Discrete Applied Mathematics 157 (2009) 164–169

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

The minimum number of e-vertex-covers among hypergraphs with e edges of given ranks

F.H. Chang [a,*], H.L. Fu [b], F.K. Hwang [b], B.C. Lin [c]

[a] Department of Applied Mathematics, National Chiayi University, Chiayi City, 60004, Taiwan
[b] Department of Applied Mathematics, National Chiao Tung University, Hsinchu, 30050, Taiwan
[c] Department of Math, National Central University, Chung-Li, 32054, Taiwan

ARTICLE INFO

Article history:
Received 25 May 2004
Received in revised form 28 August 2007
Accepted 21 May 2008
Available online 30 June 2008

Keywords:
Vertex-cover
Hypergraph
Pooling design

ABSTRACT

We study the problem that among all hypergraphs with e edges of ranks $l_1, \dots, l_e$ and v vertices, which hypergraph has the least number of vertex-covers of size e. The problem is very difficult and we only get some partial answers. We show an application of our results to improve the error-tolerance of a pooling design proposed in the literature.

1. Introduction

Let H(V, E) denote a hypergraph with vertex-set V and edge set $E = \{E_1, \dots, E_e\}$. Let l_i denote the rank of E_i, i.e. $l_i = |E_i|$. Then $l_i = 1$ is allowed. A subset $V' \subseteq V$ is called a *vertex-cover* of H if it intersects every $E_i \in E$. It is called a *d-vertex-cover* if $|V'| = d$. Let f_d (H) denote the number of *d-vertex-covers* in H.

Define

$$H(v; l_1, \dots, l_e) = \{H(V, E) : |V| = v, |E| = e, |E_i| = l_i, i = 1, \dots, e\}$$

and

$$f_d(v; l_1, \dots, l_e) = \min\{f_d(H) : H \in H(v; l_1, \dots, l_e)\}.$$

We study $f_d(v; l_1, \dots, l_e)$ in this paper.

This problem is motivated by the construction of error-detecting and error-correcting pooling designs (used in clone library screening). Ngo and Du [4] observed that the number of *e-vertex-covers* has bearing on the error-correcting ability of a pooling design.

2. The hypergraph case

A vertex is called an isolated vertex if it is not in any edge. A hypergraph H(V, E) is called optimal in $H(v; l_1, \dots, l_e)$ if it achieves $f_d(v; l_1, \dots, l_e)$.

Lemma 2.1. *For any given* $H(v; l_1, \dots, l_e)$*, there exists an* H^* *such that* H^* *does not have both an isolated vertex and a vertex shared by two edges.*

* Corresponding author. Tel.: +886 5 2717917.
E-mail addresses: fei@mail.ncyu.edu.tw (F.H. Chang), hlfu@math.nctu.edu.tw (H.L. Fu), fkhwang@gmail.com (F.K. Hwang), beychi.lin@gmail.com (B.C. Lin).

0166-218X/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2008.05.006

Proof. Let H be a hypergraph in $H(v; l_1, \ldots, l_e)$ with an isolated vertex i and a vertex j shared by two edges a and b. Let H' be obtained from H by eliminating i and splitting j into two vertices j_a and j_b such that $j_a \in a$ and $j_b \in b$, i.e., a and b no longer share the vertex j. We show $f_e(H') \le f_e(H)$ by mapping all *e-vertex-covers* C' of H' to distinct *e-vertex-covers* C of H.

We consider several cases:

(i) C' contains neither j_a nor j_b. Set $C' = C$.
(ii) C' contains both j_a and j_b. C is obtained by replacing j_a with j and j_b with i.
(iii) C' contains j_a but not j_b. C is obtained by replacing j_a with j.
(iv) C' contains j_b but not j_a, and no other vertex of b. C is obtained by replacing j_b with j.
(v) C' contains j_b and another vertex of b, but no j_a. C is obtained by replacing j_b with i.

Clearly, C is an *e-vertex-cover* of H. To check that the mapping is injective, it is obvious that the only time two distinct *e-vertex-covers* C' and C'' of H' can map to the same C is when C' and C differ only in one vertex, i.e., $C' = S \cup \{j_a\}$, when S is a set of $e-1$ vertices containing neither j_a nor j_b. Consider C'', S must contain a vertex of a other than j_a. Hence S contains both a vertex of a and a vertex of b. From our mapping rule, C' will be mapped to $S \cup \{j\}$, while C'' to $S \cup \{i\}$, two distinct *e-vertex-covers* in H.

By repeating this procedure, and since $|H(v; l_1, \ldots, l_e)|$ is finite, eventually we reach a hypergraph which has either no isolated vertices or no vertices shared by two edges. Since this argument holds for all $H(v; l_1, \ldots, l_e)$, Lemma 2.1 follows. ■

Define $l = \sum_{i=1}^{e} l_i$ and $L = \prod_{i=1}^{e} l_i$. For $v \ge l$, from Lemma 2.1, $f_e(v; l_1, \ldots, l_e)$ is nondecreasing in v. We next show that it reaches maximum at a certain v.

Theorem 2.2. *$f_e(v; l_1, \ldots, l_e) = L$ for $v \ge l$.*

Proof. Since any $H \in H(v; l_1, \ldots, l_e)$ with two intersecting edges must have an isolated vertex, by Lemma 2.1, there exists an optimal hypergraph H^* with no intersecting edges, i.e., H^* contains e disjoint edges of ranks $l_1, \ldots, l_e$ and $v - l$ isolated vertices. Clearly, H^* has L *e-vertex-covers*. ■

Theorem 2.3. *$f_e(l-1; l_1, \ldots, l_e) = \prod_{i \ne 1,2} l_i \left[l_1 l_2 - 1 + \frac{1}{2} \sum_{i \ne 1,2} (l_i - 1) \right]$.*

Proof. Without loss of generality, assume $l_1 \ge l_2 \ge \cdots \ge l_e$. By Lemma 2.1, it suffices to consider H with no isolated vertices.

Since $v = l - 1$, H contains exactly two edges intersecting in one vertex. Let H_{mn} denote the hypergraph where E_m and E_n intersect, and $f_e(H_{mn})$ its number of *e-vertex-covers*. Then

$$\begin{aligned} f_e(H_{mn}) &= [(l_m - 1)(l_n - 1) + l_m + l_n - 2] \prod_{i \ne m,n} l_i + \sum_{k \ne m,n} \binom{l_k}{2} \prod_{i \ne m,n,k} l_i \\ &= (l_m l_n - 1) \prod_{i \ne m,n} l_i + \frac{1}{2} \sum_{k \ne m,n} l_k (l_k - 1) \prod_{i \ne m,n,k} l_i \\ &= (l_m l_n - 1) \prod_{i \ne m,n} l_i + \frac{1}{2} \sum_{i \ne m,n} (l_i - 1) \prod_{i \ne m,n} l_i \\ &= \prod_{i \ne m,n} l_i \left[l_m l_n - 1 + \frac{1}{2} \sum_{i \ne m,n} (l_i - 1) \right]. \end{aligned}$$

If $l_1 = l_2 = \cdots = l_e$, then Theorem 2.3 obviously holds; if not, we prove:
Suppose $l_y > l_z$. Then $f_e(H_{xy}) < f_e(H_{xz})$ for all other $x \in \{1, \ldots, e\}$.

$$\begin{aligned} f_e(H_{xy}) &= \prod_{i \ne x,y} l_i \left[l_x l_y - 1 + \frac{1}{2} \sum_{i \ne x,y} (l_i - 1) \right] \\ &= \prod_{i \ne x,y,z} l_i \left[l_x l_y l_z - l_z + \frac{1}{2} l_z (l_z - 1) + \frac{1}{2} l_z \sum_{i \ne x,y,z} (l_i - 1) \right] \\ &= \prod_{i \ne x,y,z} l_i \left[l_x l_y l_z + \frac{1}{2} l_z (l_z - 3) + \frac{1}{2} l_z \sum_{i \ne x,y,z} (l_i - 1) \right] \\ &< \prod_{i \ne x,y,z} l_i \left[l_x l_y l_z + \frac{1}{2} l_y \left(l_y - 3\right) + \frac{1}{2} l_y \sum_{i \ne x,y,z} (l_i - 1) \right] \\ &= f_e(H_{xz}). \end{aligned}$$

Theorem 2.3 follows immediately. ■

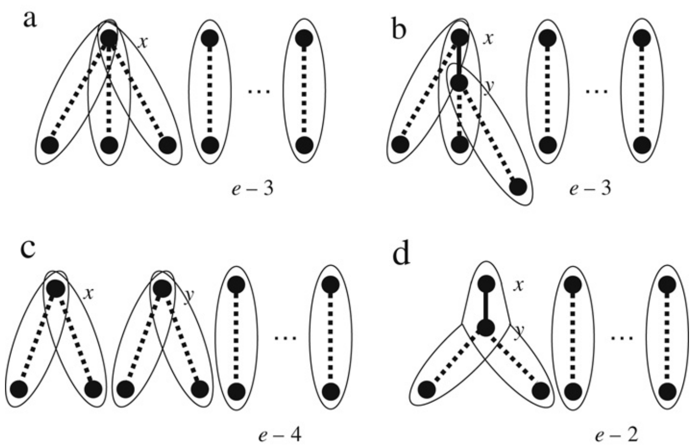

Fig. 1. Graphs with $v = l - 2$ vertices and e edges.

For $v = l - 2$, we have to restrict our attention to the case $l_1 = l_2 = \cdots = l_e = k$. Assume $e \geq 4$, then there are four hypergraphs as shown in Fig. 1.

We count the number of *e-vertex-covers* for each of them.

Lemma 2.4. *For $v = l - 2, l_1 = l_2 = \cdots = l_e = k$, the number of e-vertex-covers for the graphs in Fig. 1 is listed respectively as follows:*

(i)

$$f_e((a)_k) = (k-1)^3 k^{e-3} + \left\{ \binom{3k-3}{2} k^{e-3} + \binom{3k-3}{1} \binom{e-3}{1} \binom{k}{2} k^{e-4} + \binom{3k-3}{0} \left[\binom{e-3}{1} \binom{k}{3} k^{e-4} + \binom{e-3}{2} \binom{k}{2}^2 k^{e-5} \right] \right\}.$$

(ii)

$$f_e((b)_k) = (k-1)^2 (k-2) k^{e-3} + 2 \left\{ \left[\binom{3k-4}{2} - \binom{2k-3}{2} \right] k^{e-3} + \binom{k-1}{1} \binom{e-3}{1} \binom{k}{2} k^{e-4} \right\} + \left[\binom{3k-4}{1} k^{e-3} + \binom{3k-4}{0} \binom{e-3}{1} \binom{k}{2} k^{e-4} \right].$$

(iii)

$$f_e((c)_k) = (k-1)^4 k^{e-4} + 2 \left\{ \left[2 \binom{k-1}{2} \binom{k-1}{1} + \binom{k-1}{1}^2 \binom{2k-2}{1} \right] k^{e-4} + \binom{k-1}{1}^2 \binom{e-4}{1} \binom{k}{2} k^{e-5} \right\} + \left\{ \binom{4k-4}{2} k^{e-4} + \binom{4k-4}{1} \binom{e-4}{1} \binom{k}{2} k^{e-5} + \binom{4k-4}{0} \left[\binom{e-4}{1} \binom{k}{3} k^{e-5} + \binom{e-4}{2} \binom{k}{2}^2 k^{e-6} \right] \right\}.$$

(iv)

$$f_e((d)_k) = (k-2)^2 k^{e-2} + 2 \left[\binom{2k-4}{1} k^{e-2} + \binom{2k-4}{0} \binom{e-2}{1} \binom{k}{2} k^{e-3} \right] + k^{e-2}.$$

Proof. We start with the proof of (i) for Fig. 1(a). Clearly, if an e-vertex-cover does not contain the vertex x, then each of the three edges incident to x has $k - 1$ choices as vertex covers and all the other edges have k choices. Thus, the number of e-vertex-covers is equal to $(k-1)^3 k^{e-3}$ which gives the first term. On the other hand, if an e-vertex-cover does contain x, then it contains at most two other vertices in the x-tree since we have $e-2$ components. So, the second term can be obtained by considering the number of extra vertices in the x-tree (different from x) which are chosen for the e-vertex-cover. Now, it is not difficult to see that if the e-vertex-cover contains three vertices of the x-tree, then it contains exactly one vertex of the other $e-3$ edges; if the e-vertex-cover contains two vertices of the x-tree, then one of the remaining $e-3$ edges contains exactly two vertices and each of the other $e-4$ edges contains exactly one vertex; and if the e-vertex-cover contains only x in the x-tree, then we have choices of $(3, 1, 1, \ldots, 1)$ or $(2, 2, 1, \ldots, 1)$ for the other $e-3$ edges. Therefore, we have the second term.

Following the same line of reasoning, in each of (ii), (iii), (iv), the terms are broken down into taking none of x, y, one of them, and both of them. It is worth of noting that in (ii) $\binom{3k-4}{2} - \binom{2k-3}{2}$ represents the case that, assuming x is taken but not y, then two other vertices are taken from the (x, y)-tree, at least one of them from the edge not incident to x; and in (iii) the last three terms represent taking both x and y, and 2 or 1 or 0 other vertices from the x-tree and the y-tree. ■

Theorem 2.5. *$f_e(l-2, k^e) = f_e((b)_k)$ for $e \geq 4$ and $k \geq 2$.*

Proof. Let $f_e(l-2, k^e)$ denote the case that all e edges have length k. By using MAPLE, we obtain

$$\begin{aligned} f_e((\mathrm{a})_k) - f_e((\mathrm{b})_k) &= k^{e-3}\left[3(k-1)^2 e^2 - (5k+11)(k-1)e + 12(k+1)\right]/24 \\ &= k^{e-3}\{[3(k-1)e - (2k+8)][(k-1)e - (k+1)] - 2(k-2)(k+1)\}/24. \end{aligned}$$

$f_e((\mathrm{a})_k) - f_e((\mathrm{b})_k)$ is clearly increasing in e. So it suffices to prove $f_e((\mathrm{a})_k) - f_e((\mathrm{b})_k) \geq 0$ for $e > 4$.

For $e = 4$,

$$\begin{aligned} f_e((\mathrm{a})_k) - f_e((\mathrm{b})_k) &= k[10(k-2)(3k-5) - 2(k-2)(k+1)]/24 \\ &= k(k-2)[5(3k-5) - (k+1)]/12 \\ &= k(k-2)(7k-13)/6 \geq 0 \quad \text{for } k \geq 2. \\ f_e((\mathrm{c})_k) - f_e((\mathrm{b})_k) &= k^{e-4}\left\{(k-1)\left[3(k-1)e^2 - (11k+5)e + 4(2k+5)\right] + 24\right\}/24 \\ &= k^{e-4}\{(k-1)[[3(k-1)e - 4(2k+5)](e-1) + 12e] + 24\}/24 \\ &= k^{e-4}\{(k-1)[[3(k-1)e - 4(2k+5) + 12](e-1) + 12] + 24\}/24 \\ &= k^{e-4}\{(k-1)[[3(k-1)e - 8(k+1)](e-1) + 12] + 24\}/24. \end{aligned}$$

$f_e((\mathrm{c})_k) - f_e((\mathrm{b})_k)$ is clearly increasing in e. So it suffices to prove $f_e((\mathrm{c})_k) - f_e((\mathrm{b})_k) \geq 0$ for $e > 4$.

For $e = 4$,

$$\begin{aligned} f_e((\mathrm{c})_k) - f_e((\mathrm{b})_k) &= \{(k-1)[[12(k-1) - 8(k+1)] \cdot 3 + 12] + 24\}/24 \\ &= [(k-1)(12k-48) + 24]/24 \\ &= \left(k^2 - 5k + 6\right)/2 \\ &= (k-2)(k-3)/2 \geq 0 \quad \text{for } k \geq 2. \\ f_e((\mathrm{d})_k) - f_e((\mathrm{b})_k) &= k^{e-3}(ke - e - k - 1)/2 \geq 0. \quad ■ \end{aligned}$$

While we have no result on general $f_e(v; l_1, \ldots, l_e)$, the following lemma helps us to obtain lower bounds.

Lemma 2.6. *Suppose $l_i \geq l'_i$ for $1 \leq i \leq e$. Then $f_e(v; l_1, \ldots, l_e) \geq f_e(v; l'_1, \ldots, l'_e)$.*

In particular, $f_e(v; l_1, \ldots, l_e) \geq f_e(v, k^e)$ if $l_i \geq k$ for $1 \leq i \leq e$.

Lemma 2.7. *Let C be a vertex-cover of G such that $|C| = c < e$ and S be the set of vertices in G such that each vertex is incident to at least two edges of G. Let $|E_i \setminus S| = l'_i$, $i = 1, 2, \ldots, e$. Then $f_e(v; l_1, \ldots, l_e) \geq \binom{v-c}{e-c} + \prod l'_i$.*

Proof. $\binom{v-c}{e-c}$ represents the number of e-covers which contains C and $\prod l'_i$ is the number of e-covers which are different from the above e-covers (since $c < e$, C contains at least one vertex in S). ■

Proposition 2.8. *If $e(k-1) + 1 \leq v < ek$, then $f_e(v, k^e) \geq (k-2)k^{e-3}$.*

Proof. Since $|S| \leq k-1$,

$$\sum_{i=1}^{e} l'_i \geq k(e-2).$$

Hence

$$\prod_{l'_i > 0} l'_i \geq 1 \cdot 1 \cdot (k-2)\,k^{e-3}. \quad ■$$

3. A bound for the graph case

For the graph case $l_i = 2$ for all i; hence no isolated vertex can be an edge. We will write G instead of H for a graph. In particular, $H(v; l_1, \ldots, l_e)$ will be written as $G(v, e)$ and $f_d(v; l_1, \ldots, l_e)$ as $f_d(v, e)$. Theorems 2.2 and 2.3 then yield $f_e(v, e) = 2^e$ for $v \geq 2e$ and $f_e(2e-1, e) = 2^{e-1} + e2^{e-3}$.

Further, we have

Lemma 3.1. $f_e(e+1, e) = e+1$.

Proof. Since an edge has two vertices. Any set of e vertices must be a *vertex-cover* as it leaves at most one vertex out in an edge. Clearly, there are $e+1$ sets of e vertices. ■

For the general case, we give a lower bound.

Theorem 3.2. $f_e(v, e) \geq 2^{v-e-1}(2e - v + 2)$ *for* $e + 1 \leq v \leq 2e - 1$.

Proof. By Lemma 2.1, it suffices to consider graphs with no isolated vertex. Suppose such a graph G has c components. Then $c \geq v - e$, where equality prevails only when each component is a tree.

Let component C_i have v_i vertices and e_i edges. Then any choice of $v_i - 1$ vertices is a *vertex-cover* of C_i. Further, any G has at least $v - e$ trees. Fix $v - e - 1$ trees, say, $C_1, \ldots, C_{v-e-1}$ of G and let G' consist of the remaining components (G' is a tree if G is a forest) with v' vertices. Then any choice of $v' - 1$ vertices of G' is a *vertex-cover* of G', and there are v' such set. By taking $v_i - 1$ vertices from each C_i, and $1 \leq i \leq v - e - 1$, and $v' - 1$ vertices of G', we obtain an *e-vertex-cover* of G, and there are $v' \prod_{i=1}^{v-e-1} v_i$ of them, with $\sum_{i=1}^{v-e-1} v_i + v' = v$.

Since for $a < b$, $ab \leq (a+1)(b-1)$, $v' \prod_{i=1}^{v-e-1} v_i$ is minimized by the most uneven distribution of v_i, namely, all $v_i = 2$ except one $v_i = 2e - v + 2$. ■

Note that for $v = e+1$, this bound yields a value of $2^0(e+1) = e+1$, which matches $f_e(e+1, e)$ as shown in Lemma 3.1. But for $v - 1 > 1$, the bound can certainly be strengthened by allowing some trees or G' to have all their vertices taken and other C_i with $v_i \geq 3$ to have more than one vertex not taken. In particular, we have

Corollary 3.3. *Suppose* $v < 2e$. *Then* $f_e(v, e) \geq 2^{v-e-1}\,(2e - v + 2) + \sum_{x=\min\{e+1-\lceil v/2 \rceil, 3e-2v+2\}}^{2e-v} 2^{x-(3e-2v+2)}$.

Proof. It is easily verified that a connected graph with n vertices has an *x-vertex-cover* for every $x \geq \lfloor n/2 \rfloor$. Since G' has $2e-v+2$ vertices, it has an *x-vertex-cover* for every $e+1-\lceil v/2 \rceil \leq x \leq 2e-v$. This *x-vertex-cover* plus one vertex from each of the $v-e-1$ 2-trees constitute a $(x+v-e-1)$-*vertex-cover* for G. Suppose $e-(x+v-e-1) = 2e-x-v+1 \leq v-e-1$, or $3e - 2v + 2 \leq x$. Then we can choose two vertices from $2e - x - v + 1$ 2-trees and one from the rest 2-trees to obtain an *e-vertex-cover* of G. The number of such choices is $2^{x-(3e-2v+2)}$. ■

4. An application

A long DNA molecule M is often cut into short segments called *clones* for easy storage and reproduction. Typically, it is cut more than once, with each cutting having independent cutpoints, to facilitate reconstruction of M. One approach of reconstruction is to make use of many sequence-tagged-sites (STS) each is assumed to have a unique appearance in M. By identifying for each clone which set of STS it contains, we can use this information to sequence overlapped clones. The identification is done one STS at a time. A clone is called *positive* if it contains this STS and *negative* if not. Suppose d cuttings have been made. Due to the unique presence of an STS in M, at most d clones can be positive ("at most" because an STS can be cut into half in a cutting, or a clone can be damaged).

Given a set of n clones containing e ($e \leq d$) positive clones, where n can be in the thousands while e is single-digit, the currently most efficient way of identifying the positive clones is through *group testing* [1]. A group test applies to a subset of the n clones with two possible outcomes: a positive outcome indicates that the subset contains a positive outcome (not knowing which or how many), while a negative outcome indicates otherwise. Since each test is a biological experiment taking several hours, it is crucial that all tests can be performed parallelly. This implies that all subsets under testing must be determined simultaneously, known as *nonadaptive group testing* in the literature, or a *pooling design* as preferred by biologists.

A major tool in constructing a pooling design is the d-disjunct matrix, which is a binary matrix such that if a column is viewed as a set of row indices (those rows with a 1-entry), then no column is covered by the union of any d columns. Let M denote the incidence matrix between test-subsets (rows) and clones (columns). Kautz and Singleton [2] proved that if M is d-disjunct, then it can identify the e positive clones if $e \leq d$ by simply noting that any clone which appears in a negative pool is a negative clone, and the others are positive clones.

Macula [3] generalized the notion of d-disjunctness to d^z-disjunctness where every column has at least z 1-entries not covered by the union of any other d columns. A d^z-disjunct matrix allows a negative clone C to be identified even though up to $\lfloor (z-1)/2 \rfloor$ errors can occur to outcomes. Namely, in the worst case $\lfloor (z-1)/2 \rfloor$ negative pools in which C appears are all recorded as positive, but C still appears in at least $\lceil z/2 \rceil$ negative pools (correctly recorded) to be identified. Note that a d-disjunct matrix is a d^1-disjunct matrix and offers no error tolerance.

Ngo and Du gave a construction of d^z-disjunct matrices. Consider $2k$ vertices. A matching is called an m-matching if it consists of m matches. Then there are

$$g(m,k) = \binom{2k}{2m} \frac{2k!}{k!2^k}$$

m-matchings, where $m \leq k$. Construct a $g(d,k) \times n$ matrix M by indexing its rows by all the d-matchings, $d < k$, and its columns by n arbitrary (but distinct) k-matchings. M has a 1-entry in cell (i,j) if and only if the index of row i is contained in the index of column j. For each column C and a set $D = \{D_1, \ldots, D_d\}$ of other columns, define a hypergraph H(C,D) whose vertices are the matches in the k matching of C, and edge E_i consists of the matches in $C \setminus D_i$, $1 \leq i \leq d$. Since each pair of maximum matchings differ in at least two matches, $l_i \equiv |E_i| \geq 2$. Ngo and Du proved that M is d^z-disjunct with $z = d + 1$. By Theorems 2.2 and 3.2 and Lemma 2.6, we improve z to

$$\begin{cases} 2^{k-d-1}\,(2d-k+2) & \text{for } d+1 \leq k \leq 2d-1, \text{ and} \\ 2^d & \text{for } 2d \leq k. \end{cases}$$

Acknowledgements

The authors would like to thank the referees for their helpful comments in revising this paper.

The first, third and fourth authors' research was partially supported by a Republic of China National Science Council grant NSC 92-2115-M-009-014. The second author's research was partially supported by a Republic of China National Science Council grant NSC 92-2115-M-009-004.

References

[1] D.Z. Du, F.K. Hwang, Pooling Designs and Nonadaptive Group Testing, World Scientific, Singapore, 2006.
[2] W.H. Kautz, R.C. Singleton, Nonrandom binary superimposed codes, IEEE Trans. Inform. Thy. 10 (1964) 363–377.
[3] A.J. Macula, Error-correcting nonadaptive group testing with de-disjunct matrices, Discrete Appl. Math. 80 (1997) 217–222.
[4] H.Q. Ngo, D.Z. Du, New constructions of non-adaptive and error-tolerance pooling designs, Discrete Math. 243 (2002) 161–170.