Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 00-006

New Constructions of Non-Adaptive and Error-Tolerance Pooling Designs

Hung Q. Ngo and Ding-zhu Du

February 04, 2000

New Constructions of Non-Adaptive and Error-Tolerance Pooling Designs

Hung Q. Ngo* Ding-Zhu Du*

Abstract

We propose two new classes of non-adaptive pooling designs. The first one is guaranteed to be d-error-detecting and thus $\lfloor \frac{d}{2} \rfloor$-error-correcting given any positive integer d. Also, this construction induces a construction of a binary code with minimum Hamming distance at least $2d+2$. The second design is the q-analogue of a known construction on d-disjunct matrices.

1 Introduction

The basic problem of group testing is to identify the set of defectives in a large population of items. We assume some testing mechanism exists which if applied to an arbitrary subset of the population gives a *negative outcome* if the subset contains no defective and *positive outcome* otherwise. Objectives of group testing vary from minimizing the number of tests, limiting number of pools, limiting pool sizes to tolerating a few errors. It is conceivable that these objectives are often contradicting, thus testing strategies are application dependent.

Group testing algorithms can roughly be divided into two categories : *Combinatorial Group Testing* (CGT) and *Probabilistic Group Testing* (PGT). In CGT, it is often assumed that the number of defectives among n items is equal to or at most d for some fixed positive integer d. In PGT, we fix some probability p of having a defective. Group testing strategies can also be either *adaptive* or *non-adaptive*. A group testing algorithm is non-adaptive if all tests must be specified without knowing the outcomes of other tests. A group testing algorithm is *error tolerant* if it can detect or correct some e errors in test outcomes. Test errors could be either $0 \to 1$, i.e. a negative pool is identified as positive, or $0 \to 1$ in the contrast.

In this paper, we propose two new classes of non-adaptive and error-tolerance CGT algorithms. Non-adaptive algorithms found its applications in a wide range of practical areas such as DNA library screening [2, 5] and multi-access communications [13], etc.

*Department of Computer Science and Engineering, University of Minnesota, 200 Union street, EE/CS Building, room 4-192, Minneapolis, MN 55455, USA. e-mail: {hngo, dzd}@cs.umn.edu. Support in part by by the National Science Foundation under grant CCR-9530306.

For a general reference on CGT, the reader is referred to a monograph by Du and Hwang [6]. Ngo and Du [11] gave a survey on non-adaptive pooling designs.

The rest of the paper is organized as follows. Section 2 presents basic definitions, notations and related works. Section 3 provides our results and section 4 concludes the paper.

2 Preliminaries

Throughout this paper, for any positive integer v we shall use $[v]$ to denote $\{1, 2, \dots v\}$. Also, given any set X and $k \in \mathbb{N}$ $\binom{X}{k}$ denotes the collection of all k-subsets of X.

2.1 The Matrix Representation

Consider a $v \times n$ 01-matrix M. Let R_i and C_j denote row i and column j respectively. Abusing notation, we also let R_i (resp. C_j) denote the set of column (resp. row) indices corresponding to the 1-entries. The *weight* of a row or a column is the number of 1's it has.

Definition 1 *M is said to be d-*disjunct* if the union of any d columns does not contain another column.*

A d-disjunct $v \times n$ matrix M can be used to design a non-adaptive group testing algorithm on n items by associating the columns with the items and the rows with the pools to be tested. If $M_{ij} = 1$ then item j is contained in pool i (and thus test i). If there are no more than d defectives and the test outcomes are error-free, then it is easy to see that the test outcomes uniquely identify the set of defectives. We simply identify the items contained in negative pools as *negatives* (good items) and the rest as *positives* (defected items). Notice that d-disjunct property implies that each set of $\leq d$ defectives corresponds uniquely to a test outcome vector, thus decoding test outcomes involves only a table lookup. The design of a d-disjunct matrix is thus also naturally called a *non-adaptive pooling design*. We shall use this term interchangeably with the long "non-adaptive combinatorial group testing algorithm".

Let $S(\bar{d}, n)$ denotes the set of all subsets of n items (or columns) with size at most d, called the set of *samples*. For $s \in S(\bar{d}, n)$, let $P(s)$ denote the union of all columns corresponding to s, i.e. $P(s) = \bigcup_{i \in s} C_i$. A pooling design is e-error detecting (correcting) if it can detect (correct) up to e errors in test outcomes. In other words, if a design is e-error detecting then the test outcome vectors form a v-dimensional binary code with minimum Hamming distance at least $e + 1$. Similarly, if a design is e-error correcting then the test outcome vectors form a t-dimensional binary code with minimum Hamming distance at least $2e + 1$. The following remarks are simple to see, however useful later on.

Remark 1 *Suppose M has the property that for any $s, s' \in S(\bar{d}, n), s \neq s'$, $P(s)$ and $P(s')$ viewed as vectors have Hamming distance $\geq k$. In other words, $|P(s) \oplus P(s')| \geq k$*

where $\oplus$ denotes the symmetric difference. Then, M is $(k-1)$-error detecting and $\lfloor \frac{k-1}{2} \rfloor$-error correcting.

Remark 2 *M being d-disjunct is equivalent to the fact that for any set of $d+1$ distinct columns $C_{j_0}, \dots C_{j_d}$ with one column (say C_{j_0}) designated, C_{j_0} has a 1 in some row where all C_{j_k}'s, $1 \le k \le d$ contain 0's.*

2.2 Related Works

Previous works on error-tolerance designs are those of Aigner [1], Muthukrishnan [10], Balding and Torney [3] and Macula [9]. Aigner [1] and Muthukrishnan [10], discussed optimal strategies when $d = 1$ and the number of errors is small, although in a slightly more general setting where each test outcome could be q-ary instead of binary. Balding and Torney [3] studied several instances of the problem when $d \le 2$. In some specific case, they showed that an optimal strategy is possible if and only if certain Steiner system exists. Macula [9] showed that his construction is error-tolerant with high probability.

On construction of disjunct matrices, the most well-known method is to construct the matrix from *set packing designs*. This method was introduced by Kautz and Singleton [7] in the context of superimposed codes. A t-(v,k,λ) packing is a collection $\mathcal{F}$ of k-subsets of $[v]$ such that any t-subset of $[v]$ is contained in at most λ members of $\mathcal{F}$. By limiting $\lambda = 1$, we can construct a d-disjunct matrix from a t-$(v,k,1)$ packing if $k > d(t-1)$. Little is known about optimal set packing designs except for the case $t < 4$ (see, for example, [4, 11] for more details). Besides taking results directly from Design Theory, the only other work known on directly constructing d-disjunct matrices is that of Macula [8].

3 Main Results

We first describe our d-disjunct matrices. Given integers $m \ge k > d \ge 1$. A matching of size l (i.e. it has l edges) is called an l-matching.

Definition 2 *Let $M(m,k,d)$ be the 01-matrix whose rows are indexed by the set of all d-matchings on K_{2m}, and whose columns are indexed by the set of all k-matchings on K_{2m}. All matchings are to be ordered lexicographically. $M(m,k,d)$ has a 1 in row i and column j if and only if the i^{th} d-matching is contained in the j^{th} k-matching.*

For q being a prime power, let F_q denotes $GF(q)$. Let $\binom{\mathrm{F}_q^m}{l}$ denotes the set of all l-dimensional subspaces (l-subspaces for short) of the m-dimensional vector space on F_q.

Definition 3 *Let $M_q(m,k,d)$ be the 01-matrix whose rows (resp. columns) are indexed by elements of $\binom{\mathrm{F}_q^m}{d}$ (resp. $\binom{\mathrm{F}_q^m}{k}$). We also order elements of these set lexicographically. $M_q(m,k,d)$ has a 1 in row i and column j if and only if the i^{th} d-subspace is a subspace of the j^{th} k-subspace of F_q^m.*

We now show that $M(m,k,d)$ and $M_q(m,k,d)$ are d-disjunct.

Theorem 1 *Let $g(m,l) = \binom{2m}{2l}\frac{(2d)!}{2^d d!}$, $v = g(m,d)$, and $n = g(m,k)$. For $m > k > d \geq 1$, $M(m,k,d)$ is a $v \times n$ d-disjunct matrix with row weight $g(m-d,k-d)$ and column weight $g(k,d)$.*

Proof. It is easy to see that $g(m,l)$ is the number of l-matchings of K_{2m}. Thus, $M(m,k,d)$ is a $v \times n$ matrix with row weight $g(m-d,k-d)$ and column weight $g(k,d)$.

To show $M(m,k,d)$ is d-disjunct, we recall Remark 2. Consider $d+1$ distinct columns $C_{j_0}, C_{j_1}, \ldots C_{j_d}$ of $M(m,k,d)$. Since all these columns are distinct k-matchings, for each i, $1 \leq i \leq d$ there exists an edge $e_i \in C_{j_0} \setminus C_{j_i}$. Clearly, there exists a d-matching $R \subset C_{j_0}$ which contains all e_i's. We simply add more edges in C_{j_0} to $\{e_i : 1 \leq i \leq d\}$ if necessary. Since $R \notin C_{j_i}, \forall i \in [d]$, C_{j_0} has a 1 in row R where all other C_{j_i} contains 0.

□

Theorem 2 *Let $\begin{bmatrix} m \\ l \end{bmatrix}_q := \frac{(q^m-1)(q^{m-1}-1)\ldots(q^{m-k+1}-1)}{(q^k-1)(q^{k-1}-1)\ldots(q-1)}$, $v = \begin{bmatrix} m \\ d \end{bmatrix}_q$, and $n = \begin{bmatrix} m \\ k \end{bmatrix}_q$. For $m > k > d \geq 1$, $M_q(m,k,d)$ is a $v \times n$ d-disjunct matrix with row weight $\begin{bmatrix} m-d \\ k-d \end{bmatrix}_q$ and column weight $\begin{bmatrix} k \\ d \end{bmatrix}_q$.*

Proof. It is standard that the Gaussian coefficient $\begin{bmatrix} m \\ l \end{bmatrix}_q$ counts the number of l-subspaces of F_q^m (see, for example, Chapter 24 of [12]). The weight of any column C of $M_q(m,k,d)$ is the number of d-subspaces of C, hence it is $\begin{bmatrix} k \\ d \end{bmatrix}_q$. The weight w of any row R is the number of k-subspaces of F_q^m which contains the d-subspace R. To show $w = \begin{bmatrix} m-d \\ k-d \end{bmatrix}_q$, we employ a standard trick (double counting). Let $I(m,k,d)$ be the number of ordered tuples $(v_1, \ldots, v_{k-d})$ of $k-d$ vectors in $\mathrm{F}_q^m \setminus R$ such that each v_i is not in the span of R and other v_j's, $j \neq i$. Notice that $|\mathrm{F}_q^m| = q^m$ and $|R| = q^d$. Counting $I(m,k,d)$ directly, there are $q^m - q^d$ ways to choose v_1, then $q^m - q^{d+1}$ ways to choose v_2 and so on. Thus,

$$I(m,k,d) = (q^m - q^d)(q^m - q^{d+1})\ldots(q^m - q^{k-1}) \tag{1}$$

Now, $(v_1, \ldots, v_{k-d})$ can be obtained by first picking a k-subspace C of F_q^m which contains R in w ways, then $(v_1, \ldots, v_{k-d})$ is chosen from $C \setminus R$ in $I(k,k,d)$ ways. This yields

$$I(m,k,d) = wI(k,k,d) \tag{2}$$

Combining (1) and (2) gives $w = \begin{bmatrix} m-d \\ k-d \end{bmatrix}_q$ as desired. The fact that $M_q(m,k,d)$ is d-disjunct can be shown in a similar fashion to the previous theorem.

□

The following lemma tells us how to choose k so that the test to item ratio ($\frac{v}{n}$) is minimized. The proof is easy to see and we omit it here.

Lemma 1 *For $l \in [m]$, we have*

(i) The sequence $g(m,l)$ is unimodal and gets its peak at $l = \lfloor m - \sqrt{\frac{m+1}{2}} \rfloor$.

(ii) The sequence $\begin{bmatrix} m \\ l \end{bmatrix}_q$ is unimodal and gets its peak at $l = \lfloor \frac{m}{2} \rfloor$.

The following results further explore properties of $M(m,k,d)$. We first need two more definitions.

Definition 4 *Let $C_0, C_1, \ldots C_d$ be any $d+1$ distinct columns of $M(m,k,d)$. A d-matching R is said to be* private *for C_0 with respect to (wrt for short) $C_1, \ldots C_d$ if $R \in C_0 \setminus \bigcup_{j \geq 1} C_j$. Let $p(C_0; C_1, \ldots C_d)$ denote the number of private d-matchings of C_0 wrt $C_1, \ldots C_d$.*

Theorem 3 *Given $m > d \geq 1$, and any set of $d+1$ distinct columns $C_0, C_1, \ldots C_d$ of $M(m,m,d)$, then $p(C_0; C_1, \ldots C_d) \geq d+1$.*

Proof. Observe that for $1 \leq j \leq d$, $C_0 \cup C_j$ is a loopless multigraph which is 2-regular. $C_0 \cup C_j$ consists of cycles with even lengths. Moreover, $C_0 \neq C_j$ implies that $C_0 \cup C_j$ must have a cycle of length at least $d+1$; consequently, $|C_0 \setminus C_j| \geq 2, \forall j \in [d]$.

Let $E_i = C_0 \setminus C_i$, $i \in [d]$. We can assume $|E_i| = 2, \forall i$. We just remove edges from E_i to reduce its size to be exactly 2 for the ease of analysis. Let G be the graph with $V(G) = C_0$, $E(G) = \{E_1, \ldots E_d\}$. Then, G is a simple graph having m vertices and $\leq d$ edges. $|E(G)| \leq d$ because the E_i's are not necessarily distinct. Any d-subset R of C_0 such that $R \cap E_i \neq \emptyset, \forall i$ is a private d-matching of C_0. Note that R is nothing but a vertex cover of size d (d-cover for short) of G. To show $p(C_0; C_1, \ldots C_d) \geq d+1$, we shall show that the number of d-covers of G is at least 4. Since adding more edges into G can only decrease the number of d-covers, we also assume that G has exactly d edges.

Decompose G into its connected components. Suppose $G_1, \ldots, G_x$ are connected components which are not trees, and $G'_1, \ldots, G'_y$ are the rest of the components. Isolated points are also considered to be trees. For $i = 1, \ldots x$, let $v_i = |V(G_i)|$ and $e_i = |E(G_i)|$. For $i = 1, \ldots y$, let $v'_i = |V(G'_i)|$ and $e'_i = |E(G'_i)|$. The following equations are straight from the definitions :

$$\sum_{i=1}^{x} v_i + \sum_{i=1}^{y} v'_i = m \qquad (3)$$

$$\sum_{i=1}^{x} e_i + \sum_{i=1}^{y} e'_i = d \qquad (4)$$

hence,

$$0 \leq \sum_{i=1}^{x} e_i - \sum_{i=1}^{x} v_i = y - (m - d) \tag{5}$$

Note that for any connected simple graph H, picking any $|V(H)| - 1$ vertices out of $V(H)$ gives us a vertex cover. Hence, the number of $(|V(H)| - 1)$-covers of H is at least $\binom{|V(H)|}{|V(H)|-1} = |V(H)|$. Now, a d-cover of G could be formed by two methods as follows.

(a) *Method 1*. For each $i \in [x]$, pick in v_i ways a $(v_i - 1)$-cover for G_i, then cover all other G_j, $j \neq i$ with all their vertices. We have used up $(\sum_{i=1}^{x} v_i) - 1$ vertices, and need $d - (\sum_{i=1}^{x} v_i) + 1$ more to cover all G_i''s. Firstly, there should be enough vertices. Indeed,

$$\sum_{i=1}^{y} v_i' = m - \sum_{i=1}^{x} v_i \geq d + 1 - \sum_{i=1}^{x} v_i$$

Secondly, to cover all G_i''s, we need at most $\sum_{i=1}^{y}(v_i' - 1)$ vertices. (3) and (5) assure that

$$\sum_{i=1}^{y}(v_i' - 1) = m - \sum_{i=1}^{x} v_i - y < d + 1 - \sum_{i=1}^{x} v_i$$

In conclusion, this method gives us at least $(\sum_{i=1}^{x} v_i)$ d-covers for G.

(b) *Method 2*. This time, we are greedier by first taking all vertices in G_i's, $i \in [x]$ to cover them. $a = d - \sum_{i=1}^{x} v_i$ vertices are needed to cover the rest. These a vertices can be chosen as follows. For each $(m - d)$-subset Y of $[y]$, cover each $G_i', i \in Y$ with $v_i' - 1$ vertices. Cover each $G_i', i \notin Y$ using all of its vertices. Indeed, the total number of vertices used is

$$\sum_{i \in Y}(v_i' - 1) + \sum_{i \notin Y} v_i' = \sum_{i \in [y]} v_i' - |Y| = (m - \sum_{i=1}^{x} v_i) - (m - d) = a$$

Moreover, obviously there are at least $\prod_{i \in Y} v_i'$ ways to pick d-covers for each particular Y. In total, noticing that $y \geq m - d \geq 1$, method 2 gives us at least

$$
\begin{aligned}
\sum_{Y\in\binom{[y]}{m-d}}\prod_{i\in Y} v'_i &= \sum_{Y\in\binom{[y]}{m-d}}\prod_{i\in Y}(e'_i+1) \\
&\geq \sum_{i=1}^{y} e'_i + \binom{y}{m-d} \\
&\geq (\sum_{i=1}^{y} v'_i - y) + (y-m+d+1) \\
&= d+1-\sum_{i=1}^{x} v_i
\end{aligned}
$$

Hence, Methods 1 and 2 combined yields at least $(d+1)$ private d-matchings of C_0.

□

Corollary 1 *Given $m > d \geq 1$, the following holds :*

(i) $M(m,m,d)$ *is d-error detecting and $\lfloor\frac{d}{2}\rfloor$-error correcting.*

(ii) Moreover, if the number of defectives is known to be exactly d, then $M(m,m,d)$ is $2d+1$-error-detecting and d-error-correcting.

Proof. For any $s, s' \in S(\bar{d}, n), s \neq s'$, without loss of generality we can assume there exists $C_0 \in s \setminus s'$. Theorem 3 implies $|P(s) \oplus P(s')| \geq d+1$, hence Remark 1 shows (i). If the number of defectives is exactly d, we need to only consider $|s| = |s'| = d$; hence there exists $C_0 \in s \setminus s'$ and $C'_0 \in s' \setminus s$. Again, Theorem 3 and Remark 1 yields (ii).

□

Corollary 2 *Given $m > d \geq 1$, then there exists a binary d-error-correcting code of dimension $g(m,d)$ and size $\binom{g(m,m)}{d}$.*

Proof. The code can be constructed by taking all the unions of d columns in $M(m,m,d)$.

□

Borrowing an idea from Macula [9], we get the following algorithm which uses $M(m,k,2)$ for the at most d defective problem, and show that with very high probability, our algorithm gives the correct answer. Notice that each row of $M(m,k,2)$ is a 2-matching consisting of some two parallel edges $(e1, e2)$ of K_{2m}.

Algorithm 1 *Use $M(m,k,2)$ to design the pools as usual. For each edge $e \in E(K_{2m})$ such that the total number of positive outcomes involving e is $k-1$, i.e. $|\{(e,x) :$ the test (e,x) is positive $\}| = k-1$, identify the item $C = \{e\} \cup \{x : (e,x)$ is positive $\}$ as a defective.*

Theorem 4 *Algorithm 1 gives correct answer with probability* $p(m,k,d)$ *where*

$$p(m,k,d) \geq \left[\frac{\sum_{j=1}^{k}(-1)^{j+1}\binom{k}{j}\binom{\sum_{i=0}^{j}(-1)^{i}\binom{j}{i}g(m-i,k-i)}{d-1}}{\binom{g(m,k)-1}{d-1}} \right]^{d}$$

For example, $p(8,6,9) \geq 98.5\%$, which means that we could use $M(8,6,2)$ to solve the at most 9 defectives problem with 98.5% of success.

Proof. Given a set of d distinct columns $C_1, C_2, \ldots C_d$. $e \in E(K_{2m})$ is called a mark of C_i if $e \in C_i$ but $e \notin C_j, j \neq i$, in which case C_i is called marked. If C_i is marked by e then exactly $k-1$ tests involving e and another edge in C_i is positive. Thus, algorithm 1 gives correct answer if the set of d defectives is a marked set, i.e. every element is marked.

The probability that algorithm 1 gives correct answer is thus the probability that a random d set of columns of $M(m,k,2)$ is marked. For a fixed C_1, there are $\binom{g(m,k)-1}{d-1}$ ways to pick the other $d-1$ columns. Let E_i be the event that C_i is marked relative to the other $d-1$ columns, then

$$p(m,k,d) = P(E_1)P(E_2|E_1)P(E_3|E_1,E_2)\ldots \geq (P(E_1))^d$$

To calculate $P(E_1)$, we count number of ways to pick $d-1$ columns other than C_1 such that C_1 is marked by some $e \in C_1$. Let A_i be the collection of all $d-1$-set of columns other than C_1 such that $e_i \in C_1$ marks C_1 with respect to A_i. The answer is then $|\bigcup\{A_i, 1 \leq i \leq k\}|$. This number can be obtained by applying inclusion-exclusion principle twice. Dividing it by $\binom{g(m,k)-1}{d-1}$ gives us $P(E_1)$ and proves the theorem.

$\square$

4 Discussions

We have given the constructions of two different classes of pooling designs. $M(m,k,d)$ has good performance when the number of defectives is rare comparing to the number of items. Deterministically, a larger ratio of defectives to items is sometime preferred. Probabilistically, however, $M(m,k,2)$ could be used to solve the $S(\bar{d},n)$ problem with very high probability of success. The main strength of this construction is that $M(m,m,d)$ is d-error-detecting. It also yields the construction of a d-error-correcting code. $M_q(m,k,d)$ is the q-analogue of the construction given by Macula [8]. An interesting question is: "what is the q-analogue of a matching?"

One could think of several different variations of the matching idea. For example, a possible generalization is to index the rows (resp. columns) of a matrix $M(m,k,d,l)$ with all graphs having d (resp. k) edges whose vertex degrees are $\leq l$. $M(m,k,d)$ is nothing but $M(m,k,d,1)$. Further investigations in this direction might lead to better designs.

Lastly, in reality given a specific problem with certain parameters, m and k has to be chosen appropriately to suit one's need. More careful analysis need to be done to help pick the *best* m and k given n, d and/or any other constraints from practice. We need some reasonably good asymptotic formulas to estimate them.

References

[1] M. AIGNER, *Searching with lies*, J. Combin. Theory Ser. A, 74 (1996), pp. 43–56.

[2] D. J. BALDING, W. J. BRUNO, E. KNILL, AND D. C. TORNEY, *A comparative survey of non-adaptive pooling designs*, in Genetic mapping and DNA sequencing (Minneapolis, MN, 1994), Springer, New York, 1996, pp. 133–154.

[3] D. J. BALDING AND D. C. TORNEY, *Optimal pooling designs with error detection*, J. Combin. Theory Ser. A, 74 (1996), pp. 131–140.

[4] T. BETH, D. JUNGNICKEL, AND H. LENZ, *Design theory*, Cambridge University Press, Cambridge, 1986.

[5] W. J. BRUNO, E. KNILL, D. J. BALDING, D. C. BRUCE, N. A. DOGGETT, W. W. SAWHILL, R. L. STALLINGS, C. C. WHITTAKER, AND D. C. TORNEY, *Efficient pooling designs for library screening*, Genomics, (1995), pp. 21–30.

[6] D. Z. DU AND F. K. HWANG, *Combinatorial group testing and its applications*, World Scientific Publishing Co. Inc., River Edge, NJ, 1993.

[7] W. H. KAUTZ AND R. C. SINGLETON, *Nonrandom binary superimposed codes*, IEEE Trans. Inf. Theory, 10 (1964), pp. 363–377.

[8] A. J. MACULA, *A simple construction of d-disjunct matrices with certain constant weights*, Discrete Math., 162 (1996), pp. 311–312.

[9] ———, *Probabilistic nonadaptive group testing in the presence of errors and dna library screening*, Annals of Combinatorics, (1999), pp. 61–69.

[10] S. MUTHUKRISHNAN, *On optimal strategies for searching in presence of errors*, in Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (Arlington, VA, 1994), New York, 1994, ACM, pp. 680–689.

[11] H. Q. NGO AND D. Z. DU, *A survey on combinatorial group testing algorithms with applications to dna library screening*, in Proceedings of the DIMACS Workshop on Discrete Mathematical Problems and Medical Applications, Providence, RI, 1999, Amer. Math. Soc.

[12] J. H. VAN LINT AND R. M. WILSON, *A course in combinatorics*, Cambridge University Press, Cambridge, 1992.

[13] J. K. WOLF, *Born again group testing: multiaccess communications*, IEEE Transaction on Information Theory, IT-31 (1985), pp. 185–191.