Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Theoretical Computer Science 306 (2003) 223–243

Theoretical
Computer Science

www.elsevier.com/locate/tcs

Constructions of generalized superimposed codes with applications to group testing and conflict resolution in multiple access channels[☆]

Annalisa De Bonis*, Ugo Vaccaro

Dipartimento di Informatica e Applicazioni, Università di Salerno, 84081 Baronissi (SA), Italy

Received 6 September 2002; received in revised form 31 March 2003; accepted 2 April 2003
Communicated by D-Z. Du

Abstract

In this paper we introduce a parameterized generalization of the well known superimposed codes. We give algorithms for their construction and provide non-existential results. We apply our new combinatorial structures to the efficient solution of new group testing problems and access coordination issues in multiple access channels.

Keywords: Group testing; Multiaccess channels; Superimposed codes; Cover-free families

1. Introduction

Superimposed codes were introduced in the seminal paper by Kautz and Singleton [29]. Since then, they have been extensively studied both in the coding theory community (see [16–19,33,34,44] and the references quoted therein) and the combinatorics community under the name of *cover free* families [21,23,40,41]. Informally, a collection of subsets of a finite set is *r-cover free* if no subset in the collection is included in the union of *r* others. One gets the binary vectors of a superimposed code by considering the characteristic vectors of members of an *r*-cover free family.

[☆] This work was partially supported by Italian Ministry of Education, University and Research under the PRIN project: "RESOURCE ALLOCATION IN WIRELESS NETWORKS (REALWINE)". An extended abstract of this paper was presented at ESA 2002.

* Corresponding author.

E-mail addresses: debonis@dia.unisa.it (A. De Bonis), uv@dia.unisa.it (U. Vaccaro).

0304-3975/03/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0304-3975(03)00281-0

Superimposed codes are a very basic combinatorial structure and find application in an amazing variety of situations, ranging from cryptography and data security [1,10,20,37,31,42] to computational molecular biology [2,13,15,24], from multiaccess communication [15,29] to database theory [29], from pattern matching [27] to distributed coloring [32], circuit complexity [7], broadcasting in radio networks [11], and other areas of computer science.

In this paper we introduce a parameterized generalization of superimposed codes. For particular values of the parameters our codes reduce to the classical Kautz and Singleton superimposed codes. Our codes also include a first generalization of superimposed codes proposed in [19], and the combinatorial structures considered in [28]. For our generalized superimposed codes we provide constructions and non-existential results, both aspects have relevance to the application areas considered in this paper. Our motivations to introduce the generalization comes from new algorithmic issues in *Combinatorial Group Testing* and *Conflict Resolution in Multiple Access Channels*.

1.1. Combinatorial group testing

In group testing, the task is to determine the *positive* members of a set of objects $\mathcal{O}$ by asking subset queries of the form "does the set $Q \subseteq \mathcal{O}$ contain a positive object?". The very first group testing problem arose almost sixty years ago in the area of chemical analysis [14], where it was employed as a blood test technique to detect the infected members of a population. Since then, combinatorial group testing has exhibited strong relationships with several computer science subjects: algorithms, complexity theory, data compression, computational geometry, and computational learning theory, among others. The recent monograph by Du and Hwang [15] gives an excellent account of these aspects. Combinatorial group testing is also experiencing a renaissance in Computational Molecular Biology where it finds ample applications for screening libraries of clones with hybridization probes [4,6] and sequencing by hybridization [35,39]. We refer to [15,22,24] for an account of the fervent development of the subject.

More to our points, recent work [22] suggests that classical group testing procedures should take into account also the possibility of the existence of "inhibitory items", that is, items whose presence in the tested set could render the outcome of the test meaningless, as far as the detection of positive items is concerned. In other words, if during the execution of an algorithm we tested a subset $Q \subseteq \mathcal{O}$ containing positive objects *and* inhibitory items, we would get the same answer as Q did not contain any positive object. Similar issues were considered in [12,25] where additional motivations for the problem were given. In Section 4 we show that our generalized superimposed codes play a crucial role in estimating the computational complexity of this new group testing problem. More precisely, our codes represent both a basic tool in designing efficient algorithmic solutions for the problem at hand, and also in deriving a general lower bound on the number of subset queries to be performed by any algorithm solving it.

1.2. Conflict resolution in multiple access channels

Loosely speaking, with multiple access channels we intend communication media that interconnect a number of users and in which any packet transmission by a single user is broadcasted to all other users connected to the channel. Multiple access channels have been implemented on coaxial cable, fiber optics, packet radio, or satellite transmission media; a well known example is the ETHERNET (see for instance [5] for references in this area).

A model commonly taken as basis for mathematical studies of multiple access channels assumes that the whole system is synchronous, and that at each time instant any number of users can transmit a packet of data over the channel. There is no central control. If just one user transmit its packet in a given time unit, the packet is successfully broadcasted to every other user, if more than one user transmit in a *same* time unit, then all packets are lost because of interference. All users on the channel have the capability of detecting which one of the following events hold: no packet transmission, successful transmission of just one packet, interference due to packet conflict. A key ingredient for an efficient use of multiple access channels is a *conflict resolution algorithm*, that is, a protocol that schedule retransmissions so that each of the conflicting users eventually transmit singly (and therefore successfully) on the channel. A conflict resolution algorithm may be used to coordinate access to the channel in the following way. Access alternates between time instants in which access is unrestricted and time instants in which access is restricted to resolve conflicts. Initially access is unrestricted and all users are allowed to transmit packets at their wish. When a conflict arises, only the involved users execute an algorithm to resolve it and the other users abstain from transmitting. After conflict resolution, access to the channels is again unrestricted (more on this scenario in Section 5).

Conflict resolution in multiple access channels is a source of many challenging algorithmic problems, we refer the reader to the survey paper [9] for a nice account of the vast literature on the topic. The great majority of this body of work assumes the standard hypothesis that conflict arises if more than one user try to transmit at the same time on the channel. However, already in the 1980s Tsybakov et al. [43] studied multiple access channels in which simultaneous transmission of up to $c \geqslant 2$ users is allowed, and conflict arises if strictly more that c users try to transmit at the same time instant. Also, a somewhat similar scenario has been considered in [36], where there are *servers* and *clients*, and each server can successfully fulfill up to c simultaneous client requests; if more than c client requests are submitted to a same server then *none* of them are fulfilled. The problem here is to schedule all client requests so as to satisfy all of them. It is clear that to fully exploit these new capabilities, new conflict resolutions algorithms must be devised.

The contributions of our paper to this issue are presented in Section 5, where it is essentially shown that our generalized superimposed codes are in a sense equivalent to totally non-adaptive conflict resolution protocols for these more powerful multiple access channels, just as like classical superimposed codes corresponds to totally non-adaptive conflict resolution protocols on the standard multiple access channel [29,30]. Informally, with totally non-adaptive conflict resolution protocols we mean the following: The retransmission schedule of each user is fixed (i.e., does not depend on the time in which the conflict occurs and on the set of conflicting users), and known beforehand the conflict event occurs. Therefore, the behavior of each user is fixed and does not need to adapt to the behavior of other users. In contrast, adaptive conflict resolution protocols are more flexible; for instance, they can query other users to find out the identities of the conflicting ones and, on the basis of this acquired knowledge, schedule the retransmissions to solve the conflict. Totally non-adaptive conflict resolution protocols have obvious advantages over adaptive ones, of course at the expenses of possibly longer conflict resolution schedules. Adaptive conflict resolution protocols in our scenario have been given in [8].

1.3. Structure of the paper and summary of results

In Section 2 we introduce the basic concepts and we define our generalized superimposed codes; we also point out their relationships with previously known combinatorial structures. In Section 3 we present upper and lower bounds on the length of generalized superimposed codes; this is equivalent to giving algorithms for constructing generalized superimposed codes with "many codewords" and to proving non-existential results. It is worth pointing out that our upper bounds, that holds in much more generality, also imply the best known upper bound $O(r^2 \log n)$ on the length of classical superimposed codes [18,21,26].

In Section 4 we present the application of our codes to the design of efficient algorithms for group testing in presence of inhibitors. Moreover, we show that our codes play an important role also in bounding from below the complexity of *any* algorithm for group testing in presence of inhibitors. In Section 5 we formally define the multiple access channel under study, we provide an algorithm for conflict resolution and we estimate its performance in terms of the codeword length of our generalized superimposed codes.

2. Basic definitions

A set $\mathscr{C}=\{c_1,\ldots,c_n\}$ of n binary vectors of length N is called a *binary code* of size n and length N. Each c_j is called *codeword* and for any i, $1\leqslant i\leqslant N$, $c_j(i)$ denotes the ith entry of c_j. A binary code $\mathscr{C}$ can be represented by an $N\times n$ binary matrix $\mathscr{M}_{\mathscr{C}}=\|c_j(i)\|$, $i=1,\ldots,N$ and $j=1,\ldots,n$, with codewords as columns. A binary code is said k-uniform if all columns have exactly k entries equal to 1.

For each binary vector c_j of length N, let S_{c_j} denote the subset of $\{1,\ldots,N\}$ defined as $S_{c_j}=\{i\in\{1,\ldots,N\}\colon\ c_j(i)=1\}$. Therefore, to any binary code $\mathscr{C}=\{c_1,\ldots,c_n\}$ of length N we can associate a family $\mathscr{F}=\{S_{c_1},\ldots,S_{c_n}\}$ of subsets of $\{1,\ldots,N\}$. It is clear that this association is invertible, that is, from a family of subsets of $\{1,\ldots,N\}$ one uniquely gets a binary code $\mathscr{C}$ of length N. The set $\{1,\ldots,N\}$ will be called the *ground set* of $\mathscr{F}$.

Given $q>1$ codewords (columns) $c_{\ell_1},\ldots,c_{\ell_q}$, we denote with $(c_{\ell_1}\vee\cdots\vee c_{\ell_q})$ the boolean sum (OR) of $c_{\ell_1},\ldots,c_{\ell_q}$. We say that the column c_h is covered by the column c_j if any 1 entry of c_h corresponds to a 1 entry of c_j.

Definition 1. Let p, r and d be positive integers and let $d\leqslant r$. We call a binary code $\mathscr{C}=\{c_1,\ldots,c_n\}$, with $n\geqslant p+r$, (p,r,d)-*superimposed* if for any distinct $p+r$

codewords $c_{h_1},\dots,c_{h_p},c_{\ell_1},\dots,c_{\ell_r}$ there exist *distinct* $r-d+1$ indices $j_1,\dots,j_{r-d+1}\in\{\ell_1,\dots,\ell_r\}$ such that $(c_{h_1}\vee\cdots\vee c_{h_p})$ is not covered by $(c_{j_1}\vee\cdots\vee c_{j_{r-d+1}})$. The minimal length of a k-uniform (p,r,d)-superimposed code of size n is denoted by $N(p,r,d,k,n)$, whereas that of an arbitrary (p,r,d)-superimposed code of size n is denoted by $N(p,r,d,n)$.

Informally, the family of subsets associated to the binary vectors of a (p,r,d)-superimposed code is such that for any p subsets and any r subsets, there exist $r-d+1$ subsets among the r's such that the union of the p subsets are not included in the union of the $r-d+1$'s. Notice that (p,r,d)-superimposed codes are a generalization of the superimposed codes introduced by Kautz and Singleton [29] which corresponds to our definition for the case $p=d=1$. The families of sets associated to such codes are often referred to with the name of r-cover free families and have been extensively studied in the field of Extremal Set Theory [21,40]. An r-cover free family is such that no member of the family is contained in the union of any other r members of the family.

Dyachkov and Rykov [19] generalized r-cover families by introducing (p,r)-cover free families. A family is said (p,r)-cover free if the union of any p members of the family is not contained in the union of any other r members of the family. A (p,r)-cover free family corresponds exactly to our codes with parameter $d=1$. Finally the combinatorial structure considered in [28] coincides with ours for $p=1$ and $d=r$.

3. Bounds on the length of (*p, r, d*)-superimposed codes

In this section we will present upper and lower bounds on the length $N(p,r,d,n)$ of (p,r,d)-superimposed codes with n codewords.

Theorem 1. *Let* $n\geqslant r\geqslant d\geqslant 1$ *and* $p\geqslant 1$.
If $2r<(d-1)p$ *then it results*

$$N(p,r,d,n)=\mathrm{O}\left((r+p)\log\frac{n}{r+p}\right). \tag{1}$$

If $2r\geqslant(d-1)p$ *then it results*

$$N(p,r,d,n)=\mathrm{O}\left(\frac{r(r+p)}{pd}\log\frac{n}{r+p}\right). \tag{2}$$

Proof. For the simple case $d=1$ the upper bound (2) follows immediately from Theorem 5 of [19]. For that reason we will prove the theorem only for the case when $d\geqslant 2$. The theorem will be proved by using the probabilistic method.

Let $\mathscr{M}_{\mathscr{C}}=\|c_j(i)\|$ be an $N\times n$ a random binary matrix where each entry has probability b of being 1 and probability $1-b$ of being 0. For $h_1,\dots,h_p\in\{1,\dots,n\}$ and $\ell_1,\dots,\ell_r\in\{1,\dots,n\}\backslash\{h_1,\dots,h_p\}$, we say that the columns $c_{h_1},\dots,c_{h_p}$ are *bad* for

$c_{\ell_1},\dots,c_{\ell_r}$, if there is no $i\in\{1,\dots,N\}$ such that the ith entry of $(c_{h_1}\vee\cdots\vee c_{h_p})$ is equal to 1 and $c_{j_1}(i)=c_{j_2}(i)=\cdots=c_{j_q}(i)=0$, for some subset $\{j_1,\dots,j_q\}\subseteq\{\ell_1,\dots,\ell_r\}$ with $q\geqslant r-d+1$. In other words, the columns $c_{h_1},\dots,c_{h_p}$ are bad for $c_{\ell_1},\dots,c_{\ell_r}$ if for any $r-d+1$ pairwise distinct indices $j_1,\dots,j_{r-d+1}\in\{\ell_1,\dots,\ell_r\}$, one has that $(c_{h_1}\vee\cdots\vee c_{h_p})$ is covered by $(c_{j_1}\vee\cdots\vee c_{j_{r-d+1}})$. The binary matrix $\mathscr{M}_{\mathscr{C}}$ represents a (p,r,d)-superimposed code if and only if there do not exist $r+p$ pairwise distinct columns $c_{h_1},\dots,c_{h_p},c_{\ell_1},\dots,c_{\ell_r}$ such that $c_{h_1},\dots,c_{h_p}$ are bad for $c_{\ell_1},\dots,c_{\ell_r}$. In the following we will derive an upper bound on the probability that $\mathscr{M}_{\mathscr{C}}$ contains such $r+p$ columns and will show that for large enough values of N this probability can be made smaller than one. It follows that, for suitable values of N, the probability that the matrix $\mathscr{M}_{\mathscr{C}}$ does not represent a (p,r,d)-superimposed code is smaller than one and consequently there must exist a binary matrix with N rows that does represent a (p,r,d)-superimposed code.

For any $i\in\{1,\dots,N\}$ and any $r+p$ pairwise distinct indices $h_1,\dots,h_p,\ell_1,\dots,\ell_r\in\{1,\dots,n\}$, we define P_i as the probability that the ith entry of $(c_{h_1}\vee\cdots\vee c_{h_p})$ is 1 and that there exists a subset $\{j_1,\dots,j_q\}\subseteq\{\ell_1,\dots,\ell_r\}$, with $q\geqslant r-d+1$, such that $c_{j_1}(i)=c_{j_2}(i)=\cdots=c_{j_q}(i)=0$. The probability P_i is equal to

$$P_i=Pr\{(c_{h_1}\vee\cdots\vee c_{h_p})(i)=1\}$$
$$\cdot Pr\{\text{at most } d-1 \text{ of } c_{\ell_1}(i),\dots,c_{\ell_r}(i) \text{ are equal to } 1\}. \tag{3}$$

We notice that the probability P_i is indeed independent from the value of i. The probability that columns $c_{h_1},\dots,c_{h_p}$ are bad for columns $c_{\ell_1},\dots,c_{\ell_r}$ is

$$\prod_{i=1}^{N}(1-P_i).$$

By the union bound the probability that the matrix $\mathscr{M}_{\mathscr{C}}$ does not represent a (p,r,d)-superimposed code is upper bounded by

$$\binom{n}{r+p}\binom{r+p}{p}\prod_{i=1}^{N}(1-P_i). \tag{4}$$

We want to determine an upper bound on the above expression by deriving a lower bound on P_i.

We first obtain a lower bound on

$$Pr\{\text{at most } d-1 \text{ of } c_{\ell_1}(i),\dots,c_{\ell_r}(i) \text{ are equal to } 1\}. \tag{5}$$

Let X_i denote the random variable that counts how many entries among $c_{\ell_1}(i),\dots,c_{\ell_r}(i)$ are equal to 1. The random variable X_i is the sum of r Bernoulli random variables each having probability b of being 1. Hence, X_i has a binomial distribution with expectation $\mu=br$. One has

$$Pr\{\text{at most } d-1 \text{ of } c_{\ell_1}(i),\dots,c_{\ell_r}(i) \text{ are equal to } 1\}$$
$$=Pr\{X_i\leqslant d-1\}=1-Pr\{X_i>d-1\}. \tag{6}$$

Chernoff bound (see Theorem 4.1 of [38]) says that for any $\delta>0$ the following inequality holds:

$$Pr\{X_i > (1+\delta)\mu\} < \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}. \tag{7}$$

By setting the probability b of 1 to $(d-1)/2r$ and putting $\delta=1$ in expression (7) one gets

$$Pr\{X_i > d-1\} < \left(\frac{e}{4}\right)^{(d-1)/2}. \tag{8}$$

For $d\geqslant 2$ the right hand-side of the above inequality is smaller than or equal to $(e/4)^{1/2}<0.83$. Hence one has that

$$Pr\{\text{at most } d-1 \text{ of } c_{\ell_1}(i),\dots,c_{\ell_r}(i) \text{ are equal to } 1\} > 17/100. \tag{9}$$

In order to derive a lower bound on P_i we need also to derive a lower bound on

$$Pr\{(c_{h_1}\vee\cdots\vee c_{h_p})(i)=1\} = 1-(1-b)^p. \tag{10}$$

Since $b=(d-1)/2r$, one gets that expression (10) is equal to $1-(1-(d-1)/2r)^p$ which we limit from below by considering the following two cases.

Case 1: $2r<(d-1)p$.

Under this hypothesis we also have $p\geqslant 2$. Since it is $(d-1)/2r>1/p$ then one has

$$\left(1-\frac{d-1}{2r}\right)^p < \left(1-\frac{1}{p}\right)^p < \left(1-\frac{1}{p}\right)^{p-1} = \frac{1}{(1+1/(p-1))^{p-1}} \leqslant 1/2,$$

for any $p\geqslant 2$. Consequently one has

$$Pr\{(c_{h_1}\vee\cdots\vee c_{h_p})(i)=1\} = 1-\left(1-\frac{d-1}{2r}\right)^p > 1/2. \tag{11}$$

From inequalities (9) and (11) one has that $P_i>17/200$. It follows that the value of expression (4), representing an upper bound on the probability that $\mathcal{M}_{\mathscr{C}}$ does not constitute a (p,r,d)-superimposed code, is less than

$$\binom{n}{r+p}\binom{r+p}{p}(183/200)^N.$$

In order to make the above value smaller than 1, it is sufficient that

$$N > \frac{\ln\left(\binom{n}{r+p}\binom{r+p}{p}\right)}{-\ln(183/200)}.$$

Therefore,

$$N(p,r,d,n) \leqslant \left\lceil \frac{1}{\ln(200/183)} \ln\left(\binom{n}{r+p}\binom{r+p}{p}\right)\right\rceil + 1. \tag{12}$$

The asymptotic bound (1) now follows from inequality (12) and the well known inequality

$$\binom{a}{b} \leqslant \left(e\,\frac{a}{b}\right)^{b}. \tag{13}$$

Case 2: $2r \geqslant (d-1)p$.
It results

$$\begin{aligned}
&Pr\{(c_{h_1} \vee \cdots \vee c_{h_p})(i) = 1\}\\
&\quad = 1 - \left(1 - \frac{d-1}{2r}\right)^{p}\\
&\quad = \sum_{j=0}^{p} \binom{p}{j} \left(1 - \frac{d-1}{2r}\right)^{p-j} \left(\frac{d-1}{2r}\right)^{j} - \left(1 - \frac{d-1}{2r}\right)^{p}\\
&\quad = \sum_{j=1}^{p} \binom{p}{j} \left(1 - \frac{d-1}{2r}\right)^{p-j} \left(\frac{d-1}{2r}\right)^{j}\\
&\quad \geqslant p \left(1 - \frac{d-1}{2r}\right)^{p-1} \left(\frac{d-1}{2r}\right).
\end{aligned} \tag{14}$$

Since it is $(d-1)/2r \leqslant 1/p$ then one has

$$\left(1 - \frac{d-1}{2r}\right)^{p-1} \geqslant \left(1 - \frac{1}{p}\right)^{p-1} = \frac{1}{(1 + 1/(p-1))^{p-1}} \geqslant \frac{1}{e}. \tag{15}$$

It follows that $1-(1-(d-1)/2r)^p \geqslant p1/e(d-1)/2r$. The previous inequality and (9) imply that $P_i > 17/100\,p1/e(d-1)/2r$. It follows that the value of expression (4) is at most

$$\binom{n}{r+p}\binom{r+p}{p}\left(1 - \frac{17}{200e}\,\frac{p(d-1)}{r}\right)^{N}.$$

In order to make the above expression smaller than 1, it is sufficient that

$$N > \frac{\ln\left(\binom{n}{r+p}\binom{r+p}{p}\right)}{-\ln(1 - 17/200e\,p(d-1)/r)}.$$

Since for $0 \leqslant x < 1$ it is $-\ln(1-x) \geqslant x$, then it follows that the expression on the right hand-side of the above inequality is less than or equal to

$$\left(\frac{200e}{17}\,\frac{r}{p(d-1)}\right)\ln\left(\binom{n}{r+p}\binom{r+p}{p}\right).$$

Therefore,

$$N(p,r,d,n) \leqslant \left\lceil \frac{200e}{17}\,\frac{r}{p(d-1)}\ln\left(\binom{n}{r+p}\binom{r+p}{p}\right)\right\rceil + 1. \tag{16}$$

The asymptotic bound (2) follows from inequalities (13) and (16). □

In the following we will provide a greedy construction for (p,r,d)-superimposed codes which attains a better bound than Theorem 1 for $r = \mathrm{o}(d^2 + d\sqrt{p})$ and for $2r \geqslant p(d-1)$ if $p = \mathrm{o}(d)$.

Let $\mathscr{C} = \{c_1,\dots,c_n\}$ be a binary code of length N. For any c_j, $j = 1,\dots,n$, let k_j denote the number of entries equal to 1 in c_j and let $\underline{k} = \min_{j=1,\dots,n}\{k_j\}$. Further, for any pair of codewords c_h and c_j, let $\alpha_{h,j}$ denote the dot product of c_h and c_j, i.e., the number of entries both c_h and c_j have a 1. We define $\bar{\alpha} = \max_{h,j=1,\dots,n,h\neq j}\{\alpha_{h,j}\}$.

Lemma 1. *Let $\mathscr{C} = \{c_1,\dots,c_n\}$ be a binary code of length N. If $\lfloor r\bar{\alpha}/\underline{k}\rfloor < d$ then $\mathscr{C}$ is a $(1,r,d)$-superimposed code.*

Proof. Let $c_h, c_{\ell_1},\dots,c_{\ell_r} \in \mathscr{C}$ and let $i_1,\dots,i_{k_h}$ denote the indices of the entries c_h has a 1. Each of columns $c_{\ell_1},\dots,c_{\ell_r}$ has at most $\bar{\alpha}$ entries equal to 1 among those with indices in $\{i_1,\dots,i_{k_h}\}$. Consequently, the submatrix formed by restricting $c_{\ell_1},\dots,c_{\ell_r}$ to rows with indices in $\{i_1,\dots,i_{k_h}\}$, will contain at most $r\bar{\alpha}$ entries equal to 1. Hence, there must exist a row of such submatrix with at most $\lfloor r\bar{\alpha}/k_h\rfloor \leqslant \lfloor r\bar{\alpha}/\underline{k}\rfloor < d$ entries equal to 1. It follows that there exist at least $r-d+1$ columns among $c_{\ell_1},\dots,c_{\ell_r}$ whose union does not cover c_h. □

Theorem 2. *Let n, r and d be positive integers and let $d \leqslant r \leqslant n$. It results that*

$$N(1,r,d,n) < 24\left(\frac{r}{d}\right)^2(\log_2 n + 2).$$

Proof. In order to prove the theorem, we will show how to construct a code $\mathscr{C}$ verifying the hypothesis of Lemma 1. To this aim, we will use a construction technique similar to that of Hwang and Sos [26].

Let $\{0,1\}_k^N$ denote the set of all binary columns of length N with k entries equal to 1. The procedure to construct $\mathscr{C}$ works as follows. Initially we set $\mathscr{C}_0 = \emptyset$ and $T_0 = \{0,1\}_k^N$. For $t = 1,2,\dots,$ we select a codeword c_t arbitrarily from T_{t-1} and set $\mathscr{C}_t = \mathscr{C}_{t-1} \cup \{c_t\}$. Then, we define T_t as the set obtained by discarding from T_{t-1} all codewords c_j with $\alpha_{t,j} \geqslant \lceil dk/r\rceil$. The procedure will continue until $T_t = \emptyset$.

Since it is $|\mathscr{C}| = |\mathscr{C}_t| = t$ then the size of the resulting code is given by the number of steps executed by the procedure.

Notice that at each step the number of discarded codewords is at most

$$\sum_{i=\lceil dk/r \rceil}^{k} \binom{k}{i}\binom{N-k}{k-i}.$$

As a consequence,

$$|\mathscr{C}| \geqslant \frac{\binom{N}{k}}{\sum_{i=\lceil dk/r \rceil}^{k} \binom{k}{i}\binom{N-k}{k-i}}.$$

Let $b_i = \binom{k}{i}\binom{N-k}{k-i}$, $i = 1, \ldots, k$. For $i = \lceil dk/rc \rceil, \ldots, k$, one has that

$$\frac{b_i}{b_{i-1}} = \frac{(k-i+1)^2}{i(N-2k+i)} \leqslant \frac{(k-\lceil dk/rc \rceil + 1)^2}{\lceil dk/rc \rceil (N - 2k + \lceil dk/rc \rceil)}. \tag{17}$$

Let us set $N = \lfloor 3c^2kr/d \rfloor$ for some constant integer $c \geqslant 2$. Then it is

$$\frac{(k-\lceil dk/rc \rceil + 1)^2}{\lceil dk/rc \rceil (N - 2k + \lceil dk/rc \rceil)} < \frac{1}{c}$$

and consequently it results

$$\begin{aligned}\sum_{i=\lceil \frac{dk}{r} \rceil}^{k} b_i &< \sum_{i=\lceil \frac{dk}{r} \rceil}^{k} \left(\frac{1}{c}\right)^{i-\lceil dk/rc \rceil} b_{\lceil dk/rc \rceil} \\ &< b_{\lceil dk/rc \rceil} \left(\frac{1}{c}\right)^{-\lceil dk/rc \rceil} \left(\frac{1-(1/c)^{k+1}}{1-1/c} - \frac{1-(1/c)^{\lceil dk/r \rceil}}{1-1/c} \right) \\ &< b_{\lceil dk/rc \rceil} \frac{c^2}{c-1} \left(\frac{1}{c}\right)^{\lceil dk/r \rceil((c-1)/c)}.\end{aligned}$$

It is easy to verify that

$$\binom{N}{k} > b_{\lceil dk/rc \rceil},$$

and consequently,

$$|\mathscr{C}| \geqslant \frac{\binom{N}{k}}{\sum_{i=\lceil dk/r \rceil}^{k} b_i} > \frac{c-1}{c^2} c^{\lceil dk/r \rceil (c-1/c)} \geqslant \frac{c-1}{c^2} c^{(d/r)^2 N/6c^2}.$$

Hence, it results

$$N < 6c^2\left(\frac{r}{d}\right)^2\left(\log_c|\mathscr{C}| + \log_c\left(\frac{c^2}{c-1}\right)\right).$$

By setting $c=2$ in the previous inequality one gets

$$N < 24\left(\frac{r}{d}\right)^2(\log_2|\mathscr{C}|+2),$$

from which the theorem follows. □

Since $N(p,r,d,n) \leqslant N(1,r,d,n)$ then Theorem 2 implies the following corollary.

Corollary 1. *Let n, p, r and d be positive integers and let $d \leqslant r$ and $p+r \leqslant n$. It results that $N(p,r,d,n) < 24(r/d)^2(\log_2 n+2)$.*

It is worth pointing out that in the above results, for $p=d=1$, we recover the best known upper bound $O(r^2\log n)$ on the length of superimposed codes [18,21,26].

3.1. Non-existential results on (p,r,d)-superimposed codes

In the previous section we have given upper bounds on the length of the shortest (p,r,d)-superimposed codes. We now prove that (p,r,d)-superimposed codes cannot be "too short" by providing a lower bound on their optimal length. Our lower bound holds for the more general case when a given codeword may occur more than once in the code. Since we will use this generalization in the next section, we will express our lower bound directly in the case of possible multiple occurrences of codewords. We will refer to such codes with the term of multi-codes.

Definition 2. Let p, r and d be positive integers and let $d \leqslant r$. A binary multi-code $\tilde{\mathscr{C}} = \{c_1,\dots,c_n\}$, with $n \geqslant p+r$, is called (p,r,d)-superimposed if for any distinct $p+r$ indices $h_1,\dots,h_p,\ell_1,\dots,\ell_r$ there exist $r-d+1$ *distinct* indices $j_1,\dots,j_{r-d+1} \in \{\ell_1,\dots,\ell_r\}$ such that $(c_{h_1} \vee \cdots \vee c_{h_p})$ is not covered by $(c_{j_1} \vee \cdots \vee c_{j_{r-d+1}})$. The minimal length of a k-uniform (p,r,d)-superimposed multi-code of size n is denoted by $\tilde{N}(p,r,d,k,n)$ whereas that of an arbitrary (p,r,d)-superimposed multi-code of size n is denoted by $\tilde{N}(p,r,d,n)$.

Observe that a (p,r,d)-superimposed multi-code may contain at most $d+p-1$ occurrences of a given codeword c. Indeed let $c_{\ell_1},\dots,c_{\ell_{d+p}}$ be $d+p$ identical codewords, and let $c_{\ell_{d+p+1}},\dots,c_{\ell_{r+p}}$ be any other codewords. One has that $c_{\ell_1} \cup \cdots \cup c_{\ell_p}$ is covered by the union of any $r-d+1$ codewords belonging to $\{c_{\ell_{p+1}},\dots,c_{\ell_{r+p}}\}$. Hence, one has that the following theorem holds.

Theorem 3. *Let n, p, r and d be positive integers and let $d \leqslant r$ and $n \geqslant p+r$. It results that*

$$\tilde{N}(p,r,d,n) \leqslant N(p,r,d,n) \leqslant \tilde{N}(p,r,d,(d+p-1)n).$$

The following lemma establishes an upper bound on the size of a k-uniform (p,r,d)-superimposed multi-code for a fixed value of the length N.

Lemma 2. *Let $\tilde{\mathscr{C}}$ be a k-uniform (p,r,d)-superimposed multi-code of length N. If $\lfloor r/(pd)\rfloor$ is larger than* 1 *and divides k then*

$$|\tilde{\mathscr{C}}| \leqslant \frac{\binom{N}{k/\lfloor r/(pd)\rfloor}}{\binom{k}{k/\lfloor r/(pd)\rfloor}} rp + p - 1.$$

Proof. Let $\mathscr{F}$ be the family associated with a k-uniform (p,r,d)-superimposed multi-code and let $F \in \mathscr{F}$. If $\lfloor r/(pd)\rfloor$ divides k then Baranyai's theorem [3] implies that there exists $s=\binom{k}{k/\lfloor r/(pd)\rfloor}/\lfloor r/(pd)\rfloor$ ways to partition F into $\lfloor r/(pd)\rfloor$ sets of size $k/\lfloor r/(pd)\rfloor$ so that no subset of F of size $k/\lfloor r/(pd)\rfloor$ belongs to two distinct partitions. Let $\mathscr{P}_1^F,\ldots,\mathscr{P}_s^F$ denote these s partitions of F.

Let $F_1,\ldots,F_p$ be any p sets of $\mathscr{F}$. For $v=1,\ldots,p$, and $j=1,\ldots,s$, let $\mathscr{P}_j^{F_v} = \{A_{1,j}^v,\ldots,A_{\lfloor r/(pd)\rfloor,j}^v\}$. For each $j=1,\ldots,s$, there exists two integers $v\in\{1,\ldots,p\}$ and $i\in\{1,\ldots,\lfloor r/(pd)\rfloor\}$ such that $A_{i,j}^v$ is contained in at most $d-1$ other members of $\mathscr{F}$. Suppose by contradiction that for any $v=1,\ldots,p$ and $i=1,\ldots,\lfloor r/(pd)\rfloor$, there exist d other family members $F_{i,j}^{v,1},\ldots,F_{i,j}^{v,d}$ having $A_{i,j}^v$ as a subset. Let $f=r \bmod (pd)$ and let $\hat{F}_1,\ldots,\hat{F}_f$ denote any f members of $\mathscr{F}$ distinct from $F_{i,j}^{v,1},\ldots,F_{i,j}^{v,d}$, for $v=1,\ldots,p$ and $i=1,\ldots,\lfloor r/(pd)\rfloor$. Let us consider the r family members $F_{1,j}^{1,1},\ldots,F_{1,j}^{1,d},\ldots,F_{\lfloor r/(pd)\rfloor,j}^{1,1},\ldots,F_{\lfloor r/(pd)\rfloor,j}^{1,d},\ldots,F_{1,j}^{p,1},\ldots,F_{1,j}^{p,d},\ldots,F_{\lfloor r/(pd)\rfloor,j}^{p,1},\ldots,F_{\lfloor r/(pd)\rfloor,j}^{p,d},\hat{F}_1,\ldots,\hat{F}_f$. Since it is $r-d=d(p\lfloor r/(pd)\rfloor-1)+f$, then any collection of $r-d+1$ members of $\{F_{1,j}^{1,1},\ldots,F_{1,j}^{1,d},\ldots,F_{\lfloor r/(pd)\rfloor,j}^{1,1},\ldots,F_{\lfloor r/(pd)\rfloor,j}^{1,d},\ldots,F_{1,j}^{p,1},\ldots,F_{1,j}^{p,d},\ldots,F_{\lfloor r/(pd)\rfloor,j}^{p,1},\ldots,F_{\lfloor r/(pd)\rfloor,j}^{p,d},\hat{F}_1,\ldots,\hat{F}_f\}$ has a non-empty intersection with $\{F_{i,j}^{v,1},\ldots,F_{i,j}^{v,d}\}$, for any $v=1,\ldots,p$ and $i=1,\ldots,\lfloor r/(pd)\rfloor$. Consequently, the union of any such $r-d+1$ sets contains $F_1\cup\cdots\cup F_p$, thus contradicting the hypothesis that $\mathscr{F}$ is associated with a (p,r,d)-superimposed multi-code.

It follows that for any p members $F_1,\ldots,F_p$ of $\mathscr{F}$ there are at least s subsets of $\{1,\ldots,N\}$ in $\mathscr{P}_1^{F_1}\cup\cdots\cup\mathscr{P}_s^{F_1}\cup\cdots\cup\mathscr{P}_1^{F_p}\cup\cdots\cup\mathscr{P}_s^{F_p}$ which are contained in at most $d-1$ other family members. Moreover, at least $\lceil s/p\rceil$ such subsets are pairwise distinct, since for any $v=1,\ldots,p$, each subset belongs to at most one of partitions $\mathscr{P}_1^{F_v},\ldots,\mathscr{P}_s^{F_v}$. Let us associate each subfamily of $\mathscr{F}$ of size p with $\lceil s/p\rceil$ such subsets of $\{1,\ldots,N\}$ and let us consider $\lfloor|\mathscr{F}|/p\rfloor$ pairwise disjoint subfamilies of $\mathscr{F}$ of size p. Then, each of these pairwise disjoint subfamilies is associated with $\lceil s/p\rceil$ subsets of $\{1,\ldots,N\}$ of size $k/\lfloor r/(pd)\rfloor$ and each of these subsets is associated with at most $d-1$ of the remaining $\lfloor|\mathscr{F}|/p\rfloor-1$ subfamilies. Since these $\lceil s/p\rceil$ subsets are pairwise distinct, then it results

$$\lfloor|\mathscr{F}|/p\rfloor\cdot\lceil s/p\rceil \leqslant d\binom{N}{\frac{k}{\lfloor r/(pd)\rfloor}}$$

from which the lemma follows. □

A lower bound on the length of non-uniform (p,r,d)-superimposed codes can be derived from Lemma 2 by resorting to the following simple lemma.

Lemma 3. *Let p, r, d and n be positive integers with $d \leqslant r$ and $n \geqslant p+r$. Then, if there exists a (p,r,d)-superimposed multi-code of size n and length N then it is possible to build an N-uniform (p,r,d)-superimposed multi-code of size n and length $2N$.*

Proof. Let $\tilde{\mathscr{C}} = \{c_1,\dots,c_n\}$ be a non-uniform (p,r,d)-superimposed multi-code of size n and length N. For $j=1,\dots,n$, let c'_j denote the column of length $2N$ having $c'_j(i) = c_j(i)$ and $c'_j(i+N) = 1 - c_j(i)$, for $i = 1,\dots,N$. The lemma follows from observing that the multi-code $\tilde{\mathscr{C}}' = \{c'_1,\dots,c'_n\}$ is an N-uniform (p,r,d)-superimposed multi-code of length $2N$. □

Next Theorem 4 is an immediate consequence of Lemmas 2, 3, inequality (13) and of the following inequality holding for any pair of non-negative integers a and b.

$$\left(\frac{a}{b}\right)^b \leqslant \binom{a}{b}.$$

Theorem 4. *Let p, r and d be positive integers and let $d \leqslant r$ and $n \geqslant p+r$. It results that*

$$\tilde{N}(p,r,d,n) \geqslant \frac{1}{1+\log_2 e}\left\lfloor \frac{r}{pd} \right\rfloor \log_2 \frac{n-p+1}{rp},$$

where e is the base of the natural logarithm.

The following corollary follows from Theorems 3 and 4.

Corollary 2. *Let p, r and d be positive integers and let $d \leqslant r$ and $n \geqslant p+r$. It results that*

$$N(p,r,d,n) \geqslant \frac{1}{1+\log_2 e}\left\lfloor \frac{r}{pd} \right\rfloor \log_2 \frac{n-p+1}{rp},$$

where e is the base of the natural logarithm.

4. Efficient algorithms for group testing with inhibitors

The classical group testing scenario consists of a set $\mathscr{S} = \{s_1,\dots,s_n\}$ of items p of which are *defective* (positive), while the others are *good* (negative). The goal of a group testing strategy is to identify all defective items. To this aim, items of $\mathscr{S}$ are pooled together for testing. A test yields a positive feedback if the tested pool contains one or more positive members of $\mathscr{S}$ and a negative feedback otherwise. The group testing strategy is said *non-adaptive* if all tests are performed in parallel whereas it is said *adaptive* when the tests are performed sequentially, and which test to perform at a given step may depend on the feedbacks of the previously executed tests. In non-adaptive strategies each test must be decided beforehand and cannot depend on the feedbacks of previous tests.

We briefly describe how Kautz and Singleton superimposed codes [29] provide a non-adaptive group testing strategy for such a scenario; this will give intuitions for our more complicated case. The correspondence is obtained by associating the columns of the superimposed code with items and the rows with tests. Entry (i, j) of the matrix is 1 if s_j belongs to the pool used for the ith test, and 0 otherwise. Let y denote the column of length N with $y(i) = 1$, $i = 1, \ldots, N$, if and only if the response to the ith test is positive. This column is equal to the boolean sum of the columns associated to the p defective items. The code provides a strategy to uniquely identify the p defectives if the boolean sums of p columns are all distinct. Codes with this property, which is weaker than the cover-free property illustrated in Section 2, have been considered by Kautz and Singleton as well. However, the cover-free property allows a more efficient detection of defective items. Indeed, the feedback column y will cover only the columns associated to the p defective items. For that reason, it will be sufficient to inspect individually the columns associated to the n items instead of inspecting the boolean sums associated to all distinct $\binom{n}{p}$ p-tuples of items.

4.1. Group testing with inhibitors

In this section we consider the variation of classical group testing which has been introduced by Farach et al. [22]. A related model was considered in [12]. In this search model, which we call *Group Testing with Inhibitors* (GTI), the input set consists not only of positive items and negative items, but also of a group of r items called *inhibitors*.

A pool tests positive if and only if it contains one or more positive items and no inhibitor. The problem is to identify the set of the positive items. Farach et al. [22] have proved that this problem has the same asymptotic lower bound of the apparently harder problem of identifying both the set of the positives and that of the inhibitors. They have also described a *randomized* algorithm to find the p positives which achieve the information theoretic bound when $p + r \ll n$. In [13] the authors improved on the results given in [22].

4.2. The threshold model

We introduce a generalization of the GTI model presented in the previous section. In this new model the presence of positives in a test set can be detected only if the test set contains a number of inhibitors smaller than a fixed threshold d. Our goal is to identify all positive items using as few tests as possible.

4.2.1. Our algorithm

Our algorithm consists of four phases.

- *Phase* 1. Find a group of items Q which tests positive.
- *Phase* 2. Find a group of items containing exactly $d-1$ inhibitors and at least one positive item.
- *Phase* 3. Find $r-d+1$ inhibitors and discard them.
- *Phase* 4. Find all positives.

In the following we will describe how to perform each of the above phases.

Phase 1: The search strategy performed during this phase is provided by a (p,r,d)-superimposed code of size n. We associate the columns of the code with the n items and the rows with the tests. Entry (i,j) of the matrix is 1 if s_j belongs to the pool used for the ith test, and 0 otherwise. Let y denote the feedback column, i.e., $y(i)=1$, $i=1,\dots,N$, if and only if the response to the ith test is positive. Hence, one has that $y(i)=1$, $i=1,\dots,N$, if and only if among the items pooled for the ith test there are at least one positive and no more than $d-1$ inhibitors. Let $c_{h_1},\dots,c_{h_p}$ be the p codewords associated to the p defective items and let $c_{\ell_1},\dots,c_{\ell_r}$ those associated with the r inhibitors. Then, for any $i=1,\dots,N$, one has that $y(i)=1$ if and only if $(c_{h_1}\vee\cdots\vee c_{h_p})(i)=1$ and there exist $r-d+1$ indices $j_1,\dots,j_{r-d+1}\in\{\ell_1,\dots,\ell_r\}$ such that $(c_{j_1}\vee\cdots\vee c_{j_{r-d+1}})(i)=0$. Since the code is (p,r,d)-superimposed then one has that this condition is verified for at least one index $i\in\{1,\dots,N\}$. Moreover, such an index i exists for any choice of $c_{h_1},\dots,c_{h_p}$ and any choice of $c_{\ell_1},\dots,c_{\ell_r}$. Consequently, the strategy associated with the (p,r,d)-superimposed code guarantees a positive feedback for any choice of the p defectives and for any choice of the r inhibitors.

Notice that the search strategy performed during this phase is completely non-adaptive, a feature of some interest in practical applications.

Phase 2: In the following we will denote with $\textsc{Half}_{\text{L}}$($\textsc{Half}_{\text{R}}$, resp.) a function which takes in input a set $A=\{a_1,\dots,a_m\}$ and returns the set consisting of the first $\lfloor m/2\rfloor$ (the last $\lceil m/2\rceil$, resp.) elements of A. Let Q be the group of items returned by Phase 1. Then, Phase 2 consists of the following procedure:

```
$A \leftarrow \mathscr{S} \setminus Q$
$B \leftarrow Q$
while($|A| > 1$)
 $T \leftarrow B \cup \textsc{Half}_{\text{L}}(A)$
 if $T$ tests positive then $B \leftarrow T$
                              $A \leftarrow \textsc{Half}_{\text{R}}(A)$
                         else $A \leftarrow \textsc{Half}_{\text{L}}(A)$
return($B$)
```

The above procedure preserves the invariant that B contains at most $d-1$ inhibitors and $B\cup A$ contains at least d inhibitors. Since the algorithm terminates as soon as $|A|$ becomes equal to 1, then it follows that the set B returned by the procedure contains exactly $d-1$ inhibitors. Moreover, since $Q\subseteq B$, then B contains at least one positive item.

Phase 3: A variant of the classical group testing is used to find the $r-d+1$ inhibitors $s_{\ell_1},\dots,s_{\ell_{r-d+1}}$ contained in $S\backslash B$. The variant consists in adding the items in the set B returned by Phase 2 to each tested group T. This assures a positive feedback if the tested group T contains no inhibitor, and a negative response if the tested group T contains at least 1 inhibitor.

Then, the $r-d+1$ inhibitors $s_{\ell_1},\dots,s_{\ell_{r-d+1}}$ found in $S\backslash B$ are discarded from S.

Phase 4: $S\backslash\{s_{\ell_1},\dots,s_{\ell_{r-d+1}}\}$ is searched to find the p positives. Since the undiscarded $d-1$ inhibitors do not interfere with the tests, then a classical group testing algorithm can be applied to find all positives.

Observe that the number of tests executed in Phase 1 is as small as $N(p,r,d,n)$. Phase 2 requires $\lceil\log|\mathscr{S}\backslash Q|\rceil\leqslant\lceil\log(n-1)\rceil$ tests since the search space reduces by one half at each step. Phase 3, as well as Phase 4, perform a standard group testing strategy. The cost of the optimal group testing strategy (see Chapter 2 of [15]) to find q defectives in a set of size n is $\mathrm{O}(q\log(n/q))$. Consequently, Phase 3 requires $\mathrm{O}((r-d+1)\log(n/(r-d+1)))$ tests, whereas Phase 4 requires $\mathrm{O}(p\log(n/p))$ tests. Combining all the above estimates one gets the following theorem.

Theorem 5. *There exists a strategy to find the* p *positives which uses*

$$N(p,r,d,n)+\mathrm{O}\left(\log n+(r-d+1)\log\left(\frac{n}{r-d+1}\right)+p\,\log\,\frac{n}{p}\right)$$

tests.

Plugging in the above upper bound either the expression for $N(p,r,d,n)$ given in Theorem 1 or that of Theorem 2 provides an explicit estimate of the cost of our strategy. We remark that for many values of the involved parameters, the leading term in the estimation of the number of tests required by our strategy is just $N(p,r,d,n)$.

4.3. A lower bound on the number of tests

The introduced generalization of superimposed codes intervenes in our group testing problem not only in establishing an upper bound on its optimal cost, but also in determining a lower bound on it. Namely, we have the following results.

Theorem 6. *Any strategy to find all positives requires at least* $N(p,r,d,n-1)$ *tests.*

Proof. Fix any algorithm which finds all p positives and let $T_1,\dots,T_t,\dots$ be a sequence of tested pools. For every item $a\in\mathscr{S}$ and $t\geqslant1$, let $I_t(a)$ be a binary column of length t with $I_t(a)(i)=1$ if $a\in T_i$ and $I_t(a)(i)=0$ otherwise, for $i=1,\dots,t$.

Let $\tilde{\mathscr{C}}_t=\{I_t(a)\colon\ a\in\mathscr{S}\}$. $\tilde{\mathscr{C}}_t$ is a binary multi-code of length t. If $\tilde{\mathscr{C}}_t$ is not a (p,r,d)-superimposed multi-code, then there are $r+p$ items $s_{h_1},\dots,s_{h_p},s_{\ell_1},\dots,s_{\ell_r}$ such that $(I_t(s_{h_1})\vee\cdots\vee I_t(s_{h_p}))$ is covered by $(I_t(s_{j_1})\vee\cdots\vee I_t(s_{j_{r-d+1}}))$ for any $r-d+1$ indices $j_1,\dots,j_{r-d+1}\in\{\ell_1,\dots,\ell_r\}$. This means that if $(I_t(s_{h_1})\vee\cdots\vee I_t(s_{h_p}))(i)=1$ for some $i\in\{1,\dots,t\}$, then there are at least d columns among $I_t(s_{\ell_1}),\dots,I_t(s_{\ell_r})$ having the ith entry equal to 1. As a consequence, one has that for $i=1,\dots,t$, $\{s_{h_1},\dots,s_{h_p}\}\cap T_i\neq\emptyset$ only if $|\{s_{\ell_1},\dots,s_{\ell_r}\}\cap T_i|\geqslant d$. Then, an adversary could make $s_{h_1},\dots,s_{h_p}$ be the p positives and $s_{\ell_1},\dots,s_{\ell_r}$ be the r inhibitors, and force the tests on $T_1,\dots,T_t$ to receive negative feedbacks.

If $t<N(p,r,d,n-1)$, then $\tilde{\mathscr{C}}_t$ is not a (p,r,d)-superimposed multi-code. Then, there are $r+p$ items $s_{h_1},\dots,s_{h_p},s_{\ell_1},\dots,s_{\ell_r}$ such that $(I_t(s_{h_1})\vee\cdots\vee I_t(s_{h_p}))$ is covered by $(I_t(s_{j_1})\vee\cdots\vee I_t(s_{j_{r-d+1}}))$ for any $r-d+1$ indices $j_1,\dots,j_{r-d+1}\in\{\ell_1,\dots,\ell_r\}$. Moreover, any $\tilde{\mathscr{C}}\setminus\{s_{h_i}\}$, for $i=1,\dots,p$, is also not a (p,r,d)-superimposed multi-code. Therefore, there exists also an $(r+p)$-tuple $s_{h'_1},\dots,s_{h'_p},s_{\ell'_1},\dots,s_{\ell'_r}$ with $s_{h_i}\notin\{s_{h'_1},\dots,s_{h'_p}\}$, such that $(I_t(s_{h'_1})\vee\cdots\vee I_t(s_{h'_p}))$ is covered by $(I_t(s_{j'_1})\vee\cdots\vee I_t(s_{j'_{r-d+1}}))$ for any $r-d+1$ indices $j'_1,\dots,j'_{r-d+1}\in\{\ell'_1,\dots,\ell'_r\}$.

From the above discussion, it follows that for $t<N(p,r,d,n-1)$, there exist at least two distinct p-tuples of items which could force the first t tests to receive negative responses. Obviously such a sequence of t tests would not allow us to determine which one is the p-tuple of the positive items. □

Notice that the basic information theoretic lower bound implies that $\Omega(p\log(\frac{n}{p}))$ tests are required to find all p positives. The following result is an immediate consequence of the information theoretic bound and of Theorems 4 and 6.

Theorem 7. *Any strategy to find all positives requires* $\Omega(p\log n/p+r/pd\log(n-p+1)/rp)$ *tests.*

5. Conflict resolution in multiple access channels

In this section we show how our codes can be used for resolving conflicts in multiple access communication when simultaneous transmissions of up to d users on the same channel is allowed. We first define the mathematical model formally.

5.1. The multiaccess model

The contemplated scenario consists of a system comprising a set of n users $u_1,\dots,u_n$ and a single channel which allows up to d users to successfully transmit at the same time. We make the following standard assumptions.

- *Slotted system.* We assume that the time be divided into time slots and that the transmission of a single packet require one time slot. Simultaneous transmissions are those occurring in the same time slot.
- *Threshold conflict.* If no more than d users transmit during the same time slot then their transmissions are *successful*. Collisions arise if more than d users attempt to transmit at the same time.
- *Immediate feedback.* We assume that at the end of each slot, the system provides each user with a feedback which says whether packets have been transmitted during that slot and whether a conflict has occurred.

- *Retransmission of conflicts.* We assume that packets involved in the conflict must be retransmitted until they are successfully received. Users involved in the conflict are said *backlogged*.
- *Bounded number of backlogged users.* We assume that the number of backlogged users does not exceed a given bound q.
- *Blocked access.* We assume that when a conflict occurs only users involved in the conflict are allowed to transmit until the conflict is resolved. Collision resolution algorithms using this assumption are called *blocked access algorithms* and the time employed to resolve the conflict is called *conflict resolution period.* We define the *length* of the conflict resolution period as the number of time slots the conflict resolution period is divided into.

See [9,5] for an extensive discussion on the implications of the above assumptions.

5.2. Complexity of non-adaptive conflict resolution

A conflict resolution algorithm schedules users' transmissions so that for each user there is a time slot during which her transmission is successful. This property guarantees that a conflict is resolved within the conflict resolution period.

We present a conflict resolution algorithm for the multiaccess model described in the previous section. In our conflict resolution algorithm, each user u_j is permanently associated with a set of time slots. When a new conflict occurs a conflict resolution period starts and the conflict is resolved by having each backlogged user transmit only during the time slots allocated to her. Algorithms like ours are *totally non-adaptive*, in contrast to *adaptive* conflict resolution protocols. The latter may query other users to find out the identities of the conflicting ones and, on the basis of this acquired knowledge, schedule the retransmissions to solve the conflict. Totally non-adaptive conflict resolution protocols have obvious advantages over adaptive ones.

Our algorithm works as follows. Let $u_{\ell_1},\dots,u_{\ell_s}$, $d<s\leqslant q$, denote the users involved in the conflict. Then, transmissions from users other than $u_{\ell_1},\dots,u_{\ell_s}$ are blocked, whereas, for each $h\in\{1,\dots,s\}$, user u_{ℓ_h} transmits only during the time slots which have been allocated to her. The conflict resolution algorithm should guarantee that for each $h\in\{1,\dots,s\}$, there is a time slot among those associated to user u_{ℓ_h} during which at most $d-1$ users in $\{u_{\ell_1},\dots,u_{\ell_s}\}\backslash\{u_{\ell_h}\}$ are allowed to transmit. We use a $(1,q-1,d)$-superimposed code of size n to construct the time slot subsets to be associated to the n users. To this aim, we associate each user with a distinct codeword of the $(1,q-1,d)$-superimposed code. The time slots assigned to a given user are those corresponding to the 1-entries in the associated codeword. The length N of the $(1,q-1,d)$-superimposed code coincides with the number of time slots in the conflict resolution period. For any integer $i\in\{1,\dots,N\}$, a backlogged user transmits during the ith time slot of the conflict resolution period if and only if the ith entry of her codeword is equal to 1.

It is rather easy to see that, if no more than q users are involved in the conflict, then the above algorithm resolves the conflict within N time slots. Let $\{u_{\ell_1},\dots,u_{\ell_s}\}$ be any s users involved in the conflict, $s\leqslant q$. By definition of $(1,q-1,d)$-superimposed code, one has that for each $h \in \{1, \dots, s\}$, there exists an integer $i \in \{1, \dots, N\}$ such that the codeword associated to u_{l_h} has the ith entry equal to 1, whereas at most $d-1$ of the $s-1$ codewords associated to users $\{u_{\ell_1}, \dots, u_{\ell_s}\} \setminus \{u_{\ell_h}\}$ have the ith entry equal to 1. Consequently, for each $h \in \{1, \dots, s\}$, there is a time slot $i \in \{1, \dots, N\}$ among those assigned to user u_{ℓ_h} which has been assigned to at most other $d-1$ backlogged users. As a consequence, there is a time slot within the conflict resolution period during which u_{ℓ_h}'s transmission is successful.

Therefore, a generalized superimposed code can be used as a tool for a totally non-adaptive conflict resolution algorithm in the multiple access channel previously described. Actually, we can prove that in our scenario any totally non-adaptive conflict resolution algorithm corresponds to a generalized superimposed (multi)code. This will allow us to estimate also from below the complexity of non-adaptive conflict resolution algorithms.

Theorem 8. *In any non-adaptive conflict resolution algorithm, the length of the conflict resolution period coincides with the length of a* $(1, q-1, d)$ *superimposed multi-code of size* n.

Proof. Fix any non-adaptive conflict resolution algorithm and suppose that this algorithm divide the conflict resolution period into t slots. The conflict resolution algorithm allows each user u_j to transmit only at given time slots within the conflict resolution period. For each user u_j, $j = 1, \dots, n$, let us define a binary column c_{u_j} of length t such that $c_{u_j}(i) = 1$, $i = 1, \dots, t$, if and only if u_j is allowed to transmit at time slot i. It is rather easy to see that $c_{u_1}, \dots, c_{u_n}$ form a $(1, q-1, d)$ multi-code of size n. Suppose by contradiction that $\tilde{\mathscr{C}}$ is not $(1, q-1, d)$-superimposed. Then, there exist q columns $c_{u_h}, c_{u_{\ell_1}}, \dots, c_{u_{\ell_{q-1}}}$ such that c_h is covered by $(c_{j_1} \vee \cdots \vee c_{j_{q-d}})$ for any $q-d$ indices $j_1, \dots, j_{q-d} \in \{\ell_1, \dots, \ell_{q-1}\}$. This means that if $c_h(i) = 1$ for some $i \in \{1, \dots, t\}$, then there are at least d columns among $c_{\ell_1}, \dots, c_{\ell_{q-1}}$ having the ith entry equal to 1. As a consequence, one has that there is no time slot in the conflict resolution period during which the transmission of c_{u_j} is successful. Then, an adversary could make $c_h, c_{\ell_1}, \dots, c_{\ell_{q-1}}$ be the backlogged users thus forcing c_h's transmissions to be all unsuccessful during the conflict resolution period. □

From the above theorem one has that the length $\tilde{N}(1, q-1, d, n)$ of the shortest $(1, q-1, d)$-superimposed multi-code represents exactly the minimum number of time slots required to resolve a conflict non-adaptively. Therefore, our explicit upper and lower bounds on $\tilde{N}(1, q-1, d, n)$ (recall that $\tilde{N}(\cdot) \leqslant N(\cdot)$) given in Section 3 yield explicit estimates on the goodness of the conflict resolution protocol presented in this section.

Acknowledgements

The authors wish to thank the anonymous referees for very useful comments and suggestions.

References

[1] G. Ateniese, C. Blundo, A. De Santis, D.R. Stinson, Visual cryptography for general access structures, Inform. and Comput. 129 (2) (1996) 86–106.

[2] D.J. Balding, W.J. Bruno, E. Knill, D.C. Torney, A comparative survey of non-adaptive pooling design, in: T.P. Speed, M.S. Waterman (Eds.), Genetic Mapping and DNA Sequencing, IMA Volumes in Mathematics and its Applications, Springer, Berlin, 1996, pp. 133–154.

[3] Zs. Baranyai, On the factorization of the complete uniform hypergraph, in: Infinite and Finite Sets, Proc. Coll. Math. Soc. Jànos Bòlyai, Vol. 10, 1975, pp. 91–108.

[4] E. Barillot, B. Lacroix, D. Cohen, Theoretical analysis of library screening using an n-dimensional pooling strategy, Nucleic Acids Res. (1991) 6241–6247.

[5] D. Bertsekas, R. Gallager, Data Networks, Prentice-Hall, Englewood Cliffs, NJ, 1992.

[6] W.J. Bruno, D.J. Balding, E. Knill, D. Bruce, C. Whittaker, N. Dogget, R. Stalling, D.C. Torney, Design of efficient pooling experiments, Genomics 26 (1995) 21–30.

[7] S. Chaudhuri, J. Radhakrishnan, Deterministic restrictions in circuit complexity, in: Proc. of the 28th Annu. ACM Symp. on the Theory of Computing (STOC 96), Philadelphia, PA, 1996, pp. 30–36.

[8] R.W. Chen, F.K. Hwang, K-definite group testing and its application to polling in computer networks, Congr. Numer. 47 (1985) 145–149.

[9] B.S. Chlebus, Randomized communication in radio networks, in: P.M. Pardalos, S. Rajasekaran, J. Reif, J.D.P. Rolim (Eds.), Handbook of Randomized Computing, Vol. I, Kluwer Academic Publishers, Dordrecht, 2001, pp. 401–456.

[10] B. Chor, A. Fiat, M. Naor, Tracing traitors, in: Proc. of Crypto 94, Santa Barbara, CA, Lecture Notes in Computer Science, Vol. 839, Springer, Berlin, 1994, pp. 257–270.

[11] A.E.F. Clementi, A. Monti, R. Silvestri, Selective families, superimposed codes, and broadcasting on unknown radio networks, in: Proc of Symp. on Discrete Algorithms (SODA'01), Washington, DC, pp. 709–718.

[12] P. Damaschke, Randomized group testing for mutually obscuring defectives, Inform. Process. Lett. 67 (3) (1998) 131–135.

[13] A. De Bonis, U. Vaccaro, Improved algorithms for group testing with inhibitors, Inform. Process. Lett. 67 (1998) 57–64.

[14] R. Dorfman, The detection of defective members of large populations, Ann. Math. Statist. 14 (1943) 436–440.

[15] D.Z. Du, F.K. Hwang, Combinatorial Group Testing and its Applications, 2nd Edition, World Scientific, Singapore, 2000.

[16] A.G. Dyachkov, A.J. Macula, V.V. Rykov, New constructions of superimposed codes, IEEE Trans. Inform. Theory 46 (1) (2000) 284–290.

[17] A.G. Dyachkov, A.J. Macula, V.V. Rykov, New applications and results of superimposed code theory arising from the potentialities of molecular biology in: Number, Information and Complexity, Kluwer, Dordrecht, 2000, pp. 265–282.

[18] A.G. Dyachkov, V.V. Rykov, Bounds on the length of disjunctive codes, Probl. Control Inform. Theory 11 (1982) 7–13.

[19] A.G. Dyachkov, V.V. Rykov, A survey of superimposed code theory, Probl. Control Inform. Theory 12 (4) (1983) 1–13.

[20] M. Dyer, T. Fenner, A. Frieze, A. Thomason, On key storage in secure networks, J. Cryptology 8 (1995) 189–200.

[21] P. Erdös, P. Frankl, Z. Füredi, Families of finite sets in which no set is covered by the union of r others, Israel J. Math. 51 (1985) 75–89.

[22] M. Farach, S. Kannan, E.H. Knill, S. Muthukrishnan, Group testing with sequences in experimental molecular biology, in: B. Carpentieri, A. De Santis, U. Vaccaro, J. Storer (Eds.), Proc. Compression and Complexity of Sequences 1997, IEEE Computer Society, Silver Spring, MD, 1997, pp. 357–367.

[23] Z. Füredi, On r-cover-free families, J. Combin. Theory Ser. A 73 (1996) 172–173.

[24] Hung Q. Ngo, Ding-Zhu Du, A survey on combinatorial group testing algorithms with applications to DNA library screening, in: Discrete Mathematical Problems with Medical Applications, DIMACS Ser.

Discrete Math. Theoret. Comput. Sci., Vol. 55, American Mathematical Society, Providence, RI, 2000, pp. 171–182.

[25] F.K. Hwang, A tale of two coins, Amer. Math. Monthly 94 (1987) 121–129.

[26] F.K. Hwang, V.T. Sös, Non adaptive hypergeometric group testing, Studia Sc. Math. Hungarica 22 (1987) 257–263.

[27] P. Indyk, Deterministic superimposed coding with application to pattern matching, Proc. Foundations of Computer Science (FOCS 97), IEEE Press, Miami, FL, 1997, pp. 127–136.

[28] G.O.H. Katona, T.G. Tarján, Extremal problems with excluded subgraphs in the n-cube, Graph Theory, Lagow, Poland, Lecture Notes in Mathematics, Vol. 1018, Springer, Berlin, 1983, pp. 84–93.

[29] W.H. Kautz, R.R. Singleton, Nonrandom binary superimposed codes, IEEE Trans. Inform. Theory 10 (1964) 363–377.

[30] J. Komlós, A.G. Greenberg, An asymptotically fast non-adaptive algorithm for conflict resolution in multiple-access channels, IEEE Trans. Inform. Theory 31 (2) (1985) 302–306.

[31] R. Kumar, S. Rajagopalan, A. Sahai, Coding constructions for blacklisting problems without computational assumptions, in: Proc. of CRYPTO '99, Santa Barbara, CA, Lecture Notes in Computer Science, Vol. 1666, Springer, Berlin, 1999, pp. 609–623.

[32] N. Linial, Locality in distributed graph algorithms, SIAM J. Comput. 21 (1992) 193–201.

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

[34] A.J. Macula, Error-correcting nonadaptive group testing with d^e-disjunct matrices, Discrete Appl. Math. 80 (1997) 217–222.

[35] D. Margaritis, S. Skiena, Reconstructing strings from substrings in rounds, Proc. of Foundations of Computer Science (FOCS 95), Milwaukee, WI, pp. 613–620.

[36] F. Meyer auf der Heide, C. Scheideler, V. Stemann, Exploiting storage redundancy to speed up randomized shared memory simulations, Proc. of the 12th Internat. Symp. on Theoretical Aspects of Computer Science (STACS 95), pp. 267–278.

[37] C.J. Mitchel, F.C. Piper, Key storage in secure networks, Discrete Appl. Math. 21 (1988) 215–228.

[38] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, Cambridge, 1995.

[39] P.A. Pevzner, R. Lipshutz, Towards DNA sequencing chips, in: Proc. of the 19th Conference on Mathematical Foundations of Computer Science, Kosice, Slovakia, Lectures Notes in Computer Science, Vol. 841, Springer, Berlin, 1994, pp. 143–158.

[40] M. Ruszinkó, On the upper bound of the size of the r-cover-free families, J. Combin. Theory Ser. A 66 (1994) 302–310.

[41] Yu.L. Sagalovich, Separating systems, Probl. Inform. Transmission 30 (2) (1994) 105–123.

[42] D.R. Stinson, Tran van Trung, R. Wei, Secure frameproof codes, key distribution patterns, group testing algorithms and related structures, J. Statist. Plann. Inference 86 (2000) 595–617.

[43] B.S. Tsybakov, V.A. Mikhailov, N.B. Likhanov, Bounds for packet transmissions rate in a random-multiple-access system, Probl. Inform. Transmission 19 (1983) 61–81.

[44] Hong-Gwa Yeh, d-Disjunct matrices: bounds and Lovász Local Lemma, Discrete Math. 253 (2002) 97–107.