Applications of Combinatorial Designs to Communications, Cryptography, and Networking

C. J. Colbourn, J. H. Dinitz, D. R. Stinson

Abstract

Combinatorial designs have long had substantial application in the statistical design of experiments, and in the theory of error-correcting codes. Applications in experimental and theoretical computer science have emerged, along with connections with the theory of cryptographic communication. In this paper, we focus on another collection of recent applications in the general area of communications, including cryptography and networking. Applications have been chosen to represent those in which design theory plays a useful, and sometimes central, role. Moreover, applications have been chosen to reflect in addition the genesis of new and interesting problems in design theory in order to treat the practical concerns. Of many candidates, thirteen applications areas have been included. They are as follows:

1. Optical Orthogonal Codes
2. Synchronous Multiple Access to Channels
3. Group Testing and Superimposed Codes
4. Erasure Codes and Information Dispersal
5. Threshold and Ramp Schemes
6. Authentication Codes
7. Resilient and Correlation-immune Functions
8. Multidrop Networks
9. Channel Graphs and Interconnection Networks
10. Partial Match Queries on Files
11. Software Testing
12. Disk Layout and Striping
13. (t, m, s)-Nets and Numerical Integration

Our conclusion is that the theory of combinatorial designs continues to grow, in part as a consequence of the variety of these applications and the increasing depth of the connections with challenging problems on designs.

0 Background

The theory of combinatorial designs has a long and rich history. Its origins lie in somewhat specialized problems that arose in algebra, geometry, topology, and number theory. However, applications are found in the design of experiments [8] and in the theory of error-correcting codes [7]. Both fields of application assisted in providing a framework for the fledgling field, and both have served as sources for a wide variety of research directions.

In the past few decades, combinatorial design theory has grown to encompass a wider variety of investigations, many of which are not apparently motivated by any practical application. Rather they are motivated by a desire to obtain a coherent and powerful theory of existence and properties of designs. Nevertheless, it comes as no surprise that applications in experimental design and in coding theory continue to arise, and also that designs have found applications in new areas. Cryptography in particular has provided a new source of applications of designs, and simultaneously a source of new and challenging problems in design theory [9]. Across the spectrum of theoretical and experimental computer science, one finds similar connections [3].

Arguably, many of the connections that arise are somewhat superficial, and appear to require only the translation of elementary combinatorial properties to the application domain. Naturally, when this is the case, the limited application does not provide evidence of an important role for combinatorial design theory. However, we believe that there is ample evidence not only of superficial connections of theoretical investigations on designs to applications, but of deeper and more substantial connections. The importance of these connections cannot be overstated. While one can never know which results will find a genuine application, one expects to see such applications arise. Moreover, the evolution of the field depends to a large degree upon its ability to make contributions both to other theories and to applications.

In this paper, we present a number of applications in which the connection with designs appears to be substantial. We have selected applications primarily from the area of communications, including cryptography and networking but avoiding for the most part the well understood connections with error-correcting codes. Our objectives are to present evidence that combinatorial designs continue to arise in applications areas, often in unexpected ways; that the connections involve difficult aspects of the theory of designs; and that the applications motivate new research in design theory.

In view of these objectives, we do not provide an overview of design theory, but assume the reader is familiar with the major topics in the area; see [1, 2, 4, 5, 6] for comprehensive treatments. In general, we attempt to give a sufficient introduction to the applications problem and then outline the connection with designs. Our interpretation of what is a 'combinatorial design' is a liberal one. There are numerous objects that do not share the general structure of a balanced set system or an array of symbols, yet they are an integral part

of combinatorial design theory. We adopt an inclusive view, including for example all of the specialized types of 'designs' discussed in [2].

1 Optical Orthogonal Codes

A fiber optic channel must have the ability for multiple users to simultaneously share the channel without interference. In order to facilitate this, optical orthogonal codes were developed by Salehi [19]. Viewing these codes as sets of integers modulo n leads to interesting design theoretic questions.

1.1 The Application

The study of optical orthogonal codes was first motivated by an application in a fiber optic code-division multiple access channel. Many users wish to transmit information over a common wide-band optical channel. The objective is to design a system that allows the users to share the common channel. Other approaches have included frequency division, time division, collision detection or some type of network synchronization. They have required frequent conversions between the optical domain and the electrical domain. However, employing a code-division multiple access system with optical orthogonal codes reduces the complexity of the system, enabling implementation with available technology and with potentially higher transmission efficiency [15].

An $(n, w, \lambda_a, \lambda_c)$ optical orthogonal code (OOC), C, is a family of $(0,1)$-sequences of length n and weight w satisfying the following two properties (all subscripts are reduced modulo v):

1. $\sum_{0 \le t \le v-1} x_t x_{t+i} \le \lambda_a$ for any $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1})$ and any integer $i \not\equiv 0 \bmod v$ (the *auto-correlation property*);

2. $\sum_{0 \le t \le v-1} x_t y_{t+i} \le \lambda_c$ for any $\mathbf{x} = (x_0, \ldots, x_{n-1})$, $\mathbf{y} = (y_0, \ldots, y_{n-1})$, and any integer $i \not\equiv 0 \bmod v$ (the *cross-correlation property*).

When $\lambda_a = \lambda_c = \lambda$ the code is an (n, w, λ) optical orthogonal code. Research has concentrated on this case. To simplify the discussion of the model, let C be an $(n, w, 1)$ optical orthogonal code with m codewords. The communications system can handle up to m simultaneous transmitters. Each transmitter is assigned one codeword from C, so that transmitter T_i is assigned codeword $\{s_1, s_2, \ldots, s_w\} = c_i \in C$ (s_j denotes that the jth 1 in the (0,1) sequence; that is, codeword c_i is in the s_jth position). At the transmitter, every information bit of a signal is encoded into a frame of n optical chips: If the information bit is 1, then in the corresponding frame (consisting of n optical chips), photon pulses are sent at exactly the s_1th, s_2th, $\cdots$, and s_wth chip. In the other $n - w$ chips, no photon pulses are sent. If, however, the information bit is 0, then no photon pulses are sent in the corresponding frame (still consisting of n chips). For example, if transmitter T_i wishes to send the

message 101, this gets encoded as the sequence of 3 frames of length $3n$ where photon pulses are sent at times $s_1, s_2, \ldots s_w, 2n + s_1, 2n + s_2, \ldots 2n + s_w$.

All m users are allowed to transmit at any time; there is no network synchronization required. At the receiving end, decoders are used to separate the transmitted signals. The decoder consists of a bank of m tapped delay-lines, one for each codeword (so say D_i is the decoder for transmitter T_i). These delay taps on decoder D_i are a (possibly null) cyclic shift modulo n of those on T_i.

Each tapped delay-line can effectively calculate the correlation of the received waveform with its signature sequence. By the properties of optical orthogonal codes, the correlation between different signature sequences is low. The delay-line output is high only when the intended transmitter's information bit is a 1. In particular, the output is w when decoder D_i receives the information bit 1 from transmitter T_i and they are synchronized correctly. The output is s for some $s \leq m$ when D_i is not synchronized with T_i and T_i is sending a 1 or when T_i is sending a 0. Thus the receiver can effectively determine when the corresponding transmitter is transmitting an information bit of 1. ("Bit stuffing" assures that the receiver can determine when long strings of 0's are sent).

An important aspect of optical codes is that these codes consist of truly (0,1) sequences and are intended for environments that have no negative components. Most other correlation sequences are actually $(+1, -1)$ sequences intended for systems having both positive and negative components, an important distinction according to [15].

Recent work has also been done on using optical orthogonal codes for multimedia transmission in fiber-optic LANs [18] and in multirate fiber-optic CDMA systems [17]. The mathematical theory is quite similar to the case described.

1.2 The Connection to Designs

A convenient way of viewing optical orthogonal codes is from a set-theoretic perspective. An $(n, w, \lambda_a, \lambda_c)$ optical orthogonal code C can be considered as a family of w-sets of integers modulo n, in which each w-set corresponds to a codeword and the numbers in each w-set specify the nonzero bits of the codeword. One can reformulate the correlation properties in this set-theoretic framework. As an example, $C = \{1100100000000, 1010000100000\}$ is a (13,3,1) code with two codewords. In set theoretic notation, $C = \{\{0, 1, 4\}, \{0, 2, 7\}\}$ mod 13. The code is equivalent to a $(13, 3, 1)$ difference family in Z_{13} (which gives a Steiner triple system of order 13). Considering the set-theoretic interpretation of OOCs, one can expect that many of the constructions for optical orthogonal codes are design theoretic in nature. Indeed this is true. In this section we outline some constructions for OOCs that involve designs. To be consistent with design theoretic notation, we refer to (n, w, λ) OOCs as (v, k, λ) OOCs.

The main connection is to difference packings. Let $\mathcal{B} = \{B_1, B_2, \ldots, B_t\}$,

where $B_i = \{b_{i1}, b_{i2}, \ldots b_{ik}\}, b_{ij} \in Z_v, 1 \leq i \leq t$ and $1 \leq j \leq k$. The differences in $\mathcal{B}$ are $D = \{b_{ij} - b_{is} : 1 \leq i \leq t, 1 \leq j, s \leq k, j \neq s\}$. The pair $(Z_v, \mathcal{B})$ is called a *cyclic difference packing* or CP(v, k, λ) if the cardinality of D is exactly $\lambda k(k-1)t$ and $0 \notin D$. It can be easily verified that a (v, k, λ) cyclic difference packing gives a (v, k, λ) optical orthogonal code. A CP$(v, k, 1)$ is termed g-regular if the difference leave $(Z_v \setminus D)$ along with 0 forms an additive subgroup of Z_v having order g. When $\lambda = 1$ a cyclic difference packing satisfies the bound $t \leq \lfloor (v-1)/(k(k-1)) \rfloor$, and is *optimal* if $t = \lfloor (v-1)/(k(k-1)) \rfloor$.

A similar bound pertains to optical orthogonal codes. Let $\Phi(v, k, \lambda)$ denote the maximum number of codewords in an (v, k, λ) OOC. Analogous to the Johnson bound from coding theory, we have:

Theorem 1.1 *[15]* $\Phi(v, k, \lambda) \leq ((v-1)(v-2)\cdots(v-\lambda))/(k(k-1)(k-2)\cdots(k-\lambda))$

When $\lambda = 1$ this reduces to $\Phi(v, k, 1) \leq ((v-1)/k(k-1)$. When $|C| = \lfloor (v-1)/(k(k-1)) \rfloor$ the code is an *optimal* OOC.

Theorem 1.2 *[21] The existence of an optimal $(v, k, 1)$ optical orthogonal code is equivalent to the existence of an optimal $(v, k, 1)$ cyclic difference packing.*

Hence results on (optimal) cyclic difference packings directly relate to results on (optimal) OOCs.

When $k = 3$ and $\lambda = 1$ optimal OOCs arise from cyclic Steiner triple systems. In fact, when $v \equiv 1, 3 \pmod 6$ they coincide. Chung, Salehi and Wei [15] solve all cases when $v \not\equiv 2 \pmod 6$:

Theorem 1.3 $\Phi(v, 3, 1) = \lfloor \frac{(v-1)}{6} \rfloor$ *if* $v \not\equiv 2 \pmod 6$.

The proof is by a direct construction of $(v, 3, 1)$ cyclic difference packing and is very reminiscent of the construction of cyclic triple systems from Skolem sequences (see [4]). In a similar vein, Yin [21] summarizes results concerning optimal $(v, k, 1)$ cyclic difference packings, and hence gives many optimal OOCs. Most of these have $k = 4$, but there are some results for other $k \leq 11$.

Another class of optimal OOCs comes from projective geometry. Chung, Salehi and Wei use PG(d, q) to construct a $(v, k, 1)$ OOC where $v = (q^{d+1} - 1)/(q-1)$ and $k = q+1$. Let α be a primitive element of GF(q^{d+1}) and say that $\log \beta = e$ if $\beta = \alpha^e$. Now, in the vector space $V(d+1, q)$ the nonzero vectors on a line ℓ through the origin are $\ell = \{\alpha^i, \alpha^{i+v}, \alpha^{i+2v}, \ldots \alpha^{i+(q-2)v}\}$ where again $v = (q^{d+1} - 1)/(q-1)$. For any point $p \in$ PG(d, q), let $\log p$ denote the log of any vector on the line corresponding to p in $V(d+1, q)$ modulo v. Hence each line in the projective geometry corresponds to a subset of the integers modulo v.

Let a *cyclic shift* of a line L in PG(d, q) be the set of points $\{p | \log p = 1 + \log p' \pmod v$ for some point $p' \in L\}$. The cyclic shift of a line is also

a line in PG(d,q) so this creates a number of orbits of lines. If the number of lines in a orbit is v, then the orbit is termed *full*, otherwise it is a *short* orbit.

To construct an $(v,k,1)$ OOC from the projective geometry, take one representative line from each full orbit and map each of these lines to the set of integers modulo v under the action of the log. These sets satisfy the autocorrelation and cross-correlation restrictions. The interesting fact is that the optical orthogonal codes formed in this manner are optimal:

Theorem 1.4 *$\Phi(v,k,1) = \lfloor \frac{(v-1)}{k(k-1)} \rfloor$ when q is an odd prime power, $v = (q^{d+1}-1)/(q-1)$ and $k = q+1$.*

This construction can be extended to $\lambda > 1$ by using s-dimensional subspaces instead of 1-dimensional subspaces (lines).

A new construction by Chen, Ge and Zhu [14] gives $(6v,4,1)$ optimal OOCs for infinitely many odd values of v. They construct these OOCs directly from skew starters. A *starter* in the cyclic group Z_v (v odd) is a set of unordered pairs $S = \{\{x_i,y_i\} : 1 \le i \le (v-1)/2\}$ which satisfies the two properties: $(1)\{x_i : 1 \le i \le (v-1)/2\} \cup \{y_i\, 1 \le i \le (v-1)/2\} = Z_v \backslash \{0\}$ and (2) $\{\pm(x_i - y_i) : 1 \le i \le (v-1)/2\} = Z_v \backslash \{0\}$. The starter is a *skew* starter if in addition it satisfies the property (3) $\{\pm(x_i + y_i) : 1 \le i \le (v-1)/2\} = Z_v \backslash \{0\}$. Skew starters have been useful in the construction of Room squares, Hamiltonian path balanced tournament designs and other combinatorial designs. See [16] for a survey.

Assume that $\gcd(v,6) = 1$ and that $S = \{\{x_i,y_i\} : 1 \le i \le (v-1)/2\}$ is a skew starter in Z_v. In $Z_v \times Z_6$ (which is isomorphic to Z_{6v}), let $c_i = \{(x_i,0),(y_i,0),(x_i+y_i,1),(0,4)\}$ for $1 \le i \le (v-1)/2$. They show that the set $C = \{c_1, c_2 \ldots c_{(v-1)/2}\}$ forms an $(6v,4,1)$ optimal OOC. Using results on the existence of skew starters they derive:

Theorem 1.5 *There exists an optimal $(6v,4,1)$ optical orthogonal code in Z_{6v} for all v such that $\gcd(v,6) = 1$.*

Many combinatorial constructions for optimal $(v,k,1)$ optical orthogonal codes are given in a paper by Yin [21]. The following is a sample of one of the many recursive constructions in that paper. For the definition of difference matrices see [2].

Theorem 1.6 *Suppose that there exists a g-regular CP$(v,k,1)$, a (k,m)-difference matrix and an optimal CP$(gm,k,1)$, then there exists an optimal $(mv,k,1)$ OOC.*

Yin also gives a recursive construction for optimal OOCs using group divisible designs.

2 Synchronous Multiple Access to Channels

In the previous section, an application involving sharing an optical channel was examined. In that context, users acted asynchronously, and hence the channel decoding involved cyclic shifts of codewords. In this section, we examine an analogous problem. In this variant, however, rather than multiplexing by partitioning the channel's capacity into discrete time slots, we employ multiplexing based on available frequencies. This avoids some of the issues that arise in synchronization, but introduces some additional complexities.

2.1 The Application

Each user is to be able to send one of m different messages in a channel, or can remain silent. The channel is capable of carrying any subset of v different pulses or tones simultaneously, and can be equipped with intensity detection devices that determine not only the presence of a particular tone, but also the intensity with which this tone has been employed. The latter is usually measured in multiples of some basic nonzero intensity, and accurately distinguishes large variations in intensity. However, small variations are not considered to be significant, in order to allow for noise.

Each message for each user is mapped to a *codeword*, which indicates a selection of k of the v available tones (i.e., the scheme is *multi-tone*). When the transmitter is silent, no signal is sent. When active, the transmitter sends the combination of k tones corresponding to the desired message. One typically expects that a small number of transmitters is active. As with optical orthogonal codes, a receiver must be able to detect the presence of a message from a particular transmitter. For this reason, interference resulting from the simultaneous transmission of two (or more) codewords is to be kept to a minimum. When, for example, every transmitter is assigned only one message, we require that codewords for two different users share at most one tone. If all users are assigned one message only, and there is no intensity detection, then the task of the receiver is precisely that of solving a nonadaptive group testing problem (see §3). Intensity detection enables one to determine (with some degree of accuracy) the number of users who have transmitted a particular tone. In this case, the problem is the variant of nonadaptive group testing in which tests report not just the presence of a defective, but also the number of defectives (see [38, Chapter 5]). Codes for this type of "spread spectrum" signaling system are described in [22, 23, 26, 27].

The m-ary problem is described in some detail in [28, 29]. Transmitters have a collection of m different messages and can choose any one to send or remain silent. The intended application here is to signaling systems in which, despite the large number of users, traffic from each user is bursty (i.e., high volume but short duration). The design of the system optimizes the handling of traffic when a single user is active, but permits multiple access by a small number of users. We retain the requirement that codewords assigned to differ-

ent users share at most one tone. We enforce in addition a requirement that two codewords associated with different messages used by the same transmitter share no tone at all. This "orthogonality" requirement permits the most accurate decoding when a single user is active.

In order to accommodate multiple users, again it is necessary to be able to determine which combination of messages is present in the channel. For this reason, it is generally considered to be a poorer signaling design if one tone is used much more often than another. A secondary, but still important, criterion is therefore that all available tones appear in approximately the same number of assigned codewords.

2.2 The Connection to Designs

When every transmitter has a unique codeword, a simple design theoretic problem arises. Associate with each of the v tones an element, and with each codeword a block which is a subset of k elements. Blocks then have the property that they intersect in at most one element, and hence no pair occurs in more than one block. The result is a packing of index one, block size k, and order v. Maximizing the number of transmitters requires simply the choice of a maximum packing.

However, even this basic signaling problem poses some difficulties. When multiple transmitters are active concurrently, the received signal is the union of the transmitted signals. With intensity detection, the received signal is the multiset union. In the former situation, our task is to recover from the union the constituent sets; as noted, this is a nonadaptive group testing problem. It is not known whether, by assuming the availability of multiset unions rather than set unions, better packings can be found. Codes can be found for which multiset unions permit proper reception while set unions do not; however, in the case $k = 3$, the largest number of codewords can be realized by a code for which set unions suffice [24].

The multi-tone systems lead to difficult problems in design theory as well. We concentrate on the case when blocks are triples. A code to be used in a signaling system is necessarily a packing by triples. Suppose that the packing to be used is on v elements and has b triples. If every user is to be assigned m of the triples, we require that the m triples assigned form a partial parallel class (i.e., any two triples in the class are disjoint). Then $m \leq \lfloor v/3 \rfloor$, and the maximum number of users that can be supported cannot exceed $\lfloor b/m \rfloor$. A suitable code for s users consists of a packing on v elements with ms triples, partitioned into s partial parallel classes of size m. Among such packings, those in which every two elements appear in approximately the same number of triples are preferred.

In the case that $m = \lfloor v/3 \rfloor$, these packings have been extensively studied. For example, when $v \equiv 3 \pmod 6$, the solutions are Kirkman triple systems [29]. Indeed, the "frame" obtained by deleting a single element in a Kirk-

man triple system provides a solution when $v \equiv 2 \pmod 6$. When $v \equiv 0 \pmod 6$, nearly Kirkman triple systems provide solutions, and when $v \equiv 1 \pmod 6$, Hanani triple systems provide the codes. See [4] for more details about these types of triple systems.

Colbourn and Zhao [25] recently completed the solution when $v \equiv 4, 5 \pmod 6$, so that the determination of codes when the number of messages is maximum is complete. In the intended application, although it is plausible that the number of messages coincides with the maximum permitted, this is unlikely. The primary application is in systems employing a digital to analog conversion, so that a chunk of ℓ bits in an incoming datastream is converted to a message in the form of the k tones selected. Typically, then, we find that $m = 2^\ell$, so that m is a power of two. Zhao [28] observes that, beginning with a packing partitioned into maximum partial parallel classes, simple heuristics usually suffice to partition the same packing into more and shorter partial parallel classes. Despite this evidence that partitioning is more difficult when m is large, and the use of such partitions in forming codes for smaller m, the current state of affairs in our knowledge of triple systems is quite incomplete. To begin with, it is not always the case that solutions for large values of m ensures the existence of solutions for smaller values of m. The Kirkman triple system of order 9, for instance, admits a partition with 4 classes of size 3, but does not admit a partition with 6 classes of size 2. More importantly, no method with a performance guarantee appears to be available at present which permits one to massage a packing with large partial parallel classes into one that has smaller but more partial parallel classes. Indeed, the existence question given m, s, and v asking for a packing by triples on v elements and ms blocks which has a partition into s partial parallel classes of v has been solved only for certain restricted cases [25, 28, 29]. Among these are included the cases for all small values of v when m is a power of two, which form the principal cases employed in the application.

3 Group Testing and Superimposed Codes

In this section, a collection of applications to the design of codes for simultaneous communication, and to experimental design of pooling strategies, is examined. Du and Hwang [38] provide a much more detailed treatment.

3.1 The Application

A population $\mathcal{P}$ of b items contains a number d of *defective* items, and the remaining $b - d$ items are *good*. Items can be pooled together for testing: for a subset $X \subseteq \mathcal{P}$, the *group test* reports "yes" if X contains one or more defective elements, and reports "no" otherwise. The objective is to determine, using a number of group tests, precisely which items are defective. When group tests are all undertaken in parallel, the problem is *nonadaptive*; otherwise it is

adaptive. Then results from one or more tests are available while constructing further pools to be tested. Among adaptive testing methods, some operate in a limited number of stages or rounds.

Group testing was first studied in screening large populations for disease [37], and with the advent of large-scale HIV screening, it has had a substantial growth in importance in this area recently. It has also arisen in satellite communications [33, 48]. In this application, a large number of ground stations which rarely communicate share a satellite link. Rather than polling the ground stations individually, pools of the ground stations are formed as part of the system design. When the satellite enters a phase of accepting requests for reservations of time slots, it polls each pool and from the positive results on the pools it determines which ground stations wish to transmit. In this scenario, the satellite may have many positive responses within one pool, but detects only that there is at least one response. Hence, while cosmetically similar to the optical communication situation, this problem encounters unions rather than sums of colliding signals.

Another primary application arises in mapping genomes. In the human genome project, for example, information about long strings of genetic material is obtained by first forming a library of subsegments (*clones*) of the material. To determine where a particular sequence is located within the genetic material, one conducts a test to determine which of the clones it appears in. Pooling of different clones can be used [30, 31, 34].

A further application arises in the construction of frameproof codes, which are designed to avoid coalitions of users forging the signature of a user not in the coalition; see [45, 46].

3.2 The Connection to Designs

Let $\mathcal{P}$ be a set of b items, and let $\mathcal{X}$ be a collection of subsets of $\mathcal{P}$ corresponding to the group tests performed. Then $(\mathcal{P}, \mathcal{X})$ is a solution to the nonadaptive group testing problem if and only if, for any possible sets D_1 and D_2 of defective items, $\{X : D_1 \cap X \neq \emptyset,\ X \in \mathcal{X}\} = \{X : D_2 \cap X \neq \emptyset,\ X \in \mathcal{X}\}$ only if $D_1 = D_2$.

The dual of a solution $(\mathcal{P}, \mathcal{X})$ is a pair $(V, \mathcal{B})$, where the v group tests of $\mathcal{X}$ are in one-to-one correspondence with the points of V, and the b items are in correspondence with the blocks of $\mathcal{B}$ (for each item, the corresponding block contains the elements corresponding to the group tests containing the item). Typically $(V, \mathcal{B})$ is referred to as a solution to the group testing problem; the goal is to maximize the number of blocks (items tested) as a function of the number of points (group tests performed).

Often one knows with high probability that the number of defectives d does not exceed some threshold value p. In the *hypergeometric* problem, one assumes that the number of defectives never exceeds p, and hence one requires that $(V, \mathcal{B})$ has the union of any two distinct sets, each containing at most p

blocks, themselves distinct. In the *strict* problem, one is required to correctly identify the set of defective items when $d \leq p$ and is also required to report when $d > p$. In the latter case, the specific set of defective items need not be determined, however.

Now consider a solution $(V, \mathcal{B})$ to the nonadaptive group testing problem with d defectives. Form a $|V| \times |\mathcal{B}|$ incidence matrix. This matrix has the property that the unions of two sets of at most d columns are distinct. The matrix is then called $\overline{d}$-*separable* [38], and the corresponding set system is d-*union free* [40, 41]. The columns of a $\overline{d}$-separable matrix form a superimposed code [39, 42] which permits up to d simultaneously transmitted codewords to be unambiguously decoded. The decoding technique appears somewhat involved, because one could in principle be required to examine all unions of up to d columns. Hence a related family of matrices (or codes, or set systems) arises. If the incidence matrix contains no collection of d columns whose union covers a column not in the collection, then M is a d-*disjunct* matrix. If a disjunct matrix is employed, one has a simple decoding mechanism by observing that all codewords covered by the received union are 'positive'. Equivalently, one can alter the condition on the set system to require that it is d-*cover free*, i.e. that no union of d or fewer blocks contains another. Evidently, a d-cover free family is also d-union free.

Probabilistic bounds on the maximum numbers of blocks in cover free and union free families are available [38]; see [39, 44] for upper bounds for cover free families, and [43] for lower bound. See [36] for recent progress in the union free case. Erdös, Frankl, and Füredi [40] established that among cover free families with constant block size, the maximum is realized by a Steiner t-design (these have parameters $S(\ell, 2\ell - 1, m)$); indeed Balding and Torney [31] recommend the use of an S(3,5,65) in a genetic application. For union free families, Frankl and Füredi [41] note that Steiner triple systems give the largest 2-union free families when the block size is three; by permitting block size *at most* three, Vakil and Parnes [47] established a somewhat larger exact bound using group divisible designs with block size three.

In the *error correction* version of group testing, some group tests are permitted to report "false positives"; an *a priori* bound q on the number of such false positives is assumed. Balding and Torney [30] observe that $(V, \mathcal{B})$ is a solution to the strict group testing problem with threshold p and error correction for q false positives if and only if, for every union of p or fewer blocks, every other block contains at least $q+1$ points not in this union. Any packing $(V, \mathcal{B})$ of t-sets into k-sets having $k \geq p(t-1) + q + 1$ is a solution to the strict group testing problem with threshold p and error correction for q false positives. A Steiner system $S(t, 2t-1, v)$ is a solution to the strict group testing problem with $p = 2$ and $q = 0$ that has the maximum number of blocks of any solution [30].

Finally, we consider the use of combinatorial designs in two stage group testing. Here the objective in a first stage of pools is not to identify all de-

fectives precisely, but rather to identify a small subset of the items which is guaranteed to contain all defective items. Frankl and Füredi call a family of sets *d-weakly union free* if, whenever two *disjoint* sets of blocks are chosen, each containing d or fewer blocks, their unions are distinct. A 2-weakly union free family with block size three provides pools for a group testing method for $d = 2$, in which a set of at most three potential defectives are identified [35]. Moreover, while union free families have no more blocks than a Steiner triple system, weakly union free families can have twice as many blocks [41]. Chee, Colbourn, and Ling [35] establish that certain twofold triple systems realize the bound. Not any twofold triple system forms a weakly union free family; four forbidden configurations of four blocks each must be avoided. Again, while the bound of Frankl and Füredi [41] suggests that designs can realize the maximum, the particular designs needed require additional structural properties [35]. Applications of designs in general in two stage group testing appear to be just being explored; see [32] for useful observations.

4 Erasure Codes and Information Dispersal

Reliability is a major concern in the design of large disk arrays. Hellerstein et al. [57] examine erasure-resilient codes that allow one to reconstruct the original data even in the presence of disk failures. A set systems view of the problem of constructing erasure-resilient codes leads to interesting extremal problems. Solutions to some of these problems are characterized by well-known combinatorial designs. In other instances, combinatorial designs give asymptotically exact solutions.

4.1 The Application

Over the last decade, there has been a sustained exponential advance in the density and performance of semiconductor technology. With this progress came faster microprocessors as well as larger and faster primary memory devices. Improvements in secondary storage systems, on the other hand, have not kept pace. While the performance of RISC microprocessors has been increasing by more than 50% per year [60], disk transfer rates, which depend on the speed of mechanical movements and magnetic media densities, have only improved by about 20% each year [53]. This phenomenon has transformed many computationally-bound applications to being I/O-bound. Indeed, Amdahl [51] predicted about three decades ago that, unless accompanied by corresponding increases in secondary storage performance, big increases in microprocessor performance can only bring about marginal improvements in overall system performance. This disparity has led to the consideration of parallelism as a means to speed up secondary storage systems. Several ideas have been proposed as to how parallelism can be exploited. The most important and successful is the *disk array architecture*.

The *disk array architecture* organizes many independent small disks into one large logical disk. Small disks are preferable to large ones because they have a lower cost and consume less power. For improved performance, disk arrays employ the concept of *data striping* (see §12) which spreads data to multiple disks. This allows both single and multiple I/O requests to be processed in parallel by separate disks, thus improving effective transfer rates. A further advantage of disk striping is uniform load balance. The more disks we have in a disk array, the higher the performance we obtain. Unfortunately, large disk arrays have low reliability. For low disk failure rates, the failure rate of a disk array is directly proportional to the number of disks it contains. Many applications, notably database and transaction processing systems, require both high throughput and high data availability of their storage systems. The most demanding of these applications require continuous operation, which in terms of a storage system requires the ability to satisfy all requests for data even in the presence of disk failures, and the ability to reconstruct the content of a failed disk onto a replacement disk, thereby restoring itself to a fault-free state. These require redundancy to tolerate disk failures. Disk arrays which incorporate redundancy have come to be known as *Redundant Arrays of Independent Disks* (RAID).

There are three primary types of disk failures. The first, *transient errors*, arise from noise corruption and are dealt with by repeating the requests. The second, *media defects*, are caused by permanent defects in material, and are detected and masked by the manufacturer. The last are *catastrophic failures*, such as head crashes and failures of the disk controller electronics. When a disk suffers a catastrophic failure, its data is rendered unreadable, and is effectively erased. We therefore call such a disk failure an *erasure*. For convenience, we also call a set of k disk failures *a k-erasure*. Error-correcting codes can be used to correct erasures. However, components in disk arrays allow us to determine exactly where erasures have occurred (this distinction between errors and erasures was apparently first drawn by Elias [54]). It is possible to take advantage of this additional information to derive codes that are better than those based on error-correcting codes.

Hellerstein et al. [57] pioneered the study of erasure-resilient codes for large disk arrays. Earlier, Rabin [61] had investigated erasure-resilient codes for information dispersal, but his codes are not particularly suited for disk array applications. Alon et al. [50] have also studied erasure-resilient codes to combat bursty losses in packet-switched networks. The parameters of interest there are also different from those for disk arrays.

4.2 The Connection to Designs

Let $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$. The *weight* of $\mathbf{x}$, denoted $\mathrm{wt}(\mathbf{x})$, is the number $\sum_{i=1}^{n} x_i$. The *support* of $\mathbf{x}$, denoted $\mathrm{supp}(\mathbf{x})$, is the set $\{i \mid x_i = 1\}$.

A *data stripe*, or simply *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 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 (disjoint) stripes.

An $[n, c, k]$-*erasure-resilient code*, or briefly an $[n, c, k]$-*ERC*, consists of an encoding algorithm $\mathcal{E}$ and a decoding algorithm $\mathcal{D}$ with the following properties. Given an n-tuple S of stripes, $\mathcal{E}$ produces an $(n + c)$-tuple or *codeword* $\mathcal{E}(S) = (\mathcal{E}_1(S), \ldots, \mathcal{E}_{n+c}(S))$ of stripes such that for any $I \subseteq \{1, \ldots, n\}$, where $|I| = n + c - k$, the decoding algorithm $\mathcal{D}$ is able to recover S from $(I, \{\mathcal{E}_i(S) \mid i \in I\})$. We often call an $[n, c, k]$-ERC a k-ERC when the parameters n and c are not important in the context.

To see the relevance of an $[n, c, k]$-ERC to the protection of data loss in a RAID, suppose that we have a piece of data which is partitioned into an n-tuple S of stripes. Given an $[n, c, k]$-ERC, we encode S into a codeword $(\mathcal{E}_1(S), \ldots, \mathcal{E}_{n+c}(S))$, and for $1 \leq i \leq n + c$, store $\mathcal{E}_i(S)$ on disk i of a disk array with $n + c$ disks. The definition of an $[n, c, k]$-ERC ensures that we can reconstruct the original data in the presence of up to k erasures.

For performance reasons, the erasure-resilient codes studied are assumed to satisfy two conditions:

1. We restrict ourselves to *systematic* codes. An $[n, c, k]$-ERC is *systematic* if $\mathcal{E}_i(S) = S_i$, for $1 \leq i \leq n$, where $S = (S_1, \ldots, S_n)$. The stripes $E_i(S)$, for $n < i \leq n + c$, are called *checks*. This means that the encoding function leaves the data unmodified on some disks. This property is desirable to avoid read penalties associated with decoding when there are no disk failures.

2. We restrict ourselves to *linear* codes over the field GF_{2^L}, where L is the bit-size of a stripe. In this case, we interpret a stripe as an L-dimensional vector over GF_2, and $\mathcal{E}$ is a linear function. Hence, computations used to encode a stripe are restricted to component-wise modulo two arithmetic, that is, the parity operation $\oplus$. This restriction ensures that encodings and manipulations can be performed efficiently.

Restriction (i) above allows us to separate disks into *information disks*, which contain the original data, and *check disks*, which contain the checks. In fact, restrictions (i) and (ii) imply that an $[n, c, k]$-ERC 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 [7]. The matrix H is called the *parity-check matrix* of the code. Given the parity-check matrix $H = [C \mid I]$ of a k-ERC, 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.

The following are some metrics of an erasure-resilient code that are important for disk arrays.

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

Update penalty: This is the number of check disks whose content must be changed when an update is made in the content of a given information disk. We call these disks the *disks associated with the information disk*. If m check disks need to be involved in every write, then the parallelism of the disk array is reduced by a factor of $m + 1$. 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 with parity-check matrix $H = [C \mid I]$ are the column sums of C.

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 row sums of its parity-check matrix.

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

Suppose $H = [C \mid I]$ has a set of k or fewer linearly dependent columns (over GF_2). The failure of the corresponding disks makes reconstruction of data impossible. In fact, this is the only situation in which disk failures are irrecoverable [57]. It follows that H is the parity-check matrix of a k-ERC 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 [7].

This equivalence between k-ERCs 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 focused on the same metrics. Indeed, as observed in [57], many of these codes are not suitable for disk array applications because they have large update penalties. If an erasure-resilient code is able to correct all 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-ERC are at least k. In view of the importance of minimizing update penalties, we consider from here on only those k-ERCs 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.

A $(k+1)$-erasure is irrecoverable if it corresponds to the failure of an information disk and its k associated check disks. We call such $(k+1)$-erasures *bad*. With update penalty k, one can nonetheless hope to correct *all* $(k+1)$-erasures, except for bad ones [57]. In fact, it can happen that all t-erasures for some $t > k$ are recoverable except for those that contain bad $(k+1)$-erasures. A t-erasure, $t \geq k+1$, is *bad* if it includes the failure of an information disk and all of its k associated check disks. We extend the definition of ERCs to encompass this notion of higher resilience. An $[n, c, k, \ell]$-*ERC* is an $[n, c, k]$-ERC which can correct all t-erasures, for $k+1 \leq t \leq \ell$, except for bad t-erasures.

An alternative view of an $[n, c, k, \ell]$-ERC is that it is an erasure-resilient code with update penalty k that is able to correct all t-erasures, $t \leq \ell$, except bad ones. We often write (k, ℓ)-ERC for $[n, c, k, \ell]$-ERC when the parameters n and c are not important in the context. Requirements for higher reliability of disk arrays make (k, ℓ)-ERCs attractive. A (k, k)-ERC is simply a k-ERC.

Lemma 4.1 *[52] $H = [C \mid I]$ is the parity-check matrix of a (k, ℓ)-ERC if and only if for every t columns, $\mathbf{c}_1, \ldots, \mathbf{c}_t$ of C, where $2 \leq t \leq \ell$, the vector $\mathbf{x} = \bigoplus_{i=1}^{t} \mathbf{c}_i$ has weight at least $\ell + 1 - t$.*

Given c, k, and ℓ, define $F(c, k, \ell)$ to be the maximum n such that there exists an $[n, c, k, \ell]$-ERC. The maximum number of information disks that can be supported by c check disks is $F(c, k, \ell)$, if one desires an update penalty of k and wants to correct all t-erasures, $t \leq \ell$, except bad ones. The important problem is: For given k and ℓ, determine the behavior of $F(c, k, \ell)$ with respect to c; and construct $[n, c, k, \ell]$-ERCs having n as close to $F(c, k, \ell)$ as possible. An $[n, c, k, \ell]$-ERC with $n = F(c, k, \ell)$ is said to have *optimal check disk overhead*. We also abbreviate $F(c, k, k)$ to $F(c, k)$.

A set system $(X, \mathcal{A})$ *contains* a configuration $(Y, \mathcal{B})$ if there exists $Z \subseteq X$ and $\mathcal{C} \subseteq \mathcal{A}$ such that $(Z, \mathcal{C})$ is isomorphic to $(Y, \mathcal{B})$. If $(X, \mathcal{A})$ does not contain $(Y, \mathcal{B})$, then $(X, \mathcal{A})$ *avoids* $(Y, \mathcal{B})$. In this case, we also call $(Y, \mathcal{B})$ a *forbidden configuration* of $(X, \mathcal{A})$.

The *symmetric difference* of two sets A and B is denoted $A \Delta B$. A *Túran-type problem* takes the form: Given a family $\mathcal{F}$ of configurations, determine the maximum number of blocks in a (k-uniform) set system of order n that avoids all the configurations in $\mathcal{F}$. We now explain the role of Túran-type problems in the design of erasure-resilient codes.

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

Let $H = [C \mid I]$ be the parity-check matrix of an erasure-resilient code. We also call the set system of C *the set system of the erasure-resilient code*. If $(X, \mathcal{A})$ is the set system of an $[n, c, k, \ell]$-ERC, then $(X, \mathcal{A})$ is k-uniform, $|X| = c$, and $|\mathcal{A}| = n$. Therefore, the check disk overhead is $|X|/|\mathcal{A}|$, and the group sizes are one more than the replication numbers. This correspondence between

set systems and parity-check matrices gives rise to Túran-type problems in erasure-resilient codes.

Lemma 4.2 *[52] $(X, \mathcal{A})$ is the set system of a (k,ℓ)-ERC if and only if for any $2 \le t \le \ell$, there do not exist t blocks $A_1, \ldots, A_t \in \mathcal{A}$ such that $|\Delta_{i=1}^{t} A_i| \le \ell - t$.*

The construction of a (k,ℓ)-ERC with optimal check disk overhead is precisely the Túran-type problem of determining the maximum number of blocks in a set system satisfying the condition of Lemma 4.2.

When considering (k,ℓ)-ERCs, nontrivial cases arise only when $\ell \le 2k-1$. To see this, let $(X, \mathcal{A})$ be the set system of a (k,ℓ)-ERC. If $\mathcal{A}$ contains two blocks A and A' with nonempty intersection, then $|A \Delta A'| \le 2k-2$. By Lemma 4.2, $\ell - 2 < 2k-2$, and so $\ell \le 2k-1$. Hence if $\ell \ge 2k$, then $\mathcal{A}$ must consist of pairwise disjoint blocks.

Chee, Colbourn, and Ling [52] establish that, for general k and ℓ with $1 \le k \le \ell$, there exist positive constants a_1 and a_2 such that

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

for c a positive integer. The upper bound arises from the easy observation that the set system of a (k,ℓ)-ERC is a packing of strength $t = k+1-\lfloor \frac{\ell}{2} \rfloor$. Indeed, every packing with strength $t=2$ and block size k underlies a k-ERC [57], but the same statement does not extend to strength $t=3$, no matter how large the block size is [52].

The stronger connection with designs arises for particular small values of k. The set system of a 2-ERC is a graph, and indeed any graph without multiple edges suffices; hence $F(c,2) = \binom{c}{2}$ using the complete graph. For such a 2-ERC to be a (2,3)-ERC, the graph cannot contain any triangles and hence the determination of $F(c,2,3)$ is the same as the maximum number of edges in a triangle-free graph on c vertices. This maximum is well known to be $\lfloor \frac{c}{2} \rfloor \cdot \lceil \frac{c}{2} \rceil$.

When $k=3$, every set of triples forms a set system of a 3-ERC and hence $F(c,3) = \binom{c}{3}$. However, the case of (3,4)- and (3,5)-ERCs are more interesting. The set systems associated with such ERCs are packings of strength two, and some classes of Steiner triple systems such as those arising from affine spaces do yield ERCs [57]. Chee, Colbourn, and Ling [52] noted that a 2-packing on c points underlies a (3,4)- or (3,5)-ERC if and only if it does not contain a *quadrilateral*, or *Pasch configuration*; this configuration consists of four blocks isomorphic to $\{\{a,c,e\},\{a,d,f\},\{b,c,f\},\{b,d,e\}\}$. The existence of Steiner triple systems which are quadrilateral-free has been the subject of extensive study; see Ling et al. [59] for recent results and for a statement of the current situation.

For (4,ℓ)-ERCs with $4 \le \ell \le 7$, further classes of designs arise. When $\ell \in \{4,5\}$, it appears that Steiner quadruple systems avoiding a large number of small configurations *might* exist, to reach the upper bound $F(c,4,\ell) \le \frac{c(c-1)(c-2)}{24}$, but at the present time the existence of such SQSs remains open

(although the growth rate is known to be $O(c^3)$ when $\ell \in \{4,5\}$ [52]). When $\ell \in \{6,7\}$, block designs of block size four and index one arise. In this case, numerous small examples of suitable designs establish certain cases in which $F(c,4,7) = \frac{c(c-1)}{12}$ [52]. Although one recursive construction is available, the existence question for such designs remains far from solved.

Related questions arise as a result of additional requirements in the application. Of particular note is the desire for scalability, such as bringing more disks online as they become available. In practice, this leads to imbalances in the disk load and hence degrades performance. Hence erasure codes whose underlying set systems are *resolvable* are of interest. One challenging question motivated by this is the existence of quadrilateral-free resolvable Steiner triple systems; currently, at least one third of the relevant cases are settled affirmatively [52].

The study of erasure codes suggests another major reason for the study of designs (and packings) with forbidden configurations.

5 Threshold and Ramp Schemes

The idea of distributed trust mechanisms is pervasive in information security and cryptography. This concept is realized using threshold schemes and various generalizations, such as secret sharing schemes for general access structures, and ramp schemes, to name two examples.

5.1 The Application

We begin by discussing threshold schemes, which were invented independently by Blakley [62] and Shamir [69]. Let t and w be positive integers such that $t \leq w$. Informally, a (t,w)-*threshold scheme* is a method of splitting a *secret* into w shares, in such a way that the secret can be reconstructed from any t of the w shares, but no information about the secret is revealed by any $t-1$ shares. The value t is the *threshold* of the scheme. The secret is taken to be an element of a specified finite set, $\mathcal{K}$, and each share is an element of a finite set $\mathcal{S}$.

A (t,t)-threshold scheme can easily be constructed by taking $\mathcal{K} = \mathcal{S} = G$, where G is an additive abelian group. In this scheme, the secret is just the sum of the t shares.

Threshold schemes can be used in many common situations in which distributed trust is desirable. For example, the secret could be an access code for a secure area, a PIN or password to enable a specified action, or a cryptographic key to be used for either signing or encrypting data. In many scenarios, it is not prudent to trust any one individual with sensitive information, and hence it is preferable that the secret information can be obtained only through the co-operation of a specified threshold of t out of w people. For example, access

to a bank vault may require two out of three vice presidents; a $(2,3)$-threshold scheme could be used to set up an access mechanism of this type.

A threshold scheme specifies the method by which the secret is split up into shares (a *share generation algorithm*) and the method used to determine the secret given t shares (a *share reconstruction algorithm*). The share reconstruction could be done either by the holders of the shares themselves, or, alternatively, by a trusted machine which is given the values of the shares (and possibly the identities of the individuals holding the shares). The second situation may be more desirable in many applications because it permits the secret to be "re-used" (provided that the value of the secret is not revealed by the machine that computes it, or by any subsequent action that is enabled as a result of the secret being reconstructed).

Threshold schemes are also commonly used as components in other cryptographic protocols. We mention a couple of illustrative examples. Our first example concerns key escrow and key recovery. The idea of *key escrow* is to split a cryptographic key into shares, typically using a $(2,2)$-threshold scheme, which are stored securely in different locations. If at some later time, the individual possessing the key is suspected of criminal activity, then a law enforcement agency could obtain the shares, reconstruct the key, and decrypt any communications of the suspected individual. The use of the threshold scheme makes it difficult for someone to obtain the key who is not authorized to do so. (The entire idea of key escrow is very controversial: Law enforcement agencies and governments are strongly in favor of the idea, whereas most industry and civil liberties organizations are strongly opposed to it.) The idea of *key recovery* is similar and also involves splitting a secret key into shares using a threshold scheme. However, the motivation is different — a key recovery system is usually implemented by an individual or company in order to be able to recover (its own) lost keys. For example, it could be disastrous if keys that are used to store encrypted data are lost, and thus a key recovery scheme is essential in such a circumstance.

Another application of $(2,2)$-threshold schemes is in *electronic cash*, in order to prevent *double spending*. Various schemes have been proposed in which the electronic cash is anonymous (much like regular cash). In contrast to a debit card, the cash is not withdrawn at the time of purchase, and it is not necessary to know the identity of the person spending the cash. Since electronic cash is nothing more than information, the problem of double spending arises: How we do prevent someone from spending the same electronic coin twice? One solution to this problem uses threshold schemes. When the coin is spent, a share of a $(2,2)$-threshold scheme is revealed. If the coin is spent twice, then the two shares revealed can be used to determine the identity of the person holding the coin. Thus anonymity is maintained as long as the coin is not spent twice.

Sometimes it is not necessary to have a "sharp" threshold, t, in securing a secret. This motivates the idea of a ramp scheme, introduced by Blakley and

Meadows [63]. Let t_0, t_1 and w be non-negative integers such that $t_0 < t_1 \leq w$. A (t_0, t_1, w)-*ramp scheme* is a method of splitting a secret into w shares, in such a way that the secret can be uniquely reconstructed from any t_1 of the w shares, but no information about the secret is revealed by any t_0 shares. Thus a (t, w) threshold scheme is nothing more than a $(t-1, t, w)$-ramp scheme. The reason for the term "ramp scheme" is as follows. Typically, as the number of shares is increased from t_0 to t_1, the amount of information about the secret gradually increases, as well.

In a $(0, t_1, w)$-ramp scheme, there is no security requirement. The resulting scheme is the same as an information dispersal algorithm (see §4). Thus, ramp schemes interpolate between information dispersal schemes and threshold schemes.

So far, we have regarded the secret as an element of an arbitrary finite set, $\mathcal{S}$. Naor and Shamir introduced the appealing idea of *visual cryptography* in [68], where the secret and each share consists of a collection of black and white (transparent) pixels on a transparency. The secret reconstruction algorithm in this scheme consists of stacking the transparencies, so the "computation" is performed by the human visual system!

5.2 The Connection to Designs

Threshold schemes are closely related to orthogonal arrays. Suppose that M is an $\mathrm{OA}_1(t, w+1, v)$ on alphabet A. Thus M is a $(w+1) \times v^{t+1}$ array in which any t rows contain any t-tuple of symbols in exactly one column. M can be used to construct a (t, w)-threshold scheme in which $\mathcal{S} = \mathcal{K} = A$. Label the rows of M by the elements in $\{1, \ldots, w+1\}$. Each column r of M corresponds to a *distribution rule* as follows: The element $M(w+1, r)$ is the secret, and the w shares are the elements $M(i, r)$, $1 \leq i \leq w$.

The share generation algorithm works as follows. Given a secret $K \in \mathcal{K}$, a random distribution rule r is chosen such that $M(w+1, r) = K$ (there are v^{t-1} such rules to choose from, given K). Then the chosen rule r is used to determine the w shares.

The share reconstruction algorithm is straightforward. Suppose we are given the values of t shares, say $s_{i_j} = a_j$, $1 \leq j \leq t$, where $1 \leq i_1 < \ldots < i_t \leq w$. There is a unique column r of the orthogonal array such that a_j appears in row i_j for all j, $1 \leq j \leq t$. Hence, $M(w+1, r)$ is revealed as the secret.

It is also easy to see that a list of $t-1$ shares leaves the secret completely undetermined. In fact, given any possible guess K_0 as to the value of the secret, there is exactly one column r of the orthogonal array that is consistent with the $t-1$ given shares and such that $M(w+1, r) = K_0$.

It can be proved easily that $|\mathcal{S}| \geq |\mathcal{K}|$ in any (t, w)-threshold scheme. The construction described above provides schemes in which this bound is met with equality, i.e., in which the shares are as small as possible. A result of Martin [67] shows that the converse also holds:

Theorem 5.1 *There exists a (t, w)-threshold scheme in which $|\mathcal{S}| = |\mathcal{K}| = v$ if and only if there exists an $OA_1(t, w+1, v)$.*

We stated above that $|\mathcal{S}| \geq |\mathcal{K}|$ in any (t, w)-threshold scheme. This result can be easily generalized to ramp schemes, as follows:

Theorem 5.2 $|\mathcal{K}| \leq |\mathcal{S}|^{t_1 - t_0}$ *in any (t_0, t_1, w)-ramp scheme.*

Ramp schemes meeting this bound can also be constructed using orthogonal arrays:

Theorem 5.3 *If there exists an $OA_1(t, w + t_1 - t_0, v)$, then there exists a (t_0, t_1, w)-ramp scheme in which $|\mathcal{S}| = v$ and $|\mathcal{K}| = v^{t_1 - t_0}$.*

These and other results on ramp schemes can be found in [66].

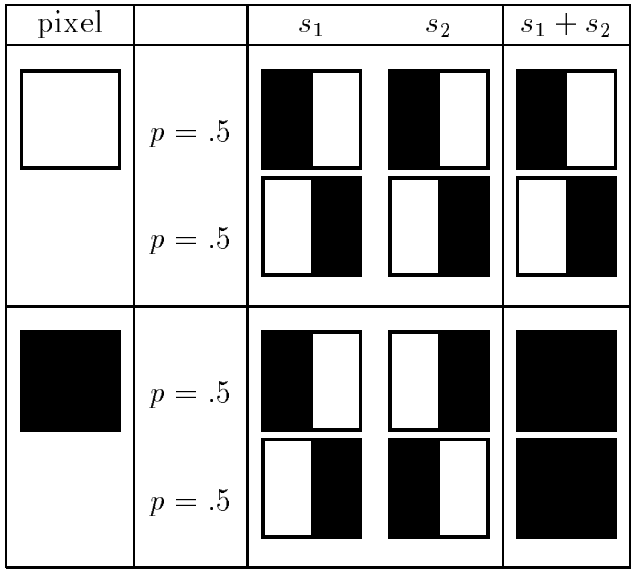
The orthogonal array-based schemes described above require knowing which share is held by which participant in order to reconstruct the secret. This is because the share reconstruction algorithm needs to be given the t relevant rows of the orthogonal array (which correspond to the identities of the t participants reconstructing the secret). Stinson and Vanstone [71] considered *anonymous* schemes, in which the values of the shares alone are sufficient to reconstruct the secret. They gave a construction for anonymous schemes based on partitionable Steiner systems. A Steiner system $\mathrm{S}(t, w, v)$ is *partitionable* if its block set can be partitioned into $N = (v - t + 1)/(w - t + 1)$ Steiner systems $\mathrm{S}(t - 1, w, v)$. Examples of partitionable Steiner systems include resolvable 2-designs (the case $t = 2$), and large sets of Steiner triple systems (the case $t = w = 3$).

A partitionable Steiner system can be used to construct an anonymous (t, w)-threshold scheme, as follows. The systems $\mathrm{S}(t - 1, w, v)$ are named $\mathcal{D}_i$, $1 \leq i \leq N$, the set of secrets is $\{1, \ldots, N\}$, and the set of shares is the set of points of the designs. If the secret to be shared is i_0, then a random block B of the system $\mathcal{D}_{i_0}$ is chosen, and the w shares are the w points in B. Any t shares determine B uniquely, and hence the secret is determined because i_0 is the unique value i such that B is a block in $\mathcal{D}_i$. On the other hand, if $t - 1$ shares are given, then, for every i, there is a unique block in $\mathcal{D}_i$ that contains the $t - 1$ shares. Thus the secret is completely undetermined.

Visual cryptography provides another example of anonymous schemes, since the reconstruction algorithm consists simply of stacking transparencies, and does not make use of the identities of the participants in the scheme. Realizing a visual cryptography scheme requires expanding each pixel into some number $m \geq 2$ of subpixels in each of the w shares. A reconstructed pixel (obtained by superimposing t transparencies) is not necessarily totally black or white. In general, there is a loss of contrast, which is measured by the difference in darkness between a reconstructed black and white pixel. Our goal is to maximize the contrast while minimizing the pixel expansion.

To illustrate, we describe the construction of the visual $(2,2)$-threshold scheme presented at EUROCRYPT '94 by Naor and Shamir (see [68]). Figure 1 illustrates the scheme, by specifying the algorithm for encoding one pixel. (This algorithm is to be applied for every pixel P in the image I in order to construct the two shares.) A pixel P is split into $m = 2$ subpixels in each of the two shares. If the given pixel P is white, then a random choice of one of the first two rows of Figure 1 is made. If the given pixel P is black, then a random choice of one of the last two rows of Figure 1 is made. Then the pixel P is encrypted as two subpixels in each of the two shares, as determined by the chosen row in Figure 1.

Figure 1: A 2-out-of-2 visual threshold scheme



Let's convince ourselves that the scheme works as desired. First, we consider the security condition. Suppose we turn our attention to a pixel P in the share s_1. One of the two subpixels in P is black and the other is white. Moreover, each of the two possibilities black-white and white-black is equally likely to occur, independent of whether the corresponding pixel in the secret image I is black or white. Thus the share s_1 gives no clue as to whether the pixel is black or white. The same argument applies to the share s_2. Since all the pixels in I were encrypted using independent random coin flips, there is no information to be gained by looking at any group of pixels on a share, either. This demonstrates the security of the scheme.

Now let's consider what happens when we superimpose the two shares (here we refer to the last column of Figure 1). Consider one pixel P in the image I. If P is black, then we get two black subpixels when we superimpose the two shares; if P is white, then we get one black subpixel and one white

subpixel when we superimpose the two shares. Thus, we could say that the reconstructed pixel (consisting of two subpixels) has a grey level of 1 if P is black, and a grey level of $1/2$ if P is white. There is a 50% loss of contrast in the reconstructed image, but it should still be visible.

In the above scheme, we say that the contrast is $\gamma = .5$. It was shown in [64] that in any visual $(2, w)$-threshold scheme, it holds that $\gamma \leq \gamma^*(w)$, where

$$\gamma^*(w) = \frac{\left\lceil \frac{w}{2} \right\rceil \left\lfloor \frac{w}{2} \right\rfloor}{w(w-1)}.$$

If we examine the behavior of the function $\gamma^*(w)$, we see that $\gamma^*(w) > 1/4$ for all $w \geq 2$, and $\lim_{w\to\infty} \gamma^*(w) = 1/4$. This raises the question if schemes can be constructed for all $w \geq 2$ which achieve relative contrast $\gamma^*(w)$. This is in fact possible [64]. Since the contrast of these schemes is always at least $1/4$, this means that the loss of contrast is at most 75%.

Several results on this problem are proved in [64] and [65] (also, see [70] for an elementary treatment). Here is an example of one of the results shown that involves designs.

Theorem 5.4 *Suppose $w \equiv 3 \bmod 4$ and there exists a visual $(2, w)$-threshold scheme with pixel expansion m and (optimal) relative difference $\gamma = \gamma^*(w)$. Then $m \geq w$, and $m = w$ if and only if there exists a $\left(w, \frac{w-1}{2}, \frac{w-3}{4}\right)$-BIBD (or, equivalently, a Hadamard matrix of order $w + 1$).*

Here is a outline of how the design is used to obtain the desired scheme. Suppose first that the pixel P to be encrypted is black. Let I_1 denote the incidence matrix of the BIBD. A random permutation of the columns is first applied to I_1. Then the w rows of the resulting matrix are used to construct the w shares of P (each share consists of w subpixels).

If P is white, then we instead begin with a matrix I_0 in which each row consists of $(w-1)/2$ 1's followed by $(w+1)/2$ 0's. As in the previous case, a random column permutation is performed, and then the rows of the resulting matrix determine the w shares.

6 Authentication Codes

Unconditionally secure authentication codes were introduced in 1974 by Gilbert, MacWilliams and Sloane [72], and Simmons developed a comprehensive theory of authentication codes in the 1980's (see, for example, [76]).

6.1 The Application

We begin with a brief motivation of this topic. Two fundamental goals of cryptography are to ensure secrecy and integrity of sensitive data. Secrecy is achieved through encryption, while data integrity is accomplished by means

such as signature schemes and message authentication codes. *Data integrity* provides a way to verify that data has not been changed by an adversary. The data in question could be communicated between two parties over an insecure network (email, for example), or it could be stored data (in a database, for example). Transmitted data is typically authenticated using a *signature scheme*, which allows anyone to verify an electronic signature using a public verification algorithm. The security of signature schemes depend on the assumed computational intractability of problems such as the discrete logarithm problem.

An alternative is to use a *message authentication code* (or MAC), which requires the used of a secret key to authenticate the data. As is the case when using a private-key cryptosystem, such as the Data Encryption Standard (DES), this key must be communicated to the intended receiver ahead of time, using a secure channel. A typical example of a MAC is constructed by using DES in *cipher block chaining* (CBC) mode. MACs can also be used to authenticate stored data, in which case the key should be stored in a separate, secure place from the data being authenticated.

The DES-based MAC is fast, but there is no known proof of security, not even one based on a plausible computational assumption. However, it is possible to construct MACs that can be proved secure, without any computational assumptions. This is called *unconditional security* and was introduced by Shannon to prove the security of the well-known one-time pad (which is used for encryption).

The simplest scenario to consider is authentication without secrecy. Here, a plaintext message s (called a *source state*) is authenticated by computing an *authentication tag*, a, which is appended to to s. The tag a is computed as $a = h_K(s)$, where h_K is a hash function chosen from a specified set of hash functions using a random secret key K. An *authenticated message*, (s, a), is verified by computing $a_0 = h_K(s)$ and checking that $a_0 = a$.

Formally, we can describe the authentication code as a 4-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{K}, \mathcal{H})$, where the following hold:

1. $\mathcal{S}$ is a finite set of possible source states;
2. $\mathcal{A}$ is a finite set of possible authentication tags;
3. $\mathcal{K}$ is a finite set of possible keys;
4. $\mathcal{H} = \{h_K : K \in \mathcal{K}\}$, where, for each $K \in \mathcal{K}$, $h_K : \mathcal{S} \to \mathcal{A}$.

As an example, $\mathcal{S}$ and $\mathcal{A}$ could consist of all binary strings of a certain length, i.e., $\mathcal{S} = (Z_2)^n$ and $\mathcal{A} = (Z_2)^m$. There is no requirement on the relative sizes of n and m, but the cases of greatest practical interest are when $n \gg m$, i.e., when we authenticate a "long" source with a "short" authenticator.

The security of an authentication code is measured by computing an opponent's *deception probabilities*. Suppose an opponent, Oscar, sees a sequence

of $i \geq 0$ authenticated messages, all of which are authenticated using the same (unknown) key. Then Oscar creates a new forged message, (s', a'), which he hopes is an authentic message. That is, Oscar wins this game if $s' = h_K(a')$, where K is the secret key. This situation is called a *substitution* of order i. In a one-time scheme (i.e., where a key is used to authenticate only one message), the case $i = 0$ is called *impersonation* and the case $i = 1$ is called *substitution.* Oscar's probability of performing a successful deception of order i is denoted Pd_i.

6.2 The Connection to Designs

The main objectives of an authentication code are to minimize the deception probabilities Pd_i as a function of $|\mathcal{A}|$, and to minimize $|\mathcal{H}|$. Authentication codes in which these quantities are simultaneously minimized turn out to be closely related to designs. The following lemma is found in [77, Theorem 5.1].

Lemma 6.1 *In any authentication code without secrecy, $(\mathcal{S}, \mathcal{A}, \mathcal{K}, \mathcal{H})$, and for any $i \geq 0$, $\mathrm{Pd}_i \geq 1/|\mathcal{A}|$.*

If a key is chosen equiprobably from $\mathcal{K}$, then we have the following characterization shown in [77, Theorems 5.3 and 5.4].

Theorem 6.2 *An authentication code $(\mathcal{S}, \mathcal{A}, \mathcal{K}, \mathcal{H})$ in which keys are chosen equiprobably and in which $\mathrm{Pd}_i = 1/|\mathcal{A}|$ for $0 \leq i \leq t-1$ is equivalent to an orthogonal array $OA_\lambda(t, |\mathcal{S}|, |\mathcal{A}|)$, where $\lambda = |\mathcal{H}|/|\mathcal{A}|^t$.*

The construction of an authentication code form an orthogonal array is easy: Let M be an $\mathrm{OA}_\lambda(t, k, v)$ on symbol set A, say. Suppose the columns of M are labeled $1, \ldots, k$ and the rows are labeled $1, \ldots, \lambda v^t$. Define $\mathcal{S} = \{1, \ldots, k\}$, $\mathcal{A} = A$, $\mathcal{K} = \{1, \ldots, \lambda v^t\}$, and for any $K \in \mathcal{K}$, define $h_K(s) = M(K, s)$ for all $s \in \mathcal{S}$. Then $(\mathcal{S}, \mathcal{A}, \mathcal{K}, \mathcal{H})$ is the desired authentication code.

In view of Theorem 6.2, minimizing $|\mathcal{H}|$ is equivalent to finding the minimum value λ in the corresponding $\mathrm{OA}_\lambda(t, k, v)$. Various classical bounds can be employed, e.g., the Rao and Bush bounds. As stated previously, the situations of greatest practical interest occur when $k \gg v$, and an orthogonal array with $\lambda = 1$ does not exist in these cases. Hence it is often necessary to use OAs with $\lambda > 1$.

The results above all assume that keys are chosen equiprobably. Various theorems have been proven which show that there is no advantage in choosing keys using a different probability distribution; the optimal codes are still derived from orthogonal arrays. Here are two results of this type in the case $t = 2$.

Theorem 6.3 *[77] An authentication code, $(\mathcal{S}, \mathcal{A}, \mathcal{K}, \mathcal{H})$, in which $\mathrm{Pd}_i = 1/|\mathcal{A}|$ for $i = 0, 1$, must have $|\mathcal{H}| \geq |\mathcal{A}|^2$. Further, if $|\mathcal{H}| = |\mathcal{A}|^2$, then there exists an orthogonal array $OA_1(2, |\mathcal{S}|, |\mathcal{A}|)$.*

Theorem 6.4 *[78] An authentication code, $(\mathcal{S}, \mathcal{A}, \mathcal{K}, \mathcal{H})$, in which $\mathrm{Pd}_i = 1/|\mathcal{A}|$ for $i = 0, 1$, must have $|\mathcal{H}| \geq |\mathcal{S}|(|\mathcal{A}| - 1) + 1$. Further, if $|\mathcal{H}| = |\mathcal{S}|(|\mathcal{A}| - 1) + 1|$, then there exists an orthogonal array $OA_\lambda(2, |\mathcal{S}|, |\mathcal{A}|)$ in which $\lambda = (|\mathcal{S}|(|\mathcal{A}| - 1) + 1)/|\mathcal{A}|^2$.*

The orthogonal arrays in Theorem 6.3 correspond to mutually orthogonal Latin squares; the OAs in Theorem 6.4 are those that meet the Rao bound with equality, e.g., simplex codes. Generalizations of these two theorems for higher values of t have been proved in [77, Theorem 5.4] and [73, Theorem 1] respectively.

So far, we have considered authentication codes without secrecy, i.e., in which the source is authenticated with an appended authentication tag. A more general setting is to study deception probabilities when the source is "encrypted". This general setting allows the study of codes which provide secrecy and authentication simultaneously, as well as codes that provide authentication with no requirement with respect to secrecy or lack thereof. There are numerous results on codes of these types, analogous to results stated above for authentication codes without secrecy. Codes which achieve the minimum deception probabilities and which have a minimum possible number of keys are closely related to t-designs. We refer the reader to [73, 74, 75, 77, 78] for further information.

7 Resilient and Correlation-immune Functions

A boolean function exhibits correlation-immunity when, despite having specified the values of a certain number of bits of its input, the output bits retain their *a priori* distribution.

7.1 The Application

Suppose $f : (Z_2)^n \to (Z_2)^m$. We refer to f as a *boolean function* with n inputs and m outputs. For an m-tuple $(y_1, \ldots, y_m) \in (Z_2)^m$, let $\mathsf{p}(y_1, \ldots, y_m)$ denote the probability that $(y_1, \ldots, y_m)$ is the output of f when $(x_1, \ldots, x_n) \in (Z_2)^n$ is chosen randomly. The function f is *balanced* if $\mathsf{p}(y_1, \ldots, y_m) = 2^{-m}$ for every $(y_1, \ldots, y_m) \in (Z_2)^m$. Equivalently, for every output m-tuple $(y_1, \ldots, y_m) \in (Z_2)^m$, there are precisely 2^{n-m} input n-tuples $(x_1, \ldots, x_n) \in (Z_2)^n$ such that $f(x_1, \ldots, x_n) = (y_1, \ldots, y_m)$.

Suppose that $t \leq n-1$, and the values of t of the inputs are fixed, say $x_{i_j} = c_j$ for $1 \leq j \leq t$, where $c_1, \ldots, c_t \in Z_2$. The remaining $n - t$ inputs are chosen independently at random, as before. Let $\mathsf{p}(y_1, \ldots, y_m | x_{i_1} = c_1, \ldots, x_{i_t} = c_t)$ denote the probability that $(y_1, \ldots, y_m)$ is the output of f in this situation.

The function f is *correlation-immune* of order t if

$$\mathsf{p}(y_1, \ldots, y_m | x_{i_1} = c_1, \ldots, x_{i_t} = c_t) = \mathsf{p}(y_1, \ldots, y_m)$$

for all choices of $y_1, \ldots, y_m$, $x_{i_1}, \ldots, x_{i_t}$, and $c_1, \ldots, c_t$. f is *resilient* of order t if it is balanced and correlation-immune of order t; this is equivalent to saying that

$$\mathsf{p}(y_1, \ldots, y_m | x_{i_1} = c_1, \ldots, x_{i_t} = c_t) = 2^{-m}$$

for all choices of the relevant variables. In most applications, it is important that t be as large as possible (given n and m), or, equivalently, that m is as large as possible (given n and t).

Two examples of resilient functions are as follows. For any n, the function

$$f(x_1, \ldots, x_n) = x_1 + \ldots + x_n$$

is resilient of order $n-1$. Also, for any h, the function

$$f(x_1, \ldots, x_{3h}) = (x_1 + \ldots + x_{2h}, x_{h+1} + \ldots + x_{3h})$$

is resilient of order $2h-1$. (Addition is modulo 2 in both these examples.)

We now describe some cryptographic applications of these concepts. In a stream cipher, a binary sequence of plaintext is exclusive-ored with a pseudorandom keystream in order to obtain ciphertext. A keystream is generated from a small random seed using a deterministic algorithm. One common way to generate a keystream is to combine the outputs of several linear feedback shift registers, which produce keystreams having maximum-length periods but which are not cryptographically secure. The combining function is a boolean function $f : (Z_2)^n \rightarrow (Z_2)^m$ where $m \leq n$ (usually $m = 1$ in this application). Such a combining function should satisfy several properties in order to produce a secure keystream. For example, the function should be balanced and should have high nonlinearity. Siegenthaler [85] suggested that correlation-immunity is another desirable property.

Block ciphers also use boolean functions in their construction. A typical example is a substitution box (or S-box) that is used in Feistel-type ciphers such as the Data Encryption Standard (DES). An S-box can be described using a boolean function as defined above. In DES, for example, there are eight S-boxes used, each of which is a boolean function from $(Z_2)^6$ to $(Z_2)^4$. Among the design criteria satisfied by the DES S-boxes are certain balance and correlation-immune properties.

Another context in which resilient functions are useful is in renewing a partially leaked key. Cryptographic scenarios in which partially leaked keys are studied include quantum key exchange (Bennett, Brassard and Robert [79]) and key distribution patterns (Stinson [87]). In both these situations, resilient functions have been proposed as a useful method of obtaining a smaller secure key from a larger partially leaked key.

Suppose Alice and Bob share a random n-bit binary key, $x_1, \ldots, x_n$, and an opponent, Oscar, has learned the values of t of the n bits. Suppose further that Alice and Bob do not know which t bits have been leaked. If $f : (Z_2)^n \rightarrow (Z_2)^m$ is a t-resilient function, then $f(x_1, \ldots, x_n)$ is a random m-bit key about which

Oscar has no information. (This is true since every output m-tuple is equally likely, given the value of t input variables.) The description of the function f does not need to be kept secret from Oscar.

7.2 The Connection to Designs

Let $f : (Z_2)^n \to (Z_2)^m$. For any $(y_1, \ldots, y_m) \in (Z_2)^m$, $f^{-1}(y_1, \ldots, y_m)$ is a set (possibly empty) of binary n-tuples. We depict $f^{-1}(y_1, \ldots, y_m)$ as a binary array with n columns, in which the rows are the n-tuples in the set $f^{-1}(y_1, \ldots, y_m)$.

The following theorems give the connection between correlation-immune and resilient functions and orthogonal arrays (see [81, 84, 86]).

Theorem 7.1 *A function $f : (Z_2)^n \to (Z_2)^m$ is correlation-immune of order t if and only if each array $f^{-1}(y_1, \ldots, y_m)$ is an orthogonal array of strength t.*

Theorem 7.2 *A function $f : (Z_2)^n \to (Z_2)^m$ is resilient of order t if and only if each array $f^{-1}(y_1, \ldots, y_m)$ is an $OA_{2^{n-m-t}}(t, n, 2)$.*

The difference between these two theorems is that the 2^m orthogonal arrays in Theorem 7.2 all have $\lambda = 2^{n-m-t}$, whereas in Theorem 7.1 the orthogonal arrays have (possibly) different values of λ. In both theorems, the orthogonal arrays are simple and disjoint, and the 2^m arrays form a partition of $(Z_2)^n$. When the OAs in question all have the same λ value, the collection is known as a *large set* of orthogonal arrays and is denoted LOA. The following corollary is immediate.

Corollary 7.3 *[86] There exists a function $f : (Z_2)^n \to (Z_2)^m$ that is resilient of order t if and only if there exists an $LOA_{2^{n-m-t}}(t, n, 2)$.*

In the correlation-immune case, where the λ values need not be identical, the corresponding collection of OAs has been termed a "large class".

Constructions of resilient functions are easily obtained from codes, using the following well-known result.

Proposition 7.4 *Suppose that $\mathcal{C}$ is a binary linear code of length n, dimension m and distance d. Then the dual code, $\mathcal{C}^\perp$, is an $OA_{2^{n-m-d+1}}(d-1, n, 2)$.*

Since $\mathcal{C}^\perp$ is a subspace of $(Z_2)^n$, it is easy to see that every coset of $\mathcal{C}^\perp$ is also an $\mathrm{OA}_{2^{n-m-d+1}}(d-1, n, 2)$, and these 2^m OAs form a large set. Applying Corollary 7.3, we have the following result, shown in [79, 82].

Theorem 7.5 *If there exists a binary linear code of length n, dimension m and distance d, then there exists a function $f : (Z_2)^n \to (Z_2)^m$ that is resilient of order $d-1$.*

Simplex codes provide a good illustration of Theorem 7.5. For any integer $m \geq 2$, the simplex code $\mathcal{S}_m$ is a linear code of length $2^m - 1$, dimension m and distance 2^{m-1} (it is in fact the dual of a Hamming code). The following corollary of Theorem 7.5 is obtained.

Corollary 7.6 *[83] For any integer $m \geq 2$, there exists a function $f : (Z_2)^{2^m-1} \rightarrow (Z_2)^m$ that is resilient of order $2^{m-1} - 1$.*

In order to know how good our constructions are, we need bounds (necessary conditions). The only known bounds for resilient functions are applications of bounds for orthogonal arrays. The two most important bounds follow:

Theorem 7.7 *[83, 80] If there exists a function $f : (Z_2)^n \rightarrow (Z_2)^m$ that is resilient of order t, then*

$$t \leq \left\lfloor \frac{2^{m-1}n}{2^m - 1} \right\rfloor - 1$$

and

$$t \leq 2 \left\lfloor \frac{2^{m-2}(n+1)}{2^{m-1}} \right\rfloor - 1.$$

Applying the first of these two bounds, we see that the resilient functions constructed in Corollary 7.6 are optimal, since the bound on t is met with equality. There are various other infinite classes of optimal resilient functions known. For example, complete results are known for $m = 1, 2$ and 3, as follows.

Theorem 7.8 *[80, 82, 83]*

1. *There exists a function $f : (Z_2)^n \rightarrow (Z_2)^1$ that is resilient of order t if and only if $t \leq n - 1$.*
2. *There exists a function $f : (Z_2)^n \rightarrow (Z_2)^2$ that is resilient of order t if and only if $t \leq \lfloor 2n/3 \rfloor - 1$.*
3. *There exists a function $f : (Z_2)^n \rightarrow (Z_2)^3$ that is resilient of order t if and only if $t \leq \lfloor 4n/7 \rfloor - 1$ and $n \not\equiv 2 \bmod 7$, or $t \leq \lfloor 4n/7 \rfloor - 2$ and $n \equiv 2 \bmod 7$.*

We have discussed correlation-immune and resilient functions defined over Z_2. These concepts can be generalized in an obvious way to non-binary alphabets, and many of the results discussed in this section hold true in a suitably generalized form; see [84].

8 Multidrop Networks

The design of networks using broadcast media so that every two sites lie on a common link, subject to constraints on the number of links at each site (degree), and the number of sites on each link (link size), is examined. This leads to the examination of pairwise balanced designs and to the use of (k, n)-arcs in projective planes.

8.1 The Application

The network design problem of interest is as follows. There are n network sites, to be connected using multidrop communication links such as Ethernets, token rings, or any broadcast medium. A *link* or *bus* is a subset of the n sites. In order to avoid congestion due to switching overhead from one link to another, it is required that every two sites appear together on at least one link. Typically, each site is equipped with a limited number of communication ports and hence can appear on at most some fixed number r of the links. Similarly, each link has a limit on the number of sites that it can connect. Reasons for such a limitation include capacity limits, and limits on acceptable routing delay within the link. With these constraints in mind, the problem can be informally stated as follows: Connect n sites so that every two sites appear together on at least one link, subject to the constraint that no link has more than k sites on it, and no site appears on more than r links.

Problems of this type have been studied extensively. Mickunas [101] considered the case when k and r are close to equal. Subsequently, Bermond and his colleagues [92, 93, 94] considered general network design problems of this type under the name "bus interconnection networks". They are primarily responsible for observing that numerous well studied combinatorial configurations lead to useful solutions to such network design problems; see also [3, 88, 97, 103, 105].

8.2 The Connection to Designs

Block designs and pairwise balanced designs lead to optimal solutions for the network design problem when $k < r$ [92, 93]. When $k = r$, projective planes yield bus networks [101]. Practical concerns dictate that the replication number r be a fixed small number, while the blocksize k can be potentially much larger than r. Since block designs and PBDs always have $k \leq r$ by Fisher's inequality [1], a technique is needed to treat cases when $k > r$. This is one of the problems treated in [104].

Bermond, Bond, Paoli, and Peyrat [92] propose the following. Suppose that we are to construct a covering with replication number at most r and blocksize at most k, and our objective is to maximize the number of elements. Choose q so that q is a power of a prime, $q+1 \leq r$, and q is as large as possible subject to these constraints. Then form $PG(2,q)$ on element set V of size q^2+q+1, and with block set $\mathcal{B}$. A *weight function* $\omega : V \mapsto Z^+$ from elements to positive integers is to be chosen, and a set of elements $W = \{(x,i) : x \in V \text{ and } 1 \leq i \leq \omega(x)\}$ defined. The weight of an element indicates the number of times that it is replicated in W. One chooses ω so that the *weight* of block B, $\omega(B) = \sum_{x \in B} \omega(x) \leq k$, for every $B \in \mathcal{B}$, and defines a new set of blocks

$$\mathcal{D} = \{\{(x,i) : x \in B \text{ and } 1 \leq i \leq \omega(x)\} : B \in \mathcal{B}\}.$$

Then $(W, \mathcal{D})$ is a covering with blocksizes at most k, replication number $q+1 \leq$

r, and $\sum_{x \in V} \omega(x)$ elements. Naturally the problem is to determine ω so as to constrain the weight of each block to k while maximizing the number of elements. Bermond *et al.* [92, 93] conjecture that the covering with replication number at most r, block sizes at most k, and the largest number of elements, arises in this manner when $r-1$ is a prime power. It follows from a theorem of Füredi [98] that such a covering can have at most $rk-(r-1)\lceil \frac{k}{r} \rceil$ elements. Now choosing ω so that all element weights are as equal as possible subject to the constraint on block weight leads to coverings for which Füredi's bound is achieved infinitely often, and approaches this bound as $k \to \infty$ for fixed r when $r-1$ is a prime power [93, 94]. Hence, although there are many potential methods for producing coverings, Füredi's result establishes that the asymptotically optimal coverings arise from replicating elements in projective planes.

Bermond *et al.* [93, 94] do not address the question of finding the largest number of elements in a covering with block size at most k and replication number at most r precisely. Yener, Ofek, and Yung [104], however, employ a similar underlying strategy but develop techniques for specifying the weight of each element in the projective plane so as to maximize the number of elements. Their method is a simple greedy strategy, which does not in general lead to the minimum block size [95].

When $k \leq r$, the existence of bus networks with restricted bus sizes amounts simply to the existence of pairwise balanced designs with specified block sizes. When $k > r$, two more specific problems arise.

The first arises when r is of the form $q+1$ for q a prime power, so that a projective plane of order q exists. Then an apparently hard question is to determine the minimum increase in the maximum blocksize. A (k,n)-*arc* in a projective plane of order q is a nonempty set K of k elements such that n is the maximum number of elements in K that appear together on a block. A $(k,2)$-arc is a k-*arc*. The existence of (k,n)-arcs has been extensively studied, but their importance here is that the maximum number $m_n(q)$ of elements in a (k,n)-arc in $PG(2,q)$ is precisely the same as the maximum number of elements that can be replicated without increasing the maximum blocksize by more than n.

Barlotti [90] established that $m_n(q) \leq (n-1)(q+1)+1$. A simple computation establishes that when $PG(2,q)$ contains a (k,n)-arc whose size meets this bound, replicating the elements of the arc yields equality in the asymptotic bound of Füredi [98] discussed earlier. Unfortunately, the determination of $m_n(q)$ is a very difficult problem in finite geometry that remains far from settled [99, 100]. Except when $n = q+1$, equality in Barlotti's bound can only be achieved when n is a divisor of q [90], and is achieved in two *trivial cases*: when $n = 1$ (by a single element), and when $n = q$ by all elements not lying on a fixed block. When q is a second or higher power of a prime, nontrivial arcs meeting Barlotti's bound *always* exist when q is a power of 2 and n is a divisor of q [96]. However, in a recent breakthrough it has been shown that

they *never* exist when q is a power of an odd prime [89].

When $m_n(q)$ does not realize Barlotti's bound, extensive research has attempted to obtain lower and upper bounds, and specific exact values; as an introduction to the literature, we suggest [99, 100]. Each lower bound can lead to a replication scheme for producing a covering, and each upper bound establishes a limit on how well such a replication scheme can do.

The second main problem arises when r is not one more than a prime power. Then the advisability of beginning with a plane of order less than r is open to question. Colbourn [95] observes that certain related combinatorial configurations can lead to better results. Consider, for example, the case when $r = 7$ and $v = 39$. There is no plane of order 6, and hence the plane $PG(2, 5)$ can be used. Then block size 9 is obtained. However, there is a covering on 39 elements with replication number 7 and blocksize 7 [102], and hence in this case replication of elements in planes does not appear to lead to the best solution. For this reason, we mention a less well studied generalization of difference sets that can lead to (slightly) smaller blocksizes for certain degree constraints.

A *difference cover* modulo v of *order* q, $D = \{d_0, \ldots, d_q\}$, has the property that $\{d_i - d_j : 0 \leq i, j \leq q \text{ and } i \neq j\}$, arithmetic modulo v, contains every nonzero integer in Z_v. Hence every nonzero difference arises at least once as the difference modulo v of two elements in D. When $v = q^2 + q + 1$, a difference cover is a difference set. However, while difference sets only exist for certain values of q, difference covers exist for every value of q. Of course, the price one pays is that the number of elements v is less than $q^2 + q + 1$ in general. Now adding each integer i to the elements of D in turn, we produce v blocks forming a covering with blocksize $q + 1$ and replication number $q + 1$. Wiedemann [102] gives a difference cover modulo 39 of order 6, which provides the illustration given above. He also presents a table of the smallest order difference cover modulo v for each value of $v \leq 133$. Unfortunately, these computational results are not at present accompanied by a useful theory. For certain small replication numbers such as 7 (a difference cover modulo 39), and 11 (a difference cover modulo 95 [102]), it appears that difference covers can improve upon the use of planes.

We have focussed on the design of bus networks with diameter one. However, transversal designs arise also in the design of bus networks of larger diameter; see [91], for example.

9 Channel Graphs and Interconnection Networks

In this section, we examine interconnection strategies for computer networks and the input-output channel graphs that they contain.

9.1 The Application

The generic *interconnection problem* is to establish communication paths between *input nodes* V_I and *output nodes* V_O; we allow the case that the input nodes and output nodes are the same. *Switch nodes* V_S may be used; these simply relay a message from one communications channel to another. Our task is to connect the nodes in $V = V_I \cup V_O \cup V_S$ using point-to-point *links*. The network must be *connecting,* in that there is a communication path from each input node to each output node. The *distance* from an input node to an output node is the number of links in the shortest communication path connecting them. The *diameter* or *depth* is the maximum distance from an input to an output. To ensure small delay in communication, we require that the diameter be small.

When all input-output paths have length equal to the diameter, the switch nodes (if any) can be partitioned into *stages*, by placing all nodes at distance i from an input in the ith stage. Networks admitting such a partitioning are *multistage interconnection networks*. A multistage interconnection network with a single input node and a single output node is a *channel graph*. In the analysis of interconnection networks, while the network designer is concerned with the overall network design, each input-output pair is concerned with the channel graph reflecting the portion of the network containing the paths from the specified input to the specified output.

In some applications, we require that the n input nodes can simultaneously communicate with the n output nodes, given a specified mapping of inputs to outputs. This requires *disjoint* communication paths, which share no common link or intermediate node. A good example of this situation arises in the design of shifting networks.

A *barrel shifter* is a network whose nodes are $\{0,1,...,n-1\}$, the integers modulo n. Given a shift distance s, $1 \leq s < n$, every node must transfer a value to the node whose label is s larger; more precisely, for each $0 \leq i < n$, node i must establish a connection to node $i+s$ (modulo n), and all n communication paths are to be disjoint. Kilian, Kipnis and Leiserson [111] develop a barrel shifter which has diameter one; when implemented in VLSI, the shift is accomplished in a single clock cycle.

We now consider an even stronger connection property of networks. An *n-superconcentrator* is a network with n inputs and n outputs in which disjoint communication paths can be established from the inputs to the outputs in *any* of the $n!$ possible orderings. We restrict superconcentrators to have only links, and no larger buses. A superconcentrator of depth one requires all n^2 connections (i.e. each input connected to each output); hence superconcentrators of depth greater than one are of interest. Nevertheless, superconcentrators are typically constructed using special types of depth-one networks in which every set of inputs is directly connected to a relatively large set of outputs (see, for example, [108]). More formally, a network $(V_I \cup V_O, E)$ with $V_I \cap V_O = \emptyset$ is

a (n, α, β)-*expander* if every set of α inputs is directly connected to at least β output nodes.

The motivation for strong expansion capability is to avoid congestion or blocking. To quantify the disruption due to such blocking, one examines the channel graphs for each input-output pair. We assume that the probability q_i of occupancy of each edge in a channel graph is known. The *blocking probability* of a channel graph is defined as the probability that every channel of that graph contains at least one occupied (blocked) edge. A channel graph with k stages is *superior* to another channel graph with k stages if the blocking probability of the former never exceeds that of the latter, independent of the occupancies for the E_i. One cannot ensure that constructing an interconnection network in which channel graphs are superior leads to an interconnection network with high expected throughput. Nevertheless, if the channel graphs have high blocking probability, this ensures poor throughput. Hence the design of superior channel graphs arises as a necessary step in interconnection network design.

9.2 The Connection to Designs

A network of diameter one is a 2-$(n, K, 1)$ covering; if we require in addition that each node i has a (disjoint) path to node $i + s$ (modulo n), the n pairs from the set $D_s = \{\{i, (i + s) \pmod n\} : 0 \leq i < n\}$ must appear in n distinct blocks. At first, this seems to be a complicated requirement, but a widely studied class of designs always has the desired property; we introduce them here. A set system $(V, \mathcal{B})$ with $V = \{0, 1, ..., n-1\}$ is *cyclic* if, whenever $\{b_1, ..., b_k\} \in \mathcal{B}$, $\{b_1 + 1, ..., b_k + 1\} \in \mathcal{B}$ (arithmetic modulo n is used). The *orbit* $\mathcal{O}(B)$ of a block B is the set $\{B + s \pmod n : 0 \leq s < n\}$; it is *full* when $|\mathcal{O}(B)| = n$. When all orbits are full, the set system is *full-cyclic.* It is easy to see that the pairs of D_s appear in at least n distinct blocks of a full-cyclic covering.

Any full-cyclic covering can then be used to design a barrel shifter. Each node finds the first orbit in which $\{0, s\}$ appears, say in block B. Node x now writes its value to the bus $B + x \pmod n$, and reads its value from the bus $B + x - s \pmod n$. In this way each node x reads the value node $x - s \pmod n$ wrote, and each communication path corresponds to a unique block in the orbit. Kilian, Kipnis and Leiserson [111] observe that to minimize the total number of buses and the number of buses incident at a node, the covering chosen is a *cyclic projective plane* (i.e. a projective plane which is cyclic).

The actual operation of a barrel shifter based on a cyclic projective plane is remarkably simple. To see this, consider the structure of cyclic projective planes. Since there are only n blocks, any two blocks B_1, B_2 satisfy $B_1 \equiv B_2 + s \pmod n$ for some $0 \leq s < n$. Consider a single block $B = \{b_1, ..., b_k\}$. Now for each element d, $1 \leq d < n$, $\{0, d\}$ appears in exactly one block. Hence B must contain exactly two elements b_i, b_j for which $b_j - b_i \equiv d \pmod n$. Every d,

$1 \leq d < n$, is the difference of two elements of B; such a set B is a *difference set* for $\{0, 1, ..., n-1\}$.

Using the difference set representation of the cyclic projective plane, the operation of a barrel shifter is straightforward. To shift a distance of s, each node finds the two elements b_i, b_j in the difference set with $b_j - b_i \equiv s \pmod{n}$. Node x then writes onto bus $x + b_i$, and reads from bus $x + b_j$.

When no cyclic projective plane on n elements exists, this very simple control logic can be retained nonetheless: This scheme requires only a set which covers all differences from 1 to $n-1$. Hence we can use a *difference cover*, in which each d, $1 \leq d < n$, is the difference of at least one pair of elements. Kilian, Kipnis and Leiserson [111] observe that they produce optimal barrel shifters of depth one. They also use difference covers to design 'permutation architectures', which realize permutations other than just cyclic shifts.

Let us turn to superconcentrators. Any depth-one network with $V_I \cap V_O = \emptyset$ can be equivalently written as a set system $(V_I, \mathcal{B})$, where $\mathcal{B} = \{\{v_i : \{v_i, v_o\} \in E\} : v_o \in V_O\}$. In this setting, an (n, α, β)-expander is a set system with n elements and n blocks, so that every set of α elements intersects at least β of the blocks. Intuitively, β is largest when the blocks intersect each other as little as possible. At the same time, however, for β to be large, each element must appear in a large number of blocks. To maximize the expansion, we choose to balance the block sizes, and balance the sizes of block intersections. Hence we consider symmetric designs.

Alon [106] proves that one class of symmetric designs, obtained from the points and hyperplanes of the projective geometries $PG(d, q)$, provides good expansion properties: In the design from $PG(d, q)$ on n elements, every set of α elements intersects $\beta \geq (\alpha n)/(\alpha + q - 1)$ blocks. Hence for all $\alpha = o(n)$, $\alpha = o(\beta)$; such a network is termed *highly expanding*. Moreover, Alon remarks that these expanders have essentially the smallest number of links of any network with equivalent expansion properties. Using projective geometries for expanders, Alon establishes the existence of n-superconcentrators of depth three with $O(n^4/3)$ links; see [106] for further uses of the expanders and superconcentrators, and [110] for a similar use of symmetric designs.

In the same way that designs lead to desirable expansion properties, they also arise in the design of superior channel graphs. Chung [107, 109] uses incidence graphs of block designs to determine connections between the second and third stage of a four-stage channel graph. She establishes that, among all channel graphs with the same numbers of nodes in each stage and the same numbers of interstage links, channel graphs arising from block designs are superior.

The design of interconnection networks employs design-theoretic tools in a number of ways. The use of designs to cover all pairs of nodes is prevalent in diameter one networks; on the other hand, the balanced intersection of blocks is shown to lead to high expansion factors, and hence to highly connecting networks.

10 Partial Match Queries on Files

In this section, we explore an application of designs to file organization. We focus on the simpler applications of designs here, to somewhat specialized problems, and content ourselves with providing some references to the more general situations.

10.1 The Application

A *file* is a collection of *records;* each record has a number of *attributes,* and we retrieve records by specifying their attributes. A primary requirement for any file organization is the support of *partial match queries*, in which values for some attributes are given and the remainder are unspecified. All records matching the values in the specified attributes are to be retrieved. Normally, records are relatively space-consuming objects; hence they are stored on a slower secondary storage device and an *accession number* records their address on this device. Our task, given a partial match query, is thus reduced to listing the relevant accession numbers.

We consider the situation in which there are n binary attributes. Moreover, we consider queries which request those records possessing certain attributes; the extension to the case in which we further stipulate that the records not have certain other attributes is not essentially more difficult. In a typical retrieval system, queries are relatively simple, in that they involve relatively few of the attributes. Hence, we first consider the case where partial match queries on up to t attributes must be supported, but queries on more than t attributes need not be.

The usual *inverted file system* creates a list of accession numbers for each attribute, and intersects these lists to reply to a partial match query. This requires the examination of very many accession numbers which do not form part of the final answer. At the other extreme, an *extended* inverted file system creates (in advance) a "bucket" of accession numbers for each partial match query. Redundancy is incurred in this scheme, but can be limited by only placing an accession number in a bucket when the partial match query is a maximal query which matches the record. The redundancy in storage pays off in retrieval, because in this scenario only the relevant accession numbers are examined. The impracticality of this approach arises from the very large number of buckets required, and a correspondingly large requirement for redundancy.

A compromise solution is to amalgamate many possible queries into a single bucket. Each bucket remains associated with a subset of the attributes, but may now contain information about many maximal partial match queries. The essential feature of the bucket subsets is that each query subset be contained in at least one bucket subset.

Rivest [122] remarks on a practical limitation, the large redundancy introduced by storing accession numbers many times. While this may be quite acceptable for small files, one would require a very large difference in the sizes

of records and their accession numbers before the storage for buckets would be less than storage for the file itself. Nevertheless, combinatorial filing schemes remain useful when the cost of retrieving a record from the secondary storage is so large that one is unwilling to retrieve any record which may prove irrelevant to the query at hand. In the absence of such a prohibitive cost, however, we need only ensure that "most" of the records retrieved prove relevant; Rivest's scheme, which we explore next, has this property.

Rivest [122] considers partial match queries of any size on a file with n binary attributes; a query specifies records which do possess certain properties, do not possess certain others, and may or may not possess the remainder. In this case, records are placed in buckets; however, here the buckets partition the records, i.e. no redundancy is permitted. A record R is placed in a bucket M_i by evaluating a hash function h; if $h(R) = i$, R is placed in M_i. The hash function is therefore a function which partitions all possible records into b buckets $M_1, ..., M_b$. To answer a partial match query Q, we determine (in a manner as yet unspecified) all buckets which could contain a record matching Q, and then linearly search all of the selected buckets. (If accession numbers rather than actual records are stored, we must access the secondary storage to retrieve all of the records in these buckets.)

Since data can be stored on many secondary storage devices, one can exploit parallelism if the records of interest lie on different disks. Suppose then that each bucket is associated with a disk. Examine the records which hash to the same bucket to determine when accesses must be made sequentially. One expects that two records meeting the requirements of a partial match query have many similar or equal attributes. This argues for the selection of a hash function which places records that are similar in many attributes in different buckets. Faloutsos and Metaxas [119] examine this problem under the assumption that attributes are binary (or can be made so by partitioning the set of values into 'high' and 'low', for example). Abdel-Ghaffar and El Abbadi [113] consider the case in which each attribute has p values (or, again, its range of values partitioned into p classes). The objective is to determine the largest value of d, given the numbers of records in a bucket (m), attributes (n) and values per attribute (p), so that one can partition the set of all possible records into buckets with no two records in the same bucket agreeing in more than $n - d$ attributes. Such a hash function has *maximum distance*.

10.2 The Connection to Designs

Now we are in a position to introduce combinatorial filing schemes. Let $A = \{a_1, ..., a_n\}$ be a set of attributes. Let $(A, \mathcal{B})$ be a t-covering, and write $\mathcal{B} = \{B_1, ..., B_m\}$. Each B_i has an associated list, or *bucket* M_i. Not all subsets of A appear in blocks of $\mathcal{B}$, but we are guaranteed that all t'-subsets with $t' \leq t$ are. A subset $A' \subseteq A$ which does appear may be a subset of many blocks; we write $f(A') = i$ if the "first" block containing the subset A' is B_i. Now many

subsets are associated with bucket M_i, and hence we partition this bucket into *subbuckets;* in particular, for each $A' \subset A$ with $f(A') = i$, we form a subbucket $M_{i,A'}$.

To enter a new record with attributes R, we place its accession number in subbucket $M_{i,A'}$ provided that $R \cap B_i = A'$ and $f(A') = i$. Each accession number thus appears in at most one subbucket of each bucket, but may appear in many different buckets. To answer a partial match query Q, we determine $i = f(Q)$ and only examine bucket M_i. The relevant accession numbers are then listed, each exactly once, by catenating all of the subbuckets $M_{i,A'}$ with $Q \subset A'$.

If only one bucket is used, this scheme reduces essentially to extended inverted filing. In fact, within each bucket, the scheme is like extended inverted filing, with one important difference. Subbuckets $M_{i,A'}$ exist even for sets A' which are too large to be partial match queries themselves; this eliminates redundancy within a bucket. The main advantage of first partitioning into buckets in this way is that the filing problems remaining within a bucket are intended to be of manageable size.

Two competing goals affect the selection of a t-covering to be used. First, the redundancy incurred by storing accession numbers in many buckets dictates that the t-covering should have few blocks; intuitively, fewer blocks lead to less redundancy. Second, larger blocks lead to more subbuckets per bucket, and hence leave larger filing problems within a bucket; intuitively, one prefers smaller blocks. The tradeoff between having few larger blocks or many smaller blocks is very application dependent. When $t = 2$, Ray-Chaudhuri [121] observes that the first goal suggests the use of projective planes. If the second goal is taken into account, block designs with small blocksize are preferable [121]. When $t > 2$, the designs to be used are not as readily available, especially in view of the requirement that the index be 1. For $t \leq 5$, many suitable designs are known to exist, but as noted earlier, existence is far from settled in general.

A most profitable direction to extend this research has been considered in [117, 121]. On each bucket, we can develop a second combinatorial filing scheme, and thereby develop an overall method which is multi-stage. To do this, on each block of a t-$(v, k, 1)$ design, we place a copy of a t-$(k, k', 1)$ design. The operation of the filing scheme is to first find the relevant block of the t-$(v, k, 1)$ design, and then within that block find the relevant block of the t-$(k, k', 1)$ design; this could naturally be repeated to form a filing scheme with any desired number of stages. In practical terms, the lack of known Steiner systems with large t and k limits the usefulness of this idea, however. Even when appropriate systems are known, one might argue that the result is just a t-$(v, k', 1)$ design. Of course, it is such a design, but has the advantage that we need not search all blocks of the design in order to locate the relevant bucket. One would employ two mappings here, one to locate the relevant block of the t-$(v, k, 1)$ design, and the second to locate within that subset the relevant block

of the t-$(k, k', 1)$ design.

A second profitable direction is to generalize the scheme to handle multiple valued attributes. The extension of the design-theoretic approach to this problem has been studied by many authors, notably Bose and Koch [117], Bose et al. [116], Ghosh and Abraham [118], and Berman [115].

The essential ingredient in Rivest's scheme is the selection of the hash function. It must have two properties. We must be able to easily determine whether records in a given bucket could possibly match a given query. In addition, we want to examine as few buckets as possible (either on average or in the worst case). These decisions are not affected by the file itself. Rivest's main theorem here shows that if we have $b = 2^w$ buckets, to minimize the average number of buckets examined we choose a function which hashes a group of records to the same bucket if they are "close" in the following sense. For bucket M_i, there exist sets S_i^0 and S_i^1 for which all records which have attributes in S_i^0 set to 0 and attributes in S_i^1 set to 1, and no others, are hashed to M_i. Moreover, the number of specified attributes $|S_i^0 \cup S_i^1|$ is w. Hence the set of records hashed to bucket M_i can be easily encoded as an n-vector with entries 0,1,* containing w digits and $n - w$ *'s; we call these vectors *signatures* of the buckets. The asterisks denote "don't care" positions.

An easy example when $n > log_2 b$ simply uses the first $log_2 b$ bits of the record to determine the bucket. While this easy method has optimal average case performance, in the worst case it may require the examination of all buckets. What yields the best worst-case complexity? Rivest [122] addresses this question by considering a novel type of designs. An *associative block design* $ABD(n, w)$ is a $2^w \times n$ array with entries from 0,1,* so that

1. each row has w digits and $n - w$ *'s,

2. for every pair of rows, there is a column in which they contain different digits, and

3. every column contains the same number, $2^w(n - w)/n$, of *'s.

Conditions (1) and (2) ensure that the signatures (=rows) form a partition of the file, which when used as a hash function delivers optimal average case performance. Condition (3) is designed to ensure that worst-case performance is also good.

Let us turn to maximum distance hash functions, following [119] and [113]. When one examines the records placed on one of the disks (buckets), treat the records as codewords in a p-ary code. This is a code of length n with m codewords, and distance as large as possible. The Singleton bound (see [7]) establishes that the distance d satisfies $d \leq n - \lceil \log_2(m) \rceil + 1$. When $m = p^\alpha$ and equality is met in the Singleton bound, the code is an MDS code, and then numerous connections with designs are well studied [123]. Cases when m is not a power of p pose challenging design-theoretic questions. In

particular, when $p < m \leq p^2$, the Singleton bound permits the possibility of a code of distance $n-1$. A p-ary code of distance $n-1$ can be interpreted as a generalization of a transversal design as follows: Let X be a set of size np partitioned into n groups of p elements each. Then a *transversal packing* of type p^n is a collection of n-sets with one element from each group, so that no two of the n-sets intersect in more than one element (equivalently, no pair of elements occurs in more than one n-set). Existence of transversal packings of type p^n essentially asks for the maximum number of blocks (n-sets). When this maximum number of blocks is p^2, the packing is a transversal design and hence equivalent to a set of mutually orthogonal latin squares. However, there are numerous cases in which the required number n exceeds the block size of any, or any known, transversal design. In these cases, the application motivates the study of dense transversal packings. An equivalent formulation in terms of mutually orthogonal *partial* latin squares is given in [112].

In closing, we mention also the related notion of perfect hash families. Again, there is a useful connection with difference matrices and other designs [114].

11 Software Testing

Testing is an important but expensive part of the software development process. In this section we describe how certain combinatorial designs are being used to reduce the number of tests needed in order to assure that a software product performs correctly in most reasonable instances.

11.1 The Application

The problem is to design a test plan for a software system. For a moderate-sized system there can billions of possible test scenarios. So it is important to find a test plan that is not too large, yet tests for most of the interactions among the possible outputs in the modules of a software system. To do this, software developers have begun using combinatorial designs to test for these interactions. This usage is closely related to the use of combinatorial designs in the design of experiments.

To design a test plan, the tester identifies possible output values from each of the stages of the software system. An example from [126] comes from the testing of a telephone switch. In this case the stages are the *call type* with output values local, long distance and international; *billing type* with values either caller, collect or 800; *access parameter* with values ISDN, PBX or loop and the *status parameter* with values success, busy or blocked. Since each of the four stages has three values, there are $3^4 = 81$ different scenarios.

The goal is to reduce the number of test scenarios while still testing all of the two-way interactions. In [126] a test model with 13 scenarios is given that covers all possible two-way interactions. Larger examples can easily be

constructed which show that the number of scenarios can be reduced substantially if only all the two-way interactions are covered. There is evidence that most errors that do occur in the development of a new software system can be found if the pairwise (and sometimes three-way) interactions are all tested.

A similar application arises in testing combinational logic circuits. Such a circuit implements a collection of functions from a set of n binary inputs to m binary outputs. When every function depends upon the value of at most d of the inputs, one wants to generate a set of binary n-vectors which include every possible assignment of values to each subset of d inputs; this is a *universal test set*. Seroussi and Bshouti [133] discuss this application further and provide numerous references. In this case, the restriction to binary values is imposed by the nature of the circuits, but d-way interactions for $d > 2$ are typically of concern.

11.2 The Connection to Designs

A *covering array* $\mathrm{CA}(t, k, g)$ is an array with k rows with each cell filled with an element of an g-set S that has the property that given any set of t rows, every $t \times 1$ column vector with symbols from S occurs at least once. The parameter t is the *strength* of the covering array and when $t = 2$ a $\mathrm{CA}(2, k, g)$ is denoted $\mathrm{CA}(k, g)$. If the requirement is strengthened to require each pair exactly once, then a covering array is an *orthogonal array*. Just as orthogonal arrays are equivalent to transversal designs, covering arrays are equivalent to *transversal covers*.

The example in the previous section asks for a covering array with 4 rows and with $|S| = 3$. Such an array with exactly 13 columns exists; each column gives a test. In general, we can design a test plan for a software system with k stages each with g outputs by using a $\mathrm{CA}(k, g)$. This test plan would test for all possible two-way interactions of outputs from two different stages.

The first result of interest is a lower bound on the number of columns needed in a $\mathrm{CA}(k, g)$, to bound the number of runs needed to test the software system. To simplify the discussion we consider only two-way interactions (i.e. in the covering array, given any two rows, every ordered pair of symbols occurs in at least one column).

Poljak and Tuza [132] proved a bound for a partition problem in set theory concerning so called *qualitatively independent* partitions. That problem is essentially the dual of covering arrays and hence the Poljak-Tuza bound can be converted to a bound for covering arrays:

Theorem 11.1 *Let $c(k, g)$ denote the minimum number of columns needed in a $CA(k, g)$, then $c(k, g) \geq \frac{g}{2} \log_2 k$ as $k \to \infty$.*

Gargano, Korner and Vaccaro [131] provide constructions which prove that $c(k, g) = \frac{g}{2} \log_2 k$ as $k \to \infty$. This asymptotic result is mainly of theoretical interest, and is not constructive. Subsequent work has focussed on finding

specific constructions for given small values of k and g (see, for example, [134, 135]). When $k \leq g+1$, one can use orthogonal arrays or, equivalently, sets of orthogonal latin squares. However, software testing applications typically have g small relative to k.

There is a nice construction in the case of $g = 2$. Let X be a set with $2n+1$ points and index the rows of an array A by the subsets $Y \subset X$ where $|Y| = n$. There are exactly $\binom{2n+1}{n}$ rows. Then for $1 \leq i \leq 2n+1$ place a 1 in row Y and column i if $i \in Y$, otherwise put a 0. It is not difficult to show that the array A is a covering array $\mathrm{CA}(\binom{2n+1}{n}, 2)$ with exactly $2n+1 = c$ columns. This is best possible as one can show (using Stirling's formula) that $c \sim \frac{2}{2}\log_2 k$ as $n \to \infty$.

The next goal is to construct covering arrays with $g > 2$. Clearly, using the result for $g = 2$ one can asymptotically make a covering array for any $g > 3$ with $c = \binom{g}{2}\log_2 k$ columns. In [130] asymptotic constructions (for k large) are given for covering arrays with $c = s\log_2 k$ columns for some specific small values of $g \leq 13$. In particular, they show that when $g = 4$, that $s \leq 3.911$; when $g = 5$, that $s \leq 5.48$ and when $g = 6$ that $s \leq 8.9$.

The first recursive construction was given in [132]. There, the authors give a construction that increases the number of rows in a covering array while leaving the value of g unchanged. Recently, this was strengthened in a paper by Cohen and Fredman [128] to the following.

Theorem 11.2 *If there exist covering arrays $CA(k_1, g)$ and $CA(k_2, g)$ with c_1 and c_2 columns respectively, then there exists a covering array $CA(k_1k_2+1, g)$ with $c_1 + c_2 - g$ columns.*

In [135] constructive techniques are given for covering arrays that are along the lines of some of the more well known design theoretic techniques. These include Wilson type theorems using covering arrays with holes and the use of pairwise balanced designs. A table of explicit values of $n(k, g)$ for all $3 \leq g \leq 7$ and $2 \leq k \leq 50$ can be found in that paper.

In [126, 127, 129], the AETG system implemented at Bellcore to design efficient test sets is discussed. This system employs generalizations of covering arrays with the property that the number of symbols allowed in each row may differ. The success of their AETG system in finding such covering arrays has led to useful testing schemes.

Covering arrays for t-way interactions when $t > 2$ are treated in [124, 133, 134] and constructions when $t = 3$ are given in a recent paper [125] using techniques developed initially for orthogonal arrays.

12 Disk Layout and Striping

In this section, the use of designs to balance access to a secondary storage device is examined.

12.1 The Application

We are again concerned with arrays of disks, as in §4. Here we concentrate on the case that erasure of any one disk can be corrected, and focus instead on balancing workload on individual disks in the disk array. In the simplest case, if there are v disks, $v-1$ blocks of data together with one *parity block* consisting of the modulo 2 sum of the data blocks, form a *parity stripe*. Placing one block from the stripe on each disk permits reconstruction if one disk fails. However, this reconstruction requires access to all remaining disks. Instead one can choose parity stripes to contain blocks on $k < v$ of the disks; then k is the *parity stripe size*. Although the overhead in storage increases from $\frac{1}{v}$ to $\frac{1}{k}$, only $k-1$ disks need be read in order to perform a reconstruction. This technique is *parity declustering*.

Parity declustering requires that we select the parity stripes, and the location of the parity block within each stripe, i.e. choosing a *data layout*. We suppose that each of the v disks can store exactly s blocks. Altogether the vs blocks must be allocated either as data or parity blocks in stripes. Holland and Gibson [139] observe that in any layout, each stripe must have all blocks on different disks to permit reconstruction; parity blocks must occur in numbers as equal as possible on each disk to balance disk workload; every two disks must be in a number of stripes together that is as equal as possible, in order to balance disk workload resulting from reconstruction; and the task of determining where a block is stored in the disk array must be efficient.

12.2 The Connection to Designs

Muntz and Lui [141] suggested that stripes of the disk layout be treated as blocks in a design whose points are the disks. Then Holland and Gibson's first and third conditions are met by any block design. Even distribution of parity units requires that each block have a point designated to serve as the disk for the parity block, but the number of times each point serves as such a representative is to be as equal as possible for all points. Holland and Gibson [139] suggest repeating the blocks of the designs k times, where k is the block size, so that each point in a block can serve as the representative of one copy. Schwabe and Sutherland [142] establish that such replication is not needed, proving via network flow techniques that occurrences of points as representatives can be equalized. Indeed, this result is implied by a stronger result of Levi [140].

In the connection with designs, the fourth requirement of Holland and Gibson is perhaps the most problematical. One cannot reasonably store the entire disk layout in array format and undertake table lookups whenever a block is to be mapped to a disk location. For this reason, recent work has concentrated on properties of designs that make them quick to generate 'on the fly' [142]. This problem is addressed by Alvarez *et al.* [136]. They also examine disk layouts which permit reconstruction from multiple failures.

Mapping data to memory arises elsewhere in connection with designs, for example in the processing of partial match queries (§10), and in the mapping of matrix data to parallel memory modules. In the latter context, latin squares in which certain subarrays always contain distinct symbols arise [137, 138].

13 (t, m, s)-Nets and Numerical Integration

Until this point, we have concentrated on connections focussing on communications, cryptography, and networking. In this final section, we examine combinatorial tools that have found recent application to computation of definite integrals. This has had particular impact in the area of finance [143]. The principal current application is in the financial arena, where they are used to price exotic options. Nevertheless, the connection described affords a technique of general applicability. Indeed, Niederreiter [154] provides applications for the same combinatorial tools in pseudo-random number generation, which in turn are employed in cryptographic protocols.

13.1 The Application

The Monte Carlo method is widely used in simulation and numerical computation, and applies particularly when a numerical approximation is needed to the value of a function whose exact value is not known or easily computed. Typically, a sample space of possible values to be explored is constructed. The dimensions of the sample space represent variables in the problem specification, and the range of values in each dimension represents the interval of possible values of the associated variable. Scaling and translating each variable so that its range is continuous from 0 to 1, and assuming that variables are treated independently of one another, the sample space can be taken to be a unit cube in the associated number of dimensions. Typical Monte Carlo sampling selects points from the unit cube uniformly distributed in each dimension, performs a calculation using the values of the variables at each point, and aggregates the function values over all data points chosen.

When the unit cube has many dimensions, or the function to be evaluated is very irregular within the sample space, the Monte Carlo method often requires the selection of a number of data points which is too large to permit effective computation. The major obstacle is that dramatic local changes within the sample space are only detected by placing a sample point 'near' the change. While we want to preserve the random nature of the point spacing in the sample space, we also want to ensure that all small regions of the sample space receive their fair share of the sample points to be placed. To do this, the cube is partitioned into a number of smaller parallelepipeds, and each parallelepiped is required to receive the correct share of points. The goal is to ensure that every parallelepiped, not just the chosen ones, receives a share of the points which is at least approximately commensurate with its share of the total volume.

Let $\mathcal{I}_s$ be the unit cube of dimension s. Let $b \geq 2$ and $m \geq 1$ be integers, let $d_1, \ldots, d_s$ be nonnegative integers, and let $a_1, \ldots, a_s$ be integers satisfying $0 \leq a_i < b^{d_i}$. An *elementary interval in base* b in $\mathcal{I}_s$ is the parallelepiped of points $(x_1, \ldots, x_s)$ satisfying $a_i \leq x_i b^{d_i} < a_i + 1$. Let $s \geq 1$, $b \geq 2$ and $m \geq t \geq 0$ be integers. A (t, m, s)-*net in base* b is a multiset M of b^m points in $\mathcal{I}_s$ so that every elementary interval in base b in which $\sum_{i=1}^{s} d_i = m - t$ (i.e., the parallelepiped has volume b^{t-m}) contains precisely b^t points in M.

For a collection of σ points in the unit s-cube, the average number of points in a parallelepiped of volume v is σv. The *local discrepancy* of a particular parallelepiped is the difference between this average and the actual number of points found. The *discrepancy* is the maximum of the local discrepancies over all parallelepipeds. The main motivation for (t, m, s)-nets is that they provide point collections with "low" discrepancy (for certain elementary intervals, in fact, the discrepancy is guaranteed to be zero) [154].

13.2 The Connection to Designs

Two (0,2,2)-nets in base 3 are shown in Figure 2. Each consists of $9 = 3^2$ points placed in a unit square. Lines have been drawn vertically and horizontally at multiples of $\frac{1}{3^2}$ to assist in the verification. Points are shown as black circles, but the actual point should be taken to be the center of the circle shown.

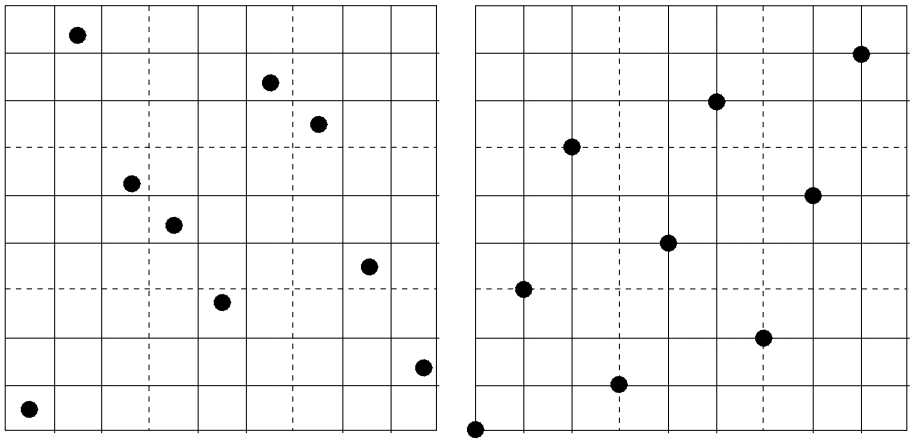


Figure 2: Two (0,2,2)-nets in base 3

To verify that each is a (0,2,2)-net in base 3, one must check that every elementary interval of area $\frac{1}{3^2}$ contains exactly one point. There are three types of such elementary intervals in this case. The first type contains horizontal stripes of width 1 and height $\frac{1}{3^2}$, defined by an integer d satisfying $0 \leq d < 9$, including the points with $0 \leq x < 1$ and $\frac{d}{3^2} \leq y < \frac{d+1}{3^2}$. The second type contains vertical stripes of height 1 and width $\frac{1}{3^2}$, defined by an integer c

satisfying $0 \le c < 9$, including the points with $\frac{c}{3^2} \le x < \frac{c+1}{3^2}$ and $0 \le y < 1$. The third type contains subsquares of height $\frac{1}{3}$ and width $\frac{1}{3}$, defined by integers c and d satisfying $0 \le c < 3$ and $0 \le d < 3$, including the points with $\frac{c}{3^1} \le x < \frac{c+1}{3^1}$ and $\frac{d}{3^1} \le y < \frac{d+1}{3^1}$. There are nine of each type of elementary interval in the example given.

Now consider the (0,2,2)-net on the left. Within each small subsquare of side $\frac{1}{3^2}$, when a point is present, any relocation of that point within the small subsquare does not alter the fact that the diagram forms a (0,2,2)-net. Indeed we can encode each such small subsquare by indicating the (x, y) coordinate of its lower left corner, so that the coordinates are integer multiples of $\frac{1}{3^2}$. To specify the position of the point, it is sufficient to specify the coordinates of the lower left corner of the small subsquare in which it lies. For practical reasons, it may be reasonable to displace points from these coordinate positions. However, when finding (t, m, s)-nets in base b, one can always assume that points are placed at positions in the unit cube in which each coordinate is a multiple of $\frac{1}{b^m}$.

Carrying this further in our example, we can write the x and y coordinates which are multiples of $\frac{1}{3^2}$ in the form $\frac{x_1}{3^1} + \frac{x_2}{3^2}$ and $\frac{y_1}{3^1} + \frac{y_2}{3^2}$, where x_1, x_2, y_1, and y_2 are each one of the integers 0, 1, or 2 (and, in general, are nonnegative integers less than b). Let us then examine the coordinates of the nine points in the first (0,2,2)-net. They are

$$\begin{array}{lll}
(\frac{0}{3^1}+\frac{0}{3^2}, \frac{0}{3^1}+\frac{0}{3^2}) & (\frac{0}{3^1}+\frac{1}{3^2}, \frac{2}{3^1}+\frac{2}{3^2}) & (\frac{0}{3^1}+\frac{2}{3^2}, \frac{1}{3^1}+\frac{2}{3^2}) \\
(\frac{1}{3^1}+\frac{0}{3^2}, \frac{1}{3^1}+\frac{1}{3^2}) & (\frac{1}{3^1}+\frac{1}{3^2}, \frac{0}{3^1}+\frac{2}{3^2}) & (\frac{1}{3^1}+\frac{2}{3^2}, \frac{2}{3^1}+\frac{1}{3^2}) \\
(\frac{2}{3^1}+\frac{0}{3^2}, \frac{2}{3^1}+\frac{0}{3^2}) & (\frac{2}{3^1}+\frac{1}{3^2}, \frac{1}{3^1}+\frac{0}{3^2}) & (\frac{2}{3^1}+\frac{2}{3^2}, \frac{0}{3^1}+\frac{1}{3^2})
\end{array}$$

For each point $(\frac{x_1}{3^1}+\frac{x_2}{3^2}, \frac{y_1}{3^1}+\frac{y_2}{3^2})$, we tabulate the values of x_1, x_2, y_1, and y_2:

x_1	0	0	0	1	1	1	2	2	2
x_2	0	1	2	0	1	2	0	1	2
y_1	0	2	1	1	0	2	2	1	0
y_2	0	2	2	1	2	1	0	0	1

What does the fact that the points are selected from a (0,2,2)-net imply? In each horizontal stripe which is an elementary interval, there is exactly one point. This requires that among the nine choices of (x_1, x_2), we find each ordered pair in which each element is 0, 1, or 2 exactly once. One can verify that this condition is met in the table given. Two tuples $(g_1, \ldots, g_{b^s})$ and $(h_1, \ldots, h_{b_s})$ with elements from $0, 1, \cdots, b-1$ are called *orthogonal* when the ordered pairs (g_i, h_i) for $1 \le i \le b^s$ contain every ordered pair of elements from $0, 1, \cdots, b-1$ exactly b^{s-2} times. Hence, the requirement on horizontal stripes can be simply stated as a requirement that the rows x_1 and x_2 are orthogonal. In the same way, requiring a single point in each vertical stripe which is an elementary interval is the same as requiring that rows y_1 and y_2

are orthogonal. Finally, the requirement that every $\frac{1}{3} \times \frac{1}{3}$ subsquare which is an elementary interval contains a single point is equivalent to the statement that rows x_1 and y_1 are orthogonal.

Selecting any four tuples of length 9 with elements 0, 1, or 2 and indexing them by x_1, x_2, y_1, and y_2, we find that when the rows x_1 and y_1 are orthogonal, the rows indexed by x_1 and x_2 are orthogonal, *and* the rows indexed by y_1 and y_2 are orthogonal, the process can be reversed to produce a (0,2,2)-net. Choosing these four tuples is not as difficult as it might seem. The reason is simple. If we choose the tuple indexed by y_2 to be the same as that indexed by x_1, and the tuple indexed by x_2 to be the same as that indexed by y_1, the requirements are met. In fact, then, we need only select two tuples in this case, to form:

x_1	0	0	0	1	1	1	2	2	2
x_2	0	2	1	1	0	2	2	1	0
y_1	0	2	1	1	0	2	2	1	0
y_2	0	0	0	1	1	1	2	2	2

Now using the entries in this table to determine locations of nine points again yields a (0,2,2)-net in base 3. In fact, it is the second (0,2,2)-net shown in Figure 2.

The natural question is to attempt to produce nets in higher dimensions. Our example of a (0,2,2)-net leads us to ask when one can form a $(0, 2, s)$-net in base 3, and in particular, what maximum value for s is achievable. With this in mind, we next describe a (0,2,4)-net in base 3. First form the array in Figure 3.

0	0	0	1	1	1	2	2	2
0	1	2	0	2	1	0	2	1
0	1	2	1	0	2	2	1	0
0	1	2	2	1	0	1	0	2

Figure 3: An array for constructing a (0,2,4)-net

This array has the important property that any two distinct rows are orthogonal. We shall construct coordinates for nine points in the unit 4-dimensional cube. Each point therefore needs four coordinates, each of which is written in the form $\frac{x_1}{3^1} + \frac{x_2}{3^2}$. We use the array given to determine the four coordinates of each point as follows. Each point corresponds to a column $(c_1, c_2, c_3, c_4)^T$ of the array. The point in the net can then be written as

$$(\frac{c_1}{3^1} + \frac{c_2}{3^2}, \frac{c_2}{3^1} + \frac{c_3}{3^2}, \frac{c_3}{3^1} + \frac{c_4}{3^2}, \frac{c_4}{3^1} + \frac{c_1}{3^2}).$$

For example, the column $(1, 2, 0, 1)^T$ corresponds to a point placed at $(\frac{5}{9}, \frac{6}{9}, \frac{1}{9}, \frac{4}{9})$.

In this way, we obtain a (0,2,4)-net in base 3. Niederreiter [154] established the equivalence between (0,2,s)-nets in base b and OA(s, b)s; indeed, we saw this correspondence earlier. The correspondence also can be used to establish nonexistence of certain (0,2,s)-nets in base b, by employing the observation that an OA(s, b) exists only if $s \leq b+1$ (see, for example, [2]). Equality can occur, and indeed is known to occur when b is a prime or a power of a prime number. Nevertheless, for certain choices of b, it is known that equality does *not* occur. Determining the largest value of s for which an OA(s, b) exists is a challenging open problem. When $b = 6$, for example, the largest value admitted for s is 3. When $b = 10$, the largest value for s is, at present, unknown. Mullen [150] discusses a number of remarkable connections with various combinatorial objects; see also [2] for results and existence tables for OA(s, b), given there in the equivalent formulation as mutually orthogonal latin squares.

Now let us examine the existence of (t, m, s)-nets in base b more generally. A (t, m, s)-net in base b gives a (u, n, r)-net in base b whenever $t \leq u \leq n \leq m$ and $1 \leq r \leq s$. The goal is to construct (t, m, s)-nets with $m - t$ "large", since they have stronger uniformity properties. When $m - t = 0$, any set of b^m points can be chosen. When $m - t = 1$, take the points $(i/b, \ldots, i/b)$ for $i = 0, 1, \ldots, b-1$, each b^{m-1} times. Hence the cases of interest are for $m \geq t+2$. Niederreiter [155] and Mullen and Whittle [153] showed that the existence of an OA$_{b^t}(2, s, b)$ is equivalent to the existence of a $(t, t+2, s)$-net in base b. The equivalence mirrors that for $(0, 2, s)$-nets in base b that we have seen. When $m - t \geq 3$, one requires a more general combinatorial model. Lawrence [146] generalized the definition of orthogonal array as follows: A *cubical orthogonal array* COA$_\lambda(t, k, n)$ is a $t \times k \times \lambda n^t$ array $C = (c_{ijk})$ with elements from a set of size n, meeting a condition on certain subarrays as follows. A set $S \subseteq \{(x_i, y_i) : 1 \leq i \leq t,\ 1 \leq x_i \leq t,\ 1 \leq y_i \leq k\}$ is a *qualifying collection* of rows when $(x_i, y_i) \in S$ and $x_i > 1$ implies that $(x_i - 1, y_i) \in S$. In Figure 4, three qualifying collections of rows with $t = 4$ and $k = 6$ are shown. The first is {(1,1), (2,1), (3,1), (4,1)}; the second is {(1,2), (1,4), (1,6), (2,6)}; and the third is {(1,3), (2,3), (3,3), (1,5)}. These are meant only to illustrate the definition, and many more qualifying collections are present. Figure 4 depicts a 2-dimensional $t \times k$ 'slice' of the 3-dimensional COA.

The qualifying collections correspond exactly to the elementary intervals. When S is a qualifying collection of rows, define a $t \times \lambda n^t$ array $A = (a_{ik})$ by setting $a_{ik} = c_{x_i, y_i, k}$; then, A contains every $t \times 1$ column vector the same number λ of times. These are often called *generalized orthogonal arrays*, but we employ the adjective 'cubical' to suggest the three-dimensional structure of the array. Cubical OAs generalize orthogonal arrays in the following sense: In a COA$_\lambda(t, k, n)$ with $t \leq k$, the $k \times \lambda n^t$ array obtained by setting the first coordinate equal to 1 is an OA$_\lambda(t, k, n)$.

The construction of a (t, m, s)-net in base b from a COA$_{b^t}(m - t, s, b)$ proceeds as follows. For each p satisfying $1 \leq p \leq b^m$, consider the $(m - t) \times s$ array A_p whose (i, j) entry is the (i, j, p) entry of the COA. Each selection of

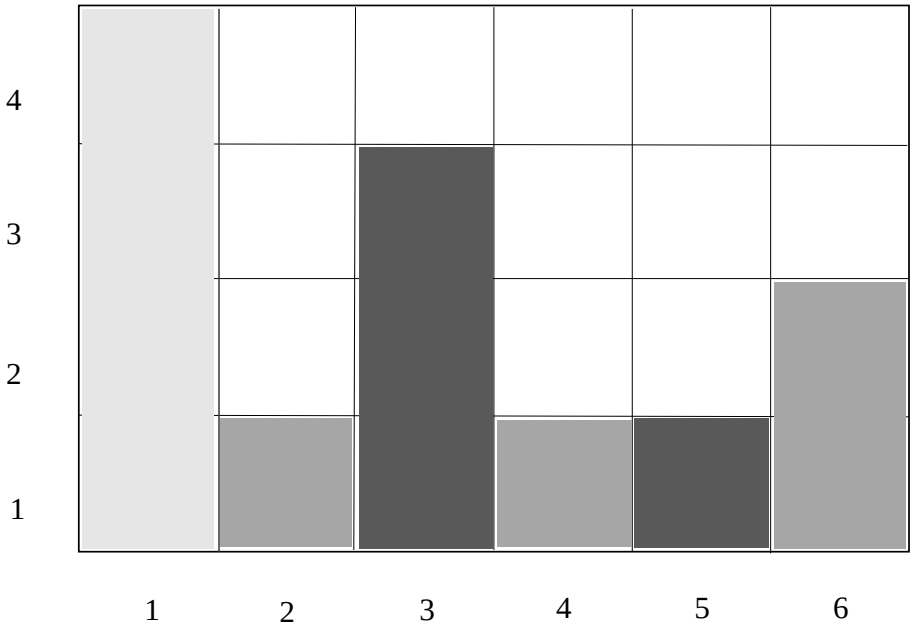


Figure 4: Qualifying Collections

p corresponds to one of the b^m points in the (t, m, s)-net, and A_p is used to determine its position in the s-dimensional unit cube as follows. Denote the entry of A_p in position (i, j) by $a_{i,j}$, for $1 \leq i \leq m - t$ and $1 \leq j \leq s$. Each column of A_p determines the coordinate of the pth point of the net. The jth column gives the coordinate of the point in the jth dimension as

$$\frac{a_{1,j}}{b^1} + \frac{a_{2,j}}{b^2} + \cdots + \frac{a_{(m-t),j}}{b^{m-t}},$$

or equivalently, as

$$\sum_{i=1}^{m-t} \frac{a_{i,j}}{b^i}.$$

Carrying out this computation for each column j with $1 \leq j \leq s$ determines the position of the point in the s-dimensional unit cube completely. Of course, the position of the point can (and, to avoid bias, should) be displaced from the 'corner' by adding (random) values less than $\frac{1}{b^{m-t}}$ to each coordinate. This process is then repeated for each of the b^m points.

It may seem unnecessarily complicated to employ a cubical OA to construct a (t, m, s)-net. However, Lawrence [146] has shown that the existence of a (t, m, s)-net in base b is, in fact, equivalent to the existence of a $\mathrm{COA}_{b^t}(m - t, s, b)$. Consequently, every construction of a (t, m, s)-net effectively relies upon the construction of some cubical orthogonal array.

For this reason, the primary question about the construction of (t, m, s)-nets is the construction of cubical OAs. We turn to this next. Since cubical

OAs have only recently been introduced, it is not surprising that a number of the available constructions are from orthogonal arrays, the latter having been studied extensively for the past five decades [2, 145].

Suppose that an $\mathrm{OA}_{b^t}(m-t, \widehat{s}, b)$ exists. Let us index the $\widehat{s}$ rows of this OA by symbols $1, 2, \ldots, \widehat{s}$. If we can form an $(m-t) \times s$ array $A = (a_{i,j})$ whose entries are from $\{1, 2, \ldots, \widehat{s}\}$, in which every qualifying collection of size $m-t$ contains $m-t$ distinct entries, Then a $\mathrm{COA}_{b^t}(m-t, s, b)$ can be formed, by placing in the (i, j, ℓ) position the entry in the $(a_{i,j}, \ell)$ position of the OA. The orthogonal array with which we began provides more structure than is really needed to get the COA. To produce a COA from the OA, it suffices to find an array A meeting the requirement on qualifying collections.

When $m-t$ is even, set $m-t = 2e$ and $s = \lfloor \frac{\widehat{s}}{e} \rfloor$. Entries of the array A are determined as follows:

1. $a_{i,j} = s(i-1) + j$ for $1 \le i \le e$, and $1 \le j \le s$;
2. $a_{2e+1-i,j-1} = a_{i,j}$ for $1 \le i \le e$, and $2 \le j \le s$;
3. $a_{2e+1-i,s} = a_{i,1}$ for $1 \le i \le e$.

When $m-t$ is odd, set $m-t = 2e+1$, and $s = \lfloor \frac{\widehat{s}-1}{e} \rfloor$. Entries of the array A are determined as follows:

1. $a_{i,j} = s(i-1) + j$ for $1 \le i \le e$, and $1 \le j \le s$;
2. $a_{e+1,j} = es + 1$ for $1 \le j \le s$;
3. $a_{2e+2-i,j-1} = a_{i,j}$ for $1 \le i \le e$, and $2 \le j \le s$;
4. $a_{2e+2-i,s} = a_{i,1}$ for $1 \le i \le e$.

For example, when $\widehat{s} = 14$ and $m-t = 5$, we find $s = 6$ and $e = 2$, and form the array

$$A = \begin{pmatrix} 2 & 3 & 4 & 5 & 6 & 1 \\ 8 & 9 & 10 & 11 & 12 & 7 \\ 13 & 13 & 13 & 13 & 13 & 13 \\ 7 & 8 & 9 & 10 & 11 & 12 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix}.$$

The symbol 14 is not used in this example, although the orthogonal array provides 14 rows. Using symbol 14 in place of some occurrences of 13 in the third row of A would result in further elementary intervals (with smaller volumes) containing their correct share of the points. There is no need in general to construct the COA explicitly to determine the point positions in the (t, m, s)-net. If an $\mathrm{OA}_\lambda(5, 14, b)$ exists, each of the λb^5 columns determines the position of a point as follows. Let $(d_1, d_2, \ldots, d_{14})^T$ be a column of the OA. Then we use A to select rows from the OA, with the jth column of A determining the coordinate of the point in the jth dimension in the net.

The first coordinate, for example, is $\frac{d_1}{b^1} + \frac{d_7}{b^2} + \frac{d_{13}}{b^3} + \frac{d_8}{b^4} + \frac{d_2}{b^5}$. This provides a construction of COAs from OAs.

There are numerous other constructions for both COAs and (t, m, s)-nets. We do not enumerate them here, but refer the reader to [144, 147, 151] for recent surveys. See [148, 149] for the strongest general necessary conditions.

14 About Things Not Said

The baker's dozen of applications described here provide compelling evidence that combinatorial design theory is finding many diverse connections to practical themes. These in turn suggest new and challenging research problems within design theory. We would be remiss in our duty, however, if we failed to point out that the thirteen topics discussed here are just a sample. Among other important applications are two-point sampling, derandomization, sequences with zero autocorrelation, and quorum systems. Somewhat further afield from our focus here, there are strong connections with sports tournaments and lotteries. See [2, 3, 9] for these and other interesting topics. Our goal has not been to be encyclopedic, so we leave to the reader the enjoyable task of finding further unexpected applications of designs.

References

Background:

[1] Th. Beth, D. Jungnickel & H. Lenz, *Design Theory*, Cambridge University Press, Cambridge (1986).

[2] C.J. Colbourn & J.H. Dinitz (editors), *CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton FL (1996).

[3] C.J. Colbourn & P.C. van Oorschot, Applications of combinatorial designs in computer science, *ACM Computing Surveys*, **21** (1989), pp. 223–250.

[4] C.J. Colbourn & A. Rosa, *Triple Systems*, Oxford University Press, Oxford (1999).

[5] J.H. Dinitz & D.R. Stinson (editors), *Contemporary Design Theory: A Collection of Surveys*, John Wiley & Sons, New York (1992).

[6] D.R. Hughes & F.C. Piper, *Projective Planes*, Springer Verlag, New York (1973).

[7] F.J. MacWilliams & N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam (1978).

[8] D. Raghavarao, *Constructions and Combinatorial Problems in Design of Experiments*, Wiley, New York (1971).

[9] D.R. Stinson, Combinatorial designs and cryptography, in *Surveys in Combinatorics, 1993* Cambridge Univ. Press, London (1993), pp. 257–287.

Optical Orthogonal Codes:

[10] C. Argon & H. Farooq Ahmad, Optimal optical orthogonal code design using difference sets and projective geometry, *Optics Commun.*, **118** (1995), pp. 505-508.

[11] C.M. Bird & A.D. Keedwell, Design and applications of optical orthogonal codes—a survey, *Bull. Inst. Combin. Appl.*, **11** (1994), pp. 21-44.

[12] S. Bitan & T. Etzion, On constructions for optimal optical orthogonal codes, *Lecture Notes in Comput. Sci.*, **781** (1994), pp. 111-125.

[13] M. Buratti, A powerful method for constructing difference families and optimal optical orthogonal codes, *Des. Codes Cryptogr.*, **5** (1995), pp. 13-25.

[14] K. Chen, G. Ge & L. Zhu, Starters and Related Codes, *J. Stat. Plann. Inference*, to appear.

[15] F.R.K. Chung, J.A. Salehi & & V.K. Wei, Optical orthogonal codes: design, analysis and applications, *IEEE Trans. Info. Theory*, **35** (1989), pp. 595-604. Correction: *IEEE Trans. Info. Theory*, **38** (1992), pp. 1429.

[16] J.H. Dinitz & D.R. Stinson, Room squares and related designs, in [5], pp. 137-204.

[17] S.V. Maric & V.K.N. Lau, Multirate fiber-optic CDMA: System design and performance analysis, *J. Lightwave Technol.*, **16** (1998), pp. 9-17.

[18] S.V. Maric, O. Moreno & C. Corrada, Multimedia transmission in fiber-optic LANs using optical CDMA, *J. Lightwave Technol.*, **14** (1996), pp. 2149-2153.

[19] J.A. Salehi, Code division multiple-access techniques in optical fibre networks – part I: Fundamental principles, *IEEE Trans. Info. Theory*, **37** (1989), pp. 824-833.

[20] G.C. Yang, Some new families of optical orthogonal codes for code-division multiple-access fibre-optic networks, *IEEE Proc. Communications*, **142** (1995), pp. 363-368.

[21] J.X. Yin, Some combinatorial constructions for optical orthogonal codes, *Discrete Math.*, **185** (1998), pp. 201-219.

Synchronous Multiple Access to Channels:

[22] G.E. Atkin & H.P. Corrales, An efficient modulation/coding scheme for MFSK systems on bandwidth controlled channels, *IEEE Trans. Sel. Areas Commun.*, **7** (1989), pp. 1396-1401.

[23] G.E. Atkin, D.A. Fares & H.P. Corrales, Coded multipulse position modulation in a noisy optical channel, *Microwave Optical Technol. Lett.*, **2** (1989), pp. 336-340.

[24] C.J. Colbourn, Weakly union-free maximum packings, preprint, University of Vermont, 1998.

[25] C.J. Colbourn & S. Zhao, Maximum Kirkman signal sets for synchronous uni-polar multi-user communication systems, preprint, University of Vermont, 1998.

[26] D.A. Fares, Concatenated coding for multipulse signaling in noisy optical channels, *Microwave Optical Technol. Lett.*, **4** (1991), pp. 359-361.

[27] D.A. Fares, W.H. Abul-Shohoud, N.A. Raslan & M.A. Nassef, $\delta_{\max}$ detection of multipulse signaling in noisy optical channels, *Microwave Optical Technol. Lett.*, **5** (1992), pp. 269-273.

[28] S. Zhao, Application of BIBDs in MT-MFSK signal set design for multiplexing bursty sources, Ph. D. thesis, University of Technology, 1998.

[29] S. Zhao, K.W. Yates & K. Yasukawa, Application of Kirkman designs in joint detection multiple access schemes, *Proc. Int. Symp. Spread Spectrum Techniques and Applications*, **2** (1996), pp. 857-861.

Group Testing and Superimposed Codes:

[30] D.J. Balding & D.C. Torney, Optimal pooling designs with error detection, *J. Combinat. Theory (A)*, **74** (1996), pp. 131–140.

[31] D.J. Balding & D.C. Torney, The design of pooling experiments for screening a clone map, *Fungal Genet. Biol.*, **21** (1997), pp. 302–307.

[32] T. Berger & J.W. Mandell, Bounds on the efficiency of two-stage group testing, preprint, Cornell University, 1998.

[33] T. Berger, N. Mehravari, D. Towsley & J. Wolf, Random multiple-access communications and group testing, *IEEE Trans. Commun.*, **32** (1984), pp. 769–778.

[34] W. J. Bruno, D. J. Balding, E. H. Knill, D. Bruce, C. Whittaker, N. Doggett, R. Stallings & D. C. Torney, Design of efficient pooling experiments, *Genomics*, **26** (1995), pp. 21–30.

[35] Y.M. Chee, C.J. Colbourn & A.C.H. Ling, Weakly union-free twofold triple systems, *Annals Combinat.*, **1** (1997), pp. 215–225.

[36] D. Coppersmith & J.B. Shearer, New bounds for union-free families of sets, *Electron. J. Combinat.*, **5** (1998), pp. #R39.

[37] R. Dorfman, The detection of defective members of a large population, *Ann. Math. Stat.*, **14** (1943), pp. 436–440.

[38] D.Z. Du & F.K. Hwang, *Combinatorial Group Testing and Its Applications*, World Scientific, Singapore (1993).

[39] A. D'yachkov, V. Rykov & A. M. Rashad, Superimposed distance codes, *Problems Control Inform. Theory*, **18** (1989), pp. 237–250.

[40] P. Erdös, P. Frankl & Z. Füredi, Families of finite sets in which no set is covered by the union of two others, *J. Combinat. Theory (A)*, **33** (1982), pp. 158–166.

[41] P. Frankl & Z. Füredi, A new extremal property of Steiner triple systems, *Discrete Math.*, **48** (1984), pp. 205–212.

[42] W. H. Kautz & R. R. Singleton, Nonrandom binary superimposed codes, *IEEE Trans. Information Theory*, **10** (1964), pp. 363–377.

[43] L. Riccio & C. J. Colbourn, An upper bound for disjunct matrices, preprint, University of Vermont, 1998.

[44] M. Ruszinkó, On the upper bound of the size of the r-cover-free families, *J. Combinat. Theory (A)*, **66** (1994), pp. 302–310.

[45] D. R. Stinson, Tran van Trung & R. Wei, Secure frameproof codes, key distribution patterns, group testing algorithms, and related structures, *J. Stat. Plann. Infer.*, in press.

[46] D. R. Stinson & R. Wei, Combinatorial properties and constructions of traceability schemes and frameproof codes, *SIAM J. Discrete Math.*, **11** (1998), pp. 41–53.

[47] F. Vakil & M. Parnes, On the structure of a class of sets useful in nonadaptive group testing, *J. Stat. Plann. Infer.*, **39** (1994), pp. 57–69.

[48] J.K. Wolf, Born again group testing: Multiaccess communications, *IEEE Trans Info. Theory*, **IT-31** (1985), pp. 185–191.

Erasure Codes and Information Dispersal:

[49] A. Albanese, J. Blömer, J. Edmonds, M. Luby & M. Sudan, Priority encoding transmission, *IEEE Trans. Info. Theory*, **42** (1996), pp. 1737–1744.

[50] N. Alon, J. Edmonds & and M. Luby, Linear time erasure codes with nearly optimal recovery, in *Proceedings of the 36th Annual Symposium on Foundations of Computer Science* IEEE, (1995), pp. 512–519.

[51] G.M. Amdahl, Validity of the single processor approach to achieving large scale computing capabilities, in *Proceedings of the 1967 Spring Joint Computer Conference* AFIPS, Washington, D. C. **30** (1967), pp. 483–485.

[52] Y.M. Chee, C.J. Colbourn & and A.C.H. Ling, Asymptotically optimal erasure-resilient codes for large disk arrays, preprint, Univ. of Vermont, 1998.

[53] P.M. Chen, E.K. Lee, G.A. Gibson, R.H. Katz & D.A. Patterson, RAID: High-performance, reliable secondary storage, *ACM Comput. Surveys*, **26** (1994), pp. 145–185.

[54] P. Elias, Coding for two noisy channels, in *Information Theory: Third London Symposium*, Buttersworth, London (1955), pp. 61–76.

[55] P. Erdös, Problems and results in combinatorial analysis, *Creation in Mathematics*, **9** (1976), pp. 25.

[56] G.A. Gibson, *Redundant Disk Arrays: Reliable, Parallel Secondary Storage*, MIT Press, Cambridge, Mass. (1992).

[57] L. Hellerstein, G.A. Gibson, R.M. Karp, R.H. Katz & D.A. Patterson, Coding techniques for handling failures in large disk arrays, *Algorithmica*, **12** (1994), pp. 182–208.

[58] H. Lefmann, P. Pudlák & P. Sacický, On sparse parity check matrices, *Designs Codes Crypt.*, **12** (1997), pp. 107–130.

[59] A.C.H. Ling, C.J. Colbourn, M.J. Grannell & T.S. Griggs, Construction techniques for anti-Pasch Steiner triple systems, preprint, Univ. of Vermont, 1997.

[60] D.A. Patterson & J.L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, Morgan Kaufmann, San Mateo, Ca. (1994).

[61] M.O. Rabin, Efficient dispersal of information for security, load balancing, and fault tolerance, *J. Assoc. Comput. Mach.*, **36** (1989), pp. 335–348.

Threshold and Ramp Schemes:

[62] G.R. Blakley, Safeguarding cryptographic keys, in *Proceedings of the National Computer Conference 1979, American Federation of Information Processing Societies Proceedings 48* (1979), pp. 313–317.

[63] G.R. Blakley and C. Meadows, Security of ramp schemes, *Lecture Notes in Computer Science*, **196** (1985), pp. 242–268.

[64] C. Blundo, A. De Santis & D.R. Stinson, On the contrast in visual cryptography schemes, *Journal of Cryptology*, in press.

[65] T. Hofmeister, M. Krause & H.U. Simon, Contrast-optimal k out of n secret sharing schemes in visual cryptography, *Lecture Notes in Computer Science*, **1276** (1997), pp. 176–185.

[66] W.-A. Jackson & K.M. Martin, A combinatorial interpretation of ramp schemes, *Australasian Journal of Combinatorics*, **14** (1996), pp. 51–60.

[67] K.M. Martin, Discrete Structures in the Theory of Secret Sharing, Ph. D. Thesis, University of London, 1991.

[68] M. Naor & A. Shamir, Visual cryptography, *Lecture Notes in Computer Science*, **950** (1995), pp. 1–12.

[69] A. Shamir, How to share a secret, *Communications of the ACM*, **22** (1979), pp. 612–613.

[70] D.R. Stinson, Visual cryptography and threshold schemes, *Dr. Dobb's Journal*, **April** (1998), pp. 36-43.

[71] D.R. Stinson & S.A. Vanstone, A combinatorial approach to threshold schemes, *SIAM Journal of Discrete Mathematics*, **1** (1988), pp. 230–236.

Authentication Codes:

[72] E. N. Gilbert, F. J. MacWilliams & N. J. A. Sloane, Codes which detect deception, *Bell System Technical Journal*, **53** (1974), pp. 405–424.

[73] K. Kurosawa, K. Okada, H. Saido & D. R. Stinson, New combinatorial bounds for authentication codes and key predistribution schemes, *Designs, Codes and Cryptography*, **15** (1998), pp. 87–100.

[74] J. L. Massey, Cryptography – a selective survey, in *Digital Communications* North-Holland, Amsterdam (1986), pp. 3–21.

[75] R. S. Rees and D. R. Stinson, Combinatorial characterizations of authentication codes II, *Designs, Codes and Cryptography*, **7** (1996), pp. 239–259.

[76] G. J. Simmons, A survey of information authentication, in *Contemporary Cryptology, The Science of Information Integrity* IEEE Press, (1992), pp. 379–419.

[77] D. R. Stinson, The combinatorics of authentication and secrecy codes, *Journal of Cryptology*, **2** (1990), pp. 23–49.

[78] D. R. Stinson, Combinatorial characterizations of authentication codes, *Designs, Codes and Cryptography*, **2** (1992), pp. 175–187.

Resilient and Correlation-immune Functions:

[79] C. H. Bennett, G. Brassard & J. M. Robert, How to reduce your enemy's information, *Lecture Notes in Computer Science*, **218** (1986), pp. 468–476.

[80] J. Bierbrauer, K. Gopalakrishnan & D.R. Stinson, Orthogonal arrays, resilient functions, error correcting codes and linear programming bounds, *SIAM J. Discrete Math.*, **9** (1996), pp. 424–452.

[81] P. Camion, C. Carlet, P. Charpin & N. Sendrier, On correlation-immune functions, *Lecture Notes in Computer Science*, **576** (1992), pp. 86–100.

[82] B. Chor, O. Goldreich, J. Håstad, J. Friedman, S. Rudich & R. Smolensky, The bit extraction problem or t-resilient functions, *26th IEEE Symp. Foundations Comp. Sci.*, (1985), pp. 396-407.

[83] J. Friedman, On the bit extraction problem, *33rd IEEE Symp. Foundations Comp. Sci.*, (1992), pp. 314-319.

[84] K. Gopalakrishnan & D.R. Stinson, Three characterizations of nonbinary correlation-immune and resilient functions, *Designs, Codes and Cryptography*, **5** (1995), pp. 241–251.

[85] T. Siegenthaler, Correlation-immunity of nonlinear combining functions for cryptographic applications, *IEEE Trans. Info. Theory*, **30** (1984), pp. 776–780.

[86] D.R. Stinson, Resilient functions and large sets of orthogonal arrays, *Congressus Numer.*, **92** (1993), pp. 105–110.

[87] D.R. Stinson, On some methods for unconditionally secure key distribution and broadcast encryption, *Designs, Codes and Cryptography*, **12** (1997), pp. 215–243.

Multidrop Networks:

[88] B.E. Aupperle & J.F. Meyer, Fault-tolerant BIBD networks, in *Proc. Eighteenth Int. Sympos. Fault Tolerant Computing* IEEE, (1988), pp. 306–311.

[89] S. Ball, A. Blokhuis & F. Mazzocca, Maximal arcs in Desarguesian planes of odd order do not exist, *Combinatorica*, **17** (1997), pp. 31–41.

[90] A. Barlotti, Su $\{k-n\}$-archi di un piano lineare finito, *Boll. Un. Mat. Ital.*, **11** (1956), pp. 553–556.

[91] J.C. Bermond, J. Bond & S. Djelloul, Dense bus networks of diameter 2, in *Interconnection Networks and Mapping and Scheduling Parallel Computations* (eds. D.F. Hsu, A.L. Rosenberg, and D. Sotteau), American Math. Society, (1994), pp. 9–16.

[92] J.C. Bermond, J. Bond, M. Paoli & C. Peyrat, Graphs and interconnection networks: diameter and vulnerability, in *Surveys in Combinatorics 1983* (ed. E.K. Lloyd), Cambridge University Press, Cambridge (1983), pp. 1–30.

[93] J.C. Bermond, J. Bond & J.F. Saclé, Large hypergraphs of diameter 1, in *Graph Theory and Combinatorics* (ed. B. Bollobás), Academic Press, London (1984), pp. 19–28.

[94] J.-C. Bermond & F.Ö. Ergincan, Bus interconnection networks, *Discrete Applied Math.*, **68** (1996), pp. 1–15.

[95] C.J. Colbourn, Projective planes and congestion-free networks, preprint, Univ. of Vermont, 1998.

[96] R.H.F. Denniston, Some maximal arcs in finite projective planes, *J. Combinat. Theory*, **6** (1969), pp. 317–319.

[97] W.W.M. Dai, Y. Kajitani & Y. Hirata, Optimal single hop multiple bus networks, in *Proc. 1993 IEEE Int. Sympos. Circuits Systems* IEEE, (1993), pp. 2541–2544.

[98] Z. Füredi, Maximum degree and fractional matchings in uniform hypergraphs, *Combinatorica*, **1** (1981), pp. 155–162.

[99] J.W.P. Hirschfeld, *Projective Geometries Over Finite Fields, 2nd Ed.*, Oxford University Press, Oxford (1998).

[100] J.W.P. Hirschfeld & L. Storme, The packing problem in statistics, coding theory and finite projective spaces, Bull. Belg. Math. Soc. Simon Stevin, (to appear).

[101] M.D. Mickunas, Using projective geometry to design bus connection networks, in *Proc. Workshop on Interconnection Networks for Parallel and Distributed Processing* (1980), pp. 47–55.

[102] D. Wiedemann, Cyclic difference covers through 133, *Congressus Numerantium*, **90** (1992), pp. 181–185.

[103] R. Yao, T. Chen & T. Kang, An investigation of multibus multiprocessor systems, *Acta Electron. Sinica*, **18** (1990), pp. 125–127.

[104] B. Yener, Y. Ofek & M. Yung, Combinatorial design of congestion-free networks, *IEEE/ACM Trans. Networking*, **5** (1997), pp. 989–1000.

[105] S.Q. Zheng, Sparse hypernetworks based on Steiner triple systems, in *Proc. 1995 Int. Conf. Parallel Processing* IEEE, (1995), pp. I.92-I.95.

Channel Graphs and Interconnection Networks:

[106] N. Alon, Expanders, sorting in rounds and superconcentrators of limited depth, *Proc. Seventeenth ACM Symposium on the Theory of Computing*, (1985), pp. 98–102.

[107] F.R.K. Chung, Zone-balanced networks and block designs, *Bell Syst. Tech. J.*, **57** (1978), pp. 2957–2981.

[108] F.R.K. Chung, On concentrators, superconcentrators, generalizers and nonblocking networks, *Bell Syst. Tech. J.*, **58** (1979), pp. 1765–1777.

[109] F.R.K. Chung, On switching networks and block designs II, *Bell Syst. Tech. J.*, **59** (1980), pp. 1165–1173.

[110] A. Ghafoor, T.R. Bashkow & I. Ghafoor, Bisectional fault-tolerant communication architecture for supercomputer systems, *IEEE Trans. Computers*, **38** (1989), pp. 1425–1446.

[111] J. Kilian, S. Kipnis & C.E. Leiserson, The organization of permutation architectures with bussed interconnections, *IEEE Trans. Comput.*, **39** (1990), pp. 1346–1358.

Partial Match Queries on Files:

[112] K.A.S. Abdel-Ghaffar, On the number of mutually orthogonal partial latin squares, *Ars Combinat.*, **42** (1996), pp. 259–286.

[113] K.A.S. Abdel-Ghaffar and A. El Abbadi, Optimal disk allocation for partial match queries, *ACM Trans. Database Systems*, **18** (1993), pp. 132-156.

[114] M. Atici, S.S. Magliveras, D.R. Stinson & W.-D. Wei, Some recursive constructions for perfect hash functions, *J. Combinat. Designs*, **4** (1996), pp. 353–363.

[115] G. Berman, The application of difference sets to the design of a balanced multiple-valued filing scheme, *Information and Control*, **32** (1976), pp. 128–138.

[116] R.C. Bose, C.T. Abraham & S.P. Ghosh, File organization of records with multiple-valued attributes for multi-attribute queries, in *Combinatorial Mathematics and Its Applications* (eds. R.C. Bose and T.A. Dowling), UNC Press, Chapel Hil (1969), pp. 277–297.

[117] R.C. Bose & G.G. Koch, The design of combinatorial information retrieval systems for files with multiple-valued attributes, *SIAM Journal on Applied Mathematics*, **17** (1969), pp. 1203–1214.

[118] S.P. Ghosh & C.T. Abraham, Application of finite geometry in file organization for records with multiple-valued attributes, *IBM Journal of Research and Development*, **12** (1968), pp. 180–187.

[119] C. Faloutsos & D. Metaxas, Declustering using error-correcting codes, *Proc. ACM Symp. Principles of Database Systems*, (1989), pp. 253–258.

[120] T. Fujiwara, M. Ito, T. Kasami, M. Kataoka & J. Okui, Performance analysis of disk allocation method using error-correcting codes, *IEEE Trans. Information Theory*, **37** (1991), pp. 379–384.

[121] D.K. Ray-Chaudhuri, Combinatorial information retrieval systems for files, *SIAM Journal on Applied Mathematics*, **16** (1968), pp. 973-992.

[122] R.L. Rivest, Partial-match retrieval algorithms, *SIAM Journal on Computing*, **5** (1976), pp. 19–50.

[123] V.D. Tonchev, Codes, in *CRC Handbook of Combinatorial Designs* (eds. C.J. Colbourn and J.H. Dinitz), CRC Press, Boca Raton FL (1996), pp. 517–542.

Software Testing:

[124] N. Alon, Explicit construction of exponential sized families of k-independent sets, *Discrete Math.*, **58** (1986), pp. 191–193.

[125] M. A. Chateauneuf, C. J. Colbourn & D. L. Kreher, Covering arrays of strength three, *Des. Codes Crypt.*, in press.

[126] D.M. Cohen, S.R. Dalal, M.L. Fredman & G.C. Patton, The AETG system: an approach to testing software based on combinatorial design, *IEEE Trans. Software Engineering*, **23** (1997), pp. 437–444.

[127] D.M. Cohen, S.R. Dalal, J. Parelius & G.C. Patton, The combinatorial design approach to automatic test generation, *IEEE Software*, **13** (1996), pp. 83–88.

[128] D.M. Cohen & M.L. Fredman, New techniques for designing qualitatively independent sets, *J. Combinat. Designs*, in press.

[129] S.R. Dalal & C.L. Mallows, Factor-covering designs for testing software, *Technometrics*, **40** (1998), pp. 234–243.

[130] L. Gargano, J. Körner & U. Vaccaro, Qualitative independence and Sperner problems for directed graphs, *J. Combinat. Theory (A)*, **61** (1992), pp. 173–192.

[131] L. Gargano, J. Körner & U. Vaccaro, Sperner capacities, *Graphs Combinat.*, **9** (1993), pp. 31–46.

[132] S. Poljak & Z. Tuza, On the maximum number of qualitatively independent partitions, *J. Combinat. Theory (A)*, **51** (1989), pp. 111–116.

[133] G. Seroussi & N.H. Bshouty, Vector sets for exhaustive testing of logic circuits, *IEEE Trans. Info. Theory*, **34** (1988), pp. 513–522.

[134] N. J. A. Sloane, Covering arrays and intersecting codes, *J. Combinat. Designs*, **1** (1993), pp. 51–63.

[135] B. Stevens & E. Mendelsohn, New recursive methods for transversal covers, *J. Combinat. Designs*, in press.

Disk Layout and Striping:

[136] G. A. Alvarez, W. A. Burkhard, L. J. Stockmeyer & F. Cristian, Declustered disk array architectures with optimal and near-optimal parallelism, in Proc. 25th ACM/IEEE Int. Sympos. Computer Architecture, Barcelona, Spain (1998),

[137] C.J. Colbourn & K.E. Heinrich, Conflict-free access to parallel memories, *J. Parallel Distrib. Comput.*, **14** (1992), pp. 193–200.

[138] K. Heinrich, K. Kim & V.K. Prasanna Kumar, Perfect Latin squares, *Discrete Appl. Math.*, **37/38** (1992), pp. 281–286.

[139] M. Holland & G.A. Gibson, Architectures and algorithms for on-line failure recovery in redundant disk arrays, *J. Parallel Distributed Databases*, **2** (1994),

[140] F.W. Levi, *Finite Geometrical Systems*, University of Calcutta, Calcutta (1942).

[141] R. Muntz & J. Lui, Performance analysis of disk arrays under failure, in *Proc. Conf. Very Large Data Bases* (1990), pp. 162–173.

[142] E. J. Schwabe & I. M. Sutherland, Improved parity-declustered layouts for disk arrays, *J. Comput. Syst. Sci.*, **53** (1996), pp. 328–343.

(t, m, s)-Nets and Numerical Integration:

[143] P.P. Boyle, M. Broadie & P. Glasserman, Monte Carlo methods for security pricing, *J. Economic Dynamics Control*, in press.

[144] A.T. Clayman, K.M. Lawrence, G.L. Mullen, H. Niederreiter & N.J.A. Sloane, Updated tables of parameters of (t, m, s)-nets, *J. Combinat. Des.*, in press.

[145] A.S. Hedayat, N.J.A. Sloane & J. Stufken, *Orthogonal Arrays: Theory and Practice*, in press.

[146] K.M. Lawrence, A combinatorial characterization of (t, m, s)-nets in base b, *J. Combinat. Designs*, **4** (1996), pp. 275–293.

[147] K.M. Lawrence, A. Mahalanabis, G.L.Mullen & W.C. Schmid, Construction of digital (t,m,s)-nets from linear codes, in *Finite Fields and their Applications* (eds. S. Cohen & H. Niederreiter), Cambridge University Press, Cambridge (1996), pp. 189–208.

[148] W.J. Martin & D.R. Stinson, A generalized Rao bound for ordered orthogonal arrays and (t,m,s)-nets, preprint, University of Winnipeg, 1997.

[149] W.J. Martin & D.R. Stinson, Association schemes for ordered orthogonal arrays and (t,m,s)-nets, preprint, University of Winnipeg, 1997.

[150] G.L. Mullen, A candidate for the "next Fermat problem", *Math. Intelligencer*, **17** (1995), pp. 18–22.

[151] G.L. Mullen, A. Mahalanabis & H. Niederreiter, Tables of (t, m, s)-net and (t, s)-sequence parameters, *Lecture Notes in Statistics*, **106** (1995), pp. 58–86.

[152] G.L. Mullen & W.C. Schmid, An equivalence between (t, m, s)-nets and strongly orthogonal hypercubes, *J. Combinat. Theory (A)*, **76** (1996), pp. 164–174.

[153] G.L. Mullen & G. Whittle, Point sets and sequences with small discrepancy, *Monatshefte Math.*, **113** (1992), pp. 265–273.

[154] H. Niederreiter, Point sets and sequences with small discrepancy, *Monatshefte Math.*, **104** (1987), pp. 273–337.

[155] H. Niederreiter, Orthogonal arrays and other combinatorial aspects in the theory of uniform point distributions in unit cubes, *Discrete Math.*, **106/107** (1992), pp. 361–367.

[156] C.R. Rao, Factorial experiments derivable from combinatorial arrangements of arrays, *J. Royal Stat. Soc.*, **9** (1947), pp. 128–139.

Computer Science, University of Vermont
Burlington, VT 05405, U.S.A.
Charles.Colbourn@uvm.edu

Mathematics and Statistics, University of Vermont
Burlington, VT 05405, U.S.A.
Jeff.Dinitz@uvm.edu

Combinatorics and Optimization, University of Waterloo
Waterloo, Ontario, CANADA N2L 3G1
dstinson@cacr.math.uwaterloo.ca