [1, t_2]$ such that $A_2[l, v] = 0$ for $v \in I$ and $A_2[l, b_{r'}] = 1$, since $b_{r'} \in V_2 \setminus I$. Thus, in A , there exists a row block y such that $A[(y, l), (a, v)] = 0$ for all $v \in I$ and $A[(y, l), (a, b_{r'})] = 1$.

Case 1.2: $a_{r'} \neq a$. Then, the column corresponding to $(a_{r'}, b_{r'})$ is not in the column block a . Since A_1 is H_1 -In-Disjunct, there exists a row $x \in [1, t_1]$ in A_1 such that $A_1[x, a] = 0$ and

$A_1[x, a_{r'}] = 1$. Let $y \in [1, t_2]$ be one of the rows in A_2 such that $A_2[y, b_{r'}] = 1$. Thus in A , (x, y) is a row such that $A[(x, y), (a, v)] = 0$ for all $v \in I$ since the column block x in A is a zero matrix and $A[(x, y), (a_{r'}, b_{r'})] = 1$.

Case 2: $e = \{(a_1, b_1), (a_2, b_2), \dots, (a_r, b_r)\}$ where $\{a_1, a_2, \dots, a_r\} \in E_1$ and there exists an edge $f \in E_2$ such that $\{b_1, \dots, b_r\}$ is a multiset of elements of f . Taking any $e' \subseteq e$ such that the projection of e' with respect to the first coordinate, $p_1(e') = I \subseteq \{a_1, a_2, \dots, a_r\}$, and $(a_{r'}, b_{r'}) \in V(H_1 \widehat{\boxtimes} H_2) \setminus e'$, we consider the following two subcases.

Case 2.1: $a_{r'} \notin \{a_1, a_2, \dots, a_r\}$. Since A_1 is $\overline{H_1}$ -disjunct, there exists a row x such that $A_1[x, a_i] = 0$ for $a_i \in I$ and $A_1[x, a_{r'}] = 1$. Let y be a row in A_2 such that $A_2[y, b_{r'}] = 1$. Then, $A[(x, y), (a_i, b_i)] = 0$ for $a_i \in I$ and $A[(x, y), (a_{r'}, b_{r'})] = 1$.

Case 2.2: $a_{r'} = a_i$ for some $i \in [1, r]$. Suppose, without loss of generality, $a_{r'} = a_r$. Let $I' \subseteq \{a_1, a_2, \dots, a_r\} \setminus \{a_r\}$ such that for each $a_i \in I'$, we have $(a_i, b_i) \in e'$. We again split this into two subcases:

Case 2.2.1: $(a_r, b_r) \notin e'$. Since A_1 is H_1 -In-disjunct, there exists a row x in A_1 such that $A_1[x, a_i] = 0$ for $a_i \in I'$ and $A_1[x, a_r] = 1$. For a $b_{r'}$, let y be a row in A_2 such that $A_2[y, b_{r'}] = 1$. Thus, $A[(x, y), (a_i, b_i)] = 0$ for $a_i \in I'$ such that $(a_i, b_i) \in e'$ and $A[(x, y), (a_r, b_{r'})] = 1$.

Case 2.2.2: $(a_r, b_r) \in e'$. Then, $b_{r'} \neq b_r$. Since A_1 is H_1 -In-disjunct, there exists a row x in A_1 such that $A_1[x, a_i] = 0$ for $a_i \in I'$ and $A_1[x, a_r] = 1$. Since A_2 is H_2 -In-disjunct, there exists a row y in A_2 such that $A_2[y, b_r] = 0$ and $A_2[y, b_{r'}] = 1$. Thus, $A[(x, y), (a_i, b_i)] = 0$ for all $(a_i, b_i) \in e'$, and $A[(x, y), (a_r, b_{r'})] = 1$.

So, we establish that A is $(H_1 \widehat{\boxtimes} H_2)$ -disjunct, which implies that $t(H_1 \widehat{\boxtimes} H_2) \leq t_1 t_2$. This also means $t(H_1 \widehat{\boxtimes} H_2) \leq t(H_1) \cdot t(H_2)$. ■

If $H_i = K_{n_i}^d$ for $i = 1, 2$, then $H_1 \widehat{\boxtimes} H_2 = H_1 \boxtimes_d H_2$ is a strong subhypergraph of $K_{n_1 n_2}^d$. Thus, by Theorem 3.2.12 and Lemma 5.1.4, we have $t(H_1 \boxtimes_d H_2) \leq t(d, n_1 n_2) \leq t(d, n_1) \times t(d, n_2) = t(K_{n_1}^d) \times t(K_{n_2}^d)$.

Now, we present an optimized Kronecker-type construction, as shown in Theorem 5.5.20, to obtain a result, which generalizes the result presented in Theorem 3.2.15 to the case of CFFs on hypergraphs.

Theorem 5.5.20. Let $H_1 = ([1, n_1], E_1)$ and $H_2 = ([1, n_2], E_2)$ be r -uniform hypergraphs. Then, $t(H_1 \boxtimes_r H_2) \leq t(r-1, \chi(H_2)) \cdot t(H_1) + t(H_2)$.

Proof. Let A_1 and A_2 be a $t_1 \times n_1$ $\overline{H_1}$ -disjunct matrix and a $t_2 \times n_2$ $\overline{H_2}$ -disjunct matrix, respectively. Let C be a $t_3 \times \chi(H_2)$ $(r-1)$ -disjunct matrix. We label the rows of A_1 , A_2 , and C with elements in $[1, t_1]$, $[1, t_2]$, and $[1, t_3]$, respectively. Now, we construct a matrix M as follows. Let M_1 and M_2 be matrices with columns corresponding to vertices (i, j) such that $i \in [1, n_1]$ and $j \in [1, n_2]$. For convenience, we keep the vertices of (i, j) together in the following order:

$$(1, 1), \dots, (n_1, 1), (1, 2), \dots, (n_1, 2), \dots, (1, n_2), \dots, (n_1, n_2).$$

Let C_* be a $t_3 \times n_2$ matrix with columns corresponding to vertices $[1, n_2]$ such that $C_*^j = C^k$, where C^k denotes the k^{th} column of C , if j belongs to the k^{th} color class of H_2 , for $k \in [1, \chi(H_2)]$. Let $M_1 = C_* \otimes A_1$, where the rows are labeled (a, b) with $a \in [1, t_3]$ and $b \in [1, t_1]$ and M_2 be the matrix such that $M_2^{(i,j)} = A_2^j$ with rows labeled as $[1, t_2]$. Let M be the vertical concatenation of M_1 and M_2 .

Now, we show that M is $(\overline{H_1 \boxtimes_r H_2})$ -disjunct. Let $e = \{(x_1, y_1), (x_2, y_2), \dots, (x_r, y_r)\} \in E(H_1 \boxtimes_r H_2)$. We show that M is cover-free and in-cover-free for the edge e . We consider the following cases for each type of e with $e' \subseteq e$ and $(x_{r+1}, y_{r+1}) \in V(H_1 \boxtimes_r H_2) \setminus e'$:

Case 1: $e = \{(x_1, y), (x_2, y), \dots, (x_r, y)\}$. This implies $I = \{x_1, x_2, \dots, x_r\} \in E_1$. Let $e' = \{(x_{i_1}, y), (x_{i_2}, y), \dots, (x_{i_{r'}}, y)\}$ for some $r' \leq r$ and $I' = \{x_{i_1}, x_{i_2}, \dots, x_{i_{r'}}\} \subseteq I$.

Case 1.1: $y_{r+1} = y$. Since A_1 is $\overline{H_1}$ -disjunct and $x_{r+1} \in [1, n_1] \setminus I'$, there exists a row b in A_1 such that $A_1[b, x_l] = 0$ for all $x_l \in I'$ and $A_1[b, x_{r+1}] = 1$. Let a be a row in C_* such that $C_*[a, y] = 1$. Thus, $M_1[(a, b), (x_l, y)] = 0$ for all $(x_l, y) \in e'$ and $M_1[(a, b), (x_{r+1}, y)] = 1$.

Case 1.2: $y_{r+1} = y' \neq y$. Since A_2 is H_2 -In-disjunct, there exists a row z in A_2 such that $A_2[z, y] = 0$ and $A_2[z, y'] = 1$. Then, for $x_{r+1} = x'$, we have $M_2^{(x', y')} = A_2^{y'}$ and for any $(x_l, y) \in e'$, we have $M_2^{(x_l, y)} = A_2^y$. Thus, $M_2[z, (x_l, y)] = 0$ for all $(x_l, y) \in e'$ and $M_2[z, (x', y')] = 1$.

Case 2: $e = \{(x, y_1), (x, y_2), \dots, (x, y_r)\}$. This implies $I = \{y_1, y_2, \dots, y_r\} \in E_2$. Let $e' = \{(x, y_{i_1}), (x, y_{i_2}), \dots, (x, y_{i_{r'}})\}$ for some $r' \leq r$ and $I' = \{y_{i_1}, y_{i_2}, \dots, y_{i_{r'}}\} \subseteq I$.

Case 2.1: $x_{r+1} = x$. Then $y_{r+1} \in [1, n_2] \setminus I'$. Since A_2 is $\overline{H_2}$ -disjunct, there exists a row z such that $A_2[z, y_l] = 0$ for all $y_l \in I'$ and $A_2[z, y_{r+1}] = 1$. Since $M_2^{(x, y_l)} = A_2^{y_l}$, we have $M_2[z, (x, y_l)] = 0$ for all $(x, y_l) \in e'$, and $M_2[z, (x, y_{r+1})] = 1$.

Case 2.2: $x_{r+1} \neq x$. We split this case into two subcases, depending on the choice of y_{r+1} .

Case 2.2.1: $y_{r+1} = y_i$ for some $y_i \in [1, n_2] \setminus I'$. This can be proven using the same technique as in Case 2.1.

Case 2.2.2: $y_{r+1} = y_i$ for some $y_i \in I'$. Since $I' \subseteq I$, the vertices of I' belong to distinct color classes of H_2 , and let the set of these color classes be $\{c_1, c_2, \dots, c_{r'}\} \subseteq [1, \chi(H_2)]$. Assume that each $y_l \in I'$ belongs to c_l for $l \in [1, r']$. Then, $C_*^{y_l} = C^{c_l}$. Since $|I'| \leq r$ and C is $(r-1)$ -disjunct, there exists a row a such that $C_*[a, y_l] = 0$ for all $y_l \in I' \setminus \{y_i\}$ and $C_*[a, y_i] = 1$. Since A_1 is H_1 -In-disjunct, there exists a row b in A_1 such that $A_1[b, x] = 0$ where $(x, y_i) \in e'$ and $A_1[b, x_{r+1}] = 1$. Thus, $M_1[(a, b), (x, y_l)] = 0$ for all $(x, y_l) \in e'$ and $M_1[(a, b), (x_{r+1}, y_i)] = 1$.

Case 3: $e = \{(x_1, y_1), (x_2, y_2), \dots, (x_r, y_r)\}$, where $I_1 = \{x_1, x_2, \dots, x_r\} \in E_1$ and $I_2 = \{y_1, y_2, \dots, y_r\} \in E_2$. Let $e' = \{(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \dots, (x_{i_{r'}}, y_{i_{r'}})\}$ for $r' \leq r$, where $I'_1 = \{x_{i_1}, x_{i_2}, \dots, x_{i_{r'}}\} \subseteq I_1$ and $I'_2 = \{y_{i_1}, y_{i_2}, \dots, y_{i_{r'}}\} \subseteq I_2$.

Case 3.1: $y_{r+1} \notin I'_2$. We have $(x_{r+1}, y_{r+1}) \in V(H_1 \boxtimes_r H_2) \setminus e'$. Since $M_2^{(x_d, y_d)} = A_2^{y_d}$ and A_2 is $\overline{H_2}$ -disjunct, there exists a row z in A_2 such that $A_2[z, y_l] = 0$ for all $y_l \in I'$ and $A_2[z, y_{r+1}] = 1$. Thus, there exists a row z in M_2 such that $M_2[z, (x_l, y_l)] = 0$ for $(x_l, y_l) \in e'$ and $M_2[z, (x_{r+1}, y_{r+1})] = 1$.

Case 3.2: $y_{r+1} \in I'_2$. Without loss of generality, let $y_{r+1} = y_{i_1}$. Thus, $x_{r+1} \neq x_{i_1}$. Using the same approach as in Case 2.2.2, we have $C_*^{y_l} = C^{c_l}$ where every vertex $y_l \in I'_2$ belongs to color class c_l . Thus, there exists a row a such that $C_*[a, y_l] = 0$ for all $y_l \in I' \setminus \{y_{i_1}\}$ and $C_*[a, y_{i_1}] = 1$. Since A_1 is H_1 -In-disjunct, there exists a row b in A_1 such that $A_1[b, x_{i_1}] = 0$ and $A_1[b, x_{r+1}] = 1$. Thus, $M_1[(a, b), (x_l, y_l)] = 0$ for all $(x_l, y_l) \in e'$ and $M_1[(a, b), (x_{r+1}, y_{i_1})] = 1$.

Thus, we establish that M is $(\overline{H_1 \boxtimes_r H_2})$ -disjunct, which implies that $t(H_1 \boxtimes_r H_2) \leq t_3 \cdot t_1 + t_2$. This also means $t(H_1 \boxtimes_r H_2) \leq t(r-1, \chi(H_2)) \cdot t(H_1) + t(H_2)$. ■

Let $H_i = K_{n_i}^d$ for $i = 1, 2$. Since $\chi(H_2) = n_2$ and $t(H_i) = t(d, n_i)$, by Lemma 5.1.5 and Theorem 5.5.20, $t(H_1 \boxtimes_d H_2) \leq t(K_{n_1 n_2}^d) \leq t(d-1, n_2) \times t(d, n_1) + t(d, n_2)$, which generalizes Theorem 3.2.15. The following corollary is also an immediate consequence of Theorem 5.5.20.

Corollary 5.5.21. Let $G_1 = ([1, n_1], E_1)$ and $G_2 = ([1, n_2], E_2)$ be two loopless graphs without isolated vertices. Then,

$$t(G_1 \boxtimes G_2) \leq t_s(G_2) \cdot t(G_1) + t(G_2).$$

Proof. Graphs G_1 and G_2 are 2-uniform hypergraphs and by Corollary 4.1.10, $t(1, \chi(G_2)) = t_s(G_2)$. The desired result then follows from Theorem 5.5.20. ■

The following theorem is applicable only to graphs and it is a generalization of the result presented in Theorem 3.2.16. If $G = K_n$, then $G \boxtimes K_2 = K_n \boxtimes K_2 = K_{2n}$. Since $t(K_{2n}) = t(2, 2n)$ and $\chi(K_n) = n$, Theorem 5.5.22 implies Theorem 3.2.16.

Theorem 5.5.22. Let $G = ([1, n], E)$ be a graph and A_1 be a $t_1 \times n$ $\overline{G}$ -disjunct matrix. Then, there exists a $t \times 2n$ $(G \boxtimes K_2)$ -disjunct matrix with $t = t_1 + t(1, \chi(G)) + 2$. If $t(1, \chi(G))$ is odd, then there exists a $t \times 2n$ $(G \boxtimes K_2)$ -disjunct matrix with $t = t_1 + t(1, \chi(G)) + 1$. Thus,

$$t(G \boxtimes K_2) \leq \begin{cases} t(G) + t(1, \chi(G)) + 2, & \text{if } t(1, \chi(G)) \text{ is even,} \\ t(G) + t(1, \chi(G)) + 1, & \text{if } t(1, \chi(G)) \text{ is odd.} \end{cases}$$

Proof. Let $V(K_2) = \{a, b\}$. Let $t(1, \chi(G)) = t'$ and A_2 be a $t' \times \chi(G)$ $\overline{1}$ -disjunct matrix which is the incidence matrix of the Sperner family consisting of subsets of size $\lceil \frac{t'}{2} \rceil$ of a t' -set. Label the columns of A_2 as elements in $[1, \chi(G)]$. Now, we construct matrices $\mathcal{M}_1$ and $\mathcal{M}_2$ as follows. Let M_1, M_2, M_3 , and M_4 be matrices where each column is labeled by the vertices of the graph $G \boxtimes K_2$, that is, (v, a) and (v, b) for $v \in V(G)$. For convenience, we arrange the vertices (v, a) for all $v \in V(G)$, followed by the vertices (v, b) for all $v \in V(G)$. Let $M_1^{(v,a)} = M_1^{(v,b)} = A_1^v$ for each $v \in V(G)$. If a vertex $v \in V(G)$ belongs to the i^{th} vertex-color class of G , then $M_2^{(v,a)} = A_2^i$ and $M_2^{(v,b)} = \overline{A_2^i}$ where $\overline{A_2^i}$ is the complement of A_2^i . Let M_3 and M_4 be row matrices such that $M_3[1, (v, a)] = 1, M_3[1, (v, b)] = 0, M_4[1, (v, a)] = 0$, and $M_4[1, (v, b)] = 1$ for all $v \in V(G)$. Let $\mathcal{M}_1$ be a vertical concatenation of M_1, M_2, M_3 , and M_4 and let $\mathcal{M}_2$ be a vertical concatenation of M_1, M_2 , and M_4 , as illustrated in Figure 5.10.

	(v, a)	(v, b)
	A_1	A_1
	A_2	$\overline{A_2}$
	$\mathbf{1}_{1 \times n}$	$\mathbf{0}_{1 \times n}$
	$\mathbf{0}_{1 \times n}$	$\mathbf{1}_{1 \times n}$

(a) In the case of even $t(1, \chi(G))$

	(v, a)	(v, b)
	A_1	A_1
	A_2	$\overline{A_2}$
	$\mathbf{0}_{1 \times n}$	$\mathbf{1}_{1 \times n}$

(b) In the case of odd $t(1, \chi(G))$

Figure 5.10: A $\overline{(G \boxtimes K_2)}$ -disjunct matrix

We show that $\mathcal{M}_1$ and $\mathcal{M}_2$ (in case of $t(1, \chi(G))$ odd) are $\overline{(G \boxtimes K_2)}$ -disjunct. Since $\delta(G \boxtimes K_2) \geq 2$, by Proposition 4.2.10, any $(G \boxtimes K_2)$ -disjunct matrix is also $\overline{(G \boxtimes K_2)}$ -disjunct. Hence, it is enough to show that $\mathcal{M}_1$ and $\mathcal{M}_2$ are $(G \boxtimes K_2)$ -disjunct. Let $\{(i_1, j_1), (i_2, j_2)\}$ be an edge in $G \boxtimes K_2$. We need to show that for any vertex $(i_3, j_3) \in V(G \boxtimes K_2) \setminus \{(i_1, j_1), (i_2, j_2)\}$, there exists a row r such that $\mathcal{M}_l[r, (i_1, j_1)] = \mathcal{M}_l[r, (i_2, j_2)] = 0$ and $\mathcal{M}_l[r, (i_3, j_3)] = 1$ for $l = 1$ and $l = 2$ when $t(1, \chi(G))$ is odd.

Case 1: $(i_1, j_1) = (i_1, a)$ and $(i_2, j_2) = (i_2, a)$.

Case 1.1: $(i_3, j_3) = (i_3, a)$. This implies $i_3 \notin \{i_1, i_2\}$. Then, $M_1^{(i_1, a)} = A_1^{i_1}$, $M_1^{(i_2, a)} = A_1^{i_2}$, and $M_1^{(i_3, a)} = A_1^{i_3}$. Since A_1 is G -disjunct and $\{i_1, i_2\} \in E(G)$, there exists a row r such that $A_1[r, i_1] = A_1[r, i_2] = 0$ and $A_1[r, i_3] = 1$. Thus, $M_1[r, (i_1, a)] = M_1[r, (i_2, a)] = 0$ and $M_1[r, (i_3, a)] = 1$.

Case 1.2: $(i_3, j_3) = (i_3, b)$. It is easy to see that $M_4[1, (i_1, a)] = M_4[1, (i_2, a)] = 0$ and $M_4[1, (i_3, b)] = 1$.

Case 2: $(i_1, j_1) = (i_1, b)$ and $(i_2, j_2) = (i_2, b)$.

Case 2.1: $(i_3, j_3) = (i_3, b)$. This implies $i_3 \neq i_1, i_2$. This case is similar to Case 1.1, so r is a row of M_1 .

Case 2.2: $(i_3, j_3) = (i_3, a)$. It is easy to see that $M_3[1, (i_1, b)] = M_3[1, (i_2, b)] = 0$ and $M_3[1, (i_3, a)] = 1$. Now we consider the case when $t(1, \chi(G))$ is odd showing that we do not need to rely on M_3 . This is further split into two cases.

Case 2.2.1: $i_3 \notin \{i_1, i_2\}$. This case can be proven using the same argument in Case 1.1, using M_1 .

Case 2.2.2: $i_3 \in \{i_1, i_2\}$. Without loss of generality, $i_3 = i_1$. Since $\{i_1, i_2\} \in E(G)$, let us say i_1 and i_2 belong to color classes c_1 and c_2 , respectively, for some $c_1, c_2 \in [1, \chi(G)]$ with $c_1 \neq c_2$. Then, $M_2^{(i_1, b)} = \overline{A_2^{c_1}}$, $M_2^{(i_2, b)} = \overline{A_2^{c_2}}$, and $M_2^{(i_1, a)} = A_2^{c_1}$. By Proposition 3.2.3, there

exists a row r such that $\overline{A_2}[r, c_1] = \overline{A_2}[r, c_2] = 0$. This also implies $A_2[r, c_1] = 1$. Thus, $M_2[r, (i_1, b)] = M_2[r, (i_2, b)] = 0$ and $M_2[r, (i_1, a)] = 1$.

Case 3: $(i_1, j_1) = (i, a)$ and $(i_2, j_2) = (i, b)$. Without loss of generality, assume $(i_3, j_3) = (i', a)$ for some $i' \neq i$. Then, $M_1^{(i,a)} = A_1^i$, $M_1^{(i,b)} = A_1^i$, and $M_1^{(i',a)} = A_1^{i'}$. Since A_1 is G -Sperner, there exists a row r such that $A_1[r, i] = 0$ and $A_1[r, i'] = 1$. Thus, $M_1[r, (i, a)] = M_1[r, (i, b)] = 0$ and $M_1[r, (i', a)] = 1$.

Case 4: $(i_1, j_1) = (i, a)$ and $(i_2, j_2) = (j, b)$, where $i \neq j$. Then, we have $\{i, j\} \in E(G)$.

Case 4.1: $(i_3, j_3) = (i, b)$. In this case, i, j belong to color classes c_1 and c_2 , respectively, where $c_1, c_2 \in [1, \chi(G)]$ with $c_1 \neq c_2$. Then, $M_2^{(i,a)} = A_2^i$, $M_2^{(j,b)} = \overline{A_2^j}$, and $M_2^{(i,b)} = \overline{A_2^i}$. Since $\overline{A_2}$ is $\overline{1}$ -disjunct, there exists a row r such that $\overline{A_2}[r, j] = 0$ and $\overline{A_2}[r, i] = 1$. This implies that $A_2[r, i] = 0$. Thus, $M_2[r, (i, a)] = M_2[r, (j, b)] = 0$ and $M_2[r, (i, b)] = 1$. We can also prove a similar result for the case when $(i_3, j_3) = (j, a)$.

Case 4.2: Without loss of generality, $(i_3, j_3) = (i', a)$ where $i' \notin \{i, j\}$. It is clear that $M_1^{(i,a)} = A_1^i$, $M_1^{(j,b)} = A_1^j$, and $M_1^{(i',a)} = A_1^{i'}$. The remainder of the proof follows as in Case 1.1.

Thus, we show that $\mathcal{M}_1$ is $(\overline{G \boxtimes K_2})$ -disjunct whereas if $t(1, \chi(G))$ is odd, $\mathcal{M}_2$ is $(G \boxtimes K_2)$ -disjunct. ■

5.6 Future Work

In this section, we list several future directions, most of which are currently under investigation by us.

1. In Section 5.2, we studied relationships between cover-free families and separable families on hypergraphs, and illustrated their relationship in Figure 5.5, analogously to Figure 5.2. We also showed that $\delta_{ex}(H) \geq 2$ is a necessary and sufficient condition for a clutter H so that any H -ECFF set system is also H -CFF. We are further interested in exploring bounds on $t(H)$ for such hypergraphs H when $\delta_{ex}(H) < 2$ and possibly generalize the results obtained in Theorem 4.2.11. We also ask for a hypergraph H , under what condition any H -separable matrix is also $\overline{H}$ -separable. Such a result would be a generalization of Theorem 5.2.2 item 1.
2. In Section 5.3, we characterized the existence of d -CFFs based on the existence of a directed hypergraph homomorphism $\overrightarrow{H}_{CF} \rightarrow \overrightarrow{CF}^{r+1}(t)$. We are interested in an analogous study about the existence of H -ECFFs. In this case, the corresponding underlying directed hypergraph is clearly less dense than $\overrightarrow{H}_{CF}$. We are motivated to study the properties of such directed hypergraphs, which could help us understand the structure of H -ECFFs as we show $t_e(H)$ to be close to $t(H)$ in many cases. Note that by Theorem 5.2.11, we have the following: if H is a clutter hypergraph with $\delta_{ex}(H) \geq 2$, then $t_e(H) = t(H)$.

3. Furthermore, we generalized the doubling construction in Theorem 3.2.16 for the case of graphs, obtaining an analogous bound on $t(G \boxtimes K_2)$ in Theorem 5.5.22. An interesting question is to identify some infinite families of graphs meeting some of the upper bounds on $t(G \boxtimes K_2)$.
4. Generalization of (H, r) -CFF: Motivated by the application of (H, r) -CFFs in combinatorial group testing arising from real-world scenarios, we are further interested in the generalization of (H, r) -CFF(t, n) by assuming a bound on the number of infected items inside a hyperedge. In the application related to infectious diseases within a community, there would be up to r communities infected with up to d infected items within each community.
5. Finally, investigating probabilistic existence of H -CFF(t, n) would be another interesting avenue of research.

Chapter 6

Gray codes and their generalizations

Sections 6.1 and 6.2 of this chapter provide an expository overview of Gray codes and their generalizations. The classical binary reflected Gray code was patented by Frank Gray [28] in 1953. Williams [73] provided a simple greedy algorithm that unifies the classical Gray codes in the literature into a general framework. Mixed-radix Gray codes, which are a generalization of the binary Gray code, have also been studied [11, 24]. An excellent resource for studying Gray codes and their generalizations is Donald Knuth’s *The Art of Computer Programming 4a* [43], which also describes two extensions of binary reflected Gray codes into a mixed-radix setting: the mixed-radix reflected Gray code and the mixed-radix modular Gray code. In Section 6.1, we define Gray codes and their connections to Hamming graphs. We begin Section 6.2 with a discussion related to lexicographic ordering in a mixed-radix setting and further move our discussion to mixed-radix reflected Gray codes and mixed-radix modular Gray codes in Sections 6.2.1 and 6.2.2, respectively. In Section 6.2.1, we present a recursive construction of reflected Gray codes and show its equivalence to a greedy increment/decrement based approach presented by Qiu et al. in [59]. In Section 6.2.2, we provide a recursive construction for modular Gray codes, namely *local recursion*. We also provide a greedy cyclic increment algorithm, generalized from the greedy Gray code algorithm introduced by Williams [73], and further conclude that the local recursion method, and the greedy cyclic increment algorithm generate the same order.

While the work presented in Sections 6.1 and 6.2 has been instrumental in proving the results related to CFFs on paths and cycles in Chapter 4 (in particular, Theorem 4.4.11), our greedy reinterpretation of modular Gray code by Knuth [43] led to an interesting observation in Section 6.3 and the novel work presented in Sections 6.4, and 6.5. The new work presented in these latter sections is largely independent of the rest of the thesis and so this chapter is somewhat self-contained. Given that this work has significant scope in the area of combinatorial generation and forms part of our joint work titled “One Sequence to Rule Them All: $\mathcal{O}(1)$ -Time Parallel Generation of Mixed-Radix Gray Codes” [51], which has been published at the 37th International Workshop in Combinatorial algorithms (IWOCA 2026), we present an overview along with relevant discussion and illustrative examples to help understand the new results presented in this paper, without going deeply into the proofs. Readers interested in the full proofs are encouraged to consult the complete version of the

paper when it becomes available. Moreover, Chapter 4 gives us a motivation for this work, as we were interested in constructing a cyclic ternary Gray code of length k to obtain a CFF on the cycle of length 3^k that improved best known bounds. The author of this thesis wrote Section 6.4, which also contributed to a section of the corresponding paper [51], and also contributed to the development of the algorithm presented in Section 6.5.2. The remaining parts of Sections 6.3 and 6.5 are adapted from the joint work [51] based on collaborative discussions.

We give a brief overview of Sections 6.3, 6.4, and 6.5 as follows. It is well known that binary ruler sequences guide the generation of the binary lexicographic order and the binary reflected Gray code. We further generalize the definition of the binary ruler sequence to a mixed-radix setting and discuss that the mixed-radix ruler sequence guides the generation of the mixed-radix lexicographic order, the mixed-radix reflected Gray codes, and the mixed-radix modular Gray codes. In addition, we show that the generation of these three orders can be fully parallelized, and moreover, we also show that the reflected and modular Gray codes can be generated in parallel in worst-case $\mathcal{O}(1)$ time per codeword. Furthermore, we provide a generalization of the modular Gray code that greedily cyclically increments any codeword that is lexicographically greater than or equal to a given codeword w on a mixed-radix base b . We show that, while the modified modular Gray code may not be a suffix of the modular Gray code, its change sequence is indeed a suffix of the mixed-radix ruler sequence. We further provide a loopless algorithm (one that generates each new codeword in constant time) for generating this modified modular Gray code.

6.1 Preliminaries

A *list* L is an ordered sequence of elements, that is, $L = [x_1, x_2, \dots, x_l]$ where x_i is the element at position i , for $1 \leq i \leq l$. In this case, the *length* of L , denoted by $|L|$, is l . In this chapter, we will be discussing lists of n -tuples from some finite alphabet. The *Hamming distance* between two n -tuples is the number of positions in which the corresponding coordinates differ. A *binary Gray code* G_n is a list of the 2^n binary n -tuples such that any two consecutive tuples have Hamming distance exactly one. Analogously, a mixed-radix Gray code G_n^b with $b = (m_1, m_2, \dots, m_n)$ is a list of n -tuples $a_1 a_2 \dots a_n$ with $0 \leq a_i < m_i$ such that only one digit is changed at a time. Hence, every binary Gray code G_n is G_n^b where $b = (\underbrace{2, 2, \dots, 2}_{n \text{ times}})$.

Throughout the rest of this thesis, the adjective “mixed-radix” will be dropped from the mixed-radix Gray code, and we will simply refer to the Gray code G_n^b and often omit the superscript for binary Gray codes.

Example 6.1.1. We provide an example of a binary Gray code G_4 :

$$G_4 = G_4^{(2,2,2,2)} = \begin{bmatrix} 0000, & 0001, & 0011, & 0010, & 0110, & 0111, & 0101, & 0100, \\ 1100, & 1101, & 1111, & 1110, & 1010, & 1011, & 1001, & 1000 \end{bmatrix}$$

In Example 6.1.1, note that the first and last codewords also have Hamming distance 1.

Minimal change ordering	Characteristic vector
$\emptyset$	000
$\{3\}$	001
$\{2, 3\}$	011
$\{2\}$	010
$\{1, 2\}$	110
$\{1, 2, 3\}$	111
$\{1, 3\}$	101
$\{1\}$	100

Table 6.1: The G_3 corresponds to a minimal change ordering of the subsets of $\{1, 2, 3\}$

Definition 6.1.2 (*Cyclic Gray code*). A Gray code is *cyclic* if the Hamming distance between the first and last codewords in the list is 1.

Constructing a Gray code (cyclic Gray code) is equivalent to finding a Hamilton path (cycle) in its *flip graph*, which is the graph where each vertex is a tuple and an edge connects two tuples if and only if they have Hamming distance 1 [52]. The flip graph for a binary Gray code of length n is the hypercube graph of dimension n (Q_n). Similarly, the flip graph for a Gray code $G_n^{(m_1, m_2, \dots, m_n)}$ is the Hamming graph $H_{m_1, m_2, \dots, m_n}$ (defined in Definition 2.2.14).

Binary Gray codes can also be viewed as the characteristic vectors of the subsets of an n -set in a *minimal change ordering* [44], where one set differs from the next by the addition or the removal of an element.

Example 6.1.3. In this example with $n = 3$, Table 6.1 provides a list of subsets in $\mathcal{P}([1, 3])$ using a minimal change ordering, while the corresponding characteristic vectors in this order produce a $G_3 = G_3^{(2, 2, 2)}$.

6.2 Lexicographic order and Gray codes

Unless otherwise specified, throughout the rest of the section, let $b = (m_1, m_2, \dots, m_n)$. We denote $\prod_{i=1}^n m_i$ by $b!$.

Notation 6.2.1. Denote by $\mathbb{W}(b)$ the set of mixed-radix codewords with $b = (m_1, m_2, \dots, m_n)$.

For example, $\mathbb{W}(b) = \{00, 01, 02, 10, 11, 12\}$ for $b = (2, 3)$. The number of such codewords is $|\mathbb{W}(b)| = b!$. A list of tuples in *lexicographic ordering*, denoted by L_n^b , is the ordering of the n -tuples $a_1 \dots a_n$ with $0 \leq a_i < m_i$, obtained by arranging the tuples in increasing order of their unique integer value $j \in [0, b! - 1]$ where $a_1 \dots a_n$ is the representation of j using base b ; that is, j is uniquely determined by the formula:

$$j = \frac{b!}{m_1} a_1 + \frac{b!}{m_1 m_2} a_2 + \dots + \frac{b!}{m_1 m_2 \dots m_i} a_i + \dots + m_n a_{n-1} + a_n.$$

j	L_3^b
0	000
1	001
2	002
3	010
4	011
5	012
6	020
7	021
8	022
9	100
10	101
11	102
12	110
13	111
14	112
15	120
16	121
17	122

Table 6.2: L_3^b for $b = (2, 3, 3)$

Definition 6.2.2. The *lexicographic rank* of $w \in \mathbb{W}(b)$, denoted by $\text{Rank}^b(w)$, is the number of codewords in $\mathbb{W}(b)$ that are lexicographically smaller than w . In other words, if $L_n^b = [w_0, w_1, \dots, w_{b!-1}]$, then $\text{Rank}^b(w) = r$ where $w_r = w$.

Definition 6.2.3. For a given base $b = (m_1, m_2, \dots, m_n)$, two n -tuples $v, w \in \mathbb{W}(b)$, v is *lexicographically greater than* w if $\text{Rank}^b(v) > \text{Rank}^b(w)$.

Let $\mathbb{W}_{(w)}(b) \subseteq \mathbb{W}(b)$ be the subset of codewords that are lexicographically greater than or equal to w . We also denote this set by $\mathbb{W}_r(b)$, where $r = \text{Rank}^b(w)$. Thus, we have $\mathbb{W}_r(b) = \mathbb{W}_{(w)}(b)$, since the codewords in $\mathbb{W}_r(b)$ and $\mathbb{W}_{(w)}(b)$ are precisely those with lexicographic rank at least r .

Example 6.2.4. Let $b = (2, 3, 3)$. Table 6.2 provides the orders L_3^b and L_3^b . Note that $\text{Rank}^b(101) = 10$. Let $w = 101$. Then, $\mathbb{W}_{(101)}(b) = \mathbb{W}_{10}(b) = \{101, 102, 110, 111, 112, 120, 121, 122\}$.

In the next sections, we discuss the reflected Gray code and the modular Gray code constructed on a base $b = (m_1, m_2, \dots, m_n)$, which are denoted by $R_n^{(m_1, m_2, \dots, m_n)}$ (or R_n^b) and $M_n^{(m_1, m_2, \dots, m_n)}$ (or M_n^b), respectively. If $m_i = k$ for all $i \in [1, n]$, then we denote the corresponding reflected Gray code and the modular Gray code by R_n^k and M_n^k , respectively. If $k = 2$, then it is easy to see that $R_n^2 = M_n^2$.

6.2.1 Reflected Gray codes and the greedy increment/decrement algorithm

For a list of m codewords, $C = [C_0, C_1, \dots, C_{m-2}, C_{m-1}]$, the *reflection* of C is denoted $C^R = [C_{m-1}, C_{m-2}, \dots, C_1, C_0]$. For an n -tuple $C_i = (a_1, a_2, \dots, a_n)$, $x \cdot C_i$ denotes $(x, a_1, a_2, \dots, a_n)$ and $x \cdot C$ denotes $[x \cdot C_0, x \cdot C_1, \dots, x \cdot C_{m-2}, x \cdot C_{m-1}]$. Similarly, $C_i \cdot x$ denotes $(a_1, a_2, \dots, a_n, x)$ and $C \cdot x$ denotes $[C_0 \cdot x, C_1 \cdot x, \dots, C_{m-2} \cdot x, C_{m-1} \cdot x]$.

A simple recursive construction of the *reflected Gray code* R_n^b is provided below.

Construction 6.2.5. (Reflected Gray code)

$$R_1^{(m_n)} = \mathcal{C}_1 := [0, 1, 2, \dots, m_n - 1]. \text{ For } i \geq 2,$$

$$R_{i+1}^{(m_n-i, m_n-i+1, \dots, m_n)} = \mathcal{C}_{i+1} := \begin{cases} [0 \cdot \mathcal{C}_i, 1 \cdot \mathcal{C}_i^R, 2 \cdot \mathcal{C}_i, \dots, (m_n-i-1) \cdot \mathcal{C}_i^R], & \text{if } m_n-i \text{ is even} \\ [0 \cdot \mathcal{C}_i, 1 \cdot \mathcal{C}_i^R, 2 \cdot \mathcal{C}_i, \dots, (m_n-i-1) \cdot \mathcal{C}_i], & \text{if } m_n-i \text{ is odd} \end{cases}$$

R_n^2 is famously known as the *binary reflected Gray code*. In Examples 6.1.1 and 6.1.3, the $G_4^{(2,2,2,2)}$ and $G_3^{(2,2,2)}$ are also the reflected Gray codes $R_4^{(2,2,2,2)}$ (or R_4^2) and $R_3^{(2,2,2)}$ (or R_3^2), respectively.

Qiu, Sawada, and Williams [59] observe that R_n^b can also be “greedily” generated starting from the zero-word 0^n , as discussed next. For each $1 \leq l \leq n$, Let $\pm \text{bit}_l$ denote the operation that increments or decrements the l^{th} bit of an n -tuple in R_n^b , where increment takes precedence over decrement. Thus, we define the following sequence of prioritized operations for R_n^b :

$$\text{Bit}(n) = (\pm \text{bit}_n, \pm \text{bit}_{n-1}, \dots, \pm \text{bit}_2, \pm \text{bit}_1).$$

Using $\text{Bit}(n)$, we can generalize the greedy approach introduced by Williams [73] for generating R_n^2 , which prioritizes flipping bits from right to left; that is, at each step, it flips the rightmost bit of the current codeword whose flip produces a new codeword. In R_n^b , the bit is either incremented or decremented, instead of flipped.

Example 6.2.6. The order of the codewords in the reflected Gray code $R_4^{(2,3,2,2)}$ is provided below:

$$R_4^{(2,3,2,2)} = \begin{array}{l} [0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 0200, 0201, 0211, 0210, \\ 1210, 1211, 1201, 1200, 1100, 1101, 1111, 1110, 1010, 1011, 1001, 1000] \end{array}$$

Table 6.3 lists the codewords of $R_4^{(2,3,2,2)}$ that are greedily generated via the greedy increment/decrement algorithm. At each step, we specify the operation with the highest priority, which also determines the bit to be incremented or decremented. The bit that is changed at each step is marked in **bold**.

$R_4^{(2,3,2,2)}$	prioritized operation
0000	+bit ₄
0001	+bit ₃
0011	−bit ₄
0010	+bit ₂
0110	+bit ₄
0111	−bit ₃
0101	−bit ₄
0100	+bit ₂
0200	+bit ₄
0201	+bit ₃
0211	−bit ₄
0210	+bit ₁
1210	+bit ₄
1211	−bit ₃
1201	−bit ₄
1200	−bit ₂
1100	+bit ₄
1101	+bit ₃
1111	−bit ₄
1110	−bit ₂
1010	+bit ₄
1011	−bit ₃
1001	−bit ₄
1000	

Table 6.3: The greedy interpretation of $R_4^{(2,3,2,2)}$

Example 6.2.7. An example of the reflected Gray code $R_4^{(4,3,3,2)}$ is given below:

$$R_4^{(4,3,3,2)} = [0000, 0001, 0011, 0010, 0020, 0021, 0121, 0120, 0110, 0111, 0101, 0100, 0200, 0201, 0211, 0210, 0220, 0221, 1221, 1220, 1210, 1211, 1201, 1200, 1100, 1101, 1111, 1110, 1120, 1121, 1021, 1020, 1010, 1011, 1001, 1000, 2000, 2001, 2011, 2010, 2020, 2021, 2121, 2120, 2110, 2111, 2101, 2100, 2000, 2001, 2011, 2010, 2020, 2021, 2121, 2120, 2110, 2111, 2101, 2100, 2200, 2201, 2211, 2210, 2220, 2221, 3221, 3220, 3210, 3211, 3201, 3200, 3100, 3101, 3111, 3110, 3120, 3121, 3021, 3020, 3010, 3011, 3001, 3000]$$

Example 6.2.8. An example of the reflected Gray code $R_4^{(3,3,3,2)}$ is given below:

$$R_4^{(3,3,3,2)} = \begin{aligned} &[0000, 0001, 0011, 0010, 0020, 0021, 0121, 0120, 0110, \\ &0111, 0101, 0100, 0200, 0201, 0211, 0210, 0220, 0221, \\ &1221, 1220, 1210, 1211, 1201, 1200, 1100, 1101, 1111, \\ &1110, 1120, 1121, 1021, 1020, 1010, 1011, 1001, 1000, \\ &2000, 2001, 2011, 2010, 2020, 2021, 2121, 2120, 2110, \\ &2111, 2101, 2100, 2200, 2201, 2211, 2210, 2220, 2221] \end{aligned}$$

In Examples 6.1.1, 6.1.3, 6.2.6, and 6.2.7, the first and last codewords have Hamming distance 1, so these Gray codes are cyclic. The following proposition provides a necessary and sufficient condition for the reflected Gray code to be cyclic.

Proposition 6.2.9. The reflected Gray code $R_n^{(m_1, m_2, \dots, m_n)}$ is cyclic if and only if m_1 is even.

Proof. From Construction 6.2.5, it is easy to see that the first codeword is always the all-zero tuple, and if m_1 is even, the last codeword is an n -tuple starting with $m_1 - 1$, followed by all zeros, since we append the reverse order of $R_{n-1}^{(m_2, \dots, m_n)}$ to $m_1 - 1$. Thus, $R_n^{(m_1, m_2, \dots, m_n)}$ is cyclic.

For the other direction, assume $R_n^{(m_1, m_2, \dots, m_n)}$ is cyclic. Then, the first and last codewords also have Hamming distance 1. Hence, by Construction 6.2.5, the last codeword must be the n -tuple starting with $m_1 - 1$ followed by all 0 and this is possible only if m_1 is even. ■

Construction 6.2.5 implies that each codeword is greedily generated by identifying the rightmost bit that can be changed by $+1$ or -1 to produce a new valid codeword. In general, the transitions for the reflected Gray code at the i^{th} co-ordinate are $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow \dots \rightarrow m_i - 2 \rightarrow m_i - 1 \rightarrow m_i - 2 \rightarrow \dots \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$.

6.2.1.1 Strongly cyclic reflected Gray codes

Knuth also discusses a stronger notion of cyclic Gray codes in [43]. In Example 6.2.6, the codeword after 1000 is 0000, thus following the natural transition $0 \rightarrow 1 \rightarrow 0$ for the first bit. In Example 6.2.7, the codeword after 3000 is 0000, which does not follow the transition for the first bit, which is $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$.

Definition 6.2.10 (*Strongly cyclic reflected Gray Code*). A reflected Gray code is *strongly cyclic* if the transition from the last codeword to the first follows the usual transition in the bit i that changes.

Therefore, the Gray code presented in Example 6.2.6 is strongly cyclic, while the one in Example 6.2.7 is not. Given the Gray code $R_n^{(m_1, m_2, \dots, m_n)}$, by Construction 6.2.5, the first and last codewords are 0^n and $(m_1 - 1)0^{n-1}$ if m_1 is even. Since the natural transition at the first co-ordinate is $0 \rightarrow 1 \rightarrow \dots \rightarrow m_1 - 1 \rightarrow \dots \rightarrow 1 \rightarrow 0$ and since the first codeword is the zero-word, it implies that $R_n^{(m_1, m_2, \dots, m_n)}$ is strongly cyclic if and only if the first co-ordinate from the last codeword to the first codeword follows the transition: $m_1 - 1 \rightarrow 0$. It follows that $m_1 - 1$ must be 1, that is, $m_1 = 2$. This proves the following theorem provided in [43].

Theorem 6.2.11. [43] The reflected Gray code R_n^b is strongly cyclic if and only if $m_1 = 2$.

It is easy to see that if a Gray code is strongly cyclic, then it is also cyclic, whereas the reverse is not necessarily true.

6.2.2 Modular Gray codes

Knuth also discusses *modular Gray codes* in [43] where successive codewords differ in one bit (say at the i^{th} position) by $+1$ modulo its base m_i . The term ‘modular’ indicates that the ordering is generated by a sequential rule in $\mathbb{Z}_{m_i}$ or ‘cyclically wrapping around’. Hence, the transition for the modular Gray code at i^{th} co-ordinate is $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow \dots \rightarrow m_i - 1 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow \dots$.

Let $G_n^{(m_1, m_2, \dots, m_{n-1}, m_n)}$ be an ordered list of tuples of length n . For $n \geq 2$, we express the construction of $G_n^{(m_1, m_2, \dots, m_{n-1}, m_n)}$ recursively by referring to the individual codewords in $G_{n-1}^{(m_1, m_2, \dots, m_{n-1})}$ and we refer to this method as *local recursion*. Now, we provide a simple local recursive construction for modular Gray codes, M_n^b .

Notation 6.2.12. For a codeword C in $G_{n-1}^{(m_1, m_2, \dots, m_{n-1})}$ and $j \in \mathbb{Z}_{m_n}$, denote $C \cdot j^+ = [C \cdot (j \bmod m_n), C \cdot ((j+1) \bmod m_n), \dots, C \cdot ((j+m_n-1) \bmod m_n)]$.

Construction 6.2.13. (Local recursion method) Let $m_1, m_2, \dots, m_n > 1, n \geq 1$.

$$M_1^{(m_1)} = [0, 1, \dots, m_1 - 1].$$

For $j \geq 2$, if $M_{j-1}^{(m_1, m_2, \dots, m_{j-1})} = [C_0, C_1, \dots, C_{(m_1 m_2 \dots m_{j-1})-1}]$,

$$\text{then } M_j^{(m_1, m_2, \dots, m_{j-1}, m_j)} = [C_0 \cdot 0^+, C_1 \cdot (m_j - 1)^+, \dots, C_i \cdot ((m_j - i) \bmod m_j)^+, \dots, C_{(m_1 m_2 \dots m_{j-1})-1} \cdot ((m_j - m_1 m_2 \dots m_{j-1} + 1) \bmod m_j)^+].$$

Example 6.2.14. An example of the modular Gray code $M_2^{(3,3)}$ given below:

$$M_2^{(3,3)} = [00, 01, 02, 12, 10, 11, 21, 22, 20].$$

Example 6.2.15. An example of the modular Gray code $M_4^{(4,2,2,4)}$ is given below:

$$M_4^{(4,2,2,4)} = [0000, 0001, 0002, 0003, 0013, 0010, 0011, 0012, 0112, 0113, 0110, 0111, 0101, 0102, 0103, 0100, 1100, 1101, 1102, 1103, 1113, 1110, 1111, 1112, 1012, 1013, 1010, 1011, 1001, 1002, 1003, 1000, 2000, 2001, 2002, 2003, 2013, 2010, 2011, 2012, 2112, 2113, 2110, 2111, 2101, 2102, 2103, 2100, 3100, 3101, 3102, 3103, 3113, 3110, 3111, 3112, 3012, 3013, 3010, 3011, 3001, 3002, 3003, 3000].$$

6.2.2.1 The greedy cyclic increment algorithm

We provide the generation of M_n^b using a greedy cyclic increment approach [51].

Construction 6.2.16. 1. **Initialization:** Let $\mathcal{L}$ be an empty list. Set the initial codeword to $C_0 = 0^n$. Add C_0 to $\mathcal{L}$.

2. **Generation:** While the size of $\mathcal{L}$ is less than $\prod_{i=1}^n m_i$, let C_j be the current last codeword in $\mathcal{L}$, for some $0 \leq j < (\prod_{i=1}^n m_i) - 1$. For each bit in C_j , starting from the rightmost bit (highest index) to the leftmost bit (lowest index), change the bit c_l for some $l \in [1, n]$ in a cyclic manner ($0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow m_l - 1 \rightarrow 0 \rightarrow \dots$) by incrementing the bit value modulo m_l . After changing the bit, check if the resulting codeword is already in $\mathcal{L}$. If it is not, add it to $\mathcal{L}$. If it is in $\mathcal{L}$, continue to the bit on its left. This process is continued until a new codeword is found and added to $\mathcal{L}$.

3. Terminate when $\mathcal{L}$ has $\prod_{i=1}^n m_i$ distinct codewords.

4. Return $\mathcal{L}$.

Let us denote $\mathcal{L} = \text{Greedy}_{\text{Bit}(n)}^b(0^n)$. For each $1 \leq l \leq n$, let bit_l be the operation that increments the l^{th} bit of an n -tuple, that is, for a codeword $C = c_1 c_2 \dots c_{n-1} c_n$, we define $\text{bit}_l(C) = c_1 c_2 \dots c_{l-1} (c_l + 1 \bmod m_l) c_{l+1} \dots c_{n-1} c_n$. As we prioritize increments from right to left, we define the sequence of prioritized operations as

$$\text{Bit}(n) = (\text{bit}_n, \text{bit}_{n-1}, \dots, \text{bit}_2, \text{bit}_1).$$

Example 6.2.17.

$$M_3^3 = \text{Greedy}_{\text{Bit}(3)}^{(3,3,3)}(0^3) = \begin{bmatrix} 00\bar{0}, & 00\bar{1}, & 0\bar{0}2, & 01\bar{2}, & 01\bar{0}, & 0\bar{1}1, & 02\bar{1}, & 02\bar{2}, & \bar{0}20, \\ 12\bar{0}, & 12\bar{1}, & 1\bar{2}2, & 10\bar{2}, & 10\bar{0}, & 1\bar{0}1, & 11\bar{1}, & 11\bar{2}, & \bar{1}10, \\ 21\bar{0}, & 21\bar{1}, & 2\bar{1}2, & 22\bar{2}, & 22\bar{0}, & 2\bar{2}1, & 20\bar{1}, & 20\bar{2}, & 200 \end{bmatrix}$$

where overlines reflect the bit to be incremented in order to create the next codeword.

Proposition 6.2.18. Given integers $m_1, m_2, \dots, m_n > 1$ and $n \geq 1$,

$$M_n^b = \text{Greedy}_{\text{Bit}(n)}^b(0^n).$$

Proof. The proof is by induction on n . For $n = 1$, it is clear that

$$\text{Greedy}_{\text{Bit}(1)}^{(m_1)}(0^1) = [0, 1, 2, \dots, (m_1 - 1)] = G_1^{(m_1)}.$$

Induction Hypothesis: For some $n \geq 2$, assume that

$$\text{Greedy}_{\text{Bit}(n-1)}^{(m_1, m_2, \dots, m_{n-1})}(0^{n-1}) = [C_0, C_1, \dots, C_{m_1 m_2 \dots m_{n-1} - 1}] = G_{n-1}^{(m_1, m_2, \dots, m_{n-1})}.$$

Since $C_0 = 0^{n-1}$, then C_i must be $0^{n-2}i$ for $1 \leq i \leq m_{n-1} - 1$. Then the first $m_{n-1}m_n$ n -tuples generated by $\text{Greedy}_{\text{Bit}(n)}^{(m_1, m_2, \dots, m_{n-1}, m_n)}(0^n)$ are

$$\begin{aligned}
& 0^{n-1}0, 0^{n-1}1, \dots, 0^{n-1}(m_n - 1), \\
& 0^{n-2}1(m_n - 1), 0^{n-2}10, 0^{n-2}11, \dots, 0^{n-2}1(m_n - 2), \\
& 0^{n-2}2(m_n - 2), 0^{n-2}2(m_n - 1), \dots, 0^{n-2}2(m_n - 3), \\
& \vdots \\
& 0^{n-2}(m_{n-1}-1)((l+1) \bmod m_n), 0^{n-2}(m_{n-1}-1)((l+2) \bmod m_n), \dots, 0^{n-2}(m_{n-1}-1)(l \bmod m_n).
\end{aligned}$$

where $l = m_n - m_{n-1}$. The above n -tuples can also be represented as

$$C_0 \cdot 0^+, C_1 \cdot (m_n - 1)^+, C_2 \cdot (m_n - 2)^+, \dots, C_{m_{n-1}-1} \cdot ((m_n - m_{n-1} + 1) \bmod m_n)^+.$$

Now, let us assume that $\text{Greedy}_{\text{Bit}(n)}^{(m_1, m_2, \dots, m_{n-1}, m_n)}(0^n)$ starts with

$$\begin{aligned}
& C_0 \cdot 0^+, C_1 \cdot (m_n - 1)^+, C_2 \cdot (m_n - 2)^+, \dots, \\
& C_{m_{n-1}-1} \cdot ((m_n - m_{n-1} + 1) \bmod m_n)^+, \dots, C_i \cdot ((m_n - i) \bmod m_n)^+.
\end{aligned} \tag{6.2.1}$$

for some $m_{n-1} \leq i < (\prod_i m_i) - 1$.

The n -tuple following $C_i \cdot (m_n - i - 1 \bmod m_n)$ in (6.2.1) must also have $m_n - i - 1$ as its rightmost bit. Note that the tuples ending with $m_n - i - 1$ within (6.2.1) are precisely $C_0, C_1, \dots, C_{m_{n-1}-1}, \dots, C_i$. Assume that $\text{Greedy}_{\text{Bit}(n-1)}^{(m_1, m_2, \dots, m_{n-1})}(0^{n-1})$ generates $C_0, C_1, \dots, C_{m_{n-1}-1}, \dots, C_i, C_{i+1}, \dots$ with C_{i+1} resulting from C_i via a prioritized operation bit_l for some $l \in [1, n]$. This implies that the subsequent n -tuples of $\text{Greedy}_{\text{Bit}(n)}^{(m_1, m_2, \dots, m_{n-1}, m_n)}(0^n)$ appearing in (6.2.1) must be codewords in the ordering $C_{i+1} \cdot ((m_n - i - 1) \bmod m_n)^+$. In particular, the successor of $C_i \cdot (m_n - i - 1)$ in $\text{Greedy}_{\text{Bit}(n)}^{(m_1, m_2, \dots, m_{n-1}, m_n)}(0^n)$ must be $C_{i+1} \cdot (m_n - i - 1)$, which follows from the prioritized operation bit_l .

Therefore, (6.2.1) holds for $i + 1$, and by induction, it follows for $i = q^{n-1} - 1$. Hence, $\text{Greedy}_{\text{Bit}(n)}^q(0^n)$ and G_n^q produce the same Gray code. This completes the induction proof. ■

Corollary 6.2.19. Let M_n^b be a mixed-radix modular Gray code constructed using Construction 6.2.13. Then, its last codeword/tuple is

$$\begin{aligned}
C_{m_1 m_2 \dots m_{n-1}} = & \left(m_1 - 1, (m_2 - m_1) \bmod m_2, (m_3 - m_1 m_2) \bmod m_3, \dots, \right. \\
& \left. \left(m_k - \prod_{i=1}^{k-1} m_i \right) \bmod m_k, \dots, \left(m_n - \prod_{i=1}^{n-1} m_i \right) \bmod m_n \right).
\end{aligned}$$

Proof. The proof is by induction on n . According to Construction 6.2.13, the result is true for $n = 1$, since the last codeword is $m_1 - 1$.

Induction hypothesis: Assume that the result is true for a mixed-radix modular Gray code $M_{n-1}^{(m_1, m_2, \dots, m_{n-1})}$ constructed recursively, for some $n \geq 2$. Thus, we have

$$C_{m_1 m_2 \dots m_{n-1}-1} = \left(m_1 - 1, (m_2 - m_1) \bmod m_2, (m_3 - m_1 m_2) \bmod m_3, \dots, \right.$$

$$\left(m_{n-1} - \prod_{i=1}^{n-2} m_i \right) \bmod m_{n-1}.$$

In the inductive step, we would like to find the codeword $C_{m_1 m_2 \dots m_{n-1} m_n - 1}$ in $G_n^{(m_1, m_2, \dots, m_n)}$. By induction hypothesis and the definition of $C \cdot j^+$, the sublist $C_{m_1 m_2 \dots m_{n-1} - 1} \cdot ((m_n - m_1 m_2 \dots m_{n-1} + 1) \bmod m_n)^+$ begins with the codeword

$$\left(m_1 - 1, (m_2 - m_1) \bmod m_2, \dots, \left(m_{n-1} - \prod_{i=1}^{n-2} m_i \right) \bmod m_{n-1}, \left(m_n - \prod_{i=1}^{n-1} m_i + 1 \right) \bmod m_n \right)$$

and ends with the codeword

$$\left(m_1 - 1, (m_2 - m_1) \bmod m_2, \dots, \left(m_{n-1} - \prod_{i=1}^{n-2} m_i \right) \bmod m_{n-1}, \left(m_n - \prod_{i=1}^{n-1} m_i \right) \bmod m_n \right),$$

which is exactly $C_{m_1 m_2 \dots m_{n-1} m_n - 1}$. ■

6.2.2.2 Strongly cyclic modular Gray code

Analogous to the definition for strongly cyclic reflected Gray code, we also provide the same for the modular Gray code.

Definition 6.2.20 (*Strongly cyclic modular Gray code*). A modular Gray code is *strongly cyclic* if the transition from the last codeword to the first follows the usual transition in the first bit, that is, $0 \rightarrow 1 \rightarrow \dots \rightarrow m_1 - 1 \rightarrow 0$.

In Examples 6.2.14 and 6.2.15, the last and first codewords follow the natural transition for modular Gray code for the first bit, which is $0 \rightarrow 1 \rightarrow 2 \rightarrow 0$ and $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$, respectively.

From Corollary 6.2.19, note that the modular Gray code $M_n^{(m_1, m_2, \dots, m_{n-1}, m_n)}$ is strongly cyclic if and only if $C_{(m_1 m_2 \dots m_n) - 1} = (m_1 - 1, 0, 0, \dots, 0) = (m_1 - 1)0^{n-1}$. This happens if and only if

$$\begin{aligned} m_2 &| m_1 \\ m_3 &| m_1 m_2 \\ &\vdots \\ m_n &| \prod_{i=1}^{n-1} m_i. \end{aligned}$$

This provides an alternate proof to the following theorem proven in [43].

Theorem 6.2.21. [43] The modular Gray code $M_n^{(m_1, m_2, \dots, m_n)}$ is strongly cyclic if and only if $m_1 m_2 \dots m_j \bmod m_{j+1} = 0$ for $1 \leq j \leq n - 1$.

6.3 Change sequences of reflected and modular Gray codes and their loopless generations

6.3.1 Loopless algorithm for binary reflected Gray codes

While two consecutive codewords in a Gray code differ in exactly one bit, two consecutive codewords in lexicographic order can differ in every bit. For example, in L_n^b for $b = (m_1, m_2, \dots, m_n)$, the codeword $0(m_2 - 1)(m_3 - 1) \dots (m_n - 1)$ is followed by 10^{n-1} . However, it is possible to explain this order of changing bits based on b through a sequence, called the *ruler sequence* and we denote it by $\text{Ruler}(b)$. To further discuss this, we compare L_3^b and R_3^b for $b = (2, 2, 2)$.

$$L_3^b = 00\bar{0}, 00\bar{1}, 01\bar{0}, 01\bar{1}, 10\bar{0}, 10\bar{1}, 11\bar{0}, 11\bar{1} \quad (6.3.1)$$

$$M_3^b = R_3^b = 00\bar{0}, 00\bar{1}, 01\bar{1}, 01\bar{0}, 11\bar{0}, 11\bar{1}, 10\bar{1}, 10\bar{0} \quad (6.3.2)$$

$$\text{Ruler}(b) = \quad 1, \quad 2, \quad 1, 3, \quad 1, \quad 2, \quad 1 \quad (6.3.3)$$

Here, the overlines denote bits that are flipped to create the next codeword. Note that when L_3^b flips the rightmost i bits, R_3^b only flips the i^{th} rightmost bit. Successive values of i create a mutual *change sequence*, denoted by $\text{Ruler}(b) = 1, 2, 1, 3, 1, 2, 1$, famously known as the *binary ruler sequence* (OEIS A001511) [66]. It is well known that, the binary ruler sequence provides the index of the single bit that changes in R_n^2 and the indices of the bits from right-to-left in L_n^2 . This sequence gives the largest power of 2 that divides $2n, n \geq 1$:

$$1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 6, 1, \dots$$

Given sequence $S = s_1, s_2, \dots, s_k$, a *subsequence* is any sequence of the form $s_i, s_{i+1}, \dots, s_j$; it is a *prefix* if $i = 1$, a *suffix* if $j = k$, *non-empty* if $i \leq j$, and *strict* if $i \neq 1$ or $j \neq k$.

The greedy generation of R_n^2 is not suitable for its efficient generation since it needs to memorize an exponential number of codewords relative to n . However, the concept of change sequence of R_n^2 leads to an efficient algorithm provided by Bitner, Ehrlich, and Reingold [7] to generate R_n^2 where it takes a worst-case $\mathcal{O}(1)$ -time to generate the successor of the current codeword. Such an algorithm is called a *loopless algorithm*. We have presented it in Algorithm 1, with the codeword positions indexed starting from 0.

Algorithm 1 produces each new codeword from its predecessor by efficiently computing the position of the bit to be flipped using a constant number of operations, independent of n . For this purpose, the algorithm uses an array F with $n + 1$ positions, called *focus pointers*, F to keep track of the bits to be changed in each iteration. In particular, in every iteration of the algorithm, $j = F[0]$ corresponds to the index of the bit in the current codeword R that is to be flipped to generate the next codeword. After the execution of this particular step, F is updated in a way so that the next required index to be flipped eventually moves to 0^{th} position of F . Such sequence of j is indeed the change sequence of R_n^2 , which also guides the generation of L_n^2 . Since we use 0-based indexing in the algorithm, each such j is one less

than the corresponding value in the binary ruler sequence. In particular, the generation of F interacts particularly well with the generation of L_n^2 .

Let $b = (\underbrace{2, \dots, 2}_{n \text{ times}})$ and $r = \text{Rank}^b(w)$. In other words, w is the binary representation of r . When the rank is incremented by 1, a suffix of w of the form $01 \dots 1$ changes to $10 \dots 0$, corresponding to the effect of “carry” in binary base. The length l of such suffix is the value of the element of rank r in $\text{Ruler}(b)$. The focus pointers, F , are used to keep track of maximal strings of 1s as a codeword w changes according to L_n^2 , without storing w explicitly, in such a way that an increment in w can be equivalently updated in F in constant time.

We provide a formal definition of focus pointers as follows.

Definition 6.3.1. Let $w = a_{n-1} \dots a_1 a_0$ be a codeword of length n in L_n^2 . Let S be the set of pair of indices (i, j) where $a_i a_{i+1} \dots a_j$ is a maximal substring of 1’s in w ; that is, $S = \{(i, j) : a_i = a_{i+1} = \dots = a_j = 1 \text{ and } (a_{i-1} = 0 \text{ or } i = 0) \text{ and } (a_{j+1} = 0 \text{ and } j = n - 1)\}$. Then, the focus pointer $f = f_n f_{n-1} \dots f_1 f_0$ is

$$f_i = \begin{cases} j + 1, & \text{if } (i, j) \in S, \\ i, & \text{otherwise.} \end{cases} \quad (6.3.4)$$

Example 6.3.2. Let $n = 6$ and $w = a_5 a_4 a_3 a_2 a_1 a_0 = 011011$. In this case, $S = \{(0, 1), (3, 4)\}$. Hence, by Definition 6.3.1, we have $f_0 = 1 + 1 = 2, f_1 = 1, f_2 = 2, f_3 = 4 + 1 = 5, f_4 = 4, f_5 = 5$, and $f_6 = 6$. Thus, $f = 6545212$. Since $f_0 = 2$, thus we flip the bit $a_2 a_1 a_0$ to generate the next word in L_6^2 , which is $w' = 011100$.

Now, we compute the focus pointer f corresponding to $w' = a_5 a_4 a_3 a_2 a_1 a_0 = 011100$. Since $S = \{(2, 4)\}$, by Equation 6.3.4, we have $f = 6543510$. With $f_0 = 0$, we flip a_0 to get the successor of w' , which is 011101 .

Algorithm 1 is the loopless algorithm to generate R_n^2 using focus pointers.

Algorithm 1 The classic loopless algorithm [7] for generating R_n^2 . Array R has length n while array F has length $n + 1$ and both use 0-based indices.

```

function looplessBinary( $n$ )
1:  $R \leftarrow [0, 0, \dots, 0]$                                 ▷ current codeword in  $R_n^2$ 
2:  $F \leftarrow [0, 1, \dots, n]$                                 ▷ focus pointers  $|F| = n + 1$ 
3: yield  $R$                                                     ▷ first codeword in  $R_n^2$ 
4: while  $F[0] < n$  do                                        ▷  $R[n]$  is invalid
5:    $j \leftarrow F[0]$                                           ▷ index of change
6:    $R[j] \leftarrow 1 - R[j]$                                     ▷ flip that bit
7:    $F[0] \leftarrow 0$                                           ▷ reset focus pointer
8:    $F[j] \leftarrow F[j + 1]$                                     ▷ inherit focus pointer
9:    $F[j + 1] \leftarrow j + 1$                                     ▷ reset focus pointer
10: yield  $R$                                                   ▷ next codeword in  $R_n^2$ 
    
```

We argue the correctness of Algorithm 1, with the understanding that at iteration $r = \text{Rank}^b(w)$, with w in L_n^2 , the index of the bit that changes in R is the largest index that

changes when incrementing w . That is precisely j , the length of the largest suffix of 1's in the binary representation of w . For instance, as discussed in Example 6.3.2, the next iteration after incrementing $w = 011011$, we get $w' = 011100$. In R_n^2 , we must flip the bit with index $j = 2$. The focus pointers are keeping track of the positions of the maximal substrings of 1's in w . So, $S_w = \{(0, 1), (3, 4)\}$ and $f_w = 6545212$. A position i in f_w that contains value $j \neq i$, indicates $(i, j - 1) \in S_w$. Thus, $(f_w)_0 = 2$ and $(f_w)_3 = 5$ indicate $S_w = \{(0, 1), (3, 4)\}$. Because $(f_w)_0 = 2$, we know $a_2a_1a_0$ will change in w from 011 to 100 and this means the corresponding codeword in R_n^2 must flip the digit with index $(f_w)_0 = 2$. That explains the updates in lines 5 and 6 of the algorithm:

$$\begin{aligned} j &\leftarrow F[0] \\ R[j] &\leftarrow 1 - R[j] \end{aligned}$$

We now argue that the focus pointers are updated correctly. As the suffix $a_ja_{j-1}\dots a_0$ changed from $01\dots 1$ in w to $10\dots 0$ in w' ; the last suffix of 1's starting at position 0 was destroyed, so we must do this update in line 7:

$$F[0] \leftarrow 0$$

This corresponds to removing $(0, 1)$ from S_w in Example 6.3.2. However, a new substring of 1's now starts at position j . Where it ends, depends on the value of $F[j+1]$. If $F[j+1] = j+1$, it indicates $a_{j+1} = 0$ and (j, j) must be added to S_w . If $F[j+1] = z \neq j+1$, it indicates the next bit equals to 0 is $a_z = 0$ and $(j, z-1)$ must be added to S_w and $(j+1, z-1)$ must be removed. In both cases, this can be accomplished with the updates in lines 8 and 9:

$$\begin{aligned} F[j] &\leftarrow F[j+1] \\ F[j+1] &\leftarrow j+1 \end{aligned}$$

In our example, since $j = 2$ and $(f_{w'})_2 = 5$, we have $S_{w'} = S_w \setminus \{(0, 1), (3, 4)\} \cup \{(2, 4)\}$, which corresponds to $w' = 011100$. Line 10 uses the command **yield R**, which indicates the current codeword R is "visited" at this point.

Example 6.3.3. We provide a simulation of Algorithm 1 for $n = 3$ in Figure 6.1. Note that the array positions are displayed in their natural order, opposite to their corresponding positions in words f and r .

6.3.2 Loopless algorithm for mixed-radix Gray codes

Now, we discuss the loopless generation of reflected and modular Gray codes over a mixed-radix base, b . Throughout the remainder of this chapter, unlike in the previous two Sections 6.1 and 6.2, we write $b = (b_n, \dots, b_1)$ a mixed-radix base, or $b = (b_{n-1}, \dots, b_0)$ when using 0-based indexing. The concept of focus pointers is generalized in the following way. While in the binary case, focus pointers keep track of substrings of 1's in the binary representation of w in L_n^2 , in the mixed-radix case, focus pointers keep track of substrings of digits i attaining their maximum values $b_i - 1$ in the mixed-radix representation of w in L_n^b . The definition of the focus pointers is given next.

iteration# (rank)	F	j = F[0]	R	$w=a_2a_1a_0$ in L_n^2	$f = f_3f_2f_1f_0$	$z=r_2r_1r_0$ in R_n^2
0	[0, 1, 2, 3]	—	[0, 0, 0]	000	3210	000
1	[1, 1, 2, 3]	0	[1, 0, 0]	001	3211	001
2	[0, 2, 2, 3]	1	[1, 1, 0]	010	3220	011
3	[2, 1, 2, 3]	0	[0, 1, 0]	011	3212	010
4	[0, 1, 3, 3]	2	[0, 1, 1]	100	3310	110
5	[1, 1, 3, 3]	0	[1, 1, 1]	101	3311	111
6	[0, 3, 2, 3]	1	[1, 0, 1]	110	3230	101
7	[3, 1, 2, 3]	0	[0, 0, 1]	111	3213	100

Figure 6.1: A simulation of Algorithm 1 for $n = 3$ that generates R corresponding to z .

Definition 6.3.4. Let $w = a_{n-1} \dots a_1 a_0$ be an n -tuple in L_n^b where $b = (b_{n-1}, \dots, b_1, b_0)$. Let S be the set of pair of indices (i, j) where $a_i a_{i+1} \dots a_j$ is a maximal substring of $b_l - 1$ in w with $i \leq l \leq j$. That is, $S = \{(i, j) : a_l = b_l - 1, i \leq l \leq j \text{ and } (a_{i-1} \neq b_{i-1} - 1 \text{ or } i = 0) \text{ and } (a_{j+1} \neq b_{j+1} - 1 \text{ or } j = n - 1)\}$. Then, the focus pointer $f = f_n f_{n-1} \dots f_1 f_0$ is

$$f_i = \begin{cases} j + 1, & \text{if } (i, j) \in S, \\ i, & \text{otherwise.} \end{cases} \quad (6.3.5)$$

Example 6.3.5. $b = (6, 2, 7, 5, 4, 3)$ and $w = 40521\bar{1}$, the rightmost digit with a bar over it is given the highest priority to increment. Since each digit is strictly less than the maximum allowed digit corresponding to the base, we have $f = 6543210$.

Example 6.3.6. $b = (6, 2, 7, 5, 4, 3)$ and $w = 403432$. Since the digits a_0, a_1, a_2 are at their maximum, the digit a_3 is the top priority to increment. Thus, $f = 6543213$.

Example 6.3.7. $b = (2, 3, 6, 2, 5, 4, 3, 4)$ and $w = 10014123$. Since the digits a_0, a_1, a_3, a_4, a_7 are the maximum corresponding to their bases, $f_0 = 2$ as the digit a_2 has the highest priority to increment. Furthermore, $f_3 = 5$ since the digits a_3 and a_4 are at their maximum while the digit a_5 is not. Similarly, $f_7 = 8$. Thus, $F = 886545212$.

Examples 6.3.5–6.3.7 are summarized below.

$$b = (6, 2, 7, 5, 4, 3) \quad w = 405211 \quad f = 6543210 \quad (6.3.6)$$

$$b = (6, 2, 7, 5, 4, 3) \quad w = 403432 \quad f = 6543213 \quad (6.3.7)$$

$$b = (2, 3, 6, 2, 5, 4, 3, 4) \quad w = 10014123 \quad f = 886545212 \quad (6.3.8)$$

While in the binary case, focus pointers have 1-to-1 correspondence with words in L_n^2 , the same is not true for the mixed-radix case. The reason is because a digit is now incremented $b_i - 2$ times until it reaches its maximum value $b_i - 1$.

For the reflected Gray code, we use R and an array D , storing current directions for each digit in R , in order to detect when a bit reached its “limit” (maximum value) and

to trigger the update of the focus pointers. Basically, $R[j]$ reached its **limit** if $(D[j] = +1$ and $R[j] = b_j - 1)$ or $(D[j] = -1$ and $R[j] = 0)$. After this, direction $D[j]$ is updated to the opposite direction. Algorithm 2 shows this modified algorithm; Knuth [43] gives this algorithm (Algorithm H in *The Art of Computer Programming 4a* - Page 300).

Algorithm 2 The classic loopless algorithm for generating R_n^B . Arrays have length n (except for F of length $n + 1$) and use 0-based indices.

```

function looplessReflected(B)
1:  $R \leftarrow [0, 0, \dots, 0]$                                 ▷ current codeword in  $R_n^B$ 
2:  $D \leftarrow [1, 1, \dots, 1]$                                 ▷ directions for  $R$ 
3:  $F \leftarrow [0, 1, \dots, n]$                                 ▷ focus pointers  $|F| = n + 1$ 
4: yield  $R$                                                     ▷ first codeword in  $R_n^B$ 
5: while  $F[0] < n$  do                                          ▷  $R[n]$  is invalid
6:    $j \leftarrow F[0]$                                           ▷ index of change
7:    $b \leftarrow B[j]$                                           ▷ base of change
8:    $F[0] \leftarrow 0$                                           ▷ reset focus pointer
9:    $R[j] \leftarrow R[j] + D[j]$                                 ▷ change digit by  $\pm 1$ 
10:   $\text{limit} \leftarrow R[j] = 0$  or  $R[j] = b - 1$               ▷ test
11:  if  $\text{limit}$  then                                           ▷  $R[j]$  at limit
12:     $D[j] \leftarrow -D[j]$                                     ▷ direction for  $R[j]$ 
13:     $F[j] \leftarrow F[j + 1]$                                 ▷ inherit focus pointer
14:     $F[j + 1] \leftarrow j + 1$                                 ▷ reset focus pointer
15:  yield  $R$                                                   ▷ next codeword in  $R_n^B$ 
    
```

For the modular Gray code M_n^b , we use M and an array of targets T to detect when a digit achieved its **limit**. The targets are initialized with maximum bit values, that is, $T = [b_0 - 1, b_1 - 1, \dots, b_{n-1} - 1]$. However, the targets will change dynamically. Basically, when a focus pointer is updated for a digit j , $M[j]$ had just changed and reached the value of the target $T[j]$, so the target must be decremented modulo b_j , to make sure that $M[j]$ only changes again after it has been incremented $b_j - 1$ times. These ideas are reflected in Algorithm 3. This algorithm is given as Exercise 77 in Section 7.2.1.1 of *The Art of Computer Programming 4a* [43].

Example 6.3.8. We provide a simulation of Algorithms 2 and 3 for $b = (2, 3, 3)$. Let F_i, R_i, D_i, M_i , and T_i be the corresponding arrays in i^{th} iteration for $0 \leq i \leq 17$. The algorithms start with the codewords $R = M = 000$ and $F = [0, 1, 2, 3]$. The corresponding directions D , and targets T , in the respective algorithms are initialized as $[1, 1, 1]$ and $[2, 2, 1]$, respectively. Let us consider the iteration # 8, where $f = 3212$. Thus, in iteration # 9 of Algorithm 2, $j = 2$. Since the corresponding $D_8[j] = 1$, then $R_9[j] = R_8 + D_8[j] = 0 + 1 = 1$. Thus, $R_9 = 221$ and corresponding codeword in R_3^b is 122. Similarly, in iteration # 9 of Algorithm 3, we also have $j = 2$. Since the corresponding $T_8[j] = 1$, then $M_9[j] = (M_8 + 1) \bmod 2 = 0 + 1 = 1$. Thus, $M_9 = 021$ and corresponding codeword in R_3^b is 120. Since in both cases, R and M hit their respective **limit** as provided in Algorithms 2 and 3, respectively, F is subsequently updated as in Equation 6.3.5.

#	R	R_3^b	D	$j = f_0$	F	$f = f_3 f_2 f_1 f_0$	L_3^b
0	000	000	+++		0123	3210	000
1	100	001	+++	0	0123	3210	001
2	200	002	-++	0	1123	3211	002
3	210	012	-++	1	0123	3210	010
4	110	011	-++	0	0123	3210	011
5	010	010	+++	0	1123	3211	012
6	020	020	+ - +	1	0223	3220	020
7	120	021	+ - +	0	0223	3220	021
8	220	022	--+	0	2123	3212	022
9	221	122	---	2	0133	3310	100
10	121	121	---	0	0133	3310	101
11	021	120	+ --	0	1133	3311	102
12	011	110	+ --	1	0133	3310	110
13	111	111	+ --	0	0133	3310	111
14	211	112	---	0	1133	3311	112
15	201	102	- + -	1	0323	3230	120
16	101	101	- + -	0	0323	3230	121
17	001	100	++-	0	3123	3213	122

 Table 6.4: A simulation of Algorithm 2 for $b = (2, 3, 3)$

#	M	M_3^b	T	$j = f_0$	F	$f = f_3 f_2 f_1 f_0$	L_3^b
0	000	000	221		0123	3210	000
1	100	001	221	0	0123	3210	001
2	200	002	121	0	1123	3211	002
3	210	012	121	1	0123	3210	010
4	010	010	121	0	0123	3210	011
5	110	011	021	0	1123	3211	012
6	120	021	011	1	0223	3220	020
7	220	022	011	0	0223	3220	021
8	020	020	211	0	2123	3212	022
9	021	120	210	2	0133	3310	100
10	121	121	210	0	0133	3310	101
11	221	122	110	0	1133	3311	102
12	201	102	110	1	0133	3310	110
13	001	100	110	0	0133	3310	111
14	101	101	010	0	1133	3311	112
15	111	111	000	1	0323	3230	120
16	211	112	000	0	0323	3230	121
17	011	110	200	0	3123	3213	122

 Table 6.5: A simulation of Algorithm 3 for $b = (2, 3, 3)$

Algorithm 3 The classic loopless algorithm for generating M_n^B . Arrays have length n (except for F of length $n + 1$) and use 0-based indices.

```

function looplessModular(B)
1:  $M \leftarrow [0, 0, \dots, 0]$                                  $\triangleright$  current codeword in  $M_n^B$ 
2:  $T \leftarrow [b_0-1, b_1-1, \dots, b_{n-1}-1]$                  $\triangleright$  targets for  $M$ 
3:  $F \leftarrow [0, 1, \dots, n]$                                  $\triangleright$  focus pointers  $|F| = n+1$ 
4: yield  $M$                                                      $\triangleright$  first codeword in  $M_n^B$ 
5: while  $F[0] < n$  do                                         $\triangleright$   $M[n]$  is invalid
6:    $j \leftarrow F[0]$                                            $\triangleright$  index of change
7:    $b \leftarrow B[j]$                                            $\triangleright$  base of change
8:    $F[0] \leftarrow 0$                                            $\triangleright$  reset focus pointer
9:    $M[j] \leftarrow (M[j] + 1) \bmod b$                              $\triangleright$  cyclic inc
10:   $limit \leftarrow M[j] = T[j]$                                  $\triangleright$  equivalent test
11:  if  $limit$  then                                            $\triangleright$   $M[j]$  at limit
12:     $T[j] \leftarrow (T[j]-1) \bmod b$                              $\triangleright$   $M[j]$  target
13:     $F[j] \leftarrow F[j + 1]$                                  $\triangleright$  inherit focus pointer
14:     $F[j + 1] \leftarrow j + 1$                                  $\triangleright$  reset focus pointer
15: yield  $M$                                                      $\triangleright$  next codeword in  $M_n^B$ 
    
```

From our previous explanation and Example 6.3.8, we thereafter observe that, for a given base b , the focus pointer F is identical at every step in Algorithms 2 and 3 and this further implies that the change sequence j also remains the same. Such j also provides the change sequence of L_3^b , which is the mixed-radix representation of the iteration number. For instance, see Example 6.3.11.

Now, we define the ruler sequence $\text{Ruler}(b)$, for a mixed-radix base, b . It is defined recursively for bases $b = (b_n, b_{n-1}, \dots, b_2, b_1)$ where $b' = (b_{n-1}, \dots, b_1)$ if $n > 1$.

Definition 6.3.9.

$$\text{Ruler}(b) = \begin{cases} \overbrace{1, 1, \dots, 1}^{b_1-1 \text{ copies}} & \text{if } n = 1 \\ \underbrace{\text{Ruler}(b'), n, \text{Ruler}(b'), \dots, n, \text{Ruler}(b')}_{b_n-1 \text{ copies of } n} & \text{if } n > 1 \end{cases} \quad (6.3.9)$$

Example 6.3.10. $\text{Ruler}(b) = \text{Ruler}(b'), n, \text{Ruler}(b'), n, \text{Ruler}(b') = 1, 2, 1, 2, 1$ for $b = (3, 2)$ and $b' = (2)$.

Note that the mixed-radix ruler sequence $\text{Ruler}(b)$ has $b! - 1$ numbers. In Example 6.3.11, we show that $\text{Ruler}(b)$ provides the index that changes in R_3^b and M_3^b . Furthermore, $\text{Ruler}(b)$ gives the number of digits that change in L_3^b . To generalize this further, let $w = w_n w_{n-1} \dots w_1 \in \mathbb{W}(b)$ for any mixed-radix base, $b = (b_n, b_{n-1}, \dots, b_1)$. Then, the next codeword in L_n^b can potentially be created by incrementing w_i for some $i \in [1, n]$ and setting $w_{i-1} \dots w_1$, for $i > 1$, to 0 whenever they are at their maximum values corresponding to their base b_i . Equivalently, the digits $w_i w_{i-1} \dots w_1$ are cyclically incremented modulo their base b_i .

for some $i \in [1, n]$. These observations further imply that L_n^b is indeed generated starting from 0^n by greedily cyclically incrementing the shortest suffix allowed of the codeword $a_n \dots a_1$ at each iteration r , and the suffix length is determined by $\text{Ruler}(b)$; specifically, it is precisely the value at the r^{th} position of $\text{Ruler}(b)$. Example 6.3.11 clarifies this discussion.

Example 6.3.11. Let $b = (2, 2, 3)$. We generate L_3^b , R_3^b , and M_3^b where overlines and underlines indicate how the digits change. Note that underlines are especially used in R_3^b where overlines and underlines correspond to increments and decrements, respectively. For every codeword w , we describe it using the index order as in $w_3w_2w_1$.

$$L_3^b = 00\bar{0}, 00\bar{1}, 00\bar{2}, 01\bar{0}, 01\bar{1}, \bar{0}1\bar{2}, 10\bar{0}, 10\bar{1}, \bar{1}0\bar{2}, 11\bar{0}, 11\bar{1}, 11\bar{2} \quad (6.3.10)$$

$$R_3^b = 00\bar{0}, 00\bar{1}, 00\bar{2}, 01\bar{2}, 01\bar{1}, \bar{0}1\bar{0}, 11\bar{0}, 11\bar{1}, 1\bar{1}2, 10\bar{2}, 10\bar{1}, 100 \quad (6.3.11)$$

$$M_3^b = 00\bar{0}, 00\bar{1}, 00\bar{2}, 01\bar{2}, 01\bar{0}, \bar{0}1\bar{1}, 11\bar{1}, 11\bar{2}, \bar{1}1\bar{0}, 10\bar{0}, 10\bar{1}, 102 \quad (6.3.12)$$

$$\text{Ruler}(b) = \quad 1, \quad 1, \quad 2, \quad 1, \quad 1, 3, \quad 1, \quad 1, \quad 2, \quad 1, \quad 1 \quad (6.3.13)$$

The main conclusion of this section (also observed in [51]), is that for a base $b = (b_n, \dots, b_1)$; the ruler sequence $\text{Ruler}(b)$, rules the digit that changes in the orders L_n^b , R_n^b , and M_n^b . In addition, focus pointers can be used to generate the ruler sequence. Moreover, R_n^b and M_n^b can be generated looplessly.

6.4 Ranking and unranking algorithms

In the context of lexicographic orders and Gray codes, the *rank* of an object is its position in the order. Similarly, the *unrank* of a position is the object in that position. To start Algorithm 1, 2, or 3 at its r^{th} iteration, we can set its variables to their respective values at this iteration. That is, we can unrank the algorithm's internal state. Similarly, if Algorithm 1, 2, or 3 unexpectedly terminates and we only have the most recently generated word in R_n^b or M_n^b , then we can first rank the word relative to R_n^b or M_n^b and resume the algorithm starting from that rank. The same concept also applies to an algorithm for generating L_n^b . We provide $\mathcal{O}(n)$ -time Python code for these tasks in [57].

In this section, we derive $\mathcal{O}(n)$ -time ranking and unranking algorithms for mixed-radix codewords in L_n^b , R_n^b , and M_n^b . Then, we unrank the data structures from the classic loopless Algorithms 1, 2, and 3 in $\mathcal{O}(n)$ -time. While similar results for mixed-radix codewords appear in the literature [43], we believe that the data structure results have not been previously published. The most difficult and novel of these problems is unranking the focus pointers. The work in this section supports efficient resumption and the loopless generation of our ‘modified’ modular Gray codes discussed in Section 6.5.

Throughout this section, we use 0-based indexing to match the indexing of most programming languages. More specifically, we index and denote bases and individual codewords as $b = (b_{n-1}, \dots, b_1, b_0)$ and $a = a_{n-1} \dots a_1 a_0$, respectively. We denote by m_i the value $b_i - 1$ for $0 \leq i \leq n - 1$. In our pseudocode, the modular operation is denoted `mod`. We also refer to the *position* i in an order of codewords as the sequence of digits in the i^{th} index of the codewords in that order.

6.4.1 Ranking and unranking mixed-radix codewords

Here we efficiently rank and unrank mixed-radix codewords in the lexicographic order, the reflected Gray code, and the modular Gray code. Note that these results are relative to the three individual orders, so there are a total of $2 \cdot 3 = 6$ distinct algorithms. For a base $(b_{n-1}, \dots, b_1, b_0)$, we define $p(i)$ as $p(i) = 1$ if $i = 0$ and $p(i) = \prod_{j=0}^{i-1} b_j$ if $1 \leq i \leq n$. In other words, $p(i)$ is the product of the i least significant base values where the entire product is $p(n) = b!$.

First, we consider the ranking and unranking algorithms for the mixed-radix codewords in the lexicographic order, L_n^b . Consider $b = (2, 3, 3)$ and L_3^b as shown in Table 6.4. The middle digit (that is, position 1) of L repeats the pattern 0, 0, 0, 1, 1, 1, 2, 2, 2 twice. We denote this $(0^3, 1^3, 2^3)^2$ or simply $(0^3 1^3 2^3)^2$ where exponents denote repetition. Then in the order of L_3^b , at any i^{th} position, the following sequence $(0^{p(i)} 1^{p(i)} \dots m_i^{p(i)})^{\frac{b!}{p(i+1)}}$ occurs. This provides the basis for the ranking and unranking algorithms of L_3^b provided in Algorithm 4.

Algorithm 4 Unranking and ranking algorithms of codewords in L_n^B . The arrays B and a have length n . For base B , codeword a , and its corresponding rank r , **rankLex**(B, a) returns its rank in L_n^B while **unrankLex**(B, r) returns the corresponding word in the order.

function unrankLex (B, r)	function rankLex (B, a)
1: $a = [0, 0, \dots, 0]$	1: $p \leftarrow [1, B[0], \dots, B[0] \cdot B[1] \cdots B[n-2]]$
2: $p \leftarrow [1, B[0], \dots, B[0] \cdot B[1] \cdots B[n-2]]$	2: $r \leftarrow 0$ ▷ Initial rank
3: for $i \leftarrow n-1, \dots, 1, 0$ do	3: for $i \leftarrow n-1, \dots, 1, 0$ do
4: $b = \lfloor \frac{r}{p[i]} \rfloor$	4: $r \leftarrow r + p[i] \cdot a[i]$ ▷ Update r
5: $a[i] = b \bmod B[i]$	5: return r
6: return a	

Example 6.4.1. Let $b = (5, 4, 5)$ and $a = a_2 a_1 a_0 = 332$. We compute the rank r of a in L_3^b . Since $a_2 = 3$ the 3 must appear in position 2 in L_3^b , starting from rank $60 = (b_0 \cdot b_1) \cdot a_2$, after the sequence $0^{20} 1^{20} 2^{20}$. Hence, $r \geq 60$. At rank 60, position 1 must have the digit 0 with 3 being the previous element in that sequence as $60 \bmod 4 = 0$. Since a_1 is 3, beginning at rank 60 it appears first after the sequence $0^5 1^5 2^5$. This implies $r \geq 60 + (b_0 \cdot a_3) = 60 + (5 \cdot 3) = 75$. Now, at rank 75, the digit corresponding to position 0 in L_3^b must be 0 as $75 \bmod 5 = 0$. Thus, $r = 75 + (1 \cdot a_0) = 75 + (1 \cdot 2) = 77$.

Example 6.4.2. Given $b = (5, 4, 5)$ and rank 77, we determine the corresponding word $a = a_0 a_1 a_2$ in L_3^b . First, since $\lfloor \frac{77}{20} \rfloor = 3$, it means that a_2 is 3 as it appears in position 2 in L_3^b after $0^{20} 1^{20} 2^{20}$, starting at rank 60. Since $\lfloor \frac{77}{5} \rfloor = 15$ and $15 \bmod 4 = 3$, the digit in position 1 in L_3^b at ranks from 75 to 79 is 3. Thus, $a_1 = 3$. Since the position 0 has digits in the sequence $012012012 \dots$ and $77 \bmod 5 = 2$, it implies that $a_0 = 2$. Thus, $a = a_2 a_1 a_0 = 332$.

The idea behind the ranking and unranking algorithms of reflected and modular Gray codes are slightly different. In the order of R_n^b , at position i , the values alternately go up and down, that is,

$$(0^{p(i)} 1^{p(i)} \dots m_i^{p(i)} m_i^{p(i)} \dots 1^{p(i)} 0^{p(i)})^{\lfloor \frac{b!}{2p(i+1)} \rfloor}.$$

However, in case of M_n^b , the values cyclically wrap around modulo b_i . For example, let $b = 3, 3, 2$. Observing the generation of M_n^b in Table 6.5, we notice the following sequence of digits. At position 2, the sequence is $0^9 1^9$. At position 1, it is $0^3 1^3 2^3 2^3 0^3 1^3$ while at position 0, it is $(012201120)^2$. We provide the corresponding ranking and unranking algorithms in Algorithms 5 and 6.

Example 6.4.3. Let $b = (b_2, b_1, b_0) = (5, 3, 4)$ and $a = a_2 a_1 a_0 = 413$. We compute its rank r in R_3^b . We have $\mathbf{p} = [p(0), p(1), p(2)] = [1, 4, 12]$. Since $a_2 = 4$, the 4 must appear in position 2 in R_3^b , starting from rank 48, after the sequence $0^{12} 1^{12} 2^{12} 3^{12}$. Thus, $r \geq 48$. Since $\lfloor \frac{48}{4} \rfloor = 12$, it implies that in position 1 in R_3^b , the sequence $0^4 1^4 2^4 2^4 1^4 0^4 0^4 1^4 2^4 2^4 1^4 0^4$ appears before rank 48. Note that it is also concluded from the fact that $\lfloor \frac{12}{3} \rfloor = 4$ is even and since the sequences $0^4 1^4 2^4$ (initially covering ranks 0 to 11, then 24 to 35 and so on) and $2^4 1^4 0^4$ (initially covering ranks 12 to 23, then 36 to 47, and so on) appear alternately, the sequence $0^4 1^4 2^4$ appears starting at rank 48. Since $a_1 = 1$, $r \geq 48 + 4 \cdot 1 = 52$. Since position 0 in R_3^b has sequences 0123 and 3210 appearing alternately and $\lfloor \frac{52}{4} \rfloor = 13$ is odd, it is clear that rank 52 has digit 3 in this position. Since $a_0 = 3$, $r = 52 + 1 \cdot (3 - 3) = 52 + 1 \cdot 0 = 52$.

Example 6.4.4. Let $b = (b_2, b_1, b_0) = (5, 3, 4)$ and $r = 52$. We determine the corresponding word in R_3^b . Note $\mathbf{p} = [p(0), p(1), p(2)] = [1, 4, 12]$. Since $\lfloor \frac{52}{12} \rfloor = 4$ and in position 2 in R_3^b the sequence $0^{12} 1^{12} 2^{12} 3^{12}$ appears covering ranks 0 to 47, $a_2 = 4$. Again, we have $\mathbf{z} = \lfloor \frac{r}{p(1)} \rfloor = \lfloor \frac{52}{4} \rfloor = 13$. Further calculating $\mathbf{k} = \lfloor \frac{\mathbf{z}}{b_1} \rfloor = \lfloor \frac{13}{3} \rfloor = 4$, which is even and observing that the position 1 in R_3^b has sequence $0^4 1^4 2^4$ and $2^4 1^4 0^4$ alternately, tells us that the digit corresponding to rank 52 in position 1 must be within the sequence $0^4 1^4 2^4$. Since $1 = \mathbf{z} \bmod b_1 = 13 \bmod 3 = 1$, we further conclude that the digit is within the subsequence 1^4 , thus, $a_1 = 1$. Since $\mathbf{z} = \lfloor \frac{r}{p(0)} \rfloor = \lfloor \frac{52}{1} \rfloor = 52$ and $\mathbf{k} = \lfloor \frac{\mathbf{z}}{b_0} \rfloor = \lfloor \frac{52}{4} \rfloor = 13$ is odd, the digit in position 0 in rank 52 must be in the sequence 3210. Since $\mathbf{z} \bmod b_0 = 52 \bmod 4 = 0$, we have $a_0 = (4 - 1 - 0) \bmod 4 = 3$. Thus, $a = a_2 a_1 a_0 = 413$.

Algorithm 5 Unranking and ranking algorithms of codewords in R_n^B . The arrays $\mathbf{B}$ and $\mathbf{a}$ have length n . For base $\mathbf{B}$, word $\mathbf{a}$, and its corresponding rank r , `rankReflected( $\mathbf{B}$ ,  $\mathbf{a}$ )` returns its rank in R_n^B , while `unrankReflected( $\mathbf{B}$ ,  $r$ )` returns the corresponding word in the order.

function unrankReflected($\mathbf{B}$, r)	function rankReflected($\mathbf{B}$, $\mathbf{a}$)
1: $\mathbf{a} = [0, 0, \dots, 0]$	1: $\mathbf{p} \leftarrow [1, \mathbf{B}[0], \dots, \mathbf{B}[0] \cdot \mathbf{B}[1] \cdots \mathbf{B}[n-2]]$
2: $\mathbf{p} \leftarrow [1, \mathbf{B}[0], \dots, \mathbf{B}[0] \cdot \mathbf{B}[1] \cdots \mathbf{B}[n-2]]$	2: $r \leftarrow 0$
3: for $i \leftarrow n-1, \dots, 1, 0$ do	3: for $i \leftarrow n-1, \dots, 1, 0$ do
4: $\mathbf{z} \leftarrow \lfloor \frac{r}{\mathbf{p}[i]} \rfloor$	4: $\mathbf{z} \leftarrow \lfloor \frac{r}{\mathbf{p}[i]} \rfloor$
5: $\mathbf{l} \leftarrow \mathbf{z} \bmod \mathbf{B}[i]$	5: $\mathbf{k} \leftarrow \lfloor \frac{\mathbf{z}}{\mathbf{B}[i]} \rfloor$
6: $\mathbf{k} \leftarrow \lfloor \frac{\mathbf{z}}{\mathbf{B}[i]} \rfloor$	6: if $\mathbf{k} \bmod 2 = 0$ then
7: if $\mathbf{k} \bmod 2 = 0$ then	7: $r \leftarrow r + \mathbf{p}[i] \cdot \mathbf{a}[i]$
8: $\mathbf{a}[i] \leftarrow 1 \bmod \mathbf{B}[i]$	8: else
9: else	9: $r \leftarrow r + \mathbf{p}[i] \cdot (\mathbf{B}[i] - 1 - \mathbf{a}[i])$
10: $\mathbf{a}[i] \leftarrow (\mathbf{B}[i] - 1 - 1) \bmod \mathbf{B}[i]$	10: return r
11: return $\mathbf{a}$	

Example 6.4.5. Let $b = (b_2, b_1, b_0) = (5, 3, 4)$ and $a = a_2 a_1 a_0 = 413$. We compute the rank r of a in M_3^b . Note $\mathbf{p} = [p(0), p(1), p(2)] = [1, 4, 12]$. We compute its rank r in M_3^b . Since $a_2 = 4$, the 4 must appear in position 2 in M_3^b , starting from rank 48, after the sequence $0^{12} 1^{12} 2^{12} 3^{12}$. Thus, $r \geq 48$. Since $\lfloor \frac{48}{4} \rfloor = 12$, in position 1 in M_3^b , the sequence $0^4 1^4 2^4 2^4 0^4 1^4 1^4 2^4 0^4 0^4 1^4 2^4$ appears before rank 48. Since $\mathbf{k} = \lfloor \frac{12}{3} \rfloor = 4$, a_1 appears after the sequence $0^4 1^4 2^4 2^4 0^4 1^4 1^4 2^4 0^4 0^4 1^4 2^4$, that is, in the sequence $2^4 0^4 1^4$. Since in position 1, $0^4 1^4 2^4$, $2^4 0^4 1^4$, $1^4 2^4 0^4$ keep rotating, for $\mathbf{k} \bmod 3 = 1$, a_1 appears in the sequence starting with i^4 such that $i = (b_1 - \mathbf{k}) \bmod b_1 = 2$. In this particular subsequence, 1^4 appears after $2^4 0^4$, therefore $r \geq 48 + 8 = 56$. (the reader can check that $8 = 4 \cdot [(a_1 - (b_1 - k)) \bmod b_1]$). In position 0 of M_3^b , the sequences $0123, 3012, 2301, 1230$ appear cyclically. Since $\mathbf{k} = \lfloor \frac{56}{4} \rfloor = 14$, it means that the digit $a_0 = 3$ appears after the sequence $(01230123011230)^3 0123012$. Furthermore, $14 \bmod 4 = 2$, which confirms that the sequence we are looking at is 2301 . Thus, $r = 56 + 1 = 57$.

Example 6.4.6. Let $b = (b_2, b_1, b_0) = (5, 3, 4)$ and $r = 57$. Since $\lfloor \frac{57}{3 \cdot 4} \rfloor = \lfloor \frac{57}{12} \rfloor = 4$, we conclude $a_2 = (b_2 - 0 + 4) \bmod b_2 = 4$. Now, $\lfloor \frac{57}{4} \rfloor = 14$ and $\mathbf{k} = \lfloor \frac{14}{3} \rfloor = 4$. The digit a_1 corresponding to the rank 57 appears in the sequence $i^4(i+1)^4(i+2)^4$ for some $i \in \{0, 1, 2\}$. Furthermore, since $1 = 14 \bmod 3 = 2$, it must appear in the last subsequence $(i+2)^4$ of the sequence $i^4(i+1)^4(i+2)^4$. In position 1 of M_3^b , the sequences $0^4 1^4 2^4, 2^4 0^4 1^4, 1^4 2^4 0^4$ appear cyclically. Noticing this pattern, it is easy to see that since $\mathbf{k} \bmod 3 = 1$, $i = b_1 - 1 = 2$ (or $(b_1 - \mathbf{k}) \bmod b_1$). Thus, $a_1 = (b_1 - \mathbf{k} + 1) \bmod b_1 = 1$. We further proceed to find a_0 . We have $\mathbf{k} = \lfloor \frac{57}{4} \rfloor = 14$. Assume the digit a_0 corresponding to rank 57 appears in the sequence $i(i+1)(i+2)(i+3)$ for some $i \in \{0, 1, 2, 3\}$. Moreover, since $1 = 57 \bmod 4 = 1$, the digit $a_0 = i + 1$. Note that in position 1 of M_3^b , the sequences $0123, 3012, 2301, 1230$ rotate cyclically. Since $\mathbf{k} \bmod 4 = 2$, we know $i = (b_0 - \mathbf{k}) \bmod b_0 = 2$. Then, $a_0 = 2 + 1 = 3$. So, we have $a = 314$.

Algorithm 6 Unranking and ranking algorithms of codewords in M_n^B . The array B and a has length n . For base B , word a , and its corresponding rank r , `rankModular(B, a)` returns its rank in M_n^B while `unrankModular(B, r)` returns the corresponding word in the order.

function `unrankModular(B, r)`

```

1:  $\mathbf{a} = [0, 0, \dots, 0]$ 
2:  $\mathbf{p} \leftarrow [1, B[0], \dots, B[0] \cdot B[1] \cdots B[n-2]]$ 
3: for  $i \leftarrow n-1, \dots, 1, 0$  do
4:    $\mathbf{z} \leftarrow \lfloor \frac{\mathbf{r}}{\mathbf{p}[i]} \rfloor$ 
5:    $\mathbf{l} \leftarrow \mathbf{z} \bmod B[i]$ 
6:    $\mathbf{k} \leftarrow \lfloor \frac{\mathbf{z}}{B[i]} \rfloor$ 
7:    $\mathbf{a}[i] \leftarrow (B[i] - \mathbf{k} + 1) \bmod B[i]$ 
8: return  $\mathbf{a}$ 
```

function `rankModular(B, a)`

```

1:  $\mathbf{p} \leftarrow [1, B[0], \dots, B[0] \cdot B[1] \cdots B[n-2]]$ 
2:  $\mathbf{r} \leftarrow 0$ 
3: for  $i \leftarrow n-1, \dots, 1, 0$  do
4:    $\mathbf{z} \leftarrow \lfloor \frac{\mathbf{r}}{\mathbf{p}[i]} \rfloor$ 
5:    $\mathbf{k} \leftarrow \lfloor \frac{\mathbf{z}}{B[i]} \rfloor$ 
6:    $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{p}[i] \cdot ((\mathbf{k} + \mathbf{a}[i]) \bmod B[i])$ 
7: return  $\mathbf{r}$ 
```

6.4.2 Unranking Data Structures

We give a brief summary of the data structures used in Algorithms 1, 2, and 3. In their loopless generation algorithm for R_n^2 , Bitner, Ehrlich, and Reingold [7] introduced an array of

focus pointers, denoted by F , to keep track of the digits to be flipped. Generalizing this idea, Knuth [43] gave loopless algorithms for reflected and modular Gray codes over a non-binary base b , using additional data structures called *directions* for the reflected case and *targets* for the modular case.

6.4.2.1 Focus pointers

In Algorithms 1, 2, and 3, F is initialized as an array with a sequence of integers in the range $[0, n]$. The first index of F , $F[0]$, in each step points to the digit to be flipped (in case of R_n^2) or altered in a “certain” way (in case of R_n^b and M_n^b for a non-binary base b). Since L_n^b shares the underlying data structures with R_n^b and M_n^b , as discussed in Examples 6.3.8 and 6.3.11), $F[0] = i$ means that the i^{th} digit of the word is to be changed while the first $i - 1$ digits are at their maximum corresponding to their bases.

Algorithm 7 follows the ideas explained in Examples 6.3.5, 6.3.6, and 6.3.7 and returns the focus pointer corresponding to the codeword of rank r in L_n^b . Let $w = w_{n-1} \dots w_0$ be the codeword in L_n^b such that $\text{Rank}^b(w) = r$. In the algorithm, we consider three arrays $\mathbf{b}$, $\mathbf{x}$, and $\mathbf{y}$. Array $\mathbf{b}$ stores the indices i of the word w whenever $w_i = b_i - 1$, its maximum value. Array $\mathbf{b}$ may have some sequences of consecutive indices. Furthermore, the default focus pointer F is set as $[0, 1, \dots, n - 1, n]$. Let S_w be the set of pair of indices (i, j) as defined in Definition 6.3.4. If $S_w = \{(i_1, j_1), (i_2, j_2), \dots, (i_k, j_k)\}$, then lines 6–8 computes $\mathbf{b} = [i_1, i_1 + 1, \dots, j_1, i_2, i_2 + 1, \dots, j_2, \dots, i_k, i_k + 1, \dots, j_k]$. Note that $|\mathbf{b}| = |S_w| = k$. If $S_w = \emptyset$, this implies $|\mathbf{b}| = 0$ and by definition, $F = [0, 1, \dots, n - 1, n]$, which explains lines 9–10 of the algorithm. Lines 12–19 compute $\mathbf{x}$ and $\mathbf{y}$. That is, $\mathbf{x} = [i_1, i_2, \dots, i_k]$ and $\mathbf{y} = [j_1 + 1, j_2 + 1, \dots, j_k + 1]$. Since by Definition 6.3.4, $F(i_l) = j_l + 1$ with $1 \leq l \leq k$, we have $\mathbf{y} = [F(i_1), F(i_2), \dots, F(i_k)]$, which explains lines 20–21 of the algorithm.

Revisiting Example 6.3.7, since $S_w = \{(0, 1), (3, 4), (7, 7)\}$, we have $\mathbf{b} = [0, 1, 3, 4, 7]$. Then, $\mathbf{x}$ stores the minimum value in each pair in S_w , that is, $\mathbf{x} = [0, 3, 7]$ while $\mathbf{y}$ stores 1 higher than the maximum value in each pair in S_w , that is, $\mathbf{y} = [2, 5, 8]$. Then, we update the default focus pointer F as $F[0] = 2$, $F[3] = 5$, and $F[7] = 8$.

6.4.2.2 Directions

At position i in R_n^b , the sequence $0^{p(i)} 1^{p(i)} \dots m_i^{p(i)}$ is followed by $m_i^{p(i)} \dots 1^{p(i)} 0^{p(i)}$. Intuitively, position i at direction D corresponding to the sequence $0^{p(i)} 1^{p(i)} \dots (m_i - 1)^{p(i)} m_i^{p(i)}$ in R_n^b is $+^{p(i)} +^{p(i)} \dots +^{p(i)} -^{p(i)}$ because the values in this position in R_n^b go up following the direction $0 \rightarrow 1 \rightarrow \dots \rightarrow m_i$. Moreover, $-^{p(i)}$ corresponding to the subsequence $m_i^{p(i)}$ allows the algorithm to start decrementing following the direction $m_i \rightarrow \dots \rightarrow 1 \rightarrow 0$. Similarly, D must have $-^{p(i)} \dots -^{p(i)} +^{p(i)}$ corresponding to the following sequence $m_i^{p(i)} \dots 1^{p(i)} 0^{p(i)}$ in R_n^b . Hence, position i in D starts with the following sequence $(+^{p(i)})^{m_i} (-^{p(i)})^{b_i} (+^{p(i)})^{b_i} (-^{p(i)})^{b_i} \dots$. Note that in `unrankDirections(B, r)` in Algorithm 8, the definition of $\mathbf{p}'$ also means that $\mathbf{p}'[i] = p(i)b_i = p(i + 1)$ if $0 \leq i \leq n$. For a rank r , if $r \geq m_i \cdot p(i)$ and we calculate $x = \lfloor \frac{r - (m_i \cdot p(i))}{\mathbf{p}'[i]} \rfloor$, it tells us whether the rank is covered within

Algorithm 7 Unranking the focus pointers F at iteration r of Algorithm 1, 2, or 3. Lists b , x , y are built based on runs of consecutive digits at their maximum values.

```

function unrankFocus( $B, r$ )                                ▷ Bases  $B$  rank  $r$ 
1:  $a \leftarrow \text{unrankLex}(B, r)$                                 ▷ Rank  $r$  word
2:  $F \leftarrow [0, 1, \dots, n]$                                 ▷ Default focus pointers
3:  $b \leftarrow []$                                               ▷ Indices in  $a$  at max value
4:  $x \leftarrow []$                                               ▷ Start indices of runs
5:  $y \leftarrow []$                                               ▷ Indices after each run
6: for  $i \leftarrow 0, 1, \dots, n-1$  do                            ▷ Check all indices
7:   if  $a[i] = B[i]-1$  then                                    ▷ Value at max?
8:      $\text{APPEND}(b, i)$                                           ▷ Add it index to list
9:   if  $|b| = 0$  then                                          ▷ No max values?
10:  return  $F$                                                   ▷ Return default values
11: else
12:    $z \leftarrow b[0]$                                           ▷ Start current run
13:   for  $j \leftarrow 1, 2, \dots, |b|-1$  do                    ▷ Indices at max
14:     if  $b[j] \neq b[j-1] + 1$  then                            ▷ End of run?
15:        $\text{APPEND}(x, z)$                                           ▷ Start of run
16:        $\text{APPEND}(y, b[j-1] + 1)$                                 ▷ End of run
17:        $z \leftarrow b[j]$                                       ▷ Start new run
18:    $\text{APPEND}(x, z)$                                           ▷ Start of last run
19:    $\text{APPEND}(y, b[|b|-1] + 1)$                                 ▷ End of last run
20: for  $i \leftarrow 0, 1, \dots, |x|-1$  do
21:    $F[x[i]] \leftarrow y[i]$                                 ▷ Update the focus pointer
22: return  $F$ 

```

the sequence $(-p(i))^{b_i}$ (if x is even) and $(+p(i))^{b_i}$ (if x is odd). It indeed follows from the fact that after the sequence $(+p(i))^{m_i}$ in R_n^b , $(-p(i))^{b_i}$ and $(+p(i))^{b_i}$ appear alternately.

6.4.2.3 Targets

Similarly, at position i in M_n^b , the sequence $0^{p(i)} 1^{p(i)} \dots m_i^{p(i)}$ is followed by $m_i^{p(i)} 0^{p(i)} 1^{p(i)} \dots (m_i - 1)^{p(i)}$. Clearly, position i at T corresponding to the sequence $0^{p(i)} 1^{p(i)} \dots (m_i - 1)^{p(i)}$ in M_n^b is $m_i^{p(i)} m_i^{p(i)} \dots m_i^{p(i)} (m_i - 1)^{p(i)}$ because the values in this position in M_n^b increment following the direction $0 \rightarrow 1 \rightarrow \dots \rightarrow m_i$. Additionally, $(m_i - 1)^{p(i)}$ in T corresponding to the subsequence $m_i^{p(i)}$ allows the algorithm to start incrementing cyclically following the direction $m_i \rightarrow 0 \rightarrow 1 \rightarrow \dots \rightarrow m_i - 1$. Thus, T must have $(m_i - 1)^{p(i)} (m_i - 1)^{p(i)} (m_i - 1)^{p(i)} \dots (m_i - 1)^{p(i)} (m_i - 2)^{p(i)}$ corresponding to the following sequence $m_i^{p(i)} 0^{p(i)} 1^{p(i)} \dots (m_i - 2)^{p(i)} (m_i - 1)^{p(i)}$ in M_n^b . Hence, position i in T starts with the following sequence $(m_i^{p(i)})^{m_i} ((m_i - 1)^{p(i)})^{b_i} ((m_i - 2)^{p(i)})^{b_i} \dots$. For rank $r \geq m_i \cdot p(i)$, $x = \lfloor \frac{r - (m_i \cdot p(i))}{p'[i]} \rfloor$ tells us the number of times sequences of the form $(j^{p(i)})^{b_i}$

for some $0 \leq j \leq m_i$ have appeared after $(m_i^{p(i)})^{m_i}$. Also, note that after the sequence $(m_i^{p(i)})^{m_i}$, $((m_i - 1)^{p(i)})^{b_i}$, $((m_i - 2)^{p(i)})^{b_i} \dots (1^{p(i)})^{b_i}$, $(0^{p(i)})^{b_i}$, $(m_i^{p(i)})^{b_i}$ appear cyclically. Thus, calculating $(m_i - 1 - x) \bmod b_i$ gives us the exact value of the target at position i .

Algorithm 8 Unranking algorithms of directions and targets in L_n^B . The array B has length n . For base B , `unrankDirections`(B, r) returns the corresponding direction d in R_n^B , while `unrankTarget`(B, r) returns the corresponding target T in M_n^B . d and T have length n . Also, in both the algorithms, p' and t have length n .

function `unrankDirections`(B, r)
1: $d = [1, 1, \dots, 1]$
2: $p' = [B[0], B[0] \cdot B[1], \dots, B!]$
3: $t = [m_0, m_1 \cdot p'[0], \dots, m_{n-1} \cdot p'[n-2]]$
4: **for** $i \leftarrow 0, 1, \dots, n-1$ **do**
5: **if** $r \geq t[i]$ **then**
6: $x \leftarrow \lfloor \frac{r-t[i]}{p'[i]} \rfloor$
7: **if** $x \bmod 2 = 0$ **then**
8: $d[i] \leftarrow -1$
9: **return** d

function `unrankTarget`(B, r)
1: $T = [m_0, m_1, \dots, m_{n-1}]$
2: $p' = [B[0], B[0] \cdot B[1], \dots, B!]$
3: $t = [m_0, m_1 \cdot p'[0], \dots, m_{n-1} \cdot p'[n-2]]$
4: **for** $i \leftarrow 0, 1, \dots, n-1$ **do**
5: **if** $r \geq t[i]$ **then**
6: $x \leftarrow \lfloor \frac{r-t[i]}{p'[i]} \rfloor$
7: $j \leftarrow (B[i] - 2 - x) \bmod B[i]$
8: $T[i] \leftarrow j$
9: **return** T

6.4.3 Loopless parallel generation of R_n^b and M_n^b

Suppose that we want p processors to each looplessly generate roughly $c = \lceil \frac{b!}{p} \rceil$ distinct mixed-radix words from $W(b)$. We can accomplish this with the results of Sections 6.4.1 and 6.4.2.

Let `looplessReflected`(B, r, c) be the algorithm obtained by modifying Algorithm 2 so that it initializes $R \leftarrow \text{unrankReflected}(B, r)$, $D \leftarrow \text{unrankDirections}(B, r)$, and $F \leftarrow \text{unrankFocus}(B, r)$; in addition, the main loop is modified to iterate $c - 1$ times. Thus, the algorithm starts at the r th iteration of Algorithm 2 and produces c codewords. We can achieve fully parallelized generation by invoking `looplessReflected`(B, r, c) independently for different values of r . Suppose we invoke `looplessReflected`(B, r, c) in parallel with $r = 0, c, 2c, \dots, (p-1)c$, with each invocation producing c codewords. Then, for each $i \in [0, p-1]$, invoking `looplessReflected`(B, ic, c) produces the codewords of R_n^B having ranks $ic, ic+1, \dots, (i+1)c-1$.

Similarly, let `looplessModular`(B, r, c) be the algorithm obtained by modifying Algorithm 3 so that it initializes $M \leftarrow \text{unrankModular}(B, r)$, $T \leftarrow \text{unrankTarget}(B, r)$, and $F \leftarrow \text{unrankFocus}(B, r)$, and stops after $c - 1$ iterations. Following the explanation above, invoking `looplessModular`(B, ic, c) for $i \in [0, p-1]$, produces the codewords of M_n^B having ranks $ic, ic+1, \dots, (i+1)c-1$, making the parallelization of the generation of M_n^B possible.

In both cases above, the codeword generated at each iteration of Algorithms `looplessReflected`(B, r, c) and `looplessModular`(B, r, c) also takes worst-case $\mathcal{O}(1)$ time, making the generation loopless.

6.5 The ‘modified’ modular Gray code

Notation 6.5.1. We use $[n]^k$ to denote the set of sequences of length k whose entries are in $[1, n]$.

For example, $(1, 3, 4, 4, 1) \in [4]^5$ and we prefer to simply write it as 1, 3, 4, 4, 1 by omitting the parentheses.

We further study several interesting properties of $\text{Ruler}(b)$ and its generalization $\text{Ruler}_r(b)$ (defined later in this section) in our work [51], which helped to prove Proposition 6.5.3, and Theorems 6.5.7 and 6.5.8. We omit these proofs.

6.5.1 Reinterpretation of modular Gray codes and further generalization

Motivated by Example 6.3.11, we can generate the modular Gray code M_n^b as follows. Given a word $w = w_n \cdots w_2 w_1 \in \mathbb{W}(b)$ with bases $b = (b_n, \dots, b_2, b_1)$, we let $\oplus_i(w)$ be the result of incrementing digit i modulo its base, that is,

$$\oplus_i(w) = w_n \dots w_{i+1} v_i w_{i-1} \dots w_1 \quad (6.5.1)$$

where $v_i = 0$ if $w_i = b_i - 1$ and $v_i = w_i + 1$ otherwise. Given a codeword $w \in \mathbb{W}(b)$ and a sequence $S = s_1, s_2, \dots, s_k \in [n]^k$ we let $w \oplus S$ be the sequence of words obtained by repeatedly cyclically incrementing the digits given in the sequence. That is,

$$w \oplus S = w, \oplus_{s_1}(w), \oplus_{s_2}(\oplus_{s_1}(w)), \dots \quad (6.5.2)$$

For example, M_n^b for $b = (2, 2, 3)$ in (6.3.12) uses $\text{Ruler}(b)$ in (6.3.13).

Definition 6.5.2. The *modular order* for b is $M_n^b = 0^n \oplus \text{Ruler}(b)$.

In other words, M_n^b starts at $w = 0^n$ and then repeatedly increments digit w_i modulo b_i , where i is the next value in $\text{Ruler}(b)$.

Proposition 6.5.3. [51] M_n^b is a Gray code of $\mathbb{W}(b)$ using cyclic increments modulo b .

Definition 6.5.4. (Mixed-radix ruler suffix sequence) Let $w = w_n \dots w_2 w_1 \in \mathbb{W}(b)$ and $b = (b_n, \dots, b_2, b_1)$ with $w' = w_{n-1} \dots w_1$ and $b' = (b_{n-1}, \dots, b_1)$ if $n > 1$. Then,

$$\text{Ruler}_{(w)}(b) = \begin{cases} \overbrace{1, 1, \dots, 1}^{b_1-1-w_1 \text{ copies}} & \text{if } n = 1 \\ \underbrace{\text{Ruler}_{(w')}(b'), n, \text{Ruler}(b'), \dots, n, \text{Ruler}(b')}_{b_n-1-w_n \text{ copies of } n} & \text{if } n > 1; \end{cases} \quad (6.5.3)$$

observe that $\text{Ruler}_{(w')}(b')$ is a (possibly strict) suffix of $\text{Ruler}(b')$.

If $\text{Ruler}(b) = s_0, s_1, \dots, s_k$, then $\text{Ruler}_r(b) = s_r, s_{r+1}, \dots, s_k$ is its suffix starting at the r^{th} entry. Alternatively, we specify the suffix using the corresponding word in L_n^b . That is, $\text{Ruler}_{(w)}(b) = \text{Ruler}_r(b)$ if $r = \text{Rank}^b(w)$. Furthermore, $\text{Ruler}(b)$ in (6.5.3) is obtained with $w = 0^n$ (or $r = 0$).

Now, we present the definition of Gray code for $\mathbb{W}_{(w)}(b)$.

Definition 6.5.5. Let $w = w_n \dots w_2 w_1 \in \mathbb{W}(b)$ and $b = (b_n, \dots, b_2, b_1)$. The *modified modular order* for b is $M_n^b(w) = w \oplus \text{Ruler}_{(w)}(b)$.

In other words, $M_n^b(w)$ starts at w and then repeatedly increments digit w_i cyclically modulo its base b_i , where i is the next value in the mixed-radix ruler suffix, $\text{Ruler}_{(w)}(b)$. Thus, $M_n^b = M_n^b(0^n)$ since 0^n is the first codeword in M_n^b . Therefore, applying $\text{Ruler}_{(0^n)}(b) = \text{Ruler}(b)$ gives the same order. We can also say that $M_n^b(r) = M_n^b(w)$ for $r = \text{Rank}^b(w)$. Also, $\text{Ruler}_{(w)}(b)$ has $b! - 1 - r$ numbers if and only if $r = \text{Rank}^b(w)$.

Example 6.5.6. (Revisiting Example 6.3.11) $b = (2, 2, 3)$ and $w = 011$. Then, $\text{Rank}^b(011) = 4$ and $\mathbb{W}_{(w)}(b) = \{011, 012, 100, 101, 102, 110, 111, 112\}$. Furthermore, we provide the order $M_3^b(w)$, where overlines correspond to the bits cyclically incremented. Let $L_3^b(w)$ be the suffix of L_3^b starting from the codeword w .

$$L_3^b(w) = L_3^b(4) = \quad 01\bar{1}, \bar{0}1\bar{2}, 10\bar{0}, 10\bar{1}, \bar{1}0\bar{2}, 11\bar{0}, 11\bar{1}, 112 \quad (6.5.4)$$

$$M_3^b(w) = M_3^b(4) = \quad 01\bar{1}, \bar{0}1\bar{2}, 11\bar{2}, 11\bar{0}, 11\bar{1}, 10\bar{1}, 10\bar{2}, 100 \quad (6.5.5)$$

$$\text{Ruler}_4(b) = \quad 1, 3, \quad 1, \quad 1, \quad 2, \quad 1, \quad 1 \quad (6.5.6)$$

$$M_3^b = 00\bar{0}, 00\bar{1}, 00\bar{2}, 01\bar{2}, 01\bar{0}, \bar{0}1\bar{1}, 11\bar{1}, 11\bar{2}, 1\bar{1}0, 10\bar{0}, 10\bar{1}, 102 \quad (6.5.7)$$

$$\text{Ruler}(b) = \quad 1, \quad 1, \quad 2, \quad 1, \quad 1, 3, \quad 1, \quad 1, \quad 2, \quad 1, \quad 1 \quad (6.5.8)$$

$$L_3^b = 00\bar{0}, 00\bar{1}, 00\bar{2}, 01\bar{0}, 01\bar{1}, \bar{0}1\bar{2}, 10\bar{0}, 10\bar{1}, \bar{1}0\bar{2}, 11\bar{0}, 11\bar{1}, 112 \quad (6.5.9)$$

Note that $M_3^b(w)$ in 6.5.5 is not a suffix (or even a subsequence) of M_3^b in 6.3.12 whereas $\text{Ruler}_4(b)$ in 6.5.6 is a suffix of $\text{Ruler}(b)$ in 6.3.13. Furthermore, $M_3^b(w)$ greedily reorders the codewords of $L_3^b(w)$ in 6.5.4 using cyclic increment. Interestingly, the generations of $M_3^b(w)$ and $L_3^b(w)$ are guided by the same sequence, $\text{Ruler}_4(b)$.

The observations from Example 6.5.6 motivates the following theorem, which is generalized from Proposition 6.5.3.

Theorem 6.5.7. [51] $M_n^b(w)$ is a Gray code of $\mathbb{W}_{(w)}(b)$ using cyclic increments.

We further prove that $M_n^b(w)$ can instead be created greedily, as described in the following theorem.

Theorem 6.5.8. [51] The Gray code $M_n^b(w)$ is created greedily starting from w by cyclically incrementing the least significant digit giving a new word greater than w .

Theorem 6.5.8 and Proposition 6.2.18 further clarify that Constructions 6.2.13, 6.2.16, and Algorithm 3 produce the identical object, M_n^b . Moreover, our *modified modular Gray code* $M_n^b(w)$ (or $M_n^b(r)$) generalizes the original modular Gray code M_n^b in two ways:

1. **Greedy cyclic increment algorithm:** $M_n^b(w)$ starts at w and cyclically increments the rightmost digit that gives a new codeword lexicographically greater than w (that is, higher lexicographic rank).
2. **Change sequence:** $\text{Ruler}_r(b)$ is simply the suffix of $\text{Ruler}(b)$ starting at rank $r = \text{Rank}^b(w)$.

6.5.2 Loopless generation of our modified modular Gray code

We further provide a loopless algorithm for generating $M_n^b(w)$ for any $w \in \mathbb{W}(b)$. Algorithm 9 initializes the current word $\mathbf{M}$ and the focus pointers $\mathbf{F}$ to be the smallest in $\mathbb{W}(\mathbf{B})$ using `unrankLex` and `unrankFocus` from Algorithms 4 and 7, respectively. Furthermore, the targets $\mathbf{T}$ are initialized for each digit w_i to be $B_i - 1$ or one less if it is already equal to this maximum value (that is, $B_i - 1$); this is compactly represented in line 3 of Algorithm 9.

Algorithm 9 A loopless algorithm for generating $M_n^B(\mathbf{r})$. As usual, arrays have length n (except for $\mathbf{F}$ of length $n + 1$) and use 0-based indices.

```

function looplessModularModded( $\mathbf{B}, \mathbf{r}$ )
1:  $\mathbf{M} \leftarrow \text{unrankLex}(\mathbf{B}, \mathbf{r})$                                 ▷ codeword of rank  $\mathbf{r}$  in  $L_n^B$ 
2:  $\mathbf{F} \leftarrow \text{unrankFocus}(\mathbf{B}, \mathbf{r})$                             ▷ rank of  $\mathbf{F}$  in  $L_n^B$ 
3:  $\mathbf{T}_i \leftarrow B_i - 1 - \mathbf{1}\{\mathbf{M}_i = B_i - 1\}, \quad \text{for } i = 0, \dots, |\mathbf{B}| - 1$     ▷ initial targets for  $\mathbf{M}$ 
4: yield  $\mathbf{M}$                                                     ▷ first codeword in  $M_n^B(\mathbf{r})$ 
5: while  $\mathbf{F}[0] < n$  do                                        ▷  $\mathbf{M}[n]$  is invalid
6:    $j \leftarrow \mathbf{F}[0]$                                         ▷ index of change
7:    $b \leftarrow \mathbf{B}[j]$                                        ▷ base of change
8:    $\mathbf{F}[0] \leftarrow 0$                                          ▷ reset focus pointer
9:    $\mathbf{M}[j] \leftarrow (\mathbf{M}[j] + 1) \bmod b$                        ▷ cyclic inc
10:   $\text{limit} \leftarrow \mathbf{M}[j] = \mathbf{T}[j]$                             ▷ equivalent test
11:  if  $\text{limit}$  then                                           ▷  $\mathbf{M}[j]$  at limit
12:     $\mathbf{T}[j] \leftarrow (\mathbf{T}[j] - 1) \bmod b$                        ▷  $\mathbf{M}[j]$  target
13:     $\mathbf{F}[j] \leftarrow \mathbf{F}[j] + 1$                                 ▷ inherit focus pointer
14:     $\mathbf{F}[j + 1] \leftarrow j + 1$                                 ▷ reset focus pointer
15:  yield  $\mathbf{M}$                                                 ▷ next codewords in  $M_n^B(\mathbf{r})$ 

```

6.6 Future work

We summarize our work in Sections 6.3, 6.4, and 6.5 and future interest as follows:

Given a base $b = (b_n, \dots, b_2, b_1)$, the generations of L_n^b , R_n^b , and M_n^b are guided by the same change sequence, $\text{Ruler}(b)$. Our main contribution in this work is to show that each of these three orderings can be generated in parallel. Furthermore, each of R_n^b and M_n^b can also be generated fully in parallel in $\mathcal{O}(1)$ -time in the worst case. Our future interest is to provide an implementation of such loopless parallel generation.

We also introduced a new family of modular Gray code, $M_n^b(w)$, whose change sequence is a suffix of $\text{Ruler}(b)$ and provided a loopless algorithm for its generation. We are further interested in ranking and unranking of codewords in $M_n^b(w)$.

Bibliography

- [1] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. 2nd edition, Wiley, 2001.
- [2] Giorgio Ausiello and Luigi Laura. Directed hypergraphs: Introduction and fundamental algorithms—a survey. *Theoretical Computer Science*, 658:293–306, 2017.
- [3] Mohammad A. Bahmanian and Mateja Šajna. Connection and separation in hypergraphs. *Theory and Applications of Graphs*, 2(2):5, 2015.
- [4] Afonso S. Bandeira. Ten lectures and forty-two open problems in the mathematics of data science. MIT OpenCourseWare, December 2015.
- [5] B. R. Barwell. Cutting string and arranging counters. *Journal of Recreational Mathematics*, 4:1664–168, 1971.
- [6] B. S. Beevers. The greatest product. *The Mathematical Gazette*, 77:91, 1993.
- [7] James R. Bitner, Gideon Ehrlich, and Edward M. Reingold. Efficient generation of the binary reflected Gray code and its applications. *Communications of the ACM*, 19(9):517–521, 1976.
- [8] Alain Bretto. *Hypergraph Theory: An Introduction*. Mathematical Engineering. Springer, 2013.
- [9] Nader H. Bshouty. Linear time constructions of some d -restriction problems. In Vangelis Th. Paschos and Peter Widmayer, editors, *Algorithms and Complexity - 9th International Conference, CIAC 2015, Paris, France, May 20-22, 2015. Proceedings*, volume 9079 of *Lecture Notes in Computer Science*, pages 74–88. Springer, 2015.
- [10] Hong-Bin Chen and Frank K. Hwang. Exploring the missing link among d -separable, $\bar{d}$ -separable and d -disjunct matrices. *Discrete Applied Mathematics*, 155(5):662–664, 2007.
- [11] Martin Cohn. Affine m -ary Gray codes. *Information and Control*, 6(1):70–78, 1963.
- [12] Marco Dalai, Stefano Della Fiore, Adele A. Rescigno, and Ugo Vaccaro. An efficient algorithm for group testing with runlength constraints. *Discrete Applied Mathematics*, 360:181–187, 2025.

- [13] Annalisa De Bonis. Group testing in arbitrary hypergraphs and related combinatorial structures. In Henning Fernau, Serge Gaspers, and Ralf Klasing, editors, *SOFSEM 2024: Theory and Practice of Computer Science - 49th International Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM 2024, Cochem, Germany, February 19-23, 2024, Proceedings*, volume 14519 of *Lecture Notes in Computer Science*, pages 154–168. Springer, 2024.
- [14] Matthew Joseph Demczyk. Algorithms for generating cover-free families. Master’s thesis, University of Ottawa, 2025.
- [15] Dameng Deng, Douglas R. Stinson, and Ruizhong Wei. The Lovász local lemma and its applications to some combinatorial arrays. *Designs, Codes and Cryptography*, 32(1):121–134, 2004.
- [16] Megan Dewar, David Pike, and John Proos. Connectivity in hypergraphs. *Canadian Mathematical Bulletin*, 61(2):252–271, 2018.
- [17] Robert P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, 51(1):161–166, 1950.
- [18] Robert P. Dilworth. A decomposition theorem for partially ordered sets. In *Classic Papers in Combinatorics*, pages 139–144. Springer, 1987.
- [19] Ding-Zhu Du and Frank Kwang-ming Hwang. *Combinatorial group testing and its applications*. World Scientific, 1999.
- [20] Arkadii Georgievich D’yachkov and Vladimir Vasil’evich Rykov. Bounds on the length of disjunctive codes. *Problemy Peredachi Informatsii*, 18(3):7–13, 1982.
- [21] Paul Erdős, Peter Frankl, and Zoltán Füredi. Families of finite sets in which no set is covered by the union of two others. *Journal of Combinatorial Theory, Series A*, 33(2):158–166, 1982.
- [22] Paul Erdős, Peter Frankl, and Zoltán Füredi. Families of finite sets in which no set is covered by the union of r others. *Israel Journal of Mathematics*, 51(1-2):79–89, 1985.
- [23] Philippe Flajolet and Robert Sedgewick. *Analytic combinatorics*. Cambridge University Press, 2009.
- [24] Ivan Flores. Reflected number systems. *IRE Transactions on Electronic Computers*, (2):79–82, 2009.
- [25] Zoltán Füredi. On r -cover-free families. *Journal of Combinatorial Theory, Series A*, 73(1):172–173, 1996.
- [26] Luisa Gargano, Adele A. Rescigno, and Ugo Vaccaro. Low-weight superimposed codes and their applications. In Jianer Chen and Pinyan Lu, editors, *Frontiers in Algorithmics - 12th International Workshop, FAW 2018, Guangzhou, China, May 8-10, 2018, Proceedings*, volume 10823 of *Lecture Notes in Computer Science*, pages 197–211. Springer, 2018.

- [27] Mira Gonen, Michael Langberg, and Alex Sprintson. Group testing on general set-systems. In *IEEE International Symposium on Information Theory, ISIT 2022, Espoo, Finland, June 26 - July 1, 2022*, pages 874–879. IEEE, 2022.
- [28] Frank Gray. Pulse code communication. US patent 2632058, 1953.
- [29] S. L. Greitzer. *International Mathematical Olympiads 1959–1977*. MAA, Washington, 1978.
- [30] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential Coding Theory*. Draft online version, University at Buffalo, 2025.
- [31] P. R. Halmos. *Problems for Mathematicians Young and Old*. MAA, Washington, 1991.
- [32] Richard Hammack, Wilfried Imrich, and Sandi Klavžar. *Handbook of product graphs*. CRC press, 2011.
- [33] Pavol Hell and Jaroslav Nešetřil. *Graphs and Homomorphisms*, volume 34 of *Oxford Graduate Texts in Mathematics Series*. Oxford University Press, 2026.
- [34] Marc Hellmuth, Lydia Ostermeier, and Peter F Stadler. A survey on hypergraph products. *Mathematics in Computer Science*, 6:1–32, 2012.
- [35] Harold V. Henderson and Shayle R. Searle. The vec-permutation matrix, the vec operator and Kronecker products: A review. *Linear and multilinear algebra*, 9(4):271–288, 1981.
- [36] Frank K. Hwang and Vera T. Sós. Non-adaptive hypergeometric group testing. *Studia Sci. Math. Hungar*, 22(1-4):257–263, 1987.
- [37] Thaís Bardini Idalino and Lucia Moura. Nested cover-free families for unbounded fault-tolerant aggregate signatures. *Theoretical Computer Science*, 854:116–130, 2021.
- [38] Thaís Bardini Idalino and Lucia Moura. Structure-aware combinatorial group testing: A new method for pandemic screening. In Cristina Bazgan and Henning Fernau, editors, *Combinatorial Algorithms - 33rd International Workshop, IWOCA 2022, Trier, Germany, June 7-9, 2022, Proceedings*, volume 13270 of *Lecture Notes in Computer Science*, pages 143–156. Springer, 2022.
- [39] Thaís Bardini Idalino and Lucia Moura. A Survey of Cover-Free Families: Constructions, Applications, and Generalizations. In Charles J. Colbourn and Jeffrey H. Dinitz, editors, *New Advances in Designs, Codes and Cryptography*, volume 86 of *Fields Institute Communications*, pages 195–239. Springer, 2024.
- [40] Thaís Bardini Idalino and Lucia Moura. Cover-free families on hypergraphs and combinatorial group testing. *Journal of Combinatorial Optimization*, 51(5):55, 2026.
- [41] Wilfried Imrich, Sandi Klavžar, and Douglas F. Rall. *Topics in graph theory: Graphs and their Cartesian product*. CRC Press, 2008.

- [42] William Kautz and Roy Singleton. Nonrandom binary superimposed codes. *IEEE Transactions on Information Theory*, 10(4):363–377, 1964.
- [43] Donald E. Knuth. *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1*. Pearson Education India, 2011.
- [44] Donald L. Kreher and Douglas R. Stinson. *Combinatorial algorithms: generation, enumeration, and search*. CRC press, 1998.
- [45] L. C. Larson. *Problem-Solving Through Problems*. Springer Verlag, Berlin, 1983.
- [46] P.C. Li, G.H.J. Van Rees, and R. Wei. Constructions of 2-cover-free families and related separating hash families. *Journal of Combinatorial Designs*, 14(6):423–440, 2006.
- [47] David Lubell. A short proof of Sperner’s lemma. *Journal of Combinatorial Theory, Series A*, 1(2):299, 1966.
- [48] Karen Meagher and Brett Stevens. Covering arrays on graphs. *Journal of Combinatorial Theory, Series B*, 95(1):134–151, 2005.
- [49] Robin A. Moser. A constructive proof of the lovász local lemma. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 343–350. ACM, 2009.
- [50] Robin A. Moser and Gábor Tardos. A constructive proof of the general Lovász local lemma. *Journal of the ACM (JACM)*, 57(2):1–15, 2010.
- [51] Lucia Moura, Prangya Parida, Brett Stevens, and Aaron Williams. One sequence to rule them all: $\mathcal{O}(1)$ -time parallel generation of mixed-radix Gray codes. In Florent Foucaud and Aline Parreau, editors, *Combinatorial Algorithms - 37th International Workshop, IWOCA 2026, Clermont-Ferrand, France, June 8-11, 2026, Proceedings*, volume 16587 of *Lecture Notes in Computer Science*, pages 470–485. Springer, 2026.
- [52] Torsten Mütze. Combinatorial Gray codes—an updated survey. *The Electronic Journal of Combinatorics*, Dynamic Surveys:#DS26, 2023.
- [53] D. J. Newman. *A Problem Seminar*. Springer Verlag, Berlin, 1982.
- [54] Pavlos Nikolopoulos, Sundara Rajan Srinivasavaradhan, Tao Guo, Christina Fragouli, and Suhas N. Diggavi. Group testing for connected communities. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 2341–2349. PMLR, 2021.
- [55] Pavlos Nikolopoulos, Sundara Rajan Srinivasavaradhan, Tao Guo, Christina Fragouli, and Suhas N. Diggavi. Group testing for overlapping communities. In *ICC 2021 - IEEE International Conference on Communications, Montreal, QC, Canada, June 14-23, 2021*, pages 1–7. IEEE, 2021.

- [56] Prangya Parida and Lucia Moura. Cover-free families on graphs. *arXiv:2605.12634*, 2026.
- [57] Prangya Parida and Aaron Williams. Mixed-radix ruler sequence Gray codes. <https://github.com/ppari017/Mixed-radix-Ruler-Sequence-Gray-Codes>, 2026. GitHub repository.
- [58] Ely Porat and Amir Rothschild. Explicit nonadaptive combinatorial group testing schemes. *IEEE Transactions on Information Theory*, 57(12):7982–7989, 2011.
- [59] Yuan Qiu, Joe Sawada, and Aaron Williams. Maximize the rightmost digit:gray codes for restricted growth strings. In Shin-ichi Nakano and Mingyu Xiao, editors, *WALCOM: Algorithms and Computation - 19th International Conference and Workshops on Algorithms and Computation, WALCOM 2025, Chengdu, China, February 28 - March 2, 2025, Proceedings*, volume 15411 of *Lecture Notes in Computer Science*, pages 296–311. Springer, 2025.
- [60] Sebastian Raaphorst. *Variable strength covering arrays*. PhD thesis, University of Ottawa (Canada), 2013.
- [61] Adele A. Rescigno and Ugo Vaccaro. Bounds and algorithms for generalized superimposed codes. *Information Processing Letters*, 182:106365, 2023.
- [62] Miklós Ruszinkó. On the upper bound of the size of the r -cover-free families. *Journal of Combinatorial Theory, Series A*, 66(2):302–310, 1994.
- [63] J. Scholes. 18th IMO 1976, Problem 4, 1976. Available electronically at <https://prase.cz/kalva/imo/isoln/isoln764.html>.
- [64] J. Scholes. 40th Putnam 1979, Problem a1, 1979. Available electronically at <https://prase.cz/kalva/putnam/psoln/psol791.html>.
- [65] N. J. A. Sloane and The OEIS Foundation Inc. A000792. <https://oeis.org/A000792>. On-Line Encyclopedia of Integer Sequences.
- [66] N. J. A. Sloane and The OEIS Foundation Inc. A001511. <https://oeis.org/A001511>. On-Line Encyclopedia of Integer Sequences.
- [67] M. Sonntag. *Hamiltonsche Eigenschaften von Produkten von Hypergraphen*. Habilitation, Fakultät für Mathematik und Naturwissenschaften, Bergakademie Freiberg, 1991.
- [68] Emanuel Sperner. Ein Satz über Untermengen einer endlichen Menge. *Mathematische Zeitschrift*, 27(1):544–548, 1928.
- [69] Douglas R. Stinson and Ruizhong Wei. Generalized cover-free families. *Discrete Mathematics*, 279(1-3):463–477, 2004.
- [70] Douglas R. Stinson, Ruizhong Wei, and Lie Zhu. Some new bounds for cover-free families. *Journal of Combinatorial Theory, Series A*, 90(1):224–234, 2000.

- [71] Vadim G. Vizing. On an estimate of the chromatic class of a p -graph. *Diskret. analiz.*, 3:25–30, 1964.
- [72] Ruizhong Wei. On cover-free families. *arXiv:2303.17524*, 2006.
- [73] Aaron Williams. The greedy Gray code algorithm. In Frank Dehne, Roberto Solis-Oba, and Jörg-Rüdiger Sack, editors, *Algorithms and Data Structures*, volume 8037 of *Lecture Notes in Computer Science*, pages 525–536. Springer, 2013.
- [74] Hong-Gwa Yeh. d -disjunct matrices: Bounds and Lovász local lemma. *Discrete mathematics*, 253(1-3):97–107, 2002.