# Non-Adaptive Complex Group Testing with Multiple Positive Sets

Francis Y.L. Chin, Henry C.M. Leung, S.M. Yiu

*Department of Computer Science, The University of Hong Kong, Pokfulam, Hong Kong*

**Abstract**

Given $n$ items with at most $d$ of them having a particular property (referred as positive items), a test on a selected subset of them is positive if and only if the subset contains at least one positive item. The non-adaptive group testing problem is to design how to group the items to minimize the number of tests required to identify all positive items in which all tests are performed in parallel. This problem is well-studied and algorithms exist that match the lower bound with a small gap of $\log d$ asymptoticically. An important generalization of the problem is to consider the case that individual positive item cannot make a test positive, but a combination of them (referred as positive subsets) can do. The problem is referred as the *non-adaptive complex group testing*. Assume there are at most $d$ positive subsets whose sizes are at most $s$, existing algorithms either require $\Omega(\log^s n)$ tests for general $n$ or $O(\binom{s+d}{d} \log n)$ tests for some special values of $n$ . However, the number of items in each test cannot be very small or very large in real situation. The above algorithms cannot be applied because there is no control on the number of items in each test. In this paper, we provide a novel and practical derandomized algorithm to construct the tests with two important properties. (1) Our algorithm requires only $O\left((d+s)^{d+s+1}/(d^d s^s) \log n\right)$ tests for all positive integers $n$ which matches the upper bound on the number of tests when all positive subsets are singletons, i.e. $s = 1$. (2) All tests in our algorithm can have the same number of tested items $k$. Thus, our algorithm can solve the problem with additional constraints on the number of tested items in each test, such as maximum or minimum number of tested items.

*Keywords:* pooling design, non-adaptive complex group testing, knock-out study, combinatorial group testing

## 1. Introduction

In biological studies, there are many situations in which we need to identify a subset of items with a particular property (called positive items) from a large set of items. Instead of testing each item one by one, we can group and test several items together in one experiment. If the outcome is negative, we can conclude that all items in the group do not have that property using only one experiment. By grouping the items carefully, biologists can save a lot of experiments. For example, during World War II, biologists needed to identify people with syphilitic antigen from a large population using the Wasserman-type blood test [8]. Instead of performing the test on each blood sample, they performed tests on grouped blood samples in order to reduce the total number of tests. In DNA library screening [6, 13], biologists need to identify from the DNA library a subset of cloned DNA segments containing a particular substring, called probe. Instead of performing an experiment on each clone-probe pair, biologists group several cloned DNA segments together and perform a single experiment on them. In phenotype knockout studies [17, 21, 26], biologists need to identify genes causing a particular phenotype from a set of genes. Instead of knocking out genes one by one in each experiment, biologists can knockout several genes at the same time and check whether the phenotype still appears in one experiment.

The *group testing problem* [9, 15], which has been studied since World War II on the Wasserman-type blood test mentioned above, is to find the best way of grouping items in each test so as to minimize the total number of tests needed in the worst case. If the tests can be performed sequentially after knowing the results of the previous tests, the problem was solved more than 30 years ago and there exist algorithms [15, 18, 23] for which the number of tests required is close to the optimal (in term of exact number of tests). However, some experiments are time-consuming, e.g. each phenotype knockout experiment requires several months, and we cannot afford the time to perform tests one after another. Instead, it is desirable to perform all tests in parallel without knowing the results of others. In this case, the *non-adaptive group testing problem*, also called *pooling design* [5, 11, 12, 19], is needed. In this paper, we focus on this non-adaptive version.

Given a set of $n$ items with at most $d$ hidden positive items $P$, the result of a test on a subset $S$ of items is positive if $P \cap S \neq \emptyset$, otherwise, the result is

[1]This research was partial supported by HK GRF grant HKU 7117/09E.

negative. The *non-adaptive group testing problem* is to design the minimum number of tests $t$, as a function of $n$ and $d$, for determining all positive items $P$ from the results of the tests assuming that all tests are performed in parallel and designed without any knowledge of other test results.

The non-adaptive version of the problem seems to be more difficult than the adaptive version. Only recently, there were some breakthrough results for solving the problem. Porat and Rothschild [20] solved the problem by constructing an Error Correction Code (ECC). ECC encodes the alphabets in a message into binary strings with the Hamming distance between any pair of strings is at least $d$. Thus up to $d/2$ errors in each string can be detected and corrected. By picking a suitable alphabet size, they can convert the ECC into $O(d^2 \log n)$ tests for the non-adaptive group testing problem which almost matches the lower bound of $O(d^2 \frac{\log n}{\log d})$ [4]. Indyk et al. [16] provided another solution also with $O(d^2 \log n)$ tests based on concatenated code. They first construct a Reed-Solomon code with suitable parameters, then encode it with another independent random binary code and convert it into $O(d^2 \log n)$ tests. By decoding the test results in two levels, they can determine the positive items in $O(polylog(n))$ time which is faster than other algorithms which takes $O(poly(n))$ time.

However, many important biological applications cannot be modeled by the above group testing problem becasue of two reasons. First, because of the sensitivity of the experiments, we may not be able to group many items in a test. Similarly, there are cases for which we cannot group too few items. One example is the phenotype knockout experiment. We cannot knock out many genes and leave too few for the test, otherwise the tested individual cannot survival. Therefore, there may be a minimum (or maximum) requirement on the number of tested items in a test. Second, in many real biological cases, instead of individual items, a combination of items (forming a *positive subset*) is required to make the test positive. That is, the test will show a positive result only if all items in a positive subset are all present in the test. For example, in DNA hybridization [22], the test result is positive with the presence of some pairs of hybridized DNA strands (positive subset of size 2). In two-hybrid screening [27] for detecting protein-protein interaction, the test result is positive if the test sample contains some pairs of interacting proteins (another example of positive subset of size 2). Similarly in three-hybrid screening [1], the test result is positive if the test sample contains some sets of interacting proteins and RNA (positive subset of size 3). Thus,

we need a generalization to model these applications.

Given a set of $n$ items with at most $d$ distinct hidden positive subsets $S_i$ with $|S_i| \leq s$, the result of a test on a subset $S$ of items is positive if there is a positive subset $S_i \subseteq S$, otherwise, the result is negative. The *complex group testing problem* [10, 14] is to find the best way of grouping items in each test, so as to minimize the total number of tests needed in the worst case for finding all the hidden positive subsets $S_i$. In practice, we sometimes require $|S| \geq k$ (or $|S| \leq k$) for some $k$. In the following, we only consider the case for $|S| \geq k$ as the other case is symmetric (Our algorithm can calculate the optimal $k$ for a given range).

The group testing problem is a special case of the *complex* group testing problem with $s = 1$ and no requirement on $|S|$. However, none of the algorithms [5, 11, 12, 19] for solving the non-adaptive group testing problem can be extended to solve the complex version with $s > 1$ even without any restriction on the size of $S$. At first glance, one may replace the $n$ items by the $\binom{n}{s}$ combinations of items and apply the above algorithms to design a set of tests. As the $\binom{n}{s}$ combinations of items are not independent, e.g. we cannot test $\{1, 2\}$ without testing $\{1\}$ or $\{2\}$, this trivial reduction does not work. This complex version of the group testing problem seems to be even more difficult and only limited results exist [2, 14, 24, 25]. None of these solutions can handle the requirement on the size of $S$. And they either require many tests ($\Omega(\log^s n)$ tests) [14], or designed only for specific $n$ values [24, 25, 7], or there is no guarantee on the running time or the number of tests [2]. For example, Gao et al. [14] represent each item by a distinct polynomial $g_i(x)$ of degree $z$ in a finite field $GF(q)$ with $n \leq q^z$ and $sdz \leq q$. They select $sdz$ distinct elements in $GF(q)$ and perform a test on those items with the same value of $g_i(y)$ for each element $y$. By choosing the value of $q$ and $z$ carefully, they can solve the problem with $O(s(d \log_q n)^{s+1})$ tests. Stinson et al. [24, 25] construct a set of tests using perfect hash family. They construct a separating hash family with $n = 7^{2^i}$ elements by recursion on integer $i$. Then they encode this hash family into $O(\binom{s+d}{s} \log n)$ tests. However, their method cannot solve the non-adaptive complex group testing problem (or require much more tests) when $n \neq 7^{2^i}$. Bishop et al. [2] solved the problem with $s = 2$ by assigning each item to a test with probability $p$. By setting a suitable probability $p$ and number of tests $t$, they can find all $S_i$ with some false positive subsets. Thus, another round of experiments are needed to identify the false positive subsets.

In this paper, we introduce a deterministic algorithm based on randomiza-

tion and derandomization to solve the non-adaptive complex group testing problem for all possible number of items $n$ and using no more than $t_0 = O\left(\frac{d+s}{r^s(1-r)^d}\log n\right)$ tests, where $r = \max\{\frac{k}{n-d+1}, \frac{s}{d+s}\}$. When there is no restriction on $k$ i.e. $k = 1$, our algorithm requires $O\left((d+s)^{d+s+1}/(d^d s^s)\log n\right)$ tests which matches the lower bound [4] of $O(d^2 \log n)$ for $s = 1$. When compared with Porat and Rothschild's algorithm [20] and Indyk et al.'s algorithm [16], our algorithm is more flexible because it can handle the cases when $s > 1$ and $k > 1$. Our main contributions can be summarized as follows.

1. Our approach is novel, different from any of the previous work even though the techniques used for this approach are not new. The novelty stems from the following observation. It is known that solving the non-adaptive group testing problem is equivalent to designing a binary $t \times n$ $\bar{d}$-separable matrix [5, 11, 12, 19] with the minimum number of rows. We first extend this concept to a $(\bar{d}, \bar{s})$-separable matrix for the complex version of the problem (see the definition in Section 2), then we show that the probability of a random binary $t \times n$ matrix with $t \geq t_0$ being a $(\bar{d}, \bar{s})$-separable matrix is non-zero, i.e. there always exists such a matrix. We use a greedy approach to fill the matrix row by row and guarantee that every time we fill an entry, there must still exist a solution to fill the rest of the entries to make it $(\bar{d}, \bar{s})$-separable.
2. Our approach can solve more general group testing problem, none of previous approaches can be modified to solve the general problem. In particular, an additional advantage of our solution is that we can guarantee every test has exactly $k'$ items where $k' \geq k$ which can handle the cases when there is a restriction on the size of $S$.
3. Our approach is practical and gives an optimal design in the sense that the number of tests matches the lower bound of the special case.

The paper is organized as follows. In Section 2, we define what a $(\bar{d}, \bar{s})$-separable matrix is and the relationship between such a matrix and the non-adaptive complex group testing problem. Section 3 shows a sufficient condition for a matrix to be $(\bar{d}, \bar{s})$-separable and proves the the existence of $t_0$ tests to solve the non-adaptive complex group testing problem. Then, we will describe a derandomized algorithm which constructs no more than $t_0$ tests in Section 4. Section 5 concludes the paper.

## 2. Preliminaries

**Definition** The Non-adaptive Complex Group Testing (NCGT) Problem: Given $n$ items and $d'$ hidden distinct positive subsets of items, $F = S_1, S_2, \ldots, S_{d'}, d' \leq d$, $|S_i| \leq s$, $S_i \not\subseteq S_j$ for all $i \neq j$. The result of a test on a set of items $T$ is positive if and only if there is at least one positive subset $S_i \subseteq T$. The NCGT problem is to design the minimum number of non-adaptive tests for discovering all the positive subsets in $F$.

When the size of each positive set $s = 1$, the NCGT problem is equivalent to the classical *non-adaptive group testing problem (pooling design)* [19]. When $s = 2$, it is equivalent to the *non-adaptive group testing for disjoint pairs problem* [2]. Any solution with $t$ tests to the NCGT problem can be represented as a $t \times n$ binary matrix $M$ and item $j$ is included in the $i$-th test if $M(i, j) = 1$ (column or item and test or row will be used interchangeably if no confusion arises).

For any family $F = S_1, S_2, \ldots, S_{d'}$, $d' \leq d$ given in the NCGT problem, each $S_i$ corresponds to a subset of at most $s$ columns in $M$ and $F$ corresponds to a collection of at most $d$ subsets of columns. For any $F$, we first take the and-product of the columns corresponding to each $S_i$, then take the or-product of all these and-products. The resulting bit vector is denoted as $R(F)$. Note that since the outcome of a test is positive if and only if all items in a positive subset are included in the test, the outcomes of the $t$ tests are the same as $R(F)$ for any family $F$ of positive subsets. If such a matrix represents a solution to the NCGT problem, for any two families $F_1$, $F_2$, $R(F_1)$ and $R(F_2)$ must be different otherwise it is no way to distinguish whether $F_1$ or $F_2$ is the collection of the positive subsets only based on the outcomes of the tests. This motivates us to define a $(\bar{d}, \bar{s})$-separable matrix as follows.

**Definition** A $(\bar{d}, \bar{s})$-*separable matrix* is a binary matrix, such that for any family $F$ of at most $d$ subsets of columns, each subset $S_i$ has at most $s$ columns, the or-product of the $\leq d$ and-products of $\leq s$ columns corresponding to those subsets in $F$, denoted as $R(F)$, is distinct.

Given a $t \times n$ $(\bar{d}, \bar{s})$-separable matrix $M$ and a property represented by family $F$, $R(F)$) is the test results on the subset of items represented by each row in $M$ with '1' means positive and '0' means negative. Since $R(F)$ is distinct for different $F$, it is easy to see that $M$ is a solution of the NCGT problem

using $t$ tests. Thus, the NCGT problem is equivalent to designing a $t \times n$ $(\bar{d}, \bar{s})$-separable matrix with the minimum number of rows.

## 3. Existence of $(\bar{d}, \bar{s})$-Separable Matrix

In this section, we show that there always exists a $t \times n$ $(\bar{d}, \bar{s})$-separable matrix with $t \leq t_0$ and all rows have *at least* $k$ '1', i.e. each test has at least $k$ tested items.

$$t_0 = \frac{(d+s)\ln n - d\ln(d+s) - s\ln s + d + 2s}{r^s(1-r)^d} = O\left(\frac{(d+s)\ln n}{r^s(1-r)^d}\right)$$

where $r = \max\{\frac{k}{n-d+1}, \frac{s}{d+s}\}$. We first describe a sufficient condition for a matrix with *exactly* $k$ '1' in each row to be a $(\bar{d}, \bar{s})$-separable matrix. (Theorem 3.1). Based on this sufficient condition, we prove that there always exists such a $t_0 \times n$ $(\bar{d}, \bar{s})$-separable matrix with exactly $k$ '1' in each row (Theorem 3.4).

**Theorem 3.1.** *Given a $t \times n$ binary matrix $M$, if $M$ has the property that for any $d+s$ distinct columns, there are $\binom{d+s}{s}$ rows such that the induced $\binom{d+s}{s} \times (d+s)$ matrix contains different set of $s$ '1' entries in each row, then $M$ is a $(\bar{d}, \bar{s})$-separable matrix.*

***Proof*** *Let $A = \{A_i\}$ and $B = \{B_j\}$ be two distinct families of $\leq d$ subsets of $\leq s$ columns in $M$ respectively. Remove those subsets $A_i$ and $B_j$ with $A_i = B_j$. W.L.O.G. assume a subset $A_{min} \in A$ contains the minimum number of columns among the remaining subsets. Since $A_i \not\subseteq A_{min}$ and $S \not\subseteq A_{min}$ for all $A_i \neq A_{min}$ and remaining subsets $S \neq A_{min}$ respectively, $B_j \not\subseteq A_{min}$ for all $B_j \in B$. There always exist at most $s+d$ columns containing all columns in $A_{min}$ and one column from each distinct subset $B_j$ as $|A_{min}| \leq s$ and $|B| \leq d$. Since there are $\binom{d+s}{s}$ rows in $M$ such that the induced $\binom{d+s}{s} \times (d+s)$ matrix contains a row with '1' at these $\leq s$ columns in $A_{min}$ and '0' at the rest $\leq d$ columns. Thus the values of $R(A)$ and $R(B)$ are different (1 and 0 respectively) on that row.* □

In particular when $s = 1$, Theorem 3.1 reduces to the existence of $(d+1) \times (d+1)$ identity matrix for any $d+1$ columns. It is because given two distinct sets of positive items $A$ and $B$, we should always find a row such that the $\leq d$

positive items in $B$ is '0' and a positive item in $A$ is '1'. For example, when $n = 9$, $d = 4$ and $s = 1$, $A = \{3, 5, 7, 9\}$, $B = \{1, 5, 6, 9\}$, the corresponding $d + 1$ columns can be $\{3, 1, 5, 6, 9\}$ and the row must have 1 at positions 3 and 0 at the others such that the test results on this set for properties with positive items $A$ and $B$ are positive and negative respectively.

By considering a $t \times n$ random matrix where each row is assigned with $k$ '1' randomly, we find that the probability that a $t \times n$ random matrix with exactly $k$ '1' in each row satisfies the sufficient condition of Theorem 3.1 is non-zero when $t \geq t_0$ (Theorem 3.4). Thus, there always exists such $t_0 \times n$ $(\bar{d}, \bar{s})$-separable matrix with exactly $k$ '1' in each row; otherwise, the probability should be zero. Similar theorems as Theorem 3.4 are shown in [3, 25]. However, since Bonis proved the theorem by considering a hypergraph while Stinson and Wei proved it by partitioning the matrix into submatrices, these proofs cannot be used to construct the derandomized algorithm.

**Lemma 3.2.** *Given a $t \times n$ binary matrix $M$ with exactly $k$ randomly selected '1' in each row, the probability that $M$ being a $(\bar{d}, \bar{s})$-separable matrix is at least*

$$1 - \left(\frac{en}{d+s}\right)^{d+s} \left(\frac{e(d+s)}{s}\right)^{s} \left(1 - r^s(1-r)^d\right)^t$$

*where $r = \frac{k}{(n-d+1)}$ and $e \approx 2.71$ is the Euler's number*

***Proof*** *When the $k$ '1 are assigned randomly among $n$ columns, the probability that exactly $s$ '1' are assigned in some particular positions in a subset of $d+s$ columns is $\binom{n-(d+s)}{k-s} / \binom{n}{k}$. Thus the probability that a particular combination of $s$ out of a particular subset of $d + s$ columns are not assigned '1' is $1 - \binom{n-(d+s)}{k-s} / \binom{n}{k}$.*

$$\begin{aligned}
& Pr(M \text{ is } (\bar{d}, \bar{s})\text{-separable}) \\
\geq\ & Pr(M \text{ satisfies Theorem 3.1}) \\
\geq\ & 1 - Pr(\text{There are } d+s \text{ columns s.t. any induced } \tbinom{d+s}{s} \times (d+s) \text{ matrix} \\
& \text{does not contain all possible combinations of } s \text{ out of } d+s \text{ columns}) \\
\geq\ & 1 - \binom{n}{d+s}\binom{d+s}{s}\left(1 - \frac{\binom{n-(d+s)}{k-s}}{\binom{n}{k}}\right)^t \\
\geq\ & 1 - \left(\frac{en}{d+s}\right)^{d+s}\left(\frac{e(d+s)}{s}\right)^{s}\left(1 - r^s(1-r)^d\right)^t
\end{aligned}$$

□

**Lemma 3.3.**

$$\frac{-1}{\ln(1-r^s(1-r)^d)} < \frac{1}{r^s(1-r)^d}$$

***Proof***

$$\begin{aligned} & \frac{-1}{\ln(1-r^s(1-r)^d)} \\ = \; & \frac{-1}{-r^s(1-r)^d - \frac{[r^s(1-r)^d]^2}{2} - \frac{[r^s(1-r)^d]^3}{3} - \cdots} \\ < \; & \frac{1}{r^s(1-r)^d} \end{aligned}$$

**Theorem 3.4.** *There always exists a $t_0 \times n$ $(\bar{d}, \bar{s})$-separable matrix $M$ with exactly $k$ '1's in each row where*

$$t_0 = \frac{(d+s)\ln n - d\ln(d+s) - s\ln s + d + 2s}{r^s(1-r)^d} = O\left(\frac{d+s}{r^s(1-r)^d}\log n\right)$$

*where $r = \frac{k}{n-d+1}$.*

***Proof*** *Consider a random binary $t \times n$ matrix $M$ with exactly $k$ randomly selected '1' in each row. By Lemma 3.2*

$$\begin{aligned} & 1 - \left(\frac{en}{d+s}\right)^{d+s}\left(\frac{e(d+s)}{s}\right)^{s}\left(1-r^s(1-r)^d\right)^t > 0 \\ \Leftrightarrow \; & -t\ln(1-r^s(1-r)^d) > (d+s)[\ln n - \ln(d+s) + 1] + s[\ln(d+s) - \ln s + 1] \\ \Leftrightarrow \; & t > \frac{(d+s)\ln n - d\ln(d+s) - s\ln s + d + 2s}{-\ln(1-r^s(1-r)^d)} \end{aligned}$$

*When $t$ satisfies the above inequality, the probability that such a $t \times n$ random binary matrix is a $(\bar{d}, \bar{s})$-separable matrix is larger than 0, i.e. there always exists a $t \times n$ $(\bar{d}, \bar{s})$-separable matrix $M$. The theorem is proved using Lemma 3.3.* □

**Corollary 3.5.** *There always exists a $t_0 \times n$ $(\bar{d}, \bar{s})$-separable matrix $M$ with at least $k$ '1's in each row, where $t_0$ is defined in Theorem 3.4 and $r = \max\{\frac{k}{n-d+1}, \frac{s}{d+s}\}$.*

***Proof*** *By differentiating the equation in Theorem 3.4 with respect to $r$, $r^s(1-r)^d$ has the maximum value and $t_0$ the minimum value when $r = s/(d+s)$. Thus, when $k/(n-d+1) \leq s/(d+s)$, we can increase the value of $k$ to $s(n-d+1)/(d+s)$ and achieve the minimum $t_0 = (d+s)^{d+s}[(d+s)\ln n - d\ln(d+s) - s\ln s + d + 2s]/(d^d s^s)$. Note that the assumption "at least $k$ 1's" is still satisfied.* □

Note that when solving the classical non-adaptive group testing problem with $s = 1$ and $k = 1$, $t_0 = O(d^2 \log n)$ which matches with the lower bound. When solving the non-adaptive group testing for disjoint pairs problem with $s = 2$ and $k = 1$, $t_0 = O(d^3 \log n)$.

## 4. Constructing a $(\bar{d}, \bar{s})$-Separable Matrix

Theorem 3.4 shows that there is a $t_0 \times n$ $(\bar{d}, \bar{s})$-separable matrix with exactly $k$ '1's in each row. In this section, we will introduce a deterministic algorithm for constructing such a $t \times n$ $(\bar{d}, \bar{s})$-separable matrix with $t \leq t_0$ by derandomization.

Recall that the sufficient condition for a matrix $M$ being a $(\bar{d}, \bar{s})$-separable matrix is that for any $d+s$ columns, there are $\binom{d+s}{s}$ rows such that the induced $\binom{d+s}{s} \times (d+s)$ matrix represents all possible combinations of $s$ '1' out of $d+s$ columns (Theorem 3.1). Therefore, if all the $\binom{n}{d+s}$ combinations of columns satisfy this requirement, matrix $M$ is a $(\bar{d}, \bar{s})$-separable matrix. We first show by Lemma 4.1 that for a random matrix with exactly $k$ entry '1' in each row, the expected number of combinations of $d+s$ columns satisfying the requirement is larger than $\binom{n}{d+s} - 1$. Based on Lemma 4.2, we can fill in each entry of the matrix with '0' and '1' to each row one by one such that the expected number of groups of $d+s$ columns (out of $\binom{n}{d+s}$) satisfying the requirement does not decrease. Thus, we can construct a $t_0 \times n$ $(\bar{d}, \bar{s})$-separable matrix in a greedy manner.

### *4.1. The Derandomized Algorithm*

Let $C$ be a subset of $d+s$ columns in a $t \times n$ binary matrix $M$ and $M(C)$ be the $t \times (d+s)$ binary matrix by restricting the columns in $C$.

**Lemma 4.1.** *For some $t \leq t_0$, the expected number of combinations of columns (out of $\binom{n}{d+s}$) of a random $t \times n$ matrix satisfying the requirement in Theorem 1 is larger than $\binom{n}{d+s} - 1$.*

**Proof** For any subset $C$ of $d+s$ columns, the probability that $M(C)$ is a $(\bar{d}, \bar{s})$-separable matrix is at least

$$1 - \binom{d+s}{s}\left(1 - \frac{\binom{n-(d+s)}{k-s}}{\binom{n}{k}}\right)^{t_0}$$

Thus, the expected number of combinations of columns (out of $\binom{n}{d+s}$) satisfying the requirement is at least

$$\begin{aligned} & \binom{n}{d+s}\left[1 - \binom{d+s}{s}\left(1 - \frac{\binom{n-(d+s)}{k-s}}{\binom{n}{k}}\right)^{t_0}\right] \\ > \; & \binom{n}{d+s} - 1 \end{aligned}$$

□

Now, we want to show that we can fill in the matrix in a greedy manner in order to obtain a $(\bar{d}, \bar{s})$-separable matrix with $t \leq t_0$. We order the entries of the matrix from top to bottom and from left to right (i.e., we fill the entry from $M(1,1)$ to $M(t,n)$). Assume that all entries proceeding $M(i,j)$ have been filled. Let $E_0(i,j)$ be the expected number of combinations of columns (out of $\binom{n}{d+s}$) satisfying the requirement in Theorem 1 assuming that we fill the entry $M(i,j)$ with '0'. And $E_1(i,j)$ is defined similarly assuming that we fill the entry $M(i,j)$ with '1'.

For any subset $C$ of $(d+s)$ columns in the matrix $M$ with some entries filled, let $p(M,C)$ be the probability that $M(C)$ contains $\binom{d+s}{s}$ rows such that the induced $\binom{d+s}{s} \times (d+s)$ matrix represents all combinations of $s$ '1' out of $d+s$ positions when each row of $M$ is assigned with exactly $k$ '1' randomly (how to compute $p(M,C)$ will be described in the next subsection). The expected number of subsets $C$ with $M(C)$ satisfying Theorem 3.1 is $\sum_C p(M,C)$. Thus, $E_0(i,j) = \sum_C p(M,C)$ when all previously assigned entries are fixed and $M(i,j)$ is assigned '0', similarly for $E_1(i,j)$.

**Lemma 4.2.** $\max\{E_0(i,j), E_1(i,j)\} \geq \max\{E_0(i',j'), E_1(i',j')\}$ *where* $M(i',j')$ *is the entry just before* $M(i,j)$*, i.e.,* $i' = i$ *and* $j' = j-1$ *if* $j \leq n$*, otherwise* $i' = i-1$*,* $j' = n$ *and* $j = 1$*.*

Construct a $t_0 \times n$ matrix $M$ with all entries marked as '$x$';
**for** $i \leftarrow 1$ **to** $t_0$ **do**
  $q' \leftarrow 0$;
  **for** $j \leftarrow 1$ **to** $n$ **do**
    Calculate $E_0(i,j) = \sum_C p(M,C)$ when $M(i,j) = 0$;
    Calculate $E_1(i,j) = \sum_C p(M,C)$ when $M(i,j) = 1$;
    **if** $E_0(i,j) \geq E_1(i,j)$ *or* $q' \geq k$ **then**
      $M(i,j) \leftarrow 0$;
    **else**
      $M(i,j) \leftarrow 1$;
      $q' \leftarrow q' + 1$;
    **end**
    **if** $\max\{E_0(i,j), E_1(i,j)\} = \binom{n}{d+s}$ **then**
      Assign 0 to all entries marked as '$x$';
      Return the first $i$-th rows of $M$ (an $i \times n$ matrix);
    **end**
  **end**
**end**

**Algorithm 1:** derandomized algorithm for constructing $(\bar{d}, \bar{s})$-separable matrix

**Proof** We first let $j = 2, 3, \ldots, n$. Since $E_0(i, j-1)$ and $E_1(i, j-1)$ are calculated based on the assumption that $M(i,j)$ is assigned '0' and '1', $\max\{E_0(i,j-1), E_1(i,j-1)\} = p_0 E_0(i,j) + (1-p_0) E_1(i,j)$ for some real number $0 \leq p_0 \leq 1$. Thus $\max\{E_0(i,j), E_1(i,j)\} \geq \max\{E_0(i,j-1), E_1(i,j-1)\}$. Similarly, we have $\max\{E_0(i,1), E_1(i,1)\} \geq \max\{E_0(i-1,n), E_1(i-1,n)\}$. □

Based on Lemma 4.2, we can assign values to $M(i,j)$ according to the larger value of $E_0(i,j), E_1(i,j)$. Algorithm 1 shows the details of the construction. Initially, we mark all unassigned entries by '$x$'. Since the value of $\sum_C p(M,C)$ increases monotonically with the assignment of $M(i,j)$ and the initial value of $\sum_C p(M,C)$ with no entry being assigned is larger than $\binom{n}{d+s} - 1$, the correctness of Algorithm 1 is guaranteed. The following theorem follows. Note also that when $\max\{E_0(i,j), E_1(i,j)\} = \binom{n}{d+s}$, it means that we can assign anything to the remaining entries, so we assign '0' to these entries.

**Theorem 4.3.** *Algorithm 1 outputs a $t \times n$ $(\bar{d}, \bar{s})$-separable matrix with $t \leq t_0$.*

*4.2. Computing the probability*

In this subsection, we show how to compute $p(M, C)$. Given a $t \times (d+s)$ binary matrix $M(C)$ with all entries in the first $i-1$ rows and the first $j$ entries of the $i$-th row assigned, the probability $p(M, C)$ that $M(C)$ containing $\binom{d+s}{s}$ rows such that the induced $\binom{d+s}{s} \times (d+s)$ matrix represents all combinations of $s$ '1' out of $d+s$ positions can be calculated by the following arguments.

When the last column of $C$ has been assigned, i.e. we are considering the case when all the entries in the first $i$ rows of $M(C)$ have been assigned, we can identify the set of distinct rows of the $\binom{d+s}{s}$ combinations of $s$ '1' out of $d+s$ positions already existed in the first $i$ rows of $M(C)$. Let $R$ be the set of $r$ out of $\binom{d+s}{s}$ combinations that do not exist in the first $i$ rows of $M(C)$. $p(M, C)$ is equal to the probability $p_{row}(i, r)$ that these $r$ combinations in $R$ appear in the remaining $t-i$ rows of $M(C)$. If all the $\binom{d+s}{s}$ rows have already existed in the first $i$ rows of $M(C)$, then $R = \emptyset$, $r = 0$ and $p(M, C) = 1$. The probability that none of the $t-i$ rows equals to a particular row in $R$ is $(1 - \binom{n-(d+s)}{k-s}/\binom{n}{k})^{t-i}$ and the probability that none of the $t-i$ rows equals to any of the $r$ particular rows in $R$ is $(1 - r\binom{n-(d+s)}{k-s}/\binom{n}{k})^{t-i}$. By inclusion and exclusion principle

$$p_{row}(i, r) = 1 + \sum_{\alpha=1}^{r} (-1)^{\alpha} \binom{r}{\alpha} \left(1 - \alpha \frac{\binom{n-(d+s)}{k-s}}{\binom{n}{k}}\right)^{t-i}$$

When the last column of $C$ has not been assigned yet, we can calculate $p(M, C) = p_{rc}$ with the following parameters
$r$ = number of rows in $R$ that do not exist in the first $i-1$ rows of $M(C)$
$r'$ = number of rows (out of $r$) in $R$ can occur in the $i$-th row by assigning the rest entries properly
$w$ = number of entries in $C$ have not been assigned any value in the $i$-th row
$q$ = number of unassigned entries in $C$ have to be assigned with '1' such that exactly $s$ '1' appear in the $i$-th row of $C$
$q'$ = number of entries in the $i$-th row have been assigned with '1'

$$p_{rc} = \begin{cases} \left(\frac{r'\binom{(n-j)-w}{k-q'-q}}{\binom{n-j}{k-q'}}\right) p_{row}(i, r-1) + \left(1 - \frac{r'\binom{(n-j)-w}{k-q'-q}}{\binom{n-j}{k-q'}}\right) p_{row}(i, r) & q + q' \leq k \\ 0 & q + q' > k \end{cases}$$

Since the values of $0 \leq w, q \leq d+s$, $0 \leq r' \leq r \leq \binom{d+s}{s} \leq (d+s)^s$, $1 \leq i \leq t$, $1 \leq j < n$ and $1 \leq q' \leq k$, there are $O(t(d+s)^s)$ $p_{row}(i,r)$ and $O(nkt(d+s)^{2+2s})$ $p_{rc}$ needed to be precomputed. All possible values of $\binom{n'}{q'}$ and $(1-\alpha\binom{n-(d+s)}{k-s}/\binom{n}{k})^{t-i}$ for different parameters can be precomputed in $O(n^2)$ and $O(t(d+s)^s)$ times. Each $p_{row}(i,r)$ can be calculated in $O((d+s)^s)$ time after the above precomputation. Thus, the $O(t(d+s)^s)$ $p_{row}(i,r)$ elements can be calculated in $O(t(d+k)^{2s})$ times. Since each $p_{rc}$ element can be calculated in constant time after the precomputation, the $O(nkt(d+s)^{2+2s})$ possible $p_{rc}$ elements can be calculated in $O(nkt(d+s)^{2+2s})$ times. The total time complexity for pre-calculating all possible $p(M,C)$ is $O(n^2+nkt(d+s)^{2+2s})$.

## 5. Conclusions

In this paper, we have introduced a deterministic algorithm for constructing tests with the constraint that at most (or at least) $k$ tested items in each test for the non-adaptive complex group testing problem. The algorithm matches with the lower bound $O(d^2 \log n)$ for the unconstrained classical non-adaptive group testing problem. In the future, more complicated constraints, such as inhibition and errors, should be modeled and considered.

## References


[1] Bernstein, D.S., Buter, N., Stumpf, C., Wickens, M.: Analyzing mRNA-Protein Complexes Using a Yeast Three-Hybrid System. Methods 26:123–141 (2002)

[2] Bishop, M.A., Macula, A.J., Renz, T.E., Ufimtsev, V.V.: Hypothesis Group Testing for Disjoint Pairs. Jour. Comb. Optim. 15, 7–16 (2008)

[3] Bonis, A.: New Combinatorial Structures with Applications to Efficient Group Testing with Inhibiters. Jour. Comb. Optim. 15, 77–94 (2008).

[4] Chaudhuri, S., Radhakrishnan, J.: Deterministic Restrictions in Circuit Complexity. ACM Symposium on Theory of Computing, 30–36 (1996)

[5] Deng, P., Hwang, F.K., Wu, E., MacCallum, D., Wang, F., Znati, T.: Improved Construction for Pooling Design. Jour. Comb. Optim. 15, 123–126 (2008)

[6] D'yachkov, A.G., Macula, A.J., Torney, D.C., Vilenkin, P.A.: Two Models of Nonadaptive Group Testing for Designing Screening Experiments. Proc. 6th Int. Workshop on Model-Orented Designs and Analysis, 63–75 (2001)

[7] D'yachkova, A., Vilenkina, P., Torneyb, D., Macula, A.: Families of Finite Sets in which No Intersection of $\ell$ Sets Is Covered by the Union of s Others. Jour. Comb. Thy. Series A, 99(2), 195–218 (2002).

[8] Dorfman, R.: The detection of defective members of large population. The Annals of Mathematical Statistics 14(4), 436–440 (1943)

[9] Du, D.Z., Hwang, F.: Combinatorial Group Testing and Its Applications. 2nd edition, World Scientific, Singapore (2000)

[10] Du, D.Z., Hwang, F.: Pooling Design and Nonadaptive Group Testing: Important Tools for DNA Sequencing. World Scientific, Singapore (2006)

[11] Du, D.Z., Hwang, F.K., Wu, W., Znati, T.: New construction for transversal design. Jour. of Comput. Bio. 13(4), 990–995 (2006)

[12] Eppstein, D., Goodrich, M.T., Hirschberg, D.S.: Improved Combinatorial Group Testing Algorithms for Real-World Problem Sizes. SIAM Jour. on Comput. Arch. 36(5), 1360–1375 (2006)

[13] Farach, M., Kannan, S., Knill, E., Muthukrishnan, S.: Group Testing Problem with Sequences in Experimental Molecular Biology. Proc. Compression and Complexity of Sequences, 357–367 (1997)

[14] Gao, H., Hwang, F.K., Thai, M., Wu, W., Znati, T.: Construction of d(H)-disjunct matrix for group testing in hypergraphs. Jour. of Combin. Optim. 12(3), 297-301 (2006)

[15] Hwang, F.K.: A Method for Detecting All Defective Members in a Population by Group Testing. Jour. Amer. Statist. Assoc. 67, 605–608 (1972)

[16] Indyk, P., Ngo, H.Q., Rudra, A.: Effciently Decodable Non-adaptive Group Testing. SODA(2010)

[17] 15. Jendreyko, N., Popkov, M., Rader, C., Barbas III C.F.: Phenotypic Knockout of VEGF-R2 and Tie-2 with an Intradiabody Reduces Tumor Growth and Angiogenesis in vivo. PNAS, 102(23), 8293–8298 (2005)

[18] Li, C.H.: A Sequential Method for Screening Experimental Variables. Jour. Amer. Statist. Assoc. 57, 455–477 (1962)

[19] Ngo, H.Q., Du, D.Z.: A Survey on Combinatorial Group Testing Algorithms with Applications to DNA Library Screening. Discrete Mathematical Problems with Medical Applications, DIMACS Series, 55, American Mathematical Society, Providence, RI (2000)

[20] Porat, E. and Rothschild, A.: Explicit Non-adaptive Combinatorial Group Testing Schemes. ICALP, 748–759 (2008)

[21] Ruberti, F., Capsoni, S., Comparini, A., Daniel, E.D., Franzot, J., Gonfloni, S., Rossi, G., Berardi, N., Cattaneo, A.: Phenotypic Knockout of Nerve Growth Factor in Adult Transgenic Mice Reveals Severe Deficits in Basal Forebrain Cholinergic Neurons, Cell Death in the Spleen, and Skeletal Muscle Dystrophy. Jour. Neurosci. 20(7), 2589–2601 (2000)

[22] Sibley, C.G., Ahlquist, J.E.: The Phylogeny of the Hominoid Primates, as Indicated by DNA-DNA Hybridization. Jour. Mole. Evolu. 20, 2-15 (1984)

[23] Sobel, M., Groll, P.A.: Group Testing to Elimate Efficiently All Defectives in a Binormal Sample. Bell System Tech. Jour. 38, 1179–1252 (1959)

[24] Stinson, D.R., Trung, T., Wei, R.: Secure Frameproof Codes, Key Distribution Patterns, Group Testing Algorithms and Related Structures. Jour. Stat. Plann. Infer. 86, 595–617 (2000)

[25] Stinson, D.R., Wei, R.: Generalized Cover-Free Families. Discrete Mathematics 279, 463–477 (2004)

[26] Yang, A.G., Bai, X., Huang, X.F., Yao, C., Chen, S.Y.: Phenotypic Knockout of HIV Type 1 Chemokine Coreceptor CCR-5 by Intrakines as Potential Therapeutic Approach for HIV-1 Infection. Proc. Natl. Acad. Sci. 94, 11567–572 (1997)

[27] Young, K.: Yeast Two-Hybrid: So Many Interactions, (in) So Little Time. Biol. Reprod. 58(2), 302–311 (1998)