Born Again Group Testing: Multiaccess Communications

Communications
Communications

Abstract—A brief summary of the basic notions of group testing is presented together with a brief historical account. One of the early papers on group testing is shown to include a description of the tree-search polling algorithm of Hayes. The classical group testing problem is formulated, including a criterion for optimality of test plans. A restricted class of tests, called nested testing, is described, and a complete description for an optimal nested strategy is given for both a finite number and an infinite number of Bernoulli distributed random variables. A generalization of group testing applicable to the random access communications problem is presented.

I. Introduction

THE USUAL model for multiaccess communications incorporate a community of users, some small fraction of which have information to be transmitted at any given time. We call the set of users with information to be transmitted the *active set*, and a user in this set will be called an *active user*. Two types of protocols for multiaccess communications have been studied. In the first type, called a *reservation protocol*, some subset of the active users is first identified, and only then are these users allowed to transmit their information. In the second type, called a *direct transmission protocol*, the information itself is used to schedule the transmission, and the protocol effectively partitions the community of users into subsets of no more than one active user.

If the total number of users is finite, one straightforward but inefficient method of scheduling the transmissions is to allow each user to transmit individually in sequential order. In a reservation protocol this would be called polling; in a direct transmission protocol it would be called time-division multiplexing (TDM).

Hayes [1], in a seminal paper published in 1978, suggested that a more efficient method of polling would be to query groups of users simultaneously. The basic idea in Hayes' scheme is to quickly eliminate sets of inactive users which can be identified by a single query-response. Hayes called this technique "probing" and used subtrees of a binary tree to create his groups.

This same idea was utilized to improve the efficiency of a direct transmission protocol. Capetanakis [2], [3] also used a tree to create groups of users which were allowed to transmit their information simultaneously. When more than one active user was found to be in the group, the group was broken up into subgroups using subtrees of the tree. For an infinite number of users, Capetanakis suggested a random coin toss mechanism to determine whether or not a user would be contained in a given subtree. Capetanakis' ideas were extended to other than tree-like groupings by Gallager [4], Massey [5], Tsybakov *et al.* [6], and Humblet and Mosely [7], where in each case more and more efficient protocols were developed for effecting the partition.

The purpose of this paper is to give additional evidence to support the well-known phenomenon that a good new idea is often the reincarnation of a good old idea. In this case, however, the rebirth of the idea occurred before the end of its previous life. The problem which triggered its initial birth was the need to administer syphillis tests to millions of persons being inducted into the U.S. military services during World War II. The test for syphillis was a blood test called the Wasserman test. In 1943, Dorfman [8] suggested pooling the blood samples from S persons and applying the Wasserman test to a sample from the resultant pool. The Wasserman test had sufficient sensitivity that, for values of S of interest, the test would yield a negative result if and only if none of the individual samples in the pooled sample were diseased. Dorfman further suggested that if the pooled sample yielded a positive result then these S samples should be tested individually. Using a Bernoulli model with parameter p for the blood samples (where p is the probability that an individual sample is diseased), Dorfman calculated the value of S which yielded the smallest ratio of the number of tests to persons tested. For small p, the best group size S is approximately $p^{-1/2}$ and the resulting smallest ratio of tests to persons tested is approximately $2p^{1/2}$. Dorfman's paper was the beginning of a research area which has become known as *group testing*. (Note that the word "group" merely means a set of items and does not imply any mathematical structure.)

In 1957, Sterrett [9] suggested an improvement to Dorfman's procedure. In this improved procedure, once an individual diseased sample is identified, the remainder of the pool is again tested as a group. The probability that this second group test is positive is equal to the probability that two or more diseased samples were in the original pool of size S. For small p and S chosen as indicated above, this probability is very small.

Manuscript received April 26, 1984; revised June 26, 1984. This work was supported by the National Science Foundation under Grant ECS83-10771.

The author is with the Department of Electrical Engineering and Computer Science, University of California, San Diego, La Jolla, CA 92097, USA.

92097, USA.

0018-9448/85/0300-0185$01.00 ©1985 IEEE

0018-9448/85/0300-0185$01.00 ©1985 IEEE

The beginning of a general theory for group testing occurred in the 1959 paper of Sobel and Groll [10]. In addition to giving new testing strategies, Sobel and Groll listed a large number of applications for this theory. The multi-access application was not one of these, however.

In subsequent years a large number of papers were published on the general subject of group testing; the publication of these papers continues today. In the next section we survey the results from a few of these papers. We concentrate on Sobel and Groll's 1959 paper, which includes a tree search algorithm comparable to the one described by Hayes in 1978. In particular, Sobel and Groll introduced an interesting subclass of testing strategies and gave a procedure for determining the best testing strategy within this subclass. They then compared the efficacy of this strategy with several poorer strategies—one of these being a version of the tree search algorithm of Hayes.

It should be stated that two different mathematical models have been considered in classical group testing. The first model corresponds to the assumption of Bernoulli random variables, as in the original Dorfman paper. In the second model (sometimes called hypergeometric testing), it is assumed that the exact number of items in each of the two states is known prior to testing. Here we discuss only results from the Bernoulli model.

II. Classical Group Testing and Probing

Suppose one is given an ordered set of N items to be tested where, unless stated to the contrary, N is assumed finite. Each item is in one of two states, denoted "0" and "1", respectively. The state of the ith item will be governed by a Bernoulli random variable X_i where $P[X_i = 0] = (1 - p)$ and $P[X_i = 1] = p$. The N random variables are assumed to be statistically independent.

Tests are performed on subsets of the N items. The outcome of each test is either a 0 or a 1, the output 0 occurring if and only if all items in the subset tested are in the 0 state. Thus if the jth test is performed on the subset $(j_1, j_2, \cdots, j_s)$, then the outcome of this test is the random variable $Y_j = X_{j_1} + X_{j_2} + \cdots + X_{j_s}$ where the plus sign denotes the "inclusive or" operation. A test plan is a sequence of tests such that, at the completion of the test plan, the outcomes of these tests uniquely determine the states of all N items. We assume that we know the outcomes $Y_1, Y_2, \cdots, Y_{j-1}$ prior to specifying the test Y_j $(j \geq 2)$. The number L of tests in a test plan is itself a random variable which is a complicated function of the details of the test plan and is determined by the N random variables $X_1, X_2, \cdots, X_N$.

We are interested in finding the *optimal test plan*, that is the test plan which minimizes the expected number of tests, $\bar{L}$. We are also interested in the average number of tests required for this optimal test plan, $\bar{L}_{\min}$. Unfortunately, no general techniques are known for finding the optimal test plan or for computing $\bar{L}_{\min}$.

A specific test plan for four items is shown in Fig. 1. Table I shows the correspondence between the states of the four items and the outcomes of the tests. For this specific test plan, we can compute the average number of tests $\bar{L}$ as a function of p as

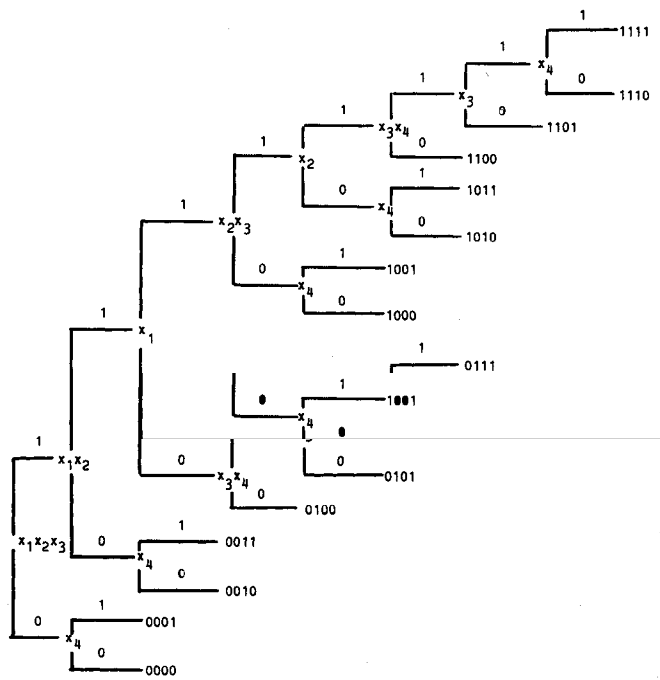

Fig. 1. A specific test plan.

TABLE I
Encoding of States by the Test Plan of Fig. 1

State	Outcome of Tests
1111	11111111
1110	11111110
1101	1111110
1100	111110
1011	111101
1010	111100
1001	11101
1000	11100
0111	110111
0110	110110
0101	11010
0100	1100
0011	101
0010	100
0001	01
0000	00

$$\bar{L} = 8p^4 + 27p^3(1-p) + 31p^2(1-p)^2 + 14p(1-p)^3 + 2(1-p)^4.$$

No claim is made for the optimality of this test plan.

It may be quickly recognized that the outcomes of the tests for a complete test plan can be considered as a binary variable length source code for the N Bernoulli random variables $X_1, X_2, \cdots, X_N$. Thus, from Shannon's source coding theorem we know that

$$\bar{L} \geq \bar{L}_{\min} \geq N h_2(p)$$

where $h_2(p)$ is the binary entropy function. Furthermore $\bar{L}_{\min}$ can be no smaller than the average length of a Huffman code [11] for N Bernoulli random variables. Both of these lower bounds for $\bar{L}_{\min}$ are contained in Sobel and Groll's 1959 paper. Unfortunately, no closed form solution

have been determined. Furthermore, let $\varnothing$ denote the empty set.

(3)

is known for the average length of a Huffman code for N Bernoulli random variables with parameter p.

One might be tempted to conjecture that $\bar{L}_{\min}$ is equal to the average length of a Huffman code—that is, that a Huffman code can be utilized to specify the optimal test plan. That this is not the case can be seen by considering the Huffman code for $N = 3$ random variables with parameter $p = 0.2$ as shown in Fig. 2. The root node (node A) of this tree corresponds to the test of all three items. If this first test fails (i.e., if $Y_1 = 1$), however, then there is no group test that corresponds to the next node (node B) of the tree, since there is no test that would fail when either $x_1 = 1$ or $x_2 = 1$ but not when both $x_1 = 1$ and $x_2 = 1$.

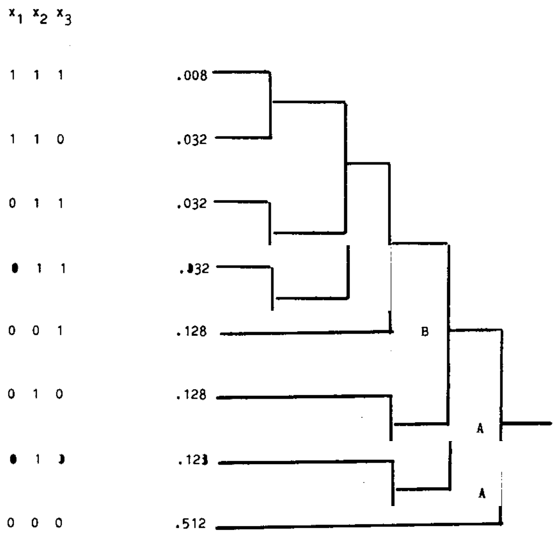

Fig. 2. Huffman code for $N = 3$ and $p = 0.2$.

The optimal test plan has been found for small values of N and for arbitrary p. To the best of our knowledge, the largest value of N for which the optimal test plan has been specified for all values of p $(0 < p < 1)$ is $N = 6$ [12].

The optimal test plan is known for $(\sqrt{5} - 1)/2 \le p < 1$ for arbitrary N [13]. The optimal test plan in this case consists of testing each item individually, resulting in $\bar{L}_{\min} = N$. To show that this is the case for $N = 2$ is a simple exercise in algebra. The proof for arbitrary N is given by Ungar [13].

Sobel and Groll [10] introduced a restricted set of test plans which have become known as *nested test plans*. They also gave a recursive algorithm for obtaining the optimal plan within this restricted set. A nested procedure is governed by the restriction that, once a test of two or more items results in a test outcome equal to 1, then the next test must be on a proper subset of the previously tested group.

A complete nested test plan can be described by the following algorithm. Let Z denote the set of N items to be tested, let U (the unknown set) be a set of items whose states are Bernoulli distributed, let A (the active set) be a set of items known to contain at least one item in state 1, and let K (the known set) be the set of items whose states have been determined. Furthermore, let $\varnothing$ denote the empty set.

Nested Test Plan Algorithm

Step 0) $U := Z$; $K := \varnothing$; $A := \varnothing$

Step 1) If $U = \varnothing$, then goto Step 5.

Step 2) Query a subset X of U of cardinality x (x may depend on the cardinality of U),
  if result = 0
    then $K := K + X$;
      $U := U - X$; goto Step 1
  if result = 1
    then $A := X$;
      $U := U - X$

Step 3) If cardinality of $A = 1$
  then $K := K + A$;
    $A := \varnothing$; goto Step 1

Step 4) Query a proper subset W of A of cardinality w (w may depend on the cardinality of A),
  if result = 0
    then $K := K + W$;
      $A := A - W$; goto Step 3
  if result = 1
    then $U := A - W + U$;
      $A := W$; goto Step 3

Step 5) End.

The explanation for almost all of the steps of this algorithm follows directly from the restriction of nested testing. The one step which deserves some mention is the step which occurs after one queries the proper subset W of the set A and the test result is 1. Note that the items in the set $A - W$ (that is items in the set A which are not in the subset W) are returned to the unknown set U. This is permissible since it is easy to verify that the random variables describing these items are again Bernoulli random variables so that these items can be treated as untested items.

Note that the only unspecified quantities in this algorithm are the chosen cardinalities of the sets X and W, namely x and w. These quantities are determined by the following recursion relations which describe the optimal algorithm. Let $G(a, b)$ be the expected number of additional queries required to complete the optimal test plan when there are a items in set A and b items in the union of sets A and U prior to a query. Then

$$G(0,0) = 0 \qquad G(0,1) = 1;$$

$$G(1,b) = G(0, b-1), \qquad b \ge 1, \tag{1}$$

$$G(0,n) = 1 + \min_{1 \le x \le n} \left[(1-p)^x G(0, n-x) + \left(1 - (1-p)^x\right) G(x, n) \right], \qquad n \ge 2, \tag{2}$$

$$G(a,n) = 1 + \min_{1 \le w < a} \left[\frac{(1-p)^w - (1-p)^a}{1 - (1-p)^a} \cdot G(a-w, n-w) + \frac{\left(1 - (1-p)^w\right)}{1 - (1-p)^a} G(w, n) \right], \qquad n \ge a \ge 2, \tag{3}$$

where (1) gives the initial conditions and describes Step 3 of the algorithm, and where (2) and (3) describe Steps 2 and 4 of the algorithm, respectively. The order in which these quantities are determined is: $G(0,0)$, $G(1,1)$, $G(0,1)$, $G(1,2)$, $G(2,2)$, $G(0,2), \cdots$. Note that the minimizing values of x and w obtained in (2) and (3) yield the optimum size groups to be tested. Furthermore note that, in order to find the optimum nested test plan for N items, one first finds the optimum nested test plan for $2, 3, \cdots, (N-1)$ items. The average number of tests required to test N items is given as $\bar{L}_{\text{minnest}} = G(0, N)$.

Sobel and Groll gave tables listing numerical values of $\bar{L}_{\text{minnest}}$, x and w for many values of p and N. They noted that the optimal nested test plan requires knowledge of x and w for all possible values of the cardinality of the sets A and U. They suggested that simpler suboptimal tests should be considered, and one of these simpler tests which they proposed was a version of the binary tree search of Hayes.

The tree search algorithm of Hayes is most easily described when $N = 2^k$. Then one can label the items $x_1, x_2, \cdots, x_N$ and treat the items as if they were leaves of a binary tree. Initially, one queries all 2^k items (i.e., those items stemming from the root node). If the response to this query is a 0, one has identified the states of all N items. If the response to the query is a 1, we know that there is at least one item in the 1 state. The items in the upper half of the tree are then queried. If the response is a 0 we query the items in the lower half of the tree. If the response is a 1, we again subdivide the previously queried set and first query the top half of the set. This process continues until the states of all 2^k items have been identified. Consider, as an explicit example, $N = 16$ items with items x_2, x_{11} and x_{12} in the 1 state (as shown in Fig. 3). The items queried and the responses for this example are given in Table II. Hayes noted that sometimes it is advantageous to treat a tree with 2^k modes as if it were 2^m trees each consisting of 2^{k-m} nodes. Hayes called the scheme *adaptive* if it chose m to minimize the average number of queries. The best choice of m depends upon the parameter p of the Bernoulli distribution.

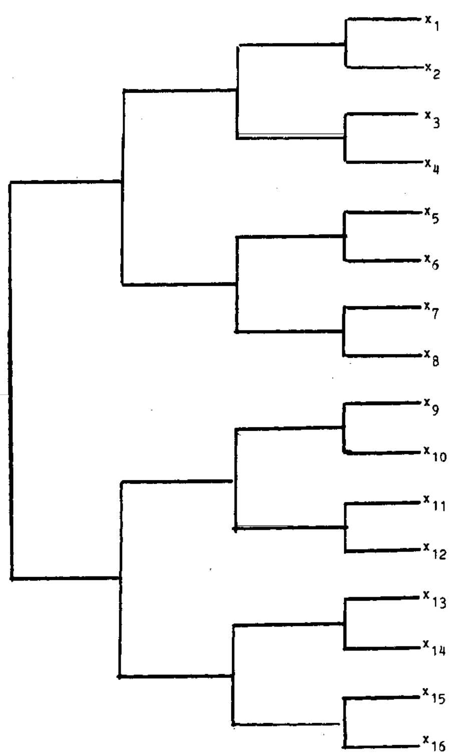

Fig. 3. Hayes tree for $N = 16$.

TABLE II
QUERIES AND RESPONSES FOR HAYES ALGORITHM WITH x_2, x_{11}, AND x_{12} IN 1 STATE

Query Number	Items Queried	Responses
1	$x_1, x_2, \cdots, x_{16}$	1
2	$x_1, x_2, \cdots, x_8$	1
3	x_1, x_2, x_3, x_4	1
4	x_1, x_2	1
5	x_1	0
6	x_2	1
7	x_3, x_4	0
8	x_5, x_6, x_7, x_8	0
9	$x_9, x_{10}, \cdots, x_{16}$	1
10	$x_9, x_{10}, x_{11}, x_{12}$	1
11	x_9, x_{10}	0
12	x_{11}, x_{12}	1
13	x_{11}	1
14	x_{12}	1
15	$x_{13}, x_{14}, x_{15}, x_{16}$	0

Hayes' algorithm can be improved further in a manner not described in his paper. The improvement is to omit queries whose answers can be predicted from previous answers (Queries 6 and 12 of Table II are of this type). Sobel and Groll used this method in their 1959 tree search algorithm, thus giving it an advantage over the Hayes algorithm. A comparison of the average number of tests for the adaptive Hayes algorithm, the improved adaptive Hayes algorithms (with unnecessary question omitted) and the optimal nested group testing is given in Table III for $N = 2, 4, 8, 16$ and $p = 0.1, 0.2, 0.3, 0.4, 0.5$.

The recursive equations of (1), (2) and (3) do not conveniently give information regarding the optimum nested algorithm in the limit as the number of items N approaches

TABLE III
AVERAGE NUMBER OF QUERIES FOR THREE TESTING PROCEDURES

N \ p	0.1	0.2	0.3	0.4	0.5
2	[1.38][a] (1.29)[b] ⟨1.29⟩[c]	[1.72] (1.56) ⟨1.56⟩	[2] (1.81) ⟨1.81⟩	[2] (2) ⟨2⟩	[2] (2) ⟨2⟩
4	[2.49] (2.11) ⟨2.05⟩	[3.44] (3.07) ⟨3.01⟩	[4] (3.62) ⟨3.60⟩	[4] (4) ⟨4⟩	[4] (4) ⟨4⟩
8	[4.98] (4.14) ⟨3.76⟩	[6.88] (6.14) ⟨5.91⟩	[8] (7.24) ⟨7.17⟩	[8] (8) ⟨8⟩	[8] (8) ⟨8⟩
16	[9.96] (8.28) ⟨7.22⟩	[13.76] (12.28) ⟨11.73⟩	[16] (14.48) ⟨14.28⟩	[16] (16) ⟨16⟩	[16] (16) ⟨16⟩

[a] [·]—Best adaptive Hayes.
[b] (·)—Best adaptive modified Hayes.
[c] ⟨·⟩—Optimal nested group testing result.

infinity. In this limiting case, however, one can completely specify the optimal testing strategy and its performance.

The following discussion is based upon joint research conducted with J. L. Massey. It is closely related to the work of Hwang [14] but was done independently. We define the optimal nested strategy for testing an infinite number of items as that nested algorithm which results in the minimum *ratio* of the average number of tests to the number of items categorized. (This ratio involves a limiting operation but we leave the details to the reader.) We assume that the items are assigned indices $1, 2, \cdots$ and that, when we choose a subset of size x, we choose that subset corresponding to those x unidentified items with the smallest indices. Let us for the moment consider another type of testing strategy which attempts to identify the *first* item in state 1; that is, the item with the smallest index which is in state 1. Let us again restrict ourselves to a nested strategy and let T^* be that nested test which identifies this single item in the smallest average number of tests. Our reason for considering such a test becomes clear when we note that the optimal nested test for processing all items is obtained by the repeated use of test T^*. Such is the case since any nested procedure will identify the items in state 1 in the order of their assigned indices and, after identifying an item in state 1, all items assigned a greater index will be Bernoulli points. Thus we need only to specify test T^* and evaluate its performance. But using our observation that the outcome of a test plan is a source code, we note that the optimal nested test plan for identifying the first item in state 1 can have an average number of tests no smaller than the average length of a binary Huffman code for an infinite source alphabet having a geometric distribution. Fortunately, in this case the Huffman code corresponds to a realizable group testing strategy. Gallager and Van Voorhis [15] have specified the Huffman code for a geometrically distributed alphabet. This Huffman code corresponds to the following group testing strategy for T^*.

1) Test $x = \lceil -\log(2-p)/\log(1-p) \rceil$ items until a response of 1 is obtained. Here $\lceil y \rceil$ is the ceiling function.
2) Define $\alpha = \lfloor \log_2 x \rfloor$ where $\lfloor y \rfloor$ is the floor function. Split the group of size x into two groups A and B as follows:
   if $x \le 3 \cdot 2^{\alpha-1}$, $A = \{\text{first } 2^{\alpha-1} \text{ items}\}$, $B = \{\text{last } x - 2^{\alpha-1} \text{ items}\}$
   if $x > 3 \cdot 2^{\alpha-1}$, $A = \{\text{first } x - 2^{\alpha} \text{ items}\}$, $B =$

WOLF: BORN AGAIN GROUP TESTING 189

Test groups A and B using a binary tree search. Group A requires $(\alpha - 1)$ tests and Group B, α tests. The average number of tests per item classified for this test plan is given as

$$\lim_{N\to\infty} \bar{L}(N)/N = p\left(\alpha + 1 + (1-p)^k/(1-(1-p)^x)\right) \tag{4}$$

where

$$k = 2^{\lfloor \log_2 \alpha \rfloor + 1} - \alpha. \tag{5}$$

TABLE IV
PERFORMANCE OF OPTIMAL NESTED ALGORITHMS ($N = \infty$)

p	Best Group Size x	Average Number of Tests to Find First Defective	Average Number of Tests per Item	Entropy Bound $h(p)$
0.01	69	8.105	0.0811	0.0808
0.05	14	5.762	0.2881	0.2864
0.1	7	4.725	0.4725	0.4690
0.15	4	4.092	0.6138	0.6098
0.2	3	3.639	0.7278	0.7219
0.25	2	3.286	0.8215	0.8113
0.3	2	2 961	0.8882	0.8813
0.4	1	2.500	1.0000	0.9710
0.45	1	2.222	1.0000	0.9928
0.5	1	2.000	1.0000	1.0000

This result serves as another lower bound for the ratio $\bar{L}_{\min}/N$ for any nested test plan for testing N items since the latter test, by repetition, is also a test for the infinite sample case. Note that $x = 1$ in Step 1 for $p > (\sqrt{5} - 1)/2$, a result which agrees with the previously discussed result by Ungar. Table IV compares the result given in (4) to the binary entropy function $h_2(p)$ for various values of p.

III. Extended Group Testing and Multiaccess Communications

In classical group testing, the outcome of each test is assumed to be a two-valued function of the states of the items tested. In multiaccess communications operating in a direct transmission mode, a three-valued outcome is usually assumed. That is, after a set of users has been given permission to transmit, one usually lumps the possible consequences of this happening as three events: no user transmits (an *idle*), one user transmits (a *success*) or two or more users transmit (a *collision*). We are thus led to develop a group testing strategy which incorporates such a three-valued test outcome.

As before, consider that N items are to be tested where each item is in one of two states, denoted 0 and 1, governed by statistically independent Bernoulli random variables with parameter p. Tests (or "enablings") are performed on subsets of the N items, the outcome of each test being a three-valued function with values 0, 1, and 2^+. Denoting by Y_j the outcome of the jth test,

$$Y_j = \begin{cases} 0, & \text{all items tested are in the 0 state,} \\ 1, & \text{exactly one item tested is in the 1 state,} \end{cases}$$

Now, a test plan is defined as a sequence of tests such that, at the completion of the test plan, each item in the 1 state appears in an enabled subset containing no other such item (i.e., each appears in a successful transmission). An optimal test plan is a test plan which requires the smallest average number of tests.

As for the case of classical group testing, very little is known about the synthesis of optimal test plans or their performance. An information-theoretic lower bound to the average number of tests for an optimal test plan follows directly from the work of Pippenger [16]. In work with D. Towsley reported in another paper [17], we defined a restricted class of optimal tests which are a natural generalization of the nested tests of Sobel and Groll to this situation. This class of optimal test plans is described by an algorithm slightly more complicated than the description of the nested test plan given in the previous section. Now, however, after each query, in addition to the known set K of completely classified items (where we note that an item in state 1 is not considered to be completely classified until it appears in an enabled set with no other such item) and the unknown set U of Bernoulli distributed items, there are two intermediate sets $A(1)$ and $A(2)$. Set $A(1)$ is the smallest subset of the remaining items certain to contain one or more items in the 1 state, and $A(2)$ is the smallest subset certain to contain two or more items in the 1 state. If $A(1)$ and $A(2)$ are both nonempty, then $A(1)$ is a proper subset of $A(2)$. The restriction on the tests are such that

a) If $A(1)$ is nonempty and $A(2)$ is empty, one must test either a subset of $A(1)$ or the union of the entire set $A(1)$ and a subset of U.
b) If $A(1)$ and $A(2)$ are both nonempty, one must test either a subset of $A(1)$ or the union of the entire set $A(1)$ and a subset of $A(2) - A(1)$.
c) If $A(1)$ is empty but $A(2)$ is not empty, one must test a proper subset of $A(2)$.

A description of the full algorithm follows. Note that the tests described in both b) and c) above are included in Step 2 of the algorithm since $A(1) = \varnothing$ implies $Y \subseteq A(2)$. Note also that Step 2 of the algorithm is never entered unless $A(2) \neq \varnothing$ and that Step 3 of the algorithm is never entered unless $A(1) \neq \theta$ and $A(2) = \varnothing$.

Algorithm

```
Step 0) U:= Z; K:= ∅; A(1):= ∅; A(2):= ∅;
Step 1) if U = ∅ then goto Step 4
          else enable a subset X ⊆ U;
            if test result is 0 or 1
              then K:= K + X;
                   U:= U − X;
                   goto Step 1
            if test result is 2+
              then A(2):= X;
                   U:= U − X;
                   goto Step 2
Step 2) enable a subset X ⊆ A(1) or X = A(1) + Y where
        Y ⊆ (A(2) − A(1))
          if test result is 0
            then K:= K + X;
                 A(2):= A(2) − X;
                 A(1):= A(1) − X;
                 goto Step 2
          if test result is 1
            then K:= K + X;
                 A(1):= A(2) − X;
                 A(2):= ∅;
          if test result is 2+
            then U:= U + (A(2) − X);
                 A(2):= X;
                 if X ⊆ A(1) then A(1):= ∅;
                 goto Step 2
Step 3) enable subset X ⊆ A(1) or X = A(1) + Y where
        Y ⊆ U
          if test result is 0
            then K:= K + X;
                 A(1):= A(1) − X;
                 goto Step 3
          if test result is 1
            then K:= K + X;
                 U:= U + (A(1) − X);
                 A(1):= ∅;
                 goto Step 1
          if test result is 2+
            then A(2):= X;
                 U:= U + (A(1) − X);
                 if X ⊆ A(1) then A(1):= ∅;
                 goto Step 2
Step 4) End.
```

The average number of tests for this test plan is a function of the size of the enabled sets in Steps 1–3 of the algorithm. To find the size of the enabled sets that minimizes this average, recursion relations similar to (1)–(3) are written for this algorithm. This has been done and the results are reported elsewhere [17].

We have also considered other situations where the outcomes of the tests are again two-valued, but where the two values differ from what has been described in the previous section. The interested reader is referred to Berger *et al.* [18] for details.

IV. Summary

A brief history of group testing has been presented. It was found that, in one of the earliest group testing papers (published in 1959), reference was made to a testing procedure based upon a tree search that is very closely related to testing procedures used in multiaccess communications (as described in papers published in 1978 and 1979).

The notions of group testing are interesting in their own right. However, they are particularly relevant to multiple-access communications problems–thus, this paper.

Acknowledgment

I was first told of the commonality between group testing and multi-access communications by Sandy Cutler, a graduate student. My colleague Don Towsley and I, along with several graduate students including Sandy Cutler, Shiv Panwar and Bob Trismen, have explored this relationship. We later learned of similar work being carried out at Cornell University by Toby Berger and Nader Mehravari, and we have collaborated with the Cornell group on another paper [18]. The work on finding the

190 IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-31, NO. 2, MARCH 1985

have been determined. Furthermore, let $\emptyset$ denote the empty set.

(3)

random variables was carried out with James Massey at the Eidgenössiche Technische Hochschule (ETH), Zurich. I would like to thank Frank Hwang of Bell Laboratories for making me aware of the vast literature on classical group testing and for some very interesting discussions. Finally, I would like to acknowledge the assistance of a very competent reviewer.

References

[1] J. H. Hayes, "An adaptive technique for local distribution," *IEEE Trans. Commun.*, vol. COM-21, pp. 1178–1186, Aug. 1978.

[2] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 505–515, Sept. 1979.

[3] ——, "Generalized TDMA: The multi-accessing tree protocol," *IEEE Trans. Commun.*, vol. COM-27, pp. 1476–1484, Oct. 1979.

[4] R. G. Gallager, "Conflict resolution in random access broadcast networks," in *Proc. AFOSR Workshop on Communication Theory and Applications*, pp. 74–76, Sept. 1978.

[5] J. L. Massey, "Collision-resolution algorithms and random-access communications," in *Multi-User Communication Systems*, G. Longo, Ed. New York: Springer-Verlag, 1981, pp. 74–137.

[6] B. S. Tsybakov, M. A. Berkovskii, N. D. Vredenskaja, V. A. Mikhailov, and S. P. Ferdorzov, "Methods of random multiple access," in *Proc. 5th Int. Symp. Information Theory*, USSR, July 1979.

[7] P. A. Humblet and J. Mosely, "Efficient accessing of a multiaccess channel," in *Proc. 19th IEEE Conf. Decision and Control*, Dec. 1980, pp. 624–627.

[8] R. Dorfman, "The detection of defective members of large populations," *Ann. Math. Statist.*, vol. 14, pp. 436–440, Dec. 1943.

[9] A. Sterrett, "On the detection of defective members of large populations," *Ann. Math. Statist.*, vol. 28, pp. 1033–1036, Dec. 1957.

[10] M. Sobel and P. A. Groll, "Group testing to eliminate efficiently all defectives in a binomial sample," *Bell Syst. Tech. J.*, vol. 38, pp. 1178–1252, Sept. 1959.

[11] D. A. Huffman, "A method for the construction of minimum redundancy codes," in *Proc. Inst. Radio Engineers*, vol. 40, p. 1098, Sept. 1952.

[12] Arnon Boneh, private communication.

[13] P. Ungar, "The cutoff point for group testing," *Commun. Pure Appl. Math.*, vol. XIII, pp. 49–54, Feb. 1960.

[14] F. K. Hwang, "On finding a single defective in binomial group testing," *J. Amer. Statist. Ass., Theory Methods Sect.*, vol. 69, pp. 146–150, June 1974.

[15] R. G. Gallager and D. C. Van Voorhis, "Optimal source codes for geometrically distributed integer alphabets," *IEEE Trans. Inform. Theory*, vol. 21, pp. 228–229, Mar. 1975.

[16] N. Pippenger, "Bounds on the performance of protocols for a multiple-access broadcast channel," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 145–151, Mar. 1981.

[17] D. Towsley and J. K. Wolf, "An application of group testing to the design of multi-user access protocols," in *Proc. 19th Ann. Allerton Conf. Communications, Control and Computing*, Urbana, IL, Oct. 1981, pp. 397–403.

[18] T. Berger, N. Mehravari, D. Towsley, and J. K. Wolf, "Random multiple-access communication and group testing," to be published in *IEEE Trans. Commun.*, July, 1984.

$x_1 = 1$ or $x_2 = 1$ but not when both $x_1 = 1$ and $x_2 = 1$.

then $K := K + A$;
$A := \emptyset$; goto Step 1