Turán-Type Problems in Group Testing, Coding Theory and Cryptography

by

Yeow Meng Chee

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

Waterloo, Ontario, Canada, 1996

I hereby declare that I am the sole author of this thesis.

I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Waterloo to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

Turán-type problems are those which ask for the maximum number of blocks in a set system of a given order, which avoids a given set of configurations. We study the problem of designing low complexity nonadaptive algorithms for group testing, and the problem of constructing efficient erasure-resilient codes for large disk arrays, and frameproof codes for digital fingerprinting. These three problems are shown to yield a common treatment as Turán-type problems. The goal of this thesis is to develop bounds as well as to characterize optimal solutions to these problems. For the former, we focus on giving bounds that are at least asymptotically optimal, that is, optimal up to constant factors.

We obtain characterizations of optimal r-cover-free k-uniform set systems for $(r, k) \in \{(2,3),(3,4),(4,5)\}$. When $k = r+1$ or $k \equiv 1 \pmod 2$, we exhibit bounds that are stronger than previous ones of Erdös, Frankl, and Füredi.

Next, weakly union-free twofold triple systems are shown to be equivalent to optimal nonadaptive algorithms for a certain group testing problem requiring only approximate identification. We investigate the spectrum of these set systems, and prove that the elementary necessary conditions are also sufficient for their existence, except perhaps for

a small finite number of cases. This settles a conjecture of Frankl and Füredi in the affirmative.

We also study a category of set systems arising from fault-tolerant nonadaptive group testing algorithms. Optimal solutions are obtained for all compositions of block sizes in $\{1, 2, 3\}$. In the process, we complete the spectrum of a class of designs (called quasidesigns) introduced by Frankl and Füredi.

The next application area we investigate is the design of erasure-resilient codes for disk arrays. We prove general upper and lower bounds on the maximum size of such codes. The lower bound comes from a construction based on expander graphs. By studying set systems associated with (k, l)-erasure-resilient codes, asymptotically optimal bounds for such codes are established for all $k \leq l \leq 2k - 1$, when $k = 3$ and 4. We then study the problem of controlling group sizes in erasure-resilient codes, which leads to resolvability properties of the associated set systems. All these results improve, generalize, and/or extend previous results of Hellerstein, Gibson, Karp, Katz, and Patterson. It is also shown that erasure-resilient codes can be used to construct r-difference-free set systems, which correspond to nonadaptive group testing algorithms for the parity model. We prove asymptotically optimal bounds for 2-difference-free 3-uniform set systems.

The final results of this thesis concern r-frameproof codes. These codes can be used to fingerprint digital data so that unauthorized use and copying of a piece of data can be traced back to its user. Moreover, no coalitions of at most r users can frame other users of unauthorized actions. We give improved bounds on 2-frameproof codes, and exhibit for every r, the first explicit family of r-frameproof codes whose rate is bounded away from zero.

The results of this thesis indicate the pertinent role of Turán-type problems in group testing, erasure-resilient codes, and frameproof codes.

Credits

Charlie Colbourn and Alan Ling have made some helpful suggestions during the preliminary part of research that culminated in Chapter 6. The electronic MOLS table that Charlie made available to me has been extremely useful. The material in Chapter 8 is joint work with Charlie Colbourn and Alan Ling. I thank them for this collaboration.

My thesis committee comprises Gord Agnew, Charlie Colbourn, Frank Hwang, Ron Mullin, Paul Schellenberg, and Scott Vanstone.

Acknowledgements

After earning my Master's degree from Waterloo in 1989, I was back in Singapore for a period of about five years. Many important decisions were made during that time. Amidst all these was the decision to come back to Waterloo for my Ph.D. During the span of five years in Singapore and two years that followed in Waterloo, I have been fortunate to meet many people who have influenced me greatly and who have made it possible for me to realize my dream. One of the delights of finally finishing my Ph.D. is this opportunity to thank them.

I am greatly indebted to Charlie Colbourn, my advisor, mentor, and friend. Charlie gave me new directions in a time when I was wandering in the jungle of research. He got me thinking about many problems upon which this thesis is based. This thesis would not have come into being if he had not introduced me to group testing, and formulated the problem on erasure-resilient codes. Charlie's generosity with time also made it possible for me to discuss whatever ideas I have whenever I like. It is well known that most of Charlie's students graduated in relatively short time. Many attribute this to his folklorish ability to push his students. Having worked with Charlie in two degree programs, I have gotten to know sufficiently many of his students to know what the truth is. Charlie's commitment to research is such an inspiration that his students are propelled to work even harder. I am also grateful to Charlie for his financial support which enabled me to attend many interesting conferences.

A special thank goes to Ron Mullin who is ever so willing to help and support. It was Ron who, nine years ago when I was an undergraduate, taught me what research is by offering me a reading course in cryptography. Many of the ideas I encountered in that course have been useful in my later work. Ron's phenomenally good temper and patience also merit mention. Ron will be turning 60 this August. I wish him a very happy 60th birthday and many healthy years ahead.

Gord Agnew, Paul Schellenberg, and Scott Vanstone all served on my thesis committee. I am grateful to Gord for taking the time to make a careful reading of the thesis, despite his busy wedding preparations. Paul is to be commended for his readiness to serve on the committee as a substitute for Gord despite his heavy administrative duties. Anyone who has tried looking for Scott would appreciate how busy his schedule is. I am thankful to him for agreeing to serve on my committee without hesitation.

Thanks also to Frank Hwang who consented to be my external examiner.

The first year of my Ph.D. studies was supported financially by Anna Lubiw who was also my advisor. I would like to express my thankfulness to her for providing resources which made it possible for me to come to Waterloo.

San Ling, at the National University of Singapore, played an important role in getting me to work on applications of algebraic number theory in combinatorics through his excellent series of lectures on elliptic curves. I thank him for several happy collaborations.

The last two years I had in Waterloo were a time of hard work but great pleasure. I would like to thank my academic siblings, Zhike Jiang and Alan Ling, for stimulating discussions on combinatorics, and their friendship. The oyster dinner we had together

To Angeline,

the inspiration of all things wondrous.

Table of Contents

List of Figures

Chapter 1

Introduction

The investigation of the structure of finite set systems constitutes a large part of the development of combinatorics. Several approaches have been taken, but the one that is oldest and yet continues to yield deep insights and a rich source of interesting problems, has come to be known as the branch of Turán-type problems. In a Turán-type problem, we are given a class $\mathcal{G}$ of set systems, and an invariant μ (usually the number of blocks) for another class $\mathcal{H}$ of set systems. The problem is to determine the maximum of μ over all set systems in $\mathcal{H}$ that contain none of the elements in $\mathcal{G}$ as a subsystem. Turán-type problems are not merely mathematical curiosities. Many of them correspond naturally to problems of practical interest. In this dissertation, we study several Turán-type problems that arise from three areas of computer science and engineering which have received much attention recently: group testing, erasure-resilient codes for disk arrays, and frameproof codes for digital fingerprinting. Our goal is to construct nonadaptive group testing algorithms, erasure-resilient codes, and frameproof codes that are as efficient as possible. The nonadaptive group testing algorithms, erasure-resilient codes, and frameproof codes that we build are improvements on previous results. Characterizations of some classes of optimal nonadaptive group testing algorithms are also obtained.

1.1 Group Testing

The concept of group testing was introduced to deal with a laborious and expensive process in clinical medicine. During World War II, the United States Public Health Service and the Selective Service System found it necessary to sieve out syphilitic Americans who were inducted into military service. A sample of blood drawn from each prospective inductee was subjected to a laboratory analysis which revealed the presence or absence of syphilitic antigen. The presence of syphilitic antigen was taken as an indication of infection. Instead of performing such a test on each blood sample, it was proposed that blood samples be pooled in groups and analyzed. If a pool showed no trace of syphilitic antigen, then all the individuals contributing to that pool could be assumed to be uninfected. If, however, syphilitic antigen was detected in a pool, then each individual contributing to that pool must be tested again. The merit of this proposal is that while a test is wasted when the pool contains blood samples of one or more infected individuals, many tests can be saved if the pool turns out to be free from syphilitic antigen. This idea is usually attributed to Dorfman, who wrote the first paper [46] in the area. However, it seems that Rosenblatt is the first to suggest the idea. More interesting history can be found in a recent book of Du and Hwang [48].

A number of industrial inspection problems share many similarities with the blood testing problem described above:

1. Detecting gas leakage in devices [135].

2. Testing electronic components for faults [135].

3. Locating electrical shorts [35, 67, 132].

The same technique of pooling several objects in groups and inspecting them collectively can be used to reduce the cost of these inspection processes. The phrase "group testing" is coined by Sobel and Groll [135] to describe this general technique.

More recently, group testing has found new applications in network communication. In a multiple access channel, there are two or more users sending information to a common receiver using the channel. Such channels provide a way for a large number of geographically dispersed stations to communicate. They have many attractive features, including low cost and potential for high bandwidth. An example of this channel is a satellite receiver with many independent ground stations. One problem with multiple access channels surfaces when two or more users transmit simultaneously. A common model [114] for multiple access channels assumes that the transmissions interfere destructively. A simple protocol that resolves such conflicts is time division multiplexing (TDM). In a TDM protocol, the time horizon is divided into units called slots, such that any message of unit length can be transmitted in one slot. If there are n users, we define a step to be a period of n slots. During each step, the TDM protocol allocates a slot to each user during which the user can send any message of unit length. The step is then repeated. Time division multiplexing can be highly inefficient since slots are allocated even to users who do not wish to transmit. To overcome this drawback, Hayes [77] suggested that users who wish to transmit should first be identified and then allotted slots. We are thus faced with a set of users from which we want to identify those who wish to transmit. The role of group testing is now evident. A group test comprises a poll to a subset of users to determine if any of them wishes to transmit. How these subsets of users are to be constructed to allow fast identification is addressed by many researchers [14, 28, 29, 30, 36, 71, 77, 87, 144].

Even more recently, applications of group testing have come a full circle. From its initial inception in a clinical problem, group testing is now widely used in several efforts of the Human Genome Project. The goal of the Human Genome Project is to analyze the

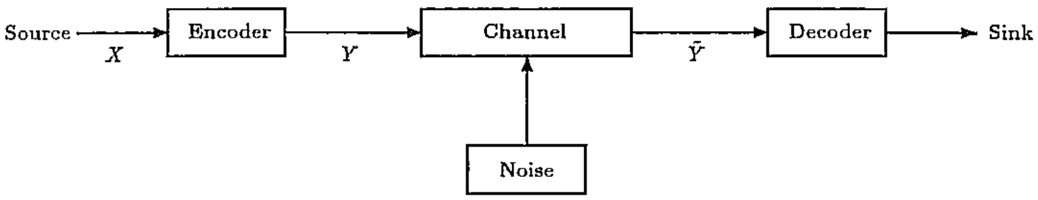


Figure 1.1: A communication system.

structure of human DNA and to determine the location of the estimated 10^5 human genes. Much of the current effort of this project involves screening large libraries of recombinant DNA in order to isolate clones containing a particular DNA sequence. The experimental test used is known as *polymerase chain reaction.* This screening is an important preliminary to disease-gene mapping and large scale clone mapping [109]. The isolation of clones containing DNA sequences of interest is a tedious process and has been done using the idea of group testing to reduce the amount of work involved [10, 25]. Group testing has also been considered for sequencing by hybridization [112].

In each of the applications discussed above, the objective is always to minimize the number of tests used while still being able to identify those objects with the desired property.

1.2 Erasure-Resilient Codes

In Shannon's seminal work on information theory [130], the transfer of digital information from a source to a sink is modeled mathematically by a communication system depicted in Figure 1.1. A source signal is a vector X, where its components belong to a finite set called the source alphabet. The encoder transforms X to another vector Y with components also from a finite set called the channel input alphabet. This vector Y, called a codeword, is then transmitted through the channel which is occasionally corrupted by noise. At the

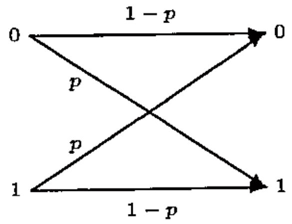


Figure 1.2: Binary symmetric channel.

end of the channel, the decoder tries to reconstruct the source signal from the received signal $\tilde{Y}$. Such a communication system is known as a discrete channel.

It is necessary that enough redundant information be added by the encoder if the decoder is to be able to reconstruct the source signal from the distorted received signal. The fundamental problem in coding theory [130] is how to construct for a given channel with a specified noise function, an encoder/decoder pair so that even if part of the transmitted codeword is distorted by noise in the channel, the decoder can nevertheless deduce what the source signal is. The objective here is for the encoder to introduce as little redundant information as possible, and for the decoder to tolerate as much distortion as possible. We cannot, of course, fulfil both of these conditions at the same time. The problem is how good we can reconcile these aims.

The theory of error-correcting codes (see [96]) addresses this problem for a discrete channel whose noise function may distort a codeword by replacing some of its components with other symbols in the channel input alphabet. A distorted component is called an error. The most frequently studied channel in the theory of error-correcting codes and information theory is the binary symmetric channel (Figure 1.2), where the channel input alphabet is $\{0, 1\}$, and input symbols are complemented with probability p. The theory of error-correcting codes is well developed and Spielman's recent breakthrough [139] gives an asymptotically good family of error-correcting codes that can be encoded and decoded in linear time.

The binary erasure channel is illustrated in Figure 1.3. In this channel, the channel input alphabet is also $\{0,1\}$, but the noise function changes any symbol in $\{0,1\}$ to a special symbol '#' with probability p. This has the effect of erasing symbols in a codeword. An erased symbol is called an erasure. Hence, in a binary erasure channel, we know exactly where erasures have occurred just by looking at the received vector $\bar{Y}$, but we receive no values for those erased components. The binary erasure channel is also well analyzed in information theory but work on the coding-theoretic aspect of this channel has begun only recently [7, 79, 115]. There are two main reasons for this. Firstly, any error-correcting code that tolerates up to e errors also tolerates up to e erasures. So it seems more important that we understand and develop first the theory of error-correcting codes. Secondly, most of the communication systems in use during the inception of coding theory behave more like binary symmetric channels. This is no longer true. Today's packet-switched networks (the Internet, for example) face applications that generate bursty traffic, causing congestions and buffer overflows that can lead to unpredictable losses [3]. This is undesirable for real-time multimedia applications. A substantial part of the Priority Encoding Transmission (PET) project [3] at the International Computer Science Institute in Berkeley focuses on developing erasure-resilient codes for packet-switched networks.

Another manifestation of binary erasure channels involves viewing storage devices as communication channels. The requirement for high-performance, highly available storage for file servers and supercomputing systems led to the development of Redundant Arrays of Inexpensive Disks (RAID) [111]. The reliability of a large array of disks can be an issue, even if each disk in the array is highly reliable. A recent survey of Özden, Rastogi, and Silberschatz [110] gives the design of fault-tolerant disk arrays as a significant research area in multimedia applications. The failure of any disk in a disk array corresponds to an erasure of data stored on that disk. Erasure-resilient codes for handling failures in disk

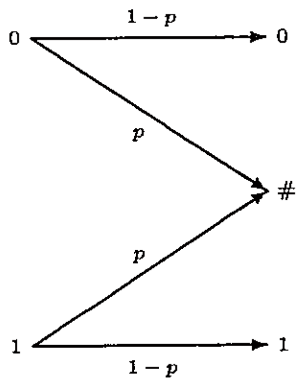


Figure 1.3: Binary erasure channel.

arrays thus become an important study.

There are essentially two approaches to measuring the performance of a code. The first is an average-case analysis, which gives the performance of a code in terms of its error probability, that is, the probability that an error cannot be corrected. The second approach is a worst-case analysis which measures the performance of a code by the number of errors appearing in any codeword which can be corrected with certainty. Although the two measures are closely related, it is usually more convenient to describe a code by its worst-case performance [133]. This is also the approach adopted in this dissertation.

1.3 Frameproof Codes

The goal of cryptography is to solve the problem of communication in the presence of adversaries [121]. In this dissertation, we focus on the problem of protecting digital data against unauthorized use or copying. Here, the user of the data plays the role of the adversary, and the data tries to communicate to its distributor whether any unauthorized use or copying has taken place. For nondigital products, this problem is often solved by physically incorporating an identifier, called a fingerprint, into each product. These identifiers help to trace the products back to their users, and hence act as a deterrent of unauthorized use.

The design of fingerprinting mechanisms is much harder for digital data, since these can be processed and manipulated easily. Many procedures which are infeasible for users to perform on nondigital products are simply accomplished for digital ones. For example, the ability of the users to collude and compare every bit of a digital data allows them to detect the position of the fingerprints, and subsequently to modify them. If the design of fingerprints is not careful, it is also possible for users to collude and produce legitimate fingerprints, allowing them to frame another user (not in the collusion) of unauthorized actions.

The problem of constructing fingerprints that are tolerant to the adversarial behaviour of users discussed above has been addressed recently by Boneh and Shaw [19]. They introduced a class of codes, called frameproof codes, that would prevent collusions of users from framing other users. Frameproof codes are an interesting deterrent of software piracy, and warrant investigation.

1.4 Organization of This Thesis

We begin in Chapter 2 with an introduction to some basic notation, terminology, and results in analysis, combinatorics, algebra, and number theory that are used in this dissertation.

Chapter 3 is devoted to formalizing group testing models and group testing problems. The purpose is to provide a formal context for the results in subsequent chapters. The chapter ends with a section containing a list of problems that are of interest in this dissertation.

In Chapter 4, we show how Turán-type problems arise from nonadaptive group testing problems. In particular, we see the applications of r-union-free and r-cover-free set systems in nonadaptive group testing problems where exact identification of the target set is required.

Chapter 5 is devoted to the study of uniform r-cover-free set systems. We present a characterization of several classes of optimal r-cover-free uniform set systems. In some other situations where we did not manage to achieve such characterizations, new upper bounds on the number of blocks are obtained. These improve on previous bounds of Erdös, Frankl, and Füredi [57, 58].

Chapter 6 deals with the existence problem for weakly union-free twofold triple systems. This problem was first studied by Frankl and Füredi [62] as a generalization of an old result of Erdös [54] on graphs. We show that this Turán-type problem arises naturally in a certain nonadaptive group testing problem where only an approximate identification of the target set is required. We solve the existence problem for weakly union-free twofold triple systems completely except for a small finite number of cases, thus settling a conjecture of Frankl and Füredi in the affirmative. We also determine completely those orders for which there exists a twofold triple system that avoids all twofold triple systems of smaller order.

The theme of Chapter 7 is fault-tolerant group testing. We study nonadaptive group testing algorithms that can identify target sets of size at most two even in the presence of an erroneous test, subjecting each of the elements to no more than three tests. It is found that the nonadaptive algorithm with the lowest test complexity involves each element in precisely three tests. The 2-union-free 3-uniform set systems constructed by Frankl and Füredi [62] is found to have the desired properties. We also complete the spectrum of a class of designs introduced by Frankl and Füredi.

We shift our attention in Chapter 8 to the problem of designing erasure-resilient codes for large disk arrays. It turns out that the problem can be expressed equivalently as a Turán-type problem. We provide a general lower bound obtained from a construction based on expanders. Viewing the design of erasure-resilient codes as a Turán-type problem, we improve on this lower bound for some parameter situations by constructing erasure-resilient codes that are better than any presently known. The results we obtain here are optimal up to a constant factor. In this respect, we make heavy use of techniques from design theory. Often, looking at the associated Turán-type problem gives extremely simple and direct proofs of existing results (and even their improvements). This gives evidence that perhaps treating Turán-type problems is the right approach to the design of erasure-resilient codes for large disk arrays. The chapter also contains an application of erasure-resilient codes to the nonadaptive group testing problem where the test function used is the MOD_2 test function.

In Chapter 9 we consider yet another area in which Turán-type problems arise naturally. Very recently, Boneh and Shaw have considered cryptographic techniques for protecting unauthorized use and copying of digital data, with the requirement that we be able to trace unauthorized actions back to their originators, and no coalitions below a certain size can frame other users. Special codes, known as r-frameproof codes, can be used to solve the problem. Stinson and Wei have observed that the problem of designing r-frameproof codes is equivalent to a Turán-type problem. We give a probabilistic construction of 2-frameproof codes, improving earlier bounds. We also exhibit for every r, the first explicit constructible family of r-superimposed codes, and hence also r-frameproof codes, whose rate is bounded away from zero.

The final chapter summarizes the results of this dissertation and discusses some of its ramifications.

CHAPTER 2

Mathematical Preliminaries

This chapter summarizes mathematical background material from analysis, combinatorics, coding theory, algebra, and number theory used in this thesis. The definitions and results listed are mainly meant for reference.

2.1 Basic Notation

By $\mathbf{R}$ ($\mathbf{Z}$, $\mathbf{N}$) we denote the set of real (integral, natural) numbers. The set $\mathbf{N}$ of natural numbers does not contain zero. $\mathbf{R}_+$ ($\mathbf{Z}_+$) denotes the nonnegative real (integral) numbers.

Let M be a set. For $n \in \mathbf{N}$, we denote by M^n the set of vectors with n components (or n-dimensional vectors) with entries in M. We sometimes call a vector in M^n an *M-vector.*

Definition 2.1.1 The *weight* of a vector $\mathbf{v} = (v_1, v_2, \ldots, v_n) \in \{0,1\}^n$, denoted $\mathrm{wt}(\mathbf{v})$, is the number $\sum_{i=1}^{n} v_i$.

Definition 2.1.2 The *support* of a vector $\mathbf{v} = (v_1, v_2, \ldots, v_n) \in \{0,1\}^n$, denoted $\mathrm{supp}(\mathbf{v})$, is the set $\{i \mid v_i = 1\}$.

Definition 2.1.3 Given a vector $\mathbf{v} = (v_1, v_2, \ldots, v_n) \in \mathbf{R}^n$, and a set $I = \{i_1, i_2, \ldots, i_m\} \subseteq \{1, 2, \ldots, n\}$, such that $i_1 < i_2 < \cdots < i_m$, the *restriction of* $\mathbf{v}$ *to* I, denoted $\mathbf{v}|_I$, is the vector $(v_{i_1}, v_{i_2}, \ldots, v_{i_m})$.

Definition 2.1.4 A vector $\mathbf{u} = (u_1, u_2, \ldots, u_n) \in \mathbf{R}^n$ is said to *precede* another vector $\mathbf{v} = (v_1, v_2, \ldots, v_n) \in \mathbf{R}^n$, and we write $\mathbf{u} \preceq \mathbf{v}$, if $u_i \leq v_i$ for all $i \in \{1, 2, \ldots, n\}$.

The j-th unit vector in $\mathbf{R}^n$, whose j-th component is one while all other components are zero, is denoted by $\mathbf{e}_j$. The zero vector is denoted by $\mathbf{0}$ and the vector of all ones is denoted by $\mathbf{1}$.

The component-wise Boolean sum of two $\{0, 1\}$-vectors $\mathbf{u}$ and $\mathbf{v}$ is written $\mathbf{u} \vee \mathbf{v}$.

For any set M, we denote by $M^{m \times n}$ the set of $m \times n$ matrices with entries in M. The identity matrix is denoted by I. The transpose of a matrix A is denoted by A^T.

For a real number α, the symbol $\lfloor \alpha \rfloor$ denotes the largest integer not larger than α, and $\lceil \alpha \rceil$ denotes the smallest integer not smaller than α.

For two sets M and N, we write $M \setminus N$ for the set-theoretic difference $\{x \in M \mid x \notin N\}$, $M \Delta N$ for the symmetric difference $(M \setminus N) \cup (N \setminus M)$, and 2^M for the set of all subsets of M. If $k \in \mathbf{Z}_+$, a *k-subset of M* is a set $N \subseteq M$ such that $|N| = k$. The set of all k-subsets of M, $\{N \subseteq M \mid |N| = k\}$, is denoted by $\binom{M}{k}$. If $M \cap N = \varnothing$, the expression $M \dot\cup N$ is often used in place of $M \cup N$ to emphasize that the two sets are disjoint.

The discrete probability measure is denoted by $\mathbf{Pr}$ and $\mathbf{E}[X]$ denotes the expectation of a random variable X.

2.2 Analysis

We use the classical notations for asymptotic analysis. Let $f, g : \mathbf{R} \to \mathbf{R}_+$.

(i) We say that $f(n) = O(g(n))$ if there exist positive numbers c and N such that, for all $n \geq N$, $f(n) \leq cg(n)$.

(ii) We say that $f(n) = \Omega(g(n))$ if there exist positive numbers c and N such that, for all $n \geq N$, $f(n) \geq cg(n)$.

(iii) We say that $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$ both hold.

(iv) We say that $f(n) = o(g(n))$ if $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$.

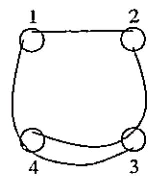
The natural logarithm is denoted by "ln" while "log" denotes the logarithm to base two. We use "exp" to denote the exponential function.

2.3 Combinatorics

Definition 2.3.1 Let X be a finite set. A *set system* or *configuration* is a pair $(X, \mathcal{A})$, where $\mathcal{A} \subseteq 2^X$. The *order* of the set system is $|X|$. The elements of X are called *points* and the elements of $\mathcal{A}$ are called *blocks*.

Our definition of a set system precludes repeated blocks. A set system $(X, \mathcal{A})$ is represented diagrammatically by a set of points corresponding to the elements of X and each block $A \in \mathcal{A}$ is drawn as a continuous curve passing through precisely those points comprising A.

Example 2.3.1 Consider the set system $(X, \mathcal{A})$, where $X = \{1, 2, 3, 4\}$ and $\mathcal{A} = \{\{1, 2\}, \{1, 3, 4\}, \{2, 3, 4\}\}$. This set system is represented diagrammatically as follows.



Definition 2.3.2 A set system $(X, \mathcal{A})$ is *k-uniform* if $\mathcal{A} \subseteq \binom{X}{k}$.

We sometimes say that a set system is *uniform* if it is k-uniform for some k.

Definition 2.3.3 Two set systems $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ are *isomorphic* if there exists a bijection, called an *isomorphism*, $\pi : X \to Y$ such that $A \in \mathcal{A}$ if and only if $\{\pi(a) \mid a \in A\} \in \mathcal{B}$.

Definition 2.3.4 If $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ are set systems such that $Y \subseteq X$ and $\mathcal{B} \subseteq \mathcal{A}$, we say that $(Y, \mathcal{B})$ is a *subsystem* of $(X, \mathcal{A})$.

Definition 2.3.5 A set system $(X, \mathcal{A})$ is said to *contain* a configuration $(Y, \mathcal{B})$ if there exists a subsystem of $(X, \mathcal{A})$ that is isomorphic to $(Y, \mathcal{B})$.

Definition 2.3.6 A set system $(X, \mathcal{A})$ *avoids* a configuration $(Y, \mathcal{B})$ if $(X, \mathcal{A})$ does not contain $(Y, \mathcal{B})$. In this case, we also say that $(Y, \mathcal{B})$ is a *forbidden configuration* of $(X, \mathcal{A})$.

The following definitions pertain to group actions on set systems.

Definition 2.3.7 An *automorphism* of a set system $(X, \mathcal{A})$ is an isomorphism from $(X, \mathcal{A})$ onto itself.

Definition 2.3.8 The set of all automorphisms of a set system forms a group Γ, called its *full automorphism group*, under functional composition. Any subgroup of Γ is simply referred to as *an automorphism group*.

Definition 2.3.9 Let Γ be a group acting on a set X. For $S \subseteq X$, the *development of S with Γ*, denoted $\text{dev}_\Gamma(S)$, is the set $\{\{\gamma(s) \mid s \in S\} \mid \gamma \in \Gamma\}$.

Definition 2.3.10 A collection of *starter blocks* for a set system $(X, \mathcal{A})$, with automorphism group Γ, is a subset $\mathcal{A}' \subseteq \mathcal{A}$ such that $\mathcal{A} = \bigcup_{A \in \mathcal{A}'} \text{dev}_\Gamma(A)$.

2.4 Coding Theory

A *code*, or more specifically a *q-ary code of length n*, is any subset $\mathcal{C} \subseteq \{0, 1, \ldots, q-1\}^n$. The elements of $\mathcal{C}$ are called *codewords*. An important parameter of a code is its *Hamming distance*.

Definition 2.4.1 The *Hamming distance between two codewords* $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ and $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ is the quantity $\mathrm{dist}(\mathbf{u}, \mathbf{v}) = |\{i \mid u_i \neq v_i\}|$.

Definition 2.4.2 The *minimum distance* of a code $\mathcal{C}$ is

$$\mathrm{d}(\mathcal{C}) = \min_{\substack{\mathbf{u},\mathbf{v} \in \mathcal{C} \\ \mathbf{u} \neq \mathbf{v}}} \mathrm{dist}(\mathbf{u}, \mathbf{v}).$$

Definition 2.4.3 The *relative minimum distance* of a code $\mathcal{C}$ of length n is $\delta(\mathcal{C}) = \mathrm{d}(\mathcal{C})/n$.

Another important parameter in coding theory is the *rate* of a code. This is defined as follows.

Definition 2.4.4 The *rate* of a q-ary code, $\mathcal{C}$, of length n is $\mathrm{Rate}(\mathcal{C}) = \frac{\log_q |\mathcal{C}|}{n}$.

We define a *family of codes* to be an infinite sequence of codes that contain at most one code of any length.

Definition 2.4.5 A family of codes, $\{\mathcal{C}_i\}_{i=1}^{\infty}$, is said to have *rate* R if $\mathrm{Rate}(\mathcal{C}_i) \geq R$ for all i.

2.5 Algebra and Number Theory

The finite field with q elements, where q is a prime power, is denoted by $\mathrm{GF}(q)$. $\mathbf{Z}_n$ represents the ring of integers modulo n.

If σ is a permutation, the group generated by σ is denoted $\langle\sigma\rangle$.

The following result on the gap between consecutive primes is useful.

Theorem 2.5.1 (Mozzochi [101]) Let p_n denote the n-th prime. Then $p_{n+1} - p_n = O(p_n^{1051/1921+\epsilon})$, for any $\epsilon > 0$.

CHAPTER 3

Models of Group Testing

3.1 A Group Testing Game

Consider the one-player *simple group testing game.* The object of the game is to identify an unknown subset U of a finite set X, where $|U| \leq r$. We shall call U the *target set.* The player receives information about U only through the following process. The player chooses an arbitrary subset P of X and is told whether P contains at least one element of U or no elements of U. We call P a *pool* and the process of obtaining information a *test on* P. We often view a test on P as the evaluation of some *test function* $f_U : 2^X \rightarrow R$ on P. The appearance of the subscript U is to remind the reader that U is generally a parameter of the test function. We usually write f instead of f_U, unless we find it necessary to emphasize the dependency on U. In this game, R can be taken to be $\{0, 1\}$.

The goal of the player is to use as few tests as possible, and as little computation as possible, to pick a set U' which is a "close" approximation to U. To motivate the group testing game, consider Dorfman's blood testing application explained in Section 1.1. We can express this application as an instance of the game. The set X comprises all blood samples, and the target set U comprises those blood samples that contain syphilitic antigen. A subset P of X is a positive pool if and only if P contains at least one blood sample showing presence of syphilitic antigen. The objective here is to identify exactly the set U.

There are a number of features of the simple group testing game that are essential to any model of group testing. We highlight them before delving into the general definitions.

- The goal of the group testing game is to identify an unknown target set. The target set is not arbitrary, but contains at most r elements of X.
- Finding a solution occurs through performing tests on pools.
- The solution supplied by the player must satisfy a specified criterion.
- We are interested in a player who is efficient: not many questions need to be asked to obtain the solution.

Our intent is to state a metamodel of group testing that shares and formalizes the properties listed above. We begin by developing and motivating the necessary definitions.

3.2 Definition of the Metamodel

Many models of group testing have been proposed (see [48]). Unfortunately, these different proposals seem rather ad hoc. It is possible, however, to describe them as derivations of a common metamodel. Treating them in this view provides us with a formal setting to discuss various results in group testing.

Let X be a finite set called the *object space*, and let r be a positive integer called the *a priori guarantee* of X. A *target set* of X is just a subset $U \subseteq X$ such that $|U| \leq r$. The *test function* employed is $f : 2^X \to R$. Our *group testing algorithm* (corresponding to the player of our simple group testing game) is an *XPRAM*, a model introduced by Valiant [148, 149] for parallel computation.

An XPRAM is a machine that consists of a number of processors, each one with a local memory. Each processor is a universal sequential machine with its local instruction set which can access words from the local memory. In addition, there is a set of global parallel instructions that allow accesses by processes to the memories of other processors. The main global instructions are reads and writes, which enable processors to simultaneously read from or write to places in the whole memory space. Each processor is assumed to execute its own, possibly unique, program. An XPRAM executes operations in steps. In each step, each processor may execute any number of local instructions, and access other memories using global reads and writes. The processors know whether or not a step is completed at the end of every period of a specified length. Within such a period, there is no synchronization among the processors.

In addition to the above defining properties of an XPRAM given by Valiant, we allow each processor to have access to an oracle. We call such a machine an *oracle XPRAM*. We think of the *oracle* as a procedure $\mathcal{O}(\cdot)$ that implements the test function $f : 2^X \to R$. When presented with a pool $P \subseteq X$, the oracle $\mathcal{O}$ performs a test on P, that is, it returns $f(P)$. Other than the oracle, the XPRAM knows, X, r, and the solution criterion, but has no other knowledge.

For formality, the *solution criterion* is specified as a predicate $\Pi(\varphi)$ containing a variable φ. The solution U' is said to satisfy the solution criterion if $\Pi(U')$ is true. The predicate $\Pi(\varphi)$ will, in general, involve U and possibly r.

Let us now take a moment to see how an oracle XPRAM with p processors, numbered one to p, plays the simple group testing game. The oracle in this case implements the function $f : 2^X \to \{0, 1\}$ such that $f(P) = 1$ if $|P \cap U| \geq 1$ and $f(P) = 0$ otherwise. The global communication pattern of the p processors is in the form of a *star network* (Figure

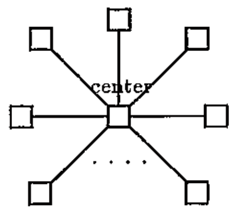


Figure 3.1: Star network.

3.1). Processor one, called the *center*, communicates with the other $p-1$ *leaf processors* via global reads and writes. The other $p-1$ processors do not communicate with each other. In each step, the processors work as follows.

Center: Depending on results from previous steps, either output a solution and halt, or compute p pools of X, $P_1, P_2, \ldots, P_p$. Using global writes, write P_i into the local memory of processor i, for $2 \leq i \leq p$. Now present P_1 to the oracle. Check local memory for results written by the leaf processors.

Leaf processor *i*: read local memory for P_i written by the center. Present P_i to the oracle. Using global write, write the answer supplied by the oracle to the local memory of the center.

Up to now, we have not mentioned how the complexity of the group testing algorithm is measured. Valiant [149] gives a formula for the running time of an XPRAM without oracles. In most applications of group testing, the cost of performing a test (calling the oracle) far exceeds the cost of gathering results of tests and distributing the tests to be performed (global reads and writes). Inferring the final solution or what subsequent pools to test based on the results of previous tests is not a simple process, but is usually not as tedious as performing the tests. Hence, it has been common practice in the group testing literature (see [48]) to assume that the most important factor is the number of tests performed. The time complexity of the inference procedure is of secondary importance. This is also the view we take throughout this dissertation. The *test complexity* of an oracle XPRAM is defined to be the total number of calls made to the oracles.

We are now ready to give a definition for the group testing metamodel.

Definition 3.2.1 (Group Testing Metamodel) Let X be a finite set with a priori guarantee r, $f : 2^X \to R$ be a test function, and $\Pi(\varphi)$ be a solution criterion. Let $\mathcal{O}$ be an oracle implementing f. For positive integers p and t, we say that (X, r, f, Π, p, s) is *t-group testable* if there exists a p-processor s-step group testing algorithm with access to $\mathcal{O}$ having test complexity at most t, that outputs a solution P' such that $\Pi(P')$ is true. The hextuple (X, r, f, Π, p, s) is called a *group testing problem.*

Given a group testing problem (X, r, f, Π, p, s), the objective is to determine the minimum t such that the problem is t-group testable. A dash "–" is used to indicate that a parameter is unconstrained. In the literature, a $(X, r, f, \Pi, 1, -)$ problem is called *sequential*, and a $(X, r, f, \Pi, -, 1)$ problem is called *nonadaptive*. Hence, in a sequential problem, the algorithm has only one processor and at each step, its query to the oracle can depend on the results of all previous queries. In contrast, an algorithm for a nonadaptive problem must specify all its queries in one step. Since the values of both p and s are implied, sequential and nonadaptive problems are simply defined by a quadruple (X, r, f, Π). Also, by a sequential or nonadaptive algorithm, we mean an algorithm for a sequential or nonadaptive problem, respectively.

3.3 Some Interesting Models

Various models of group testing can be derived from the metamodel (Definition 3.2.1) by specifying f and Π in different ways. Let us briefly discuss some of the most important types.

3.3.1 Test Functions

An n-ary test function is a function $f : 2^X \to R$, where $|R| = n$. The most popular models for group testing have binary or ternary test functions [48]. Among these, binary test functions form the majority. Nonadaptive problems with binary test functions also give rise to interesting Turán-type problems, which are the subject of this dissertation. We therefore restrict ourselves to binary test functions here and in subsequent chapters. Without loss of generality, we assume that the range of each test function is $\{0, 1\}$.

Another assumption we make concerns a particular property of tests. In all applications of group testing encountered, one can observe that adding an element not in the target set cannot change the test result of a pool from zero to one. For example, in Dorfman's blood testing application, the addition of a nonsyphilitic blood sample to a pool which shows no trace of syphilitic antigen cannot render the pool syphilitic. We formalize this property as follows.

Definition 3.3.1 Let X be a finite set and $U \subseteq X$. A function $f : 2^X \to \{0,1\}$ is *coherent* with respect to U if $f(\varnothing) = 0$ and whenever $P \subseteq X$ such that $f(P) = 0$, we have $f(P \cup \{x\}) = 0$ for all $x \in X \setminus U$.

Henceforth, we assume that the test functions for our group testing problems are all coherent with respect to the target sets.

The test function that is most frequently studied in group testing is

$$f(P) = \begin{cases} 1, & \text{if } |P \cap U| \geq 1; \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

A useful generalization of (3.1) is the *τ-threshold function*

$$f(P) = \begin{cases} 1, & \text{if } |P \cap U| \geq \tau; \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

To motivate the definition of this test function, let us refer back to Dorfman's blood testing application in Section 1.1. The precision of any test has a limit. Hence, a test cannot detect the presence of syphilitic antigen in a blood pool, unless the number of syphilitic blood samples it contains exceeds a certain threshold. The function in (3.2) models this situation. It is also possible that the precision of the test depends on the concentration of syphilitic antigen in the pool, and not on the minimum amount of antigen that would trigger the test. This scenario gives rise to the test function

$$f(P) = \begin{cases} 1, & \text{if } |P \cap U| \geq \gamma|P|; \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

Another test function with potential applications is the *MOD_m function*,

$$f(P) = \begin{cases} 1, & \text{if } |P \cap U| \equiv 1 \pmod{m}; \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

Suppose that a factory produces *inverters* (Figure 3.2), which may suffer from faults. Instead of inverting its input, a *faulty inverter* simply outputs its input. The factory would like to sieve out those faulty inverters before putting its inverters out on the market. A possible group test for inverters is to construct pools, each consisting of several inverters connected in series. A test comprises inputing a bit to a pool and observing its output. It is not hard to verify that a pool produces an incorrect output if and only if the pool

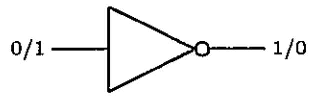


Figure 3.2: An inverter.

contains an odd number of faulty inverters. The group test just described corresponds to a MOD_2 test function.

The three test functions (3.2), (3.3), and (3.4) we introduce do not appear to have been studied before.

3.3.2 Solution Criteria

The most important solution criterion is the *exact identification* criterion:

$$\Pi(\varphi) \quad \equiv \quad (\varphi = U). \tag{3.5}$$

In some situations, it may be sufficient to find a reasonably small set containing U. This leads to the following *α-approximate identification* criterion:

$$\Pi(\varphi) \quad \equiv \quad (\varphi \supseteq U \quad \text{and} \quad |\varphi| \leq \alpha r), \tag{3.6}$$

where r is the a priori guarantee.

3.3.3 Restrictions

Further restrictions are often imposed on group testing algorithms. Typical restrictions are:

- There is a given set L, and for every pool P, we must have $|P| \in L$.

- For each $x \in X$, the number of tests involving x must not exceed some number k.

The first restriction is motivated by applications in which the test kit is manufactured only with sizes in L. The second restriction is appropriate in situations when there is insufficient material for test, or when the quality of the object degrades with each test. For example, in Dorfman's blood testing application, each blood sample may be enough for only k tests. In some applications, it is also desirable that the number of tests involving each $x \in X$ is constant. We call a group testing problem *k-restricted* if each $x \in X$ is involved in exactly k tests.

3.4 A List of Group Testing Problems

We end this chapter with a list of group testing problems that are of interest in this dissertation. Each problem is parametrized by the a priori guarantee r which appears at the end of the name of the problem in parentheses. All the problems are nonadaptive. We therefore only specify the test function, solution criterion, and restrictions (if any) for each problem.

UNRESTRICTED NONADAPTIVE EXACT IDENTIFICATION PROBLEM(r)

TEST FUNCTION: 1-threshold function.

SOLUTION CRITERION: $(\varphi = U)$.

k-RESTRICTED NONADAPTIVE EXACT IDENTIFICATION PROBLEM(r)

TEST FUNCTION: 1-threshold function.

SOLUTION CRITERION: $(\varphi = U)$.

RESTRICTIONS: Every object is tested exactly k times.

k-Restricted Nonadaptive α-Approximate Identification Problem(r)

TEST FUNCTION: 1-threshold function.

SOLUTION CRITERION: $(\varphi \supseteq U \quad \text{and} \quad |\varphi| \leq \alpha r)$.

RESTRICTIONS: Every object is tested exactly k times.

k-Restricted Nonadaptive Exact Identification Parity Problem(r)

TEST FUNCTION: MOD_2 function.

SOLUTION CRITERION: $(\varphi = U)$.

RESTRICTIONS: Every object is tested exactly k times.

CHAPTER 4

Nonadaptive Group Testing and Turán-Type Problems

4.1 Why Nonadaptive Group Testing?

The group testing problems that are the primary concern of this dissertation are nonadaptive. It is clear that the best sequential algorithm for a group testing problem (X, r, f, Π) must perform at least as well as any nonadaptive algorithm for (X, r, f, Π), since a one-processor oracle XPRAM can simulate a p-processor oracle XPRAM without increasing the test complexity. It is therefore not surprising that most past research efforts on group testing had focused on sequential problems. Moreover, machines with many processors were not a reality until relatively recently.

The advent of massively parallel computers have prompted Hwang and Sós [81] to make a case for the study of nonadaptive group testing problems. Further support of this case is given by Knill and Muthukrishnan [85], who observed that the following three features in the screening of clone libraries with hybridization probes strongly encourage nonadaptive algorithms:

- A set X of clones is screened many times. Each time the concept of positive clones in X is different. The aim is to identify the positive clones, with respect to the different concepts. During each screening, a different probe suited to a particular concept of positivity, is used to test pools of clones.

- It is expensive to prepare a pool for testing the first time. Once a pool is prepared, however, it can be tested many times with different probes.

- Testing one pool at a time is expensive but testing many pools in parallel with the same probe is much cheaper per pool.

An example of a real-life screening effort can be found in [25].

In addition to the above biological application, nonadaptive group testing is also closely related to the theory of *superimposed codes*. Superimposed codes were first studied by Kautz and Singleton [83]. These codes have applications in information retrieval [83] and multiple access communication [11]. We refer the interested reader to [51, 52] for more information.

4.2 The Role of Turán-Type Problems

A *Turán-type problem* takes the following form. Given a family $\mathcal{F}$ of configurations, determine the maximum number of blocks in a set system of order n that avoids all the configurations in $\mathcal{F}$. This problem is so-named in memory of Turán, who proved one of the most important results in the area [145, 146]. A recent survey of Füredi [65] provides a good summary of work in the area. In this section, we explain the role of Turán-type problems in nonadaptive group testing.

Let $\mathcal{A}$ be any nonadaptive algorithm for a group testing problem. The only essential factor of $\mathcal{A}$ is whether the pools it tests would yield enough information for the determination of a solution. Hence, we may consider a nonadaptive algorithm $\mathcal{A}$ for a group testing problem (X, r, f, Π) as being completely specified by a set system $\mathcal{S} = (X, \mathcal{P})$, where $\mathcal{P}$ is the set of all pools tested by $\mathcal{A}$. The test complexity of $\mathcal{A}$ is then given by $t = |\mathcal{P}|$. $\mathcal{S}$ is called the *primal system associated with* $\mathcal{A}$. It has been more natural to consider the *dual* of $\mathcal{S}$, which is constructed as follows. Let $P_1, P_2, \ldots, P_t$ be the blocks in $\mathcal{P}$. For each $x \in X$, let $B_x = \{i \mid x \in P_i\}$. The *dual* of $\mathcal{S} = (X, \mathcal{P})$, denoted $\mathcal{S}^*$, is the set system $(Y, \mathcal{B})$, where $Y = \{1, 2, \ldots, t\}$ and $\mathcal{B} = \{B_x \mid x \in X\}$. A block $B_x \in \mathcal{B}$ contains exactly all those pools in which x is involved. We call $(Y, \mathcal{B})$ the *dual system associated with* $\mathcal{A}$. Since $(\mathcal{S}^*)^*$ is isomorphic to $\mathcal{S}$ for any set system $\mathcal{S}$, a nonadaptive algorithm is also determined by its dual system.

In the following subsections, we define for any test function f, set system $(X, \mathcal{P})$, and $U \subseteq X$, the set

$$f_{\mathcal{P}}^{+}(U) = \{P \in \mathcal{P} \mid f_U(P) = 1\}.$$

4.2.1 Exact Identification Problems

Let (X, f, r, Π) be a nonadaptive group testing problem with the exact identification criterion. Let U and U' be two distinct subsets of X, each containing at most r elements. A necessary and sufficient condition for $(X, \mathcal{P})$ to be the primal system of a nonadaptive algorithm for an exact identification problem is given in the following lemma.

Lemma 4.2.1 Let (X, r, f, Π) be a group testing problem with the exact identification criterion (3.5). A set system $(X, \mathcal{P})$ is the primal system of a nonadaptive algorithm for (X, r, f, Π) if and only if the following condition holds. For any two distinct subsets U and U' of X, each containing at most r elements, we have $f_{\mathcal{P}}^{+}(U) \neq f_{\mathcal{P}}^{+}(U')$.

Proof. Necessity is easy. If $f_{\mathcal{P}}^{+}(U) = f_{\mathcal{P}}^{+}(U')$, then the algorithm cannot decide whether U or U' is the target set, since the tests yield the same results regardless of whether U or U' is the target set.

To prove sufficiency, let $S = \{P \in \mathcal{P} \mid \text{the result of the test on pool } P \text{ is } 1\}$. Since $f_{\mathcal{P}}^{+}(U) \neq f_{\mathcal{P}}^{+}(U')$, there exists precisely one subset $U \subseteq X$, $|U| \leq r$, such that $f_{\mathcal{P}}^{+}(U) = S$. This subset is the required target set. □

The condition in Lemma 4.2.1 can be translated into a condition for the dual system by observing that $P_i = \{x \in X \mid i \in B_x\}$.

Lemma 4.2.2 Let (X, r, f, Π) be a group testing problem with the exact identification criterion (3.5). A set system $(Y, \mathcal{B})$ is the dual system of a nonadaptive algorithm for (X, r, f, Π) if and only if the following condition holds. For any two distinct subsets U and U' of X, each containing at most r elements, we have

$$\{i \in Y \mid f_U(\{x \in X \mid i \in B_x\}) = 1\} \neq \{i \in Y \mid f_{U'}(\{x \in X \mid i \in B_x\}) = 1\}. \tag{4.1}$$

In a nonadaptive group testing problem, $|\mathcal{B}|$, the number of objects in X, is usually fixed and we want to minimize the algorithm's test complexity $t = |Y|$. Equivalently, we can keep the test complexity of the algorithm fixed and try to maximize the number of objects for which we can still solve the problem. This latter view defines a Turán-type problem. The condition (4.1) dictates which configurations are to be avoided in the dual system. We illustrate this with the UNRESTRICTED NONADAPTIVE EXACT IDENTIFICATION PROBLEM(r).

Definition 4.2.1 A set system $(X, \mathcal{A})$ is *r-union-free* if there do not exist $A_1, A_2, \ldots, A_r, B_1, B_2, \ldots, B_r \in \mathcal{A}$, not necessarily distinct, such that

$$\bigcup_{i=1}^{r} A_i = \bigcup_{i=1}^{r} B_i,$$

unless $\{A_1, A_2, \ldots, A_r\} = \{B_1, B_2, \ldots, B_r\}$.

Lemma 4.2.2 gives the following.

Corollary 4.2.1 (Hwang and Sós [81]) Solving the UNRESTRICTED NONADAPTIVE EXACT IDENTIFICATION PROBLEM(r) is equivalent to determining the maximum number of blocks in an r-union-free set system of order n.

Proof. For f the 1-threshold function, (4.1) reduces to

$$\bigcup_{x \in U} B_x \neq \bigcup_{x \in U'} B_x \tag{4.2}$$

for the dual system $(Y, \mathcal{B})$, where U and U' are distinct subsets of X, each containing at most r elements. When $|U| = |U'| = r$, (4.2) is exactly the condition for the dual system being r-union-free. So suppose the dual system is r-union-free and $\bigcup_{x \in U} B_x = \bigcup_{x \in U'} B_x$ for some U and U' distinct subsets of X, each containing at most r elements. Now increase the multiplicity of any $x \in U$ and any $x' \in U'$ until $|U| = |U'| = r$. It is obvious that $\bigcup_{x \in U} B_x = \bigcup_{x \in U'} B_x$. But this contradicts the assumption that the dual system is r-union-free. □

From Corollary 4.2.1, the result below follows easily.

Corollary 4.2.2 Solving the k-RESTRICTED NONADAPTIVE EXACT IDENTIFICATION PROBLEM(r) is equivalent to determining the maximum number of blocks in an r-union-free

k-uniform set system of order n.

Let $u(n, r)$ and $u(n, k, r)$ denote the maximum number of blocks in an r-union-free set system and r-union-free k-uniform set system, respectively. Not many results concerning these two functions are known, other than those implied by r-cover-free set systems (see Section 4.3). In general, it is known (see [48]) that

$$\binom{u(n,r)}{r} \leq \sum_{j=r}^{n} \binom{n}{j}.$$

The problem of estimating $u(n, 2)$ was raised by Erdös and Moser [61]. Frankl and Füredi [63] proved

$$2^{(n-3)/4} \leq u(n,2) \leq 2^{(n+1)/2} + 1.$$

A 2-union-free 2-uniform set system is a graph without cycles of length three and cycles of length four. Reiman [118] proved

$$\frac{1}{2\sqrt{2}} n^{3/2} < u(n,2,2) < \frac{1}{2} n^{3/2}.$$

Erdös [55] has made the conjecture that

$$u(n,2,2) = \frac{1+o(1)}{2\sqrt{2}} n^{3/2}.$$

Surprisingly, the case $k = 3$ is easier. Frankl and Füredi [62] established the following exact result:

$$u(n,3,2) = \left\lfloor \frac{n(n-1)}{6} \right\rfloor. \tag{4.3}$$

For general k, employing symmetric functions over finite fields, Frankl and Füredi [64] obtained the result below.

Theorem 4.2.1 (Frankl and Füredi [64]) For any fixed k, there exist positive constants a_1 and a_2 such that

$$a_1 n^{\lceil 4k/3 \rceil/2} \leq u(n, k, 2) \leq a_2 n^{\lceil 4k/3 \rceil/2}.$$

4.2.2 α-Approximate Identification Problems

Let (X, r, f, Π) be a nonadaptive group testing problem with the α-approximate identification criterion. We have the following analogue of Lemma 4.2.1.

Lemma 4.2.3 Let (X, r, f, Π) be a group testing problem with the α-approximate identification criterion (3.6). A set system $(X, \mathcal{P})$ is the primal system of a nonadaptive algorithm for (X, r, f, Π) if and only if the following condition holds. There exists a subset $Z \subseteq X$, $|Z| \leq \alpha r$, such that for any two distinct subsets U and U' of X, each containing at most r elements, we have either $f_{\mathcal{P}}^{+}(U) \neq f_{\mathcal{P}}^{+}(U')$ or $U \cup U' \subseteq Z$.

Proof. First we prove necessity. Suppose there does not exist such a subset Z and $f_{\mathcal{P}}^{+}(U) = f_{\mathcal{P}}^{+}(U')$. Then, one of the following two situations must occur:

(i) $|U \cup U'| > \alpha r$;

(ii) there exists $U'' \subseteq \mathcal{P}$ such that $f_{\mathcal{P}}^{+}(U) = f_{\mathcal{P}}^{+}(U'')$ and there is no set of size αr containing both $U \cup U'$ and $U \cup U''$.

If $|U \cup U'| > \alpha r$, then no set of size at most αr can contain both U and U'. So the solution obtained by the algorithm can contain at most one of U and U'. We can obtain a contradiction by taking the target set to be the one that is not contained in the solution given by the algorithm. For the second situation, the algorithm cannot obtain a solution that contains both $U \cup U'$ and $U \cup U''$. If the solution contains $U \cup U'$, we let the target set be U''. If the solution contains $U \cup U''$, we let the target set be U'. In both cases we have a contradiction.

To prove sufficiency, let $S = \{P \in \mathcal{P} \mid \text{the result of the test on pool } P \text{ is } 1\}$. Then there exists a subset $Z \subseteq X$, $|Z| \leq \alpha r$, such that for all $U \subseteq X$, $|U| \leq r$, and $f^+_{\mathcal{P}}(U) = S$, we have $U \subseteq Z$. This subset Z is the required solution since it contains the target set and has size at most αr. □

In Chapter 6, we shall see that Lemma 4.2.3 gives rise to a Turán-type problem for the 3-RESTRICTED NONADAPTIVE (3/2)-APPROXIMATE IDENTIFICATION PROBLEM(2).

4.3 r-Cover-Free Set Systems

The proof of Corollary 4.2.1 shows that every nonadaptive algorithm based on an r-union-free set system is able to solve the (UNRESTRICTED or k-RESTRICTED) NONADAPTIVE EXACT IDENTIFICATION PROBLEM(r). How does one actually go about finding the solution after obtaining the results for the set of pools $\mathcal{P}$? This is the task of the *inference procedure.* One approach is to build a table of $f^+_{\mathcal{P}}(U)$, for all $U \subseteq X$, $|U| \leq r$. The table can be sorted with respect to $f^+_{\mathcal{P}}(U)$ so that a search for the target set can be done in $O(\log |X|)$ time, for constant r. However, the time and space required to build and store the table is $\Omega(|X|^r)$. The tabulation of $f^+_{\mathcal{P}}(U)$ can thus be a serious bottleneck in terms of both space and time for some applications when r is large, even though once built, the table can be used over and over again. It is not known whether there is a more efficient inference procedure of determining the target set for nonadaptive algorithms based on r-union-free set systems [48].

Another problem commonly encountered in practical applications is that the a priori guarantee is often an estimate and hence can be wrong at times. Ideally, we would like to be able to tell as soon as possible whether the a priori guarantee given is correct. An inference procedure for a nonadaptive algorithm based on an r-union-free set system only reveals the correctness of the a priori guarantee after having succeeded or failed the search in the table of $f_{\mathcal{P}}^{+}(U)$. This is often undesirable.

The two problems discussed above can be alleviated by nonadaptive algorithms based on set systems that satisfy a stronger property.

Definition 4.3.1 A set system $(X, \mathcal{A})$ is *r-cover-free* if there do not exist $A_0, A_1, \ldots, A_r \in \mathcal{A}$, not necessarily distinct, such that

$$A_0 \subseteq \bigcup_{i=1}^{r} A_i,$$

unless $A_0 \in \{A_1, A_2, \ldots, A_r\}$.

Let $\mathcal{A}$ be a nonadaptive algorithm whose dual system $(Y, \mathcal{B})$ is r-cover-free. Since every r-cover-free set system is also r-union-free, $\mathcal{A}$ solves the (UNRESTRICTED or k-RESTRICTED) NONADAPTIVE EXACT IDENTIFICATION PROBLEM(r) (X, r, f, Π).

Lemma 4.3.1 Let U be the target set and $x \in X$. Then U contains x if and only if x does not appear in any pool $P \subseteq X$ such that $f(P) = 0$.

Proof. It is obvious that $x \in U$ implies that x cannot appear in any pool P such that $f(P) = 0$.

So assume that every pool P containing x is such that $f(P) = 1$. Suppose to the

all $x \in X$ are not marked;
for $i = 1$ **to** t **do**
    if $f(P_i) = 0$ **then**
        mark x **for all** $x \in P_i$;
for $x \in X$ **do**
    if x is not marked **then**
        $x \in U$;

Figure 4.1: Inference procedure.

contrary that $x \notin U$. For all $i \in B_x$, we have $f(P_i) = 1$. But

$$\{i \in Y \mid f(P_i) = 1\} = \bigcup_{u \in U} B_u,$$

where U is the target set. Hence

$$B_x \subseteq \bigcup_{u \in U} B_u.$$

This contradicts the fact that $(Y, \mathcal{B})$ is r-cover-free, since $x \notin U$ and $|U| \leq r$. □

An inference procedure based on Lemma 4.3.1 can be developed to find the target set given the test results. This is given in Figure 4.1. This procedure is more efficient than the search table technique used for nonadaptive algorithms based on r-union-free set systems. The time complexity of the procedure is easily bounded by $O(t|X|)$. It is known [58, 81] that there exist r-cover-free set systems for which t is as small as $O(\log |X|)$ for unrestricted problems and $O(|X|^{1/k})$ for k-restricted problems. This gives a marked improvement over the $\Omega(|X|^r)$ time procedure employed by nonadaptive algorithms based on r-union-free set systems. If at the end of the procedure above, we have more than r objects in U, then we can also conclude that the a priori guarantee is wrong. This observation is first made by Schultz [129]. Since the property of being r-cover-free is stronger than the property of being r-union-free, the tradeoff here is between the test complexity of the nonadaptive algorithm and the time complexity of the inference procedure.

This advantage of nonadaptive algorithms based on r-cover-free set systems has encouraged the study of r-cover-free set systems over r-union-free set systems in the group testing literature. The next chapter in this dissertation presents some new results on r-cover-free set systems.

CHAPTER 5

Characterizations and Improved Bounds for r-Cover-Free Set Systems

5.1 Preliminaries

Let $c(n, k, r)$ denote the maximum number of blocks in an r-cover-free k-uniform set system of order n. Any r-cover-free k-uniform set system of order n having $c(n, k, r)$ blocks is said to be *optimal.*

If an r-cover-free k-uniform set system has at least $r + 1$ blocks, then each block in a collection of $r + 1$ blocks must contain a point that is contained in no other blocks of the collection. It follows that there are at least $(k - 1) + (r + 1) = k + r$ points. Hence, we assume throughout this chapter that $n > k + r$, since we have

$$c(n,k,r) = \begin{cases} 0, & \text{if } n < k; \\ \min\left\{\binom{n}{k}, r\right\}, & \text{if } k \le n \le k + r - 1; \\ r + 1, & \text{if } n = k + r. \end{cases} \tag{5.1}$$

The problem of determining $c(n, k, r)$ was introduced by Kautz and Singleton [83], and studied extensively by Erdös, Frankl, and Füredi [57, 58]. To state their main result, and also our results in subsequent sections, we require the following definitions from design theory.

Definition 5.1.1 A t-(v, k, λ) *design* is a k-uniform set system, $(X, \mathcal{B})$, of order v, such that every t-subset of X is contained in precisely λ of the blocks in $\mathcal{B}$.

Definition 5.1.2 A t-(v, k, λ) *packing* is a k-uniform set system, $(X, \mathcal{B})$, of order v, such that every t-subset of X is contained in at most λ of the blocks in $\mathcal{B}$.

The *packing number* $D_\lambda(v, k, t)$ is the maximum number of blocks in any t-(v, k, λ) packing. A t-(v, k, λ) packing, $(X, \mathcal{B})$, is *optimal* if $|\mathcal{B}| = D_\lambda(v, k, t)$. If $\lambda = 1$, often one writes $D(v, k, t)$ for $D_1(v, k, t)$.

Packings play an important role in the study of r-cover-free set systems because of the following simple observation.

Lemma 5.1.1 (Kautz and Singleton [83]) A t-$(n, k, 1)$ packing is an r-cover-free set system of order n if $k \geq r(t-1) + 1$.

Proof. Consider any block B in the packing. At most $r(t-1)$ of the points in B can be contained in the union of r other blocks. But $k \geq r(t-1) + 1$. Hence no block is contained in the union of r others. □

Definition 5.1.3 A Δ-*system with nucleus* A is a set system $(X, \mathcal{B})$ in which $B \cap B' = A$ for all $B, B' \in \mathcal{B}$, $B \neq B'$. The sets $B \setminus A$, for all $B \in \mathcal{B}$, are called the *rays* of the Δ-system.

In [58], Erdös, Frankl, and Füredi established the following result.

Theorem 5.1.1 (Erdős, Frankl, and Füredi [58]) Let $k = r(t-1)+1+d$, where $0 \le d < r$. There exists an integer $n_0(k)$ such that for all $n > n_0(k)$, we have

$$(1-o(1))\binom{n-d}{t}\Big/\binom{k-d}{t} \le c(n,k,r) \le \binom{n-d}{t}\Big/\binom{k-d}{t} \tag{5.2}$$

whenever any one of the following conditions holds:

(i) $d = 0$ or 1,

(ii) $d < r/2t^2$,

(iii) $t = 2$ and $d < \lceil 2r/3 \rceil$.

Moreover, equality holds in (5.2) if and only if a t-$(n-d, k-d, 1)$ design exists.

Apart from the characterization of r-cover-free k-uniform set systems meeting the upper bound in (5.2) in terms of t-$(n-d, k-d, 1)$ designs, no other exact behaviour of $c(n,k,r)$ is known. The purpose of this chapter is to determine exactly the function $c(n,k,r)$ for some values of r and k, and to characterize their associated set systems. New upper bounds, improving that of Theorem 5.1.1, are also obtained.

We begin in the next section with the instructive case of 2-cover-free 3-uniform set systems.

5.2 2-Cover-Free Triple Systems

The asymptotic behaviour of $c(n,3,2)$ was determined by Erdős, Frankl, and Füredi in [57]:

$$c(n,3,2) = \frac{1}{6}n^2 - O(n).$$

Here, we determine $c(n,3,2)$ exactly for all n. More specifically, we show that $c(n,3,2) = D(n,3,2)$ for all $n \geq 6$. The function $D(n,3,2)$ has been completely determined by Schönheim [127] and Spencer [136] who proved

$$D(n,3,2) = \begin{cases} U(n,3,2) - 1, & \text{if } n \equiv 5 \pmod 6; \\ U(n,3,2), & \text{otherwise,} \end{cases}$$

where

$$U(v,k,t) = \left\lfloor \frac{v}{k} \left\lfloor \frac{v-1}{k-1} \cdots \left\lfloor \frac{v-t+1}{k-t+1} \right\rfloor \cdots \right\rfloor \right\rfloor .$$

The idea behind our determination of $c(n,3,2)$ is based on the observation that every Δ-system with a nucleus of size two in a 2-cover-free 3-uniform set system precludes all of the points in its rays from appearing in any other blocks. Hence, we cannot have too many Δ-systems with a nucleus of size two in a 2-cover-free 3-uniform set system. We can also delete all Δ-systems with a nucleus of size two to obtain a 2-$(v,3,1)$ packing. The number of blocks in this packing can be bounded. The details are as follows.

Theorem 5.2.1 For all $n \geq 6$, $c(n,3,2) = D(n,3,2)$.

Proof. First note that any 2-$(n,3,1)$ packing is 2-cover-free (Lemma 5.1.1). This shows that $c(n,3,2) \geq D(n,3,2)$.

Now let $(X, \mathcal{B})$ be any 2-cover-free 3-uniform set system of order n. For $A \in \binom{X}{2}$, define

$$\mathcal{B}(A) = \{B \in \mathcal{B} \mid B \supset A\} \quad \text{and} \quad T(A) = \{x \in X \mid A \cup \{x\} \in \mathcal{B}\}.$$

Note that there is a bijection between $\mathcal{B}(A)$ and $T(A)$:

$$x \in T(A) \iff A \cup \{x\} \in \mathcal{B}(A).$$

Further, define

$$G_i = \left\{ A \in \binom{X}{2} \,\middle|\, |T(A)| = i \right\}, \quad \text{for} \quad 0 \leq i \leq n-2.$$

Let $g_i = |G_i|$. Clearly, $\{G_0, G_1, \ldots, G_{n-2}\}$ is a partition of $\binom{X}{2}$. Let $G_{\geq 2} = \bigcup_{i=2}^{n-2} G_i$. Observe that if $A \in G_{\geq 2}$, then $T(A)$ contains points, each of which appears in only one block of $\mathcal{B}$; for otherwise there would exist $x \in T(A)$ that is contained in the block $A \cup \{x\}$ and some other block $B \in \mathcal{B}$, and we can take a point $y \in T(A)$ different from x (this is possible because $|T(A)| \geq 2$) to obtain $(A \cup \{x\}) \subseteq (A \cup \{y\}) \cup B$, hence contradicting the assumption that $(X, \mathcal{B})$ is 2-cover-free. This observation implies that

$$T(A) \cap T(A') = \varnothing \quad \text{and} \quad \mathcal{B}(A) \cap \mathcal{B}(A') = \varnothing, \quad \text{for any} \quad A, A' \in G_{\geq 2},\ A \neq A'. \tag{5.3}$$

Let

$$\mathcal{B}' = \mathcal{B} \setminus \left(\bigcup_{A \in G_{\geq 2}} \mathcal{B}(A) \right) \quad \text{and} \quad X' = X \setminus \left(\bigcup_{A \in G_{\geq 2}} T(A) \right).$$

We claim that $(X', \mathcal{B}')$ is a set system. Suppose not. Then there exists $B \in \mathcal{B}'$ such that $B \not\subseteq X'$. Hence, B contains a point $x \in T(A)$, for some $A \in G_{\geq 2}$. It follows that $A \cup \{x\}$ is a block of $\mathcal{B}$ and there exists also another point y such that $A \cup \{y\}$ is a block of $\mathcal{B}$. But then $(A \cup \{x\}) \subseteq (A \cup \{y\}) \cup B$, contradicting the assumption that $(X, \mathcal{B})$ is 2-cover-free. It is also easy to see that for any two distinct blocks $B, B' \in \mathcal{B}'$, we have $|B \cap B'| \leq 1$; for otherwise there would exist $A \in \binom{X}{2}$ such that $A \subset B$, $A \subset B'$, and hence $B, B' \in \bigcup_{A \in G_{\geq 2}} \mathcal{B}(A)$. We conclude, therefore, that $(X', \mathcal{B}')$ is a 2-$(|X'|, 3, 1)$ packing. Also, (5.3) allows us to compute the size of X':

$$|X'| = |X| - \left| \bigcup_{A \in G_{\geq 2}} T(A) \right| = |X| - \sum_{i=2}^{n-2} i g_i.$$

Similarly, we have

$$|\mathcal{B}'| = |\mathcal{B}| - \sum_{i=2}^{n-2} i g_i.$$

It follows that

$$\begin{aligned} |\mathcal{B}| &= |\mathcal{B}'| + \sum_{i=2}^{n-2} i g_i \\ &= D\left(n - \sum_{i=2}^{n-2} i g_i, 3, 2\right) + \sum_{i=2}^{n-2} i g_i \\ &\leq D(n,3,2) \quad \text{for all} \quad n \geq 6. \end{aligned} \tag{5.4}$$

This completes the proof that $c(n,3,2) = D(n,3,2)$. □

The proof of Theorem 5.2.1 gives a characterization of optimal 2-cover-free 3-uniform set systems. If $n \geq 7$, then for equality to hold in (5.4), we must have $\sum_{i=2}^{n-2} i g_i = 0$. This is possible only if $g_2 = g_3 = \cdots = g_{n-2} = 0$. Hence, every 2-subset of X appears in at most one block of $\mathcal{B}$. Consequently, we have the following result.

Corollary 5.2.1 For $n \geq 7$, a 2-cover-free 3-uniform set system of order n is optimal if and only if it is an optimal 2-$(n, 3, 1)$ packing.

This innocent-looking proof of Theorem 5.2.1 has two useful generalizations that yield results on $c(n, r+1, r)$ and $c(n, 2t-1, 2)$ that are stronger than any presently known. We discuss these next.

5.3 r-Cover-Free $(r+1)$-Uniform Set Systems

Our goal in this section is to obtain a result similar to Theorem 5.2.1 for r-cover-free $(r+1)$-uniform set systems. To this end, we generalize the idea behind the proof of Theorem 5.2.1 as follows. For every Δ-system with a nucleus of size two appearing in an r-cover-free $(r+1)$-uniform set system, we show that there is a *sufficiently large* number of points in the rays that do not appear in any other blocks. We can then follow the same bounding technique in the proof of Theorem 5.2.1.

First, Lemma 5.1.1 implies that any 2-$(n, r+1, 1)$ packing is r-cover-free. Hence $c(n, r+1, r) \geq D(n, r+1, 2)$.

Now, let $(X, \mathcal{B})$ be any r-cover-free $(r+1)$-uniform set system of order n. For $A \in \binom{X}{2}$, define

$$\mathcal{B}(A) = \{B \in \mathcal{B} \mid B \supset A\} \quad \text{and} \quad T(A) = \left\{F \in \binom{X}{r-1} \,\middle|\, A \cup F \in \mathcal{B}\right\}.$$

Note again the existence of a bijection between $\mathcal{B}(A)$ and $T(A)$. Further define

$$G_i = \left\{A \in \binom{X}{2} \,\middle|\, |T(A)| = i\right\}, \quad \text{for} \quad 0 \leq i \leq n-2.$$

Let $g_i = |G_i|$ and $G_{\geq 2} = \bigcup_{i=2}^{n-2} G_i$.

Now suppose $A \in G_{\geq 2}$. If $F \in T(A)$, then at least one point of F is not contained in any block of $\mathcal{B}$ other than $A \cup F$; for otherwise we can find $r-1$ blocks of $\mathcal{B}$ whose union contains F. These $r-1$ blocks together with the block $A \cup F'$ for some $F' \in T(A)$ different from F then contain the block $A \cup F$, contradicting the assumption that $(X, \mathcal{B})$ is r-cover-free.

For each $F \in T(A)$, define $S_A(F)$ to be the subset of points in F, each of which is contained in no blocks of $\mathcal{B}$ other than $A \cup F$.

Lemma 5.3.1 For any k distinct $(r-1)$-subsets $F_1, F_2, \ldots, F_k \in T(A)$, we have

$$|S_A(F_1) \cup S_A(F_2) \cup \cdots \cup S_A(F_k)| \geq k.$$

Proof. The proof is by induction on k. The case $k = 1$ follows easily from our observation before that every block $A \cup F$, $F \in T(A)$, must contain a point that is contained in no other block of $\mathcal{B}$.

Now consider $S_A(F_1)$, $S_A(F_2)$, $\ldots$, $S_A(F_k)$ for k distinct $(r-1)$-subsets F_1, F_2, $\ldots$, $F_k \in T(A)$. By the induction hypothesis, $S = S_A(F_1) \cup S_A(F_2) \cup \cdots \cup S_A(F_{k-1})$ contains at least $k-1$ points. We claim that $S_A(F_k)$ contains a point not in S. Assume the contrary. Then every point of $S_A(F_k)$ is contained in S, and hence is contained in the union of at most $|S_A(F_k)|$ of the $(r-1)$-subsets $F_1, F_2, \ldots, F_k$. Clearly, the points in $F_k \setminus S_A(F_k)$ are contained in the union of at most $r-1-|S_A(F_k)|$ blocks from $\mathcal{B}$. Hence, the block $A \cup F_k$ is contained in the union of at most $r-1$ blocks of $\mathcal{B}$, a contradiction. Therefore, $|S \cup S_A(F_k)| \geq k$. □

Corollary 5.3.1 There are at least $|T(A)|$ points in $\bigcup_{F \in T(A)} S_A(F)$.

From the definition of $S_A(F)$, it also follows that

$$\left(\bigcup_{F \in T(A)} S_A(F) \right) \cap \left(\bigcup_{F \in T(A')} S_{A'}(F) \right) = \varnothing, \tag{5.5}$$

if $A \neq A'$.

Let

$$\mathcal{B}' = \mathcal{B} \setminus \left(\bigcup_{A \in G_{\geq 2}} \mathcal{B}(A) \right)$$

and

$$X' = X \setminus \left(\bigcup_{A \in G_{\geq 2}} \bigcup_{F \in T(A)} S_A(F) \right).$$

First we show that $(X', \mathcal{B}')$ is a set system. Suppose not. Then there exists a block $B \in \mathcal{B}'$ such that $B \not\subseteq X'$. Hence, B contains a point $x \in S_A(F)$, for some $A \in G_{\geq 2}$ and $F \in T(A)$. By definition of $S_A(F)$, x is contained in no blocks other than $A \cup F$. So we must have $B = A \cup F$. This is a contradiction since $A \in G_{\geq 2}$, and $\mathcal{B}'$ contains no blocks of $\bigcup_{A \in G_{\geq 2}} \mathcal{B}(A)$. Next, for any two distinct blocks B and B' in $\mathcal{B}'$, we have $|B \cap B'| \leq 1$; for otherwise there would exists $A \in \binom{X}{2}$ such that $A \subset B$, $A \subset B'$, and hence $B, B' \in \bigcup_{A \in G_{\geq 2}} \mathcal{B}(A)$. Consequently, $(X', \mathcal{B}')$ is a 2-$(|X'|, r+1, 1)$ packing.

Since

$$\begin{aligned} \left| \bigcup_{A \in G_{\geq 2}} \mathcal{B}(A) \right| &\leq \sum_{A \in G_{\geq 2}} |\mathcal{B}(A)| \\ &= \sum_{A \in G_{\geq 2}} |T(A)| \qquad \text{(by the bijection between } \mathcal{B}(A) \text{ and } T(A)) \\ &= \sum_{i=2}^{n-2} i g_i, \end{aligned}$$

and

$$\begin{aligned}\left|X \setminus \left(\bigcup_{A\in G_{\geq 2}} \bigcup_{F\in T(A)} S_A(F)\right)\right| &= |X| - \left|\bigcup_{A\in G_{\geq 2}} \bigcup_{F\in T(A)} S_A(F)\right| \\ &= n - \sum_{A\in G_{\geq 2}} \left|\bigcup_{F\in T(A)} S_A(F)\right| \qquad \text{(by (5.5))} \\ &\leq n - \sum_{A\in G_{\geq 2}} |T(A)| \qquad \text{(by Corollary 5.3.1)} \\ &= n - \sum_{i=2}^{n-2} i g_i,\end{aligned}$$

we have

$$\begin{aligned}|\mathcal{B}| &= |\mathcal{B}'| + \left|\bigcup_{A\in G_{\geq 2}} \mathcal{B}(A)\right| \\ &\leq D\left(n - \sum_{i=2}^{n-2} i g_i, r+1, 2\right) + \sum_{i=2}^{n-2} i g_i. \end{aligned} \tag{5.6}$$

Let $\gamma = \sum_{i=2}^{n-2} i g_i$ and define, for fixed n and r, the function

$$\Phi(\gamma) = \binom{n-\gamma}{2} \Big/ \binom{r+1}{2} + \gamma.$$

It is easy to see that Φ is a convex function. Hence, the maximum of Φ over any closed interval occurs at one of its boundary points. Since

$$\Phi(2) = \frac{n^2 - 5n + 2(r^2 + r + 3)}{3(r+1)} \quad \text{and} \quad \Phi(n) = n,$$

we have $\Phi(2) \geq \Phi(n)$ if and only if $n \geq r^2+r+3$. In particular, this shows that

$$\arg\max\{\Phi(\gamma) \mid \gamma \in [2,n]\} = \begin{cases} n, & \text{if } n \leq r^2+r+2; \\ 2, & \text{if } n \geq r^2+r+3. \end{cases} \tag{5.7}$$

By counting the number of t-subsets in two ways, one has

$$D(n,k,t) \leq \binom{n}{t} \Big/ \binom{k}{t}.$$

So (5.6) now implies

$$|\mathcal{B}| \leq D(n-\gamma, r+1, 2) + \gamma \leq \Phi(\gamma). \tag{5.8}$$

Also, since γ is either zero or at least two, it follows from (5.7) that

$$|\mathcal{B}| \leq \begin{cases} \max\{D(n,r+1,2), n\}, & \text{if } n \leq r^2+r+2; \\ \max\{D(n,r+1,2), \binom{n-2}{2}/\binom{r+1}{2}+2\}, & \text{if } n \geq r^2+r+3. \end{cases} \tag{5.9}$$

The following result due to Johnson [82] can be used to simplify the bound in (5.9).

Lemma 5.3.2 (Johnson [82]) If $v < k^2/(t-1)$, then

$$D(v,k,t) \leq \left\lfloor \frac{(k+1-t)v}{k^2-(t-1)v} \right\rfloor.$$

Corollary 5.3.2 For $n \leq r^2+r+1$, we have $D(n,r+1,2) \leq n$.

Proof. If $n \le r^2 + r + 1$, then Lemma 5.3.2 applies and we have

$$\begin{aligned} D(n, r+1, 2) &\le \left\lfloor \frac{rn}{r^2 + 2r + 1 - n} \right\rfloor \\ &\le \left\lfloor \frac{rn}{r^2 + 2r + 1 - (r^2 + r + 1)} \right\rfloor \\ &= \left\lfloor \frac{rn}{r} \right\rfloor \\ &= n. \end{aligned}$$

□

For the range $r^2 + r + 2 \le n \le r^2 + 2r$, we apply the following result of Schönheim [127].

Lemma 5.3.3 (Schönheim [127]) $D(v, k, t) \le U(v, k, t)$.

Corollary 5.3.3 For $r^2 + r + 2 \le n \le r^2 + 2r$, $D(n, r+1, 2) \le n$.

Proof. If $n = r^2 + r + a$, $2 \le a \le r$, then by Lemma 5.3.3, we have

$$\begin{aligned} D(n, r+1, 2) &\le \left\lfloor \frac{r^2 + r + a}{r+1} \left\lfloor \frac{r^2 + r + a - 1}{r} \right\rfloor \right\rfloor \\ &= \left\lfloor \frac{r^2 + r + a}{r+1} \left\lfloor r + 1 + \frac{a-1}{r} \right\rfloor \right\rfloor \\ &= \left\lfloor \frac{r^2 + r + a}{r+1} (r+1) \right\rfloor \\ &= r^2 + r + a \\ &= n. \end{aligned}$$

□

Finally, we observe that $\binom{n-2}{2} / \binom{r+1}{2} + 2 \ge n$, for all $n \ge r^2 + r + 3$. This, together with

Corollary 5.3.2 and Corollary 5.3.3, establishes the following theorem.

Theorem 5.3.1 For any positive integers n and r such that $n \geq 2(r+1)$, we have

$$c(n, r+1, r) \leq \begin{cases} n, & \text{if } 2(r+1) \leq n \leq r^2+r+2; \\ \binom{n-2}{2} / \binom{r+1}{2} + 2, & \text{if } r^2+r+3 \leq n \leq r^2+2r; \\ \max\left\{ D(n, r+1, 2), \binom{n-2}{2} / \binom{r+1}{2} + 2 \right\}, & \text{if } (r+1)^2 \leq n. \end{cases}$$

The bound of Theorem 5.3.1 is stronger than that supplied by Theorem 5.1.1 of Erdös, Frankl, and Füredi. A consequence of Theorem 5.3.1 is the characterization of optimal 3-cover-free 4-uniform set systems and optimal 4-cover-free 5-uniform set systems of sufficiently large order.

5.3.1 Optimal 3-Cover-Free 4-Uniform Set Systems

The following is a consequence of Theorem 5.3.1 and existing results on 2-$(n, 4, 1)$ packings.

Corollary 5.3.4 For $n \geq 19$, a 3-cover-free 4-uniform set system of order n is optimal if and only if it is an optimal 2-$(n, 4, 1)$ packing.

Proof. Brouwer [22] has shown that $D(n,4,2) = U(n,4,2) - \epsilon$, where

$$\epsilon = \begin{cases} 1, & \text{if } n \equiv 7 \text{ or } 10 \pmod{12},\ n \neq 10, 19; \\ 1, & \text{if } n = 9 \text{ or } 17; \\ 2, & \text{if } n = 8, 10, \text{ or } 11; \\ 3, & \text{if } n = 19; \\ 0, & \text{otherwise.} \end{cases}$$

Since $\binom{n-2}{2}/\binom{4}{2} + 2 < D(n,4,2)$ when $n \geq 19$, we have $c(n,4,3) = D(n,4,2)$ if and only if $\gamma = 0$ in (5.8). This happens if and only if $g_2 = g_3 = \cdots = g_{n-2} = 0$, in which case we have a 2-$(n,4,1)$ packing. □

We can now determine $c(n,4,3)$ completely.

Theorem 5.3.2

$$c(n,4,3) = \begin{cases} n-3, & \text{if } 8 \leq n \leq 12; \\ D(n,4,2), & \text{if } n \geq 13. \end{cases}$$

Proof. Let $(X, \mathcal{B})$ be a 3-cover-free 4-uniform set system of order n. We can classify $(X, \mathcal{B})$ as follows:

(i) $(X, \mathcal{B})$ is a 2-$(n,4,1)$ packing.

(ii) $(X, \mathcal{B})$ has a pair of blocks that intersect in at least two points.

If $(X, \mathcal{B})$ is a 2-$(n,4,1)$ packing, then $|\mathcal{B}| \leq D(n,4,2)$. Otherwise, let $B, B' \in \mathcal{B}$ be such that $|B \cap B'| \geq 2$. Then there must be at least two points in $B \Delta B'$ that are not contained in any block in $\mathcal{B} \setminus \{B, B'\}$. Hence, $|\mathcal{B} \setminus \{B, B'\}| \leq c(n-2,4,3)$, implying

$|\mathcal{B}| \le c(n-2,4,3)+2$. With the base cases $c(6,4,3)=3$ and $c(7,4,3)=4$ provided by (5.1), it follows by induction that if $(X,\mathcal{B})$ satisfies condition (ii), then $|\mathcal{B}| \le n-3$ for $n \ge 6$. Hence, for any $n \ge 6$, we have

$$c(n,4,3) \le \max\{n-3, D(n,4,2)\}.$$

For $8 \le n \le 12$, we have $D(n,4,2) \le n-3$. The blocks $\{1,2,3,4\}$, $\{1,2,3,5\}$, $\ldots$, $\{1,2,3,n\}$ give a 3-cover-free 4-uniform set system of order n having $n-3$ blocks. Hence, $c(n,4,3) = n-3$ for $8 \le n \le 12$.

For $n \ge 13$, observe that the value of $D(n,4,2)$ meets the upper bound on $c(n,4,3)$ given by Theorem 5.3.1.

This completes the proof. □

5.3.2 Optimal 4-Cover-Free 5-Uniform Set Systems

We begin with the following result concerning the function $D(n,5,2)$.

Theorem 5.3.3 There exist positive constants a and N such that for all $n > N$, we have $D(n,5,2) \ge U(n,5,2) - a$.

Proof. The result for $n \equiv 1$ or 5 (mod 20) follows from the existence of 2-$(n,5,1)$ designs [75]. For $n \equiv 3$, 9, or 17 (mod 20), the result can be found in [105]. The result for $n \equiv 13$ (mod 20) is implied by the results of [74]. When $n \equiv 7$, 11, or 15 (mod 20), the result is obtained by Yin [155]. The result for $n \equiv 0$ (mod 4) is obtained by Yin [157], and the result for $n \equiv 2$ (mod 4) is obtained by Ling [91]. The proof for the remaining case $n \equiv 19$ (mod 20) is relegated to Corollary A.0.2 in Appendix A, as it is not central to the theme of this chapter. □

Corollary 5.3.5 There exists a constant N such that for all $n > N$, a 4-cover-free 5-uniform set system of order n is optimal if and only if it is an optimal 2-$(n, 5, 1)$ packing.

Proof. Since, for any constant a, $\binom{n-2}{2}/\binom{5}{2} + 2 < U(n, 5, 2) - a$ for all sufficiently large n, we have $c(n, 5, 4) = D(n, 5, 4)$ if and only if $\gamma = 0$ in (5.8). This happens if and only if $g_2 = g_3 = \cdots = g_{n-2} = 0$, in which case we have a 2-$(n, 5, 1)$ packing. □

5.4 2-Cover-Free Set Systems With Odd Block Size

We generalize the proof of Theorem 5.2.1 in a different direction. Instead of considering Δ-systems with nuclei of size two, we now consider Δ-systems with *nuclei of larger size.*

Let $(X, \mathcal{B})$ be any 2-cover-free $(2t - 1)$-uniform set system of order n. For $A \in \binom{X}{t}$, define

$$\mathcal{B}(A) = \{B \in \mathcal{B} \mid B \supset A\} \quad \text{and} \quad T(A) = \left\{F \in \binom{X}{t-1} \,\middle|\, A \cup F \in \mathcal{B}\right\}.$$

Further define

$$G_i = \left\{A \in \binom{X}{t} \,\middle|\, |T(A)| = i\right\}, \quad \text{for} \quad 0 \le i \le n - t.$$

Let $g_i = |G_i|$ and $G_{\geq 2} = \bigcup_{i=2}^{n-t} G_i$.

Suppose $A \in G_{\geq 2}$. If $F \in T(A)$, then F cannot be contained in any block of $\mathcal{B}$ other than $A \cup F$; for otherwise if F is contained in $B \in \mathcal{B}$, $B \neq A \cup F$, we can take $F' \in T(A)$, $F' \neq F$ (which exists because $A \in G_{\geq 2}$), to obtain

$$(A \cup F) \subseteq (A \cup F') \cup B.$$

Definition 5.4.1 The *upper shadow* of a k-uniform set system $(X, \mathcal{B})$ is the $(k+1)$-uniform set system $(X, \partial_u(\mathcal{B}))$, where

$$\partial_u(\mathcal{B}) = \left\{ A \in \binom{X}{k+1} \,\middle|\, A \supset B \quad \text{for some } B \in \mathcal{B} \right\}.$$

Lemma 5.4.1 (Sperner [138]) Let $(X, \mathcal{B})$ be a k-uniform set system. Then

$$|\partial_u(\mathcal{B})| \geq \frac{n-k}{k+1}|\mathcal{B}|.$$

Our observation above implies that no subset in the upper shadow of the $(t-1)$-uniform set system

$$\left(X, \bigcup_{A \in G_{\geq 2}} T(A) \right)$$

can be contained in any block of

$$\mathcal{B}' = \mathcal{B} \setminus \left(\bigcup_{A \in G_{\geq 2}} \mathcal{B}(A) \right).$$

The number of t-subsets of X contained in $\mathcal{B}'$ is therefore at most

$$\begin{aligned}
&\binom{X}{t} - \left| \partial_u \left(\bigcup_{A \in G_{\geq 2}} T(A) \right) \right| - |G_{\geq 2}| \\
&\leq \binom{X}{t} - \frac{n-t+1}{t} \left| \bigcup_{A \in G_{\geq 2}} T(A) \right| - \sum_{i=2}^{n-t} g_i \qquad \text{(by Lemma 5.4.1)} \\
&= \binom{X}{t} - \frac{n-t+1}{t} \sum_{i=2}^{n-t} i g_i - \sum_{i=2}^{n-t} g_i \\
&= \binom{X}{t} - \sum_{i=2}^{n-t} \frac{(n-t+1)i + t}{t} g_i.
\end{aligned}$$

Hence, the number of blocks in $\mathcal{B}'$ is at most

$$\frac{\binom{X}{t} - \sum_{i=2}^{n-t} \frac{(n-t+1)i+t}{t} g_i}{\binom{2t-1}{t}}.$$

It follows that

$$\begin{aligned}
|\mathcal{B}| &= |\mathcal{B}'| + \left| \bigcup_{A \in G_{\geq 2}} \mathcal{B}(A) \right| \\
&\leq \frac{\binom{X}{t} - \sum_{i=2}^{n-t} \frac{(n-t+1)i+t}{t} g_i}{\binom{2t-1}{t}} + \sum_{i=2}^{n-t} i g_i \\
&= \frac{\binom{X}{t} - \sum_{i=2}^{n-t} \left[\left(\frac{n-t+1}{t} - \binom{2t-1}{t} \right) i + 1 \right] g_i}{\binom{2t-1}{t}}.
\end{aligned}$$

Now, if $n \geq t\left(\binom{2t-1}{t} + 1\right) - 1$,

$$\sum_{i=2}^{n-t} \left[\left(\frac{n-t+1}{t} - \binom{2t-1}{t} \right) i + 1 \right] g_i \tag{5.10}$$

is either zero or at least

$$2\left(\frac{n-t+1}{t} - \binom{2t-1}{t} \right) + 1.$$

If the quantity in (5.10) is zero, then $g_2 = g_3 = \cdots = g_{n-t} = 0$, implying that $(X, \mathcal{B})$ is a t-$(n, 2t-1, 1)$ packing. This results in the following.

Theorem 5.4.1 For $n \geq t\left(\binom{2t-1}{t}+1\right)-1$,

$$c(n,2t-1,2) \leq \max\left\{D(n,2t-1,t), \frac{\binom{n}{t}-2\left(\frac{n-t+1}{t}-\binom{2t-1}{t}\right)-1}{\binom{2t-1}{t}}\right\}.$$

For $n \geq t\left(\binom{2t-1}{t}+1\right)-1$, Theorem 5.4.1 represents an improvement over the upper bound for $c(n,2t-1,2)$ given by Theorem 5.1.1.

Further strengthening of Theorem 5.4.1 is possible. For one thing, better lower bounds on the size of the upper shadow of $\bigcup_{A \in G_{\geq 2}} T(A)$ would improve Theorem 5.4.1. Our bound on the size of the upper shadow uses a rather weak result of Sperner. The following stronger bound can be obtained as a consequence of the Kruskal-Katona Theorem (see [17]).

Lemma 5.4.2 Let $(X, \mathcal{B})$ be a k-uniform set system. Then

$$|\partial_u(\mathcal{B})| \geq |\partial_u(\mathcal{A})|,$$

where $\mathcal{A}$ is the set of the last $|\mathcal{B}|$ k-subsets of $\binom{X}{k}$ in the colexicographic order.

At this point, however, it is not clear how the size of the upper shadow of the last m subsets in the colexicographic order can be determined.

5.5 Remarks

We see in Section 5.3 how some r-cover-free $(r+1)$-uniform set systems are characterized by 2-$(n, r+1, 1)$ packings. The following plausible conjecture (a packing analogue of Wilson's theorem for the asymptotic existence of 2-$(n, k, 1)$ designs), if true, would enable all r-cover-free $(r+1)$-uniform set systems of sufficiently large order $n \equiv 0, 1, 2$ or 3 (mod r) to be completely characterized.

Conjecture 5.5.1 For every k, there exists an N depending only on k, such that for all $n > N$, there is a 2-$(n, k, 1)$ packing with at least $U(n, k, 2) - o(n)$ blocks.

Lemma 5.5.1 If Conjecture 5.5.1 is true, then for every r, there exists an integer N depending only on r, such that for all $n > N$, $n - 1 \equiv 0, 1, 2$ or 3 (mod r), an r-cover-free $(r+1)$-uniform set system of order n is optimal if and only if it is an optimal 2-$(n, r+1, 1)$ packing.

Proof. If Conjecture 5.5.1 is true, then for all sufficiently large n, $D(n, r+1, 2) \geq U(n, r+1, 2) - o(n)$. If $n - 1 \equiv 0, 1, 2$ or 3 (mod r),

$$\begin{aligned} U(n, r+1, 2) &= \left\lfloor \frac{n}{r+1} \left\lfloor \frac{n-1}{r} \right\rfloor \right\rfloor \\ &\geq \left\lfloor \frac{n(n-4)}{(r+1)r} \right\rfloor \\ &\geq \frac{n(n-4) - (r+1)r + 1}{(r+1)r}. \end{aligned}$$

But $\frac{n(n-4) - (r+1)r + 1}{(r+1)r} - o(n) > \binom{n-2}{2} / \binom{r+1}{2} + 2$ for all sufficiently large n. This completes the proof. □

Notice that for the proof of Lemma 5.5.1, we need only the truth of Conjecture 5.5.1 for the congruence classes $n \equiv 0, 1, 2$ or 3 (mod $k - 1$). Nevertheless, we feel that it is likely that the conjecture in its full generality remains true. Erdös and Hanani [59] have shown that Conjecture 5.5.1 is true if $U(n, k, 2) - o(n)$ is replaced by $(1 - o(1))U(n, k, 2)$. Recent progress in probabilistic methods for constructing packings [70, 137, 154] also falls short of proving Conjecture 5.5.1.

CHAPTER 6

The Spectrum of Weakly Union-Free Twofold Triple Systems

6.1 Preliminaries

Definition 6.1.1 A set system $(X, \mathcal{A})$ is *weakly union-free* if there do not exist four distinct blocks $A_1, A_2, A_3, A_4 \in \mathcal{A}$ such that $A_1 \cup A_2 = A_3 \cup A_4$.

The problem of determining the maximum number of blocks in a weakly union-free 3-uniform set system was first studied by Frankl and Füredi [62]. This is a Turán-type problem since a 3-uniform set system is weakly union-free if and only if it avoids all of the configurations in Figure 6.1. The problem of determining the maximum number of blocks in a 3-uniform set system that avoids just the first configuration in Figure 6.1 has also been investigated by Lefmann, Phelps, and Rödl [90]. The motivation of Frankl and Füredi was to generalize Erdös' result [54] on the maximum number of edges in a graph that avoids cycles of length four. We shall see in the next section that this problem also has applications in nonadaptive group testing.

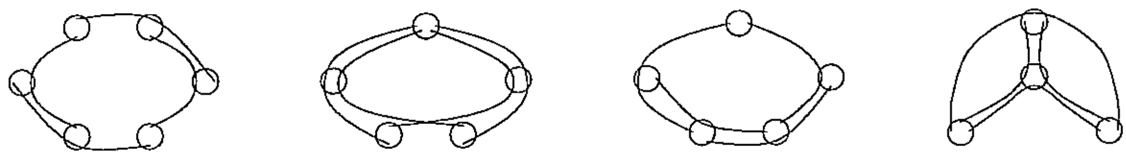


Figure 6.1: Forbidden configurations for weakly union-free 3-uniform set systems.

Definition 6.1.2 A *twofold triple system of order n*, denoted $\mathsf{TTS}(n)$, is a 2-$(n,3,2)$ design.

Frankl and Füredi showed in [62] that the maximum number of blocks in a weakly union-free 3-uniform set system is at most $n(n-1)/3$, with equality if and only if there is a weakly union-free TTS(n). A necessary condition for the existence of a TTS(n) is $n \equiv 0$ or $1 \pmod 3$. The following result was obtained by Frankl and Füredi.

Theorem 6.1.1 (Frankl and Füredi [62]) There is a constant N such that for all $n > N$, $n \equiv 1 \pmod 6$, there exists a weakly union-free $\mathsf{TTS}(n)$.

Theorem 6.1.1 settles only about a quarter of the admissible orders. In fact, in a remark of [62], Frankl and Füredi posed the problem of determining those orders n for which a weakly union-free TTS(n) exists, and made the conjecture that the condition $n \equiv 0$ or $1 \pmod 3$ is asymptotically sufficient.

In this chapter, we prove this conjecture of Frankl and Füredi and make substantial progress on the existence of weakly union-free twofold triple systems. In fact, we prove that with at most 7064 exceptions, weakly union-free twofold triple systems of all orders exist. We begin by describing an application to group testing in the next section.

6.2 Application to Approximate Identification

We focus on 3-RESTRICTED NONADAPTIVE (3/2)-APPROXIMATE IDENTIFICATION PROBLEM(2), where weakly union-free twofold triple systems play an important role.

Lemma 6.2.1 The dual system $(Y, \mathcal{B})$ of an algorithm for 3-RESTRICTED NONADAPTIVE (3/2)-APPROXIMATE IDENTIFICATION PROBLEM(2) must be weakly union-free.

Proof. Suppose not. Then there are four distinct blocks $B_{x_1}, B_{x_2}, B_{x_3}, B_{x_4} \in \mathcal{B}$ such that $B_{x_1} \cup B_{x_2} = B_{x_3} \cup B_{x_4}$. This implies that in the primal system $(X, \mathcal{P})$ of the algorithm, the sets $U = \{x_1, x_2\}$ and $U' = \{x_3, x_4\}$ satisfy $f_{\mathcal{P}}^{+}(U) = f_{\mathcal{P}}^{+}(U')$, where f is the 1-threshold function. But $|U \cup U'| = 4$, violating the condition of Lemma 4.2.3. □

Lemma 6.2.2 Any weakly union-free twofold triple system is the dual system of an algorithm for 3-RESTRICTED NONADAPTIVE (3/2)-APPROXIMATE IDENTIFICATION PROBLEM(2).

Proof. We verify that any weakly union-free twofold triple system $(Y, \mathcal{B})$ is the dual of a set system $(X, \mathcal{P})$ satisfying the conditions of Lemma 4.2.3. It suffices to verify for $|U| = |U'| = 2$ since $f_{\mathcal{P}}^{+}(U) = f_{\mathcal{P}}^{+}(U')$ for $|U| \neq |U'|$ would mean that $\mathcal{B}$ contains repeated blocks.

Lemma 4.2.3 implies that $(Y, \mathcal{B})$ is a dual system of an algorithm if and only if there exists $\mathcal{C} \subseteq \mathcal{B}$, $|\mathcal{C}| \leq 3$, such that whenever $B_1 \cup B_2 = B_1 \cup B_3$, for distinct blocks $B_1, B_2, B_3 \in \mathcal{B}$, we have $\{B_1, B_2, B_3\} \subseteq \mathcal{C}$. Note that we cannot have $B_1 \cup B_2 = B_3 \cup B_4$ for distinct blocks $B_1, B_2, B_3, B_4 \in \mathcal{B}$ since $(Y, \mathcal{B})$ is weakly union-free. So suppose we have distinct blocks $B_1, B_2, B_3 \in \mathcal{B}$ such that $B_1 \cup B_2 = B_1 \cup B_3 = F$. Hence $\mathcal{C} = \{B_1, B_2, B_3\}$. Suppose there exist $B, B' \in \mathcal{B}$ such that $B \cup B' = F$. Because $(Y, \mathcal{B})$ is weakly union-free, we must have $\{B, B'\} = \{B_1, B_4\}$ or $\{B, B'\} = \{B_2, B_3\}$. We consider $\{B, B'\} = \{B_1, B_4\}$.

We know that $|B_1 \cap B_2| \neq 0$ or 3 because $\mathcal{B}$ contains no repeated blocks. If $|B_1 \cap B_2| = 2$, then $|F| = 4$, giving $\{B_1, B_2, B_3, B_4\} = \binom{F}{3}$. This is a contradiction since $(F, \binom{F}{3})$ is not weakly union-free. It follows that $|B_1 \cap B_2| = 1$. But then $B_2 \setminus B_1$ is a 2-subset that must also be contained in the blocks B_3 and B_4, thus contradicting the assumption that $(Y, \mathcal{B})$ is a twofold triple system.

Thus, only the case $\{B, B'\} = \{B_2, B_3\}$ can occur, and we have $\{B, B'\} \subseteq \mathcal{C}$. □

6.3 PBD-Closure

Let W be the *spectrum* of weakly union-free twofold triple systems, that is,

$$W = \{n \mid \text{there exists a weakly union-free TTS}(n)\}.$$

Definition 6.3.1 Let K be a set of positive integers. A *pairwise balanced design* (PBD) *of order v with block sizes from K*, denoted PBD(v, K), is a set system $(X, \mathcal{B})$ of order v such that $|B| \in K$ for all $B \in \mathcal{B}$, and every 2-subset of X is contained in exactly one block of $\mathcal{B}$.

Definition 6.3.2 A set S of positive integers is *PBD-closed* if the existence of a PBD(v, S) implies that $v \in S$.

Definition 6.3.3 Let K be a set of positive integers and let $B(K) = \{v \mid$ there exists a PBD$(v, K)\}$. Then $B(K)$ is the *PBD-closure* of K.

The theory of PBD-closure is developed by Wilson in his series of ground-breaking work on the existence of PBDs [151, 152, 153]. The importance of this theory lies in the following result of Wilson [153].

Theorem 6.3.1 (Wilson [153]) Let K be a PBD-closed set. Then there exists a constant $N(K)$, such that for every $k \in K$, $\{v \mid v \geq N(K) \text{ and } v \equiv k \pmod{\beta(K)}\} \subseteq K$, where $\beta(K) = \gcd\{k(k-1) \mid k \in K\}$.

The next result, stated without proof in [62], shows the relevance of PBD-closure to weakly union-free twofold triple systems.

Lemma 6.3.1 (Frankl and Füredi [62]) The set W is PBD-closed.

Proof. Suppose that $(X, \mathcal{G})$ is a PBD(n, W). For each block $G \in \mathcal{G}$, replace G by the blocks of a weakly union-free twofold triple system, $(G, \mathcal{B}_G)$. This gives a twofold triple system $(X, \mathcal{T})$ of order n. Now suppose $(X, \mathcal{T})$ is not weakly union-free. Then there are four distinct blocks $A, B, C, D \in \mathcal{T}$ such that $A \cup B = C \cup D$. Without loss of generality, assume that $|A \cap C| = 2$ and $|B \cap D| = 2$. Hence, A and C are blocks of $\mathcal{B}_G$, and B and D are blocks of $\mathcal{B}_{G'}$ for some $G, G' \in \mathcal{G}$. Since $(G, \mathcal{B}_G)$ is weakly union-free, we cannot have $G = G'$. But $|(A \cup C) \cap (B \cup D)| = |(A \cap B) \cup (A \cap D) \cup (B \cap C) \cup (C \cap D)| \geq 2$. Hence G and G' intersect in at least two points. This is impossible since $(X, \mathcal{G})$ is a PBD. Therefore, $(X, \mathcal{T})$ is weakly union-free, that is, $n \in W$. □

Our proof of the asymptotic existence of weakly union-free twofold triple systems uses the following idea. First we determine some subset $L \subseteq W$ which contains at least one integer from each of the congruence classes 0, 1, 3, and 4 (mod 6). According to Theorem 6.3.1, there then exists a constant $N(L)$ such that for all $n > N(L)$, $n \in L$ if and only if $n \equiv 0$ or 1 (mod 3). Unfortunately, Theorem 6.3.1 does not supply any explicit upper bound on $N(L)$. Indeed, it has only been shown recently that $N(\{k\}) \leq \exp(\exp(k^{k^2}))$ [32]. Instead, we compute the PBD-closure of L with the help of a set of recursive constructions. This gives us an upper bound on $N(L)$ that is reasonably small.

6.4 Nonexistence and Some Direct Constructions

Obviously, the trivial TTS(0) and TTS(1) are both weakly union-free. So we assume throughout that the order is at least three. All twofold triple systems of order at most ten (without repeated blocks) have been enumerated. There is a unique 2-$(6, 3, 2)$ design, a unique 2-$(7, 3, 2)$ design, 13 nonisomorphic 2-$(9, 3, 2)$ designs [98], and 394 nonisomorphic 2-$(10, 3, 2)$ designs [39]. A quick computer search on these designs establishes the following.

Lemma 6.4.1 There do not exist any nontrivial weakly union-free twofold triple systems of order ten or less.

An infinite class of weakly union-free twofold triple systems have been constructed by Frankl and Füredi [62].

Lemma 6.4.2 (Frankl and Füredi [62]) Let $n \geq 13$, $n \equiv 1 \pmod 6$, be a prime power. Then there exists a weakly union-free TTS(n).

Proof. Let $1, \zeta$, and ζ^2 be the solutions to $x^3 = 1$ in GF(n). Let $\mathcal{B} = \{\{a, b, c\} \in \binom{\mathrm{GF}(n)}{3} \mid a + b\zeta + c\zeta^2 = 0\}$. Then $(\mathrm{GF}(n), \mathcal{B})$ is a weakly union-free TTS(n). □

We now construct some small weakly union-free TTS(n), where $n \equiv 0, 3$ or $4 \pmod 6$.

Lemma 6.4.3 There exists a weakly union-free TTS(16).

Proof. Let the point set be $X = \mathbf{Z}_8 \times \{0, 1\}$ and define the permutation σ on X so that

$$\sigma : (x, i) \mapsto (x + 1 \pmod 8, i).$$

Develop the starter blocks

$$\{(0,0),(1,0),(3,1)\}\ \ \{(0,0),(4,0),(0,1)\}\ \ \{(0,0),(2,0),(5,0)\}\ \ \{(0,0),(2,0),(1,1)\}$$
$$\{(3,0),(0,1),(1,1)\}\ \ \{(0,0),(1,1),(3,1)\}\ \ \{(0,0),(1,0),(5,1)\}\ \ \{(0,1),(2,1),(5,1)\}$$
$$\{(0,0),(2,1),(6,1)\}\ \ \{(0,0),(0,1),(7,1)\}$$

with the group $\langle\sigma\rangle$ to obtain a TTS(16). That this design is weakly union-free can easily be checked with a computer. □

Lemma 6.4.4 There exists a weakly union-free $\mathsf{TTS}(n)$ for $n \in \{21, 24, 30\}$.

Proof. These designs are all 1-rotational and are constructed as follows. In each case, the point set is taken to be $X = \mathbf{Z}_{n-1} \cup \{\infty\}$, where n is the order of the design. Let σ be the permutation

$$\sigma : x \mapsto \begin{cases} x + 1 \pmod{n-1}, & \text{if } x \neq \infty; \\ \infty, & \text{otherwise.} \end{cases}$$

For $n = 21$, take the following as starter blocks:

$$\{0,4,9\} \quad \{0,2,4\} \quad \{0,1,7\} \quad \{0,3,8\} \quad \{0,1,9\} \quad \{0,3,13\} \quad \{0,6,\infty\}$$

For $n = 24$, take the following as starter blocks:

$$\{0,8,15\} \quad \{0,1,4\} \quad \{0,3,12\} \quad \{0,2,13\} \quad \{0,6,16\} \quad \{0,1,18\} \quad \{0,4,18\} \quad \{0,2,\infty\}$$

For $n = 30$, take the following as starter blocks:

$$\begin{array}{ccccc} \{0,3,20\} & \{0,14,27\} & \{0,20,\infty\} & \{0,21,27\} & \{0,24,25\} \\ \{0,4,5\} & \{0,10,22\} & \{0,15,26\} & \{0,7,13\} & \{0,11,19\} \end{array}$$

Developing each set of starter blocks with the group generated by the appropriate σ yields the required weakly union-free twofold triple systems. □

Let us denote by Q the set of prime powers congruent to 1 (mod 6) and at least 13, together with the numbers 16, 21, 24, and 30. By Lemma 6.3.1, we have $B(Q) \subseteq W$, and Theorem 6.3.1 gives $n \in B(Q)$ for all sufficiently large $n \equiv 0$ or $1 \pmod 3$. So, at this point, the conjecture of Frankl and Füredi is already established in the affirmative.

6.4.1 Computational Details

It may appear that the set systems we presented in the foregoing lemmata are constructed magically. Therefore, an explanation is in order. The first thing we tried is a hill-climbing algorithm [140] that generates random TTS(n) (we are not claiming with uniform distribution). For each $n \equiv 0$ or 1 (mod 3), n not an odd prime power, and $12 \leq n \leq 33$, one million TTS(n) were generated and checked for the weakly union-free property. Rather surprisingly, this procedure yields no weakly union-free TTS(n). So it seems that weakly union-free TTS(n) are quite rare. It is well-known that hill-climbing algorithms tend to generate set systems without large automorphism groups [141]. We decided to restrict our search to several classes of TTS(n) having a certain degree of symmetry, in the hope that we have a better chance of finding one there which is weakly union-free. The three primary classes we focused on were

(i) 1-rotational TTS(n): those that have the group generated by the permutation

$$(\ 0\ \ 1\ \ \cdots\ \ n-2\ )(\infty)$$

as an automorphism group;

(ii) cyclic TTS(n): those that have the group generated by the permutation

$$(\ 0\ \ 1\ \ \cdots\ \ n-1\ )$$

as an automorphism group; and

(iii) bicyclic TTS(n): those that have the group generated by the permutation

$$(\ 0\ \ 1\ \ \cdots\ \ \frac{n}{2}\ )(\ \frac{n}{2}+1\ \ \frac{n}{2}+2\ \ \cdots\ \ n\ )$$

as an automorphism group.

All nonisomorphic 1-rotational TTS(n) have been enumerated by Chee and Royle [34], for $3 \leq n \leq 19$. We checked all of these set systems, for $n = 12$, 15, 16, and 18, but found none that were weakly union-free. We pushed further to the next case $n = 21$. Employing essentially the same algorithm as in [34], we generated a set of 1-rotational TTS(21) which is guaranteed to contain all the nonisomorphic 1-rotational TTS(21). We found that it is much faster to check each of these set systems, as it is being generated, for the weakly union-free property, than to first carry out isomorph rejection, and then check the remaining designs. We chose the faster option. Here, our persistence paid off; we found our first example of a weakly union-free TTS(n), where n is not an odd prime power. Encouraged by our result for $n = 21$, we continued with the examination of 1-rotational TTS(22). However, there does not exist a weakly union-free 1-rotational TTS(22). For $n \geq 24$, the resources required to enumerate all 1-rotational TTS(n) are quite demanding. We therefore settled for the examination of randomly generated 1-rotational TTS(n), using a hill-climbing algorithm similar to that described by Gibbons and Mathon [69]. Again, one million 1-rotational TTS(n) are constructed, for each $n \equiv 0$ or 1 (mod 3), and $24 \leq n \leq 33$. Only for $n = 24$ and $n = 30$ did we obtain any weakly union-free 1-rotational TTS(n) using this method.

Cyclic TTS(n) exist only if $n \equiv 0$, 1, 3, 4, 7, or 9 (mod 12) [44]. For $3 \leq n \leq 21$, nonisomorphic cyclic TTS(n) have been completely enumerated by Colbourn [43]. Since we already have weakly union-free TTS(n), for $n = 13$, 16, 19, and 21, we only checked the nonisomorphic cyclic TTS(12) and TTS(15). These are all not weakly union-free.

Hence, we move on to TTS(n) with smaller automorphism groups.

Bicyclic TTS(n) can only exist if $n \equiv 0$ or 4 (mod 6). We enumerated all nonisomorphic bicyclic TTS(12) and bicyclic TTS(18) and found none that were weakly union-free. However, we obtained a weakly union-free bicyclic TTS(16). Since the automorphism group involved is small, the complete enumeration of nonisomorphic bicyclic TTS(n), for $n \geq 22$, seems to require much more time. It involves finding all $\{0,1\}$-vectors $\mathbf{x}$ that satisfy $A\mathbf{x} = 21$, where A is the Kramer-Mesner matrix (see [89]) on the orbits of $\binom{X}{2}$ and $\binom{X}{3}$. The matrix A has dimension 21×140 for the case $n = 22$. So again, we resort to hill-climbing algorithms that construct random bicyclic TTS(n). We did not manage to find any weakly union-free TTS(n) this way.

The existence of small ingredients is very important in the recursive construction of PBDs. They can usually affect the asymptotic existence of PBDs in a drastic manner. We paid particular attention to the existence of weakly union-free TTS(12). It is hopeless to enumerate all nonisomorphic TTS(12). Royle [122] has constructed one million nonisomorphic TTS(12) with a hill-climbing algorithm. We attempted to enumerate all nonisomorphic TTS(12) with a nontrivial automorphism group. This was done for those whose full automorphism group has order divisible by an odd prime. None of them is weakly union-free. The amount of work required to enumerate the remaining case, where two divides the order of the full automorphism group, seems prohibitive at present. A more detailed account of this enumeration effort appears in Appendix B.

Definition 6.4.1 A *Steiner triple system of order n,* denoted STS(n), is a 2-(n, 3, 1) design.

It is well-known that an STS(n) exists if and only if $n \equiv 1$ or 3 (mod 6) [15].

Definition 6.4.2 Two STS(n), $(X, \mathcal{A})$ and $(X, \mathcal{B})$ are *orthogonal* if

(i) $\mathcal{A} \cap \mathcal{B} = \varnothing$, and

(ii) if $\{u,v,w\},\{x,y,w\} \in \mathcal{A}$, and $\{u,v,s\},\{x,y,t\} \in \mathcal{B}$, then $s \neq t$.

It is known that there exists a pair of orthogonal STS(n) for all $n \equiv 1$ or 3 (mod 6) [40].

Definition 6.4.3 A TTS(n), $(X, \mathcal{A})$, is *decomposable* if there exists a partition $\mathcal{A} = \mathcal{B} \dot\cup \mathcal{C}$ such that $(X, \mathcal{B})$ and $(X, \mathcal{C})$ are both STS(n).

Another avenue we explored is based on the following observation.

Lemma 6.4.5 If $(X, \mathcal{A})$ is a weakly union-free TTS(n) that is decomposable into two STS(n), then these two STS(n) must be orthogonal.

Proof. Let $(X, \mathcal{A})$ be decomposable into $(X, \mathcal{B})$ and $(X, \mathcal{C})$. If $(X, \mathcal{B})$ and $(X, \mathcal{C})$ are not orthogonal, then there exist $\{u,v,w\},\{x,y,w\} \in \mathcal{B}$ and $\{u,v,s\},\{x,y,s\} \in \mathcal{C}$. But $\{u,v,w\} \cup \{x,y,s\} = \{x,y,w\} \cup \{u,v,s\}$, contradicting the fact that $(X, \mathcal{A})$ is weakly union-free. □

All nonisomorphic pairs of orthogonal STS(15) have been enumerated by Gibbons [68]. We checked all TTS(15) that are the union of a pair of orthogonal STS(15), but none of these is weakly union-free either. At this point, we decided to move on to the computation of the PBD-closure.

6.5 Product Constructions

In this section, we describe two product constructions for weakly union-free TTS(n).

Theorem 6.5.1 Let $m \equiv 1$ or 3 (mod 6). If there exist a weakly union-free TTS(m) and a weakly union-free TTS(n), then there exists a weakly union-free TTS(mn).

Proof. Let $(X, \mathcal{A})$, $(Y, \mathcal{B})$, and $(X, \mathcal{C})$ be, respectively, a weakly union-free TTS(m), a weakly union-free TTS(n), and an STS(m). Define $Z = X \times Y$, and

$$\begin{aligned}\mathcal{D} = \{&\{(a, y), (b, y), (c, y)\} \mid \{a, b, c\} \in \mathcal{A}, y \in Y\}\\ &\cup \{\{(x, a'), (x, b'), (x, c')\} \mid \{a', b', c'\} \in \mathcal{B}, x \in X\}\\ &\cup \{\{(a, a'), (b, b'), (c, c')\} \mid \{a, b, c\} \in \mathcal{C}, \{a', b', c'\} \in \mathcal{B}\}.\end{aligned}$$

It is a routine matter to verify that $(Z, \mathcal{D})$ is an TTS(mn). We now show that it is weakly union-free. The proof is by case analysis. Suppose on the contrary that we have four distinct blocks $A, B, C, D \in \mathcal{D}$ such that $A \cup B = C \cup D$. There are four cases to consider.

Case (i). The four blocks A, B, C, and D have the following form:

$$\begin{aligned}A &= \{(a, a'), (b, b'), (c, c')\},\\ B &= \{(d, d'), (e, e'), (f, f')\},\\ C &= \{(a, a'), (b, b'), (f, f')\},\\ D &= \{(c, c'), (d, d'), (e, e')\},\end{aligned}$$

where $|A \cap B| = 0$. Consider the number of elements in the set $S = \{a', b', c', d', e', f'\}$. The definition of $\mathcal{D}$ implies that each of the sets $\{a', b', c'\}$, $\{d', e', f'\}$, $\{a', b', f'\}$, and $\{c', d', e'\}$ has size one or three. Hence, $|S| \in \{1, 3, 6\}$.

If $|S| = 1$, then it must be the case that $\{a, b, c\}, \{d, e, f\}, \{a, b, f\}, \{c, d, e\} \in \mathcal{A}$. But then $(X, \mathcal{A})$ is not weakly union-free, a contradiction.

If $|S| = 3$, we must have $\{a', b', c'\} = \{d', e', f'\}$ and $c' = f'$. If $|\{a, b, c, d, e, f\}| > 1$, then $\{a, b, c\}, \{d, e, f\}, \{a, b, f\}, \{c, d, e\} \in \mathcal{C}$. Since $(X, \mathcal{C})$ is an STS(m), this implies $c = f$. Hence, $(c, c') = (f, f')$, contradicting the assumption that $|A \cap B| = 0$. If

$|\{a,b,c,d,e,f\}| = 1$, then it must be the case that $\{a',b',c'\}$, $\{d',e',f'\}$, $\{a',b',f'\}$, $\{c',d',e'\} \in \mathcal{B}$. But then $(Y,\mathcal{B})$ is not weakly union-free, a contradiction.

If $|S| = 6$, then it must be the case that $\{a',b',c'\}, \{d',e',f'\}, \{a',b',f'\}, \{c',d',e'\} \in \mathcal{B}$. But then $(Y,\mathcal{B})$ is not weakly union-free, a contradiction.

Case (ii). The four blocks A, B, C, and D have the following form:

$$A = \{(a,a'),(b,b'),(c,c')\},$$
$$B = \{(a,a'),(d,d'),(e,e')\},$$
$$C = \{(a,a'),(b,b'),(e,e')\},$$
$$D = \{(a,a'),(c,c'),(d,d')\},$$

where $|A \cap B| = 1$. Consider the number of elements in the set $S = \{a',b',c',d',e'\}$. A bit of reflection reveals that $|S| \in \{1,3,5\}$.

If $|S| = 1$, then it must be the case that $\{a,b,c\}$, $\{a,d,e\}$, $\{a,b,e\}$, $\{a,c,d\} \in \mathcal{A}$. But then $(X,\mathcal{A})$ is not weakly union-free, a contradiction.

If $|S| = 3$, we may assume without loss of generality that a', b', and c' are all distinct. Then we must have $b' = d'$ and $c' = e'$. If $|\{a,b,c,d,e\}| = 1$, then we have $\{a',b',c'\}$, $\{a',d',e'\}$, $\{a',b',e'\}$, $\{a',c',d'\} \in \mathcal{B}$. But then $(Y,\mathcal{B})$ is not weakly union-free, a contradiction. So we must have $|\{a,b,c,d,e\}| > 1$. Hence, $\{a,b,c\}$, $\{a,d,e\}$, $\{a,b,e\}$, $\{a,c,d\} \in \mathcal{C}$. Since $(X,\mathcal{C})$ is an STS(m), this implies $b = d$ and $c = e$. Thus $(b,b') = (d,d')$ and $(c,c') = (e,e')$, contradicting the assumption that $|A \cap B| = 1$.

If $|S| = 5$, then it must be the case that $\{a',b',c'\}$, $\{a',d',e'\}$, $\{a',b',e'\}$, $\{a',c',d'\} \in \mathcal{B}$. But then $(Y,\mathcal{B})$ is not weakly union-free, a contradiction.

Case (iii): The four blocks A, B, C, and D have the following form:

$$A = \{(a, a'), (b, b'), (c, c')\},$$
$$B = \{(a, a'), (d, d'), (e, e')\},$$
$$C = \{(a, a'), (b, b'), (d, d')\},$$
$$D = \{(b, b'), (c, c'), (e, e')\},$$

where $|A \cap B| = 1$. Consider the number of elements in the set $S = \{a', b', c', d', e'\}$. A bit of reflection reveals that $|S| \in \{1, 3, 5\}$.

If $|S| = 1$, then it must be the case that $\{a, b, c\}$, $\{a, b, d\}$, $\{a, c, d\}$, $\{b, c, d\} \in \mathcal{A}$. But then $(X, \mathcal{A})$ is not weakly union-free, a contradiction.

If $|S| = 3$, we may assume without loss of generality that a', b', and c' are all distinct. Then we must have $b' = e'$ and $c' = d'$. If $|\{a, b, c, d, e\}| = 1$, then we have $\{a', b', c'\}$, $\{a', d', e'\}$, $\{a', b', e'\}$, $\{a', c', d'\} \in \mathcal{B}$. But then $(Y, \mathcal{B})$ is not weakly union-free, a contradiction. So we must have $|\{a, b, c, d, e\}| > 1$. Hence, $\{a, b, c\}$, $\{a, d, e\}$, $\{a, b, d\}$, $\{b, c, e\} \in \mathcal{C}$. Since $(X, \mathcal{C})$ is an STS(m), we must have $b = e$ and $c = d$. Thus $(b, b') = (e, e')$ and $(c, c') = (d, d')$, contradicting the assumption that $|A \cap B| = 1$.

If $|S| = 5$, then it must be the case that $\{a', b', c'\}$, $\{a', d', e'\}$, $\{a', b', e'\}$, $\{a', c', d'\} \in \mathcal{B}$. But then $(Y, \mathcal{B})$ is not weakly union-free, a contradiction.

Case (iv): The four blocks A, B, C, and D have the following form:

$$A = \{(a, a'), (b, b'), (c, c')\},$$
$$B = \{(a, a'), (b, b'), (d, d')\},$$
$$C = \{(a, a'), (c, c'), (d, d')\},$$
$$D = \{(b, b'), (c, c'), (d, d')\},$$

where $|A \cap B| = 2$. Consider the number of elements in the set $S = \{a', b', c', d'\}$. A bit of reflection reveals that $|S| \in \{1, 4\}$.

If $|S| = 1$, then it must be the case that $\{a, b, c\}$, $\{a, b, d\}$, $\{a, c, d\}$, $\{b, c, d\} \in \mathcal{A}$. But then $(X, \mathcal{A})$ is not weakly union-free, a contradiction.

If $|S| = 4$, then it must be the case that $\{a', b', c'\}$, $\{a', b', d'\}$, $\{a', c', d'\}$, $\{b', c', d'\} \in \mathcal{B}$. But then $(Y, \mathcal{B})$ is not weakly union-free, a contradiction.

This completes the proof. □

The next construction is a singular direct product-type construction.

Theorem 6.5.2 Let $m \equiv 4 \pmod 6$. If there exist a weakly union-free TTS(m) and a weakly union-free TTS(n), then there exists a weakly union-free TTS($(m-1)n+1$).

Proof. Let $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ be, respectively, a weakly union-free TTS(m) and a weakly union-free TTS(n). Let $x^* \in X$ be a distinguished element and let $(X \setminus \{x^*\}, \mathcal{C})$ be an

STS$(m-1)$. Define $Z = ((X \setminus \{x^*\}) \times Y) \cup \{\infty\}$, and

$$\begin{aligned}\mathcal{D} = &\{\{(a,y),(b,y),(c,y)\} \mid x^* \notin \{a,b,c\} \in \mathcal{A}, y \in Y\}\\ &\cup \{\{(a,y),(b,y),\infty\} \mid \{a,b,x^*\} \in \mathcal{A}.y \in Y\}\\ &\cup \{\{(x,a'),(x,b'),(x,c')\} \mid \{a',b',c'\} \in \mathcal{B}, x \in X, x \neq x^*\}\\ &\cup \{\{(a,a'),(b,b'),(c,c')\} \mid \{a,b,c\} \in \mathcal{C}, \{a',b',c'\} \in \mathcal{B}\}.\end{aligned}$$

It is straightforward to verify that $(Z,\mathcal{D})$ is a TTS$((m-1)n+1)$. The proof of Theorem 6.5.1 shows that there are no four distinct blocks $A,B,C,D \in \mathcal{D}$ such that $A \cup B = C \cup D$, unless at least one of A, B, C, or D contains the point ∞. If follows that if $A \cup B = C \cup D$, then

$$\begin{aligned}A,B,C,D \in &\{\{(a,y),(b,y),(c,y)\} \mid x^* \notin \{a,b,c\} \in \mathcal{A}, y \in Y\}\\ &\cup \{\{(a,y),(b,y),\infty\} \mid \{a,b,x^*\} \in \mathcal{A}.y \in Y\}.\end{aligned}$$

It is not hard to see that this is also impossible unless $(X,\mathcal{A})$ is not weakly union-free. □

For a given set K of positive integers, let us define a sequence of sets, $(K_i)_{i \geq 0}$, as follows:

$$\begin{aligned}&K_0 = K, \quad \text{and for } i \geq 1,\\ &K_i = K_{i-1}\\ &\quad \cup \{k \mid k = mn \text{ for some } m,n \in K_{i-1}, \text{ and either } m \text{ or } n \text{ is 1 or 3 (mod 6)}\}\\ &\quad \cup \{k \mid k = (m-1)n+1 \text{ for some } m,n \in K_{i-1}, \text{ and either } m \text{ or } n \text{ is 4 (mod 6)}\}.\end{aligned}$$

Let $L = Q_\infty$. Theorem 6.5.1 and Theorem 6.5.2 imply that $L \subseteq W$. Hence $B(L) \subseteq W$. We shall improve on the asymptotic existence of weakly union-free twofold triple systems given in Section 6.4 by considering the PBD-closure of L.

6.6 Eventual Periodicity of W

In this section, we investigate the set W. We prove that for all $n \equiv 0$ or $1 \pmod 3$, $n \in W$ except for a set of at most 7064 values of n, the largest of which is 137628.

6.6.1 Recursive Constructions for PBDs

In this section, we describe several recursive constructions for PBDs. First, we need to introduce some terminology.

Definition 6.6.1 Let K be a set of positive integers. A *group divisible design* (GDD) *of order* v, denoted K-GDD, is a triple $(X, \mathcal{G}, \mathcal{B})$, where $\mathcal{G}$ is a partition of X into parts, called groups, and $(X, \mathcal{B})$ is a set system which satisfies the properties:

(i) if $B \in \mathcal{B}$, then $|B| \in K$;

(ii) every 2-subset of X occurs in exactly one block or one group, but not both;

(iii) $|\mathcal{G}| > 1$.

The *type* of a GDD $(X, \mathcal{G}, \mathcal{B})$ is the multiset $\{|G| \mid G \in \mathcal{G}\}$. We usually use an "exponential" notation to describe types: a type $g_1^{u_1} g_2^{u_2} \cdots g_t^{u_t}$ denotes u_i occurrences of g_i, $1 \leq i \leq t$.

A K-GDD of order v and type $g_1^{u_1} g_2^{u_2} \cdots g_t^{u_t}$ can be viewed as a PBD$(v, K \cup \{g_1, g_2, \ldots, g_t\})$ by considering the groups of the GDD to be blocks of the PBD also. A K-GDD of order v and type $g_1^{u_1} g_2^{u_2} \cdots g_t^{u_t}$ can also be used to create a PBD$(v+1, K \cup$

$\{g_1+1, g_2+1, \ldots, g_t+1\}$) by *adjoining* a new point to each group and considering the resulting subsets as blocks.

Definition 6.6.2 A *transversal design* $\mathrm{TD}(k,n)$ is a $\{k\}$-GDD of type n^k.

It is well-known that a $\mathrm{TD}(k,n)$ is equivalent to $k-2$ mutually orthogonal Latin squares (MOLS) of order n. For a list of lower bounds on the number of MOLS of all orders up to 10000, we refer the reader to [1].

We also need to define various types of incomplete designs.

Definition 6.6.3 An *incomplete transversal design* $\mathrm{TD}(k,n)-\mathrm{TD}(k,m)$ is a quadruple $(X, \mathcal{G}, H, \mathcal{B})$, where

(i) $(X, \mathcal{B})$ is a set system of order kn;

(ii) $\mathcal{G}$ is a partition of X into k parts, called groups, each of size n;

(iii) H is a subset of X, called a hole, with the property that $|G \cap H| = m$ for each $G \in \mathcal{G}$;

(iv) every 2-subset of X is

- contained in the hole, and contained in no blocks; or
- contained in a group, and contained in no blocks; or
- contained in neither the hole nor a group, and contained in exactly one block.

We also need PBDs containing subdesigns, or flats. Let $(X, \mathcal{A})$ be a PBD. If a set of points $Y \subseteq X$ has the property that, for any $A \in \mathcal{A}$, either $|Y \cap A| \leq 1$ or $A \subseteq Y$, then we say that Y is a *flat* of $(X, \mathcal{A})$. The *order* of the flat is $|Y|$. If Y is a flat, then we can delete all blocks $A \subseteq Y$, replace them by a single block, Y, and obtain a PBD. Any block or point of a PBD is itself a flat. Often we do not require that the flat be present. This gives rise to the notion of incomplete PBDs.

Definition 6.6.4 Let K be a set of positive integers and h a nonnegative integer. An *incomplete pairwise balanced design* (IPBD) *of order v with a hole of order h*, denoted IPBD(v, h, K), is a triple $(X, H, \mathcal{B})$, where $|B| \in K$ for all $B \in \mathcal{B}$, $H \subseteq X$ such that $|H| = h$, and $(X, \mathcal{B} \cup \{H\})$ is a PBD.

We begin with a useful construction for PBDs with two consecutive block sizes.

Lemma 6.6.1 (Truncation of a Group in a Transversal Design (see [103])) Let k be a positive integer. Let $K = \{k, k+1\}$. Suppose that there exists a TD$(k+1, n)$. Then there exists a K-GDD of type $n^k m$, for $0 \leq m \leq n$.

Proof. Delete $n - m$ points from one group of a TD$(k+1, n)$. □

If instead of deleting points from a group, we delete points from a block, then we obtain the following well-known result.

Lemma 6.6.2 (Truncation of a Block in a Transversal Design) Let k and m be integers such that $0 \leq m \leq k$. Let $K = \{k, m\}$. Suppose that there exists a TD(k, n). Then there exists a K-GDD of type $n^m(n-1)^{k-m}$.

Below are two further constructions for PBDs with two consecutive block sizes.

Lemma 6.6.3 (Bennett [13]) If n is a prime power, and $1 \leq k \leq n$, then for $0 \leq t \leq n - k$, there exists a PBD$(kn + t, K)$, where $K = \{k, k+1, k+t, n\}$.

Lemma 6.6.4 (Spike Construction (see [103])) Let k and n be positive integers. Let $K = \{k, k+1, k+n\}$. If there exists a TD$(k+n, m)$, then there exists a K-GDD of type $m^k 1^n$.

The following constructions are also useful.

Lemma 6.6.5 (Greig (see [103])) Let k and n be positive integers. If there exists a TD$(k+n, k+n-1)$, then there exists a $\{k-1, k+1\}$-GDD of type $(k+n-2)^k n^1$.

Lemma 6.6.6 (Brouwer [23]) Let q be a prime power, and let t be an integer satisfying $0 < t < q^2 - q + 1$. Then there exists a $\{t, q+t\}$-GDD of type t^{q^2+q+1}.

By adjoining a point to the GDDs constructed in Lemma 6.6.1, Lemma 6.6.2, and Lemma 6.6.5, we obtain the following three results.

Lemma 6.6.7 Let k be a positive integer and suppose that there exists a TD$(k+1, n)$. Let $0 \leq m \leq n$. Then there exists a PBD$(kn+m+1, K)$, where $K = \{k, k+1, n+1, m+1\}$. If $m = 0$, there are no blocks of size $k+1$. If $m = n$, there are no blocks of size k.

Lemma 6.6.8 Let k and m be integers such that $0 \leq m \leq k$ and suppose that there exists a TD(k, n). Then there exists a PBD$(k(n-1)+m+1, K)$, where $K = \{k, m, n, n+1\}$. If $m = 0$, there are no blocks of size $n+1$, and if $m = k$, there are no blocks of size n.

Lemma 6.6.9 Let k and n be positive integers. If there exists a TD$(k+n, k+n-1)$, then there exists a PBD$(k(k+n-2)+n+1, K)$, where $K = \{k-1, k+1, n+1, k+n-1\}$.

The remaining constructions are product type constructions. The most general of these constructions is the singular indirect product construction due to Mullin [102, 104].

Lemma 6.6.10 (Singular Indirect Product) Let K be a set of positive integers and $k \in K$. Let h be a nonnegative integer and suppose that the following designs exist:

(i) a TD$(k, m+n)$−TD(k, m);

(ii) an IPBD$(m+n+h, m+h, K)$; and

(iii) a PBD$(km+h, K)$.

Then there exists a PBD($k(m+n)+h, K$) containing flats of order k and $km+h$.

If we let $m=0$ in Lemma 6.6.10, we obtain the singular direct product construction.

Lemma 6.6.11 (Singular Direct Product) Let K be a set of positive integers and $k \in K$. Let h be a nonnegative integer and suppose there exist a TD(k, n), an IPBD($n+h, h, K$), and a PBD(h, K). Then there exists a PBD($kn+h, K$) containing flats of order k, $n+h$, and h.

In order to apply the singular indirect product construction, we need incomplete transversal designs. We rely on the following result of Wilson (see [24]) to supply these.

Lemma 6.6.12 (Wilson (see [24])) Let u and t be integers such that $0 \leq m \leq t$. If there exist a TD(k, n), a TD($k, n+1$) and a TD($k+1, t$), then there exists a TD($k, tn+m$)−TD(k, m).

We also use a well-known result of MacNeish [95].

Lemma 6.6.13 (MacNeish [95]) Let $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ be the prime power factorization of n. Then there exists a TD(k, n) for all $k \leq 1 + \min\{p_i^{e_i} \mid 1 \leq i \leq k\}$.

Finally, another set of important ingredients for our constructions is provided by Brouwer.

Lemma 6.6.14 (Brouwer (see [1])) If $k \leq 32$, then there exists a TD(k, n) for all $n \geq 52503$.

6.6.2 PBDs up to Order Thirteen Million

In this section, we describe the construction of PBD($n, B(L)$) for $n \equiv 0$ or $1 \pmod 3$, $n \leq 13000000$, in preparation for determining W. This was accomplished with a computer program that applied the constructions given in Section 6.6.1. It would take too much space to write down all the constructions, but we give a brief description which will enable anyone to easily duplicate our computations. Our computer program has a knowledge of all the results in Section 6.6.1. The transversal designs employed are those that exist by the MOLS table in [1], Lemma 6.6.13, and Lemma 6.6.14. The incomplete transversal designs used are those whose existence is given by Lemma 6.6.12. Given an integer $n \equiv 0$ or 1 (mod 3), the program attempts to construct a PBD($n, B(L)$) by applying the following constructions (in the order indicated):

(1) Lemma 6.6.1 (truncate a group of a transversal design),

(2) Lemma 6.6.7 (adjoin a point to a group-truncated transversal design),

(3) Lemma 6.6.2 (truncate a block of a transversal design),

(4) Lemma 6.6.8 (adjoin a point to a block-truncated transversal design),

(5) Lemma 6.6.3 (Bennett's construction),

(6) Lemma 6.6.4 (spike construction),

(7) Lemma 6.6.5 (Greig's construction),

(8) Lemma 6.6.9 (adjoin a point to the GDD obtained by Greig's construction),

(9) Lemma 6.6.6 (Brouwer's construction),

(10) Lemma 6.6.11 (singular direct product),

(11) Lemma 6.6.10 (singular indirect product).

The singular indirect product construction is a somewhat complicated construction and we apply it only with $k \in \{13, 16, 19, 24, 25, 31\}$ and $m + h \in \{13, 16, 19,$

21, 24, 25, 31, 37, 43, 49}. Our program also keeps track of all flats appearing in the constructed PBDs. This information is used immediately by all subsequent constructions. Our computational results up to this point can be summarized as follows.

Theorem 6.6.1 If $12 \leq n \leq 13000000$, $n \equiv 0$ or 1 (mod 3), then $n \in B(L)$ with at most 8507 exceptions.

Let L' be the set of integers in $B(L)$ given by Theorem 6.6.1. Note that $L' \subseteq W$. Since our interest is in the set W, we next compute $B(L'_\infty) \subseteq W$ using the same program as in the computation of $B(L)$. Let E be the set of 7058 numbers given in Appendix C. The result of this stage of computation is given below.

Theorem 6.6.2 If $12 \leq n \leq 13000000$, $n \equiv 0$ or 1 (mod 3), and $n \notin E$, then $n \in W$.

6.6.3 The Spectrum

In this section, we show that all $n \equiv 0$ or 1 (mod 3) exceeding 13000000 are in W.

Lemma 6.6.15 If $n \equiv 0$ or 1 (mod 3) and $n \geq 1283140$, then $n \in W$.

Proof. We prove this theorem by induction on n. First notice that for all $n \equiv 1$ (mod 3) and $52504 \leq n \leq 13000000$, we have $n \in W$. This can be verified easily with the list given in Appendix C. Now, any $n \equiv 0$ or 1 (mod 3) and at least 1283140 can be written in the form $n = 72m + 24 + g$, where $m \in [17501, 4333333]$ and $g \equiv 0$ or 1 (mod 3), $g \in [23044, 23115]$. Note that $g \in W$ for all g in this interval. By Lemma 6.6.14, there exists a TD$(24, 3m+1)$. We can therefore apply Lemma 6.6.2 to obtain a $\{24, 25\}$-GDD of type $(3m+1)^{24}g^1$. Our induction hypothesis gives $3m+1 \in W$. This implies $72m + 24 + g \in W$ by the PBD-closure of W. By induction, the proof is complete. □

We can now give the result for the spectrum of weakly union-free twofold triple systems.

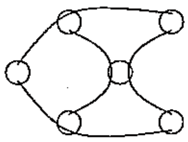
Theorem 6.6.3 For all $n \equiv 0$ or $1 \pmod 3$, there exists a weakly union-free $\mathrm{TTS}(n)$, provided $n \geq 137629$. Below this bound, there are 7058 values of n (appearing in Appendix C) for which the existence of a $\mathrm{TTS}(n)$ is not decided. There are no nontrivial weakly union-free $\mathrm{TTS}(n)$ for $n \leq 10$.

Proof. Follows directly from Theorem 6.6.2 and Lemma 6.6.15. □

6.7 Subsystem-Free Twofold Triple Systems

Definition 6.7.1 A (j,k)-*configuration* is a set system $(X, \mathcal{A})$ such that $|X| = j$, $|\mathcal{A}| = k$, and $\bigcup_{A \in \mathcal{A}} = X$.

The problem of constructing Steiner triple systems of every admissible order avoiding $(j+2, j)$-configurations, $2 \leq j \leq r$, for every fixed r, was proposed by Erdös [56]. For $r = 4$, the problem asks for the existence of Steiner triple systems avoiding the $(6,4)$-configuration below, known as the *Pasch configuration.*



Such Steiner triple systems are called *anti-Pasch*, and we shall see more of them in the next few chapters.

At the recent Tenth Ontario Combinatorics Workshop which was held at the Fields Institute for Research in Mathematical Sciences, Terry Griggs mentioned (April 27, 1996) to the author that one natural analogue of anti-Pasch Steiner triple systems for twofold triple systems is those that avoid TTS(4), the last configuration in Figure 6.1. The reason is that the Pasch configuration is the $(j, 4)$-configuration with the minimum possible j that a Steiner triple system can contain, while a TTS(4) is the $(j, 4)$-configuration with the minimum possible j that a twofold triple system can contain. A TTS(v) that avoids TTS(4) is called *TTS(4)-free*, and is called *subsystem-free* if it avoids TTS(w), for all $w < v$.

Definition 6.7.2 Let $(X, \mathcal{A})$ be a TTS(v). The *neighbourhood* of a point $x \in X$ is the graph $G = (V, E)$, where $V = X \setminus \{x\}$ and $E = \{A \setminus \{x\} \mid x \in A \text{ and } A \in \mathcal{A}\}$.

There is an intimate connection between embeddings of the complete graph K_n on orientable surfaces and twofold triple systems of order n. Heffter [78] seems to be the first to realize this connection. Heffter's ideas were later used by Emch [53] to compute the automorphism groups of some twofold triple systems. The article of Alpert [8] is a nice exposition on this topic. The following theorem of Ducrocq and Sterboul [49] is obtained by observing that some embeddings of K_n constructed by Ringel and Youngs [120] (see also [119]) give TTS(n) with the desired property.

Theorem 6.7.1 (Ducrocq and Sterboul [49]) For every $v \equiv 0$ or $1 \pmod 3$, $v \geq 4$, there exists a TTS(v) in which the neighbourhood of every point is a cycle of length $v - 1$.

Colbourn [37] observed that the TTS(v) in Theorem 6.7.1 is TTS(4)-free and avoids even the second configuration in Figure 6.1. In fact, more is true. We show that these TTS(v) are subsystem-free.

Theorem 6.7.2 (Chee and Colbourn) There exists a subsystem-free TTS(v) for all $v \equiv 0$ or $1 \pmod 3$, except when $v = 3$.

Proof. Suppose the TTS(v) in Theorem 6.7.1 contains a TTS(w). The neighbourhood of any point in this TTS(w) is a graph on $w - 1$ vertices with $w - 1$ edges, and must be a subgraph of a cycle of length $v - 1$. This is possible only if $w = v$. □

The corresponding problem for Steiner triple systems has been solved by Doyen [47].

CHAPTER 7

Fault-Tolerant Group Testing

7.1 Introduction

One of the most important issues in group testing that demands further investigation is fault-tolerance. In real life applications, tests are affected by too many factors to be rarely error-free. Let (X, r, f, Π) be a group testing problem and $\mathcal{O}$ an oracle implementing f. A test on a pool $P \subseteq X$ performed by an algorithm with access to $\mathcal{O}$ is called *erroneous* if the result returned is not $f(P)$. Erroneous tests can happen as a result of incorrect implementation of $\mathcal{O}$ or noise in the channel between the oracle and the processors. Very recently, the problem of designing nonadaptive group testing algorithms that can tolerate a certain number of erroneous tests has been studied by Balding and Torney [9].

Definition 7.1.1 A set system $(X, \mathcal{A})$ is called (r,s)*-fault-tolerant* if for any $\mathcal{A}', \mathcal{A}'' \subseteq \mathcal{A}$ with $|\mathcal{A}'| \le r$, $|\mathcal{A}''| \le r$, we have

$$\left| \left(\bigcup_{A \in \mathcal{A}'} A \right) \Delta \left(\bigcup_{A \in \mathcal{A}''} A \right) \right| > s,$$

unless $\mathcal{A}' = \mathcal{A}''$.

An important observation made in [9] is the following characterization.

Lemma 7.1.1 (Balding and Torney [9]) Let (X, r, f, Π) be a group testing problem with f the 1-threshold test function, and Π the exact identification criterion. Then $(Y, \mathcal{B})$ is the set system of a nonadaptive algorithm for (X, r, f, Π) that can tolerate up to s erroneous tests if and only if it is (r, s)-fault-tolerant.

In fact, Balding and Torney [9] studied the more stringent set systems which are defined as in Definition 7.1.1, except with the condition $|\mathcal{A}''| \leq r$ removed. These set systems have the additional property that the nonadaptive algorithms they define can detect when the a priori guarantee r is violated. Such set systems have also been studied by Dyachkov, Rykov, and Rashad [52] in the context of random multiple access communication systems. The focus of Dyachkov, Rykov, and Rashad [52], and Balding and Torney [9] is on set systems without any restriction on block sizes. In this chapter, we are concerned with $(2,1)$-fault-tolerant set systems whose block sizes do not exceed three. Such set systems correspond to nonadaptive algorithms using the 1-threshold function that can exactly identify target sets of at most two elements, even in the presence of one erroneous test, and moreover each element is involved in at most three tests. To avoid triviality, we assume the order of the set systems to be at least three.

Let us begin with some easy observations. First, there can be no blocks of size one in any $(2,1)$-fault-tolerant set system $(X, \mathcal{A})$, since the existence of $\{x\} \in \mathcal{A}$ implies that $|A \Delta (\{x\} \cup A)| \leq 1$ for any $A \in \mathcal{A}$. Second, a $(2,1)$-fault-tolerant 2-uniform set system is a graph $G = (V, E)$ consisting of only independent edges, since the existence of edges $e_1 = \{a, b\}$ and $e_2 = \{b, c\}$ in E implies $|e_1 \Delta (e_1 \cup e_2)| = 1$. Hence the maximum number of blocks in a $(2,1)$-fault-tolerant 2-uniform set system of order n is $\lfloor n/2 \rfloor$. Next, we examine the case of 3-uniform set systems. We show that with the same number of blocks in an optimal 2-union-free 3-uniform set system, we can construct a $(2,1)$-fault-tolerant 3-uniform set system.

7.2 (2, 1)-Fault-Tolerant 3-Uniform Set Systems

Let $\beta(n)$ denote the maximum number of blocks in a $(2,1)$-fault-tolerant 3-uniform set system of order n. Obviously, any $(2,1)$-fault-tolerant 3-uniform set system is 2-union-free. It therefore follows from (4.3) that

$$\beta(n) \leq \left\lfloor \frac{n(n-1)}{6} \right\rfloor . \tag{7.1}$$

We show that equality in (7.1) can be met for all n.

Definition 7.2.1 A set system $(X, \mathcal{A})$ is a *quasi-design* $\mathrm{QD}(n, \{3,4\})$, if $|X| = n$, $\mathcal{A} \subset (\binom{X}{3} \cup \binom{X}{2})$, such that

(i) $|A \cap A'| \leq 1$ for all $A, A' \in \mathcal{A}$; and

(ii) there is at most one 2-subset of X that is not contained in any block of $\mathcal{A}$.

The concept of quasi-designs $\mathrm{QD}(n, \{3,4\})$ is first introduced by Frankl and Füredi [62] to settle the existence problem for optimal 2-union-free 3-uniform set systems. We use the same construction as that given in [62] and check that it is $(2,1)$-fault-tolerant.

Lemma 7.2.1 Suppose that $(X, \mathcal{A})$ is a $\mathrm{QD}(n, \{3,4\})$. Let

$$\mathcal{B} = \{A \in \mathcal{A} \mid |A| = 3\} \cup \{\{a,b,c\}, \{a,c,d\} \mid \{a,b,c,d\} \in \mathcal{A}\}.$$

Then $(X, \mathcal{B})$ is a $(2,1)$-fault-tolerant set system with $\lfloor n(n-1)/6 \rfloor$ blocks.

Proof. That $|\mathcal{B}| = \lfloor n(n-1)/6 \rfloor$ has been shown in [62]. Trivially, the symmetric difference of any two blocks in $\mathcal{B}$ contains at least two points.

Suppose that $A, B, C \in \mathcal{B}$, $B \neq C$, such that $|A\Delta(B \cup C)| \leq 1$. If $|B \cap C| \leq 1$, then $|B \cup C| \geq 5$, and hence $B \cup C$ contains at least two points not in A, a contradiction. If $|B \cap C| = 2$, then any block A must intersect $B \cup C$ in at most one point. Hence, A contains two points not in $B \cup C$, a contradiction.

Next, suppose that $A, B, C, D \in \mathcal{B}$, $\{A, B\} \neq \{C, D\}$, such that $|(A \cup B)\Delta(C \cup D)| \leq 1$. We may assume that A, B, C, and D are all distinct, for otherwise we can reduce to the previously considered cases.

Case (i): If $|A \cap B| \leq 1$, then each of C and D must be contained in $A \cup B$. Without loss of generality, $|A \cap C| = 2$ and $|B \cap C| = 1$, where

$$(B \cap C) \not\subset (A \cap C). \tag{7.2}$$

Hence $A \cup C \in \mathcal{A}$. We cannot have $|A \cap D| = 2$ since it would mean $A \cup D \in \mathcal{A}$ and we have two blocks in $\mathcal{A}$, namely $A \cup C$ and $A \cup D$ which intersect in three points. So we must have $|A \cap D| = 1$ and $|B \cap D| = 2$, implying $B \cup D \in \mathcal{A}$. But the block $B \cup D$ contains two points, that in $A \cap D$, and that in $B \cap C$. We claim that these two points are distinct. Suppose not, then $A \cap D = B \cap C$. It follows that $A \cap C \supseteq A \cap B \cap C = A \cap D = B \cap C$, which is impossible by our assumption (7.2). Hence, the two blocks $B \cup D$ and $A \cup C$ of $\mathcal{A}$ intersect in two distinct points, a contradiction.

Case (ii): If $|A \cap B| = 2$, then $A \cup B \in \mathcal{A}$. Since $\{A, B\} \neq \{C, D\}$, at least one of C and D must contain at most one point of $A \cup B$. This block then has two points not in $A \cup B$. □

It was shown in [62] that there exists a QD(n, $\{3, 4\}$) for all n, except when $n = 5$, 6 and 8, and possibly when $n = 20$ and 32. We settle the two remaining cases here.

Lemma 7.2.2 There exists a QD(n, $\{3, 4\}$) for all $n \equiv 2 \pmod 6$, $n \geq 14$.

Proof. Let $(X, \mathcal{G}, \mathcal{A})$ be a $\{3\}$-GDD of type $3^{(n-2)/3}1^1$, which is known to exist [41]. Let $\infty \notin X$ and define $Y = X \cup \{\infty\}$ and $\mathcal{B} = \mathcal{A} \cup \{G \cup \{\infty\} \mid G \in \mathcal{G}$ and $|G| = 3\}$. Then $(Y, \mathcal{B})$ is a QD(n, $\{3, 4\}$). □

Corollary 7.2.1 There exists a QD(n, $\{3, 4\}$) for all $n \in \mathbb{N} \setminus \{5, 6, 8\}$.

Proof. The cases for $n \notin \{20, 32\}$ are settled by Frankl and Füredi [62]. Existence of a QD(20, $\{3, 4\}$) and a QD(32, $\{3, 4\}$) follows from Lemma 7.2.2. □

Corollary 7.2.2 For all n, $\beta(n) = \lfloor n(n-1)/6 \rfloor$.

7.3 (2, 1)-Fault-Tolerant Set Systems With Block Sizes Two and Three

Let $\rho(n)$ denote the maximum number of blocks in a (2, 1)-fault-tolerant set system, $(X, \mathcal{A})$, such that $|A| \in \{2, 3\}$ for every $A \in \mathcal{A}$. From Corollary 7.2.2, we have

$$\rho(n) \geq \left\lfloor \frac{n(n-1)}{6} \right\rfloor.$$

Let $(X, \mathcal{A})$ be a (2, 1)-fault-tolerant set system with block sizes two and three. Let $\mathcal{B} = \{A \in \mathcal{A} \mid |A| = 2\}$. The same argument for 2-uniform set systems shows that $B \cap B' = \varnothing$ for all distinct $B, B' \in \mathcal{B}$. Let $A \in \mathcal{A}$ be any block of size three. Then

$A \cap B = \varnothing$ for all $B \in \mathcal{B}$, since $A \Delta (A \cup B) = B \setminus A$. It follows that if

$$X' = X \setminus \bigcup_{B \in \mathcal{B}} B, \quad \text{and} \quad \mathcal{A}' = \mathcal{A} \setminus \mathcal{B},$$

then $(X', \mathcal{A}')$ is a $(2,1)$-fault-tolerant 3-uniform set system. This gives the inequality

$$\rho(n) \leq \beta(n - 2b) + b,$$

where $b = |\mathcal{B}|$. But

$$\beta(n - 2b) + b = \left\lfloor \frac{n(n-1)}{6} - \frac{2b(n-b-2)}{3} \right\rfloor .$$

Since $\dfrac{2b(n-b-2)}{3} \geq 0$ for all $n \geq 3$, we have $\rho(n) = \lfloor n(n-1)/6 \rfloor$ for all $n \geq 3$. Trivially, this also holds for $n \in \{1, 2\}$. We record this result below.

Lemma 7.3.1 For all positive integers n, $\rho(n) = \lfloor n(n-1)/6 \rfloor$.

The set systems of order n achieving $\rho(n)$ blocks do not contain blocks of size two.

7.4 Remarks

In this chapter, we have seen that optimal $(2,1)$-fault-tolerant 3-uniform set systems are optimal even within the larger class of set systems where blocks of size two are allowed. $(2,0)$-fault-tolerant set systems with block sizes two and three have been characterized by Vakil and Parnes [147]. However, it seems hard to obtain a detailed characterization of $(2,1)$-fault-tolerant 3-uniform set systems. We know from [62] that in such a set system, the number of 2-subsets not contained in any block must be equal to the number of 2-subsets contained in precisely two blocks, and that no 2-subset is contained in more than two blocks. But there remain many flexibilities. Firstly, we can construct many nonisomorphic quasi-designs QD$(n, \{3,4\})$. Secondly, there are many ways to replace the blocks of size four with two blocks of size three.

CHAPTER 8

Erasure-Resilient Codes for Redundant Arrays of Inexpensive Disks

8.1 An Overview of Disk Arrays

A phenomenal increase in processor speed has occurred over the last decade and this trend is likely to continue. Meanwhile, the performance of input/output (I/O) systems has lagged behind. Providing raw processing speed and large memories without balancing I/O capabilities is not sufficient in solving many real-world problems. This imbalance has transformed traditionally computation-bound applications to I/O-bound applications. To achieve application speedup, the bandwidth of I/O systems must be improved. This has led to the development of parallel I/O systems. Issues that must be addressed by any parallel I/O system include storage, support hardware, networking, and software technology.

The most successful approach to the storage problem is an architecture known as a *Redundant Array of Inexpensive Disks* (RAID) [84, 92, 111, 124]. Rather than building one large expensive disk, the RAID architecture increases I/O bandwidth by using a

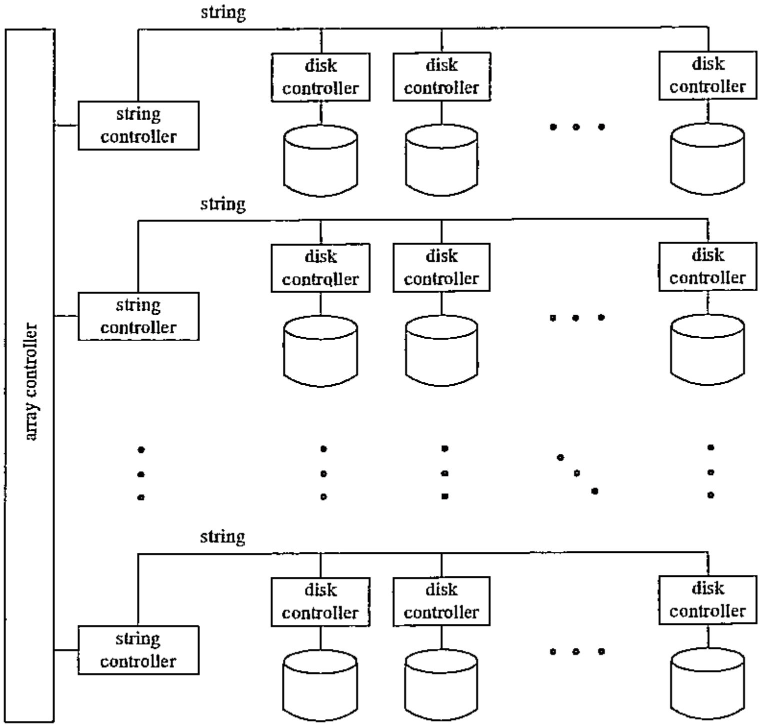


Figure 8.1: A RAID layout.

large array of small magnetic disks linked together as a single data store (see Figure 8.1). Small disks are preferable to large ones because they have a lower cost and consume less power. The idea is to spread data across these small disks (*disk striping*) so that subsequent data access can be done in parallel to reduce access time. Many commercial RAID systems exist today, for example, Fujitsu's DynaRAID, Storage Computer's RAID 7, Sun's SPARCstorage Array, and Thinking Machine Corporation's ScaleArray. It is estimated that the market for RAID systems will exceed thirteen billion dollars by the year 1997 [45].

Large disk arrays, however, are prone to failures, even though each individual disk making up the array may be highly reliable. If the probability of failure of a disk is p, then the probability of failure for a disk array with N disks is $1-(1-p)^N$, assuming that disk failures are uncorrelated. Thus, for p close to zero, a disk array with N disks is about N times more likely to fail than an individual disk. Many applications, notably database and transaction processing, require both high throughput and high data availability of their storage systems. The most demanding of these applications requires continuous operation, which in terms of a storage system requires

(i) the ability to satisfy all user requests for data even in the presence of disk failures, and

(ii) the ability to reconstruct the content of a failed disk onto a replacement disk, thereby restoring itself to a fault-free state.

The solution is to introduce redundancy into the system.

The taxonomy of RAID is based on the amount of redundancies as well as the method of incorporating them. Eight levels of RAID organizations exist at present. We briefly describe these.

RAID Level 0: Offers disk striping with redundancy. Generally not considered a RAID.

RAID Level 1: Uses the traditional method of *mirroring*. All data is copied onto two separate disks. The disadvantage is in the overhead because twice as many physical drives are required. Tolerates one disk failure in the worst case.

RAID Level 2: Uses multiple dedicated parity disks in a Hamming code scheme. All disks are synchronized, which means that all disks must be accessed in parallel. This is ineffective for applications requiring many small reads and writes. For this reason, level 2 RAID is not commercially viable. Tolerates one disk failure in the worst case.

RAID Level 3: Like level 2, disks are synchronized. Data is interleaved bit-wise over the disks. All parity data is stored on a single parity drive.

RAID Level 4: Disks are not synchronized so that multiple reads to disks can be done independently. Data is interleaved block-wise over the disks. All parity data is still stored on a single parity drive.

RAID Level 5: Similar to level 4 RAID, but parity data is spread over all disks.

RAID Level 6: Similar to level 4 RAID but uses Reed-Solomon codes to tolerate up to two disk failures.

RAID Level 7: This is a patented architecture of Storage Computer Corporation that incorporates a totally asynchronous hardware environment with a multi-tiered cache memory controlled by an embedded real-time operating system. Parity data is held on one or more dedicated drives.

One noticeable characteristic of these RAID organizations is that all of them, except level 6, are able to tolerate only one disk failure, and even level 6 can tolerate only two disk failures. This can be a serious problem for mission-critical applications, where very high reliability of data storage is required. This has prompted Hellerstein, Gibson, Karp, Katz, and Patterson [79] to examine coding in RAID that protects against catastrophic disk failures.

When one deals with fault-tolerance issues in date storage systems, it is typical to model the data store as a binary symmetric channel (refer to Section 1.2). This enables one to use techniques from the theory of error-correcting codes to protect against data loss. However, disk controllers can easily identify which disk has failed. This makes the binary erasure channel (Figure 1.3) a more appropriate model for the RAID architecture. The purpose of this chapter is to generalize, as well as to extend, the work of Hellerstein et al. [79]. In particular, we provide a new view of the design of *erasure-resilient codes* for RAID systems, and develop new efficient coding schemes that tolerate a higher number of disk failures than those treated in [79].

Modern large-capacity, high-speed memory units also use erasure-resilient codes for error control [116]. The metrics of interest there are different from those in disk arrays.

8.2 Terminology and Important Metrics

A *data stripe* is the minimum amount of contiguous user data allocated to one disk before any data is allocated to any other disk. The size of a data stripe must be an integral number of sectors, and is often the minimum unit of update used by system software. Because of this, we can view each disk as a collection of data stripes.

Definition 8.2.1 An $[n, c, k]$-*erasure-resilient code* is a function E that encodes n-tuples $D = (D_1, D_2, \dots, D_n)$ of data stripes onto $(n + c)$-tuples $E(D) = (E_1(D), E_2(D), \dots, E_{n+c}(D))$ of data stripes called *codewords* so that any $n+c-k$ data stripes $E_{i_1}(D)$, $E_{i_2}(D)$, $\dots$, $E_{i_{n+c-k}}(D)$ of $E(D)$ together with the indices i_j uniquely determine D.

We often call an $[n, c, k]$-erasure-resilient code a k-erasure-resilient code when the parameters n and c are not important in the context.

To see the relevance of an $[n, c, k]$-erasure-resilient code to the protection of data loss in RAID, suppose that we have a piece of data which is partitioned into an n-tuple $D = (D_1, D_2, \dots, D_n)$ of data stripes. Given an $[n, c, k]$-erasure-resilient code E, we can form the codeword $(E_1(D), E_2(D), \dots, E_{n+c}(D))$ and store this onto a disk array of $n+c$ disks (see Figure 8.2). The definition of an $[n, c, k]$-erasure-resilient code ensures that we can reconstruct the original data in the presence of up to k disk failures. We often call a disk failure an *erasure*, and the failure of a set of k disks a k-*erasure*.

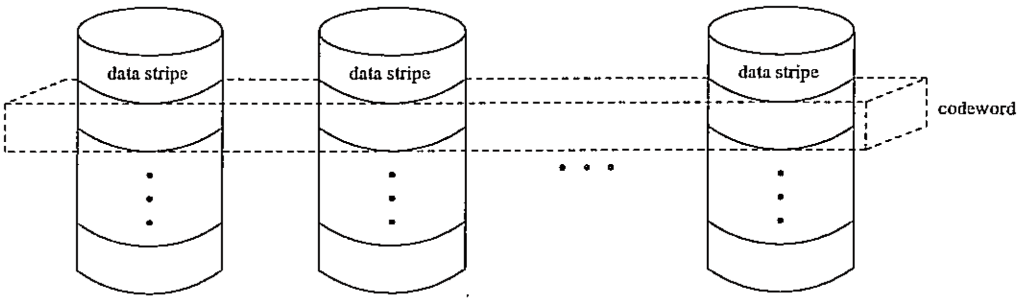


Figure 8.2: Data layout on a disk array.

For performance reasons, we make the following two restrictions on erasure-resilient codes, as in [79].

(i) We restrict ourselves to *systematic* erasure-resilient codes. These are erasure-resilient codes for which $E_i(D) = D_i$ for $1 \leq i \leq n$. The encodings $E_i(D)$ for $n+1 \leq i \leq n+c$ are called *checks*. This means that the encoding leaves the original data unmodified on some disks. This property is desirable to avoid read penalties when there are no disk failures.

(ii) We restrict ourselves to *binary linear* erasure-resilient codes over the field GF(2^L), where L is the bit-size of a data stripe. In this case, each data stripe is interpreted as an L-dimensional vector over GF(2), and E is a linear function. Hence, calculations used to form the encodings are restricted to modulo two arithmetic, that is, parity operations, $\oplus$. This ensures that encodings can be computed efficiently.

Restriction (i) above allows us to separate disks into *information disks*, which contain the original data, and *check disks*, which contain the parity checks. In fact, both restrictions (i) and (ii) imply that an $[n, c, k]$-erasure-resilient code can be described in terms of a $c \times (n + c)$ matrix $H = [C \mid I]$ over GF(2), where I is the $c \times c$ identity matrix and C is a $c \times n$ matrix that determines the equations for the checks. This is a well-known result in the theory of error-correcting codes [96]. The matrix H is called the *parity-check matrix* of the code. Given the parity-check matrix $H = [C \mid I]$ of a k-erasure-resilient code, we can think of the rows of C (as well as the rows and columns of I) as being indexed by the check disks of a disk array, and the columns of C as being indexed by the information disks. The content of check disk i is the modulo two sum of the content of those information disks, whose columns they index in C have a one in row i.

We consider the following metrics for the performance of an erasure-resilient code [79].

Check disk overhead: This is the ratio of the number of check disks to information disks. An $[n, c, k]$-erasure-resilient code has a check disk overhead of c/n.

Update penalty: This is the number of check disks whose content must be changed when a change is made in the content of a given information disk. We call these disks the *check disks associated with the information disk.* If N check disks need to be involved in every write, then the parallelism of the disk array is reduced by a factor of $N + 1$. Since parallelism is the reason behind using disk arrays, update penalties should be kept as small as possible. The update penalties of an erasure-resilient code are the numbers of ones in the columns of its parity-check matrix.

Group size: This is the number of disks that must be accessed during the reconstruction of a single failed disk. The cost of reconstruction makes small group size desirable, while for load balancing reasons, uniform group size is desirable. The group sizes of an erasure-resilient code are the numbers of ones in the rows of its parity-check matrix.

Since updates of data are usually much more frequent than the reconstruction of data due to disk failures, the update penalties are typically of more concern than the group size.

Another assumption we make is that disk failures are uncorrelated. This assumption is valid for *catastrophic failures*, which are head crashes or failures of the read/write or controller electronics [79]. It should be pointed out that disk failures can be correlated. For example, the disks on a string are usually connected to the same power supply. So the failure of the power supply causes all disks on the string to fail simultaneously. We refer the reader to [106] for more information on this topic. Our interest in this chapter is solely on uncorrelated disk failures.

8.3 Properties of Parity-Check Matrices

Let us consider the failure of k disks (both information disks and check disks can fail). If $H = [C \mid I]$ has a set of k or fewer linearly dependent columns (over GF(2)), then the failure of the corresponding disks makes reconstruction of data impossible. In fact, this is the only scenario for which disk failures are irrecoverable.

Lemma 8.3.1 (Hellerstein et al. [79]) A set of disk failures is recoverable if and only if the corresponding set of columns in the parity-check matrix is linearly independent.

Therefore, H is the parity-check matrix of a k-erasure-resilient code if and only if every set of k columns of H contains no nonempty set of linearly dependent columns. Precisely the same condition determines when H is the parity-check matrix of a k-error-detecting code [96].

Corollary 8.3.1 A code is k-erasure-resilient if and only if it is k-error-detecting.

This equivalence between k-erasure-resilient and k-error-detecting codes means that results on error-detecting codes can be brought to bear. However, the study of codes for error detection has not focussed on the metrics discussed in the previous section. Indeed, as observed in [79], many of these codes are not suitable for disk arrays because they have large update penalties. Recently, erasure-resilient codes have also been constructed to combat bursty losses in packet-switched networks [3, 7]. Again, the metrics of interest there are different from those in disk array applications.

Corollary 8.3.2 $H = [C \mid I]$ is the parity-check matrix of a k-erasure-resilient code if and only if for every $t \leq k$ columns, $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_t$ of C, the vector $\mathbf{x} = \mathbf{c}_1 \oplus \mathbf{c}_2 \oplus \cdots \oplus \mathbf{c}_t$ has weight at least $k + 1 - t$.

Proof. The condition is exactly what is needed for every set of at most k columns of H to be linearly independent. □

We have earlier discussed the importance of update penalties. It is easy to see that if an erasure-resilient code is able to tolerate k erasures, then every update must affect the content of at least $k+1$ disks (one information disk and k check disks). Thus, the update penalties of a k-erasure-resilient code are at least k. Henceforth, we consider only those k-erasure-resilient codes for which the update penalties are all equal to k, the minimum possible. We speak, therefore, of the update *penalty*, instead of the update *penalties* of an erasure-resilient code. The corresponding parity-check matrix $H = [C \mid I]$ has column sums for C all equal to k.

Although an erasure-resilient code with update penalty k cannot tolerate all $(k+1)$-erasures, it can certainly tolerate some of them. Indeed, a $(k+1)$-erasure is irrecoverable if and only if it corresponds to the failure of an information disk and its k associated check disks. We call such $(k+1)$-erasures *bad.* It is observed in [79] that with update penalty k, one can nonetheless hope to tolerate *all* $(k+1)$-erasures, except for bad ones. In fact, it can happen that all t-erasures are recoverable except for those that contain bad $(k+1)$-erasures.

Definition 8.3.1 A t-erasure, where $t \geq k+1$, is called *bad* if it includes the failure of an information disk and its k associated check disks.

With this in mind, we extend Definition 8.2.1 to encompass this notion of higher resilience.

Definition 8.3.2 An $[n, c, k, l]$-*erasure-resilient code* is an $[n, c, k]$-erasure-resilient code which can tolerate all t-erasures, for $k+1 \leq t \leq l$, except for bad t-erasures.

We often write (k, l)-erasure-resilient code for $[n, c, k, l]$-erasure-resilient code when the parameters n and c are not important in the context. Requirements for higher reliability of disk arrays make (k, l)-erasure-resilient codes attractive. Note that an $[n, c, k, k]$-erasure-resilient code is simply an $[n, c, k]$-erasure-resilient code. Corollary 8.3.2 can be extended as follows to handle the more general (k, l)-erasure-resilient codes.

Lemma 8.3.2 $H = [C \mid I]$ is the parity-check matrix of a (k, l)-erasure-resilient code if and only if for every t columns, $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_t$ of C, where $2 \leq t \leq l$, the vector $\mathbf{x} = \mathbf{c}_1 \oplus \mathbf{c}_2 \oplus \cdots \oplus \mathbf{c}_t$ has weight at least $l+1-t$.

Proof. First we prove necessity. Suppose there exists $\mathbf{x} = \mathbf{c}_1 \oplus \mathbf{c}_2 \oplus \cdots \oplus \mathbf{c}_t$ for some columns $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_t$ of C, such that $\mathrm{wt}(\mathbf{x}) \leq l-t$. Then there exists $\mathrm{wt}(\mathbf{x})$ columns of I whose sum together with $\mathbf{x}$ gives the zero vector. Hence, the corresponding s-erasure, where $s = \mathrm{wt}(\mathbf{x}) + t \leq l$, cannot be recovered. We may assume that this s-erasure is not bad, for otherwise we may discard information disks and their k associated check disks from this s-erasure and obtain an s'-erasure, for some $s' < s$, which is still irrecoverable.

For sufficiency, suppose on the contrary that there exists an r-erasure which is irrecoverable. Then there exist columns $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_t$ of C and columns $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_s$ of I, such that $\mathbf{c}_1 \oplus \mathbf{c}_2 \oplus \cdots \oplus \mathbf{c}_t \oplus \mathbf{e}_1 \oplus \mathbf{e}_2 \oplus \cdots \oplus \mathbf{e}_s = \mathbf{0}$ and $t + s = r$. This is possible if and only if the weight of $\mathbf{x} = \mathbf{c}_1 \oplus \mathbf{c}_2 \oplus \cdots \oplus \mathbf{c}_t$ is exactly s. Hence, we have $\mathrm{wt}(\mathbf{x}) = r - t \leq l - t$, a contraduction. □

Before we leave this section, let us make the following definition.

Definition 8.3.3 Given c, k, and l, define $F(c,k,l)$ to be the maximum n such that there exists an $[n,c,k,l]$-erasure-resilient code.

An $[n,c,k,l]$-erasure-resilient code with $n = F(c,k,l)$ is said to have *optimal check disk overhead.* We also abbreviate $F(c,k,k)$ to $F(c,k)$.

8.4 Turán-Type Problems in Erasure-Resilient Codes

Given any matrix $M \in \{0,1\}^{m \times n}$, one can define a set system $(X, \mathcal{A})$, where $X = \{1,2,\ldots,m\}$ and $\mathcal{A}$ contains precisely the supports of the columns of M. We call $(X,\mathcal{A})$ *the set system associated with the matrix* M.

Definition 8.4.1 Let $(X,\mathcal{A})$ be a set system. The *replication number* of a point $x \in X$ is $r_x = |\{A \in \mathcal{A} \mid x \in A\}|$.

Let $H = [C \mid I]$ be the parity-check matrix of an erasure-resilient code. The set system associated with C is called *the set system of the erasure-resilient code.* If $(X,\mathcal{A})$ is the set system of an $[n,c,k,l]$-erasure-resilient code, then with our foregoing assumptions, $(X,\mathcal{A})$ is k-uniform, $|X| = c$, $|\mathcal{A}| = n$ (and therefore the check disk overhead is $|X|/|\mathcal{A}|$), and the group sizes are one more than the corresponding replications numbers. It is this correspondence between set systems and parity-check matrices that gives rise to Turán-type problems in erasure-resilient codes.

Lemma 8.4.1 $(X,\mathcal{A})$ is the set system of a (k,l)-erasure-resilient code if and only if it satisfies the following condition. For any $2 \leq t \leq l$, there do not exist t blocks $A_1, A_2, \ldots, A_t$

in $\mathcal{A}$ such that $|A_1 \Delta A_2 \Delta \cdots \Delta A_t| \leq l - t$.

Proof. Simply translate Lemma 8.3.2 into the language of set systems and observe that $\text{supp}(\mathbf{u} \oplus \mathbf{v}) = \text{supp}(\mathbf{u}) \Delta \text{supp}(\mathbf{v})$ for any two vector $\mathbf{u}, \mathbf{v}, \in \{0,1\}^n$. □

Lemma 8.4.1 implies that the construction of a (k,l)-erasure-resilient code with optimal check disk overhead is precisely the Turán-type problem of determining the maximum number of blocks in a set system satisfying the condition of Lemma 8.4.1.

When considering (k,l)-erasure-resilient codes, we may assume $l \leq 2k - 1$ for the following reason. Let $(X, \mathcal{A})$ be the set system of a (k,l)-erasure-resilient code. If $\mathcal{A}$ contains at least two blocks A and A' with nonempty intersection, then $|A \Delta A'| \leq 2k - 2$. It follows from Lemma 8.4.1 that $l - 2 < 2k - 2$, and this implies $l \leq 2k - 1$. Hence, if $l \geq 2k$, then $\mathcal{A}$ must consist of pairwise disjoint blocks. This corresponds to the scheme where the data on each information disk is replicated on k different check disks. This scheme is able to tolerate t-erasures for all t, except for bad ones. For fixed update penalty k, this scheme has the highest reliability, but suffers from a huge check disk overhead of k. Henceforth, we restrict our attention to $l \leq 2k - 1$.

In the next section, we give a general construction for $[n, c, k, l]$-erasure-resilient codes and establish a limit on how good an $[n, c, k, l]$-erasure-resilient code can be.

8.5 An Expander-Based Construction and an Upper Bound

Given a set system $(X, \mathcal{A})$, one can construct a bipartite graph $G = (X \,\dot{\cup}\, \mathcal{A}, E)$ as follows. The vertex sets of the bipartition are X and $\mathcal{A}$. Two vertices $x \in X$ and $A \in \mathcal{A}$ are adjacent if and only if $x \in A$. This graph is called the *point-block incidence graph of* $(X, \mathcal{A})$. It is easy to see that $(X, \mathcal{A})$ can be reconstructed from its point-block incidence graph.

Definition 8.5.1 Let S be a subset of vertices in a graph. The *neighbourhood* of S, denote $N(S)$, is the set of all vertices not in S that are adjacent to some vertex in S. The elements of $N(S)$ are called the *neighbours* of S.

Definition 8.5.2 Let S be a subset of vertices in a graph. A vertex v is an *odd neighbour* of S if v is adjacent to an odd number of vertices in S.

Lemma 8.5.1 Let $1 \leq k \leq l$ and $2 \leq t \leq l$. Let $G = (U \dot{\cup} V, E)$ be a bipartite graph where each vertex in U has degree k, and such that for any subset $T \subseteq V$, $|T| = t$, we have

$$|N(T)| \geq \frac{t(k-1)+l+1}{2}.$$

Then G is the point-block incidence graph of a set system of an (k,l)-erasure-resilient code.

Proof. From Lemma 8.4.1, it suffices to show that any subset T of t vertices from V has at least $l+1-t$ odd neighbours. Suppose that there are only $s \leq l-t$ odd neighbours of T. Then there are $|N(T)| - s$ neighbours of T, each of which is adjacent to at least two vertices of T. Hence,

$$2(|N(T)| - s) + s \leq tk,$$

which gives

$$\begin{aligned} |N(T)| &\leq \frac{tk+s}{2} \\ &\leq \frac{tk+l-t}{2} \\ &= \frac{t(k-1)+l}{2}. \end{aligned}$$

This is a contradiction. □

Lemma 8.5.1 shows that bipartite graphs for which the neighbourhood of any set of vertices S is large relative to the size of S give erasure-resilient codes. This property is indeed what defines a special class of graphs known as *expanders* [16, 66]. Expanders are useful in many theoretical as well as practical applications in computer science. Unfortunately, the study of (bipartite) expanders have focussed on the case when the sizes of the two partitions are linearly related [4, 94, 97, 113, 143]. This gives trivial results in our application. The probabilistic construction we give next yields bipartite expanders where the sizes of the two partitions are polynomially related. The construction is a modification of the usual probabilistic construction for expanders (see [100]).

Theorem 8.5.1 Let k and l be constants such that $1 \leq k \leq l$, and define $\alpha = (2k+1-l)/4$. Let $2 \leq t \leq l$. There is an integer n_0 such that for all $n > n_0$, there exists a bipartite graph $G = (U \dot{\cup} V, E)$ with $|U| = n$ and $|V| = \Omega(n^\alpha)$ satisfying the following two conditions:

(i) each vertex in V has degree k;

(ii) for every subset T of t vertices from V, we have $|N(T)| \geq (t(k-1)+l+1)/2$.

Proof. Let $|V| = dn^\alpha$ for some positive constant d. Consider a random bipartite graph on the vertices in U and V, in which each vertex of V chooses its k neighbours by sampling a k-subset of vertices from U independently and uniformly from $\binom{U}{k}$. It is clear that the bipartite graph so constructed satisfies condition (i).

Let $\mathcal{E}_t$ denote the event that a subset of t vertices of V has fewer than $s = (t(k-1)+l+1)/2$ neighbours in U. Fix any subset $T \subseteq V$ of size t and any subset $S \subseteq U$ of size s. There are $\binom{dn^\alpha}{t}$ ways of choosing T and $\binom{n}{s}$ ways of choosing S. The probability that S contains $N(T)$ is $(\binom{s}{k}/\binom{n}{k})^t$. Thus, the probability of the event that all the edges emanating from some t vertices of V fall within any s vertices of U is bounded as follows:

$$\mathbf{Pr}[\mathcal{E}_t] \leq \binom{dn^\alpha}{t}\binom{n}{s}\left[\frac{\binom{s}{k}}{\binom{n}{k}}\right]^t .$$

Using the inequalities $\binom{n}{k} \leq (ne/k)^k$ and $\binom{n}{k} \geq (n/k)^k$, we obtain

$$\mathbf{Pr}[\mathcal{E}_t] \leq O(n^{-(t-2)(l+1)/4}).$$

The probability that the bipartite graph fails to satisfy (ii) is at most

$$\sum_{t=2}^{l} \mathbf{Pr}[\mathcal{E}_t],$$

which can be made to be less than one for n large enough by an appropriate choice of d. The desired result follows. □

Next, we establish an upper bound on $F(c,k,l)$.

Theorem 8.5.2 Let k, and l be constants such that $1 \leq k \leq l$. Then $F(c,k,l) = O(c^{k+1-\lfloor l/2 \rfloor})$.

Proof. Consider all the configurations of two blocks of size k intersecting in at least $k+1-\lfloor l/2 \rfloor$ points. Any set system $(X,\mathcal{A})$ for an $[n,c,k,l]$-erasure-resilient code must avoid all such configurations, for otherwise it would violate the condition of Lemma 8.3.2. Hence, any two blocks of $(X,\mathcal{A})$ intersect in at most $k-\lfloor l/2 \rfloor$ points. It follows that

$(X, \mathcal{A})$ is a $(k+1-\lfloor l/2 \rfloor)$-$(c, k, 1)$ packing. Hence,

$$|\mathcal{A}| \leq D(c, k, k+1-\lfloor l/2 \rfloor) \leq \frac{\binom{c}{k+1-\lfloor l/2 \rfloor}}{\binom{k}{k+1-\lfloor l/2 \rfloor}} = O(c^{k+1-\lfloor l/2 \rfloor}).$$

□

Theorem 8.5.1 and Theorem 8.5.2 give the following.

Corollary 8.5.1 For any fixed k and l such that $1 \leq k \leq l$, there exist positive constants a_1 and a_2 such that

$$a_1 c^{(2k+1-l)/4} \leq F(c, k, l) \leq a_2 c^{k+1-\lfloor l/2 \rfloor},$$

for all $c \in \mathbb{N}$.

For general k, the only lower bound on $F(c, k, l)$ obtained by Hellerstein et al. [79] is for the case $l = k$. We give a new short proof here.

Theorem 8.5.3 (Hellerstein et al. [79]) For any $k \in \mathbb{N}$, we have $F(c, k) \geq (1 - o(1))\binom{c}{2}/\binom{k}{2}$.

Proof. It is easy to see that every $(k-1)$-cover-free set system is the set system of a k-erasure-resilient code. By Lemma 5.1.1, any 2-$(c, k, 1)$ packing is $(k-1)$-cover-free. Hence, $F(c, k) \geq D(c, k, 2) = (1 - o(1))\binom{c}{2}/\binom{k}{2}$; the last equality being from [59]. □

The (k, l)-erasure-resilient codes we built from expanders are at least as reliable and have asymptotically better check disk overheads than that provided by Theorem 8.5.3 as long as $k \leq l \leq 2k - 8$.

The exponent in the upper bound of Corollary 8.5.1 is about twice that for the lower bound. We believe the upper bound to be the true asymptotic behaviour of $F(c,k,l)$, but tightening the lower bound in general appears to be difficult. We can give tight bounds for several cases when k is small.

8.6 $(3,l)$-Erasure-Resilient Codes

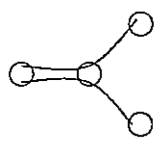
An extensive treatment of $(3,l)$-erasure-resilient codes, for $l=3$ and 4, was given in [79]. We summarize their results below.

Lemma 8.6.1 (Hellerstein et al. [79]) $(X, \binom{X}{3})$ is the set system of a 3-erasure-resilient code with optimal check disk overhead. Hence, $F(c,3)=\binom{c}{3}$.

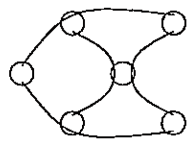
Lemma 8.6.2 (Hellerstein et al. [79]) For all $c \in \mathbb{N}$, $F(c,3,4) \leq c(c-1)/6$, with equality if $c=3^a$ for some nonnegative integer a. If $c \equiv 3 \pmod 6$, there exists a $[c(c-3)/6, c, 3, 4]$-erasure-resilient code.

We can improve on Lemma 8.6.2 by examining the set system of a $(3,4)$-erasure-resilient code. First, consider the configuration P_1 in Figure 8.3(a) for which the symmetric difference of its two blocks has size two. By Lemma 8.3.2, this configuration must be avoided by the set system of any $(3,4)$-erasure-resilient code. For $3 \leq t \leq 4$, the only configuration of t blocks of size 3 for which their symmetric difference has at most $4-t$ points and which does not contain P_1 is that given in Figure 8.3(b). Lemma 8.3.2 implies that P_2 must also be avoided in the set system of any $(3,4)$-erasure-resilient code.

Forbidding P_1 from the set system $(X,\mathcal{A})$ of an $[n,c,3,4]$-erasure-resilient code is equivalent to saying that $(X,\mathcal{A})$ is a 2-$(c,3,1)$ packing. The configuration P_2 is known in the design theory literature under various names: *quadrilateral*, *Pasch configuration*,



(a) P_1



(b) P_2

Figure 8.3: Forbidden configurations for $(3,4)$-erasure-resilient codes.

fragment, or *arrow* (see [42]). A 2-$(c,3,1)$ packing that does not contain a Pasch configuration is called *anti-Pasch*. The construction of $(3,4)$-erasure-resilient codes with optimal check disk overhead is therefore equivalent to the following problem.

Problem 8.6.1 Determine the maximum number of blocks in an anti-Pasch 2-$(v,3,1)$ packing.

An anti-Pasch 2-$(v,3,1)$ packing with $D(v,3,2)$ blocks is said to be *optimal*.

A complete solution to Problem 8.6.1 is not known. We believe that for all sufficiently large v, there exists an optimal anti-Pasch 2-$(v,3,1)$ packing. The simple observation below shows that it is sufficient to treat the cases $v \equiv 1$, 3, or 5 (mod 6).

Lemma 8.6.3 Let $v \equiv 1$, 3, or 5 (mod 6). If there exists an optimal anti-Pasch 2-$(v,3,1)$ packing, then there exists an optimal anti-Pasch 2-$(v-1,3,1)$ packing.

Proof. Schönheim [127, 128] and Spencer [136] have shown that for $v \equiv 1$, 3, or 5 (mod 6), an optimal 2-$(v-1,3,1)$ packing $(X,\mathcal{A})$ can be constructed from an optimal 2-$(v,3,1)$ packing $(Y,\mathcal{B})$ as follows. Pick an element $y \in Y$ that is contained in the least number of blocks in $\mathcal{B}$, breaking ties arbitrarily. Take $X = Y \setminus \{y\}$ and $\mathcal{A} = \{B \in \mathcal{B} \mid y \in B\}$. Since $\mathcal{A} \subseteq \mathcal{B}$, it is clear that $(X,\mathcal{A})$ does not contain a Pasch configuration if $(X,\mathcal{B})$ does not. □

Example 8.6.1 An optimal anti-Pasch 2-(17, 3, 1) packing $(X, \mathcal{A})$: X is taken to be {0, 1, ..., 16} and $\mathcal{A}$ contains the following 3-subsets of X.

$\{6,8,11\}$	$\{3,7,13\}$	$\{2,5,7\}$	$\{3,11,12\}$	$\{3,5,16\}$	$\{4,8,13\}$	$\{0,10,11\}$
$\{1,5,12\}$	$\{2,10,13\}$	$\{1,8,16\}$	$\{5,11,14\}$	$\{1,9,15\}$	$\{0,7,8\}$	$\{4,12,16\}$
$\{6,7,15\}$	$\{7,9,12\}$	$\{12,14,15\}$	$\{3,8,15\}$	$\{2,4,6\}$	$\{3,9,10\}$	$\{2,11,15\}$
$\{0,1,13\}$	$\{1,7,11\}$	$\{0,15,16\}$	$\{0,2,12\}$	$\{2,9,16\}$	$\{0,3,14\}$	$\{1,10,14\}$
$\{5,13,15\}$	$\{1,3,6\}$	$\{2,8,14\}$	$\{4,7,14\}$	$\{11,13,16\}$	$\{4,9,11\}$	$\{4,10,15\}$
$\{7,10,16\}$	$\{8,10,12\}$	$\{9,13,14\}$	$\{6,12,13\}$	$\{5,8,9\}$	$\{6,14,16\}$	$\{0,4,5\}$
$\{5,6,10\}$	$\{0,6,9\}$					

When $v \equiv 1$ or 3 (mod 6), an optimal 2-$(v, 3, 1)$ packing is a Steiner triple system, STS(v). Already twenty years ago, Erdös [56] made the conjecture that there exists an anti-Pasch STS(v) for all $v \equiv 1$ or 3 (mod 6) whenever v is sufficiently large. The unique STS(7) and the two nonisomorphic STS(13) contain Pasch configurations. Brouwer [21] refined Erdös' conjecture as follows.

Conjecture 8.6.1 (Brouwer [21]) There exists an anti-Pasch STS(v) for all $v \equiv 1$ or 3 (mod 6), except when $v = 7$ or 13.

Conjecture 8.6.1 is known to be true for $v \equiv 3$ (mod 6).

Theorem 8.6.1 (Brouwer [21], Griggs, Murphy, and Phelan [72]) There exists an anti-Pasch STS(v) for all $v \equiv 3$ (mod 6).

The results for $v \equiv 1$ (mod 6) is more fragmented and we refer the reader to [42] for a survey. It appears that Griggs has recently constructed anti-Pasch STS(v) for a large fraction of $v \equiv 1$ (mod 6). So by observing the equivalence between $(n, c, 3, 4)$-erasure-resilient codes with optimal check disk overhead and optimal anti-Pasch 2-$(c, 3, 1)$ packings, we can easily improve Lemma 8.6.2 as follows.

Lemma 8.6.4 For all $c \in \mathbb{N}$, we have $F(c, 3, 4) \leq D(c, 3, 2)$, with equality if $c \equiv 2$ or 3 (mod 6).

Proof. Follows from Theorem 8.6.1 and Lemma 8.6.3. □

We now turn our attention to $(3, 5)$-erasure-resilient codes. It turns out that there are no additional configurations to P_1 and P_2 which must be avoided by the set system of an $(3, 5)$-erasure-resilient code. Consequently, every $(3, 4)$-erasure-resilient code is a $(3, 5)$-erasure-resilient code.

Lemma 8.6.5 For all $c \in \mathbb{N}$, we have $F(c, 3, 5) = F(c, 3, 4)$.

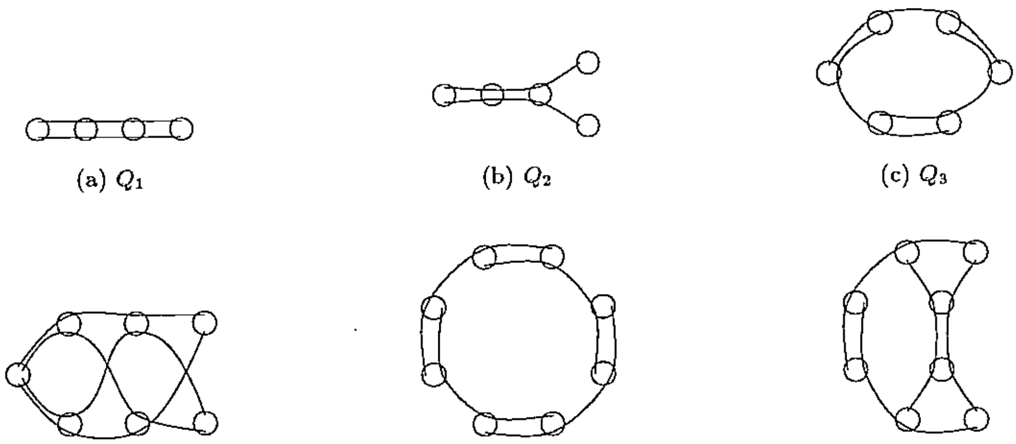
8.7 $(4, l)$-Erasure-Resilient Codes

The only previously-known result concerning $(4, l)$-erasure-resilient codes is the lower bound $F(c, 4) \geq c(c-1)/12$ given by Theorem 8.5.3. Hellerstein et al. [79] posed the open problem of determining $F(c, 4)$.

8.7.1 The Cases $l = 4$ and $l = 5$

The proof of Theorem 8.5.2 shows that any set system $(X, \mathcal{A})$ of an $[n, c, 4]$-erasure-resilient code must avoid the two configurations Q_1 and Q_2 in Figure 8.4, and hence is a 3-$(c, 4, 1)$ packing[1]. Therefore, $F(c, 4) \leq D(c, 4, 3)$. But being a 3-$(c, 4, 1)$ packing is not sufficient. Lemma 8.3.2 implies that $(X, \mathcal{A})$ must further avoid the four configurations Q_3, Q_4, Q_5, and Q_6 in Figure 8.4. It follows that $F(c, 4) = c(c-1)(c-2)/24$ if and only

[1]Our definition of a set system (see Section 2.3) automatically excludes the configuration Q_1. This configuration is given here to remind the reader that Q_1 must be avoided even if set systems with repeated blocks are allowed.



(a) Q_1 (b) Q_2 (c) Q_3 (d) Q_4 (e) Q_5 (f) Q_6

Figure 8.4: Forbidden configurations for 4-erasure-resilient codes.

if there exists a 3-$(c, 4, 1)$ design (known as a *Steiner quadruple system of order* c and denoted SQS(c)) that avoids all the configurations Q_3, Q_4, Q_5, and Q_6. At present, we do not know of any example of a nontrivial SQS(c) that avoids all these configurations. For a comprehensive survey on Steiner quadruple systems, we refer the reader to [76].

Here, we address the more difficult problem of constructing $(4, 5)$-erasure-resilient codes, and in the process, obtain asymptotically-tight bounds (up to constant factors) on both $F(c, 4)$ and $F(c, 4, 5)$. Let $(X, \mathcal{A})$ be the set system of an $[n, c, 4, 5]$-erasure-resilient code. Naturally, $(X, \mathcal{A})$ is a 3-$(c, 4, 1)$ packing that avoids the four configurations Q_3, Q_4, Q_5, and Q_6. A short computation demonstrates that there are precisely nine other configurations that must be avoided. These configurations are shown in Figure 8.5.

The remainder of this section discusses a finite field construction for $(4, 5)$-erasure-resilient codes.

Definition 8.7.1 A set system $(X, \mathcal{A})$ is *k-partite* if there is a partition of X into k parts, $X = X_1 \dot\cup X_2 \dot\cup \cdots \dot\cup X_k$, such that for every block $A \in \mathcal{A}$, we have $|A \cap X_i| \leq 1$ for $1 \leq i \leq k$.

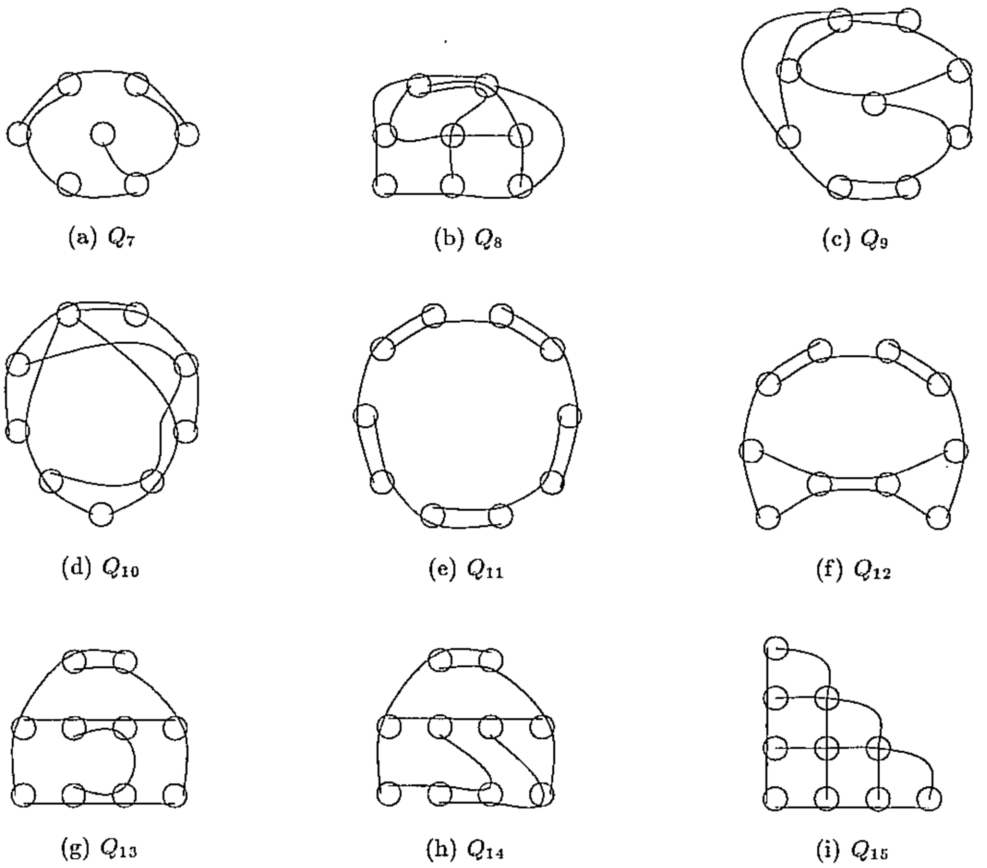


Figure 8.5: Forbidden configurations for $(4, 5)$-erasure-resilient codes.

One idea we use to simplify our construction is to restrict our attention to set systems of (4, 5)-erasure-resilient codes that are 4-partite. It is known [60] that for every k-uniform set system $(X, \mathcal{A})$, one can find a k-partite set system $(X, \mathcal{B})$, where $\mathcal{B} \subseteq \mathcal{A}$, such that $|\mathcal{B}| \geq \frac{k!}{k^k}|\mathcal{A}|$. So our restriction to 4-partite set systems is not a severe one and affects $F(c, 4, l)$ by at most a constant factor of 32/3. It is easy to verify that the configurations Q_3, Q_7, Q_8, Q_9, Q_{10}, Q_{11}, Q_{12}, Q_{13}, Q_{14}, and Q_{15} are not 4-partite. Hence, they cannot be present in any 4-partite set system. It therefore suffices to construct 4-partite set systems that do not contain any of the configurations Q_1, Q_2, Q_4, Q_5, and Q_6.

Definition 8.7.2 An *extension of a set system* $(X, \mathcal{A})$ *by a point* $\infty \notin X$ is the set system $(X \cup \{\infty\}, \mathcal{B})$, where $\mathcal{B} = \{A \cup \{\infty\} \mid A \in \mathcal{A}\}$.

We now describe the finite field construction. Let q be an odd prime power and let ω be a primitive element of GF(q). For each i, $1 \leq i \leq (q-1)/2$, define a set system $(X_i, \mathcal{B}_i)$, where

$$X_i = \mathrm{GF}(q) \times \{0, 1, 2\}, \qquad \text{and}$$
$$\mathcal{B}_i = \{\{(a, 0), (b, 1), (a + \omega^i b, 2)\} \mid a, b \in \mathrm{GF}(q) \text{ and } b \neq 0\}.$$

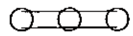
Now let $(Y_i, \mathcal{C}_i)$ be the extension of $(X_i, \mathcal{B}_i)$ by the point ∞_i, for $1 \leq i \leq (q-1)/2$. Finally, define $(Y, \mathcal{C})$ so that

$$Y = \bigcup_{i=1}^{(q-1)/2} Y_i \quad \text{and} \quad \mathcal{C} = \bigcup_{i=1}^{(q-1)/2} \mathcal{C}_i.$$

The next lemma shows that $(Y, \mathcal{C})$ is a set system, that is, $(Y, \mathcal{C})$ avoids the configuration Q_1.

Lemma 8.7.1 The pair $(Y, \mathcal{C})$ is a set system.

Proof. If $(Y, \mathcal{C})$ contains the configuration Q_1, then it would mean that some $(X_i, \mathcal{B}_i)$ contains the configuration below.

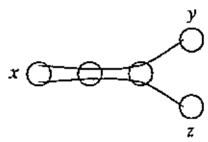


But this contradicts the fact that $(X_i, \mathcal{B}_i)$ is a 2-$(3q, 3, 1)$ packing. □

It is now clear that $(Y, \mathcal{C})$ is a 4-uniform set system. Since each block in $\mathcal{C}$ intersects each of the sets $\mathrm{GF}(q) \times \{0\}$, $\mathrm{GF}(q) \times \{1\}$, $\mathrm{GF}(q) \times \{2\}$, and $\{\infty_1, \infty_2, \ldots, \infty_{(q-1)/2}\}$ in exactly one point, and these sets partition Y, $(Y, \mathcal{C})$ is also 4-partite. The sequence of lemmata below shows that $(Y, \mathcal{C})$ avoids several other configurations.

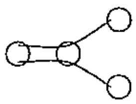
Lemma 8.7.2 The set system $(Y, \mathcal{C})$ avoids the configuration Q_2.

Proof. Suppose $(Y, \mathcal{C})$ contains the configuration below.



Without loss of generality, either $x = \infty_i$, or $y = \infty_i$ and $z = \infty_j$, for some $i \neq j$.

If $x = \infty_i$, then some $(X_i, \mathcal{B}_i)$ must contain the following configuration.

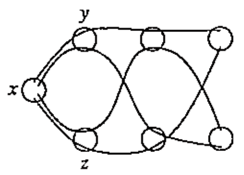


But this contradicts the fact that $(X_i, \mathcal{B}_i)$ is a 2-$(3q, 3, 1)$ packing.

If $y = \infty_i$ and $z = \infty_j$, then there exists $\{(a, 0), (b, 1), (c, 2)\} \in \mathcal{B}_i \cap \mathcal{B}_j$. This is only possible if $b = 0$. But the only set system that contains blocks of the form $\{(a, 0), (0, 1), (c, 2)\}$ is $(X_1, \mathcal{B}_1)$. This is a contradiction. □

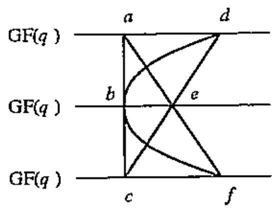
Lemma 8.7.3 The set system $(Y, \mathcal{C})$ avoids the configuration Q_4.

Proof. Suppose $(Y, \mathcal{C})$ contains the configuration below.



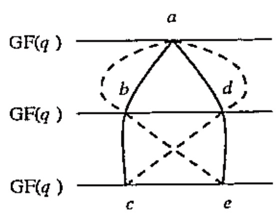
Without loss of generality, either $x = \infty_i$, or $y = \infty_i$ and $z = \infty_j$, for some $i \neq j$.

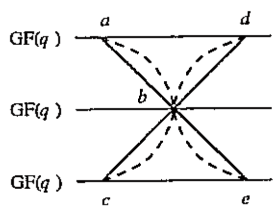
If $x = \infty_i$, then $(X_i, \mathcal{B}_i)$ contains the Pasch configuration. The only way a Pasch configuration can occur in $(X_i, \mathcal{B}_i)$ is as follows.

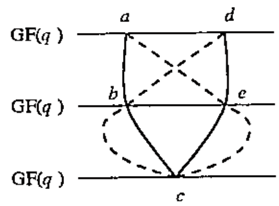


But this implies $c = a + \omega^i b = d + \omega^i e$ and $f = a + \omega^i e = d + \omega^i b$, which can only be satisfied if $b = e$. This is a contradiction.

If $y = \infty_i$ and $z = \infty_j$, then $(X_i, \mathcal{B}_i)$ and $(X_j, \mathcal{B}_j)$ must contain four blocks (two from $\mathcal{B}_i$ and two from $\mathcal{B}_j$) that occur in one of the following three ways.





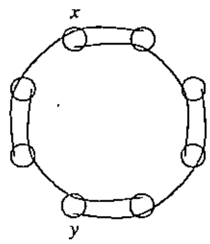


The blocks in $\mathcal{B}_i$ are shown in solid lines and the blocks in $\mathcal{B}_j$ are shown in dashed lines. In the first situation, we have $c = a + \omega^i b = a + \omega^j d$ and $e = a + \omega^i d = a + \omega^j b$, which can only be satisfied if $b - d$. In the second situation, we have $c = d + \omega^i b = a + \omega^j b$ and $e = a + \omega^i b = d + \omega^j b$, which can only be satisfied if $a = d$. For the last situation, we have $c = a + \omega^i b = d + \omega^i e = a + \omega^j e = d + \omega^j b$, which can only be satisfied if $b = e$. All these lead to contradictions.

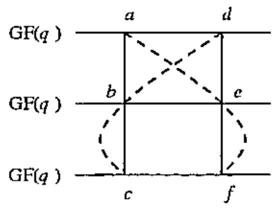
It follows that $(Y, \mathcal{C})$ cannot contain the configuration Q_4. □

Lemma 8.7.4 The set system $(Y, \mathcal{C})$ avoids the configuration Q_5.

Proof. Suppose $(Y, \mathcal{C})$ contains the configuration below.



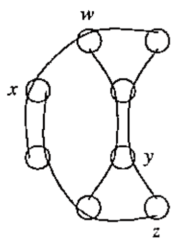
Without loss of generality, we may assume $x = \infty_i$ and $y = \infty_j$ for some $i \neq j$. Then $(X_i, \mathcal{B}_i)$ and $(X_j, \mathcal{B}_j)$ must contain four blocks (two from $\mathcal{B}_i$ and two from $\mathcal{B}_j$) that occur as follows.



The blocks in $\mathcal{B}_i$ are shown in solid lines and the blocks in B_j are shown in dashed lines. But this implies that $c = a + \omega^i b = d + \omega^j b$ and $f = a + \omega^i e = d + \omega^j e$, which can only be satisfied if $b = e$. This is a contradiction. □

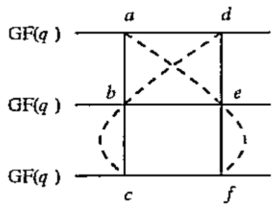
Lemma 8.7.5 The set system $(Y, \mathcal{C})$ avoids the configuration Q_6.

Proof. Suppose $(Y, \mathcal{C})$ contains the configuration below.



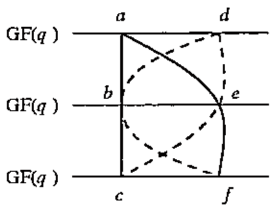
Without loss of generality, either $w = \infty_i$ and $z = \infty_j$, or $x = \infty_i$ and $y = \infty_j$, for some $i \neq j$.

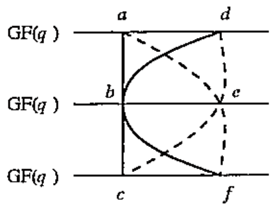
If $w = \infty_i$ and $z = \infty_j$, then $(X_i, \mathcal{B}_i)$ and $(X_j, \mathcal{B}_j)$ must contain four blocks (two from $\mathcal{B}_i$ and two from $\mathcal{B}_j$) that occur as follows.

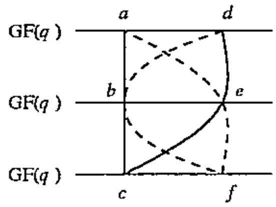


This, as we have seen in the proof of Lemma 8.7.4, is impossible.

If $x = \infty_i$ and $y = \infty_j$, then $(X_i, \mathcal{B}_i)$ and $(X_j, \mathcal{B}_j)$ must contain four blocks (two from $\mathcal{B}_i$ and two from $\mathcal{B}_j$) that occur in one of the following three ways.







The blocks in $\mathcal{B}_i$ are shown in solid lines and the blocks in $\mathcal{B}_j$ are shown in dashed lines. The first situation gives $c = a + \omega^i b = d + \omega^j e$ and $f = a + \omega^i e = d + \omega^j b$, which can only be satisfied if $b = e$ or $\omega^i = -\omega^j$. But $-\omega^j = \omega^{j+(q-1)/2}$ since q is odd, and $i \not\equiv j \pmod{(q-1)/2}$ since $1 \leq i, j \leq (q-1)/2$. The second situation gives $c = a + \omega^i b = d + \omega^j e$ and $f = d + \omega^i b = a + \omega^j e$, which can only be satisfied if $d = a$. For the last situation, we have $c = a + \omega^i b = d + \omega^i e$ and $f = a + \omega^j e = d + \omega^j b$, which can only be satisfied if $b = e$ or $\omega^i = -\omega^j$. As before, $\omega^i = -\omega^j$ is impossible. All these lead to contradictions.

It follows that $(Y, \mathcal{C})$ cannot contain the configuration Q_6. □

We can now state the main result of this section.

Theorem 8.7.1 Let q be an odd prime power, and let λ be an integer such that $1 \leq \lambda \leq (q-1)/2$. Then there exists an $[n, c, 4, 5]$-erasure-resilient code, where $c = 3q - 1 + \lambda$ and $n = \lambda q(q-1)$.

Proof. The set system $(\bigcup_{i=1}^{\lambda} Y_i, \bigcup_{i=1}^{\lambda} \mathcal{C}_i)$ is a 4-uniform 4-partite set system of order $3q - 1 + \lambda$ having $\lambda q(q-1)$ blocks, which avoids the configurations Q_2, Q_4, Q_5, and Q_6 by Lemmata 8.7.2, 8.7.3, 8.7.4, and 8.7.5. Hence, it is the set system of a $(4,5)$-erasure-resilient code. □

The asymptotic behaviour of $F(c, 4)$ and $F(c, 4, 5)$ can now be determined.

Corollary 8.7.1 $F(c, 4) = \Theta(c^3)$ and $F(c, 4, 5) = \Theta(c^3)$.

Proof. Let q be the largest odd prime power, at most $(2c+3)/7$. Taking $\lambda = (q-1)/2$ in Theorem 8.7.1 gives a $[q(q-1)^2/2, (7q-3)/2, 4, 5]$-erasure-resilient code. Hence,

$$\begin{aligned} F(c,4,5) &\geq F((7q-3)/2, 4, 5) \\ &\geq \frac{q(q-1)^2}{2} \\ &\geq \frac{4}{343}c^3 - O(c^{4893/1921+\epsilon}) \qquad \text{(by Theorem 2.5.1)} \end{aligned}$$

for any $\epsilon > 0$. This, together with the inequalities

$$F(c,4,5) \leq F(c,4) \leq D(c,4,3) \leq \frac{1}{24}c^3,$$

gives the required result. □

The bound on $F(c, 4, 5)$ in Corollary 8.7.1 is a significant improvement over the results of [79]. It is an order of magnitude better than even the bound on $F(c, 4)$ obtained in [79].

One drawback of the (4, 5)-erasure-resilient codes produced in Theorem 8.7.1 is that the group size is large and nonuniform. Among the $3q - 1 + \lambda$ points, $2q$ have replication number $\lambda(q-1)$, $q-1$ have replication number λq, and the remaining λ have replication number $q(q-1)$. When $\lambda = (q-1)/2$, all groups have size $\Theta(q^2)$ but the largest group remains about twice as big as the smallest. However, the following *splitting* process can be used to make the group sizes more uniform.

Definition 8.7.3 Suppose $(X, \mathcal{A})$ is a set system and $x \in X$. Let $\mathcal{A}_x = \{A \in \mathcal{A} \mid x \in A\}$ and $\mathcal{A}_x = \mathcal{B}_1 \dot\cup \mathcal{B}_2$ be any partition of $\mathcal{A}_x$ such that $||\mathcal{B}_1| - |\mathcal{B}_2|| \leq 1$. Define $W = X \cup \{x'\}$ and $\mathcal{D} = (\mathcal{A} \setminus \mathcal{B}_1) \cup \{(A \setminus \{x\}) \cup \{x'\} \mid A \in \mathcal{B}_2\}$. Then $(W, \mathcal{D})$ is the set system obtained by *splitting x in* $(X, \mathcal{A})$, and is denoted $\text{split}_x(X, \mathcal{A})$.

We can extend this definition to splitting a subset $S \subseteq X$ in $(X, \mathcal{A})$ as follows.

$$\text{split}_S(X, \mathcal{A}) = \begin{cases} \text{split}_x(X, \mathcal{A}), & \text{if } S = \{x\}; \\ \text{split}_{S\setminus\{x\}}(\text{split}_x(X, \mathcal{A})), & \text{if } x \in S \text{ and } |S| \geq 2. \end{cases}$$

Next, we show that splitting preserves erasure-resilience.

Lemma 8.7.6 If $(X, \mathcal{A})$ is the set system of a (k, l)-erasure-resilient code and $x \in X$, then $\text{split}_x(X, \mathcal{A})$ is also the set system of a (k, l)-erasure-resilient code.

Proof. Suppose not. Then by Lemma 8.3.2, there exist t blocks $A_1, A_2, \ldots, A_t$ in $\text{split}_x(X, \mathcal{A})$, where $2 \leq t \leq l$, such that $|A_1 \Delta A_2 \Delta \cdots \Delta A_t| \leq l - t$. For each of the blocks $A_1, A_2, \ldots, A_t$ that contains x', replace x' by x. Note that this will not increase the size of their symmetric difference. But now, all these blocks are in $\mathcal{A}$, contradicting the assumption that $(X, \mathcal{A})$ is the set system of a (k, l)-erasure-resilient code. $\square$

(a) Q_{16} (b) Q_{17}

Figure 8.6: Forbidden configuration for (4, 6)-erasure-resilient codes.

Let $\infty = \{\infty_1, \infty_2, \ldots, \infty_{(q-1)/2}\}$. It is not hard to see that $\text{split}_\infty(Y, \mathcal{C})$ is a set system of order $4q-2$ with $q(q-1)^2/2$ blocks and all replication numbers are $q^2/2$ or $q(q-1)/2$. It also follows from Lemma 8.7.6 that this is the set system of a (4, 5)-erasure-resilient code. We record this result below.

Lemma 8.7.7 Let q be an odd prime power. Then there exists a $[q(q-1)^2/2, 4q-2, 4, 5]$-erasure-resilient code where the group sizes are $q^2/2$ and $q(q-1)/2$.

8.7.2 The Cases $l = 6$ and $l = 7$

Let $(X, \mathcal{A})$ be the set system of a (4, 6)-erasure-resilient code. Lemma 8.3.2 implies that $(X, \mathcal{A})$ must avoid the configuration Q_{16} shown in Figure 8.6(a). Hence, $(X, \mathcal{A})$ must be a 2-$(c, 4, 1)$ packing and $F(c, 4, 6) \leq D(c, 4, 2)$. This obviates the need to consider many of the configurations treated for the case when $l = 5$. The only configurations that a 2-$(c, 4, 1)$ packing must avoid in order for it to be the set system of a (4, 6)-erasure-resilient code are the configuration Q_{15} shown in Figure 8.5(i) and the configuration Q_{17} shown in Figure 8.6(b).

Consider the standard construction of a transversal design TD$(4, q)$, where q is a prime power. Let

$$X = \mathrm{GF}(q) \times \{0,1,2,3\}, \tag{8.1}$$

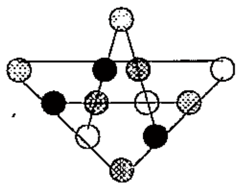
$$\mathcal{G} = \{\mathrm{GF}(q) \times \{i\} \mid i \in \{0,1,2,3\}\}, \quad \text{and}$$

$$\mathcal{B} = \{\{(a,0),(b,1),(a+b,2),(a+2b,3)\} \mid a,b \in \mathrm{GF}(q)\}. \tag{8.2}$$

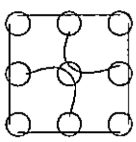
Then $(X, \mathcal{G}, \mathcal{B})$ is a TD$(4, q)$. It is easy to see that the set system $(X, \mathcal{B})$ is a 4-partite 2-$(4q, 4, 1)$ packing. Let $(X', \mathcal{G}', \mathcal{B}')$ be the TD$(3, q)$ obtained by truncating the entire group $\mathrm{GF}(q) \times \{3\}$.

Lemma 8.7.8 The set system $(X, \mathcal{B})$ avoids the configuration Q_{17}.

Proof. Suppose $(X, \mathcal{B})$ contains the configuration below.



This configuration has a unique (up to isomorphism) partition of its points into four parts so that each block contains exactly one point from each part. This partition is indicated by the different shadings in the figure above. Hence, the points of one of the parts must belong to $\mathrm{GF}(q) \times \{3\}$. It is easy to check that deleting all the points in any part gives the following configuration.



So $(X', \mathcal{B}')$ must contain the configuration above. There are six possibilities to consider, as shown below.

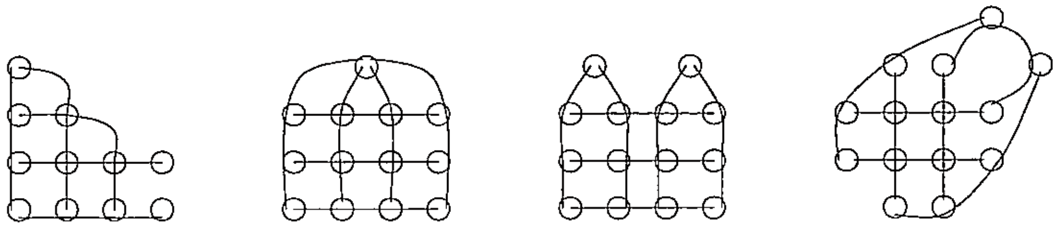
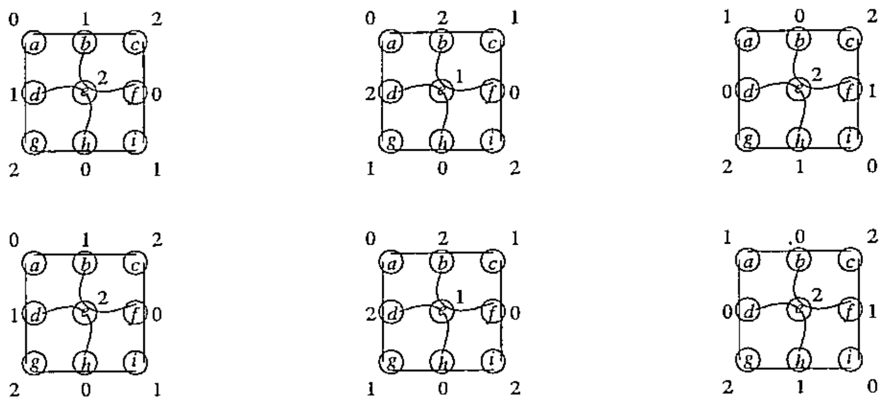


Figure 8.7: Forbidden configurations for (4, 7)-erasure-resilient codes.



Each point is an element of $\mathrm{GF}(q) \times \{0, 1, 2\}$. The label inside a point shows its first coordinate and the label outside a point shows its second coordinate.

Consider the first possibility. We have $c = a + b = f + i$, $e = b + f = d + h$, and $g = a + d = h + i$, which can only be satisfied if $b = d$. This is a contradiction.

The other five possibilities can be disposed of similarly. □

The set system of a (4, 7)-erasure-resilient code must avoid the four configurations in Figure 8.7 in addition to all the forbidden configurations for set systems of (4, 6)-erasure-resilient codes.

Theorem 8.7.2 Let q be a prime power. Then there exists an $[q^2, 4q, 4, 7]$-erasure-resilient code. Moreover, this code has uniform group size q.

Proof. It follows from Lemma 8.7.8 that the set system $(X, \mathcal{B})$ is the set system of a $(4,6)$-erasure-resilient code. It is easily verified that all the configurations in Figure 8.7 are not 4-partite. Since $(X, \mathcal{B})$ is 4-partite, these configurations are all avoided. Hence, $(X, \mathcal{B})$ is also the set system of a $(4,7)$-erasure-resilient code. It is straightforward to see that every point of X is contained in exactly q blocks of $\mathcal{B}$. □

Corollary 8.7.2 $F(c,4,6) = \Theta(c^2)$ and $F(c,4,7) = \Theta(c^2)$.

Proof. Let q be the largest prime power, at most $c/4$. Theorem 8.7.2 gives a $[q^2, 4q, 4, 7]$-erasure-resilient code. Hence,

$$\begin{aligned} F(c,4,7) &\geq F(4q,4,7) \\ &\geq q^2 \\ &\geq \frac{1}{16}c^2 - O(c^{2972/1921+\epsilon}) \qquad \text{(by Theorem 2.5.1)} \end{aligned}$$

for any $\epsilon > 0$. This, together with the inequalities

$$F(c,4,7) \leq F(c,4,6) \leq D(c,4,2) \leq \frac{1}{12}c^2,$$

gives the required result. □

8.8 Controlling Group Sizes by Balanced Orderings

Let $H = [C \mid I]$ be the parity-check matrix of an $[n,c,k,l]$-erasure-resilient code. Let $g_1, g_2, \ldots, g_c$ be the group sizes of this code. By counting the ones in H in two different ways, we obtain

$$\sum_{i=1}^{c} g_i = kn + c.$$

So the average group size is $kn/c+1$. Since the check disk overhead is c/n, the smaller the check disk overhead, the larger the average group size. In the previous sections, our focus has been on the construction of erasure-resilient codes with optimal (up to constant factors) check disk overheads. Therefore, inevitably, our codes have large average group size.

It is however, possible to trade check disk overhead for a smaller average group size. Given the parity-check matrix $[C \mid I]$ of an erasure-resilient code, one can simply delete the appropriate number of columns of C so that the desired average group size is obtained. However, this process does not guarantee that the maximum group size will be lowered. We have indicated in Section 8.2 that for load balancing reasons, uniform group size is desirable. This raises the issue of whether it is possible to construct erasure-resilient codes for which there is a way of deleting columns from its parity-check matrix so that every group size is close to the average. Let us discuss this problem more formally. The terminology we use here generalizes those used in [79].

Definition 8.8.1 Let α be a positive integer. An erasure-resilient code is said to have *α-balanced group size* if the following conditions hold:

(i) when the average group size is 1 (mod α), all groups are the same size;

(ii) when the average group size is not 1 (mod α), the maximum group size is at most α greater than the minimum group size.

Let M be an $m \times n$ matrix. For any $1 \leq i \leq n$, $M(i)$ denotes the $m \times i$ matrix comprising the first i columns of M.

Definition 8.8.2 Let $H = [C \mid I]$ be the parity-check matrix of an $[n, c, k, l]$-erasure-resilient code and α a positive integer. We say that the columns of C are arranged in an

α-balanced ordering if, for any $1 \leq i \leq n$, $H = [C(i) \mid I]$ is the parity-check matrix of an $[i, c, k, l]$-erasure-resilient code with α-balanced group size.

The existence of an α-balanced ordering for a (k, l)-erasure-resilient code allows us to derive from it other (k, l)-erasure-resilient codes with higher check disk overhead but smaller group sizes, and whose group sizes differ from one another by at most α. Another use of balanced orderings observed by Hellerstein et al. [79] is in the design of extensible disk array systems. If we have chosen a code whose parity-check matrix have more columns then we need, then as more disks are added to the system, the extra columns are put to use. The existence of an α-balanced ordering for the original parity-check matrix ensures that we have α-balanced group size at all times if disks are associated with columns according to this ordering. The case $\alpha = 1$ was considered by Hellerstein et al. [79].

Definition 8.8.3 Let α be a positive integer and $(X, \mathcal{A})$ a set system. Let $\mathcal{B} \subseteq \mathcal{A}$ be a subset of blocks. Then,

(i) $\mathcal{B}$ is an *α-resolution class* if every element of X is contained in precisely α blocks of $\mathcal{B}$;

(ii) $\mathcal{B}$ is a *partial α-resolution class* if every element of X is contained in at most α blocks of $\mathcal{B}$.

Definition 8.8.4 Let α be a positive integer. A set system $(X, \mathcal{A})$ is *α-resolvable* if $\mathcal{A}$ can be partitioned into parts $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_r$, each of which is an α-resolution class.

Definition 8.8.5 Let α be a positive integer. A set system $(X, \mathcal{A})$ is *almost α-resolvable* if $\mathcal{A}$ can be partitioned into parts $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_r$, each of which is an α-resolution class, except perhaps for one part which is a partial α-resolution class.

If $(X, \mathcal{A})$ is k-uniform, then $(X, \mathcal{A})$ is α-resolvable or almost α-resolvable only if $\alpha|X| \equiv 0 \pmod{k}$, since the number of blocks in each α-resolution class is exactly $\alpha|X|/k$.

Lemma 8.8.1 Let $H = [C \mid I]$ and $(X, \mathcal{A})$ be the parity-check matrix and set system of an $[n, c, k, l]$-erasure-resilient code, respectively. Then C has an α-balanced ordering if and only if $(X, \mathcal{A})$ is almost α-resolvable.

Proof. Suppose $(X, \mathcal{A})$ is almost α-resolvable with α-resolution classes $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{r-1}$ and a partial α-resolution class $\mathcal{A}_r$ (which can be empty). Order the matrix C so that $C = [C_1 \mid C_2 \mid \cdots \mid C_r]$, where each C_i contains precisely those columns whose supports are in $\mathcal{A}_i$. The ordering of the columns within each C_i can be arbitrary. This is an α-balanced ordering for C.

Now suppose C has an α-balanced ordering. Consider the first $\alpha c/k$ columns of C and the set of their supports $\mathcal{A}_1$. The erasure-resilient code formed by these columns has average group size $\alpha + 1$, and hence each group has size $\alpha + 1$. If follows that every point is contained in exactly α blocks in $\mathcal{A}_i$. Now consider the first $i(\alpha c/k)$ columns of C, $2 \leq i \leq \lfloor nk/\alpha c \rfloor$, and the set of their supports $\mathcal{B} \,\dot{\cup}\, \mathcal{A}_i$, where $\mathcal{A}_i$ is the set of supports of columns $(i-1)(\alpha c/k) + 1$ to $i(\alpha c/k)$ of C. The average group size of the erasure-resilient code formed by the first $i(\alpha c/k)$ columns of C is $i\alpha + 1$. Hence every point appears in exactly $i\alpha$ blocks of $\mathcal{B} \,\dot{\cup}\, \mathcal{A}_i$. By the induction hypothesis, every point appears in exactly $(i-1)\alpha$ blocks of $\mathcal{B}$. It follows that every point must appear in precisely α blocks of $\mathcal{A}_i$. Consequently, $\mathcal{A}_i$ is an α-resolution class. The supports of the remaining columns of C constitute a partial α-resolution class. □

Hellerstein et al. [79] showed that the $[3^a(3^a - 1)/6, 3^a, 3, 4]$-erasure-resilient code they constructed (see Lemma 8.6.2) has a 1-balanced ordering. In fact, the set system of this erasure-resilient code is the *affine geometry* $\mathrm{AG}_1(a, 3)$ (see, for example, [15]) whose 1-resolvability is a classical result in design theory. An STS(n) that is 1-resolvable is commonly known as a *Kirkman triple system of order n*, or KTS(n). The above discussion shows that the problem of constructing $[n, c, 3, 4]$-erasure-resilient codes with optimal check disk overhead having a 1-balanced ordering is equivalent to the following problem.

Problem 8.8.1 Determine the existence of anti-Pasch Kirkman triple systems.

The existence of KTS(n) has long been settled [93, 117]; the condition $n \equiv 3 \pmod 6$ is both necessary and sufficient. Work on the existence problem for anti-Pasch STS(n) is also well under way. However, Problem 8.8.1 appears not to have been studied, perhaps due to the lack in motivation. As we have shown, this is not the case now. Here, we settle the existence problem for anti-Pasch almost 3-resolvable Steiner triple systems.

Example 8.8.1 Let $X = \mathbf{Z}_{21}$ and

$$\mathcal{A}_1 = \{\{i, 1+i, 3+i\} \mid i \in \mathbf{Z}_{21}\},$$
$$\mathcal{A}_2 = \{\{i, 4+i, 12+i\} \mid i \in \mathbf{Z}_{21}\},$$
$$\mathcal{A}_3 = \{\{i, 6+i, 11+i\} \mid i \in \mathbf{Z}_{21}\},$$
$$\mathcal{A}_4 = \{\{i, 7+i, 14+i\} \mid 0 \leq i \leq 6\}.$$

Let $\mathcal{A} = \bigcup_{1 \leq i \leq 4} \mathcal{A}_i$. Then $(X, \mathcal{A})$ is an anti-Pasch almost 3-resolvable STS(21). The 3-resolution classes are $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$. The partial 3-resolution class is $\mathcal{A}_4$.

Lemma 8.8.2 There exists an anti-Pasch almost 3-resolvable STS($3g$) whenever g is odd and $g \not\equiv 0 \pmod 7$.

Proof. Let $X = \mathbf{Z}_g \times \{0, 1, 2\}$. Let $\mathcal{A}$ contain the following blocks:

(i) $\{(a,0),(a,1),(a,2)\}$, for all $a \in \mathbf{Z}_g$;

(ii) $\{(a,i),(b,i),((a+b)2^{-1},i+1)\}$, for all $a,b \in \mathbf{Z}_g$, $a \neq b$, and all $i \in \mathbf{Z}_3$ (reducing subscripts modulo 3 as necessary).

Brouwer [21] has shown that $(X,\mathcal{A})$ is an anti-Pasch STS($3g$) when $g \not\equiv 0 \pmod 7$. We show that $(X,\mathcal{A})$ is almost 3-resolvable. The partial 3-resolution class is taken to be $\{\{(a,0),(a,1),(a,2)\} \mid a \in \mathbf{Z}_g\}$ (which is in fact a 1-resolution class). The other $(g-1)/2$ 3-resolution classes are

$$\{\{(a,i),(a+j,i),((2a+j)2^{-1},i+1)\} \mid a \in \mathbf{Z}_g, i \in \mathbf{Z}_3\}, \quad \text{for } 1 \leq j \leq (g-1)/2.$$

□

Theorem 8.8.1 There exists an anti-Pasch almost 3-resolvable STS(n) for all $n \equiv 3 \pmod 6$.

Proof. Lemma 8.8.2 handles all cases except when $n \equiv 0 \pmod 7$.

So suppose that $n \equiv 3 \pmod 6$ and $n = 7v$. Then proceeding inductively, there is an anti-Pasch almost 3-resolvable STS(v), $(X,\mathcal{A})$, with 3-resolution classes $\mathcal{A}_1$, $\mathcal{A}_2$, ..., $\mathcal{A}_{(v-3)/6}$ and a partial 3-resolution class $\mathcal{A}^*$. Let $Y = X \times \mathbf{Z}_7$. For each $A \in \mathcal{A}^*$, construct an STS(21), $(A \times \mathbf{Z}_7, \mathcal{B}(A))$, which is isomorphic to that given in Example 8.8.1. Let $\mathcal{B}(A)_1$, $\mathcal{B}(A)_2$ and $\mathcal{B}(A)_3$ be the 3-resolution classes of this STS(21), and $\mathcal{B}(A)^*$ the partial 3-resolution class. Define

$$\mathcal{B}_i = \bigcup_{A \in \mathcal{A}^*} \mathcal{B}(A)_i, \qquad \text{for } 1 \leq i \leq 3, \quad \text{and}$$

$$\mathcal{B}^* = \bigcup_{A \in \mathcal{A}^*} \mathcal{B}(A)^*.$$

Next, define

$$\mathcal{B}_{i,j} = \bigcup_{A \in \mathcal{A}_i} \{\{(a,h),(b,h+j),(c,2h+j)\} \mid A = \{a,b,c\} \text{ and } h \in \mathbf{Z}_7\},$$

$$\text{for } 1 \le i \le (v-3)/6 \text{ and } j \in \mathbf{Z}_7.$$

Finally, let

$$\mathcal{B} = \left(\bigcup_{1 \le i \le 3} \mathcal{B}_i \right) \cup \left(\bigcup_{\substack{1 \le i \le (v-3)/6 \\ j \in \mathbf{Z}_7}} \mathcal{B}_{i,j} \right) \cup \mathcal{B}^*.$$

Brouwer [21] and Griggs, Murphy, and Phelan [72] have established that $(Y, \mathcal{B})$ is an anti-Pasch STS($7v$). It is easy to check that $\mathcal{B}_i$, $1 \le i \le 3$, and $\mathcal{B}_{i,j}$, $1 \le i \le (v-3)/6$, $j \in \mathbf{Z}_7$, are 3-resolution classes of $(Y, \mathcal{B})$, and $\mathcal{B}^*$ is a partial 3-resolution class. □

The $(4,7)$-erasure-resilient code we construct in Theorem 8.7.2 has a 1-balanced ordering. This follows from the well-known result in design theory concerning the resolvability of transversal designs produced by the standard finite field construction (see [15]). We give the proof here for the sake of completeness.

Theorem 8.8.2 The $(4,7)$-erasure-resilient code of Theorem 8.7.2 has a 1-balanced ordering.

Proof. The set system $(X, \mathcal{B})$ of the code is given by (8.1) and (8.2). This set system is 1-resolvable, with 1-resolution classes $\{\{(a,0),(b,1),(a+b,2),(a+2b,3)\} \mid a+3b=\zeta\}$, for $\zeta \in \mathrm{GF}(q)$. □

8.9 Complexity of Code Construction

One of the most important issues associated with any family of codes is the question of how hard it is to encode and decode in the family [12, 125, 126].

The erasure-resilient codes we consider are systematic binary linear codes. All systematic binary linear codes have an extremely simple encoding procedure. Suppose $H = [C \mid I]$ is the parity-check matrix of such a code. Then the encoding of a (row) vector $\mathbf{x} \in \{0,1\}^n$ is the vector $(\mathbf{x} \mid \mathbf{x}C^T)$. The decoding of erasure-resilient codes is discussed at length by Hellerstein et al. [79]. It is straightforward to see that both encoding and decoding of erasure-resilient codes can be carried out efficiently.

It has been pointed out by Bassalygo, Zyablov, and Pinsker [12] that in addition to considering the complexity of encoding and decoding procedures, we should also examine the complexity of building the encoding and decoding software and hardware. It is clear that this reduces to the complexity of constructing the parity-check matrices or their associated set systems. For the remainder of this section, we study the complexity of constructing the erasure-resilient codes described in this thesis. The model of computation we adopt is the *unit-cost random access machine* (RAM) (see [2]). The more realistic bit-cost RAM model can also be used, but this introduces only a polylogarithmic factor in our results.

8.9.1 Generating Anti-Pasch Steiner Triple Systems

Anti-Pasch Steiner triple systems are set systems associated with optimal $(3,l)$-erasure-resilient codes, for $l = 4$ and 5. In this section, we consider the construction of anti-Pasch STS(v), where $v \equiv 3 \pmod 6$. Our aim is to design an efficient algorithm which on input $v \equiv 3 \pmod 6$, outputs the blocks of an anti-Pasch STS(v). We have to be careful here with the meaning of the word "efficient". The size of the input is $O(\log v)$ and the output

```
anti-Pasch-STS(v)
    if v ≢ 0 (mod 7) then
        return the blocks obtained by applying Lemma 8.8.2;
    else if v = 21 then
        return the blocks in Example 8.8.1;
    else
        (X, B) = anti-Pasch-STS(v/7);
        with (X, B), return the blocks given in the proof of Theorem 8.8.1;
```

Figure 8.8: Algorithm for generating anti-Pasch STS(v), $v \equiv 3 \pmod 6$.

consists of $v(v-1)/6$ blocks, which has size exponential in the size of the input. Hence, we say that an algorithm is "efficient" if its running time is polynomial in the size of its output. Any algorithm for constructing anti-Pasch STS(v) must output $v(v-1)/2$ numbers in $\mathbf{Z}_v$, since each block contains precisely three elements, and there are exactly $v(v-1)/6$ blocks. It follows that any algorithm must take time $\Omega(v^2)$. We describe in Figure 8.8 an algorithm which achieves $O(v^2)$ time.

Theorem 8.9.1 Algorithm anti-Pasch-STS given in Figure 8.8 outputs the blocks of an anti-Pasch STS(v) in $O(v^2)$ time.

Proof. Correctness of the algorithm follows from Lemma 8.8.2 and Theorem 8.8.1. Let $T(v)$ denote the running time of the algorithm on input v. If $v \not\equiv 0 \pmod 7$, we can efficiently determine 2^{-1} in $\mathbf{Z}_{v/3}$ using the extended Euclidean algorithm. An additional $O(v^2)$ additions and multiplications in $\mathbf{Z}_{v/3}$ suffice to construct all the required blocks. Hence $T(v) = O(v^2)$ when $v \not\equiv 0 \pmod 7$. If $v \equiv 0 \pmod 7$, we have the recurrence

$$T(v) = T\left(\frac{v}{7}\right) + O(v^2),$$

which also gives $T(v) = O(v^2)$ by an easy induction. □

It is interesting to point out that it is only recently that Colbourn [38] began a study of complexity issues related to the construction of combinatorial designs.

8.9.2 Generating $(4, l)$-Erasure-Resilient Codes

Our goal here is to design an algorithm that on input q, outputs the blocks of a set system associated with the $(4, l)$-erasure-resilient code given by Theorem 8.7.1 and Theorem 8.7.2. It is easy to see that these set systems can be constructed using a polynomial number (in q) of arithmetic operations in $\mathrm{GF}(q)$. Therefore, the main problem here is the synthesis of the finite field $\mathrm{GF}(q)$. Let $q = p^\alpha$, where p is prime and $\alpha \in \mathbf{N}$. Shoup [131] gave an algorithm for synthesizing finite fields with a running time of $O(\sqrt{p}(\alpha + \log p)^{O(1)})$. This time bound is not polynomial in the size of q but is polynomial in the size of the output (which is at least $\Omega(q^2)$). It follows that all our erasure-resilient codes can be constructed efficiently.

8.10 From Erasure-Resilient Codes to Group Testing

We consider the k-RESTRICTED NONADAPTIVE EXACT IDENTIFICATION PARITY PROBLEM(r). Specializing Lemma 4.2.2 to MOD_2 test functions gives the following.

Lemma 8.10.1 Let (X, r, f, Π) be a group testing problem with the MOD_2 test function and the exact identification criterion. A set system $(Y, \mathcal{B})$ is the dual system of a nonadaptive algorithm for (X, r, f, Π) if and only if the following condition holds. For any blocks $A_1, A_2, \ldots, A_a \in \mathcal{B}$ and $B_1, B_2, \ldots, B_b \in \mathcal{B}$, where $a \leq r$ and $b \leq r$, we have

$$A_1 \Delta A_2 \Delta \cdots \Delta A_a \neq B_1 \Delta B_2 \Delta \cdots \Delta B_b,$$

unless $\{A_1, A_2, \ldots, A_a\} = \{B_1, B_2, \ldots, B_b\}$.

We call a set system satisfying the condition of Lemma 8.10.1 an *r-difference-free set system.* Let $d(n, k, r)$ denote the maximum number of blocks in an r-difference-free k-uniform set system of order n. No previous results on nonadaptive algorithms for the k-RESTRICTED NONADAPTIVE EXACT IDENTIFICATION PARITY PROBLEM(r) or r-difference-free k-uniform set systems are known, although some results have been obtained by Chang, Hwang, and Weng [31] for the sequential case. In this section, we show how erasure-resilient codes can be used to construct r-difference-free k-uniform set systems, and hence nonadaptive algorithms for the k-RESTRICTED NONADAPTIVE EXACT IDENTIFICATION PARITY PROBLEM(r).

Theorem 8.10.1 If there exists an $[n, c, k, l]$-erasure-resilient code, then there exists an $\lfloor l/2 \rfloor$-difference-free k-uniform set system of order c having n blocks.

Proof. Let $(X, \mathcal{A})$ be the set system of an $[n, c, k, l]$-erasure-resilient code. We claim that $(X, \mathcal{A})$ is $\lfloor l/2 \rfloor$-difference-free. Suppose not. Then there exist blocks $A_1, A_2, \ldots, A_a \in \mathcal{A}$ and $B_1, B_2, \ldots, B_b \in \mathcal{A}$, where $a \leq \lfloor l/2 \rfloor$, $b \leq \lfloor l/2 \rfloor$, such that

$$A_1 \Delta A_2 \Delta \cdots \Delta A_a = B_1 \Delta B_2 \Delta \cdots \Delta B_b.$$

This gives

$$|A_1 \Delta A_2 \Delta \cdots \Delta A_a \Delta B_1 \Delta B_2 \Delta \cdots \Delta B_b| = 0.$$

Since $a + b \leq l$, this contradicts the fact that $(X, \mathcal{A})$ is the set system of a (k, l)-erasure-resilient code. □

Corollary 8.10.1 Let k and r be positive integers such that $k \geq r$. Then $d(n,k,r) \geq \Omega(n^{(2(k-r)+1)/4})$.

Proof. Follows from the codes obtained from expanders in Corollary 8.5.1. □

We can also determine the order of $d(n,3,2)$ exactly.

Theorem 8.10.2 $d(n,3,2) = \Theta(n^2)$.

Proof. There exist $[c^2/6 - O(c), c, 3, 4]$-erasure-resilient codes (Lemma 8.6.4). Hence, by Theorem 8.10.1, $d(n,3,2) \geq n^2/6 - O(n)$.

The upper bound can be proven using the same argument in [62] for weakly union-free 3-uniform set systems. We give the proof here for completeness. Let $(X, \mathcal{A})$ be any 2-difference-free 3-uniform set system. Let us define, for $B \in \binom{X}{2}$, $T(B) = \{x \in X \mid B \cup \{x\} \in \mathcal{A}\}$. For every i, $0 \leq i \leq n-2$, let

$$G_i = \left\{ B \in \binom{X}{2} \,\middle|\, |T(B)| = i \right\}.$$

Let $g_i = |G_i|$. Clearly, $\{G_0, G_1, \ldots, G_{n-2}\}$ is a partition of $\binom{X}{2}$. Thus, we have

$$\sum_{i=0}^{n-2} g_i = \binom{n}{2}. \tag{8.3}$$

Counting the number of pairs (B, A) such that $B \in \binom{X}{2}$, $A \in \mathcal{A}$, and $B \subset A$, in two ways, we obtain

$$\sum_{i=0}^{n-2} i g_i = 3|\mathcal{A}|. \tag{8.4}$$

We claim that

$$\binom{T(B)}{2} \bigcap \binom{T(B')}{2} = \varnothing.$$

Suppose not. Let $\{x, x'\}$, $x \neq x'$, belong to the intersection. Then $A_1 = B \cup \{x\}$, $A_2 = B \cup \{x'\}$, $A_3 = B' \cup \{x\}$, $A_4 = B' \cup \{x'\}$ are four blocks of $\mathcal{A}$. But $A_1 \Delta A_2 = A_3 \Delta A_4$, a contradiction. Hence, we have

$$\sum_{i=2}^{n-2} \binom{i}{2} g_i \leq \binom{n}{2}. \tag{8.5}$$

Adding (8.3) to (8.5) gives

$$\sum_{i=0}^{n-2} \left(1 + \binom{i}{2}\right) g_i \leq n(n-1),$$

which implies

$$\sum_{i=0}^{n-2} i g_i + \sum_{i=0}^{n-2} \left(1 + \binom{i}{2} - i\right) g_i \leq n(n-1). \tag{8.6}$$

The first term of (8.6) is just $3|\mathcal{A}|$ by (8.4), while the second is nonnegative. Thus, $|\mathcal{A}| \leq n(n-1)/3$.

This completes the proof. □

8.11 Summary

In this chapter, we have considered the construction of erasure-resilient codes for increasing the reliability of large disk arrays. We adopt a set systems approach different from those considered previously. As a result, we have at our disposal many tools from design theory, which enabled us to construct some classes of erasure-resilient codes better than any known. It is also observed that previous results in erasure-resilient codes follow easily or even trivially from existing results in design theory. This suggests that perhaps the approach considered here is the natural one.

We close this chapter with a conjecture.

Conjecture 8.11.1 For any integer $k \geq 2$, we have $F(c,k,l) = \Theta(n^{k+1-\lfloor l/2 \rfloor})$, for all l such that $k \leq l \leq 2k-1$.

The 2d-parity code constructed in [79] shows that Conjecture 8.11.1 holds when $k = 2$. Our work in this chapter shows that Conjecture 8.11.1 is true for $k = 3$ and 4.

CHAPTER 9

Frameproof Codes for Digital Fingerprinting

A procedure, called *fingerprinting*, commonly practiced by suppliers is to mark their products with an identifier, called a *fingerprint*, before distributing the products to the users. By fingerprinting, it is hoped that the following two objectives can be met:

(i) products can be distinguished; and

(ii) a product can be traced back to its user.

Condition (ii) discourages users from unauthorized use of the products. So it is the wish of the users to destroy the fingerprints, while the supplier tries to prevent this from happening. Fingerprinting has been applied to a diverse spectrum of objects, including consumer goods, advertisements, explosives, mathematical tables, and maps [150]. For digital products, such as computer software or data, the difficulty in designing fingerprints is immense, since digital data can be processed and manipulated easily. For example, two or more users can compare their digital data and deduce that the fingerprints are where their copies differ. If the fingerprints are not carefully designed, it is also possible for a coalition of users to generate new fingerprints, allowing them to frame other users of unauthorized actions. The goal of the supplier is to produce undetectable and unalterable fingerprints.

The problem discussed above is first studied by Boneh and Shaw [19], who showed that certain codes, known as frameproof codes, can be used to solve the problem. The combinatorics of frameproof codes is further investigated by Stinson and Wei [142], who observe the equivalence to a certain Turán-type problem. The purpose of this chapter is to establish several improved bounds for frameproof codes.

9.1 Technical Preliminaries

Let $\mathbf{D} \in \{0,1\}^d$ be a piece of binary digital data, which is to be made available to m users. A fingerprint, $\mathbf{w}^{(i)} \in \{0,1\}^n$, is generated for user i, $1 \le i \le m$. The fingerprinted data $f(\mathbf{D}, \mathbf{w}^{(i)})$ is then distributed to user i, where f denotes the function performing the fingerprinting process. Owing to space limitations, it is not possible for us to survey the various techniques used for incorporating fingerprints into digital data. We refer the interested reader to [20, 150] for more information. We should point out, however, that the only information we obtain from two fingerprinted data $f(\mathbf{D}, \mathbf{w}^{(i)})$ and $f(\mathbf{D}, \mathbf{w}^{(j)})$ is exactly the information we would have obtained from $\mathbf{w}^{(i)}$ and $\mathbf{w}^{(j)}$. This gives rise to the first of the following three properties that a fingerprinted data should satisfy.

(i) Two users can detect the bit positions in which their respective fingerprints defer, and nothing else.

(ii) A user cannot change an undetected bit without rendering the data useless.

(iii) Any detected bit can be changed, or made unreadable.

As in [19], we assume that methods exist to produce fingerprinted data satisfying the three properties above. With this assumption, it is obvious that if users do not collude, then assigning a unique fingerprint to each user would enable us to detect any unauthorized use. This is also true if users have no knowledge of the set of all fingerprints. In reality, we have to worry about collusions, as well as the possibility that the set of all fingerprints is known to the users.

Definition 9.1.1 Let $\mathbf{w}^{(1)}$, $\mathbf{w}^{(2)}$, ..., $\mathbf{w}^{(m)} \in \{0,1\}^n$ and $C \subseteq \{1, 2, \ldots, m\}$. For $i \in \{1, 2, \ldots, n\}$, bit position i is said to be *undetectable for C* if $|\{\mathbf{w}_i^{(j)} \mid j \in C\}| = 1$. Define $U(C)$ to be the *set of undetectable bit positions for C*.

Intuitively, a bit position is undetectable for a coalition C of users if the fingerprints assigned to users in C all agree in that position.

Definition 9.1.2 Let $\Gamma = \{\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(m)}\} \subseteq \{0,1\}^n$ and C a coalition of users. The *feasible set of C* is

$$F(C,\Gamma) = \left\{ \mathbf{w} \in \{0,1\}^n \;\middle|\; \mathbf{w}|_{U(C)} = \mathbf{w}^{(i)}|_{U(C)} \text{ for all } i \in C \right\}.$$

The feasible set is the set of all possible vectors that match the undetected bits of C. Hence, the coalition C can only create a piece of data whose fingerprint lies in $F(C,\Gamma)$. It follows that a user (outside the coalition C) can be framed by C if and only if the fingerprint of his piece of data is in $F(C,\Gamma) \setminus \{\mathbf{w}^{(i)} \mid i \in C\}$. The desire for this not to happen motivates the following definition.

Definition 9.1.3 Let $\Gamma \subseteq \{0,1\}^n$. We call Γ an *r-frameproof code of length n* if for every $W \subseteq \Gamma$ such that $|W| \leq r$, we have $F(W,\Gamma) \cap \Gamma = W$. The elements of Γ are called codewords.

9.2 Bounds on Frameproof Codes

The problem of designing frameproof codes is to construct for any given r, a family of r-frameproof codes having as high a rate as possible. The observation that this problem is equivalent to a Turán-type problem is made recently by Stinson and Wei [142].

Definition 9.2.1 A set system $(X, \mathcal{A})$ is *r-frameproof* if there do not exist $r+1$ blocks A_1, A_2, ..., $A_{r+1} \in \mathcal{A}$ such that

$$\bigcap_{i=1}^{r} A_i \subseteq A_{r+1} \subseteq \bigcup_{i=1}^{r} A_i,$$

unless $A_{r+1} \in \{A_1, A_2, \ldots, A_r\}$.

Theorem 9.2.1 (Stinson and Wei [142]) There exists an r-frameproof code of length n having m codewords if and only if there exists an r-frameproof set system of order n, having m blocks.

Proof. Let M be the matrix whose columns are codewords of an r-frameproof code. Then M is the point-block incidence matrix of an r-frameproof set system. □

Let $f(n, k, r)$ denote the maximum number of blocks in an r-frameproof k-uniform set system of order n.

Theorem 9.2.2 (Boneh and Shaw [19]) For any $r \in \mathbb{N}$, there exists a family of r-frameproof codes with rate at least $1/16r^2$.

The bound in Theorem 9.2.2 was established using a probabilistic construction. For constant weight frameproof codes (or frameproof uniform set systems), we have the following result of Stinson and Wei [142].

Theorem 9.2.3 (Stinson and Wei [142]) For any $n, k \in \mathbb{N}$, we have $f(n,k,k-1) \geq D(n,k,2)$.

It is trivial to see that any r-cover-free set system is r-frameproof. Therefore, Theorem 9.2.3 is a simple corollary of Lemma 5.1.1. Indeed, many known bounds for r-cover-free set systems already supersede Theorem 9.2.2. It appears that neither Boneh and Shaw nor Stinson and Wei are aware of the work on r-cover-free set systems carried out by the data communications community [27, 50, 51, 52, 83, 107]. The r-cover-free set systems often appear under the guise of *superimposed codes*, which are introduced by Kautz and Singleton [83].

Definition 9.2.2 A set $\mathcal{S} \subseteq \{0,1\}^n$ is an *r-superimposed code of length n* if there are no $r+1$ codewords $S_1, S_2, \ldots, S_{r+1} \in \mathcal{S}$ with the property that $S_{r+1} \preceq S_1 \vee S_2 \vee \cdots \vee S_r$, unless $S_{r+1} \in \{S_1, S_2, \ldots, S_r\}$.

It is easy to see that a set of $\{0,1\}$-vectors is an r-superimposed code if and only if the supports of these vectors form an r-cover-free set system. It is known long ago (see [88, 123]) that there exists a family of r-superimposed codes of rate c/r^2, for some absolute constant c. Busschbach [27] (see also [48, 134]) gave a family of r-superimposed codes of rate $(1-o(1))/3(r+1)^2$. This was improved by Erdös, Frankl, and Füredi [58], and Hwang and Sós [81] to a rate of $\log(1+1/4r^2)$. Nguyen and Zeisel [107] obtained an even better rate of $(0.6617-o(1))/(r+1)^2$. The best lower bound currently known is the following result of Dyachkov, Rykov, and Rashad [52].

Theorem 9.2.4 (Dyachkov, Rykov, and Rashad [52]) For any $r \in \mathbb{N}$, there exists a family of r-superimposed code with rate $(1-o(1))A_r/r$, where

$$A_r = \max_{0 \leq Q \leq 1} \max_{0 \leq q \leq 1} \left[-(1-Q)\log(1-q^r) + r\left(Q \log \frac{q}{Q} + (1-Q)\log\frac{1-q}{1-Q}\right)\right].$$

Theorem 9.2.4 implies the existence of a family of r-frameproof codes of length n with a rate of $(1 - o(1))A_r/r$. It is known that $\lim_{r\to\infty} A_r = \frac{\ln 2}{r}$ [52]. So the improvement on Theorem 9.2.2 is quite drastic.

However, the less stringent defining conditions of an r-frameproof set system permit us to establish results better than those implied by r-cover-free set systems. In the next section, we give an improved bound on 2-frameproof codes using a probabilistic construction.

9.3 A Probabilistic Construction for 2-Frameproof Codes

Definition 9.3.1 An *r-frameproof array of order n and size m* is an $n \times m$ matrix with entries from $\{0, 1\}$ such that every $n \times (r + 1)$ submatrix L of M has the property that for every $i \in \{1, 2, \ldots, r + 1\}$, either $\mathbf{e}_i$ or $\mathbf{1} - \mathbf{e}_i$ appears as a row of L.

We begin with the following property of frameproof codes.

Lemma 9.3.1 The existence of an r-frameproof set system of order n with m blocks is equivalent to the existence of an r-frameproof array of order n and size m.

Proof. Let M be the point-block incidence matrix of an r-frameproof set system. It is straightforward to verify that M is an r-frameproof array. □

For any fixed $\epsilon > 0$, let M be a $2n \times (1 + \epsilon)m$ matrix[2] with entries from $\{0, 1\}$ constructed as follows. Each column of M is a vector selected uniformly at random from the set of all vectors in $\{0, 1\}^{2n}$ of weight n. Let $\mathcal{C} = \{1, 2, \ldots, (1 + \epsilon)m\}$ be the set of column indices of M. For $C \in \binom{\mathcal{C}}{3}$, define $M(C)$ to be the $2n \times 3$ submatrix of M with

[2] Strictly speaking, we should write $\lceil (1 + \epsilon)m \rceil$ instead of $(1 + \epsilon)m$. However, this only introduces notational burden, and does not affect the results in any way. If the reader is uncomfortable, he/she can replace all occurrence of ϵm with $\lceil \epsilon m \rceil$.

columns in C. Further define the indicator variables

$$X(C,i) = \begin{cases} 0, & \text{if } M(C) \text{ contains } \mathbf{e}_i \text{ or } \mathbf{1} - \mathbf{e}_i \text{ as a row;} \\ 1, & \text{otherwise.} \end{cases}$$

The sum $X = \sum_{C \in \binom{\mathcal{C}}{3}} \sum_{1 \le i \le 3} X(C,i)$, is an upper bound on the number of subsets $C \in \binom{\mathcal{C}}{3}$ for which $M(C)$ contains neither $\mathbf{e}_i$ nor $\mathbf{1} - \mathbf{e}_i$, for at least one $i \in \{1,2,3\}$.

For any $C \in \binom{\mathcal{C}}{3}$ and $i \in \{1,2,3\}$, we have

$$\mathbf{Pr}[X(C,i) = 1] = \frac{\sum_{u=0}^{n} \binom{n}{u}^2 \binom{2n-2u}{n-u}}{\binom{2n}{n}^2}. \tag{9.1}$$

To see this, suppose without loss of generality that $C = \{1,2,3\}$ and $i = 3$. Permute the rows of $M(C)$, if necessary, so that its first column consists of n zeroes followed by n ones. The total number of choices for the other 2 columns is $\binom{2n}{n}^2$. Let u be the number of common zeroes in columns one and two, and hence also the number of common ones in columns one and two. Then $X(C,i) = 1$ if and only if the zeroes in column 3 are disjoint from the common ones, and the ones in column 3 are disjoint from the common zeroes. This event can happen in $\binom{n}{u}^2 \binom{2n-2u}{n-u}$ ways.

Asymptotically, the sum (9.1) is dominated by the terms near $u = \alpha n$. Let $\mathcal{H}(x) = -x \log x - (1-x)\log(1-x)$, for $0 < x < 1$, be the *binary entropy function.* Using the well-known approximation (see, for example, [108]) $\binom{n}{\alpha n} = 2^{n\mathcal{H}(\alpha) + O(\log n)}$, we have

$$\mathbf{Pr}[X(C,i) = 1] = 2^{2n(\mathcal{H}(\alpha) - 1 - \alpha) + O(\log n)}.$$

The minimum of $\mathcal{H}(\alpha) - 1 - \alpha$ occurs at $\alpha = 1/3$. Hence,

$$\mathbf{Pr}[X(C,i) = 1] \leq 2^{2n \log \frac{3}{4} + O(\log n)} = O\left(\left(\frac{3}{4}\right)^{2n}\right).$$

It follows that

$$\mathbf{E}[X] \leq O\left(m^3 \left(\frac{3}{4}\right)^{2n}\right). \tag{9.2}$$

Now choose m to be the largest integer so that $\mathbf{E}[X] \leq \epsilon m$. It is easy to see from (9.2) that it suffices to choose

$$m = \Omega\left(\left(\frac{4}{3}\right)^{n}\right). \tag{9.3}$$

It follows that, for m taking the value in (9.3), there exists a $2n \times (1+\epsilon)m$ matrix with entries from $\{0,1\}$, in which there are at most ϵm $2n \times 3$ submatrices that contain neither $\mathbf{e}_i$ nor $\mathbf{1} - \mathbf{e}_i$ as a row, for at least one $i \in \{1,2,3\}$. Hence, by deleting at most ϵm columns from this matrix, we obtain a 2-frameproof array of order $2n$ and size at least m. This gives

$$f(2n,2) \geq \Omega\left(\left(\frac{4}{3}\right)^{n}\right).$$

or

$$f(n,2) \geq \Omega\left(\left(\frac{2}{\sqrt{3}}\right)^{n}\right). \tag{9.4}$$

We summarize the foregoing results in the theorem below.

Theorem 9.3.1 There exists a family of 2-frameproof codes with rate $(1 - o(1)) \log(2/\sqrt{3})$.

The family of 2-frameproof codes supplied by Theorem 9.2.2 has a rate of 1/64. So Theorem 9.3.1 provides a substantial improvement.

9.4 Explicit Constructions for Superimposed and Frameproof Codes

In this section, we discuss constructivity issues of superimposed codes and frameproof codes. A family of codes $\{\mathcal{C}_i\}_{i=1}^{\infty}$ is said to be *constructive* if for every $\mathcal{C}_i = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m\}$, there exists an algorithm that on every input j, where $1 \leq j \leq m$, outputs the codeword $\mathbf{c}_j$ in time that is polynomial in terms of the length of $\mathcal{C}_i$. So all codes constructed using probabilistic methods are not constructive, since the corresponding algorithms may not even halt. It is possible, however, to derandomized any such algorithm to give one that is guaranteed to halt, by sampling exhaustively the underlying sample spaces. Unfortunately, the sample spaces used in probabilistic constructions of codes often have exponential size, rendering the codes not constructive.

The frameproof codes of Theorem 9.2.2 and Theorem 9.3.1 are both not constructive. The superimposed codes of Busschbach [27], Erdös, Frankl and Füredi [58], Hwang and Sós [81], Nguyen and Zeisel [107], and Dyachkov, Rykov, and Rashad [52], mentioned in Section 9.2 all involved probabilistic arguments at some point, and is therefore also not constructive. Hwang and Sós [81] actually gave a greedy algorithm for constructing the codes, but the algorithm involves an exponential size search space, and can be viewed also as a direct derandomization of the construction of Erdös, Frankl and Füredi [58].

In applications, it is desirable that constructive superimposed codes and frameproof codes be available. The frameproof codes in Theorem 9.2.3 are constructive but have rather poor rates. Boneh and Shaw provided an explicit construction for a family of frameproof codes but the rate of this family is not even bounded away from zero.

Theorem 9.4.1 (Boneh and Shaw [19]) For any $r \in \mathbf{N}$, there exists an r-frameproof code of length n with $2^{\sqrt{n}/r}$ codewords.

Stinson and Wei [142] has given a better explicit construction based on orthogonal arrays, but the rate of the code family is still not bounded away from zero.

Theorem 9.4.2 (Stinson and Wei [142]) For any prime power q and any integer $t < q$, there exists a $\lfloor q/(t-1) \rfloor$-frameproof code of length $q^2 + q$ having q^t codewords.

A family of r-superimposed codes of similar rate can be constructed using Reed-Solomon codes, or polynomial codes in general [58].

To our knowledge, there is also no known constructive families of superimposed codes whose rate is bounded away from zero. We show in this section that for all $r \in \mathbf{N}$, there exists a constructive family of r-superimposed codes whose rate is bounded away from zero. This also implies for every $r \in \mathbf{N}$, the existence of a constructive family of r-frameproof codes whose rate is bounded away from zero. We make use of the following composition construction of Kautz and Singleton [83].

Lemma 9.4.1 (Kautz and Singleton [83]) If there exist

(i) an r-superimposed code, Γ, of length n having q codewords, and

(ii) a q-ary code, $\mathcal{C}$, of length n', where $n' > r$, with relative minimum distance at least $1 - 1/r$, having N codewords,

then there exists an r-superimposed code of length nn' having N codewords.

Proof. Let $\Gamma = \{\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(q)}\}$. For each codeword $\mathbf{v} = (v_1, v_2, \ldots, v_{n'}) \in \mathcal{C}$, let $\mathbf{u}^{(\mathbf{v})} = (\mathbf{w}^{(v_1)}, \mathbf{w}^{(v_2)}, \ldots, \mathbf{w}^{(v_{n'})})$. The set $\{\mathbf{u}^{(\mathbf{v})} \mid \mathbf{v} \in \mathcal{C}\}$ is an r-superimposed code of length nn' having N codewords. □

Boneh and Shaw [19] have the same construction for r-frameproof codes.

Let $\mathcal{H}_q(x) = x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x)$, $0 < x < 1-1/q$. The following is a well-known bound in coding theory.

Theorem 9.4.3 (Zyablov [159]) For any $\delta \in [0, 1-1/q)$, there exists a family of q-ary codes of relative minimum distance δ and of rate $R \geq R_Z(\delta)$, where

$$R_Z(\delta) = \max_{\delta \leq \mu \leq 1-1/q} (1 - \mathcal{H}_q(\mu))\left(1 - \frac{\delta}{\mu}\right).$$

We have the following approximation of $R_Z(\delta)$ for δ near $1-1/q$.

Lemma 9.4.2 Let $q \geq 2$ and $\epsilon > 0$ be fixed. Then for x sufficiently small, we have

$$R_Z\left(1 - \frac{1}{q} - x\right) \geq \frac{q^3}{16(q-1)^2 \ln q} x^3 - \epsilon \tag{9.5}$$

Proof. We have

$$R_Z\left(1 - \frac{1}{q} - x\right) \geq \left(1 - \mathcal{H}_q\left(1 - \frac{1}{q} - \frac{x}{2}\right)\right)\left(1 - \frac{1 - \frac{1}{q} - x}{1 - \frac{1}{q} - \frac{x}{2}}\right). \tag{9.6}$$

The series expansion of the right hand side in (9.6), with respect to x, about the point zero, is

$$\frac{q^3}{16(q-1)^2 \ln q} x^3 - \frac{(q-5)q^4}{96(q-1)^3 \ln q} x^4 + \cdots,$$

which gives the required result. □

We need an explicit construction of Alon, Bruck, Naor, Naor, and Roth [6].

Theorem 9.4.4 (Alon et al. [6]) Let q be a prime power. Then there exists a constructive family of q-ary codes of relative minimum distance δ and rate at least $\gamma R_Z(\delta)$, where $\gamma = 1/(22 + 10\sqrt{5})$.

Theorem 9.4.5 For any $r \in \mathrm{N}$, there exists a constructive family of r-superimposed codes whose rate is bounded away from zero.

Proof. We use Lemma 9.4.1 to produce such a family of r-superimposed codes. Let $r \in \mathrm{N}$, and q be the smallest prime power at least $r + 1$. The rows of a $q \times q$ identity matrix give an r-superimposed code of length q having q codewords. Take this as our first ingredient for Lemma 9.4.1. We now give the second ingredient required by Lemma 9.4.1.

Let $\epsilon > 0$. Choose $x \leq 1/q(q-1)$ sufficiently small so that (9.5) holds. Theorem 9.4.4 gives an explicit family of q-ary codes of relative minimum distance $1 - 1/q - x$ and rate greater than $\gamma R_Z(1 - 1/q - x)$. Therefore, the number of codewords in a code of length n in this family is at least

$$q^{\gamma n((qx)^3/16(q-1)^2 \ln q - \epsilon)} = \Omega(a^n),$$

for some constant a. Now,

$$1 - \frac{1}{q} - x \geq 1 - \frac{1}{q} - \frac{1}{q(q-1)} = 1 - \frac{1}{q-1} \geq 1 - \frac{1}{r},$$

So we can take this family of codes as the second ingredient for Lemma 9.4.1. It follows that there is a family of r-superimposed codes for which each code in this family of length qn has at least $\Omega(a^n)$ codewords. It follows that this family of r-superimposed codes has rate bounded away from zero.

The constructivity of this construction follows from that of the composition construction (Lemma 9.4.1) and the construction of Alon et al. (Theorem 9.4.4). □

Corollary 9.4.1 For any $r \in \mathbb{N}$, there exists a constructive family of r-frameproof codes whose rate is bounded away from zero.

The family of codes in Theorem 9.4.5 is in fact explicit since the construction of Alon et al. (Lemma 9.4.4) is explicit. The constant a in the proof of Theorem 9.4.5 is probably very bad. But Theorem 9.4.5 and Corollary 9.4.1 appears to be the first explicit families of r-superimposed codes and r-frameproof codes whose rates are bounded away from zero.

9.5 Remarks

In this chapter, we have given and analyzed a probabilistic construction for 2-frameproof codes. The bound we obtained with this construction is the best currently known. We have also established for every r, the first explicit families of r-superimposed codes and r-frameproof codes whose rates are bounded away from zero. Frameproof codes are studied only very recently, and many combinatorial properties remain to be discovered.

CHAPTER 10

Conclusion

This dissertation examines three problems in computer science that have received much attention recently. The first is the study of nonadaptive algorithms for different group testing models. The second is the design of erasure-resilient codes for large disk arrays. The final problem is that of constructing frameproof codes for fingerprinting of digital data. All of these problems yield a unified treatment as Turán-type problems. We obtained new and improved results on several Turán-type problems that arise from these applications. These results, when interpreted in the context of group testing, give either stronger bounds on the test complexity of nonadaptive group testing algorithms, or characterizations of nonadaptive group testing algorithms with optimal test complexity. Nonadaptive algorithms for new models of group testing are also obtained. Our results on the construction of erasure-resilient codes for large disk arrays established many families of codes with as many codewords as possible (up to a constant factor), having optimal update penalties. We have also shown how some properties of our erasure-resilient codes, namely, the existence of α-balanced orderings, can be used to trade the number of codewords for smaller α-balanced group sizes. In another connection, we illustrated that erasure-resilient codes can be used to construct nonadaptive algorithms for group testing problems in which the test function used is the MOD_2 test function. Our contribution in frameproof codes is the establishment of a better bound for 2-frameproof codes, as well as the exhibition of the first explicit family of r-frameproof codes whose rate is bounded away from zero.

The most exciting aspect of this research[3] is the serendipitous discovery that many mathematical problems (Turán-type problems in particular), studied long and not so long ago by mathematicians purely for reasons of aesthetics, are equivalent to problems faced by designers of erasure-resilient codes in practice. By this, we do not mean taking a mathematical problem and trying to come up with a problem corresponding to it that is perhaps practical. Such an approach, which unfortunately is quite pervasive, often gives artificial problems that never actually occur in real life. Rather, Hellerstein et al. [79] begin with the problem of designing erasure-resilient codes for large disk arrays, working their way through the requirements of the codes, and arrive at certain properties which must be possessed by the parity-check matrices. We carry this step further by examining the set systems corresponding to these parity-check matrices. The result is the discovery that some of what are desired are combinatorial designs that have been studied for quite some time without any apparent applications in mind. The same can be said about the work of Hwang and Sós [81] who demonstrated that r-union-free set systems correspond to nonadaptive algorithms for some group testing models.

[3]This opinion is expressed by C. J. Colbourn in his talk "Erasure Codes", presented at the University of Toronto Department of Computer Sciences Spring Colloquia on February 13, 1996 and in the Tutte Colloquium at the University of Waterloo on February 16, 1996. Part of what follows is a paraphrase of excerpts from his talk.

10.1 Open Problems

We list two categories of open problems. The first, which is contained in Section 10.1.1, concerns group testing. The second, in Section 10.1.2 concerns erasure-resilient codes. We do not attempt to suggest any open problems here for frameproof codes, since the area is still in its infancy, and almost any problem one can conceive is open.

10.1.1 Group Testing

For a given r, let us define

$$m(r) = \min\{n \mid \text{there exists an } r\text{-cover-free set system of order } n \text{ with at least } n+1 \text{ blocks}\}.$$

Erdös, Frankl, and Füredi [58] raised the following open problem.

Open Problem 10.1.1 Is $\lim_{r\to\infty} \frac{m(r)}{r} = 1$ or even $m(r) \geq (r+1)^2$?

The importance of this problem for group testing stems from the following observation. Consider an instance of UNRESTRICTED NONADAPTIVE EXACT IDENTIFICATION PROBLEM(r), (X, r, f, Π). If we have fewer than $m(r)$ elements to test, that is, $|X| \leq m(r)$, then by the definition of $m(r)$, any nonadaptive algorithm based on an r-cover-free set system must use at least $|X|$ tests. Hence, we can do no better than the naive algorithm which tests every element individually. Therefore, $m(r)$ determines when nonadaptive algorithms based on r-cover-free set systems become useful. Erdös, Frankl, and Füredi claimed (see [58]) that

$$(1+o(1))\frac{5}{6}r^2 < m(r) < r^2 + o(r),$$

but no proof for the lower bound is published. The upper bound can be obtained from a 2-$(q^2, q, 1)$ design (an affine plane of order q), where q is the smallest prime power at least $r+1$. Erdös, Frankl, and Füredi (see [58]) also claimed to have shown that $m(r) \geq (r+1)^2$ for $r \leq 3$. Again, no proof appears in print.

The next problem is suggested by our results in Chapter 5.

Open Problem 10.1.2 Is it true that for every r, there is a constant N, depending only on r, such that for all $n > N$, an r-cover-free $(r+1)$-uniform set system of order n is optimal if and only if it is an optimal 2-$(n, r+1, 1)$ packing?

We have made substantial progress on the existence problem for weakly union-free twofold triple systems. It would be interesting to improve our results.

Open Problem 10.1.3 Determine the existence of weakly union-free TTS(n) for those values of n not decided by Theorem 6.6.3.

In particular, does there exist a weakly union-free TTS(12)?

10.1.2 Erasure-Resilient Codes

In view of the equivalence between $(3, 5)$-erasure-resilient codes and anti-Pasch 2-$(n, 3, 1)$ packings, we propose the following problems.

Open Problem 10.1.4 Determine those n for which there exists an anti-Pasch optimal 2-$(n, 3, 1)$ packing.

Open Problem 10.1.5 Determine those $n \equiv 3 \pmod 6$ for which there exists an anti-Pasch Kirkman triple system of order n.

Let q be a prime power and $(X, \mathcal{A})$ be the set system defined as follows:

$$X = \mathrm{GF}(q) \times \mathbf{Z}_k, \qquad \text{and}$$

$$\mathcal{A} = \{\{(a,0),(b,1),(a+b,2),(a+2b,3),\ldots,(a+(k-2)b,k-1)\} \mid a,b \in \mathrm{GF}(q)\}.$$

It is easy to see that $(X, \{\mathrm{GF}(q) \times \{i\} \mid i \in \mathbf{Z}_k\}, \mathcal{A})$ is a $\mathrm{TD}(k,q)$.

Open Problem 10.1.6 Is the set system $(X, \mathcal{A})$ defined above that of a $[q^2, kq, k, 2k-1]$-erasure-resilient code?

A positive answer to Open Problem 10.1.6 would imply that

$$F(c,k,2k-1) \geq \frac{1}{k^2}c^2 - O(c),$$

which compares favourably with the upper bound $F(c,k,2k-1) \leq \frac{1}{k(k-1)}c^2$ of Theorem 8.5.2. We have shown that the answer to Open Problem 10.1.6 is yes if $k=3$ (this is implicit in the proof of Lemma 8.7.3) or $k=4$ (Theorem 8.7.2). A more difficult problem is:

Open Problem 10.1.7 Prove that

$$F(c,k,l) = \Theta(c^{k+1-\lfloor l/2 \rfloor}). \tag{10.1}$$

An even harder problem is to determine the constant factor in (10.1). We know that it is 1/6 for $k=3$ (and any $l \leq 5$), but we know nothing for $k \geq 4$.

10.2 Future Directions

Sequential and nonadaptive algorithms for group testing lie at the two extreme ends of a spectrum of algorithms. Sequential algorithms are not limited by the number of steps they take, but are restricted to only one processor. Nonadaptive algorithms, on the other hand, must obtain all necessary information in one step, but can have any number of processors. The goal is to minimize the unrestricted resource, that is the number of steps for sequential algorithms, and the number of processors for nonadaptive algorithms. We can define an s-step group testing algorithm to be one that is limited to s steps. The problem then is to design, for any given s, an s-step algorithm that finds the target set using the minimum possible total number of tests. Such algorithms have been considered in other areas of computer science. For example, the problem of designing s-step algorithms for sorting (called *sorting in rounds*), has been studied in [5, 18, 73].

Two-step algorithms have been considered before for group testing as well [25]. However, the concept there is different. In [25], pools are designed probabilistically for the first step. The second step is used to test individually those elements for which membership in the target set cannot be decided after the first step.

Other important issues in group testing that demand further study are given in a recent article written by Hwang [80] for CADCOM (Committee for the Advancement of Discrete and Combinatorial Mathematics).

Quite recently, Buhrman, Hemaspaandra, and Longpré [26] have used r-cover-free set systems to show that any sparse set is conjunctive truth-table reducible to a tally set, thus refuting two conjectures of Ko [86] in complexity theory. Chaudhuri and Radhakrishnan [33] have also used r-cover-free set systems to derive new lower bounds for the circuit complexity of threshold functions. We are hopeful that more intimate connections between Turán-type problems and other problems in computer science will be uncovered.

APPENDIX A

Packing Pairs by Quintuples: $n \equiv 19 \pmod{20}$

Concerning the determination of $D(n,5,2)$, there has not been any explicit work done on the case $n \equiv 19 \pmod{20}$. The reason is that no example of a 2-$(n,5,1)$ packing with $U(n,5,2)$ blocks is known in this case. For our application, the asymptotic existence of 2-$(n,5,1)$ packings with at least $U(n,5,2) - o(n)$ blocks suffices (see Section 5.3.2). In this appendix, we prove that there is a constant a so that $D(n,5,2) \geq U(n,5,2) - a$ for all sufficiently large $n \equiv 19 \pmod{20}$. We assume knowledge of various designs defined in Section 6.6.

Definition A.0.1 Let n and m be nonnegative integers. A *maximum incomplete packing*, denoted by $\mathsf{MIP}(v,w)$, is a triple $(X, Y, \mathcal{A})$, where $|X| = n$, $Y \subseteq X$ with $|Y| = m$, and $(X, \mathcal{A})$ is a 5-uniform set system with $U(n,5,2) - U(m,5,2)$ blocks so that,

(i) each 2-subset of X not contained in Y is contained in at most one block of $\mathcal{A}$, and

(ii) no block contains any 2-subset of Y.

The concept of maximum incomplete packings originated in the work of Yin [156]. The following lemmata can be found in the work of Mullin and Yin [105].

Lemma A.0.1 (Mullin and Yin [105]) Suppose that there exist a $\{5\}$-GDD of type $g_1 g_2 \cdots g_s$ and an $\mathrm{MIP}(q+g_i, q)$ for each i, $1 \le i \le s-1$. Then there exists an $\mathrm{MIP}\left(q + \sum_{i=1}^{s-1} g_i, q + g_s\right)$.

Lemma A.0.2 (Mullin and Yin [105]) Suppose that there exists a $\mathrm{TD}(6,t)$ and $0 \le u \le t$. Then an $\mathrm{MIP}(20t+4u+q, 4u+q)$ exists if an $\mathrm{MIP}(4t+q, q)$ exists.

Lemma A.0.3 (Mullin and Yin [105]) There exists an $\mathrm{MIP}(79, 19)$.

We also use a recent result of Yin et al. [158].

Theorem A.0.1 (Yin et al. [158]) There exists a $\{5\}$-GDD of type 60^s for all $s \ge 5$.

Corollary A.0.1 There exists an $\mathrm{MIP}(60s+19, 19)$ for all $s \ge 5$.

Proof. Let $s \ge 5$. Since there is a $\{5\}$-GDD of type $60^s 0^1$ (Theorem A.0.1) and an $\mathrm{MIP}(19+60, 19)$ (Lemma A.0.3), Lemma A.0.1 implies the existence of an $\mathrm{MIP}(60s+19, 19)$. □

The following is the main result of this appendix.

Theorem A.0.2 For all $n \ge 319$ such that $n \equiv 19 \pmod{20}$, there exists an $\mathrm{MIP}(n, m)$, where $m \equiv 19 \pmod{20}$ and $19 \le m \le 299$.

Proof. Every $n \ge 319$, $n \equiv 19 \pmod{20}$, can be written in the form $20t+4u+19$, where $t \equiv 0 \pmod{15}$, and $0 \le u \le 70$, $u \equiv 0 \pmod 5$. By Corollary A.0.1, there exists an $\mathrm{MIP}(4t+19, 19)$. Since there exists a $\mathrm{TD}(6,t)$ for all positive $t \equiv 0 \pmod{15}$ [1], it then follows from Lemma A.0.2 that there exists an $\mathrm{MIP}(n, 4u+19)$. □

Corollary A.0.2 There exists a constant a such that for all $n \ge 319$, $n \equiv 19 \pmod{20}$, $D(n,5,2) \ge U(n,5,2) - a$.

Proof. For all $n \geq 319$, $n \equiv 19 \pmod{20}$, we have an MIP(n, m), where $m \equiv 19 \pmod{20}$ and $19 \leq m \leq 299$. This MIP(n, m) is a 2-$(n, 5, 1)$ packing having at least $U(n, 5, 2) - U(299, 5, 2)$ blocks. □

Stronger results can be obtained but Corollary A.0.2 suffices for our purpose.

APPENDIX B

Enumeration of A Class of Twofold Triple Systems of Order 12

In this appendix, we determine all TTS(12) having an automorphism group whose order is divisible by an odd prime.

B.1 Structure of Automorphism Groups

Let Γ be the full automorphism group of a TTS(12). We develop some facts about the structure of Γ.

Lemma B.1.1 Let α be an automorphism of a TTS(12), where α has prime order $p \geq 3$. Then α fixes 0 or 1 point.

Proof: Let $(X, \mathcal{B})$ be a TTS(12) with α as an automorphism, and F the set of fixed points of α. Let $\mathcal{B}_i = \{B \in \mathcal{B} \mid |B \cap F| = i\}$, $0 \leq i \leq 3$, and let $b_i = |\mathcal{B}_i|$. Henceforth, we assume that $f = |F| \geq 2$, since if $f \leq 1$, then the lemma easily holds. So let $A \in \binom{F}{2}$ and let B be any block in $\mathcal{B}$ such that $A \subseteq B$. Then the three blocks B, $\alpha(B)$, and

$\alpha^2(B)$ all contain A, which is impossible unless $B \subseteq F$. Hence, $b_2 = 0$ and $(F, \mathcal{B}_3)$ is a TTS(f). The necessary conditions for the existence of a TTS(f) with $f < 12$ require that $f = 4, 6, 7, 9$, or 10. But to cover pairs from $X \setminus F$ we need

$$3b_0 + b_1 = \binom{12-f}{2},$$

and

$$b_0 + b_1 + b_3 = 44,$$

which cannot be satisfied for any $f \in \{4, 6, 7, 9, 10\}$. □

Theorem B.1.1 Let p be a prime dividing the order of the automorphism group Γ of a TTS(12). Then $p \in \{2, 3, 11\}$. Furthermore, for $\alpha \in \Gamma$, we have

(i) α fixes no points if α has order 3, and

(ii) α fixes 1 point if α has order 11.

Proof: Let α be an automorphism of order p of a TTS(12). If $p = 5$ or 7, then α fixes f points, where $f \in \{2, 5, 7\}$. Our result then follows from Lemma B.1.1. □

B.2 Enumeration

Having established the structure of Γ, we proceed to enumerate all TTS(12) having an automorphism group whose order is divisible by an odd prime. Henceforth, any TTS(12), $(X, \mathcal{B})$, under consideration has point set $X = \{0, 1, \ldots, 9, a, b\}$. When Γ contains an automorphism of order 11, the TTS(12) is 1-rotational and all nonisomorphic 1-rotational TTS(12) have been enumerated by Chee and Royle [34]. There are precisely five nonisomorphic 1-rotational TTS(12). So it remains to examine the case when Γ contains an automorphism α of order three. Without loss of generality, let

$$\alpha = (\ 0\ \ 1\ \ 2\ )(\ 3\ \ 4\ \ 5\ )(\ 6\ \ 7\ \ 8\ )(\ 9\ \ a\ \ b\ ),$$

and $G = \langle \alpha \rangle \leq \Gamma$.

From the orbit structure of G on $\binom{X}{3}$, we see that $\mathcal{B}$ must consist of two orbits of length one, and 14 orbits of length three on $\binom{X}{3}$. Without loss of generality, the two orbits of length one are taken to be $\{0,1,2\}$ and $\{3,4,5\}$. The pairs $\{0,1\}$ and $\{3,4\}$ must appear in some other blocks, and thus we can assume that the set of starter blocks for $\mathcal{B}$ contains $\{0,1,2\}$, $\{3,4,5\}$, $\{0,1,\star\}$, and $\{3,4,\star\}$. Filling in the stars in all possible ways leads to five nonisomorphic starting configurations which are given below.

starting configuration A	starting configuration B	starting configuration C	starting configuration D	starting configuration E
$\{0,1,2\}$	$\{0,1,2\}$	$\{0,1,2\}$	$\{0,1,2\}$	$\{0,1,2\}$
$\{3,4,5\}$	$\{3,4,5\}$	$\{3,4,5\}$	$\{3,4,5\}$	$\{3,4,5\}$
$\{0,1,3\}$	$\{0,1,3\}$	$\{0,1,3\}$	$\{0,1,6\}$	$\{0,1,6\}$
$\{0,3,4\}$	$\{1,3,4\}$	$\{3,4,6\}$	$\{3,4,6\}$	$\{3,4,9\}$

Beginning with each starting configuration above, we try to complete to a TTS(12) by adding 12 more starter blocks using a backtracking algorithm. All the TTS(12) constructed are subjected to an isomorphism test to sieve out isomorphic designs. This is carried out with McKay's Nauty program [99]. The result is that there are precisely 775 nonisomorphic TTS(12) having G as an automorphism group, 36 with starting configuration A, 16 with starting configuration B, 540 with starting configuration C, 88 with starting configuration D, and 95 with starting configuration E. The starter blocks for these designs are given in Sections B.3.1 through B.3.5. We only list the starter blocks required to complete the respective starting configurations to a TTS(12). To reduce space utilization in the tables, we omit braces and commas, and write a set $\{x, y, z\}$ as xyz.

None of the 1-rotational TTS(12) have an automorphism group whose order is divisible by three. Consequently, we have the following theorem.

Theorem B.2.1 There are exactly 780 isomorphism classes of TTS(12) admitting an automorphism group whose order is divisible by an odd prime.

An attempt was made to enumerate all TTS(12) with an automorphism of order two. In this case, we have about 80 starting configurations to try to complete to a TTS(12). When run on each of the first few starting configurations we picked, our backtracking algorithm did not stop even after a week, and we decided to abandon the search.

B.3 Catalogues

B.3.1 Starter Blocks for TTS(12) With Starting Configuration A

Design #	Starter Blocks
1	379 136 378 078 369 069 06a 38a 23b 07b 0ab 3ab
2	379 136 368 068 069 07a 38a 39a 23b 37b 07b 0ab
3	379 136 368 068 079 37a 06a 39a 23b 07b 38b 0ab
4	379 236 378 078 13a 36a 06a 09a 08b 06b 39b 38b
5	379 137 067 368 089 36a 07a 38a 08b 23b 39b 0ab
6	379 137 368 068 089 23a 07a 39a 09a 36b 07b 38b
7	379 137 368 078 239 079 36a 06a 08b 09b 38b 3ab
8	379 137 368 078 089 23a 06a 07a 39a 36b 09b 38b
9	379 237 067 368 079 13b 36a 06a 39a 08b 09b 38b
10	379 237 368 068 13a 06a 07a 39a 36b 07b 09b 38b
11	379 378 067 238 13b 089 36a 07a 39a 08b 36b 09b
12	379 378 138 078 369 069 089 23a 06a 36b 0ab 3ab
13	379 367 138 068 36a 06a 07a 38a 08b 23b 39b 09b
14	379 367 138 078 239 069 089 36a 06b 38b 0ab 3ab

15	136 378 067 069 37a 389 23a 08a 09a 36b 07b 39b
16	136 367 067 389 07a 38a 39a 09a 08b 23b 06b 37b
17	136 368 078 239 069 37a 38a 09a 06b 37b 07b 39b
18	236 378 067 369 13a 089 08a 39a 06b 37b 38b 0ab
19	236 378 067 369 13a 06a 38a 08a 08b 37b 39b 09b
20	137 378 067 389 36a 07a 08a 39a 08b 36b 23b 09b
21	137 067 368 239 37a 089 38a 09a 08b 36b 07b 39b
22	137 368 068 369 079 089 23a 07a 38a 37b 39b 0ab
23	137 368 078 37a 089 389 07a 39a 36b 23b 06b 0ab
24	237 378 067 139 369 079 36a 06a 08a 09b 38b 3ab
25	237 368 068 079 13b 37a 389 36a 07a 06b 39b 09b
26	378 367 238 139 069 089 07a 08a 39a 36b 07b 3ab
27	378 367 238 069 079 13b 36a 08a 39a 08b 07b 39b
28	378 138 068 239 369 37a 089 09a 36b 06b 07b 3ab
29	367 067 138 239 069 089 38a 39a 08b 36b 37b 0ab
30	367 238 068 369 13a 07a 38a 08a 37b 07b 39b 09b
31	367 238 078 139 069 079 36a 08a 39a 37b 38b 0ab
32	367 138 068 239 369 089 07a 38a 06b 37b 09b 3ab
33	367 138 068 239 079 37a 389 06a 08b 36b 09b 3ab
34	367 138 078 239 369 069 089 38a 06b 37b 0ab 3ab
35	238 368 068 139 079 37a 089 07a 39a 36b 37b 0ab
36	138 368 078 239 069 37a 089 39a 36b 06b 37b 0ab

B.3.2 Starter Blocks for TTS(12) With Starting Configuration B

Design #	Starter Blocks
1	379 236 378 068 089 23a 06a 07a 39a 36b 09b 38b
2	379 236 368 078 069 23a 06a 38a 09a 08b 37b 39b
3	236 378 067 239 37a 089 06a 38a 08b 36b 39b 09b
4	236 378 368 069 089 23a 07a 08a 39a 06b 37b 39b
5	236 378 068 069 37a 389 23a 08a 09a 36b 07b 39b
6	236 368 068 37a 389 07a 08a 39a 23b 06b 37b 09b
7	237 378 067 369 389 36a 07a 08a 23b 06b 09b 3ab
8	237 367 067 089 389 23a 07a 39a 09a 36b 06b 38b
9	237 367 068 239 069 079 36a 38a 07b 39b 38b 0ab
10	237 067 368 239 369 089 07a 38a 06b 37b 09b 3ab
11	237 067 078 369 069 389 23a 38a 09a 36b 37b 0ab
12	237 368 078 369 06a 07a 38a 39a 23b 06b 37b 09b
13	378 238 068 369 079 089 23a 07a 39a 36b 37b 0ab
14	367 238 078 239 079 37a 389 06a 08b 36b 09b 3ab
15	367 238 078 069 37a 389 23a 08a 09a 36b 07b 39b
16	238 368 078 239 37a 089 07a 39a 36b 06b 37b 09b

B.3.3 Starter Blocks for TTS(12) With Starting Configuration C

Design #	Starter Blocks
1	379 037 136 067 239 089 679 23a 06a 09b 38b 3ab
2	379 037 136 067 069 389 23a 08a 67a 23b 0ab 3ab
3	379 037 136 068 239 079 23a 06a 67a 09b 38b 3ab
4	379 037 136 068 069 389 23a 07a 23b 67b 0ab 3ab
5	379 037 136 068 679 23a 06a 39a 09a 23b 07b 38b
6	379 037 136 078 239 069 23a 06a 67b 09b 38b 3ab
7	379 037 136 078 069 679 389 23a 06a 23b 0ab 3ab
8	379 037 136 078 23a 06a 67a 39a 09a 23b 06b 38b
9	379 037 236 068 139 089 679 23a 06a 09a 38b 3ab
10	379 037 236 068 069 13b 389 23a 09a 08b 67b 3ab
11	379 037 236 068 13b 089 679 23a 06a 39a 09b 38b
12	379 037 236 068 13b 089 23a 67a 39a 09a 06b 38b
13	379 037 236 068 13a 089 679 23a 06a 39b 38b 0ab
14	379 037 236 068 13a 679 389 06a 08b 23b 0ab 3ab
15	379 037 236 068 13a 23a 06a 09a 08b 67b 39b 38b
16	379 037 236 068 13a 06a 39a 08b 23b 67b 38b 0ab
17	379 037 067 238 139 36a 08a 67a 09a 08b 23b 3ab
18	379 037 067 238 369 13b 089 23a 08a 67a 09b 3ab
19	379 037 067 238 369 13b 089 23a 09a 08b 67b 3ab
20	379 037 067 238 13b 36a 08a 67a 39a 08b 23b 09b
21	379 037 067 068 239 13b 089 23a 36a 09b 38b 69b
22	379 037 238 068 139 089 679 36a 07a 23b 0ab 3ab
23	379 037 238 068 139 36a 07a 09a 08b 23b 67b 3ab
24	379 037 238 068 239 079 13b 36a 67a 08b 09b 3ab
25	379 037 238 068 369 079 13b 089 23a 67a 0ab 3ab
26	379 037 238 068 369 13a 08a 67a 23b 07b 0ab 3ab
27	379 037 238 068 079 13b 36a 67a 39a 08b 23b 0ab
28	379 037 238 068 13a 089 23a 07a 67a 36b 39b 0ab
29	379 037 238 068 13a 36a 07a 08b 23b 67b 39b 0ab
30	379 037 238 078 369 069 13b 089 23a 67b 0ab 3ab
31	379 037 238 078 13b 679 36a 06a 39a 08b 23b 09b
32	379 037 238 078 13b 36a 67a 39a 09a 08b 23b 06b
33	379 037 138 068 239 679 36a 06a 08b 23b 09b 3ab
34	379 037 138 068 369 23a 06a 08a 67a 23b 09b 3ab
35	379 037 138 068 069 23a 36a 08a 67a 23b 39b 0ab
36	379 037 138 068 23a 06a 67a 39a 09a 08b 36b 23b
37	379 037 368 068 13a 23a 06a 07a 08b 23b 39b 6ab
38	379 037 368 078 239 13a 06a 69a 08b 23b 06b 3ab
39	379 037 068 078 239 13a 36a 69a 23b 06b 38b 0ab
40	379 136 237 067 039 679 23a 06a 09a 07b 38b 3ab
41	379 136 237 067 239 069 38a 67a 09a 03b 07b 3ab
42	379 136 237 067 069 389 23a 07a 09a 03b 67b 3ab

43	379	136	237	067	079	03a	23a	06a	67a	39b	09b	38b
44	379	136	237	067	679	03a	06a	39a	23b	07b	09b	38b
45	379	136	237	067	389	03a	07a	67a	23b	06b	09b	3ab
46	379	136	237	067	03a	67a	39a	09a	23b	06b	07b	38b
47	379	136	378	067	069	03a	23a	08a	69a	23b	07b	39b
48	379	136	378	067	089	03a	23a	07a	69a	23b	06b	39b
49	379	136	378	068	039	23a	06a	07a	23b	07b	69b	3ab
50	379	136	378	078	039	23a	06a	07a	69a	23b	06b	3ab
51	379	136	038	067	679	23a	06a	07a	39a	23b	37b	09b
52	379	136	067	238	079	03a	23a	67a	09a	08b	37b	39b
53	379	136	067	238	089	679	03a	23a	07a	37b	39b	09b
54	379	136	067	238	679	03a	39a	09a	08b	23b	37b	07b
55	379	136	067	068	039	23a	07a	38a	69a	23b	37b	0ab
56	379	136	238	068	079	37a	03a	67a	23b	07b	39b	0ab
57	379	136	238	078	039	069	23a	07a	37b	67b	0ab	3ab
58	379	136	238	078	239	069	679	03a	37b	07b	09b	3ab
59	379	136	238	078	03a	07a	67a	39a	23b	06b	37b	09b
60	379	036	137	078	239	079	089	23a	67a	38b	0ab	3ab
61	379	036	137	078	239	38a	67a	09a	08b	23b	07b	3ab
62	379	036	137	078	089	23a	07a	39a	23b	67b	38b	0ab
63	379	036	237	067	239	13a	089	679	07b	38b	0ab	3ab
64	379	036	237	067	239	13a	08a	67a	07b	09b	38b	3ab
65	379	036	237	067	079	13b	389	23a	67a	09a	08b	3ab
66	379	036	237	067	13b	089	679	23a	39a	09a	07b	38b
67	379	036	237	068	239	13a	07a	07b	67b	09b	38b	3ab
68	379	036	237	068	079	13b	389	23a	07a	67a	09b	3ab
69	379	036	237	068	13b	07a	38a	67a	39a	23b	07b	09b
70	379	036	237	068	13a	23a	07a	09a	07b	67b	39b	38b
71	379	036	237	078	239	069	13b	38a	67a	07b	09b	3ab
72	379	036	237	078	069	079	13b	389	23a	67a	0ab	3ab
73	379	036	237	078	079	13a	23a	06a	67a	39b	38b	0ab
74	379	036	237	078	13a	679	23a	06a	09a	07b	39b	38b
75	379	036	237	078	13a	389	07a	67a	23b	06b	0ab	3ab
76	379	036	237	078	13a	23a	06a	07a	67b	39b	09b	38b
77	379	036	378	067	13b	089	23a	07a	39a	08b	23b	6ab
78	379	036	378	078	239	13a	06a	08b	23b	07b	69b	3ab
79	379	036	378	078	069	13b	23a	08a	39a	69a	23b	07b
80	379	036	378	078	13b	089	23a	07a	39a	69a	23b	06b
81	379	036	067	078	13b	37a	389	23a	08a	69a	23b	09b
82	379	036	238	078	139	37a	08a	67a	09a	23b	07b	3ab
83	379	036	238	078	13b	37a	089	23a	07a	67b	39b	09b
84	379	036	138	068	239	37a	089	679	23b	07b	0ab	3ab
85	379	036	138	078	239	069	089	679	23a	37b	0ab	3ab
86	379	036	138	078	069	23a	08a	67a	39a	23b	37b	0ab
87	379	236	137	067	239	089	38a	67a	09a	03b	08b	3ab

88	379	236	137	067	089	03a	23a	09a	08b	67b	39b	38b
89	379	236	137	067	089	03a	38a	67a	08b	23b	39b	0ab
90	379	236	137	067	089	03a	39a	08b	23b	67b	38b	0ab
91	379	236	137	068	039	23a	08a	67a	09a	07b	38b	3ab
92	379	236	137	068	089	679	03a	23a	09a	07b	39b	38b
93	379	236	137	068	089	23a	07a	38a	67a	09a	03b	39b
94	379	236	137	068	089	07a	38a	67a	39a	03b	23b	0ab
95	379	236	137	078	239	069	089	38a	67a	03b	0ab	3ab
96	379	236	137	078	069	089	03a	23a	67b	39b	38b	0ab
97	379	236	137	078	089	679	03a	23a	06a	39b	09b	38b
98	379	236	137	078	089	03a	67a	39a	23b	06b	38b	0ab
99	379	236	137	078	679	389	03a	06a	08b	23b	09b	3ab
100	379	236	137	078	03a	06a	38a	67a	08b	23b	39b	09b
101	379	236	237	067	139	069	089	03a	67b	38b	0ab	3ab
102	379	236	237	067	039	13a	06a	08b	67b	38b	0ab	3ab
103	379	236	237	067	069	13b	389	03a	08b	67b	09b	3ab
104	379	236	237	067	13a	089	03a	67a	06b	39b	38b	0ab
105	379	236	237	067	13a	679	389	06a	09a	03b	08b	3ab
106	379	236	237	067	13a	06a	39a	09a	03b	08b	67b	38b
107	379	236	237	068	139	069	03a	09a	07b	67b	38b	3ab
108	379	236	237	068	139	079	03a	06a	67a	09b	38b	3ab
109	379	236	237	068	069	13b	03a	38a	67a	07b	39b	09b
110	379	236	237	068	13a	679	389	06a	07a	03b	09b	3ab
111	379	236	237	068	13a	389	07a	67a	09a	03b	06b	3ab
112	379	236	237	078	139	069	06a	38a	67a	09a	03b	3ab
113	379	236	237	078	069	13b	389	03a	67a	06b	09b	3ab
114	379	236	237	078	13a	06a	67a	39a	09a	03b	06b	38b
115	379	236	378	067	13a	06a	08a	39a	69a	03b	08b	23b
116	379	236	378	068	039	069	13b	23a	08a	69a	07b	3ab
117	379	236	378	068	039	079	13b	23a	06a	69a	08b	3ab
118	379	236	378	068	039	13b	089	23a	07a	69a	06b	3ab
119	379	236	378	068	239	079	13a	06a	69a	03b	08b	3ab
120	379	236	378	068	239	13a	089	07a	69a	03b	06b	3ab
121	379	236	378	068	069	13b	089	23a	07a	39a	03b	6ab
122	379	236	378	078	069	13b	089	03a	23a	69a	06b	39b
123	379	236	038	067	139	37a	06a	08a	67a	23b	09b	3ab
124	379	236	038	068	069	079	13b	23a	67a	39a	37b	0ab
125	379	236	038	068	13a	679	23a	06a	07a	37b	39b	09b
126	379	236	038	078	13b	37a	06a	67a	39a	23b	06b	09b
127	379	236	038	078	37a	13a	06a	67a	23b	06b	39b	0ab
128	379	236	067	238	13a	03a	08a	67a	08b	37b	39b	09b
129	379	236	238	068	039	13a	07a	08b	37b	67b	0ab	3ab
130	379	236	238	078	039	13a	679	06a	08b	37b	0ab	3ab
131	379	236	238	078	37a	13a	06a	08a	67a	03b	39b	09b
132	379	236	238	078	13a	679	03a	06a	08b	37b	39b	09b

133	379 236 138 068 239 069 37a 089 03b 67b 0ab 3ab
134	379 236 138 068 239 37a 089 679 06a 03b 09b 3ab
135	379 236 138 068 069 089 679 03a 23a 37b 39b 0ab
136	379 236 138 068 37a 089 679 23a 06a 09a 03b 39b
137	379 236 068 078 139 069 03a 23a 09a 37b 38b 6ab
138	379 236 068 078 039 069 13b 23a 38a 69a 37b 0ab
139	379 236 068 078 039 13a 23a 06a 37b 38b 0ab 69b
140	379 236 068 078 239 069 13a 38a 69a 03b 37b 0ab
141	379 137 038 067 369 089 23a 07a 23b 67b 0ab 3ab
142	379 137 038 067 089 23a 07a 67a 39a 36b 23b 0ab
143	379 137 038 067 23a 36a 07a 09a 08b 23b 67b 39b
144	379 137 038 068 679 23a 36a 07a 09a 23b 07b 39b
145	379 137 067 368 239 089 03a 08b 23b 07b 69b 3ab
146	379 137 067 078 039 23a 36a 08a 69a 23b 38b 0ab
147	379 137 067 078 239 089 03a 23a 69a 36b 09b 38b
148	379 137 067 078 239 089 23a 36a 09a 03b 38b 69b
149	379 137 238 078 039 089 23a 07a 67a 36b 0ab 3ab
150	379 137 238 078 369 03a 08a 67a 23b 07b 09b 3ab
151	379 137 238 078 089 23a 07a 67a 39a 09a 03b 36b
152	379 137 368 078 239 069 03a 08b 23b 07b 6ab 3ab
153	379 137 068 078 389 03a 23a 07a 69a 36b 23b 09b
154	379 237 038 067 139 36a 07a 67a 09a 23b 06b 3ab
155	379 237 038 067 069 13b 679 23a 36a 09a 07b 39b
156	379 237 038 067 13a 23a 06a 07a 67a 36b 39b 09b
157	379 237 067 238 139 089 03a 07a 67a 36b 09b 3ab
158	379 237 067 238 039 079 13b 36a 67a 08b 0ab 3ab
159	379 237 067 238 369 13a 08a 67a 09a 03b 07b 3ab
160	379 237 067 138 039 23a 06a 67a 09a 08b 36b 3ab
161	379 237 067 138 239 069 089 679 36a 03b 0ab 3ab
162	379 237 067 368 139 079 03a 06a 69a 08b 23b 3ab
163	379 237 067 368 039 079 13b 23a 06a 69a 08b 3ab
164	379 237 067 368 069 13b 03a 39a 08b 23b 07b 6ab
165	379 237 067 068 139 03a 36a 07a 23b 09b 38b 6ab
166	379 237 067 068 139 03a 36a 09a 23b 07b 38b 69b
167	379 237 067 068 239 079 13a 36a 69a 03b 38b 0ab
168	379 237 067 068 239 13a 36a 09a 03b 07b 38b 69b
169	379 237 067 068 369 13b 03a 38a 69a 23b 07b 09b
170	379 237 067 068 369 13a 03a 23b 07b 38b 0ab 6ab
171	379 237 067 068 13a 389 36a 07a 03b 23b 0ab 69b
172	379 237 067 078 139 069 03a 36a 23b 38b 0ab 6ab
173	379 237 067 078 039 13b 23a 36a 09a 69a 06b 38b
174	379 237 067 078 369 13a 06a 38a 69a 03b 23b 0ab
175	379 237 238 068 369 13a 07a 09a 03b 07b 67b 3ab
176	379 237 238 068 13a 07a 67a 39a 09a 03b 36b 07b
177	379 237 238 078 139 069 36a 07a 09a 03b 67b 3ab

178	379	237	238	078	369	13a	679	06a	09a	03b	07b	3ab
179	379	237	238	078	13a	679	36a	06a	07a	03b	39b	09b
180	379	237	138	068	369	069	03a	23b	07b	67b	0ab	3ab
181	379	237	138	068	369	679	03a	06a	23b	07b	09b	3ab
182	379	237	138	068	069	679	03a	36a	23b	07b	39b	0ab
183	379	237	138	068	069	03a	23a	67a	09a	36b	07b	39b
184	379	237	138	078	239	069	36a	67a	09a	03b	06b	3ab
185	379	237	138	078	369	069	23a	06a	09a	03b	67b	3ab
186	379	237	138	078	069	679	23a	36a	06a	09a	03b	39b
187	379	038	367	067	13a	23a	06a	07a	08b	23b	39b	6ab
188	379	038	367	068	239	069	079	13b	23a	69a	07b	3ab
189	379	038	367	068	239	13a	07a	69a	23b	06b	07b	3ab
190	379	038	367	068	13a	23a	06a	07a	23b	07b	39b	69b
191	379	038	367	078	069	13b	23a	06a	39a	23b	07b	69b
192	379	038	367	078	13a	23a	06a	07a	69a	23b	06b	39b
193	379	038	067	068	369	13b	37a	23a	09a	23b	07b	6ab
194	379	038	067	078	239	13b	37a	36a	69a	23b	06b	09b
195	379	367	067	238	079	13b	089	03a	23a	69a	08b	39b
196	379	367	067	138	039	23a	06a	08a	69a	08b	23b	3ab
197	379	367	067	138	239	069	089	23a	08a	69a	03b	3ab
198	379	367	067	138	239	089	03a	69a	08b	23b	06b	3ab
199	379	367	238	068	13a	03a	07a	08b	23b	07b	39b	6ab
200	379	367	238	068	13a	07a	08a	39a	69a	03b	23b	07b
201	379	367	238	078	039	13b	089	23a	07a	69a	06b	3ab
202	379	367	238	078	069	13b	03a	39a	08b	23b	07b	6ab
203	379	367	138	068	089	23a	06a	07a	39a	03b	23b	69b
204	379	367	138	078	239	069	089	23a	06a	03b	69b	3ab
205	379	367	138	078	069	03a	23a	06a	08b	23b	39b	6ab
206	379	067	238	078	239	13a	03a	69a	08b	36b	37b	09b
207	379	067	238	078	239	13a	36a	09a	03b	08b	37b	69b
208	379	067	138	068	239	37a	089	23a	09a	69a	03b	36b
209	379	067	138	068	239	37a	36a	08a	69a	03b	23b	09b
210	379	067	138	068	369	03a	23a	09a	08b	23b	37b	6ab
211	379	238	068	078	039	13a	23a	07a	69a	36b	37b	0ab
212	379	238	068	078	369	13a	03a	23b	37b	07b	0ab	69b
213	037	136	378	239	079	06a	67a	39a	08b	23b	06b	3ab
214	037	136	067	239	37a	089	23a	09a	06b	67b	39b	38b
215	037	136	067	239	37a	389	09a	08b	23b	06b	67b	3ab
216	037	136	067	069	37a	389	23a	08a	23b	67b	39b	0ab
217	037	136	068	239	069	079	23a	39a	37b	67b	38b	0ab
218	037	136	068	239	069	37a	38a	23b	07b	67b	39b	0ab
219	037	136	068	239	069	679	23a	38a	09a	37b	07b	39b
220	037	136	068	239	37a	679	389	09a	23b	06b	07b	3ab
221	037	136	068	239	07a	38a	67a	39a	23b	06b	37b	09b
222	037	136	068	679	389	23a	07a	39a	09a	23b	06b	37b

223	037	136	078	239	069	679	389	23a	09a	06b	37b	3ab
224	037	236	378	139	069	06a	08a	39a	08b	23b	67b	3ab
225	037	236	378	13a	06a	08a	67a	39a	08b	23b	06b	39b
226	037	236	068	239	13a	679	389	09a	08b	06b	37b	3ab
227	037	236	068	13b	37a	089	679	389	23a	09a	06b	39b
228	037	137	067	239	089	23a	38a	67a	09a	08b	36b	39b
229	037	137	067	239	36a	08a	39a	08b	23b	67b	09b	38b
230	037	137	067	369	389	23a	08a	09a	08b	23b	67b	3ab
231	037	137	368	239	069	089	679	23a	08a	07b	39b	3ab
232	037	137	068	239	369	079	089	23a	67b	38b	0ab	3ab
233	037	137	068	239	079	089	679	23a	36a	39b	38b	0ab
234	037	137	068	239	079	089	23a	67a	39a	36b	38b	0ab
235	037	137	068	239	079	679	389	36a	08b	23b	0ab	3ab
236	037	137	068	239	079	389	23a	67a	09a	08b	36b	3ab
237	037	137	068	239	079	36a	38a	67a	08b	23b	39b	0ab
238	037	137	068	239	089	679	389	23a	07a	36b	09b	3ab
239	037	137	078	239	069	389	23a	09a	08b	36b	67b	3ab
240	037	137	078	239	069	36a	38a	08b	23b	67b	39b	0ab
241	037	137	078	369	23a	06a	38a	09a	08b	23b	67b	39b
242	037	237	067	139	069	389	23a	08a	67a	09a	36b	3ab
243	037	237	067	139	06a	38a	67a	39a	09a	08b	36b	23b
244	037	237	067	239	369	13b	089	38a	67a	06b	09b	3ab
245	037	237	067	239	13a	679	389	06a	08b	36b	09b	3ab
246	037	237	067	239	13a	389	67a	09a	08b	36b	06b	3ab
247	037	237	067	369	069	13b	389	23a	08a	67b	09b	3ab
248	037	237	067	369	13b	089	23a	39a	09a	06b	67b	38b
249	037	237	067	069	13b	389	23a	08a	67a	39a	36b	09b
250	037	237	068	139	369	06a	07a	38a	23b	67b	09b	3ab
251	037	237	068	139	069	36a	07a	38a	23b	67b	39b	0ab
252	037	237	068	139	079	23a	06a	67a	39a	09a	36b	38b
253	037	237	068	139	36a	07a	39a	09a	23b	06b	67b	38b
254	037	237	068	069	13b	679	389	23a	39a	09a	36b	07b
255	037	237	068	069	13a	389	23a	07a	36b	67b	39b	0ab
256	037	237	068	13b	679	389	36a	07a	39a	23b	06b	09b
257	037	237	078	139	069	23a	06a	38a	67a	09a	36b	39b
258	037	237	078	239	069	13a	389	67a	36b	06b	0ab	3ab
259	037	237	078	239	13a	06a	67a	39a	36b	06b	09b	38b
260	037	237	078	369	13a	06a	38a	67a	23b	06b	39b	0ab
261	037	237	078	069	13b	679	389	23a	06a	39a	36b	09b
262	037	237	078	069	13b	679	389	36a	39a	23b	06b	0ab
263	037	237	078	069	13a	679	389	23a	06a	36b	39b	0ab
264	037	378	067	239	13a	36a	08a	69a	08b	23b	06b	39b
265	037	378	068	139	239	069	089	23a	07a	36b	6ab	3ab
266	037	378	068	139	239	079	36a	06a	08b	23b	69b	3ab
267	037	378	068	139	369	079	23a	06a	08a	69a	23b	3ab

268	037	378	068	139	069	23a	36a	07a	08a	23b	39b	6ab
269	037	378	078	139	369	069	23a	06a	08a	23b	6ab	3ab
270	037	378	078	239	069	13a	23a	06a	08b	36b	39b	6ab
271	037	367	238	239	069	13a	08a	08b	07b	67b	39b	3ab
272	037	367	238	069	13b	089	679	23a	08a	39a	07b	39b
273	037	367	068	139	079	23a	06a	08a	39a	69a	23b	38b
274	037	367	068	239	069	13b	38a	39a	08b	23b	07b	6ab
275	037	367	068	239	13a	089	23a	07a	06b	39b	38b	6ab
276	037	367	068	239	13a	06a	38a	08b	23b	07b	39b	69b
277	037	367	068	079	13b	389	23a	06a	39a	08b	23b	69b
278	037	367	078	139	069	23a	06a	08a	39a	23b	38b	6ab
279	037	367	078	239	069	13b	089	389	23a	06b	69b	3ab
280	037	067	238	239	13b	37a	089	39a	08b	36b	67b	09b
281	037	067	368	239	069	13a	23a	08a	08b	37b	39b	6ab
282	037	067	068	239	369	13a	23a	08a	37b	09b	38b	6ab
283	037	238	068	139	37a	08a	67a	39a	09a	36b	23b	07b
284	037	238	068	369	079	13b	089	679	23a	39a	37b	0ab
285	037	238	078	139	37a	06a	39a	09a	08b	36b	23b	67b
286	037	238	078	239	369	13a	08a	67a	06b	37b	09b	3ab
287	037	238	078	239	369	13a	09a	08b	06b	37b	67b	3ab
288	037	138	068	239	069	23a	08a	67a	39a	36b	37b	09b
289	037	368	068	239	079	13b	37a	089	23a	69a	06b	39b
290	037	368	068	239	37a	13a	08a	69a	23b	06b	07b	39b
291	037	368	078	239	069	13a	23a	08a	69a	06b	37b	39b
292	037	368	078	239	37a	13a	06a	08b	23b	06b	39b	6ab
293	037	368	078	239	13a	06a	39a	08b	23b	06b	37b	69b
294	037	068	078	239	369	13a	23a	09a	06b	37b	38b	6ab
295	037	068	078	369	13b	37a	389	23a	09a	23b	06b	6ab
296	136	237	067	039	239	07a	38a	67a	06b	37b	09b	3ab
297	136	237	067	039	679	389	23a	07a	09a	06b	37b	3ab
298	136	237	067	239	069	079	38a	67a	39a	03b	37b	0ab
299	136	237	067	239	069	679	03a	38a	37b	07b	39b	09b
300	136	038	067	239	079	37a	23a	67a	09a	06b	37b	39b
301	136	038	067	239	37a	07a	39a	23b	06b	37b	67b	09b
302	136	067	238	039	37a	679	23a	08a	09a	37b	07b	39b
303	136	067	238	039	37a	679	08a	39a	23b	37b	07b	0ab
304	136	067	238	239	079	37a	03a	08b	37b	67b	39b	09b
305	136	067	238	239	079	37a	08a	67a	39a	03b	37b	09b
306	136	067	238	239	079	37a	39a	09a	03b	08b	37b	67b
307	136	067	068	039	239	079	37a	23a	37b	38b	0ab	6ab
308	136	067	078	039	239	37a	38a	69a	23b	06b	37b	0ab
309	136	238	078	239	079	37a	03a	67a	06b	37b	39b	09b
310	036	137	378	239	079	08a	67a	39a	08b	23b	07b	3ab
311	036	137	078	239	37a	089	679	38a	23b	07b	39b	0ab
312	036	137	078	079	37a	389	23a	08a	67a	23b	39b	0ab

313	036 237 378 239 079 13a 679 06a 08b 07b 39b 3ab
314	036 237 067 139 079 23a 08a 67a 39a 09a 37b 38b
315	036 237 067 139 37a 679 389 08a 09a 23b 07b 3ab
316	036 237 067 139 37a 38a 08a 67a 09a 23b 07b 39b
317	036 237 067 239 079 13b 37a 389 08b 67b 09b 3ab
318	036 237 067 239 079 13a 679 389 08b 37b 0ab 3ab
319	036 237 067 239 13a 679 38a 09a 08b 37b 07b 39b
320	036 237 067 13b 37a 679 389 08a 39a 23b 07b 09b
321	036 237 067 37a 13a 679 389 08a 23b 07b 39b 0ab
322	036 237 067 13a 679 389 23a 07a 08a 37b 39b 09b
323	036 237 068 239 13a 679 07a 38a 37b 07b 39b 09b
324	036 237 078 239 079 13b 37a 389 67a 06b 09b 3ab
325	036 378 078 239 069 079 13b 23a 39a 08b 37b 6ab
326	036 067 138 239 37a 089 679 23a 08a 37b 39b 09b
327	036 238 078 239 37a 13a 679 08a 37b 07b 39b 09b
328	036 138 068 239 079 37a 23a 09a 08b 37b 67b 39b
329	036 138 078 239 069 37a 23a 08a 37b 67b 39b 09b
330	036 068 078 139 239 079 37a 23a 09a 37b 38b 6ab
331	036 068 078 139 239 079 37a 38a 69a 23b 37b 0ab
332	036 068 078 139 239 37a 38a 09a 23b 37b 07b 69b
333	036 068 078 239 079 13b 37a 389 23a 37b 09b 69b
334	236 137 378 239 079 089 03a 67a 08b 06b 39b 3ab
335	236 137 067 37a 089 679 389 23a 08a 09a 03b 39b
336	236 137 068 039 079 23a 08a 67a 39a 37b 38b 0ab
337	236 137 068 039 37a 679 389 08a 23b 07b 0ab 3ab
338	236 137 068 039 37a 38a 08a 67a 23b 07b 39b 0ab
339	236 137 068 039 679 389 23a 07a 08a 37b 09b 3ab
340	236 137 068 239 37a 089 07a 38a 03b 67b 39b 09b
341	236 137 068 079 37a 389 23a 08a 67a 09a 03b 39b
342	236 137 068 079 679 389 03a 39a 08b 23b 37b 0ab
343	236 137 068 37a 679 389 03a 08a 23b 07b 39b 09b
344	236 137 078 039 23a 08a 67a 39a 09a 06b 37b 38b
345	236 137 078 069 679 389 23a 08a 39a 09a 03b 37b
346	236 237 378 039 079 13b 06a 67a 39a 08b 06b 3ab
347	236 237 378 039 13a 06a 07a 08b 06b 67b 39b 3ab
348	236 237 067 139 069 679 389 03a 08a 37b 09b 3ab
349	236 237 067 37a 13a 389 08a 67a 09a 03b 06b 39b
350	236 237 068 139 039 079 37a 06a 67b 38b 0ab 3ab
351	236 237 068 139 039 07a 38a 67a 09a 06b 37b 3ab
352	236 237 068 139 069 079 679 389 03a 37b 0ab 3ab
353	236 237 068 139 069 679 03a 38a 09a 37b 07b 39b
354	236 237 068 139 079 37a 679 389 06a 09a 03b 3ab
355	236 237 068 139 079 37a 03a 06a 67b 39b 09b 38b
356	236 237 068 039 079 13b 37a 389 67a 06b 0ab 3ab
357	236 237 068 039 13a 07a 39a 06b 37b 67b 38b 0ab

358	236 237 078 139 069 03a 38a 67a 09a 06b 37b 39b
359	236 237 078 069 13b 37a 389 03a 06b 67b 39b 09b
360	236 378 067 239 069 13a 08a 39a 03b 08b 37b 6ab
361	236 378 067 239 13a 089 03a 08b 06b 37b 39b 69b
362	236 378 068 139 089 03a 07a 39a 23b 06b 37b 69b
363	236 378 068 039 13b 089 23a 07a 39a 06b 37b 69b
364	236 378 068 239 069 37a 13a 08a 03b 07b 39b 6ab
365	236 378 078 139 37a 089 03a 06a 23b 06b 39b 69b
366	236 378 078 039 13b 37a 089 23a 06a 06b 39b 69b
367	236 378 078 239 069 13b 37a 089 39a 03b 06b 6ab
368	236 038 067 139 37a 08a 67a 39a 09a 23b 06b 37b
369	236 038 068 139 239 079 37a 679 06a 37b 09b 3ab
370	236 038 068 139 239 079 37a 67a 09a 06b 37b 3ab
371	236 038 068 139 079 37a 679 06a 39a 23b 37b 0ab
372	236 038 068 239 37a 13a 07a 06b 37b 67b 39b 09b
373	236 038 078 139 37a 679 06a 39a 09a 23b 06b 37b
374	236 038 078 239 069 37a 13a 06b 37b 67b 39b 0ab
375	236 067 238 139 37a 089 679 03a 08a 37b 39b 09b
376	236 067 238 039 37a 13a 08a 08b 37b 67b 39b 0ab
377	236 067 068 139 039 37a 38a 08a 69a 23b 37b 0ab
378	236 067 068 039 239 13b 37a 38a 08b 37b 09b 6ab
379	236 238 068 139 039 079 37a 08a 67a 37b 0ab 3ab
380	236 238 068 139 039 37a 07a 08a 37b 67b 09b 3ab
381	236 238 068 139 079 37a 03a 08a 67a 37b 39b 09b
382	236 238 068 039 079 13b 37a 39a 08b 37b 67b 0ab
383	236 238 068 039 13b 37a 679 08a 39a 37b 07b 09b
384	236 238 078 039 13b 37a 089 679 39a 06b 37b 0ab
385	236 138 068 039 239 37a 089 679 06b 37b 0ab 3ab
386	236 138 068 039 239 37a 08a 67a 06b 37b 09b 3ab
387	236 138 068 239 37a 089 679 03a 06b 37b 39b 09b
388	137 237 067 039 389 23a 07a 09a 08b 36b 67b 3ab
389	137 237 067 239 079 089 679 36a 39a 03b 38b 0ab
390	137 237 067 239 089 679 03a 39a 36b 07b 09b 38b
391	137 237 067 369 079 389 03a 08b 23b 67b 0ab 3ab
392	137 237 067 369 38a 08a 67a 39a 09a 03b 23b 07b
393	137 237 067 079 389 03a 23a 67a 09a 08b 36b 39b
394	137 237 067 679 389 36a 07a 08a 39a 03b 23b 09b
395	137 237 067 389 03a 07a 39a 08b 36b 23b 67b 09b
396	137 237 368 039 069 23a 07a 08a 07b 67b 39b 3ab
397	137 237 368 039 07a 08a 67a 39a 23b 06b 07b 3ab
398	137 237 368 239 069 079 08a 67a 39a 03b 07b 3ab
399	137 237 068 039 679 389 23a 07a 09a 36b 07b 3ab
400	137 237 068 039 679 36a 07a 38a 23b 07b 39b 0ab
401	137 237 068 369 079 03a 39a 23b 07b 67b 38b 0ab
402	137 237 078 039 239 679 36a 09a 06b 07b 38b 3ab

403	137	237	078	039	369	079	23a	06a	67b	38b	0ab	3ab
404	137	237	078	039	679	389	36a	07a	23b	06b	0ab	3ab
405	137	237	078	039	23a	36a	07a	09a	06b	67b	39b	38b
406	137	237	078	039	36a	07a	39a	23b	06b	67b	38b	0ab
407	137	237	078	239	069	679	36a	38a	09a	03b	07b	39b
408	137	237	078	239	079	03a	36a	67a	06b	39b	09b	38b
409	137	237	078	239	079	36a	67a	39a	09a	03b	06b	38b
410	137	237	078	369	679	389	03a	09a	23b	06b	07b	3ab
411	137	237	078	369	03a	39a	09a	23b	06b	07b	67b	38b
412	137	237	078	069	679	389	03a	23a	09a	36b	07b	39b
413	137	378	067	039	239	36a	08a	08b	23b	07b	69b	3ab
414	137	378	067	039	23a	36a	07a	08a	08b	23b	39b	6ab
415	137	378	067	369	089	03a	23a	08a	23b	07b	39b	69b
416	137	378	068	039	239	089	23a	07a	36b	07b	69b	3ab
417	137	378	068	039	369	23a	07a	08a	23b	07b	6ab	3ab
418	137	378	078	039	23a	06a	07a	39a	08b	36b	23b	6ab
419	137	378	078	369	079	03a	23a	06a	08b	23b	39b	6ab
420	137	038	367	239	089	679	07a	39a	23b	06b	07b	3ab
421	137	038	067	239	079	37a	089	23a	67a	36b	39b	0ab
422	137	038	067	239	37a	08a	67a	39a	36b	23b	07b	09b
423	137	038	078	239	069	079	679	23a	36a	37b	39b	0ab
424	137	367	067	039	239	089	23a	08a	07b	38b	69b	3ab
425	137	367	067	039	239	38a	08a	69a	08b	23b	07b	3ab
426	137	367	068	039	239	07a	38a	08b	23b	07b	6ab	3ab
427	137	367	068	039	23a	07a	08a	39a	23b	07b	38b	6ab
428	137	367	068	239	079	03a	39a	08b	23b	07b	38b	6ab
429	137	367	078	039	239	079	23a	06a	08b	38b	6ab	3ab
430	137	367	078	039	23a	06a	07a	38a	08b	23b	39b	6ab
431	137	367	078	239	089	07a	38a	39a	69a	03b	23b	06b
432	137	067	368	239	079	089	03a	23a	08b	37b	39b	69b
433	137	067	078	039	239	37a	089	23a	36b	38b	0ab	6ab
434	137	067	078	239	369	37a	089	38a	03b	23b	0ab	6ab
435	137	238	078	039	239	079	679	36a	08b	37b	0ab	3ab
436	137	238	078	239	079	37a	089	679	36a	03b	39b	0ab
437	137	238	078	239	079	37a	089	67a	39a	03b	36b	0ab
438	137	238	078	369	079	089	679	03a	23a	37b	39b	0ab
439	137	238	078	369	679	03a	08a	39a	23b	37b	07b	09b
440	137	368	078	039	239	069	23a	08a	37b	07b	69b	3ab
441	137	368	078	039	239	079	23a	06a	08b	37b	69b	3ab
442	137	368	078	039	239	089	23a	07a	06b	37b	69b	3ab
443	137	068	078	239	369	079	03a	23a	37b	09b	38b	6ab
444	137	068	078	239	079	37a	389	23a	09a	69a	03b	36b
445	137	068	078	239	37a	389	03a	36b	23b	07b	09b	69b
446	237	378	067	139	069	03a	08a	39a	69a	36b	23b	07b
447	237	378	067	139	079	03a	36a	06a	08b	23b	39b	69b

448	237	378	067	139	089	03a	07a	39a	69a	36b	23b	06b
449	237	378	067	039	13b	089	23a	07a	39a	69a	36b	06b
450	237	378	067	239	069	13a	03a	08b	36b	07b	39b	6ab
451	237	378	068	139	239	069	079	03a	69a	36b	07b	3ab
452	237	378	078	139	039	069	23a	06a	07a	36b	6ab	3ab
453	237	378	078	139	039	36a	06a	07a	23b	06b	69b	3ab
454	237	378	078	139	069	03a	36a	07a	23b	06b	39b	6ab
455	237	378	078	039	239	069	13b	36a	06b	07b	69b	3ab
456	237	038	367	239	069	079	13b	67a	39a	06b	07b	3ab
457	237	038	367	239	069	13a	07a	06b	07b	67b	39b	3ab
458	237	038	067	239	13a	07a	67a	39a	36b	06b	37b	09b
459	237	038	067	369	13a	07a	39a	23b	06b	37b	67b	0ab
460	237	367	067	139	089	389	03a	07a	23b	06b	69b	3ab
461	237	367	067	039	239	069	13b	38a	08b	07b	6ab	3ab
462	237	367	067	039	079	13b	389	23a	06a	08b	69b	3ab
463	237	367	067	039	13b	089	389	23a	07a	06b	69b	3ab
464	237	367	067	039	13b	089	23a	07a	38a	69a	06b	39b
465	237	367	067	239	13a	089	07a	38a	69a	03b	06b	39b
466	237	367	067	069	079	13b	389	03a	23a	08b	39b	6ab
467	237	367	067	069	079	13b	389	23a	08a	39a	69a	03b
468	237	367	067	069	13a	389	23a	07a	08a	03b	39b	6ab
469	237	367	067	079	13b	389	03a	39a	69a	08b	23b	06b
470	237	367	238	079	13a	06a	07a	39a	03b	08b	67b	39b
471	237	367	138	069	079	23a	06a	08a	67a	39a	03b	39b
472	237	367	138	069	03a	08a	67a	39a	23b	06b	07b	39b
473	237	367	068	039	069	13b	23a	07a	38a	07b	39b	6ab
474	237	367	068	239	079	13a	06a	39a	03b	07b	38b	69b
475	237	367	068	069	079	13b	389	03a	23a	07b	39b	69b
476	237	367	078	139	039	06a	07a	38a	69a	23b	06b	3ab
477	237	367	078	139	069	389	03a	07a	23b	06b	6ab	3ab
478	237	367	078	139	069	06a	07a	38a	39a	03b	23b	6ab
479	237	367	078	039	069	13b	389	23a	07a	06b	6ab	3ab
480	237	367	078	039	069	13b	23a	06a	38a	07b	39b	69b
481	237	367	078	039	13a	23a	06a	07a	06b	39b	38b	6ab
482	237	067	238	139	039	37a	08a	67a	09a	36b	07b	3ab
483	237	067	238	139	369	079	089	679	03a	37b	0ab	3ab
484	237	067	238	139	369	079	03a	08a	67a	37b	09b	3ab
485	237	067	238	139	37a	089	07a	39a	09a	03b	36b	67b
486	237	067	238	369	079	13b	37a	089	39a	03b	67b	0ab
487	237	067	138	369	089	679	03a	23a	09a	06b	37b	39b
488	237	067	368	139	039	37a	06a	08a	23b	07b	69b	3ab
489	237	067	368	139	079	03a	06a	39a	08b	23b	37b	69b
490	237	067	368	139	089	03a	07a	39a	23b	06b	37b	69b
491	237	067	368	039	13b	37a	08a	39a	69a	23b	06b	07b
492	237	067	368	239	13a	089	07a	39a	03b	06b	37b	69b

493	237	067	068	139	369	079	03a	38a	69a	23b	37b	0ab
494	237	067	078	039	369	069	13b	23a	38a	37b	0ab	6ab
495	237	067	078	039	369	13b	23a	06a	38a	37b	09b	69b
496	237	067	078	369	13b	37a	389	03a	23b	06b	09b	6ab
497	237	067	078	369	13a	389	03a	23b	06b	37b	0ab	69b
498	237	238	068	139	039	37a	07a	09a	36b	07b	67b	3ab
499	237	238	068	139	079	37a	03a	67a	09a	36b	07b	39b
500	237	238	068	369	079	13b	37a	39a	09a	03b	07b	67b
501	237	238	078	369	079	13a	03a	67a	06b	37b	39b	0ab
502	237	238	078	369	13a	679	03a	09a	06b	37b	07b	39b
503	237	138	068	039	369	079	679	23a	06a	37b	0ab	3ab
504	237	138	068	039	079	37a	23a	06a	67a	36b	39b	0ab
505	237	138	068	039	679	23a	36a	07a	09a	06b	37b	39b
506	237	138	068	239	069	37a	03a	36b	07b	67b	39b	09b
507	237	138	068	369	079	03a	67a	39a	23b	06b	37b	0ab
508	237	138	068	369	37a	03a	09a	23b	06b	07b	67b	39b
509	237	138	068	369	679	03a	39a	09a	23b	06b	37b	07b
510	237	138	078	039	369	37a	06a	23b	06b	67b	0ab	3ab
511	237	138	078	369	37a	03a	06a	23b	06b	67b	39b	09b
512	237	368	068	139	039	079	23a	06a	07a	37b	69b	3ab
513	237	368	068	039	079	13b	23a	07a	39a	69a	06b	37b
514	237	368	078	139	239	069	079	37a	06a	03b	6ab	3ab
515	237	368	078	139	239	069	079	03a	69a	06b	37b	3ab
516	237	368	078	139	069	03a	07a	39a	23b	06b	37b	6ab
517	237	368	078	039	239	069	13b	37a	06b	07b	6ab	3ab
518	237	368	078	039	069	13b	23a	07a	39a	06b	37b	6ab
519	038	367	067	139	239	069	37a	08a	23b	07b	6ab	3ab
520	038	367	067	139	079	37a	23a	06a	08a	69a	23b	39b
521	038	367	067	239	079	13b	37a	089	23a	69a	06b	39b
522	038	367	067	239	37a	13a	08a	69a	23b	06b	07b	39b
523	038	367	068	239	079	13a	23a	07a	69a	06b	37b	39b
524	038	067	068	239	369	079	13b	37a	23a	37b	09b	6ab
525	367	067	238	139	079	03a	08a	39a	69a	08b	23b	37b
526	367	067	238	039	13a	23a	07a	08a	08b	37b	39b	6ab
527	367	067	238	239	079	13a	08a	39a	69a	03b	08b	37b
528	367	238	068	139	079	37a	03a	08a	69a	23b	07b	39b
529	367	238	078	139	039	37a	06a	08a	23b	07b	69b	3ab
530	367	238	078	139	069	079	03a	23a	08a	69a	37b	39b
531	367	238	078	039	239	13a	07a	08b	06b	37b	6ab	3ab
532	367	238	078	039	13b	37a	08a	39a	69a	23b	06b	07b
533	367	138	068	239	069	37a	08a	39a	03b	23b	07b	6ab
534	367	138	068	239	079	37a	089	23a	06a	03b	39b	69b
535	367	138	078	039	239	37a	06a	08b	23b	06b	6ab	3ab
536	067	238	078	039	239	37a	13a	08b	36b	37b	0ab	6ab
537	067	238	078	039	369	13b	37a	23a	08a	37b	09b	6ab

538	067 238 078 239 369 37a 13a 08a 03b 37b 09b 6ab
539	238 068 078 139 369 079 37a 03a 23b 37b 0ab 6ab
540	138 068 078 039 239 37a 23a 06a 36b 37b 09b 69b

B.3.4 Starter Blocks for TTS(12) With Starting Configuration D

Design #	Starter Blocks
1	379 037 136 067 13b 389 03a 23a 67a 23b 09b 69b
2	379 037 136 238 239 13a 679 67a 09a 03b 07b 3ab
3	379 037 136 238 239 13a 07a 67a 03b 67b 09b 3ab
4	379 037 136 238 13a 23a 07a 67a 09a 03b 67b 39b
5	379 037 136 138 679 03a 23a 06a 67a 23b 39b 09b
6	379 037 036 138 13b 089 679 23a 67a 39a 23b 0ab
7	379 037 036 078 13b 13a 389 23a 67a 23b 0ab 69b
8	379 037 236 068 039 13b 13a 23a 67a 38b 0ab 6ab
9	379 037 038 067 239 13b 13a 23a 67a 69a 36b 09b
10	379 037 038 067 369 13b 13a 23a 69a 23b 67b 0ab
11	379 037 067 368 139 13b 089 03a 23a 69a 23b 69b
12	379 037 238 078 239 13b 13a 679 36a 69a 03b 09b
13	379 136 067 138 239 37a 03a 67a 03b 23b 09b 6ab
14	379 036 237 138 13b 679 03a 67a 39a 23b 07b 09b
15	379 036 237 138 13a 23a 07a 67a 09a 03b 67b 39b
16	379 036 138 078 239 37a 13a 69a 03b 23b 67b 0ab
17	379 236 137 038 139 03a 07a 67a 23b 67b 09b 3ab
18	379 236 137 078 239 13a 03a 67a 03b 09b 38b 6ab
19	379 236 137 078 239 13a 03a 69a 03b 67b 09b 38b
20	379 236 237 067 13b 13a 679 389 03a 69a 03b 09b
21	379 236 237 138 139 069 03a 67a 09a 03b 67b 3ab
22	379 236 237 138 13a 679 03a 06a 67a 03b 39b 09b
23	379 236 238 078 039 13b 37a 13a 67a 03b 0ab 6ab
24	379 236 138 068 039 13b 679 03a 23a 69a 37b 09b
25	379 137 138 078 369 03a 23a 67a 09a 03b 23b 6ab
26	379 367 067 138 039 13b 03a 23a 69a 08b 23b 6ab
27	379 367 238 078 139 13a 03a 07a 69a 03b 23b 6ab
28	379 367 138 068 239 13a 03a 69a 03b 23b 07b 69b
29	037 136 137 039 23a 07a 38a 67a 23b 67b 39b 0ab
30	037 136 378 139 239 069 679 03a 23b 07b 6ab 3ab
31	037 136 378 139 239 079 03a 67a 69a 23b 06b 3ab
32	037 136 378 039 239 079 13b 23a 67a 69a 06b 3ab
33	037 036 237 139 239 079 13a 67a 67b 38b 0ab 3ab
34	037 036 078 139 239 13a 679 38a 69a 23b 37b 0ab
35	037 036 078 139 13b 37a 679 389 23a 09a 23b 6ab
36	037 236 137 139 089 679 03a 38a 67a 23b 39b 0ab
37	037 236 137 239 13a 679 389 08a 67a 03b 09b 3ab

38	037	236	378	039	13b	13a	23a	08a	67a	69a	06b	39b
39	037	236	238	039	13b	13a	679	39a	08b	37b	67b	0ab
40	037	236	068	139	13a	679	389	03a	23b	37b	0ab	69b
41	037	236	068	239	13b	37a	13a	389	03b	67b	09b	69b
42	037	137	038	139	239	679	36a	07a	23b	67b	09b	3ab
43	037	137	038	239	079	13b	679	23a	36a	67a	39b	09b
44	037	137	138	039	239	089	679	23a	67a	36b	0ab	3ab
45	037	137	138	039	679	23a	36a	08a	67a	23b	39b	0ab
46	037	137	138	369	23a	08a	67a	39a	09a	03b	23b	67b
47	037	137	368	139	079	03a	23a	08a	67a	69a	23b	39b
48	037	237	067	139	239	13a	03a	67a	36b	09b	38b	6ab
49	037	237	067	139	369	13b	03a	38a	67a	23b	09b	6ab
50	037	237	067	139	369	13b	03a	38a	69a	23b	67b	09b
51	037	237	067	369	13b	13a	679	389	23a	09a	03b	6ab
52	037	237	238	139	13a	03a	07a	67a	36b	67b	39b	09b
53	037	237	238	369	13b	13a	679	03a	07b	67b	39b	09b
54	037	237	138	139	239	069	03a	67a	36b	67b	09b	3ab
55	037	237	138	239	13a	679	06a	67a	39a	03b	36b	09b
56	037	237	138	369	13a	679	06a	39a	03b	23b	67b	0ab
57	037	237	368	139	239	079	13a	679	06a	69a	03b	3ab
58	037	237	368	139	13a	679	06a	07a	39a	03b	23b	6ab
59	037	378	067	139	039	13b	23a	36a	08a	69a	23b	69b
60	037	378	068	139	239	13a	36a	07a	03b	23b	69b	6ab
61	037	367	238	139	079	13b	089	679	03a	23a	69a	39b
62	037	367	238	239	079	13b	13a	67a	39a	03b	08b	6ab
63	037	367	238	239	079	13b	13a	39a	69a	03b	08b	67b
64	037	367	138	139	239	089	679	03a	69a	23b	06b	3ab
65	037	367	138	139	239	089	03a	67a	23b	06b	69b	3ab
66	037	367	138	239	13a	08a	67a	39a	69a	03b	23b	06b
67	037	138	068	039	239	13a	679	23a	69a	36b	37b	0ab
68	037	138	068	239	369	13a	23a	09a	03b	37b	67b	69b
69	036	237	378	239	079	13b	13a	679	39a	69a	03b	07b
70	036	237	378	239	079	13b	13a	67a	39a	03b	07b	69b
71	036	237	138	239	079	37a	13a	67a	03b	67b	39b	0ab
72	236	137	237	139	039	079	03a	67a	67b	38b	0ab	3ab
73	236	137	237	139	079	679	389	03a	67a	09a	03b	3ab
74	236	137	378	039	079	13b	03a	23a	67a	08b	39b	6ab
75	236	137	378	039	079	13b	03a	23a	69a	08b	67b	39b
76	236	137	378	239	13a	089	679	03a	03b	07b	39b	69b
77	236	137	038	239	079	13a	679	67a	39a	03b	37b	0ab
78	236	137	138	039	239	679	03a	08b	37b	67b	09b	3ab
79	236	137	078	039	13b	679	389	03a	23a	37b	09b	69b
80	236	237	378	139	039	069	13b	679	03a	07b	6ab	3ab
81	236	237	067	139	039	13b	37a	03a	67b	09b	38b	6ab
82	236	237	067	039	13b	37a	13a	679	389	03b	0ab	6ab

83	137 237 368 239 079 13a 679 03a 69a 03b 07b 39b
84	137 038 367 039 13b 679 23a 07a 39a 23b 07b 6ab
85	237 367 067 039 13b 13a 389 23a 07a 03b 69b 6ab
86	237 367 138 039 13b 679 03a 39a 69a 23b 06b 07b
87	237 367 138 039 13a 679 23a 06a 07a 03b 39b 6ab
88	237 067 368 039 239 13b 37a 13a 03b 07b 69b 6ab

B.3.5 Starter Blocks for TTS(12) With Starting Configuration E

Design #	Starter Blocks
1	379 037 136 138 368 03a 23a 06a 67a 23b 09b 69b
2	379 036 236 237 138 13a 03a 67a 67b 09b 38b 0ab
3	379 036 237 138 368 13a 23a 07a 67a 09a 03b 6ab
4	379 036 038 367 138 13a 23a 07a 69a 23b 67b 0ab
5	379 236 038 367 138 069 13b 03a 23a 69a 67b 09b
6	379 236 038 367 138 13a 23a 06a 67a 09a 03b 69b
7	037 136 036 378 239 13a 23a 09a 07b 67b 38b 69b
8	037 136 036 138 239 37a 38a 67a 23b 67b 09b 0ab
9	037 136 236 378 239 069 13a 38a 67a 03b 0ab 6ab
10	037 136 236 378 13a 679 389 23a 06a 09a 03b 69b
11	037 136 236 238 37a 13a 679 389 09a 03b 67b 0ab
12	037 136 137 368 079 679 389 03a 23a 69a 23b 0ab
13	037 136 378 238 139 03a 36a 07a 23b 67b 09b 69b
14	037 136 378 238 039 079 13b 679 23a 36a 69a 0ab
15	037 136 378 238 239 13a 03a 67a 36b 07b 09b 69b
16	037 136 378 238 239 13a 36a 07a 03b 67b 09b 69b
17	037 136 367 238 239 079 13a 38a 67a 69a 03b 0ab
18	037 136 238 368 139 079 679 03a 23a 09a 69a 37b
19	037 136 238 368 039 13b 37a 23a 09a 07b 67b 69b
20	037 136 138 368 239 37a 03a 69a 23b 06b 67b 09b
21	037 036 236 237 139 13a 38a 67a 09a 67b 38b 0ab
22	037 036 236 378 239 13b 13a 38a 67a 08b 09b 6ab
23	037 036 236 378 13b 13a 679 389 23a 08a 69a 09b
24	037 036 236 378 13b 13a 679 389 23a 09a 08b 6ab
25	037 036 236 378 13b 13a 389 23a 09a 08b 67b 69b
26	037 036 378 138 239 13a 23a 08a 67a 69a 36b 09b
27	037 036 378 138 239 13a 23a 67a 09a 08b 36b 6ab
28	037 036 367 138 239 13b 089 23a 38a 67a 09b 69b
29	037 236 137 368 139 03a 38a 08a 67a 69a 23b 09b
30	037 236 137 368 039 13b 23a 38a 09a 69a 08b 67b
31	037 236 137 368 039 13a 23a 08a 69a 67b 38b 0ab
32	037 236 137 368 239 13a 38a 08a 67a 69a 03b 09b
33	037 236 237 368 139 13a 679 06a 38a 09a 69a 03b
34	037 236 378 038 139 069 13b 23a 36a 09a 67b 69b

35	037	236	378	038	369	13b	13a	23a	67a	09a	06b	6ab
36	037	236	378	238	139	13a	36a	08a	67a	09a	03b	69b
37	037	236	378	238	369	13b	13a	03a	08b	67b	09b	6ab
38	037	236	378	368	139	069	13b	089	03a	23a	69b	6ab
39	037	236	038	368	139	069	13b	37a	23a	09a	67b	6ab
40	037	236	038	368	139	37a	13a	06a	23b	67b	0ab	69b
41	037	236	367	238	039	13b	13a	38a	69a	08b	67b	0ab
42	037	236	238	368	139	13a	03a	09a	08b	37b	67b	69b
43	037	236	238	368	039	13b	37a	13a	08b	67b	0ab	6ab
44	037	137	038	368	139	679	23a	36a	07a	09a	23b	6ab
45	037	137	038	368	239	13a	679	23a	09a	69a	36b	07b
46	037	237	238	368	139	13a	679	03a	09a	69a	36b	07b
47	037	237	138	368	239	13a	03a	67a	69a	36b	06b	09b
48	037	038	367	238	369	13b	13a	23a	07a	67b	09b	6ab
49	037	038	367	138	369	13a	23a	06a	23b	67b	0ab	69b
50	037	367	238	138	039	13b	23a	36a	08a	67a	09b	69b
51	037	367	238	138	239	13a	03a	69a	08b	36b	67b	09b
52	037	367	238	138	239	13a	36a	08a	67a	03b	09b	69b
53	136	236	137	378	039	03a	38a	67a	23b	07b	0ab	69b
54	136	236	137	378	039	23a	07a	38a	67a	09a	03b	6ab
55	136	236	378	238	139	079	37a	03a	09a	69a	03b	67b
56	136	236	378	238	039	37a	13a	07a	03b	67b	0ab	6ab
57	136	236	378	138	239	069	37a	03a	03b	67b	09b	6ab
58	136	367	238	138	039	079	679	03a	23a	69a	37b	0ab
59	136	367	238	138	239	079	37a	03a	67a	03b	09b	6ab
60	036	236	378	038	139	13a	679	23a	07a	09a	37b	6ab
61	036	237	378	138	139	03a	36a	07a	23b	67b	09b	69b
62	036	237	378	138	039	079	13b	23a	36a	67a	0ab	69b
63	036	237	378	138	369	13a	23a	07a	09a	03b	67b	6ab
64	036	237	367	138	079	13b	389	03a	23a	67a	09b	69b
65	036	237	138	368	239	37a	13a	679	09a	03b	07b	6ab
66	036	237	138	368	239	37a	13a	07a	03b	67b	09b	6ab
67	036	237	138	368	239	37a	13a	09a	03b	07b	67b	69b
68	036	378	367	138	239	13a	089	23a	07a	03b	69b	6ab
69	036	038	367	138	239	37a	13a	679	23b	07b	0ab	6ab
70	236	137	237	368	039	13a	07a	38a	69a	03b	67b	0ab
71	236	137	237	368	079	13a	389	03a	67a	03b	0ab	69b
72	236	137	378	038	139	03a	36a	07a	23b	67b	09b	69b
73	236	137	378	038	039	13b	679	23a	36a	09a	07b	69b
74	236	137	378	138	039	089	03a	23a	67a	36b	0ab	69b
75	236	137	378	138	039	03a	23a	08a	67a	69a	36b	09b
76	236	137	378	368	039	13a	23a	07a	08a	69a	03b	6ab
77	236	137	038	367	139	079	03a	23a	67a	09a	38b	6ab
78	236	137	038	367	139	079	03a	38a	67a	69a	23b	0ab
79	236	137	038	367	239	13a	38a	67a	09a	03b	07b	69b

80	236	137	038	367	079	13b	389	03a	23a	67a	09b	69b
81	236	137	038	368	039	13b	37a	23a	09a	07b	67b	69b
82	236	137	038	368	239	079	37a	13a	69a	03b	67b	0ab
83	236	137	367	138	039	03a	23a	08a	69a	67b	09b	38b
84	236	237	378	138	369	13a	679	03a	06a	03b	09b	6ab
85	236	237	378	138	369	13a	03a	67a	09a	03b	06b	6ab
86	236	237	138	368	039	13a	03a	67a	06b	37b	0ab	69b
87	236	038	367	238	139	37a	13a	07a	09a	03b	67b	6ab
88	236	038	367	138	139	069	679	03a	23a	09a	37b	6ab
89	236	038	367	138	139	069	03a	23a	09a	37b	67b	69b
90	236	038	367	138	139	37a	03a	09a	69a	23b	06b	67b
91	236	038	367	138	239	37a	13a	09a	69a	03b	06b	67b
92	236	367	238	138	039	13a	03a	08b	37b	67b	0ab	69b
93	137	237	138	368	239	079	03a	36a	67a	03b	09b	69b
94	137	038	367	368	139	079	03a	23a	07a	69a	23b	6ab
95	137	367	138	368	039	03a	23a	08a	69a	23b	07b	69b

APPENDIX C

List of Possible Exceptions for W

This appendix gives a list of elements in E, the set of possible exceptions for W. We use the notation $x.y$ to represent the y numbers congruent to 0 or 1 (mod 3) immediately following and including x.

12	15	18	22	27.2	33.3	39.3	45.3	51.7	63.3	69.3	75.3
81.11	99.3	105.3	111.7	123.3	129.7	141.7	153.3	159.3	165.3	171.7	183.7
195.3	201.5	210	213.7	225.3	231.7	243.3	249.5	258.9	274.2	279.3	285
288	291.6	301.2	306	309.2	315	318.5	327.3	333.2	339.3	345.3	351.7
363.3	370.2	375	381.6	391.4	402	405.3	411.7	423.7	435.2	442.6	453.2
459.3	468.5	477.3	483	486	489.5	498	501.2	505.12	526.2	531.7	543.3
549.2	555	558	561.6	573.3	579.7	591	594.4	603.3	609	612	615
618	622	627.2	634.2	639.3	645	648.2	652.6	663.7	675.9	690	693.3
699.3	705.3	711	714	718	723.2	729.3	735	738	741.2	747.2	753.3
759	762	772	778.4	786	789.3	795.11	813.3	819.3	825.3	831.7	843.7
855.3	861.3	867.3	873.3	879.3	885.2	891.2	897.3	906	910	915	921.3
927.2	933.3	939.3	945	948	952	957.2	963.3	969.5	978.9	993.3	999.5
1008	1011.7	1023.3	1030	1035.2	1041.3	1047	1050	1054	1059.2	1065.3	1071
1074	1077.7	1089.3	1095	1098.8	1113.3	1119.3	1125	1128	1132	1137.2	1143.3
1149.3	1155.3	1161.5	1170	1173.2	1179.2	1185	1188	1191	1194	1198	1203.2
1209.3	1215.3	1221.3	1227.3	1233.3	1239.3	1245.3	1251.7	1263	1266.5	1275.3	1282.5
1293.3	1299.3	1305	1308	1312	1317.2	1323.3	1329.3	1335.3	1341.19	1371.7	1383.3
1389.7	1401.4	1408.6	1419.3	1425.3	1431.11	1449.3	1455	1458	1461.2	1467.2	1473.3
1479	1482	1492	1497.2	1503.3	1509.3	1515.3	1521.3	1527.3	1534.6	1545	1548

1551	1554.9	1569.3	1575.3	1581.7	1593.3	1599.6	1611.2	1617.3	1623	1626	1630
1635	1641.3	1647	1650	1653.3	1660.2	1665.3	1671.3	1677.3	1683.7	1695.3	1701.15
1725.11	1743	1746	1749.2	1755.2	1761.3	1767	1770	1774	1779.2	1785.3	1791.3
1797.3	1803.3	1809.3	1815	1818	1821.3	1827.2	1833.3	1839.3	1845	1848	1852
1857.2	1863.3	1869.3	1875.3	1881.7	1893.2	1899.2	1905	1908	1911	1914	1918
1923.2	1929.3	1935	1938	1941.3	1947.3	1953.3	1959.3	1965.3	1971.3	1977.7	1989.3
1995.3	2001.6	2013.3	2019.3	2025	2028	2038	2043.3	2049.3	2055.3	2061.7	2073.3
2079.3	2085	2088	2091.11	2109.3	2115.3	2121.7	2133.3	2139.3	2145.11	2164.10	2181.6
2193.3	2199.3	2205	2208	2212	2217.2	2223	2226	2229.3	2235.3	2241.7	2253.3
2259.3	2265	2268	2271.7	2283.3	2290.2	2295.3	2301.7	2313.3	2319.6	2331.2	2337.3
2343	2346	2350	2355	2361.3	2367.2	2373.3	2379.3	2385	2388	2392	2397.2
2403.3	2409.3	2415	2418	2421.3	2427.3	2433.3	2439.3	2445	2448	2451.11	2469.2
2475.2	2481.3	2487	2490	2494	2499.2	2505.3	2511	2514	2517.3	2523.3	2529.3
2535	2538	2544	2547.3	2553.3	2559.7	2571	2574	2577.7	2589.3	2595.9	2610.4
2619.2	2625.3	2631	2634	2638	2643.2	2650.2	2655.3	2661.3	2668.2	2673.3	2679.3
2685.3	2691.3	2697.3	2703	2706	2709.3	2715	2718	2721	2724	2727.3	2733.3
2739.3	2745.3	2751	2754.9	2769.15	2793.3	2799.3	2805.3	2811	2814.9	2829.3	2835.11
2853.3	2859.7	2871	2874.5	2883.3	2889.5	2898	2901	2907.2	2913.3	2922	2932
2937.2	2943.3	2949.3	2955.3	2961.3	2967.3	2973.3	2979.3	2985	2988	2991	2994.5
3003.3	3009.3	3015.3	3021	3024	3027.7	3039.6	3051.2	3057.3	3063	3066	3070
3075.2	3081.3	3087	3093.3	3099.3	3105	3108	3112	3117.2	3123.3	3129.3	3135.3
3141.3	3147.3	3153.3	3159.3	3165	3168	3174	3177.3	3183	3186	3189.3	3195.5
3204.5	3213.3	3219.7	3231	3234.9	3249.3	3255.3	3261	3264.4	3273.3	3279.3	3285
3288	3292	3298	3303.3	3309.3	3315.3	3321.5	3330	3333.2	3339.2	3345	3348
3351	3354	3358	3363.2	3369.3	3375	3378	3381	3384	3387.3	3393.3	3399.3
3405.3	3411.3	3417.3	3424.2	3429.3	3435	3438	3441	3444	3447.3	3453.3	3459.3
3465.3	3471.3	3477.3	3483.3	3489.7	3501.7	3513.3	3519.3	3525.3	3531.4	3538.2	3543.3
3550.6	3561	3564.3	3570	3573.3	3579.3	3585.3	3591.7	3603.3	3609.3	3615	3618.4
3627.2	3633.3	3639	3642	3652	3657.2	3663	3666	3669.3	3675.3	3681.3	3687.3
3693.3	3699.3	3705	3708	3711	3714.3	3720	3723	3726	3729.3	3735.3	3741
3744	3747.3	3753.3	3759	3762	3765.2	3771.2	3777.3	3783	3786	3790	3795.2
3804	3807.2	3813.3	3819.3	3825	3828	3832	3837.2	3843.3	3849.3	3855	3858
3861.3	3867.3	3873.3	3879.3	3885.2	3891	3894	3897.2	3903.3	3909.2	3915.2	3921.3
3927	3930	3934	3939.2	3945.3	3951	3954	3957.3	3963.3	3969.3	3975.3	3981.3
3987.3	3993.3	3999.3	4005.3	4011.3	4017.3	4023.3	4029.3	4035.3	4041.2	4047.3	4054
4059.2	4065	4068	4071	4074	4078	4083.2	4089.3	4095	4098	4101	4104
4107	4110	4113.3	4119.3	4125.3	4131.3	4137.3	4143.3	4149.3	4155	4158	4164
4167	4173.3	4180.2	4185	4188	4192	4197.2	4203.3	4209.3	4215.3	4221.3	4227.3
4233.3	4239.3	4245	4248	4251.7	4263.3	4269.3	4275.3	4281.7	4293.3	4299.3	4305.3
4311.11	4329.5	4338	4341.2	4347.2	4354.2	4359	4362	4366	4371.2	4377.3	4383.3

4389.2	4395.3	4401.3	4407.3	4413.3	4419.3	4425.3	4434.5	4443.3	4449.3	4455.3	4461.3
4467.3	4473.3	4479.3	4485.3	4491.3	4497.3	4503.3	4509.3	4515.3	4521.3	4527	4533.3
4539.3	4545	4548	4552	4557.2	4563.3	4569.3	4575.3	4581.7	4593.3	4599.3	4608
4611.7	4623	4626	4629.2	4635.2	4641.3	4647	4650	4654	4659.2	4665.3	4671.3
4677.3	4684.2	4689.3	4695	4698	4701.3	4707.2	4713.3	4719.3	4725	4728	4732
4737.2	4743.3	4749.3	4755.3	4761.3	4767	4770	4773.2	4779.2	4786.2	4791	4794
4798	4803.2	4810.2	4815.3	4821.3	4827.3	4833.3	4839.3	4845.3	4851.3	4857.3	4863
4866	4869.3	4875	4878	4881.3	4887	4893.3	4899	4905	4908	4917	4923.3
4929.3	4935.3	4941.3	4947.3	4953.3	4959.3	4965	4968	4971.3	4978.2	4983.3	4989.3
4995.3	5001.5	5010	5013.3	5019.3	5025.3	5031.5	5040	5043.3	5049.3	5055	5062
5067.2	5073.3	5079	5082	5092	5097	5103.3	5109.3	5115.3	5122.2	5127	5130
5133.3	5139.3	5145	5148	5151.7	5163.3	5169.3	5175.3	5181.3	5188.2	5193.3	5199
5202	5205.3	5211.3	5217.3	5223.3	5229.3	5235	5238	5241.3	5247	5250	5253.3
5259.3	5265.5	5274	5277	5280	5283.3	5289.3	5295	5298.13	5319.3	5325.3	5331.7
5343.3	5349.2	5355.2	5361.3	5367	5370	5374	5379.2	5385.3	5391	5394	5397
5400	5403.3	5409.3	5415	5418	5421	5424	5427.2	5433	5436	5439.3	5445
5448	5457	5463.2	5469.3	5475.3	5481.3	5487	5493.2	5499.2	5505	5508	5511
5514	5523.2	5529.3	5535	5538	5541.3	5547.3	5553.3	5559	5562	5565.3	5571.3
5577	5580	5583	5586	5589.3	5595	5598	5601	5604	5607	5610	5613.3
5619.3	5625.3	5631	5634	5637	5640	5643.3	5649.3	5655.3	5661.2	5667.3	5673.3
5679	5688	5697.2	5703	5706	5709.7	5722.2	5727.4	5734.2	5739.3	5745.3	5751
5754.3	5760.5	5769.3	5775	5778	5781	5787.2	5793.3	5799	5802	5812	5823.3
5829.3	5835.3	5841.3	5847.3	5854.2	5859.3	5865.3	5871	5874.5	5883.3	5890.2	5895.3
5901	5907.2	5913.2	5919	5922	5925.2	5931.3	5937	5940	5944.2	5949.3	5955
5958	5961.3	5967	5973.3	5979.3	5985	5988	5997	6003	6009.3	6015.3	6021.3
6027.3	6034.2	6039	6042	6048	6051	6054	6057	6060	6063	6066	6070.2
6075	6078	6084	6087	6090	6093.3	6099.3	6105.3	6111	6114	6117	6120
6123.3	6129.3	6135	6138	6141	6147.2	6153.3	6159	6162	6166	6171.2	6177
6180	6183.3	6189	6192	6195.3	6201.3	6207.3	6213.3	6219	6222	6226.2	6231
6234	6237	6240	6243.3	6249.3	6255	6258	6261	6264	6267.3	6273.3	6279
6282	6285.3	6291	6294	6297.2	6303.3	6309.3	6315	6318	6324	6327	6333.3
6339.3	6345	6348	6357	6363.3	6369	6372	6375	6378	6381.2	6387.3	6393.3
6399	6402	6405.3	6411.3	6417.3	6423	6426	6429	6432	6435.3	6441	6444
6450	6453.3	6459	6462	6465.3	6471.3	6477	6480	6483.3	6489.3	6495	6498
6501.2	6507	6513.3	6519	6522	6526	6531	6537	6543	6546	6549	6555.3
6561.3	6567.3	6574.2	6580.2	6585	6588	6591	6594	6597.3	6603.3	6609	6612
6615	6618	6621	6624	6627.3	6633.2	6639	6642	6645.2	6651.2	6657	6660
6663	6666	6675.2	6681	6684	6687	6693.3	6699	6702	6705	6708	6717
6724.2	6729.3	6735	6738	6741.3	6747.3	6753.3	6759	6762	6765	6768	6771
6774	6777.3	6783.3	6789	6795.2	6801.3	6807	6810	6814	6819	6826.2	6831

6834	6837	6840	6843.3	6849	6852	6855	6858	6861.3	6867	6873.3	6879
6882	6885	6888	6892	6903	6906	6909	6912	6915.3	6922.2	6927	6930
6933.2	6939.2	6945	6954	6963.2	6969.3	6975	6978	6981	6984	6987.2	6994.2
6999	7002	7005.3	7012.2	7017	7020	7023	7026	7029	7032	7035.3	7044
7047	7050	7053.2	7059	7062	7066.2	7071	7074	7080	7083.3	7089	7092
7095.3	7101.3	7107.3	7113.3	7119	7122	7125	7128	7131.3	7137.3	7143	7146
7149	7152	7155.3	7161	7167	7170	7173	7179.2	7186.2	7191	7194	7197
7204.2	7212	7215	7218	7221	7224	7227	7233.3	7239	7242	7245	7248
7252	7257	7263.3	7272	7275.3	7282.2	7287.3	7293	7296	7299.3	7305	7308
7311	7314	7317.3	7323.3	7332	7335.3	7341	7344	7347.3	7353	7356	7359
7362	7365.2	7371.2	7377.3	7383	7386	7395.2	7401	7404	7407	7413.3	7419.3
7426.2	7431.3	7437	7443.3	7449	7452	7455	7458	7461.3	7467.3	7473.3	7479
7482	7491	7497.2	7503.3	7509.2	7515.2	7524	7527	7530	7539	7545.3	7551
7554	7557	7560	7563.3	7569.3	7575	7578	7584	7587.3	7593.3	7599	7602
7605	7608	7611.3	7618.2	7623	7626	7629	7632	7635.3	7642	7647	7650
7653	7656	7659.3	7665.2	7671	7674	7677	7683.3	7689	7692	7695	7698
7702	7708	7713	7716	7719	7722	7725	7728	7731.3	7740	7743	7746
7752	7755	7758	7767	7770	7773	7779	7782	7786	7791	7794	7797
7803.3	7809	7815	7818	7822	7827.3	7834.2	7839	7842	7845	7848	7851
7854	7860	7863	7866	7869.3	7875	7878	7881.3	7887	7890	7893	7899
7902	7905	7908	7911	7914	7917	7920	7923.3	7929	7932	7935	7939.2
7947.2	7953.3	7962	7971.2	7977	7983	7986	7989	7992	7995	7998	8002.2
8007	8010	8013	8016	8019.3	8025.3	8031	8034	8037	8040	8043.3	8052
8055	8058	8061	8064	8067.3	8074.2	8079	8082	8085	8091.2	8097.2	8103
8106	8115	8121	8124	8127	8133.3	8139.3	8145	8148	8157	8163.3	8169
8172	8175	8181.3	8187.2	8194.2	8199	8202	8205	8208	8211.3	8218	8223
8226	8235	8241	8247	8250	8259	8265.3	8271	8274	8277	8280	8283.3
8289	8292	8298	8301	8307	8313.3	8319	8325	8328	8332	8343	8346
8349	8352	8355	8358	8362	8367	8370	8373	8385	8391	8394	8403
8409	8412	8415	8418	8427.3	8433	8436	8439	8442	8445	8448	8451.3
8457	8460	8466	8469.3	8475.3	8484	8487	8493	8496	8499	8502	8517
8523	8526	8529	8535	8538	8541.2	8550	8556	8559	8562	8565	8571
8574	8577.3	8586	8592	8595	8598	8601.3	8607	8610	8613	8616	8619
8622	8628	8631	8634	8637	8640	8643.3	8650	8655	8658	8661	8667
8673.2	8679	8682	8697	8703.3	8709	8715	8718	8721.3	8727.3	8736	8739.3
8745	8748	8751	8754	8757.3	8764.2	8769.3	8775	8778	8781	8784	8787.3
8793	8796	8799	8802	8805	8811.2	8820	8823	8826	8844	8847	8850
8853	8859	8862	8865	8868	8871	8874	8877	8880	8883.3	8889	8892
8895	8898	8901.3	8907	8910	8916	8919	8922	8925	8928	8931	8934
8938	8946	8949	8955	8967	8970	8974	8979	8985	8988	8991	8994

8997.2	9003.3	9009	9012	9015	9018	9021	9024	9027.3	9034.2	9039	9042
9045	9048	9051.3	9063	9066	9069	9072	9075	9078	9082	9087	9090
9099	9111	9114	9118	9123	9129	9132	9135	9141	9147.3	9153.3	9159
9162	9168	9171.3	9177.3	9183	9186	9189	9192	9195	9204	9207	9213
9216	9220.2	9237	9243	9246	9249	9255	9258	9262	9267	9270	9273.3
9279	9282	9285	9291	9294	9298	9303	9306	9309	9312	9315	9318
9324	9327	9330	9333	9336	9339	9342	9363.2	9370	9375	9378	9381
9387	9393.2	9399	9402	9405	9417	9423	9429	9435	9442.2	9447	9456
9459.3	9468	9477.2	9486	9489	9492	9495	9498	9501	9504	9507	9513
9516	9519	9522	9525	9531.2	9537	9540	9543	9546	9555	9561	9564
9567	9570	9573.2	9579	9582	9588	9591	9594	9604.2	9612	9615	9618
9627.2	9639	9645	9651	9654	9657.2	9666	9669.2	9675	9678	9687	9699.2
9708	9711	9717	9720	9729	9732	9735	9738	9741	9747.2	9753.3	9759
9762	9768	9771	9774	9780	9783	9789	9792	9795	9798	9802	9807
9810	9813	9819	9825	9831	9834	9843	9849	9852	9855	9858	9861
9867	9870	9874.2	9879	9885	9888	9891.2	9900	9903	9909.3	9915	9924
9927	9933	9939.3	9948	9957	9963.3	9978	9981.2	9987.3	9993	9996	9999
10002	10005.3	10011	10014	10017.3	10026	10029.3	10035	10038	10041.3	10047	10053
10056	10059	10068	10071	10074	10077	10080	10083	10086	10095	10098	10107
10113	10119	10122	10137	10143	10146	10149	10155	10158	10162	10167	10176
10179	10188	10197.2	10209	10218	10221	10224	10228	10233	10236	10239	10245
10248	10251.3	10257	10260	10263	10266	10269	10272	10275.3	10281	10284	10287
10293	10296	10299	10302	10305	10308	10317	10323.3	10329	10332	10335	10338
10341.3	10347.3	10356	10359	10362	10365	10368	10371	10374	10378	10386	10389
10395	10401	10407	10419	10425	10428	10431	10434	10443	10452	10455	10461
10464	10467	10476	10485	10492	10497	10506	10509	10512	10515	10518	10521.3
10527	10530	10539	10545	10548	10551	10554	10569.2	10575	10578	10581	10587.2
10593	10599	10605	10608	10611.3	10617.3	10623	10629	10632	10635	10638	10644
10647	10650	10653	10656	10659.3	10671	10674	10677	10683	10686	10689	10695
10698	10701	10707	10710	10716	10722	10725	10731.3	10749	10755	10764	10773
10779	10788	10794	10797	10800	10803	10815	10818	10821	10827	10833.2	10839
10842	10845	10857	10863	10866	10869	10875	10878	10882.2	10887	10893	10896
10899.2	10908	10917.3	10923	10929	10932	10938	10941	10944	10947.2	10953	10956
10962	10965	10972	10977	10980	10986	10995	11001	11004	11007	11013	11016
11019	11028	11037	11043.3	11049	11052	11055	11058	11061.3	11067	11076	11079
11082	11085	11088	11091	11094	11097.2	11106	11115	11122.2	11127	11139	11148
11151	11154	11157	11163	11169	11172	11175	11178	11181.3	11187	11193	11196
11199	11202	11205	11208	11226	11229	11232	11235	11238	11242	11247	11250
11259	11265	11271	11274	11277	11286	11289	11295	11298	11301	11307	11319
11325	11328	11331.2	11340	11349	11352	11355	11364	11373	11376	11379	11397

11403	11415	11418	11422	11427	11436	11442	11445	11451	11454	11458	11469.2
11475	11484	11493	11499.2	11514	11517	11523	11535	11538	11541	11547	11562
11565	11571.3	11577	11580	11583	11586	11589	11595	11598	11604	11607	11616
11619.2	11628	11631	11637.2	11646	11649	11655	11658	11661	11664	11667.2	11673
11679	11685	11697.2	11703	11715	11721	11724	11727	11733	11736	11740	11745
11751	11754	11757	11760	11763.3	11769	11772	11775	11778	11781.3	11787.3	11793
11796	11802	11805	11808	11811	11814	11818.2	11823	11826	11829.3	11835	11838
11847	11850	11853	11859	11862	11865	11868	11871	11874	11877	11880	11883
11886	11890.2	11895	11898	11901	11904	11907	11916	11922	11925	11928	11932
11946	11949	11955	11958	11962	11967	11973	11979	11985	11994	12003	12009.3
12015	12018	12021	12028	12034.2	12045	12051	12069	12072	12075	12081	12084
12099	12111	12114	12123.2	12132	12135	12138	12141.2	12147	12150	12156	12159
12162	12165	12168	12171	12174	12177	12186	12189.2	12195	12198	12202.2	12207
12213	12219	12228	12231	12234	12237	12243	12255	12261	12267	12282	12285
12297	12303	12306	12309	12315	12318	12322.2	12327	12333	12339	12348	12357
12360	12364	12370	12378	12381	12387	12394.2	12399	12405	12411.2	12420	12423
12426	12429	12435	12438	12444	12447	12453	12456	12459	12465.2	12474	12477
12483.3	12489	12492	12495	12498	12501.3	12507	12516	12519	12522	12525	12531.3
12546	12549	12555	12561	12564	12567	12579	12585	12588	12591	12594	12597
12603	12612	12615	12624	12627.2	12633	12636	12642	12645	12652	12657	12663
12666	12672	12675	12678	12681.3	12687	12690	12693	12699	12705	12708	12711
12714	12723.2	12729.3	12735	12738	12741	12744	12748.2	12753	12756	12759	12762
12765	12768	12771.3	12777	12780	12783	12786	12792	12795	12798	12804	12807
12813	12816	12819.3	12831	12834	12837	12843	12846	12849	12855	12858	12861
12867	12870	12876	12879	12882	12885	12891	12894	12909	12915	12922.2	12933
12939	12948	12954	12957	12960	12963	12975	12981	12987	13002	13023	13026
13035	13038	13042.2	13047	13053	13059	13068	13080	13083.2	13089	13092	13098
13101	13107	13114.2	13119	13122	13131	13140	13143	13146	13155	13164	13167
13179	13182	13185	13188	13203.3	13209	13215	13218	13222.2	13227	13236	13239
13242	13245	13248	13252.2	13258	13266	13269	13275	13278	13281	13287	13299
13305	13308	13311	13314	13317	13323	13332	13335	13341	13347	13356	13365
13371	13386	13392	13395	13407	13410	13419	13431	13434	13437	13446	13455
13458	13461	13467	13479	13491	13500	13512	13521	13524	13533	13536	13551
13554	13557	13563.2	13572	13575	13581	13587	13590	13605	13611	13620	13626
13635	13638	13641	13644	13647	13654.2	13659	13665	13674	13683.2	13692	13695
13698	13701	13707.2	13713	13719	13722	13732	13737	13743	13746	13749	13752
13755	13762.2	13767	13770	13773	13779	13782	13785	13788	13791	13797	13809
13812	13815	13818	13821	13824	13827.2	13833.3	13839	13842	13845	13851	13857
13860	13863	13866	13875	13887	13893	13896	13899	13902	13905	13908	13917
13923	13926	13932	13938	13941	13944	13947	13959	13962	13965	13971	13974

13989	13995	13998	14001	14007	14013	14019	14028	14043	14052	14061	14064
14067	14076	14085	14091.2	14106	14115	14127	14130	14139	14154	14175	14178
14187	14199	14211	14220	14235	14244	14253	14277	14283.2	14292	14295	14302
14307	14310	14319	14322	14325	14334	14337.3	14355	14358	14361	14364	14367
14379	14388	14391	14403	14409	14412	14421	14427.2	14433	14436	14439	14442
14451.2	14457	14460	14463	14466	14469	14472	14475	14478	14481.3	14487	14490
14493	14496	14499	14502	14505	14508	14514	14517	14520	14523.3	14529	14532
14538	14547.3	14553	14556	14559	14562	14565	14571	14580	14583	14586	14595
14601	14604	14607	14613	14616	14619.3	14625	14628	14643.3	14652	14655	14658
14661	14664	14667	14670	14679	14682	14685	14691	14694	14709	14715	14722
14727	14739	14748	14751	14757	14772	14781	14784	14787	14796	14805	14811
14826	14829	14835	14847	14850	14871	14874	14898	14907	14919	14931	14940
14955	14964	14997	15003	15027	15039	15042	15054	15060	15063	15082.2	15099
15108	15117	15123	15135	15147	15159	15162	15171	15180	15183	15186	15189
15198	15204	15207	15219	15228	15279	15285	15291	15303	15315	15324	15327
15333	15342	15351	15364	15372	15387	15423	15429	15447	15450	15471	15474
15477	15483	15492	15495	15507	15516	15537	15540	15549	15562.2	15567	15573
15579	15591	15603	15606	15612	15615	15618	15627	15630	15633	15636	15639
15642	15651	15654	15657	15660	15663	15675	15684	15687	15693	15699	15702
15717	15723	15726	15735	15738	15747	15753	15756	15759	15762	15780	15783
15789	15792	15795	15804	15813.3	15819	15828	15834	15837	15840	15843	15852
15855	15858	15861	15876	15879	15882	15903	15906	15909	15915	15918	15921
15924	15927	15930	15939.3	15945	15948	15951	15954	15966	15969	15975	15981
15987	15993	15999	16005	16011.3	16017	16020	16035	16044	16047	16071	16074
16077	16083.2	16092	16095	16102.2	16107	16119	16122	16134	16161	16167	16179
16185	16188	16197	16203	16212	16215	16221	16224	16227	16236	16245	16257
16266	16272	16284	16287	16290	16293	16299	16314	16323	16332	16338	16341
16350	16353	16356	16359	16362	16371	16380	16386	16398	16404	16407	16413
16419	16422	16437	16443	16449	16455	16458	16467	16476	16479	16482	16509
16524	16533	16539	16545	16548	16554	16557	16560	16563	16572	16575	16581
16587	16596	16602	16611	16623	16626	16635	16647	16650	16659	16662	16674
16689	16701	16707	16713	16719	16731	16737	16740	16743	16764	16767	16797
16804.2	16812	16815	16824	16827	16830	16839	16842	16863	16869	16887	16890
16893	16899	16902	16908	16911	16917	16923	16926	16932	16941	16944	16947
16956	16965	16971	16980	16989	16995	17002.2	17007	17010	17013	17019	17028
17034	17052	17055	17058	17061	17067	17073	17076	17079	17097	17103	17106
17115	17118	17121	17124	17127	17133	17139	17142	17145	17163	17166	17172
17175	17178	17187	17196	17199	17223	17226	17229	17232	17241	17244	17247
17250	17253	17256	17259	17262	17268	17295	17298	17301	17307	17319	17322
17325	17337	17346	17349	17355	17358	17367	17376	17379	17388	17397	17427

17445	17451	17463	17475	17484	17487	17493	17502	17505	17514	17523	17532
17547	17559	17577	17601	17607	17628	17631	17676	17703	17706	17715	17727
17730	17733	17739	17754	17763	17772	17787	17796	17811	17820	17829	17844
17859	17871	17874	17883	17895	17898	17901	17910	17916	17919	17922	17931
17934	17940	17943	17955	17964	17979	17988	17997	18003	18015	18027	18042
18063	18066	18084	18087	18123	18147	18156	18159	18180	18183	18186	18195
18204	18207	18228	18246	18249	18252	18255	18258	18267	18276	18279	18315
18327	18351	18354	18372	18375	18381	18396	18405	18447	18450	18459	18498
18501	18507	18519	18531	18564	18597	18603	18615	18627	18639	18642	18678
18681	18687	18699	18708	18711	18723	18747	18753	18783	18786	18804	18807
18849	18858	18867	18879	18891	18903	18927	18933	18945	18966	18975	18987
19029	19059	19068	19071	19101	19113	19116	19143	19146	19155	19167	19179
19209	19218	19227	19239	19260	19284	19335	19347	19359	19377	19380	19407
19419	19428	19431	19437	19446	19449	19461	19467	19473	19476	19479	19491
19503	19506	19527	19545	19548	19575	19578	19587	19593	19599	19611	19617
19620	19644	19647	19671	19674	19683	19686	19692	19695	19704	19707	19710
19719	19734	19743	19749	19761	19767	19770	19779	19788	19791	19797	19806
19821	19824	19827	19836	19845	19848	19857	19869	19872	19875	19887	19899
19908	19932	19935	19938	19941	19947	19950	19956	19959	19962	19974	19977
19980	19983	19986	19998	20004	20013	20019	20022	20031	20037	20043	20046
20049	20058	20067	20076	20079	20082	20109	20124	20133	20139	20148	20154
20163	20175	20187	20202	20223	20226	20244	20247	20259	20283	20289	20292
20301	20316	20319	20331	20340	20343	20346	20364	20367	20388	20391	20394
20406	20409	20415	20418	20427	20436	20439	20487	20508	20514	20532	20535
20541	20544	20556	20565	20571	20577	20586	20595	20604	20607	20610	20619
20634	20652	20658	20676	20679	20682	20691	20697	20700	20703	20706	20718
20724	20727	20733	20739	20763	20769	20775	20778	20787	20796	20799	20802
20829	20832	20844	20853	20856	20859	20868	20874	20883	20895	20898	20907
20919	20922	20940	20943	20946	20964	20967	20979	21021	21027	21039	21045
21051	21060	21063	21075	21084	21087	21111	21132	21147	21150	21153	21159
21180	21183	21207	21210	21228	21231	21258	21264	21273	21276	21279	21285
21303	21315	21327	21339	21372	21375	21378	21396	21399	21405	21408	21411
21420	21429	21432	21435	21444	21459	21471	21474	21477	21483	21495	21498
21516	21519	21522	21525	21531	21534	21555	21564	21579	21585	21597	21603
21612	21615	21627	21636	21642	21651	21663	21684	21690	21702	21714	21723
21732	21747	21771	21777	21780	21783	21786	21804	21807	21828	21837	21846
21849	21852	21855	21867	21876	21879	21897	21915	21924	21927	21930	21951
21954	21972	21981	21984	21987	21993	21996	22002	22023	22029	22047	22050
22089	22098	22101	22107	22119	22128	22131	22155	22161	22173	22239	22257
22278	22281	22299	22308	22341	22347	22353	22383	22386	22407	22437	22467

22479	22491	22503	22524	22572	22587	22650	22713	22719	22779	22839	22872
22875	22881	22884	22932	22998	23001	23019	23028	23031	23037	23043	23127
23142	23157	23187	23193	23196	23223	23268	23277	23283	23286	23289	23292
23295	23298	23307	23316	23319	23322	23325	23346	23349	23355	23379	23385
23388	23403	23412	23415	23445	23451	23466	23475	23487	23499	23514	23517
23532	23538	23541	23547	23550	23556	23571	23601	23604	23613	23619	23652
23658	23676	23679	23682	23706	23718	23724	23727	23739	23748	23763	23769
23772	23799	23844	23847	23862	23865	23868	23871	23874	23877	23886	23895
23898	23913	23919	23931	23937	23997	24003	24006	24027	24030	24039	24042
24054	24069	24081	24099	24108	24123	24141	24147	24156	24165	24171	24186
24189	24195	24207	24219	24252	24255	24258	24267	24270	24276	24279	24291
24315	24321	24333	24363	24396	24399	24402	24420	24423	24438	24441	24444
24447	24450	24459	24471	24483	24495	24507	24513	24519	24543	24546	24564
24567	24576	24597	24609	24618	24627	24639	24651	24684	24732	24747	24759
24783	24789	24801	24810	24819	24828	24861	24885	24897	24915	24927	24939
24975	24978	24999	25011	25014	25023	25035	25053	25083	25149	25173	25179
25188	25227	25263	25266	25287	25299	25329	25332	25356	25359	25371	25446
25449	25452	25476	25479	25527	25548	25551	25554	25572	25575	25581	25626
25650	25659	25692	25731	25740	25767	25779	25803	25818	25827	25836	25869
25884	25893	25908	25914	25923	25935	25983	25986	26019	26049	26061	26091
26124	26166	26172	26187	26250	26268	26298	26301	26313	26319	26346	26385
26412	26436	26445	26556	26595	26625	26628	26652	26655	26667	26676	26682
26706	26724	26742	26769	26793	26799	26817	26820	26844	26847	26874	26877
26886	26892	26970	27021	27033	27069	27099	27105	27129	27132	27156	27165
27189	27213	27276	27282	27300	27324	27357	27402	27411	27420	27444	27468
27540	27543	27564	27627	27633	27639	27657	27660	27663	27681	27690	27753
27765	27774	27777	27783	27807	27819	27825	27885	27915	27933	28020	28029
28059	28077	28146	28155	28167	28188	28209	28212	28227	28236	28251	28263
28332	28356	28359	28362	28371	28407	28455	28461	28476	28485	28506	28515
28539	28554	28563	28572	28653	28671	28683	28692	28695	28698	28707	28716
28791	28815	28818	28839	28842	28866	28893	28899	28908	28911	28923	28929
28932	28938	28953	28962	28986	29025	29037	29043	29049	29052	29067	29076
29079	29082	29085	29091	29139	29145	29148	29163	29172	29181	29196	29205
29217	29226	29250	29259	29319	29337	29343	29391	29403	29415	29427	29442
29469	29472	29523	29538	29559	29583	29586	29619	29622	29625	29628	29652
29655	29679	29685	29691	29715	29766	29772	29796	29802	29811	29823	29826
29835	29847	29850	29871	29895	29901	29907	29946	29949	29967	29979	29994
30027	30060	30075	30081	30084	30093	30111	30123	30132	30147	30159	30162
30201	30207	30231	30267	30273	30306	30327	30357	30369	30372	30378	30396
30399	30402	30411	30483	30486	30489	30492	30516	30519	30525	30555	30579

30621	30663	30699	30714	30738	30756	30759	30762	30771	30780	30786	30807
30816	30837	30843	30849	30858	30867	30876	30909	30915	30924	30933	30939
30948	30963	30975	31002	31023	31026	31059	31101	31119	31131	31140	31146
31167	31215	31227	31239	31275	31287	31332	31341	31410	31419	31491	31719
31743	31746	31788	31809	31815	31932	31983	32010	32061	32085	32127	32139
32187	32211	32253	32283	32319	32367	32487	32571	32646	32649	32652	32676
32679	32697	32715	32745	32823	32826	32829	32865	32871	32874	32925	32931
32940	32997	33003	33027	33042	33057	33081	33111	33117	33141	33171	33177
33183	33186	33195	33207	33219	33237	33249	33252	33258	33279	33309	33339
33363	33366	33369	33372	33396	33399	33405	33429	33435	33438	33459	33471
33495	33501	33546	33579	33588	33591	33594	33615	33651	33657	33675	33681
33705	33717	33723	33729	33732	33738	33747	33756	33801	33804	33831	33834
33855	33867	33882	33906	33924	33927	33939	33957	33969	33978	33981	33999
34011	34083	34092	34107	34119	34191	34233	34263	34266	34338	34359	34362
34371	34443	34449	34476	34515	34524	34575	34623	34626	34659	34701	34725
34731	34740	34755	34767	34806	34812	34815	34827	34836	34884	34911	34941
34953	34956	34965	34971	34974	34983	34986	34995	35007	35019	35052	35076
35091	35100	35109	35139	35163	35196	35235	35244	35295	35343	35346	35388
35532	35547	35601	35661	35676	35706	35715	35727	35739	35748	35811	35940
35949	36099	36108	36171	36228	36291	36351	36372	36474	36516	36531	36579
36588	36675	36684	36705	36714	36891	37035	37074	37179	37188	37194	37218
37260	37335	37362	37506	37527	37539	37599	37611	37686	37692	37716	37791
37815	37866	37899	37971	37980	38043	38067	38115	38193	38223	38226	38268
38331	38340	38367	38403	38409	38427	38436	38445	38475	38481	38556	38586
38607	38619	38634	38655	38658	38679	38763	38772	38787	38838	38859	38862
38883	38907	38913	38976	38988	38997	39018	39123	39132	39159	39276	39306
39327	39453	39771	39915	39996	40047	40131	40323	40383	40452	40491	40572
40650	40857	40923	40929	41004	41067	41139	41148	41181	41211	41220	41289
41292	41355	41361	41364	41391	41394	41415	41433	41436	41442	41499	41577
41580	41598	41604	41613	41616	41619	41643	41649	41724	41745	41754	41775
41802	41823	41826	41850	41874	41901	41931	41940	41967	41997	42006	42018
42027	42036	42075	42111	42114	42135	42186	42189	42219	42234	42300	42315
42363	42375	42399	42420	42507	42519	42567	42588	42726	42732	42753	42756
42780	42795	42801	42876	42906	42927	42939	42954	43005	43020	43053	43083
43107	43197	43227	43263	43266	43287	43449	43452	43476	43515	43626	43659
43674	43803	43827	43947	43956	43983	43986	44172	44349	44379	44388	44475
44532	44559	44583	44601	44739	44748	44892	45531	45609	45675	45681	45834
45933	45963	46044	46143	46506	46539	46683	46722	47049	47175	47259	47268
47340	47355	47403	47553	47619	47625	47628	47691	47772	47817	47871	47943
47949	47985	48075	48093	48123	48159	48204	48267	48306	48327	48492	48519

48555	48612	48699	48732	48756	48771	48801	48852	48876	48924	48951	49026
49203	49206	49227	49644	49767	49779	49788	49851	49929	49935	49938	49959
49995	50001	50052	50127	50154	50253	50463	50466	50571	50859	51003	51135
51183	51186	51300	51327	51372	51387	51546	51579	51723	51756	51780	51804
51867	51876	51903	51906	51927	51945	51948	51954	52011	52017	52020	52086
52092	52107	52236	52257	52266	52269	52272	52287	52299	52308	52413	52443
52452	52476	52479	52482	52506	52509	52521	52524	52533	52587	52623	52626
52647	52659	52731	52812	52827	52911	52986	53019	53058	53079	53163	53343
53346	53388	53451	53526	53532	53559	53595	53601	53676	53706	53727	53739
53754	53811	53820	53847	53853	54099	54171	54276	54315	54351	54474	54540
54627	54684	54783	54786	54996	55347	55506	55527	55548	55611	55692	55716
55761	55836	55887	55914	55977	56427	56511	56679	56985	57165	57267	57276
57411	57579	57771	58674	58695	58809	58815	58818	58947	59001	59073	59106
59127	59148	59319	59361	59364	59385	59505	59565	59571	59580	59667	59757
59868	59919	59943	60006	60027	60141	60156	60195	60546	60732	60876	60906
61083	61905	62457	62487	62817	62919	62988	63243	63279	63324	63423	63465
63495	63540	63684	63756	63972	64002	64188	64287	64332	64401	64476	64539
64866	64908	65052	65121	65196	65340	65373	65484	65628	65772	65916	65946
65988	66129	66201	66204	66213	66306	66381	66492	66633	66852	66876	66924
66945	67137	67197	67206	67209	67212	67227	67305	67839	67932	68076	68283
68499	68652	68751	68826	68832	68874	68940	68961	69039	69183	69186	69363
69372	69615	69660	69759	69903	69906	69993	70092	70119	70221	70236	70287
70311	70371	70479	70539	71244	71346	71385	71457	71547	71631	71919	71949
72252	72351	72783	72786	72861	72891	73113	73116	73179	73245	73332	73611
73755	74076	77355	77643	77652	78183	78327	78435	78687	78759	78876	78927
79308	79839	80706	81099	82188	82332	82764	83052	84132	84201	84492	85140
85356	85407	85533	85572	85587	85746	85932	86076	86466	86652	86667	86793
86796	86847	86940	87003	87186	87228	87927	88809	88827	89244	89775	89964
90684	90966	90972	91146	91404	95436	96012	96732	96876	97452	104139	104283
104652	105228	108972	109287	109683	109692	110268	110439	110844	110943	110988	111699
111867	113292	114012	121932	127692	128412	131292	137628				

Bibliography

[1] ABEL, R. J. R., BROUWER, A. E., COLBOURN, C. J., AND DINITZ, J. H. Mutually orthogonal Latin squares (MOLS). In *The CRC Handbook of Combinatorial Designs*, C. J. Colbourn and J. H. Dinitz, Eds. CRC Press, Inc., Boca Raton, 1996, ch. II.2, pp. 111–142.

[2] AHO, A. V., HOPCROFT, J. E., AND ULLMAN, J. D. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Massachusetts, 1974.

[3] ALBANESE, A., BLÖMER, J., EDMONDS, J., LUBY, M., AND SUDAN, M. Priority encoding transmission. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science* (1994), IEEE, pp. 604–612.

[4] ALON, N. Eigenvalues and expanders. *Combinatorica 6* (1986), 83–96.

[5] ALON, N. Eigenvalues, geometric expanders, sorting in rounds, and Ramsey theory. *Combinatorica 6* (1986), 207–219.

[6] Alon, N., Bruck, J., Naor, J., Naor, M., and Roth, R. M. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Trans. Inform. Theory 38* (1992), 509–516.

[7] Alon, N., Edmonds, J., and Luby, M. Linear time erasure codes with nearly optimal recovery (extended abstract). In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science* (1995), IEEE, pp. 512–519.

[8] Alpert, S. R. Twofold triple systems and graph imbeddings. *J. Combin. Theory Ser. A 18* (1975), 101–107.

[9] Balding, D. J., and Torney, D. C. Optimal pooling designs with error detection. *J. Combin. Theory Ser. A 74* (1996), 131–140.

[10] Barillot, E., Lacroix, B., and Cohen, D. Theoretical analysis of library screening using an n-dimensional pooling strategy. *Nucleic Acids Research 19* (1991), 6241–6247.

[11] Bassalygo, L. A., and Pinsker, M. S. Limited multiple-access of a non-synchronous channel. *Problemy Peredachi Informatsii 19* (1983), 92–96.

[12] Bassalygo, L. A., Zyablov, V. V., and Pinsker, M. S. Problems of complexity in the theory of correcting codes. *Problems Inform. Transmission 13* (1977), 166–175.

[13] Bennett, F. E. Latin squares with pairwise orthogonal conjugates. *Discrete Math. 36* (1981), 117–137.

[14] Berger, T., Mehravari, N., Towsley, D., and Wolf, J. Random multiple-access communications and group testing. *IEEE Trans. Commun. 32* (1984), 769–778.

[15] BETH, T., JUNGNICKEL, D., AND LENZ, H. *Design Theory.* Cambridge University Press, Cambridge, 1993.

[16] BIEN, F. Constructions of telephone networks by group representations. *Notices Amer. Math. Soc. 36* (1989), 5–22.

[17] BOLLOBÁS, B. *Combinatorics: Set Systems, Hypergraphs, Families of Vectors, and Combinatorial Probability.* Cambridge University Press, Cambridge, 1986.

[18] BOLLOBÁS, B., AND ROSENFELD, M. Sorting in one round. *Israel J. Math. 38* (1981), 154–160.

[19] BONEH, D., AND SHAW, J. Collusion-secure fingerprinting for digital data. In *Advances in Cryptology – CRYPTO '95*, D. Coppersmith, Ed., vol. 963 of *Lecture Notes in Computer Science.* Springer-Verlag, Berline, 1995, pp. 452–465.

[20] BRASSIL, J. T., LOW, S., MAXEMCHUK, N. F., AND O'GORMAN, L. Electronic marking and identification techniques to discourage document copying. *IEEE J. Selected Areas in Commun. 13* (1995), 1495–1504.

[21] BROUWER, A. E. Steiner triple systems without subconfigurations. Technical report ZW 104/77, Mathematisch Centrum Amsterdam, Amsterdam, 1977.

[22] BROUWER, A. E. Optimal packings of K_4's into a K_n. *J. Combin. Theory Ser. A 26* (1979), 278–297.

[23] BROUWER, A. E. A series of separable designs with application to pairwise orthogonal Latin squares. *European J. Combin. 1* (1980), 39–41.

[24] BROUWER, A. E., AND VAN REES, G. H. J. More mutually orthogonal Latin squares. *Discrete Math. 39* (1982), 263–281.

[25] Bruno, W. J., Balding, D. J., Knill, E. H., Bruce, D., Whittaker, C., Doggett, N., Stallings, R., and Torney, D. C. Design of efficient pooling experiments. *Genomics 26* (1995), 21–30.

[26] Buhrman, H., Hemaspaandra, E., and Longpré, L. SPARSE reduces conjunctively to TALLY. *SIAM J. Comput. 24* (1995), 673–681.

[27] Busschbach, P. Constructive methods to solve the problems of: s-surjectivity, conflict resolution, coding in defective memories. Tech. Rep. 84D005, École Nationale Supér. Télécomm., Paris, 1984.

[28] Capetanakis, J. I. Generalized TDMA: The multi-accessing tree protocol. *IEEE Trans. Commun. 27* (1979), 1479–1485.

[29] Capetanakis, J. I. Tree algorithms for packet broadcast channels. *IEEE Trans. Inform. Theory 25* (1979), 505–515.

[30] Chang, X. M., and Hwang, F. K. The minimax number of calls for finite population multi-access channels. In *Computer Networking and Performance Evaluation*, T. Hasegawa, H. Takagi, and Y. Takahashi, Eds. Elsevier, Amsterdam, 1986, pp. 381–388.

[31] Chang, X. M., Hwang, F. K., and Weng, J. F. Optimal detection of two defectives with a parity check device. *SIAM J. Discrete Math. 1* (1988), 38–44.

[32] Chang, Y. A bound for Wilson's theorem (II). *J. Combin. Des. 4* (1996), 11–26.

[33] Chaudhuri, S., and Radhakrishnan, J. Deterministic restrictions in circuit complexity. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing* (1996), ACM, pp. 30–36.

[34] Chee, Y. M., and Royle, G. F. Enumeration of small nonisomorphic 1-rotational twofold triple systems. *Math. Comp. 59* (1992), 609–612.

[35] Chen, C. C., and Hwang, F. K. Detecting and locating electrical shorts using group testing. *IEEE Trans. Circuits Systems 36* (1989), 1113–1116.

[36] Chen, R. W., and Hwang, F. K. K-definite group testing and its application to polling in computer networks. *Congr. Numer. 47* (1985), 145–159.

[37] Colbourn, C. J. Private communication.

[38] Colbourn, C. J. Generating Kirkman triple systems quickly. Unpublished manuscript, 1995.

[39] Colbourn, C. J., Colbourn, M. J., Harms, J. J., and Rosa, A. A complete census of $(10, 3, 2)$ block designs and of Mendelsohn triple systems of order ten, III: $(10, 3, 2)$ block designs without repeated blocks. *Congr. Numer. 39* (1983), 211–234.

[40] Colbourn, C. J., Gibbons, P. B., Mathon, R., Mullin, R. C., and Rosa, A. The spectrum of orthogonal Steiner triple systems. *Canad. J. Math. 46* (1994), 239–252.

[41] Colbourn, C. J., Hoffman, D. G., and Rees, R. A new class of group divisible designs with block size three. *J. Combin. Theory Ser. A 59* (1992), 73–89.

[42] Colbourn, C. J., and Rosa, A. Leaves, excesses and neighbourhoods in triple systems. *Australas. J. Combin. 4* (1991), 143–178.

[43] Colbourn, M. J. *Cyclic Block Designs: Computational Aspects of Their Construction and Analysis.* PhD thesis, Department of Computer Science, University of Toronto, Toronto, August 1980.

[44] COLBOURN, M. J., AND COLBOURN, C. J. Cyclic block designs with block size 3. *European J. Combin. 2* (1981), 21–26.

[45] DISK/TREND, INC. *1994 DISK/TREND Report: Disk Drive Arrays.* 1925 Landings Drive, Mountain View, California, April 1994.

[46] DORFMAN, R. The detection of defective members of large populations. *Ann. Math. Statist. 14* (1943), 437–440.

[47] DOYEN, J. Sur la structure de certains systéme triples de Steiner. *Math. Z. 111* (1969), 289–300.

[48] DU, D.-Z., AND HWANG, F. K. *Combinatorial Group Testing and its Applications.* World Scientific, Singapore, 1993.

[49] DUCROCQ, P. M., AND STERBOUL, F. Les G-systémes triples. *Ann. Discrete Math. 9* (1980), 141–145.

[50] DYACHKOV, A. G., AND RYKOV, V. V. Bounds on the lengths of disjunctive codes. *Problemy Peredachi Informatsii 18* (1982), 7–13.

[51] DYACHKOV, A. G., AND RYKOV, V. V. Обзор теории дизъюнктивныx кодов. *Problems Control Inform. Theory 12* (1983), 229–242.

[52] DYACHKOV, A. G., RYKOV, V. V., AND RASHAD, A. M. Superimposed distance codes. *Problems Control Inform. Theory 18* (1989), 237–250.

[53] EMCH, A. Triple and multiple systems, their geometric configurations and groups. *Trans. Amer. Math. Soc. 31* (1929), 25–42.

[54] ERDÖS, P. On sequences of integers no one of which divides the product of two others and some related problems. *Izv. Naustno-Issl. Inst. Mat. i Meh. Tomsk. 2* (1938), 74–82.

[55] ERDÖS, P. Problems and results in combinatorial analysis. In *Colloq. Internat. Sulle Teorie Combinatorie*, vol. 2. Acad. Naz. Lincei, Roma, 1976, pp. 3–17.

[56] ERDÖS, P. Problems and results in combinatorial analysis. *Creation in Mathematics 9* (1976), 25.

[57] ERDÖS, P., FRANKL, P., AND FÜREDI, Z. Families of finite sets in which no set is covered by the union of two others. *J. Combin. Theory Ser. A 33* (1982), 158–166.

[58] ERDÖS, P., FRANKL, P., AND FÜREDI, Z. Families of finite sets in which no set is covered by the union of r others. *Israel J. Math. 51* (1985), 79–89.

[59] ERDÖS, P., AND HANANI, H. On a limit theorem in combinatorial analysis. *Publ. Math. Debrecen 10* (1963), 10–13.

[60] ERDÖS, P., AND KLEITMAN, D. J. On coloring graphs to maximize the proportion of multicolored k-edges. *J. Combin. Theory 5* (1968), 164–169.

[61] ERDÖS, P., AND MOSER, L. Problem 35. In *Proceedings of the Conference on Combinatorial Structures and Applications* (Calgary, 1969, 1970), Gordon and Breach, p. 506.

[62] FRANKL, P., AND FÜREDI, Z. A new extremal property of Steiner triple systems. *Discrete Math. 48* (1984), 205–212.

[63] FRANKL, P., AND FÜREDI, Z. Union-free hypergraphs and probability theory. *European J. Combin. 5* (1984), 12–131.

[64] FRANKL, P., AND FÜREDI, Z. Union-free families of sets and equations over fields. *J. Number Theory 23* (1986), 210–218.

[65] FÜREDI, Z. Turán type problems. In *Surveys in Combinatorics, 1991*, A. D. Keedwell, Ed., no. 166 in London Mathematical Society Lecture Note Series. Cambridge University Press, Cambridge, 1991, pp. 253–300.

[66] GABBER, O., AND GALIL, Z. Explicit construction of linear-sized superconcentrators. *J. Comput. System Sci. 22* (1981), 407–420.

[67] GAREY, M. R., JOHNSON, D. S., AND SO, H. C. An application of graph coloring to printed circuit testing. *IEEE Trans. Circuits Systems 23* (1976), 591–599.

[68] GIBBONS, P. B. A census of orthogonal Steiner triple systems of order 15. *Ann. Discrete Math. 26* (1985), 165–182.

[69] GIBBONS, P. B., AND MATHON, R. The use of hill-climbing to construct orthogonal Steiner triple systems. *J. Combin. Des. 1* (1993), 27–50.

[70] GRABLE, D. More-than-nearly-perfect packings and partial designs. Preprint, 1994.

[71] GREENBERG, A. G., AND WINOGRAD, S. A lower bound on the time needed in the worst case to resolve conflicts deterministically in multiple access channels. *J. Assoc. Comput. Mach. 32* (1985), 589–596.

[72] GRIGGS, T. S., MURPHY, J., AND PHELAN, J. S. Anti-Pasch Steiner triple systems. *J. Combin. Inform. System Sci. 15* (1990), 79–84.

[73] HÄGGKVIST, R., AND HELL, P. Parallel sorting with constant time for comparisons. *SIAM J. Comput. 10* (1981), 465–472.

[74] HAMEL, A. M., MILLS, W. H., MULLIN, R. C., REES, R., STINSON, D. R., AND YIN, J. The spectrum of PBD($\{5, k^*\}, v$) for $k = 9, 13$. *Ars Combin. 36* (1993), 7–26.

[75] HANANI, H. Balanced incomplete block designs and related designs. *Discrete Math. 11* (1975), 255–369.

[76] HARTMAN, A., AND PHELPS, K. T. Steiner quadruple systems. In *Contemporary Design Theory: A Collection of Surveys*, J. H. Dinitz and D. R. Stinson, Eds. John Wiley & Sons, New York, 1992, ch. 6, pp. 205–240.

[77] HAYES, J. F. An adaptive technique for local distribution. *IEEE Trans. Commun. 26* (1978), 1178–1186.

[78] HEFFTER, L. Über das problem der nachbargebiete. *Math. Ann. 38* (1891), 477–508.

[79] HELLERSTEIN, L., GIBSON, G. A., KARP, R. M., KATZ, R. H., AND PATTERSON, D. A. Coding techniques for handling failures in large disk arrays. *Algorithmica 12* (1994), 182–208.

[80] HWANG, F. K. Combinatorial group testing[4]

[81] HWANG, F. K., AND SÓS, V. T. Non-adaptive hypergeometric group testing. *Studia Sci. Math. Hungar. 22* (1987), 257–263.

[82] JOHNSON, S. M. A new upper bound for error-correcting codes. *IEEE Trans. Inform. Theory 8* (1962), 203–207.

[4]Available from URL `http://www.dms.auburn.edu/~rodgecl/cadcom/applications/hwansnap/hwansnap.html`. This URL may not be stable.

[83] Kautz, W. H., and Singleton, R. R. Nonrandom binary superimposed codes. *IEEE Trans. Inform. Theory 10* (1964), 363–377.

[84] Kim, M. Synchronized disk interleaving. *IEEE Trans. Comput. 35* (1986), 978–988.

[85] Knill, E., and Muthukrishnan, S. Group testing problems in experimental molecular biology (preliminary report). Los Alamos Combinatorics E-print Server[5], LACES-94B-95-26 (1995).

[86] Ko, K. Distinguishing conjunctive and disjunctive reducibilities by sparse sets. *Inform. and Comput. 81* (1989), 62–87.

[87] Komlós, J., and Greenberg, A. G. An asymptotically fast nonadaptive algorithm for conflict resolution in multiple access channels. *IEEE Trans. Inform. Theory 31* (1985), 302–306.

[88] Körner, J. On the extremal combinatorics of the Hamming space. *J. Combin. Theory Ser. A 71* (1995), 112–126.

[89] Kramer, E. S., and Mesner, D. M. t-Designs on hypergraphs. *Discrete Math. 15* (1976), 263–296.

[90] Lefmann, H., Phelps, K. T., and Rödl, V. Extremal problems for triple systems. *J. Combin. Des. 1* (1993), 379–394.

[91] Ling, A. C. H. Private communication.

[92] Livny, M., Khoshafian, S., and Boral, H. Multi-disk management algorithms. In *Proceedings of the ACM SIGMETRICS Conference* (1987), ACM, pp. 69–77.

[5] Available from URL `http://www.c3.lanl.gov/dm/cgi/docOptions.cgi?document_name=94B-95-26`. This URL may not be stable.

[93] LU, J. *Collected Works on Combinatorial Designs.* Inner Mongolia People's Press, Mongolia, 1990.

[94] LUBOTZKY, A., PHILLIPS, R., AND SARNAK, P. Ramanujan graphs. *Combinatorica 8* (1988), 261–277.

[95] MACNEISH, H. F. Euler squares. *Ann. of Math. 23* (1922), 221–227.

[96] MACWILLIAMS, F. J., AND SLOANE, N. J. A. *The Theory of Error-Correcting Codes.* North Holland, Amsterdam, 1977.

[97] MARGULIS, G. A. Explicit group theoretical constructions of combinatorial schemes and their application to the design of expanders and concentrators. *Problems Inform. Transmission 24* (1988), 39–46.

[98] MATHON, R., AND ROSA, A. A census of Mendelsohn triple systems of order nine. *Ars Combin. 4* (1977), 309–315.

[99] MCKAY, B. D. Nauty user's guide (version 1.2). Technical report TR-CS-90-02, Department of Computer Science, Australian National University, Canberra, Australia, 1990.

[100] MOTWANI, R., AND RAGHAVAN, P. *Randomized Algorithms.* Cambridge University Press, Cambridge, 1995.

[101] MOZZOCHI, J. On the difference between consecutive primes. *J. Number Theory 24* (1986), 181–187.

[102] MULLIN, R. C. A generalization of the singular direct product with application to skem Room squares. *J. Combin. Theory Ser. A 29* (1980), 306–318.

[103] MULLIN, R. C., AND GRONAU, H.-D. O. F. PBDs: Recursive constructions. In *The CRC Handbook of Combinatorial Designs*, C. J. Colbourn and J. H. Dinitz, Eds. CRC Press, Inc., Boca Raton, 1996, ch. III.2, pp. 193–203.

[104] MULLIN, R. C., SCHELLENBERG, P. J., VANSTONE, S. A., AND WALLIS, W. D. On the existence of frames. *Discrete Math. 37* (1981), 79–104.

[105] MULLIN, R. C., AND YIN, J. On packings of pairs by quintuples: $v \equiv 3$, 9, or 17 (mod 20). *Ars Combin. 35* (1993), 161–171.

[106] NEWBERG, L., AND WOLFE, D. String layouts for a redundant array of inexpensive disks. *Algorithmica 12* (1994), 209–224.

[107] NGUYEN, Q. A., AND ZEISEL, T. Bounds on constant weight binary superimposed codes. *Problems Control Inform. Theory 17* (1988), 223–230.

[108] ODLYZKO, A. M. Asymptotic enumeration methods. In *Handbook of Combinatorics*, R. L. Graham, M. Grötschel, and L. Lovász, Eds., vol. II. North-Holland, Amsterdam, 1995, ch. 22, pp. 1063–1229.

[109] OLSON, M. V., HOOD, L., CANTOR, C., AND BOTSTEIN, D. A common language for physical mapping of the human genome. *Science 245* (1989), 1334–1335.

[110] ÖZDEN, B., RASTOGI, R., AND SILBERSCHATZ, A. Research issues in multimedia storage servers. *ACM Comput. Surveys 27* (1995), 617–620.

[111] PATTERSON, D. A., GIBSON, G. A., AND KATZ, R. H. A case for redundant arrays of inexpensive disks (RAID). In *Proceedings of ACM Conference on Management of Data* (1988), ACM, pp. 109–116.

[112] Pevzner, P. A., and Lipshutz, R. Towards DNA sequencing chips. In *19th International Conference on Mathematical Foundations of Computer Science* (1994), Lecture Notes in Computer Science, Springer-Verlag, pp. 143–158.

[113] Pippenger, N. Superconcentrators. *SIAM J. Comput. 6* (1977), 298–304.

[114] Pippenger, N. Bounds on the performance of protocols for a multiple-access broadcast channel. *IEEE Trans. Inform. Theory 27* (1981), 145–151.

[115] Rabin, M. O. Efficient dispersal of information for security, load balancing, and fault tolerance. *J. Assoc. Comput. Mach. 36* (1989), 335–348.

[116] Rao, T. R. N., and Fujiwara, E. *Error-Control Coding for Computer Systems.* Prentice Hall, New Jersey, 1989.

[117] Ray-Chaudhuri, D. K., and Wilson, R. M. Solution of Kirkman's schoolgirl problem. *Proc. Symp. Pure Math. Amer. Math. Soc. 19* (1971), 187–204.

[118] Reiman, I. über ein problem von Zarankiewicz. *Acta Math. Acad. Sci. Hungar. 9* (1958), 269–278.

[119] Ringel, G. *Map Color Theorem.* Springer-Verlag, Berlin, 1974.

[120] Ringel, G., and Youngs, J. W. T. Solution of the Heawood map-coloring conjecture. *Proc. Nat. Acad. Sci. U.S.A. 60* (1968), 438–445.

[121] Rivest, R. L. Cryptography. In *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, J. van Leeuwen, Ed. Elsevier, 1990, ch. 13, pp. 717–755.

[122] Royle, G. 1000000 2-(12, 3, 2) designs. *Abstracts Amer. Math. Soc. 10* (1989), 482.

[123] RUSZINKÓ, M. On the upper bound of the size of the r-cover-free families. *J. Combin. Theory Ser. A 66* (1994), 302–310.

[124] SALEM, K., AND GARCIA-MOLINA, H. Disk stripping. In *Proceedings of the 2nd IEEE International Conference on Data Engineering* (1986), IEEE, pp. 336–342.

[125] SAVAGE, J. E. The complexity of decoders – part I: Decoder classes. *IEEE Trans. Inform. Theory 15* (1969), 689–695.

[126] SAVAGE, J. E. The complexity of decoders – part II: Computational work and decoding time. *IEEE Trans. Inform. Theory 17* (1971), 77–85.

[127] SCHÖNHEIM, J. On maximal systems of k-tuples. *Studia Sci. Math. Hungar. 1* (1966), 363–368.

[128] SCHÖNHEIM, J. On the number of mutually disjoint triples in Steiner systems and related maximal packing and minimal covering systems. In *Recent Progress in Combinatorics, Proceedings of the Third Waterloo Conference on Combinatorics, May 1968*, W. T. Tutte, Ed. Academic Press, New York, 1969, pp. 311–318.

[129] SCHULTZ, D. J. *Topics in Nonadaptive Group Testing.* PhD thesis, Department of Statistics, Temple University, Philadelphia, 1992.

[130] SHANNON, C. E. A mathematical theory of communication. *Bell System Tech. J. 27* (1948), 379–423.

[131] SHOUP, V. New algorithms for finding irreducible polynomials over finite fields. *Math. Comp. 54* (1990), 435–447.

[132] SKILLING, J. K. Method of electrical short testing and the like. *U.S. Patent 4342959* (1982).

[133] SLOANE, N. J. A. *A Short Course on Error Correcting Codes.* Springer-Verlag, Wien, 1975.

[134] SLOANE, N. J. A. Covering arrays and intersecting codes. *J. Combin. Des. 1* (1993), 51–63.

[135] SOBEL, M., AND GROLL, P. A. Group testing to eliminate efficiently all defectives in a binomial sample. *Bell System Tech. J. 28* (1959), 1179–1252.

[136] SPENCER, J. Maximal consistent families of triples. *J. Combin. Theory 5* (1968), 1–8.

[137] SPENCER, J. Asymptotic packing via a branching process. *Random Structures Algorithms 7* (1995), 167–172.

[138] SPERNER, E. Ein Satz über Untermengen einer endlichen Menge. *Math. Z. 27* (1928), 544–548.

[139] SPIELMAN, D. A. *Computationally Efficient Error-Correcting Codes and Holographic Proofs.* PhD thesis, Department of Mathematics, Massachusetts Institute of Technology, Cambridge, June 1995.

[140] STINSON, D. R. Hill-climbing algorithms for the construction of combinatorial designs. *Ann. Discrete Math. 26* (1985), 321–334.

[141] STINSON, D. R., AND FERCH, H. 2000000 Steiner triple systems of order 19. *Math. Comp. 44* (1985), 533–535.

[142] STINSON, D. R., AND WEI, R. Combinatorial properties and constructions of traceability schemes and frameproof codes. Preprint, 1996.

[143] TANNER, R. M. Explicit constructions of concentrators from generalized n-gons. *SIAM J. Algebraic Discrete Methods 5* (1984), 287–293.

[144] TSYBAKOV, B. S., AND MIKHAILOV, V. A. Free synchronous packet access in a broadcast channel with feedback. *Problems Inform. Transmission 14* (1978), 259–280.

[145] TURÁN, P. On an extremal problem in graph theory. *Mat. Fiz. Lapok 48* (1941), 436–452.

[146] TURÁN, P. On the theory of graphs. *Colloq. Math. 3* (1954), 19–30.

[147] VAKIL, F., AND PARNES, M. On the structure of a class of sets useful in nonadaptive group testing. *J. Statist. Plann. Inference 39* (1994), 57–69.

[148] VALIANT, L. G. Bulk-synchronous parallel computers. In *Parallel Processing and Artificial Intelligence*, M. Reeve and S. E. Zenith, Eds. Wiley, Chichester, 1989.

[149] VALIANT, L. G. General purpose parallel architectures. In *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, J. van Leeuwen, Ed. Elsevier, 1990, ch. 18, pp. 943–971.

[150] WAGNER, N. R. Fingerprinting. In *Proceedings of the 1983 Symposium on Security and Privacy.* IEEE Computer Society Press, Maryland, 1984, pp. 18–22.

[151] WILSON, R. M. An existence theory for pairwise balanced designs I: Composition theorems and morphisms. *J. Combin. Theory Ser. A 13* (1972), 220–245.

[152] WILSON, R. M. An existence theory for pairwise balanced designs II: The structure of PBD-closed sets and the existence conjecture. *J. Combin. Theory Ser. A 13* (1972), 246–273.

[153] WILSON, R. M. An existence theory for pairwise balanced designs III: Proof of the existence conjectures. *J. Combin. Theory Ser. A 18* (1975), 71–79.

[154] WORMALD, N. Differential equations for random processes and random graphs. Preprint, 1994.

[155] YIN, J. Private communication through R. C. Mullin.

[156] YIN, J. On the packing of pairs by quintuples with index 2. *Ars Combin. 31* (1991), 287–301.

[157] YIN, J. Packing pairs by quintuples: the case v congruent to 0 (mod 4). Preprint, 1992.

[158] YIN, J., LING, A. C. H., COLBOURN, C. J., AND ABEL, R. J. R. The existence of uniform 5-GDDs. Preprint, 1996.

[159] ZYABLOV, V. V. An estimate of the complexity of constructing binary linear cascade codes. *Problems Inform. Transmission 7* (1971), 3–10.