6
Pooling Designs on Complexes

6.1 Introduction

The pooling design we have discussed so far has a set of positive elements each can induce a positive effect. Sometimes it takes a set of elements combined to induce a positive effect. Therefore the set of positive elements is replaced by the set $\mathcal{D} = \{D_1, ..., D_d\}$ where each D_i, called a *positive complex*, is a subset of elements. It is usually assumed that $D_i \not\subseteq D_j$ for all $i \neq j$. Torney [22] first introduced the complex model and gave the complexes of eukaryotic DNA transcription and RNA translation as an example.

Besides its applicability to molecular biology, the complex model is interesting as a genuine generalization of the classic group testing from searching subsets to searching subgraphs. Consider a hypergraph $H(V, E)$ where the vertex-set V is the set of elements (clones, molecules) and each edge represents a candidate member of $\mathcal{D}$. Edges in $\mathcal{D}$ are referred to as *positive edges* while all other edges are *negative*. The problem is to identify all positive edges of H. An *edge-test* is a test on a subset S of vertices with two possible outcomes: a *positive outcome* indicates that S contains an edge of $\mathcal{D}$; a *negative outcome* indicates otherwise. Note that a biological assay on a pool corresponds to an edge-test since a positive outcome is obtained as long as the pool contains one positive complex (an edge). We do not distinguish between ”edge” and ”complex”, often honoring the usage of the quoted literature. The *rank* of an edge is the number of vertices in it. Define

(i) $H_{\bar{r}}$: rank-r graph, i.e., the maximum rank of H is r,

(ii) H_r: r-gragh, i.e., every edge is of rank r,

(iii) H^*_r: complete r-graph, i.e., a k-set is an edge if and only if $k = r$,

(iv) $H^*_{\bar{r}}$: complete rank-r graph, i.e., a k-set is an edge if and only if $k \leq r$.

While the group testing (d, n) model can be represented as the edge-testing (H^*_1, d) model, it is also well known (p.211 of [6]) that a group test on a set S in the (d, n) model is equivalent to an edge test on S in the $(H^*_d, 1)$ model since both tests divide the candidate set into two parts, one containing candidates which intersect with S, and the other containing candidates which do not (note that the positive outcome of one test corresponds to the negative outcome of the other). Therefore, there exists a one-to-one mapping between the classical group testing algorithms of identifying d ($\bar{d}$) positive vertices and the edge-testing algorithms of identifying a single edge in H_d^* ($H_{\bar{d}}^*$).

Let $t(H : \mathcal{D})$ denote the minimum number of tests given H and $\mathcal{D}$. If the only information available about $\mathcal{D}$ is its cardinality d, then write $t(H : d)$ instead, or $t(H : \bar{d})$ if d is an upper bound. From the above discussion, we have

Theorem 6.1.1 $t(H_1^* : d) = t(H_d^* : 1)$.

For $d > 1$ and $r > 1$, the edge-testing model no longer has an interpretation as a group testing model. Further, the edge-testing model encounters a difficulty not present in the group testing model. In the group testing model, once a positive vertex is found, it is removed from the set and has no impact on further testing. But in the edge-testing model, once a positive edge is found we cannot remove it since the vertices in it may still appear in other positive edges. On the other hand if a positive edge is not removed, then it may be caught repeatedly. An edge-testing algorithm has to deal with this issue.

We extend the notions of d-separable and d-disjunct matrices to the edge-testing model. For a given hypergraph H, let M be the incidence matrix of an edge-testing algorithm where rows are labeled by tests and columns by vertices. Then M is called $(H : d)$*-separable* if any two different d-sets of complexes have different outcome vectors; it is $(H : \bar{d})$*-separable* if d is just an upper bound of the cardinalities of the two sets. This implies that given an outcome vector U, there exists a unique set $\mathcal{D} = \{D_1, ..., D_j\}$, $j = d$ in the d-separable case and $j \leq d$ in the $\bar{d}$-separable case, consistent with U. In the d-separable case, we assume no two positive edges can contain each other for otherwise we can't tell whether only one or both are in the positive set.

M is called $(H : d)$*-disjunct* if all edges of H not in $\mathcal{D}$ must appear in a pool whose outcome is negative. Therefore we can eliminate all negative edges from negative pools and what left are the D_i's. For an edge X, we say a row *contains* (or *covers*) X if it intersects every vertex (column) of X. In a matrix M, $\cap X$ denotes the set of rows each containing X. Then more formally, M is $(H : d)$-disjunct if and only if for every set of $d+1$ edges $E = \{e_0, e_1, ..., e_d\}$, there exists a row in M which contains $e_0 \in E$, but none of $E \setminus \{e_0\}$, or

$$\cap e_0 \not\subseteq \cup_{i=1}^{d}(\cap e_i).$$

Note that for two edges $e \subset e'$, there does not exist a row covering e' bot not e. Hence $(H : d)$-disjunct makes sense only if the assumption that no edge contains another edge is made, which we do. In particular, $(H*_r : d)$-disjunctness is a valid concept. Further

Lemma 6.1.2 *M is $(H : d)$-disjunct if and only if for every set of $d + 1$ edges, $\{e_0, e_1, ..., e_d\}$, there exists a row which contains e_0 but none of columns $C_1, ..., C_d$ for some $C_i \in e_i$, $1 \leq i \leq d$.*

The relations between $(H : d)$-separable, $(H : \bar{d})$-separable and $(H : d)$-disjunct are much like in group testing. We list them in the following, some obvious ones are stated without proofs.

Theorem 6.1.3 *The $(H : d)$-disjunct implies the $(H : \bar{d})$-separable which imples the $(H; d)$-separable.*

Theorem 6.1.4 *$(H^*_r : d)$-separable, the $(H{*}_r : \bar{d})$separable and the $(H^*_r : d)$-disjunct properties are preserved if either d is reduced to d' or r to r'.*

Theorem 6.1.5 *Let M be $(H : d)$-disjunct, and let M' be obtained from M by deleting a row. Then M is $(H : \bar{d})$-separable.*

Proof. Consider two distinct sets $\mathcal{D}$, $\mathcal{D}'$ of complexes. Then either $\mathcal{D} \backslash \mathcal{D}'$ or $\mathcal{D}' \backslash \mathcal{D} \neq \emptyset$. Without loss of generality, assume the former and let $D_1 \in D \setminus D'$. Suppose $U(D) = U(D')$. Then D_1 is covered by $\cup_{D_i \in \mathcal{D}} D_i = U(\mathcal{D}) = U(\mathcal{D}') = \cup_{D'_i \in \mathcal{D}'} D'_i$, contradicting the assumption that M is $(H : d)$-disjunct. Next, suppose $|U(\mathcal{D}) \setminus U(\mathcal{D}')| = 1$. Then $U(\mathcal{D}') \subset U(\mathcal{D})$ implies that $\mathcal{D}'$, hence every $\mathcal{D}'_i \in \mathcal{D}'$, is covered by $\cup_{D_i \in \mathcal{D}} D_i$, again contradicting the assumption that M is $(H : d)$-disjunct. Therefore $|U(\mathcal{D}) \setminus U(\mathcal{D}')| \geq 2$; so deleting a row would preserve the distinctness of $U(\mathcal{D})$ and $U(\mathcal{D}')$. □

Lemma 6.1.6 *The $(H : \bar{d})$-separable, the $(H : d)$-separable and the $(H : d)$-disjunct properties are preserved under adding rows or dropping columns.*

Theorem 6.1.7 *$(H : \bar{d})$-separable implies $(H : d - 1)$-disjunct.*

Proof. Let M be $(H : \bar{d})$-separable. Suppose to the contrary that M is not $(H : d-1)$-disjunct. Then there exists a d-set D of edges in H and an edge $D_1 \in D$ such that D_1 is covered by the union of the other $d - 1$ edges. Then $U(\mathcal{D}) = U(\mathcal{D} \setminus \{D_1\})$, contradicting the assumption of $(H : \bar{d})$-separable. □

While the theory of $(H : d)$-disjunct is a natural extension of d-disjunct, the construction is not since if edges can have very large ranks, then even the mere requirement that for each edge there exists a row containing it may force a large matrix. To control the size of the matrix, we need to make use of an upper bound r of edge ranks. Thus we will often represent H by $H_{\bar{r}}$. Note that this representation does not lose generality since every H has a maximum rank.

Finally, we introduce the error-tolerant version of $(H : d)$-disjunct matrix. A binary matrix is $(H : d; z)$-disjunct if for any $d + 1$ edges $e_0, e_1, ..., e_d$, there exist at least z rows each covers e_0 but none of the other e_i. $(H : d; 1)$-disjunct will be simply written as $(H : d)$-disjunct. An $(H : d; z)$-disjunct matrix can identify all positive complexes with up to $\lfloor (z-1)/2 \rfloor$ errors since there exist at least $\lceil (z+1)/2 \rceil$ $(> \lfloor (z-1)/2 \rfloor)$ negative rows covering a negative edge.

6.2 A Construction of $(H : d; z)$-Disjunct Matrix

Du, Hwang, Thai, Wu and Znati [9] generalized an idea of Du, Hwang, Wu and Znati [8] in constructing d-disjunct matrix for the graph-testing model. The idea is to construct a q-nary $(H : d; z)$-disjunct matrix and then convert it to binary, while a q-nary matrix is $(H : d, z)$-disjunct if for any $d + 1$ edges $e_0, e_1, ..., e_d$, there exist at least z rows in each of which $\{$ entries of $e_0\} \not\supseteq \{$ entries of $e_i\}$ for all $1 \le i \le d$.

Consider a hypergraph $H = (V, E)$ with maximum rank r. Let $GF(q)$ be a finite field of order q. Associate each vertex $v \in V$ with a distinct polynomial p_v of degree k over $GF(q)$. Thus each edge $e \in E$ associates with a subset of polynomials, $P_e = \{p_v \mid v \in e\}$. Let T be a subset of t elements in $GF(q)$. Construct a $t \times |V|$ q-nary matrix $A_H(q, k, t)$ with rows labeled by T and columns by V. Each cell (x, v) contains an element $p_v(x)$ in $GF(q)$.

Theorem 6.2.1 *Suppose $H = H_{\bar{r}}$ and $t \ge rdk + z$. Then $A_H(q, k, t)$ is q-nary $(H : d; z)$-disjunct.*

Proof. Suppose to the contrary that no such z rows exist. Then for at least $rdk + 1$ values of $x \in T$, $P_{e_0}(x) \supseteq P_{e_i}(x)$ for some $1 \le i \le d$. Thus there exists a fixed j, $1 \le j \le d$, such that $P_{e_0}(x) \supseteq P_{e_j}(x)$ for at least $rk + 1$ values of $x \in T$. Since each $u \in e_j$ must have $p_u(x) = p_v(x)$ for some $v \in e_0$ for these $rk + 1$ x's, there exists a $v \in e_0$ such that $p_u(x) = p_v(x)$ for at least $k + 1$ x's. Since both p_u and p_v are polynomials of degree k, we have $p_u = p_v$, contradicting our assumption that p_u and p_v are distinct. □

Next, we construct a binary matrix $B_H(q, k, t)$ from $A_H(q, k, t)$. $B_H(q, k, t)$ has $|V|$ columns labeled by V. For each row x of $A_H(q, k, t)$ and each set $P_e(x)$, $e \in E$, $B_H(q, k, t)$ has a row labeled by $< x, P_e(x) >$ which has an 1-entry in cell $(< x, P_e(x) >, v)$ if $p_v \in P_e(x)$, and a 0-entry otherwise.

Theorem 6.2.2 *Suppose $t \ge rdk + z$. Then $B_H(q, k, t)$ is $(H : d; z)$-disjunct.*

Proof. Consider $d + 1$ edges $e_0, e_1, ..., e_d$. Let x be a row in $A_H(q, k, t)$ such that $P_{e_0}(x) \not\supseteq P_{e_i}(x)$ for any $1 \le i \le d$. Then row $< x, e_0 >$ in $B_H(q; k, t)$ covers e_0 but not e_i for all $1 \le i \le d$. By Theorem 6.2.1, $A_H(q, z.t)$ has z such rows. Hence $B_H(q, k, t)$ is $(H : d; z)$-disjunct. □

$B_H(q, k, t)$ has $t' = \sum_{x \in T} \sum_{e \in E} |P_e(x)|$ rows. $|P_e(x)|$ can be bounded in two ways. The first is $|P_e(x)| \le |E|$. But this is not a useful bound since that leads to a bound of $t|E|$ tests while individually testing each edge of H takes only $|E|$ tests.

Next we drive the second bound. Suppose $|\{p_v(x) \mid v \in V\}| = c_x$. Then $c_x \le q$. Each row x in $A_H(q, k, t)$ can generate at most $\sum_{i=1}^{r} \binom{c_x}{i} \le \sum_{i=1}^{r} \binom{q}{i} \le \binom{q+r-1}{r}$ rows in $A_H(q, k, t)$. Thus $t' \le t\binom{q+r-1}{r}$. Let $|V| = n$. Then $n \le q^{k+1}$. Set q to be the minimum prime power $\geq drk + z$ and approximate it by drk, and set $t = drk - z$ also approximated by drk. Assume $n \geq q^k$. Then

$$t' \leq t\binom{q+r-1}{r} \sim drk(\frac{q}{r})^r = r(dk)^{r+1} \leq r(d\log_q n)^{r+1}.$$

Note that the number of edges can certainly reach n^r. With a more detailed analysis similar to the one in Sec. 3.5, we have

Theorem 6.2.3 *$B_H(q,k,t)$ is $(H:d;z)$-disjunct with at most $q(q+1)^r$ rows, where*

$$q \leq (2+o(1))\frac{log_2 n}{\log_2(d\log_2 n)}.$$

Row x in $A_H(q,k,t)$ is transformed to c_x rows in $B_H(q,k,t)$ by replacing each element of $GF(q)$ with a binary vector of length c_x View this as a $l \times q$ matrix Q. Then the requirement on Q is that if x is a row in which $P_{e_i}(x) \not\subseteq P_{e_0}(x)$ for all $1 \leq i \leq d$, or equivalently, there exists a $C_{r+i} \in e_i$ for every $1 \leq i \leq d$ such that $p_{C_{r+i}}(x) \notin P_{e_0}(x)$, then there exists a row in Q intersecting each column $C_1, ..., C_r$, but none of $C_{r+1}, ..., C_{r+d}$. Chen, Du and Hwang [2] noted that if Q is a $(d,r]$-disjunct matrix, then it certainly meets the requirement. In fact, a $(d,r;z]$-disjunct matrix can be used to provide more error-tolerance.

Li, Thai, Liu and Wu [14] considered the special case that H is an r-partite graph and $z = 1$. We give the error-tolerant version here. The construction of $A_H(q,k,t)$ and $B_H(q,k,t)$ are same as before.

Corollary 6.2.4 *For H an r-partite graph and $t \geq dk + z$, $B_H(q,k,t)$ is $(H:d;z)$-disjunct.*

The reason that t can be reduced from $drk + z$ to $dk + z$ is the following: Suppose $e_0 = \{v_1, ..., v_r\}$ and $e_i = \{v'_1, ..., v'_r\}$ where v_j and v'_j are vertices in part j, $1 \leq j \leq r$. Then $P_{e_0} = P_{e_i}$ implies $p_{v_j}(x) = p_{v'_j}(x)$ for $1 \leq j \leq r$. Thus, if $P_{e_0}(x) = P_{e_i}(x)$ for $k+1$ values of x, then $p_{v_j} = p_{v'_j}$ for $1 \leq j \leq r$. On the other hand, this mapping between v_j and v'_j does not exist for general $H_{\bar{r}}$. Therefore $P_{e_0}(x) = P_{e_i}(x)$ must occur at least $rk + 1$ times in order to force one of j to have $p_{v_j}(x) = p_{v'_j}(x)$ at least $k+1$ times.

An analysis similar to before shows that the number of tests in $B_H(q,k,t)$ is about $(d\log_q n)^{r+1}$, a saving of a factor r from the general case.

6.3 $(d,r;z]$-disjunct matrix

We now discuss another type of disjunct matrices which is not defined on a graph, and thus seemingly unrelated to the $(H:d;z)$-disjunct matrix. Yet we will prove a surprising result that the new disjunct matrix is equivalent to an $(H_r^* : d; z)$-disjunct matrix.

A binary $t \times n$ matrix M is called $(d, r; z]$-disjunct ($(d, r; 1]$-disjunct will be written as $(d, r]$-disjunct) if for any $d+r$ columns $C_1, ..., C_{d+r}$, there exist at least z rows each intersecting $C_1, ..., C_r$, but none of $C_{r+1}, ..., C_{r+d}$, i.e.,

$$| \cap_{i=1}^{r} C_i \setminus \cup_{j=r+1}^{r+d} C_j| \geq z.$$

Note that $n \geq d + r$ is assumed.

Chen, Du and Hwang [2] proved the following theorem for $z = 1$.

Theorem 6.3.1 *$(d, r; z]$-disjunct $= (H_{\bar{r}}^* : d; z)$-disjunct.*

Proof. Suppose that M is $(d, r; z]$-disjunct. Consider $d+1$ arbitrary edges $e_0, e_1, ..., e_d$ such that $e_i \not\subseteq e_0$ for $1 \leq i \leq d$. Let $e_0 = \{C_1, ..., C_r\}$ and select C_{r+i} and select C_{r+i} from $e_i \setminus e_0$ for $1 \leq i \leq d$. Set $S = \{C_{r+i} \mid 1 \leq i \leq d\}$. Note that the cardinality of S can be less than d since $C_{r+i} = C_{r+j}$ is possible for $i \neq j$.

Since M is $(d, r; z]$-disjunct, there exist z rows intersecting $C_1, ..., C_r$ but none of $C_{r+1}, ..., C_{r+d}$, i.e., each such row covers e_0 but none of e_i, $1 \leq i \leq d$. Thus, M is $(H{*}_r : d; z)$-disjunct.

We next prove $(H_{\bar{r}}^* : d)$-disjunct $\Rightarrow$ $(d, r]$-disjunct. Suppose that M is not $(d, r]$-disjunct. Then there exist $d + r$ columns $C_1, ..., C_{r+d}$ such that

$$\cap_{i=1}^{r} C_i \subseteq \cup_{i=r+1}^{r+d} C_i.$$

Set $e_0 = \{C_1, ..., C_r\}$ and $e_i = \{C_2, ..., C_r, C_{r+i}\}$ for $1 \leq i \leq d$. Then

$$\cap e_0 = (\cap\{C_2, ..., C_r\}) \cap \{C_1\} \subseteq (\cap\{C_2, ..., C_r\}) \cap (\cup_{i=r+1}^{r+d}) = \cup_{i=1}^{d}(\cap e_i).$$

Hence M is not $(H{*}_r; d)$-disjunct.

To extend to the general z case, suppose M is not $(d, r; z]$-disjunct. Then there exist at most $z-1$ rows each intersecting $C_1, ..., C_r$ but none of $C_{r+1}, ..., C_{r+d}$. Delete these rows, then the remaining matrix is not $(d, r]$-disjunct, hence not $(H{*}_r : d)$-disjunct as we just proved. This implies that even after bring back the deleted rows, the natrix is still not $(H{*}_r : d; z)$-disjunct. □

Using Theorem 6.3.1, we can translate Theorems 6.1.3-6.1.7, some only considering the special case $H = H{*}_r$, to their $(d, r; z]$ versions. Here we only quote two such translations which have been independently studied [22] in term of $(d, r; z]$ for the case $z = 1$.

Lemma 6.3.2 *The property of $(d, r; z]$-disjunct or $(\bar{d}, r; z]$-separable or $(d, r; z]$-separable is preserved by reducing d or r.*

Note that a corollary of Theorem 6.3.1 is

Lemma 6.3.3 *$(d, r; z]$-disjunct $\Rightarrow$ $(H^*_{\bar r} : d; z)$-separable.*

The following was proved in [22] for $z = 1$:

Lemma 6.3.4 *$(H^*_{\bar r} : \bar d; z)$-separable $\Rightarrow$ $(d-1, r; z]$-disjunct and $(d, r-1; z]$-disjunct.*

Proof. We prove only for $z = 1$ and rely on the argument in Theorem 6.3.1 for extension to general z.

Suppose M is not $(d-1, r]$-disjunct. Then there exist $r+d-1$ columns $C_1, ..., C_{r+d-1}$ such that

$$\cap_{j=1}^{r} C_j \subseteq \cup_{j=r+1}^{r+d-1} C_j.$$

Set $e_0 = \{C_1, ..., C_r\}$ and $e_j = \{C_2, ..., C_r, C_{r+j}\}$ for $1 \le j \le d-1$. Let $D = \{e_j \mid 0 \le j \le d-1\}$ and $D' = \{e_j \mid 1 \le j \le d-1\}$. Then

$$\cup_{e_j \in D}(\cap e_j) = \cup_{e_j \in D'}(\cap e_j).$$

Hence M is not $(H^*_{\bar r} : \bar d)$-separable.

Next suppose M is not $(d, r-1]$-disjunct. Then there exist $r+d-1$ columns $C_1, ..., C_{r+d-1}$ such that

$$\cap_{j=1}^{r-1} C_j \subseteq \cup_{j=r}^{r+d-1} C_j.$$

Set $e_0 = \{C_1, ..., C_{r-1}\}$ and $e_j = e_0 \cup \{C_{r-1+j}\}$ for $1 \le j \le d$. Set $D = \{e_0\}$ and $D' = \{e_1, ..., e_d\}$. Then

$$\cup_{e_j \in D}(\cap e_j) = \cap e_0 = \cup_{j=1}^{d}(\cap e_j) = \cup_{e_j \in D'}(\cap e_j).$$

Hence M is not $(H^*_{\bar r} : \bar d)$-separable. □

D'yachkov, Villenkin, Macula and Torney also observed the following result for $z = 1$.

Lemma 6.3.5 *Let M be $(d, r; z]$-disjunct and M' be obtained from M by interchanging 0 and 1. Then M' is $(r, d; z]$-disjunct.*

Let $t(d, r, n; z]$ denote the minimum t in a $t \times n$ $(d, r; z]$-disjunct matrix.

Corollary 6.3.6 *$t(d, r, n; z] = t(r, d, n; z]$.*

Stinson and Wei [20], extending results of Stinson, Wei and Zhu [21], proved the following results. The first two are obvious.

Lemma 6.3.7 *Let M be a $(d, r; z]$-disjunct matrix and let M' be obtained from M by deleting a column and all rows intersecting it. Then M' is $(d-1, r; z]$-disjunct.*

Lemma 6.3.8 *Let M be a $(d, r; z]$-disjunct matrix and let M' be obtained from M by deleting a column and all rows not intersecting it. Then M' is $(d, r-1; z]$-disjunct.*

From Lemmas 6.3.7 and 6.3.8, we have

Theorem 6.3.9 $t(d, r, n; z] \geq t(d-1, r, n-1; z] + t(d, r-1, n-1; z]$.

This recursion leads to a lower bound of $t(d, r, n; z]$. First, we show a lemma. Define

$$g(d, r, n) = \frac{\binom{d+r}{r} \log n}{\log(d+r)}.$$

Lemma 6.3.10 $g(d, r, n) \leq g(d-1, r, n-1) + g(d, r-1, n-1)$.

Proof. Using the fact that $\log x / \log(x-1)$ is decreasing for $x > 1$. □

Theorem 6.3.11 *For $d + r > 2$,*

$$t(d, r, n; z] \geq 2cg(d, r, n-1) + \frac{c(z-1)}{2}\binom{d+r}{r},$$

where c is the same constant as in Theorem 2.7.8.

Proof. The proof is by induction on $r + d$. The case $r = 1$ is by Theorem 2.6.6 and (2.6.1). The case $d = 1$ is the same as the case $r = 1$ by Corollary 6.3.6. For $d \geq 2$, $r \geq 2$,

$$\begin{aligned} t(d, r, n; z] &\geq t(d-1, r, n-1; z] + t(d, r-1, n-1; z] \\ &\geq 2cg(d-1, r, n-1) + \\ &\quad \frac{c(z-1)}{2}\binom{d+r-1}{r} + 2cg(d, r-1, n-1) + \frac{c(z-1)}{2}\binom{d+r-1}{r-1} \\ &\geq 2cg(d, r, n-1) + \frac{c(z-1)}{2}\binom{d+r}{r}. \end{aligned}$$

□

Define

$$h(d, r, n) = \binom{d+r}{r}(d+r)\log n / \log\binom{d+r}{r}.$$

Then a similar argument leads to a stronger result.

Theorem 6.3.12 *There exists an integer $n^*_{d,r}$ such that for $n \geq n^*_{d,r}$,*

$$t(d, r, n; z] \geq 0.7ch(d, r, n) + c(d-1)(z-1)/2,$$

where c is the same constant as in Theorem 2.7.8.

6.4 Constructions for $(d, r; z]$-Disjunct Matrices

D'ychkov, Villenkin, Macula and Torney [10] gave a simple construction of $(d, r]$-disjunct matrices, which we will extend to the error-tolerant case.

Theorem 6.4.1 *The $\binom{n}{k} \times n$ binary matrix where the rows consists of all k-subsets of the set $[n]$, $r \leq k \leq n - d$, is $(d, r; \binom{n-d-r}{k-r}]$-disjunct.*

Proof. For a given r-set R and a d-set D, let $K = R \cup S$ where S is a set of $k - r$ elements from $[n] \setminus (R \cup D)$. Then the row corresponding to K covers R but not intersects D. Since there are $\binom{n-d-r}{k-r}$ choices of S, z is equal to that number. □

Corollary 6.4.2 $t(d, r, n) \leq \min\{\binom{n}{d}, \binom{n}{r}\}$.

Proof. The minimum of $\binom{n}{k}$ occurs at one of two extreme points. But $\binom{n}{n-d} = \binom{n}{d}$. □

One way to view this construction is that we take all unions of r rows of an $n \times n$ identity matrix. Note that the identity matrix is a d-disjunct matrix. An attempt was made in [10] to replace this identity matrix by any d-disjunct matrix M to reduce the number of rows. However, although for each member i of R, M contains a row M_i which contains i but no member of D, and the union of M_i over all $i \in R$ covers R but not intersects D, the union may be over less than r rows as M_i and M_j could be the same. Hence the union does not correspond to a row in the matrix constructed by taking all unions of r rows of M.

Stinnson, Wei and Zhu [21] suggested a different way to obtain error-tolerance for the construction in Theorem 6.4.1, which they viewed as a $(d, r]$-disjunct matrix as originally intended in [10]. By taking z copies of each row of the constructed matrix, one obtains a $(d, r; z]$-disjunct matrix. Note that this multiplication works even if the starting matrix itself is error-tolerant. Thus

Lemma 6.4.3 $t(d, r, n; zz'] \leq \min\{zt(d, r, n; z'], z't(d, r, n; z]\}$.

In particular when applied to the construction in Theorem 6.4.1, we obtain a $z\binom{n}{k} \times n$ $(d, r; z\binom{n-d-r}{k-r}]$-disjunct matrix.

q-nary codes were used in Chapter 3 to construct $(d; z)$-disjunct matrices. A similar construction also works for our current model. M is called a q-nary $(d, r; z]$-disjunct matrix if for any two disjoint set D and R of columns, $|D| = d$ and $|R| = r$, there exist at least z rows i such that

$$\{m_{ij} \mid j \in D\} \cap \{m_{ij} \mid j \in R\} = \emptyset.$$

Stinson and Wei [20], extending a result of [10] for $z = 1$, proved

Theorem 6.4.4 *The existences of a $t_1 \times n$ q-nary $(d, r; z_1]$-disjunct matrix and $t_2 \times q$ $(d, r; z_2]$-disjunct matrix imply the existence of a $t_1 t_2 \times n$ $(d, r; z_1 z_2]$-disjunct matrix.*

Proof. Let M_1 and M_2 denote the above $t_1 \times n$ and $t_2 \times q$ matrix, respectively. Replace each entry y in M_1 by column y of M_2. Then for each row i in M_1 such that

$$\{m_{ij} \mid j \in R\} \cap \{m_{ij} \mid j \in S\} = \emptyset,$$

there exist at least z_2 rows after the replacement such that $\cap R \not\subseteq \cup D$. Further, for $i' \neq i$, then the two sets of z_2 rows are disjoint. Hence the constructed matrix has $z_1 z_2$ rows satisfying $\cap R \not\subseteq \cup D$. □

Again, Theorem 6.4.3 can be viewed as a mechanism to grow a $d, r; z]$-disjunct matrix in the number of columns as well as in the error-tolerant capability. The q-nary $(d, r; z)$-disjunct matrix then becomes the engine of this mechanism. On the other hand, we also need some $(d, r; z]$-disjunct matrix for small z, perhaps just a $(d, r]$-disjunct matrix, to serve as the input of this mechanism. We will first study the construction of these matrices, and then the construction of the q-nary matrices.

As commented in Chapter 3, the incidence matrix of a T-design with blocks as rows is not good for a d-disjunct matrix since the number of rows is not smaller than the number of columns by the Fisher inequality, hence not better than the trivial d-disjunct matrix. However, for the $(d, r]$-disjunct matrix, the t-design has a second life since we only need to beat $\min\{\binom{n}{r}, \binom{n}{d}\}$. Mitchell and Piper [18] obtained

Theorem 6.4.5 *A $T - (v, k, 1)$ design yields a $t \times n$ $(d, r]$-disjunct matrix with $t = \binom{v}{T}/\binom{k}{T}$, $n = v$, $d = \lambda_{T-1} - 1$ and $r = T - 1$, where $\lambda_{T-1} = (v - T + 1)/(k - T + 1)$.*

Proof. For any set S of $T - 1$ columns, there exist λ_{T-1} rows covering S. Let $\cap S = \{R_{i_j} \mid 1 \leq j \leq \lambda_{T-1}\}$ denote these rows. Then $(R_{i_x} \setminus S) \cap (R_{i_y} \setminus S) = \emptyset$ for $x \neq y$. Hence no column can intersect more than one such row, and it takes the union of at least λ_{T-1} columns to contain $\cap S$. □

Stinson and Wei [20] observed that a $3 - (q^2 + 1, q + 1, 1)$ design (an *inverse plane*) always exists for q a prime power.

Corollary 6.4.6 *A $q(q^2 + 1) \times (q^2 + 1)$ $(q, 2]$-disjunct matrix exists for q, a prime power.*

A T-design is called *super-simple* if every pair of blocks intersect in at most t elements. Kim and Lebedev [13] proved

Theorem 6.4.7 *A super-simple $T - (v, k, \lambda)$ design yields a $t \times n$ $(d, r]$-disjunct matrix where*

$$t = \lambda \binom{v}{T} / \binom{k}{T}, n = v, d = \lambda - 1 \text{ and } r = T.$$

Proof. For any set S of T columns, there exist exactly λ rows covering it. Let $R_{i_1}, ..., R_{i_\lambda}$ denote those rows. Then $(R_{i_x} \setminus S) \cap (R_{i_y} \setminus S) = \emptyset$, $1 \leq x < y \leq \lambda$, for otherwise the two blocks (rows) i_x and i_y intersect in more than T elements. Thus a column not in S can intersect at most one such row, and it takes at least λ columns to cover $\cap S = \sum_{x=1}^{\lambda} R_{i_x}$. □

Next, we study the construction of q-nary $(d, r; z]$-disjunct matrices. First, consider the $z = 1$ case.

The MDS code was introduced in chapter 3 to construct q-nary disjunct matrix. Sagalovich [26] gave the following result.

Lemma 6.4.8 *Any MDS code which has parameters (q, kt), where $t \geq dr(k-1)+1$ and $q^k \geq d+r$, yields a $t \times q^k$ q-nary $(d, r]$-disjunct matrix.*

[10] used the Reed-Soloman code as an MDS code. Note that for any integer $k \geq 2$ and a prime power $q > k-1$, there exists a $(q, k, q+1)$ Reed=Soloman code.

Theorem 6.4.9 *Suppose $q \geq dr(k-1)+1$. Then*

$$t(d, r, q^k] \leq t(d, r, q][dr(k-1)+1].$$

Proof. Denote by M the matrix obtained from the Reed-Soloman code by removing any $q - dr(k-1)$ rows. Then M is still a MDS code with parameters $(q, k, dr(k-1)+1)$. By Lemma 6.4.8, M is q-nary $(d, r]$-disjunct. Using M as the q-nary matrix in Theorem 6.4.7, we obtain Theorem 6.4.9. □

Some examples were given in [10] to illustrate the effectiveness of Theorem 6.4.9.

$$\begin{aligned}
&t(2,2,n) \leq \binom{r}{2} \text{ for any } n \geq 4 \text{ (Corollary 6.4.2)}, \\
&t(2,2,16) = t(2,2,4^2) \leq t(2,2,4) \cdot (2 \cdot 2 + 1) \leq \binom{4}{2} \cdot 5 = 30, \\
&t(2,2,256) = t(2,2,16^2) \leq t(2,2,16) \cdot 5 \leq 30 \cdot 5 = 150, \\
&t(2,2,4096) = t(2,2,16^3) \leq t(2,2,16) \cdot (2^3+1) \leq 30 \cdot 9 = 270.
\end{aligned}$$

Another construction is to use the hashing family.

A *(t, n, q, c)-k-perfect hash family* is a $t \times n$ q-nary matrix while for any c of the n columns, there exist at least k rows such that the entrices of the c columns are distinct. Note that a (t, n, q, c)-k-perfect family is a q-nary (d, r, k)-disjunct matrix for any d, r satisfying $d + r = c$. Wang and Xing [24] proved

Lemma 6.4.10 *There exists an explicit construction for an infinite family of $(t, n, q.c)$-1-perfect hash family with $t = O(\log n)$.*

Using this perfect has family as a q-nary $(d, r, 1)$-disjoint matrix in Theorem 6.4.4, we obtain

Theorem 6.4.11 *There exists an infinite class of $t \times n$ $(d, r; k]$-disjunct matrices with $t = O(\log n)$.*

Stinson and Wei used the vertex z-cover of a hypergraph to construct $(d, r; z]$-disjunct matrices.

For a hypergraph (V, E), (note that this hypergraph has nothing to do with the underlying hypergraph H), VC_z is a *vertex z-cover* if for every edge $e \in E$, there exist at least z vertices $v \in VC_z$ such that $v \in e$. Then VC_1 is simply a vertex-cover. Stinson and Wei interpreted a $(d, r; z]$-disjunct matrix as a vertex k-cover of some hypergraph.

Let $V_{n;\ell,u} = \{X \subseteq [n] \mid \ell \leq |X| \leq u\}$, where $0 < \ell < u < n$. Define a class of *order-interval hypergraph* $H_{n;\ell,u}(V, E)$ with $V = V_{n;\ell,u}$,

$$e_{X,Y} = \{S \mid X \subseteq S \subseteq Y\},$$

and

$$E = \{e_{X,Y} \mid |X| = \ell, |Y| = u\}.$$

Theorem 6.4.12 *There exists a vertex z-cover Z of $H_{n;\ell,u}$ of size t if and only if there exists a $t \times n$ $(\ell, n-u; z]$-disjunct matrix.*

Proof. Let M denote the incidence matrix where rows are labeled by Z and columns by $[n]$. Consider any $l + n - u$ columns $C_1, ..., C_{l+n-u}$. Define $X = \{C_1, ..., C_l\}$ and $Y = \{C_i \mid i \notin \{l+1, ..., l+n-u\}\}$. Then there exist z rows $R_1, ..., R_z$ such that $X \subseteq R_i \subseteq Y$ for $1 \leq i \leq z$, or equivalently, R_i intersects $C_1, ..., C_l$ but not $C_{l+1}, ..., C_{l+n+u}$ for $1 \leq i \leq z$. Hence M is $(\ell, n-u; z]$-disjunct. □

Define

$$\tau^z_{n;\ell,u} = \min\{|Z| \mid Z \text{ is a vertex } z\text{-cover of } H_{n;\ell,u}\}.$$

A *fractional vertex d-cover* is a function $g : V \to R^+$ such that for any $e_{X,Y} \in E$,

$$\sum_{v \in e_{X,Y}} g(v) \geq z.$$

Define

$$(\tau^*)^z_{n;\ell,u} = \min\{\sum_{v \in V} g(v) \mid g \text{ is a fractional } z\text{-cover of } H_{n;\ell,u}\}.$$

(The superscript z will be omitted for $z = 1$.)

Stinson and Wei, extending ideas of Engel [23] from $z = 1$ to general z, obtained the following inequalities by choosing g properly:

$$\begin{aligned} \tau^z_{n,\ell,u} &\geq (\tau^*)^z_{n,\ell,u}, && (6.4.1)\\ (\tau^*)^z_{n,\ell,u} &= z \times (\tau^*)_{n,\ell,u}, && (6.4.2)\\ \tau^z_{n,\ell,u} &\geq (\tau^*)^z_{n,\ell,u} \times (\tau^*)_{u-\lambda;\ell-\lambda,u-\lambda}, && (6.4.3)\\ \tau^z_{n,\ell,u} &\geq (\tau^*)_{n,\ell,u} \times (\tau^*)^z_{u-\lambda;\ell-\lambda,u-\lambda}, . && (6.4.4) \end{aligned}$$

From (6.4.1) and (6.4.2), results on $(\tau^*)_{n,\ell,u}$ can be translated to results on $\tau^z_{n,\ell,u}$, and hence on $t(d, r; z, n]$. In particular, using a result on τ^* of Engel, we obtain

Theorem 6.4.13 $t(d, r, n; z] \geq \min\{z\binom{n}{z}/\binom{n-d-1}{m-r} \mid r \leq m \leq n - d\}$.

Using (6.4.3), we obtain

Theorem 6.4.14

$$t(d, r, n; z] \geq \min_{\substack{r-\lambda_1 \leq m_1 \leq n-d+\lambda_2 \\ \lambda_1 \leq m_2 \leq n-d-r+\lambda_1}} \frac{z\binom{n}{m_1}\binom{n-d-r+\lambda_1+\lambda_2}{m_2}}{\binom{n-d-r+\lambda_1+\lambda_2}{m_1-r+\lambda_1}\binom{n-d-r}{m_2-\lambda_1}},$$

where $0 < \lambda_1 < r_1, 0 < \lambda_2 < d$ *and* m_1 *and* m_2 *are integers.*

Using (6.4.4), we obtain

Theorem 6.4.15

$$t(d, r; z, n] \geq \min_{r-\lambda_1 \leq m \leq n-d+\lambda_2} \frac{\binom{n}{m}}{\binom{n-d-r+\lambda_1+\lambda_2}{m-r+\lambda_1}} t(\lambda_2, \lambda_1; z, n - d - r + \lambda_1 + \lambda_2],$$

where $0 < \lambda_1 < r$, $0 < \lambda_2 < d$ *and* m *is an integer.*

Setting $\lambda_1 = r - 1$ and $\lambda_2 = d - 1$, it is easily checked that the right hand side attains minimum at $m = n/2$. Thus

Corollary 6.4.16 $t(d, r, n; z] \geq 4(1 - \frac{1}{n})t(d - 1, r - 1, n - 2; z]$.

Using this recursion and (2.6.1), we obtain

Theorem 6.4.17

$$\begin{aligned} t(d, r, n; z] \geq\ & c4^{r-1}(1 - \frac{1}{n})(1 - \frac{1}{n-1}) \cdots (1 - \frac{1}{n-2r+2})[\frac{(d-r+1)^2}{\log(d-r+1)} \log(n - 2r) \\ & +(z - 1)(d - r + 1)], \end{aligned}$$

where c *is the same constant as in (2.6.1).*

Define $\log^*(1) = 1$ and $\log^*(n) = \log^*(\lceil \log n \rceil) + 1$ for $n > 1$. Stinson, Wei and Zhu proved

Lemma 6.4.18 *There exists an infinite class of $t \times n$ q-nary (d, r)-disjunct matrices with $t = O((dr)^{\log^*(n)} \log n)$.*

Using this q-nary matrix and z copies of the matrix in Theorem 6.4.1 as inputs to the mechanism in Theorem 6.4.4, we have

Theorem 6.4.19 *There exists a $t \times n$ $(d, r; z)$-disjunct matrix with $t = O(z\binom{n}{z}(dr)^{\log^*(n)} \log n)$.*

6.5 Random Designs

An attempt to extend the analysis of random designs (Chapter 5) from group testing to graph testing encounters some basic difficulties. Consider a hypergraph H and a fixed but unknown subgraph D. Let $\mathcal{D}$ be the sample space of D, i.e., a member of $\mathcal{D}$ is a set of edges which is a candidate of D. The problem is to identify D using edge tests. In the group testing problem, D is just a set of vertices. Due to symmetry, the analysis for D and D' are same if $|D| = |D'|$. But in the graph testing problem, even if $|D| = |D'|$, the subgraph structure plays an important part in the analysis. Then one has to deal with the large number of subgraph structures and to take average over all D in $\mathcal{D}$.

To alleviate this problem, we make the following assumptions:

(i) $H = H_r^*$ (H has n vertices),

(ii) $|D| = d$ and every edge is equally likely to be in D.

(iii) only RID is studied.

Note that even under the set of assumptions, the structure of D can vary, i.e., D could be a set of disjoint edges, or an r-star.

Let M be a testing matrix and $I_X(D, M)$ the event that M contains a row covering X but none of the edges in $D \setminus X$. When X is a negative edge and D is the positive set, $I_X(D, M)$ is the event that M identifies X. In Sec.6.6, we will see that even if X is positive, $I_X(D, M)$ is very relevant to the identification of X. Thus $I_X(D, M)$ is the focus of our study in random designs. Let $\bar{I}_X(D, M)$ denote the opposite of $I_X(D, M)$. Then

$$\bar{I}_X(D, M) = \cup_i \bar{I}_X(D, M_i)$$

where M_i is the i^{th} row of M.

Although row i and row j are independent in M, the two events $\bar{I}_X(D, M_i)$ and $\bar{I}_X(D, M_j)$, when D is a variable, are positively correlated since if D is a favorable structure, then it favors both rows i and j. Therefore,

$$E_D Prob(\bar{I}_X(D, M)) \geq \prod_i E_D Prob(\bar{I}_X(D, M_i)), \qquad (6.5.5)$$

namely, we cannot compute the left hand side by computing the much simpler the right hand side. Note that this problem does not exist in group testing since D is invariant given $|D| = d$. .

Suppose M is a $t \times n$ RID(p). Define

$$\Phi(t,n,d,r) = E_X E_{D,|D|=d} Prob(I_X(D,M)) = E_{D,|D|=d} Prob(I_X(D,M)) \qquad (6.5.6)$$

since H_r^* is edge-transitive.

From (6.5.5), we have

Lemma 6.5.1 $\Phi(t,n,d,r) \leq 1 - [1 - \Phi(1,n,d,r)]^t$.

We will study $\Phi(1,n,d,r)$ first.

Macula, Rykov and Yakhanin [16] proved a lower bound.

Lemma 6.5.2 $\Phi(1,n,d,r) \geq p^r \{\sum_{k=1}^r [\binom{n-r}{k}\binom{r}{r-k}\binom{n}{r}^{-1}](1-p^k)\}^d$.

Proof. X appears in the row with probability p^r. For each positive edge D_j in D, suppose it overlaps with X in $r-k$ vertices, with probability

$$\binom{n-r}{k}\binom{r}{r-k}\binom{n}{r}$$

then the probability that at least one of the remaining k vertices in D_j does not appear in the row is $(1-p^k)$. Since there are d positive edges, we multiply this probability d times to obtain a lower bound since the event $I_X(D_j, M)$ and $I_X(D_i, M)$ are positively correlated. □

To compute $\Phi(1,n,t,r)$ exactly, we have to enumerate the numerous structure of D as shown in (6.5.6). Surprisingly, there is a way to bypass the enumeration.

Let D^+ be the random variable representing the distribution of $X_0, X_1, ..., X_d$, namely, a specification of the set of r columns belonging to each X_i. Let w be the random variable representing the distribution of the 1-entries in the row. Let $I(D^+, w)$ denote the indicator function such that

$$I(D^+, w) = \begin{cases} 1 & \text{if the row covers } X_0 \text{ but none of } X_1, ..., X_d, \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$\Phi(1,n,d,r) = E_D E_w I(D^+, w).$$

Let W be the set of w 1-entries in the row. Then $\binom{W}{r}$ is the set of complexes covered by the row. Hence X must be chosen from $\binom{W}{r}$. Note that instead of en umerating the structures of D, it is now only a matter of counting in how many ways we can choose d complexes (other than D) from the set of $\binom{w}{r}$ complexes.

Macula, Torney and Villenkin [17] gave the formula

$$\Phi(1,n,d,r) = \sum_{w=0}^{n} \binom{n}{w} p^w (1-p)^{n-w} \binom{w}{r} \binom{n}{r}^{-1} [1 - \binom{w}{r} \binom{n}{r}^{-1}]^d \qquad (6.5.7)$$

Assume the row has weight w. Then the r columns of X must be taken from the W columns with 1-entries, which none of the complexes in D can have 1-entries in all its r columns.

From the last term of (6.5.7), it is obvious that the D complexes choose their columns independently (as in Lemma 6.5.1). In particular, some of them can choose the same set of columns which contradicts the fundamental assumption that complexes are distinct. We now modify (6.5.3) to distinct complexes.

Lemma 6.5.3

$$\Phi(1,n,d,r) = \sum_{w=0}^{n} \binom{n}{w} p^w (1-p)^{n-w} \binom{w}{r} \binom{n}{r}^{-1} \left[\binom{\binom{n}{r} - \binom{w}{r}}{d} / \binom{\binom{n}{r} - 1}{d} \right] . \quad (6.5.8)$$

Torney [22] gave a formula from the viewpoint that for fixed X and D, how to choose the $\binom{w}{r}-$ complexes (other than X) present in the row.

Lemma 6.5.4

$$\Phi(1,n,d,r) = \sum_{w=r}^{n} \binom{n-r}{w-r} p^w (1-p)^{n-w} \left[\binom{\binom{n}{r} - d - 1}{\binom{w}{r} - 1} / \binom{\binom{n}{r} - 1}{\binom{w}{r} - 1} \right] . \qquad (6.5.9)$$

Proof. Again, w is the row weight. From the bracket term, each complex in $X \cup D$ is treated as fixed (taking a fixed set of columns), which the $\binom{w}{r} - 1$ complexes present in the row other than X must be chosen outside of $X \cup D$. □

Offhand, this approach seems doubtful since a random choice of $\binom{w}{r} - 1$ complexes outside pf $X \cup D$ does not guarantee the union to be a w-set. While we do not have a direct proof of Lemma 6.5.3 with insight, we provide a mechannical proof that the two RHSs of (6.5.8) and (6.5.9) are indeed equal by noting:

(i) $\sum_{w=0}^{n}$ in (6.5.8) can be changed to $\sum_{w=r}^{n}$ due to the presence of the term $\binom{w}{r}$.

(ii) $\binom{n}{w}\binom{w}{r}\binom{n}{r}^{-1} = \binom{n-r}{w-r}$.

(iii)

$$\frac{\binom{\binom{n}{r}-\binom{w}{r}}{d}}{\binom{\binom{n}{r}-1}{d}} = \frac{[\binom{n}{r} - \binom{w}{}] \cdots [\binom{n}{r} - \binom{w}{r} - d + 1]}{[\binom{n}{r} - 1] \cdots [\binom{n}{r} - d]}$$

$$= \frac{[\binom{n}{r} - d - 1] \cdots [\binom{n}{r} - \binom{w}{r} - d + 1]}{[\binom{n}{r} - 1] \cdots [\binom{n}{r} - \binom{w}{r} + 1}$$

$$= \frac{\binom{\binom{n}{r}-d-1}{\binom{w}{r}-1}}{\binom{\binom{n}{r}-1}{\binom{w}{r}-1}}.$$

However, we cannot use the same "exchange-order" trick to compute $\Phi(t,n,d,r)$ exactly. Let w_i denote the weight variable of row i. Then

$$E_{\{w_1,\ldots,w_d\}}E_{D^+}\prod_{i=1}^{t} I(D^+,w_i) \neq \prod_{i=1}^{t} E_{\{w_1,\ldots,w_d\}}E_{D^+}I(D^+,w_i) = 1-[1-\Phi(1,n,d,r)]^t,$$

since $\prod$ is not a linear function.

Macula, Torney and Villenkin gave a bound by using a truncated inclusion-exclusion formula.

Theorem 6.5.5

$$\begin{aligned}\Phi(t,n,d,r) \quad \geq \quad & t\Phi(1,n,d,r) - \binom{t}{2}\sum_{w_2+w_1+w_1'}\binom{n}{w_2}\binom{n-w_2}{w_1}\binom{n-w_2-w_1}{w_1'}p^{2w_2} \\ & \cdot[p(1-p)]^{w_1+w_1'}(1-p)^{2(n-w_2-w_1-w_1')}\binom{w_2}{r}\binom{n}{r}^{-1} \\ & \cdot[\binom{n}{r}-\binom{w_2+w_1}{r}-\binom{w_2+w_1'}{r}+\binom{w_2}{r}]^d\binom{n}{r}^{-d}.\end{aligned}$$

Proof. $t\Phi(1,n,d,r)$ obviously over estimates $\Phi(t,n,d,r)$ since if k rows satisfy $I_X(D,M)$, then this one case is counted k times. We now explain the second term. Let i and i' be two rows both satisfying $I_X(D,M)$. Suppose w_2 columns intersect both i and i', w_1 (w_1') columns intersect i (i') but not i' (i), and $n-w_2-w_1-w_1'$ columns intersect neither. Then X must be covered by the w_2 columns. For $D_j \in D$, it can neither be covered by the w_2+w_1 columns of row i, nor the w_2+w_1' columns of row i'. But when D_j is covered by the w_2 columns, it is subtracted twice and needs to be added back once. □

6.6 Trivial Two-stage Pooling Designs for Complete r-graphs

A properly constructed random design M could have a high probability of containing a row R_i which covers a positive complex X_j but none of the other positive complexes. However R_i cannot identify X_j unless all other complexes covered in R_i are resolved negative complexes (by appearing in rows with negative outcomes). To increase its chance of happening, we may construct a set of pools obtained by taking intersection of R_i with a set M' of row vectors of the same length. Note that all these intersection rows preserve the property that no positive complex other than X_j can appear. Suppose R_i also covers a negative complex X_0. If M' has a row R' containing X_j but not X_0, then the intersection row R_iR' contains X_j but not X_0, and has a positive outcome. Of course, M' can also have a row $R"$ containing X_0 but not X_j. Then $R_iR"$ has a negative outcome. Therefore, by collecting the intersection rows with positive outcomes, the probability that X is the only complex appearing in all of them is increased.

As we do not know which row in M is R_i, typically we take intersections of every row in M with every row in M' to obtain a new pooling design M''. The pools in M'' are tested and analyzed to generate a set CP of candidates of positive complexes. Some positive complexes would be missed by CP, and some negative complexes would be wrongly picked up bu CP. So

$$P^+ = Prob(\text{ a positive complex not in } CP).$$

We eliminate unresolved negative complexes by introducing a second stage which confirms or rejects the candidates by individual testing.

A 2-stage design is evaluated by two criteria representing performance and cost, respectively.

(i) P^+,

(ii) the number of tests = the number of pools in $M'' + |CP|$.

Clearly, these two criteria depend on M, M' and CP. For the 2-stage designs studied in this section M is always an RID with parameter p, while M' can be either probabilistic or deterministic, or even related to M. We introduce two CP which have been studied in the literature.

Define

$$M''(i) = \{\text{row } (i, i') \text{ in } M'' \mid 1 \leq i' \leq t'\}$$

and

$$U(i) = \{\text{rows in } M''(i) \text{ with positive outcomes}\}.$$

Let $C(i)$ denote the set of columns containing $U(i)$. Macula, Torney and Villenkin first introduced:

The *complex* CP. $X \in CP$ if $\cap X = U(i)$ in $M''(i)$. The reason to exclude the $U(i) = \emptyset$ case is to avoid picking up too many negative clones which are simply not present in R_i.

Theorem 6.6.1 *For a 2-stage design M'' uder the complex CP,*

$$P^+ = 1 - \Phi(t, n, d-1, r) Prob(M' \textit{ contains a given complex}). \tag{6.6.10}$$

Proof. Let R_1 be a row in M which covers X_1 but none of the other positive complexes. Then each row in $U(i)$ does not cover any of the other positive complexes, and hence must cover X_1 to have a positive outcome. Hence $\cap X_1 \supseteq U(i)$ in $M''(i)$. But it is also clear that $\cap X_1 \subseteq U(i)$ since X_1 is positive. Hence $\cap X_1 = U(i)$ and $X_1 \in CP$. The second term in (6.6.10) is added to exclude the case $U(i) = \emptyset$. □

A negative complex X_0 can enter CP if both conditions (i) and (ii) are met:

(i) M has a row R_i covering X_0 and a nonempty set S of positive complexes.

(ii) Every row in M' covering X_0 covers a complex of S and vice versa, i.e., $\cap X_0 = \cup_{X_i \in S}(\cap X_i)$.

The requirements on covering positive complexes is to assure that the rows in $\cap X_0$ all have positive outcomes.

Let $P^{(-)}$ denote the probability $\cap X_0 = U(i)$, i.e., X_0 is misclassified into CP. Then

Theorem 6.6.2 *Under the complex CP,*

$$\begin{aligned} P^{(-)} &= \sum_{w=r}^{n} \binom{n}{w} p^w (1-p)^{n-w} \binom{w}{r} \binom{n}{r}^{-1} \\ &\quad \cdot \sum_{|S|=1}^{d} \sum_{s=r}^{w} \binom{w}{s} \left[\binom{\binom{s}{r} - I_{X_0 \in US}}{|S|} \binom{\binom{n}{r} - \binom{s}{r}}{d-|S|} \binom{\binom{n}{r}}{d}^{-1} \right] \cdot Prob(cond\ (ii)), \end{aligned}$$

where $I_{X_0 \in US}$ is the indicator function of $X_0 \in US$.

Proof. Suppose row i in M has weight w and covers X as well as the set S of positive complexes such that $|US| = s$. Then these s columns must be chosen from the w columns constituting the weight. The probability of choosing $|S|$ positive complexes from these s columns and the other $d - |S|$ positive complexes not from these s columns is the [ ] term. □

Corollary 6.6.3 $E(|CP|) = d\Phi(t, n, d-1, r)Prob($ *M' contains a given complex*$) + [\binom{n}{r} - d]P^{(-)}$.

Theorem 6.6.4 *Suppose M' is the complement (interchanging 0 and 1) of an r-separable matrix. Then $E(|CP|) \leq t$ under the complex CP.*

Proof. Let X and X' be two complexes covered by row i of M. Then the property $\cup X \neq \cup X'$ in an r-separable matrix is translated to $\cap X \neq \cap X'$ in M'. Hence at most one X satisfying $\cap X = U(i)$. □

In fact we can get an exact estimate of $E(CP)$ if the assumption of Theorem 6.6.4 is strengthened.

Theorem 6.6.5 *Suppose M' is the complement of an r-disjunct matrix. Then $E(|CP|) = d\Phi(t, n, d-1, r)$.*

Proof. Condition (ii) can never be realized since $\cap X_0 = \cup_{X_i \in S}(\cap X_i)$ implies the existence of an $X' \in S$ such that $\cap X' \subseteq \cap X_0$ in M', which further implies $cupX_0 \subseteq \cup X'$ in the r-disjunct matrix. Let $C \in X_0 \backslash X'$, then $C \subseteq \cup X'$, contradicting the definition of r-disjunctness. □

Note that under the assumption of Theorem 6.6.5, there is no need to conduct the second-stage since every complex in the CP set is positive.

Theorem 6.6.6 *Suppose M' is RID(p'). Then*

$$\begin{aligned} Prob(condition\ (ii)) \geq\ & \{\sum_{w=r}^{n} \binom{n}{w}(p')^w(1-p')^{n-w}[\binom{w}{r}\binom{n}{r}^{-1}(1-\binom{\binom{n}{r}-\binom{w}{r}}{|S|}) \\ & +(1-\binom{w}{r}\binom{n}{r}^{-1})\binom{\binom{n}{r}-\binom{w}{r}}{|S|}]\}^t. \end{aligned}$$

Proof. Given a row y with weight w, $\binom{w}{r}\binom{n}{r}^{-1}$ is the probability that y covers X_0, $\binom{\binom{n}{r}-\binom{w}{r}}{|S|}$ is the probability that y does not cover any complex of S. Then the formula gives the probability of the event E_y that the row covers X_0 if and only if it covers a complex of S. It is a lower bound since $\{E_y\}$ are positively correlated over the rows, while the formula treats them as independent. □

We can improve the 2-stage procedure by screening the complexes in CP before individually testing them. A complex X admitted to CP by satisfying $\cap X = U(i)$ can be removed if it appears in any test of $C(j)$ with a negative outcome. Further, suppose M has k rows covering X. Then con dition (ii) must be satisfied k times for X not to be removed. Usually, k is not very small and $P^{(-)}$ would then tend to 0, essentially eliminating the need of a second stage.

Macula, Torney and Villenkin proposed to set $M' = M$. Write M'' as M^2. Since $(i,j) = (j,i)$, M^2 contains only one of them labeled by $\{i,j\}$. Rows $\{i,j\}$, $1 \leq i \leq t'$, are also deleted. Thus M^2 has $\binom{t}{2}$ rows. Theorem 6.4.1 is reduced to

Theorem 6.6.7 *For M^2 under the complex CP, $P^+ = 1 - \Phi(t,n,d-1,r)$.*

Theorem 6.6.7 was given in [22] under the condition $|\cap X| \geq 2$ in M. But this condition is unnecessary since if $|\cap X| = 1$, then $U(i) = \emptyset$; but $\cap X_1 = \emptyset = U(i)$. So X is still in CP. The case $|\cap X| = 0$ is already counted in the formula for P^+.

For two complexes X_1 and X_2, let $\lambda(t,n,d,r)$ denote the probability that $\cap X_1 = \cap X_2$.

Theorem 6.6.8

$$\lambda(t,n,d,r) = \sum_{k=0}^{r} \binom{r}{k}\binom{n-r}{r-k}\binom{n}{r}^{-1}\{1-p^k+p^k[p^{2r-2k}+(1-p^{r-k})^2]\}^t.$$

Proof. Let $|X_1 \cap X_2| = k$. then given the r columns of X_1, $\binom{r}{k}\binom{n-r}{r-k}$ is the number of ways to choose X_2. To have $\cap X_1 = \cap X_2$, each row either covers both X_1 and X_2, or covers neither. The former event has probability $p^k p^{2(r-k)}$, while the latter has probability $(1-p^k) + p^k(1-p^{r-k})^2$, combining the probabilities of two mutually exclusive sub-events: either row misses one of the common k columns, or it has them all, but missing one of the non-common column in both X_1 and X_2. □

A formula of $|CP|$ is given in [17] using $\lambda(t, n, d, r)$.

Theorem 6.6.9 $E(|CP|) = t\binom{n}{r}\lambda(t, n, d, r)$.

Proof. For each row i, suppose there exists a positive complex X satisfying $\cap X = U(i)$. Then a complex X' can satisfying $\cap X' = U(i)$ only if $\cap X' = \cap X$ in M', with probability $\lambda(t, n, d, r)$. There are t choices for i, and $\binom{n}{r}$ (or $\binom{n}{r} - 1$) choices of X', hence the formula. □

This formula is not exact (besides the substraction of 1 from $\binom{n}{r}$) in two aspects:

(i) The probability of the condition $\cap X = U(i)$ is not included. Further, since this probability and $\lambda(t, n, d, r)$ are correlated, they cannot be simply multiplied together.

(ii) The formula does not take into consideration the event that even if $\cap X = U(i)$ is not satisfied for all positive X, we could still have $\cap X_0 = U(i)$ for some negative complex X_0, as described in the paragraph after Theorem 6.4.1.

Next we introduce the other CP. Let $C(i)$ denote the set of columns in M''_i containing $U(i)$.

The *column* CP. $X \in CP$ if $X = C(i)$ and $|C(i)| = r$.

If $C(i) = C(j)$, then only one of them needs to be tested in stage-2. By construction, $|CP| \leq t$.

Theorem 6.6.10 *For M'' under the column CP,*

$$P^+ = 1 - \Phi(t, n, d-1, r)Prob(given\ M,\ \cap X_0 \nsubseteq C\ for\ any\ column\ C\ in\ M').$$

Proof. The condition $\cap X_0 \nsubseteq C$ assures that C does not contain $U(i)$, i.e., $C \notin C(i)$. □

Corollary 6.6.11 *Suppose M' is an RID with parameter p'. Then under the column CP,*

$$P^+ = 1 - \Phi(t, n, d-1, r)\sum_{k=1}^{t'}\binom{t'}{k}[(p')^r]^k[1-(p')^r]^{t'-k}[1-(p')^k]^{n-r}.$$

Proof. k is the number of rows covering X_0 in M'. None of the $n-r$ columns not in X_0 can contain these r rows. □

Macula and Popyack used a different approach to approximate P^+. They compute the probability $\Phi(1,n,C,r)$ that a given row covers X but does not intersect the column C (call such a C a *success*):

$$\Phi(t,n,C,r) = 1 - [1-\Phi(1,n,C,r)]^t.$$

Since

$$1-\Phi(t,n,d-1,r) \neq [1-\Phi(t,n,C,r)]^{d-1},$$

due to positive correlation between $\Phi(1,n,C,r)$ and $\Phi(1,n,C',r)$, they turned to assymptotc analysis by computing the expected number of successful C to be

$$\beta = (n-r)[1-(1-p)(p)^r]^t, \tag{6.6.11}$$

which can be approximated by a Poisson variable with mean β. Then the probability that this number is 0 is $e^{-\beta}$. Thus

Theorem 6.6.12

$$P^+ \sim 1 - e^{-\beta}. \tag{6.6.12}$$

Macula and Popyack actually allowed r_i to vary from complex to complex. Let $r^* = \max r_i$. They modified the column CP by admitting $C(i)$ to CP if $|C(i)| \leq r^*$. For each $C(i) \in CP$, test $C(i) \setminus C$ for each $C \in C(i)$ and confirm $C(i)$ as a positive complex only if $C(i) \setminus C$ tests negative for all $C \in C(i)$. Note that the number of stage-2 tests is inflated to $|CP|(r^*+1)$.

Macula, Rykov and Yekhanin observed that the complement (exchanging 0 with 1) of a r-disjunct matrix satisfies the requirement that for any $r+1$ columns $C_0, C_1, ..., C_r$, there exists a row intersecting $C_1, ..., C_r$, but not C_0. Let $C_1, ..., C_r$ be the columns in a positive complex X. Then for each column C_0 not in X, M' has a row covering X but not C_0. To reduce the number of rows in M' they proposed to use the complement of an *α-almost r-disjunct matrix*, meaning the probability that a random set X of r columns has probability at least α to have no other column C containing $\cap X$. Then Theorem 6.6.10 is reduced to

Theorem 6.6.13 *Suppose M' is the complement of an α-almost r-disjunct matrix. Then under the column CP,*

$$P^+ = 1 - \Phi(t,n,d-1,r)\alpha.$$

Macula, Rykov and Yekhanin also commented that the q-nary MDS code used in Chapter 3 to construct d-disjunct matrices can be used to construct α-almost r-disjunct matrices with $r > d$ and $\alpha \to 1$.

6.7 Sequential Algorithms for H_r

Chang and Hwang [3] first cast the group testing problem on graphs. They formulated the problem of identifying a unique positive item in a set A and a unique positive item in a set B as a problem of identifying a unique edge in a bipartite graph $G(A, B)$. Throughout this section, we assume G (or H) is a graph (hypergraph) with n vertices and edge set E.

Theorem 6.7.1 $t(G,1) = \lceil \log |E| \rceil$ *if* G *is a complete bipartite graph.*

Note that $\lceil \log |E| \rceil$ is the trivial information-theoretic lower bound. Chang and Hwang also conjectured that Theorem 6.7.1 holds even if the bipartite graph is not complete, but contains exactly 2^k edges for some k.

The proof of Theorem 6.7.1 uses group tests (as we point out in Sec. 6.1 that for $d = 1$ group tests can be translated into edge-tests). Aigner [1] brought out the edge-test notion explicitly and conjectured for a general graph G,

$$t(G,1) \leq \lceil \log |E| \rceil + c.$$

Du and Hwang [6] sharpened the conjecture by setting $c = 1$, which was proved by Damaschke [5].

Theorem 6.7.2 $t(G,1) \leq \lceil \log |E| \rceil + 1.$

There are many examples of G for which $t(G,1) = \lceil \log |E| \rceil + 1$.

Triesch [23] extended Theorem 6.7.2 to H_r by proving

Theorem 6.7.3 $t(H_r,1) \leq \lceil \log |E| \rceil + r - 1.$

Triesch gave the proof using the group-test (as versus edge-test) terminology for easier description. Every vertex in the edge in D is considered a *positive vertex.*

A *vertex-cover* of H is a vertex-set which intersects every edge of H. Choose a vertex-cover $C = \{v_1, ..., v_s\}$ by the following greedy algorithm. Let $H^1 = H$ and v_1 be the vertex with maximum degree in H^1. For $2 \leq i \leq s$, let H^i be the hypergraph obtained from H^{i-1} by deleting $v_1, ..., v_{i-1}$ (and their edges), and let v_i denote the vertex with maximum degree in H^i. Let $d_{H^i}(v_i)$ denote the degree of v_i in H^i. It is easily verified

$$d_{H^1}(v_1) \geq d_{H^2}(v_2) \geq \cdots \geq d_{H^s}(v_s).$$

Define

$$\ell_i = \lceil \log |E| / d_{H^i}(v_i) \rceil.$$

Then

$$\sum_{i=1}^{s} 2^{-\ell_i} \leq \sum_{i=1}^{s} 2^{-\log \frac{|E|}{d_{H^i}(v_i)}} = \sum_{i=1}^{s} \frac{d_{H_i}(v_i)}{|E|} = 1.$$

Now, we give the proof of Theorem 6.7.3.

Proof of Theorem 6.7.3. It is proved by induction on the rank r. For $r = 1$, the problem is simply the classic group testing problem with one positive item and hence $t(H_1, 1) = \lceil \log n \rceil = \lceil \log |E| \rceil$. For $r \geq 2$, consider the vertex cover $C = \{v_1, \cdots, v_s\}$ obtained by the greedy algorithm. Since $\sum_{i=1}^{s} 2^{-\ell_i} \leq 1$, by Kraft's Inequality, there is a binary search tree T with leaves $v_1, v_2, \cdots, v_s$ ordering from left to right such that the length of the path from the root to leaf v_i is at most l_i. Each internal node u of T is associated with a test whether there exists a positive vertex among the leaves under the left son of u. Since C is a vertex cover, there must exist a positive vertex in C. Let v_i denote the positive vertex. Denote by T_a the subtree of T rooted at vertex a. One searches this v_i as follows.

```
begin
   Ininitially, set $a :=$ the root of $T$
   while $a$ is not a leaf
         do begin $b :=$ the left son of $a$;
                  test on $\{v_j \mid j$ is a leaf of $T_b\}$;
                  if the outcome is negative
                     then $a :=$ the right son of $a$
                     else $a := b$;
         end-while;
   $v_i := a$;
end.
```

Note that this algorithm can find a positive vertex v_i through at most l_i tests. Moreover, when v_i is found, all $v_1, \cdots, v_{i-1}$ have been found to be good. Therefore, the remaining positive vertices are those adjacent to v_i in H_i. The total number of them is $d_{H_i}(v_i)$. Removing v_i from them results in a hypergraph of rank at most $r - 1$. By the induction hypothesis, $\lceil \log d_{H^i}(v_i) \rceil + r - 2$ tests are enough to identify the remaining positive vertices. Therefore, the total number of tests is at most

$$l_i + \lceil \log d_{H^i}(v_i) \rceil + r - 2 \leq \lceil \log |E| \rceil + r - 1.$$

□

As commented in Sec 6.1, a group test on $S = \{v_1, ..., v_k\}$ can be connected to an edge test in $V \setminus S$ without affecting the testing tree structure, except a "yes" answer to the group test corresponds to a "no" answer to the edge-test. Johann [12] observed that the edge-test version of the Triesch algorithm can still identify a positive edge even when $d > 1$. We will refer to this modification as the *TJ-procedure.* Note that the group testing version identifies the first positive edge while the edge-test version the last; so that their excutions may result in different number of tests. But their worst-case performances are identical even through different paths are travelled.

For H a complete r-hypergraph, the $(H*_r, 1)$ problem is reduced to the classical group testing with n items including r positive ones.

Corollary 6.7.4 $t(r, n) \leq \lceil \log \binom{n}{r} \rceil + r - 1$.

Corollary 6.7.4 is known in classical group testing (Corollary 2.4.2 in [7]).

For $r = 1$, Theorem 6.7.3 is reduced to Theorem 6.7.1. For $d > 1$, Du and Hwang [7] conjectured

$$t(H : d) \leq d(\lceil \log(|E|/d) \rceil + c)$$

for some constant c. (Note that $d \log(|E|/d)$ is the leading term of the information bound.) Johann [12] proved the conjecture with $c = 7$ when H is a graph G. We improve the constant to 6.

Theorem 6.7.5 $t(G : d) \leq d((\lceil \log(|E|/d) \rceil + 6)$.

The proof of Theorem 6.7.5 is based on the construction of a class of algorithms A_ℓ for integer ℓ which requires at most $d(\lceil \log \ell \rceil + 5 - 1/\ell) + |E|/\ell + 1$ tests. By setting $\ell = \lfloor |E|/d \rfloor$, this quantity is at most $d((\lceil \log(|E|/d) \rceil + 6)$.

The algorithm A_ℓ depends crucially on three ideas. The first is that each positive edge is identified by the TJ-procedure (with $r = 2$) in $\lceil \log |E| \rceil + 1$ tests. The second is a method to avoid repeatedly identifying the same positive edge. The third is that it is possible to find a subset of untested edges with proper size under some general condition. Namely,

Lemma 6.7.6 *Suppose S and S' are two disjoint subsets of V and $1 \leq \ell \leq |E|/2$ such that*

(i) $|E(S') \cup E(S, S')| \geq \ell$,

(ii) $\forall v \in S', |E(v, S)| \leq 2\ell$.

Then there exists a subset $W \subseteq S'$ with

$$\ell \leq |E(W) \cup E(W, S)| \leq 2\ell.$$

Proof. Let $\Gamma(v)$ denote the set of v's neighbors. If there is a vertex $v \in S'$ satisfying

$$\ell \leq |\Gamma(v) \cap S| \leq 2\ell,$$

then set $W = \{v\}$ will do. So we may assume $|\Gamma(v) \cap S| < \ell$ for every $v \in S'$. Let $S' = \{v_1, ..., v_k\}$, and let

$$s = \max\{i \in \{1, ..., k\} \mid E(\{v_1, ..., v_i\}) \cup E(\{v_1, ..., v_i\}, S)| < \ell\}.$$

Because of (i), we have $1 < s < k$. Let $W_1 = \{v_1, ..., v_{s+1}\}$. Then

$$|E(W_1) \cup E(W_1, S)| \geq \ell$$

holds. If

$$|E((W_1) \cup E(W_1, S)| \leq 2\ell,$$

then set $W = W_1$ will do. Otherwise, define sets $W_2, W_3, ..., W_s$ recursively through

$$W_i = W_{i-1} \setminus \{v_{i-1}\}, i = 2, ..., s.$$

Since

$$\begin{aligned}
& |E(W_{i-1})| + |E(W_{i-1}, S)| - [|E(W_i)| + |E(W_i, S)|] \\
= \;& |E(W_{i-1})| - |E(W_i)| + |E(W_{i-1}, S)| - |E(W_i, S)| \\
\leq \;& |E(\{v_{i-1}, \{v_i, ..., v_s\})| + 1 + |E(\{v_{i-1}\}, S)| \\
\leq \;& |E(\{v_1, ..., v_s\})| + |E(\{v_1, ..., v_s\}, S))| + 1 \\
\leq \;& \ell,
\end{aligned}$$

one of W_i, $2 \leq i \leq s$, will do. □

The algorithm A_ℓ consists of two steps:

Step 1. A_ℓ partitions the n vertices into a number of sets $V_1, ..., V_h$ such that each V_i contains no positive edge and for each $v \in V_i$, $1 \leq i \leq h$, v has a positive edge with each V_j, $j < i$.

Step 2. For each $v \in V_i$, $2 \leq i \leq h$ and $1 \leq j \leq i-1$, identify all edges in $\{v\} \cup V_j$.

At the beginning, set $S = \emptyset$ and $S' = V$. In general, when $V_1, ..., V_k$ are nonempty, set $S = V_1$ and $S' = V \setminus \{V_1, ..., V_k\}$. Call a vertex $v \in S'$ *heavy* if $|\Gamma(v) \cap S| > 2\ell$.

For every heavy vertex $v \in S'$, let S_1 denote a set of 2ℓ neighbors of v in S. Test $v \cup S_1$. If positive, identify a positive edge (v, u) and assign v to the first V_i it has no neighbor. If negative, set $\Gamma(v) = \Gamma(v) \setminus S_1$ and repeat the procedure (even though v may not be heavy any more) until $\Gamma(v) = \emptyset$ (the last S_1 may contain fewer than 2ℓ vertices).

Next, we consider the case that S' contains no heavy vertex. If $|E(S') \cup E(S, S')| \geq \ell$, use Lemma 6.7.6 to find the subset $W \subseteq S'$; if $|E(S') \cup E(S, S')| < \ell$, set $W = S'$. Test $W \cup S$. If negative, assign W to V_1; otherwise, identify a positive edge (u, v). Assign u and v to satisfy the requirement stated in Step 1. Update S and S', and repeat the procedure until $S' = \emptyset$.

We now describe the "assigning" part in more detail. If the test on W is negative, assign W to V_1; otherwise identify a positive edge (u, v) by the TJ-procedure and assign u, v to some V_i and V_j satisfying the condition in Step 1. We show that we only need to go through the assigning procedure for one of them.

Suppose the TJ-procedure identifies vertex v in the vertex-cover. Among the set of vertices adjacent to u, order them so that those in V_I are at the head of order.

Now apply the TJ-procedure to identify the first vertex u such that (v, u) is a positive edge. If $u \in V_1$, then we only need to assign v. If $u \notin V_1$, then V_1 has no positive edge to any vertex in W, in particular, to either u or v, hence v can be assigned to V_1 without any testing.

To assign v, we examine whether $v \cup V_i$, $i = 2, 3, ..., k$ contains a positive edge in that order until we find a V_i such that $v \cup V_i$ contains no positive edge. Then we assign v to V_i. If no such V_i exists, we create a new V_{k+1} to store v. To examine $v \cup V_i$, it suffices to consider $\Gamma_i(v) = \Gamma(v) \cap V_i$. If $|\Gamma_i(v)| \leq \ell$, set $W = v \cup \Gamma_i(v)$. Otherwise, let W be an arbitrary ℓ-subset of $\Gamma_i(v)$. Test $v \cup \Gamma_i(v)$. If negative, set $\Gamma_i(v) = \Gamma_i(v) \setminus W$ and do the same. If we reach $\Gamma_i(v) = \emptyset$, then assign v to V_i. Otherwise, at some stage $v \cup W$ is positive. Use the TJ-procedure to identify a positive edge (v, w). Examine $v \cup V_{i+1}$.

In identifying the positive edges between v and V_j in Step 2, we first remove all u from the latter if (u, v) has already been identified, either as a positive edge or as a negative edge. Let $\Gamma'(v)$ be the set of neighbors of v in V_j. If $|\Gamma'(v)| \leq \ell$, test $v \cup \Gamma'(v)$. Otherwise, let $W \subseteq \Gamma'(v)$ be of size ℓ. Test $v \cup W$. If negative, set $\Gamma'(v) = \Gamma'(v) \setminus W$. If positive, identify a positive edge (v, u) and set $\Gamma'(v) = \Gamma'(v) \setminus \{u\}$. Repeat the procedure with update $\Gamma'(v)$.

We analyze the number of tests required in Step 1. We first count the number of tests consumed in identifying positive and negative edges. Call a test with negative outcome a *bad test* if it contains fewer than ℓ edges. In Step 1 each positive edge in an m-set is identified by at most $\lceil \log m \rceil + 2$ tests, one more than the Triesch's result due to the additional test on $W \cup S$. Each time a positive vertex is assigned, the last test could be bad (all other tests, even negative, are counted in the $\lceil \log m \rceil + 2$ tests in identifying a positive edge). Further, the test on the last S' can be bad. Note that for a heavy vertex, although the test on the last S_1 can be bad, this test must be proceeded by a test on 2ℓ negative edges. So these two tests average out to at least ℓ edges, i.e., we need not count the bad test.

Let d_1 denote the number of positive edges identified and n_1 the number of nonbad tests taken in Step 1. If a positive edge is identified in testing $E(W) \cup E(W \cup V_1)$, then it takes one test for the initial test and at most $\lceil \log 2\ell \rceil + 1$ tests in using the TJ procedure (fewer tests for a heavy item). Further, there is at most one bad test among all tests of this type. If a positive edge is identified in assigning a vertex to some V_i then it takes $\lceil \log \ell \rceil + 1$ tests in using the TJ procedure and at most one bad test can occurs among the negative tests. Therefore the total number of tests in Step 1 is at most

$$d_1(\lceil \log 2\ell \rceil + 2) + n_1 + d_1 + 1$$

In Step 2, there are at most d_1 vertices not in V_1 since each of them implies a distinct positive edge. In checking the positive edges of $v \in V_i$, the test on $v \cup V_j$ may contain one bad test. But again, each bad test can be assigned to the identification of a positive edge. Let d_2 and n_2 denote the counterparts of d_1 and n_1 in Step 2. Then the number of tests in Step 2 is at most

$$d_2(\lceil \log \ell \rceil + 2) + n_2 + d_1$$

Adding up, the total number of tests is at most

$$(d_1 + d_2)(\lceil \log \ell \rceil + 5) + n_1 + n_2 + 1.$$

By noting

$$d_1 + d_2 = d$$

$$(n_1 + n_2)\ell \leq |E| - d,$$

we obtain an upper bound

$$d(\lceil \log \ell \rceil + 5) + \frac{|E| - d}{\ell} + 1 = d(\lceil \log \ell \rceil + 5 - \frac{1}{\ell}) + \frac{|E|}{\ell} + 1.$$

The proof of Theorem 6.7.5 is completed.

Recently, Hwang [11] gave a competitive algorithm for graphs with d unknown. Chen and Hwang [4] extended to hypergraphs. They followed Johann's approach in general , but had to resolve some problems unique to hypergraphs.

In Johann's algorithm, each positive edge is broken into two subsets. Should a rank-r edge be broken into two subsets, r subsets or something in between? Later, when searching for positive edges between two subsets, Johann simply takes one vertex u from one subset and then remove $\{(v \mid (u, v)$ is an identified positive edge$\}$ from the other subset to avoid the identification of a positive edge already identified. For the hypergraph case, if an r-edge is broken into r subsets, then we need to mix vertices from more than two subsets; if not, then we need to take more than one vertex from a subset. How do we avoid the identified positive edges? Finally, there is the difficulty in analysis which, to a large degree, depends on the choice of breaking a positive edge in (i).

To avoid being overly complicated, Chen and Hwang chose the simplest setting of partitioning each positiv e edge into two subsets V_0 and V_1. At the b egining $V_0 = V$. Test V_0, if positive, use the TJ-procedure to identify a positive edge e. Assign an arbitrary vertex v of e to V_1 and set $V_0 = V_0 \setminus \{v\}$. Test V_0 again. Do this until testing outcome on V_0 is negative. There are still unidentified positive edges e with at least one vertex in V_1. These edges are identified essentially through an enumeration process. Let K be a nonempty subset of V_1. We identify all positive edges contained in $K \cup V_1$. But since all positive edges of the type $K' \cup V_1'$, where $K' \subset K$ and $V_1' \subseteq V_1$ were identified easier when K was set to be K', we only identify positive edges of the type $K \cup V_1'$, where $V_1' = V_1$. This is done by solving a subproblem where $e' \subseteq V_1$ is a positive edge in the subproblem if and only if $K \cup e'$ is a positive edge in the origin al problem.

To avoid the identified positive edges, let V' be the vertex-set of the subproblem and $C \in V'$ a vertex-cover of the positive edges. Then C is moved from V' to V_1' before any testing. Thus in testing V', we will never encounter a positive edge.

Note that the subproblem is same as the original problem except

(i) r is changed to $r - |K|$,

(ii) K and C are attached to the subproblem.

Thus we will enlarge the original problem to allow K and C (both equal to $\emptyset$ in the original problem) so that the problem can be solved recursively. Note than in the subproblem when the maximum rank is 1 and K is given. Then an identified positive edge can be avoided by removing $\{v \in V \mid K \cup V$ is an identified positive edge$\}$.

By inspecting the algorithm, we note that each positive edge is either identified by the TJ-procedure during a partition stage, or by a direct but during a search stage. Thus every positive edge is identified in $d(\log |E| + r - 1)$ tests. $CH(r)$ does not attempt to optimize the test size as in Johann's algorithm (and pay a price of increasing the leading term from $d \log(|E|/d)$ to $d \log |E|$, but still manage to control the number of negative tests. Since the parameter d is used in the algorithm only in determin ing the optimal size, $CH(r)$ assumes no knowledge of d and is thus a competitive algorithm.

We first give an algorithm for r-hypergraphs.

Let K_i denote the subset imposed on $CH(i)$, i.e., K_i is a part of every test in $CH(i)$. If the original problem is defined on an r-hypergraph, then $|K_i| = r - i$. The vertices V_i of $CH(i)$ are divided into V_{i0} and V_{i1}. Define $E(K_i) = \{e \subseteq V_i \mid e \cup K_i \in E\}$ and $D(K_i) = \{e \subseteq V_i \mid e \cup K_i \in D\}$. Finally, let I denote the set of currently identified positive edges (in the original problem). Define $I(K_i) = \{e \subseteq V_i \mid e \cup K_i \in I\}$. For $CH(r)$, $V_{i0} = V$ and $V_{i1} = K_r = I = \emptyset$. Thus $E(K_r) = E$, $D(K_r) = D$ and $I(K_r) = \emptyset$.

We first define $CH(1)$ and then give a recursive algorithm $CH(i)$.

Algorithm $CH(1)$

Input: K_1, V_{10}, V_{11}, I. Attach K_1 to every test.

Step 1. Test V_{10}. If positive, use the halving procedure, which we will treat as a special case of the TJ-procedure, to identify a positive vertex u. Set $I := I \cup \{e\}$, $V_{10} := V_{10} \setminus \{u\}$ and go back to the beginning of Step 1. If negative, go to Step 2.

Step 2. For every vertex $v \in V_{11}$, test $K_1 \cup \{v\}$. If positive, set $I := I \cup \{K_1 \cup \{v\}\}$. If negative, stop.

Algorithm $CH(i)$ $(i \geq 2)$

Input: E, D, K_i, V_{i0}, V_1, I. Attach K_i to every test.

Partition Stage:

Step 1. Test V_{i0}. If positive, use the TJ-procedure to identify a positive edge $e = \{v_1, v_2, ..., v_i\} \subseteq V_{i0}$. Add the vertex v_1 to V_{i1}. Set $V_{i0} := V_{i0} \setminus \{v_1\}$ and $I := I \cup \{K_i \cup \{v\}\}$. If $V_{i0}| \geq i$, go back to the beginning of Step 1.

Step 2. If V_{i1} is nonempty, go to the search stage.

Search Stage:

Step 1. Set $k = 1$.

Step 2. Let K be a k-subset of V_1. Set $K_{i-k} = K_i \cup K$. Construct a vertex cover C on $I(K_{i-k})$. Call subroutine $CH(i-k)$ with $V_{i-k,0} = V_{i0}$, $V_{i-k,1} = C$, K_{i-k} and I. Do this for all k-subsets K. Set $k := k+1$. If $k < i$, go back to the beginning of Step 2.

Step 3. Test all i-subsets S (except those such that $S \cup K_i \in I$) of V_{i1}. If positive, set $I := I \cup \{S \cup K_i\}$.

Step 4. Stop.

We will refer to tests in Step 3 as *direct hits.*

Theorem 6.7.7 *Let E be an arbitrary r-hypergraph which contains d positive edges, where d is not necessarily known. Then the algorithm $CH(i)$ identifies all positive edges of E with at most $d\lceil \log_2 |E| \rceil + (i-1)^{\lfloor i/2 \rfloor} d^i + o(d^i)$ tests.*

Proof. Clearly, all edges identified as positive by the algorithm are through either the TJ-procedure or direct hits, both are error-free. Thus it suffices to prove that a positive edge is always identified.

Suppose a positive edge with vertex set X is not identified at the partition stage of $CH(i)$. Then a nonempty subset $X' \subseteq X$ must lie in V_{i1}. At the search stage, the selection of K runs through all k-subsets of V_{i1} for $1 \leq k \leq i$. One such selection is $K = X'$. Suppose $|X'| = k$. Then the problem is reduced to the subroutine $CH(i-k)$ with K imposed. By induction on i, the induced positive edge $X \setminus K$ can be identified in the subroutine, which implies $(X \setminus K) \cup K = X$ is a positive edge.

It remains to count the number of tests $CH(i)$ uses. Note that TJ-procedure uses at most $\lceil \log_2 |E| \rceil + i$ tests. Since a positive edge is identified by either the TJ-procedure or a direct hits, the number of tests consumed in identifying one positive edge is bounded by $\lceil \log_2 |E| \rceil + i$. This bounded number of tests includes the possible positive test initiating the identification process, and all negative tests occurred during the process of identifying the positive edge. Thus, the number of tests identifying d positive edges is at most $d(\lceil \log_2 |E| \rceil + i)$.

Further, it suffices to count the number $N(i)$ of negative tests occured elsewhere in $CH(i)$. There are three sources for negative tests in $N(i)$: one negative test from the partition stage, those from the subroutines and direct hits.

Denote D_k as the set of all positive edges in $K \cup V(E_0(K))$. Let $d_K = |D_K|$. Note that for $K \neq K'$, D_K and $D_{K'}$ may overlap in positive edges containing some vertices in $K \cap K'$ and other vertices in $V(E_0(K))$. Hence we can only bound d_K by d. However for $|K| = 1$, D_K and $D_{K'}$ are disjoint; hence $\sum_{K:|K|=1} d_K$ is bounded by d. We count the number $N(i)$ of negative tests in $CH(i)$ by induction on i.

For $i = 1$, $N(i)$ is easily verified to be at most 1. Since each positive vertex can be identified by the halving procedure in $\lceil \log_2 |E| \rceil$ tests, Theorem 6.7.7 holds for $i = 1$.

We prove the general $i \geq 2$ case by induction.

$$\begin{aligned}
N(i) &\leq 1 + \sum_{k=1}^{i-2} \sum_{K \subseteq V_{i1}:|K|=k} N(i-k) + \sum_{K \subseteq V_{i1}:|K|=i-1} N(1) + \binom{|V_{i1}|}{i} \\
&\leq \sum_{k=1}^{i-2} \sum_{K \subseteq V_{i1}:|K|=k} ((i-k-1)^{\lfloor \frac{i-k}{2} \rfloor} d_K^{i-k} + o(d_K^{i-k})) + \sum_{K \subseteq V_{i1}:|K|=i-1} (1 + d_k) + d^i \\
&\leq \sum_{K \subseteq V_{i1}:|K|=1} (i-2)^{\lfloor \frac{i-1}{2} \rfloor} d_K^{i-1} + \sum_{k=2}^{i-2} \sum_{K \subseteq V_{i1}:|K|=k} ((i-k-1)^{\lfloor \frac{i-k}{2} \rfloor} d_K^{i-k} + d^{i-1}(1+d) + d^i + o(d^i) \\
&\leq (i-2)^{\lfloor \frac{i-1}{2} \rfloor} \sum_{K \subseteq V_{i1}:|K|=1} d_K^{i-1} + (i-3)^{\lfloor \frac{i}{2} \rfloor - 1} \sum_{k=2}^{i-2} \binom{d}{k} d^{i-k} + 2d^i + 2d^i + o(d^i) \\
&\leq (i-3)^{\lfloor \frac{i}{2} \rfloor - 1} (i-3) d^i + 2d^i + o(d^i) \\
&= ((i-3)^{\lfloor \frac{i}{2} \rfloor} + 2) d^i + o(d^i) \\
&\leq (i-3)^{\lfloor \frac{i}{2} \rfloor} d^i + o(d^i).
\end{aligned}$$

Thus, $N(i) \leq (i-1)^{\lfloor \frac{i}{2} \rfloor} d^i + o(d^i)$ holds for general i.

Let $T(i)$ denote the total number of tests required by $CH(i)$. Since $T(i) \leq d(\lceil \log_2 |E| \rceil + i) + N(i)$ and $N(i) \leq (i-1)^{\lfloor \frac{i}{2} \rfloor} d^i + o(d^i)$, we have $T(i) \leq d(\lceil \log_2 |E| \rceil + i) + (i-1)^{\lfloor \frac{i}{2} \rfloor} d^i + o(d^i)$ for $i \geq 2$. Therefore, algorithm $CH(i)$ needs at most $d(\lceil \log_2 |E| \rceil + i) + (i-1)^{\lfloor \frac{i}{2} \rfloor} d^i + o(d^i)$ tests to identify all d positive edges in E.

We now extend the algorithm to general H. Let E be a hypergraph of rank i, i.e., $|e| \leq i$ for all edges $e \in E$. To identify the set $D \subset E$ in a hypergraph, we follow the general approach in algorithm $CH(i)$ for i-hypergraph with a slight modification. Let $CH*(i)$ denote the algorithm for hypergraph of rank i.

The search stage in $CH*(i)$ will be a little different from $CH(i)$. When we choose a k-subset K of V_{i1} before constructing a vertex cover and then calling $CH*(i-k)$, we should test K itself. If the outcome is positive, then $K \in D$. By our assumption, there is no other positive edge containing K, so we do not need to call $CH*(i-k)$ further. If the outcome is negative, call $CH*(i-k)$ to identify all induced positive edges in $E_0(K)$.

Algorithm $CH*(1)$ is same as $CH(1)$. Now, we give the algorithm $CH*(i)$ recursively.

Algorithm $CH*(i)$ $(i \geq 2)$

input: E, K_i, V_{i0}, V_{i1}, I (if $CH*(i)$ is not a subroutine, then $V_{i0} := V(E)$ and $V_{i1} := K_i := I := \emptyset$).

Partition Stage

Step 1. Test V_{i0}. If positive, use the TJ-procedure to identify a positive edge $e = \{v_1, v_2, ..., v_s\} \subseteq V_{i0}$ $(s \leq i)$. Add the vertex v_1 to V_{i0}. Set $V_{i0} := V_0 \setminus \{v_1\}$ and $I := I \cup \{\{e\} \cup K_i\}$. If $V_{i0} \neq \emptyset$, go back to the beginning of Step 1.

Step 2. If $V_{i1} \neq \emptyset$, go to Search Stage.

Search Stage:

Step 1. Set $k := 1$.

Step 2. Choose a k-subset K of V_{i1}, where $G(K_i \cup K)$ does not contain any identical positive edge in I. Set $K_{i-k} := K_i \cup K$. Test K_{i-k}. If positive, let $I := I \cup \{K_{i-k}\}$. Else, construct a vertex cover C $(C \cap K_{i-k} = \emptyset)$ on $I(K_{i-k} \cup V(E_0(K_{i-k})))$. Call subroutine $CH*(i-k)$ with $E := E_0(K_{i-k})$, $V_i := V(E_0(K_{i-k}))$, $V_{i0} := V(E_0(K_{i-k})) \setminus C$ and $V_{i1} = C$. If for some $v \in V_{i0}$, $K_{i-k} \cup \{v\} \in I(K_{i-k} \cup V(E_0(K_{i-k})))$ (possibly only for $k = i-1$), delete v from V_{i0}. Attach K_{i-k} to any test in $CH*(i-k)$. Do this for all k-subsets K. Set $k := k+1$. If $k < i$, go back to the beginning of Step 2.

Step 3. Test all i-subsets S (except those such that $S \cup K_i \in I$) of V_{i1}. If positive, set $I := I \cup \{S \cup K_i\}$.

Step 4. Stop.

Theorem 6.7.8 *Let E be a hypergraph of rank r with d positive edges, where d is not necessarily known. The the algorithm $CH*(r)$ identifies all positive edges in E with at most $d\lceil \log_2 |E| \rceil + (r-1)^{\lfloor \frac{r}{2} \rfloor} d^r + o(d^r)$ tests.*

Proof. Similar to the proof of Theorem 6.7.7, we can show that $CH*(r)$ identifies all positive edges of the hypergraph E.

To count the number of tests $CH*(r)$ uses, let $N*(r)$ and $T*(r)$ be the counterparts of $N(r)$ and $T(r)$ in $CH*(r)$. The analysis of the test number of $CH*(r)$ is also similar to that of $CH(r)$. The only difference is that the subroutine of $CH*(r)$ should need $N*(r-k)+1$ tests instead of $N*(r-k)$ tests in $CH(r)$. But it does not change the result; so $N*(r) \leq (r-1)^{\lfloor \frac{r}{2} \rfloor} d^r + o(d^r)$. Consequently, $T*(r) \leq d\lceil \log_2 |E| \rceil + (r-1)^{\lfloor \frac{r}{2} \rfloor} d^r + o(d^r)$ for $r \geq 2$. Therefore, algorithm $CH*(r)$ needs at most $d\lceil \log_2 |E| \rceil + (r-1)^{\lfloor \frac{r}{2} \rfloor} d^r + o(d^r)$ tests to identify all d positive edges of E. □

References

[1] M. Aigner, Search problems on graphs, *Disc. Appl. Math.* 14 (1986) 215-230.

[2] H.B. Chen, D.-Z. Du and F.K. Hwang, An unexpected meeting of four seemily unrelated problems: graph testing, DNA complex secreening, superimposed codes and secure key distribution, preprint, 2005.

[3] G.J. Chang and F.K. Hwang, A group testing problem, *SIAM J. Alg. Disc. Methods* 1 (1980) 21-24.

[4] T. Chen and F.K. Hwang, A competitive algorithm in searching for many edges in a hypergraph, 2003, preprint.

[5] P. Damaschke, A tight upper bound for group testing in graphs, *Disc. Appl. Math.* 48 (1994) 101-109.

[6] D.-Z. Du and F.K. Hwang, *Combinatorial Group Testing and Its Applications*, World Scientific, Singapore, 1993.

[7] D.-Z. Du and F.K. Hwang, *Combinatorial Group Testing and Its Applications*, 2nd edition, World Scientific, Singapore, 2000.

[8] D.-Z. Du, F.K. Hwang, W. Wu and T. Znati, A new construction of transversal designs, to appear in *Journal of Computational Biology.*

[9] D.-Z. Du, F.K. Hwang, M. Thai, W. Wu and T. Znati, Construction of disjunct matrices for group testing in the complex model, manuscript.

[10] A. Dyachkov, P. Villenkin, A. Macula and D. Torney, On families of subsets where no intersection of ℓ-subsets is covered by the union of s others, *J. Combin. Thy. (Series A)*, 99 (2002) 195-218.

[11] F. K. Hwang, A competitive algorithm to find all defective edges in a graph, *Disc. Appl. Math.* 148 (2005) 273-277.

[12] P. Johann, A group testing problem for graphs with several defective edges, *Disc. Appl. Math.* 117 (2002) 99-108.

[13] H.K. Kim and V. Lebedev, On optimal superimposed codes, *J. Combin. Design* 12 (2004) 79-91.

[14] Y. Li, M. Thai, Z. Liu and W. Wu, Protein-to-protein interactions and group testing in bipartite graphs, to appear in *International Journal of Bioinformatics and Applications.*

[15] A. J. Macula and L. J. Popyack, A group testing method for finding patterns in data, *Disc. Appl. Math.* 144 (2004) 149-157.

[16] A.J. Macula, V.V. Rykov and S. Yekhanin, Trivial two-stage group testing for complexes using almost disjunct matrices, *Disc. Appl. Math.*

[17] A. J. Macula, D.C. Torney and P.A. Villenkin, Two-stage group testing for complexes in the presence of errors, *DIMACS Series in Disc. Math. and Theor. Comput. Sci.* 55 (2000) 145-157.

[18] C.J. Mitchell and F.C. Piper, Key storage in secure networks, *Disc. Appl. Math.* 21 (1988) 215-228.

[19] D.R. Stinson, On some methods for unconditionally secure key distribution and broadcast encryption, *Designs, Codes, Crypto.* 12(1997) 215-343.

[20] D.R. Stinson and R. Wei, Generalized cover-free families, *Disc. Math.* 27 (2004) 463-477.

[21] D.R. Stinson, R. Wei and L. Chu, Some new bounds for cover-free families, *J. Combin. Thy. Series A* 90 (2000) 224-234.

[22] D.C. Torney, Sets pooling designs, *Ann. Combin.* 3(1999) 95-101.

[23] E. Triesch, A group testing problem for hypergraphs of bounded rank, *Disc. Appl. Math.* 66 (1996) 185-188.

[24] H. Wang and C. Xing, Explicit construction of perfect hash families from algebraic curves over finite fields, *J. Combin. Thy. Ser. A* 93 (2001) 112-124.

[25] S. Yakhanin, Some properties of superimposed codes based on MDS codes, preprint, 1999.

[26] Yu. L. Sagalovich, On separating systems, *Problemy Peredachi Informatsii* 30 (1994) 14-35 (in Russian).