Combinatorics, Probability and Computing (2005) 14, 191–202. © 2005 Cambridge University Press
DOI: 10.1017/S0963548304006649 Printed in the United Kingdom

Improved Results for Competitive Group Testing

JENS SCHLAGHOFF and EBERHARD TRIESCH

RWTH Aachen, Lehrstuhl II für Mathematik, D-52062 Aachen, Templergraben 55, Germany
(e-mail: triesch@math2.rwth-aachen.de)

Received 11 April 2003; revised 27 August 2003

We consider algorithms for group testing problems when nothing is known in advance about the number of defectives. Du and Hwang suggested measuring the quality of such algorithms by its so-called (first) competitive ratio (see the Introduction). Later, Du and Park suggested a second kind of competitive ratio. For each kind of competitiveness, we improve the best-known bounds: in the first case, from 1.65 to $1.5 + \varepsilon$, and in the second from 16 to 4.

1. Introduction

Suppose that we are given some set S of n items some of which are defective. Our task is to identify the defectives by sequentially choosing subsets $T \subset S$ and testing whether T contains at least one defective item. If this is true, we call T *contaminated*, otherwise *pure*.

This is the group testing problem which was first introduced by Dorfman (*cf.* [4]) in the context of testing blood samples for containing syphilitic antigens, but has by now found many interesting applications in very different areas such as statistics, information theory, pattern recognition and computer networks. For a detailed and interesting account, we refer the reader to [7]. The books [2] and [1] also provide introductions to group testing problems.

There are many results for the minimum number of tests, $M(d, n)$, in the worst case if the number d of defectives is known (*cf.* [7]), *e.g.*, the upper bound $M(d, n) \leqslant \lceil \log \binom{n}{d} \rceil + d - 1$ due to Hwang. As lower bounds, we mention the information-theoretic bound $M(d, n) \geqslant \lceil \log \binom{n}{d} \rceil$ as well as Theorem 2.1 and Lemma 2.2 below.

On the other hand, in many applications the *a priori* knowledge of d is an unrealistic assumption. But if nothing is known about the defectives at the beginning, the worst case number of tests is obviously n, a bound which is achieved by testing all items individually.

This 'optimal' algorithm has the disadvantage that its performance is very bad if you compare it to $M(d, n)$ if d is small. Often, we want an algorithm which is not too bad compared to $M(d, n)$ whatever the number of defectives is. This led Du and Hwang [6]

https://doi.org/10.1017/S0963548304006649 Published online by Cambridge University Press

to the concept of *competitive group testing*: Assume that A is an algorithm solving the general group testing problem (no knowledge of d) and a some real number, $a \geqslant 1$. Denote by $M_A(d\,|\,n)$ the worst case number of tests which A performs if the number of defectives is d. Then A is called *a-competitive* if there is some constant b such that

$$M_A(d\,|\,n) \leqslant aM(d,n) + b,$$

for all $0 \leqslant d < n$, and a is called a competitive ratio for A. Du and Hwang [6] gave a 2.75-competitive algorithm. This ratio was improved by Bar-Noy, Hwang, Kessler and Kutten [3] to 2 and later on, Du, Xue, Sun and Cheng [9] gave an algorithm with competitive ratio 1.65. In this paper we give $(1.5+\varepsilon)$-competitive algorithms for all positive ε.

There is also another notion of competitiveness introduced by Du and Park [8]. Given an algorithm A, they consider the threshold functions

$$n(d,k) = \max\{n\,|\,M(d,n) \leqslant k\} \qquad \text{and}$$
$$n_A(d\,|\,k) = \max\{n\,|\,M_A(d\,|\,n) \leqslant k\}.$$

Bounds for $M(d,n)$ and $n(d,k)$ are closely related. For example, from the information-theoretic bound mentioned above we obtain the inequality

$$n(d,k) \leqslant \max\left\{n\,|\,\log\binom{n}{d} \leqslant k\right\}$$

which, along with similar bounds, is used in Section 3.

Algorithm A is called *strongly c-competitive* if there is some constant a such that, for all $d \geqslant 1, k \geqslant 1$, the following inequality holds:

$$n(d,k) \leqslant c\, n_A(d\,|\,k) + a. \tag{1.1}$$

There is some problem with this definition. For $d \geqslant k+1$ we have $M(d,n) \leqslant k$ if and only if $n = d$, since $M(d,d) = 0$ and $M(d,d+1) = d$. On the other hand, an algorithm without information on the number of defectives cannot prove that all elements are defective without probing them all, which means that $\{n\,|\,M_A(d\,|\,n) \leqslant k\} = \emptyset$ for $d \geqslant k+1$. Hence, inequality (1.1) does not make sense in this case. Similarly, there might be 'good' algorithms (like the one in [8]) with the property that there is some constant $\gamma > 1$ such that at least γd questions are needed to find d defectives. As a consequence, it could also happen for some other pairs (d,k) that $\{n\,|\,M_A(d\,|\,n) \leqslant k\} = \emptyset$ whereas $\{n\,|\,M(d,n) \leqslant k\} \neq \emptyset$. So we propose to relax the definition of strong competitiveness by claiming inequality (1.1) only for all (d,k) with $k \geqslant \gamma d$, where γ is some constant depending on the algorithm. In Section 3, we present a strongly c-competitive algorithm with $c = 4$. The best-known value of c up to now was 16 (see [8]).

2. Competitive ratio $1.5 + \varepsilon$

Given ε with $0 < \varepsilon < 0.01$, we are going to develop an algorithm $A = A_\varepsilon$ satisfying $M_A(d\,|\,n) \leqslant (1.5+\varepsilon)M(d,n) + \lceil 0.4(1/\varepsilon)\rceil + 6$ for all d with $0 \leqslant d < n$. In our description

https://doi.org/10.1017/S0963548304006649 Published online by Cambridge University Press

and analysis of the algorithm, we try to keep as close to the terminology in [7, Chapter 7] as possible.

As in [3], we test disjoint sets of size $4, 4^2, \ldots, 4^j, \ldots$, until a contaminated set X of size 4^i is found. If $i = 1$, three of the four items are tested individually according to the procedure 4-TEST(X), and we identify at least one defective item. If $i > 1$, the procedure BIG-4-TEST(X) is employed to find a contaminated subset Y of cardinality 4^{i-1}. By using the function DIG(Y) (binary splitting), we again identify a defective item and start testing subsets of the not yet identified elements of cardinality $4, 4^2, \ldots, 4^j, \ldots$, as before.

Two exceptions have to be noted. If the first $k-1$ tests are pure, $k = \lceil 0.4(1/\varepsilon) \rceil$, then we first test the whole remaining set. If the remaining set has cardinality at most three, the items are tested individually.

For the following more formal description of the above mentioned procedures, we fix ε with $0 < \varepsilon < 0.01$. Let k denote the number $\lceil 0.4(1/\varepsilon) \rceil$ throughout this section. The command TEST(X) means that we test whether the set X contains defective elements or not. By G (respectively D), we denote the set of identified good (respectively defective) elements.

Algorithm A;
input S;
$D \leftarrow \emptyset$;
$G \leftarrow \emptyset$;
while $|S| \geqslant 4$ **do**
$i = 1$;
repeat
$X \leftarrow \min(4^i, |S|)$ items from S;
TEST(X);
if X is pure **then** $G = G \cup X$
$S = S \setminus X$
$i = i + 1$;
if $i = k$ **then TEST**(S);
if S is pure **then** $G \leftarrow G \cup S$
$S = \emptyset$;
until $S = \emptyset$ or X is contaminated;
if X is contaminated **then**
if $i = 1$ **then 4-TEST**(X);
if $i > 1$, **then BIG-4-TEST**(X)
end-while;
while $S \neq \emptyset$ **do**
$x \leftarrow$ an item from S;
TEST($\{x\}$);
if $\{x\}$ is pure **then** $G \leftarrow G \cup \{x\}$ **else** $D \leftarrow D \cup \{x\}$;
$S \leftarrow S \setminus \{x\}$;
end-while;
end-algorithm.

https://doi.org/10.1017/S0963548304006649 Published online by Cambridge University Press

```
Procedure 4-TEST({x, y, z, w});
    TEST({x});
    TEST({y});
    TEST({z});
    if x is defective
      then D ← D ∪ {x}
      else G ← G ∪ {x};
    if y is defective
      then D ← D ∪ {y}
      else G ← G ∪ {y};
    if z is defective
      then D ← D ∪ {z}
      else G ← G ∪ {z};
    if {x, y, z} is pure
      then S ← S \ {x, y, z, w}
           D ← D ∪ {w}
      else S ← S \ {x, y, z};
end-procedure;


Function DIG(X);
    Y ← X;
    repeat Z ← ⌈| Y |/2⌉ items from Y;
           TEST(Z);
           if Z is contaminated then Y ← Z
                                else Y ← Y \ Z;
    until | Y | = 1;
    DIG ← Y;
end-function;


Procedure BIG-4-TEST(X);
    partition X into 4 pairwise disjoint sets Y, Z, U, W of equal cardinality;
    TEST(Y)
    if Y is contaminated
      then D ← D ∪ DIG(Y)
           S ← S \ DIG(Y);
      else begin G ← G ∪ Y;
                 S ← S \ Y;
                 TEST(Z);
                 if Z is contaminated
                   then D ← D ∪ DIG(Z)
                        S ← S \ DIG(Z);
                   else begin G ← G ∪ Z;
                              S ← S \ Z;
```

https://doi.org/10.1017/S0963548304006649 Published online by Cambridge University Press

```
                    TEST(U);
                    if U is contaminated
                      then D ← D ∪ DIG(U)
                           S ← S \ DIG(U);
                      else G ← G ∪ U
                           D ← D ∪ DIG(W)
                           S ← S \ (U ∪ DIG(W));
              end-else
      end-else
end-procedure;
```

For the analysis of the algorithm, we need the following results, which can all be found in [7].

Theorem 2.1. *If the ratio d/n is at least $8/21$, then $M(d,n) = n - 1$.*

This is Theorem 3.5.3 in [7]; see also [5].

Lemma 2.2. *For $0 \leqslant d < (8/21)n$,*

$$M(d,n) \geqslant d\left(\log \frac{n}{d} + 1.096\right) - 0.5 \log d - 1.222.$$

Lemma 2.3. *Let $d = d' + d''$ and $n = n' + n''$ where $d' \geqslant 0, d'' \geqslant 0, n' > 0, n'' > 0$. Then*

$$d' \log \frac{n'}{d'} + d'' \log \frac{n''}{d''} \leqslant d \log \frac{n}{d}.$$

The last two results are Corollary 7.2.1 and Lemma 7.2.2 in [7].

Now assume that for the first time algorithm A detects a contaminated set and let 4^i denote its cardinality. We are going to consider the following cases.

Case 1: $i = 1$.
This means that the first test is positive. The algorithm proceeds by testing three of the four items individually. Hence, after four tests we have the following identifications:

1(a) 3 defectives, or
1(b) 2 defectives and 1 good item, or
1(c) 1 defective and at least 2 good items.

Case 2: $k > i > 1$.
In the first $i - 1$ tests, the algorithm identified

$$4 + 16 + \ldots + 4^{i-1} = \frac{1}{3}(4^i - 1) - 1$$

good items. After the ith (positive) test, the procedure BIG-4-TEST follows, which yields: (a) 1 defective with $2i - 1$ tests, or (b) 1 defective and 4^{i-1} good items with $2i$ tests, or (c) 1 defective and at least 2.4^{i-1} good items with $2i + 1$ tests.

https://doi.org/10.1017/S0963548304006649 Published online by Cambridge University Press

To summarize: After the application of BIG-4-TEST we identified:

2(a) $(1/3)(4^i - 1)$ items with $3i - 1$ tests, or
2(b) $(1/3)(7 \cdot 4^{i-1} - 1)$ items with $3i$ tests, or
2(c) $(1/3)(10 \cdot 4^{i-1} - 1)$ items with $3i + 1$ tests,

where always one of the identified items is defective.

Case 3: $i \geqslant k$.
We arrive at Cases 3(a)–(c), which are completely analogous to Cases 2(a)–(c) above, with the only difference that one additional test of the whole (remaining) set S is executed.

We can now prove the following.

Lemma 2.4. *For all fixed* $0 < \varepsilon < 0.01$, *algorithm A satisfies*

$$M_A(d \mid n) \leqslant (1.5 + \varepsilon)d\left(\log \frac{n}{d} + 1.09\right) + k.$$

Proof. The proof is by induction on d.

If $d = 0$, this is recognized by the algorithm after at most $k = \lceil 0.4(1/\varepsilon) \rceil$ tests. (If there are $k - 1$ negative tests, the whole set S is tested and for $n \leqslant 3$, $M_A(0 \mid n) \leqslant 3 < k$.) Hence, the claimed inequality holds.

We now suppose $d > 0$ and study Cases 1–3 above.

Case 1: $i = 1$.

1(a) Three defectives were identified by 4 tests, hence

$$\begin{aligned} M_A(d \mid n) &\leqslant 4 + M_A(d - 3 \mid n - 3) \\ &\leqslant (1.5 + \varepsilon)3\left(\log \frac{3}{3} + 1.09\right) + (1.5 + \varepsilon)(d - 3)\left(\log \frac{n - 3}{d - 3} + 1.09\right) + k \\ &\qquad \text{(by the induction hypothesis)} \\ &\leqslant (1.5 + \varepsilon)d\left(\log \frac{n}{d} + 1.09\right) + k \\ &\qquad \text{(by Lemma 2.3).} \end{aligned}$$

1(b) We estimate as follows:

$$\begin{aligned} M_A(d \mid n) &\leqslant 4 + M_A(d - 2 \mid n - 3) \\ &\leqslant (1.5 + \varepsilon)2\left(\log \frac{3}{2} + 1.09\right) + (1.5 + \varepsilon)(d - 2)\left(\log \frac{n - 3}{d - 2} + 1.09\right) + k \\ &\leqslant (1.5 + \varepsilon)d\left(\log \frac{n}{d} + 1.09\right) + k. \end{aligned}$$

https://doi.org/10.1017/S0963548304006649 Published online by Cambridge University Press

1(c) Similarly we have

$$\begin{aligned} M_A(d \mid n) &\leqslant 4 + M_A(d-1 \mid n-3) \\ &\leqslant (1.5+\varepsilon)(\log 3 + 1.09) + (1.5+\varepsilon)(d-1)\left(\log \frac{n-3}{d-1} + 1.09\right) + k \\ &\leqslant (1.5+\varepsilon)\left(\log \frac{n}{d} + 1.09\right) + k. \end{aligned}$$

Case 2: $k > i > 1$.
Using the induction hypothesis and Lemma 2.3 it suffices to show the following:

2(a) $$3i - 1 \leqslant (1.5+\varepsilon)\left(\log\left(\frac{1}{3}(4^i - 1)\right) + 1.09\right),$$

2(b) $$3i \leqslant (1.5+\varepsilon)\left(\log\left(\frac{1}{3}(7.4^{i-1} - 1)\right) + 1.09\right),$$

2(c) $$3i + 1 \leqslant (1.5+\varepsilon)\left(\log\left(\frac{1}{3}(10.4^{i-1} - 1)\right) + 1.09\right).$$

These inequalities can be proved as follows.

Case 2(a)

$$\begin{aligned} (1.5+\varepsilon)\left(\log\left(\frac{1}{3}(4^i - 1)\right) + 1.09\right) &\geqslant (1.5+\varepsilon)(-\log 3 + \log 4^i - \delta + 1.09) \\ &\qquad (\text{where } \delta \leqslant \log 4^2 - \log(4^2 - 1) < 0.1) \\ &\geqslant (1.5+\varepsilon)(2i - 0.6) \\ &= 3i - 0.9 + \varepsilon(2i - 0.6) > 3i - 1. \end{aligned}$$

In the estimate of δ we used that $\log x - \log(x-1)$ is a decreasing function of x for all $x \geqslant 1$. The same estimate of δ, namely $\delta < 0.01$, is used in Cases (b) and (c).

Case 2(b)

$$\begin{aligned} (1.5+\varepsilon)\left(\log\left(\frac{1}{3}(7.4^{i-1} - 1)\right) + 1.09\right) &\geqslant (1.5+\varepsilon)(-\log 3 + \log 7.4^{i-1} - \delta + 1.09) \\ &\geqslant (1.5+\varepsilon)(2i + 0.2) \\ &\geqslant (3i + 0.3) + \varepsilon(2i + 0.2) > 3i. \end{aligned}$$

Case 2(c)

$$\begin{aligned} (1.5+\varepsilon)\left(\log\left(\frac{1}{3}(10.4^{i-1} - 1)\right) + 1.09\right) &\geqslant (1.5+\varepsilon)(-\log 3 + \log 10.4^{i-1} - \delta + 1.09) \\ &\geqslant (1.5+\varepsilon)(2i + 0.7) \\ &\geqslant (3i + 1.5) + \varepsilon(2i + 0.7) > 3i + 1. \end{aligned}$$

https://doi.org/10.1017/S0963548304006649 Published online by Cambridge University Press

Case 3: $i \geqslant k$.
Arguing as above, we have to show:

3(a) $$3i \leqslant (1.5+\varepsilon)\left(\log\left(\frac{1}{3}(4^i-1)\right)+1.09\right),$$

3(b) $$3i+1 \leqslant (1.5+\varepsilon)\left(\log\left(\frac{1}{3}(7.4^{i-1}-1)\right)+1.09\right),$$

3(c) $$3i+2 \leqslant (1.5+\varepsilon)\left(\log\left(\frac{1}{3}(10.4^{i-1}-1)\right)+1.09\right).$$

The estimates are similar to Case 2, the only difference is that this time we estimate $\delta \leqslant \log 4^k - \log 4^{k-1} < 0.001$, since $\varepsilon < 0.01$ and hence $k > 40$. We omit the details. □

The next lemma gives a bound for $M_A(d\,|\,n)$ depending on n alone.

Lemma 2.5. *For $0 \leqslant d \leqslant n$, we have*

$$M_A(d\,|\,n) \leqslant \frac{4}{3}n.$$

Proof. We proceed by induction on d.

Suppose first that $d = 0$. If $n \leqslant 3$, we need n tests and for $n = 4$, one test is sufficient. For $\frac{1}{3}(4^i-1) \leqslant n < \frac{1}{3}(4^{i+1}-1)$ and $2 \leqslant i < k$, the algorithm uses i tests and it is clear that $i < n$ in this range. If n is even larger, exactly k tests are used and our claim is trivial.

Now assume that $d > 0$ and that the ith test detects the first contaminated set. If $i = 1$, the critical case is that we recognize 3 items with 4 tests. By applying the induction hypothesis, we infer that

$$M_A(d\,|\,n) \leqslant \left(\frac{4}{3}\right)3 + \left(\frac{4}{3}\right)(n-3) = \frac{4}{3}n.$$

For $i \geqslant 2$, let j denote the number of tests until the first defective item is identified and let m denote the number of identified items after j tests. By the induction hypothesis, it suffices to show that $j \leqslant \frac{4}{3}m$. This inequality is easily checked by looking at Cases 2(a)–(c) and 3(a)–(c) described above. □

We are now ready to prove the following.

Theorem 2.6. *For $0 \leqslant d < n$ and $0 < \varepsilon < 0.01$, we have*

$$M_A(d\,|\,n) \leqslant (1.5+\varepsilon)M(d,n) + k + 6,$$

where $k = \lceil 0.4(1/\varepsilon)\rceil$ as above.

Proof. If $d = 0$, obviously $M_A(d\,|\,n) \leqslant k$.

For $d \geqslant (8/21)n$, we use Theorem 2.1 and Lemma 2.5:

$$\begin{aligned} M_A(d\,|\,n) &\leqslant \frac{4}{3}n \\ &\leqslant (1.5)(n-1)+6 \\ &= (1.5)M(d,n)+6. \end{aligned}$$

https://doi.org/10.1017/S0963548304006649 Published online by Cambridge University Press

Now suppose that $0 < d < (8/21)n$. Lemmas 2.2 and 2.4 yield the following inequalities:

$$M_A(d \mid n) \leqslant (1.5 + \varepsilon)d\left(\log \frac{n}{d} + 1.09\right) + k, \quad \text{and}$$

$$M(d, n) \geqslant d\left(\log \frac{n}{d} + 1.096\right) - 0.5 \log d - 1.222.$$

It follows that

$$M_A(d \mid n) \leqslant (1.5 + \varepsilon)M(d, n) + k + (1.5 + \varepsilon)h(d),$$

where $h(d) = (0.5) \log d + 1.222 - (0.006)d$.

It is easy to check by elementary calculus that $h(d)$ is a unimodal function for $d \geqslant 1$ with a unique maximum at $d = 1/((0.012) \ln 2) \approx 120.2$. For integer d, we may thus estimate

$$h(d) \leqslant \max\{h(120), h(121)\} \leqslant 3.972.$$

It follows that

$$\begin{aligned} M_A(d \mid n) &\leqslant (1.5 + \varepsilon)M(d, n) + k + (1.51)(3.972) \\ &\leqslant (1.5 + \varepsilon)M(d, n) + k + 6. \end{aligned}$$

The proof is complete. □

3. A strongly 4-competitive algorithm

In this section, we are going to develop the promised strongly 4-competitive algorithm. The following lemma is completely analogous to Theorem 2.3 in [8].

Lemma 3.1. *Suppose that for all* $1 \leqslant d \leqslant n$ *and some* $c > 0$, *algorithm A satisfies the inequality*

$$M_A(d \mid n) \leqslant \log \binom{n}{d} + cd. \tag{3.1}$$

Then A is 2^c*-competitive. If c is a natural number, and* (3.1) *is replaced by*

$$M_A(d \mid n) \leqslant \left\lceil \log \binom{n}{d} \right\rceil + cd, \tag{3.2}$$

then the same conclusion holds.

Proof. Assume that either (3.1) holds or (3.2) holds with integral c. In any of the two cases we have

$$\begin{aligned} n_A(d \mid k) &\geqslant \max\left\{n \mid \log \binom{n}{d} + cd \leqslant k\right\} \\ &= \max\left\{n \mid \binom{n}{d} \leqslant 2^{k-cd}\right\} \quad \text{and} \end{aligned}$$

https://doi.org/10.1017/S0963548304006649 Published online by Cambridge University Press

$$n(d,k) \leqslant \max\left\{ n \mid \log \binom{n}{d} \leqslant k \right\}$$
$$= \max\left\{ n \mid \binom{n}{d} \leqslant 2^k \right\}.$$

So suppose that $\binom{n}{d} \leqslant 2^{k-cd}$ and $\binom{n+1}{d} > 2^{k-cd}$. Then

$$\binom{2^c(n+1)}{d} = \frac{1}{d!}\Big\{[2^c(n+1)][2^c(n+1)-1]\cdots[2^c(n+1)-d+1]\Big\}$$
$$\geqslant \frac{1}{d!}\Big\{[2^c(n+1)][2^c n][2^c(n-1)]\cdots[2^c(n-d+1)]\Big\}$$
$$= 2^{cd}\binom{n+1}{d} > 2^k.$$

Hence,

$$n(d,k) < 2^c(n+1) \leqslant 2^c(n_A(d \mid k)+1). \qquad \square$$

The following result ([10, Theorem 5]) is crucial. It deals with the situation that the possible *sets* of defective elements are the edges of some hypergraph and that we are given some probability distribution on the edge set of that hypergraph. The probability $p(e)$ of some edge e can be interpreted as the probability that e is the set of defective elements we are looking for. This set of defective elements is called *the unknown edge* in the statement of Theorem 3.2. We also note that the proof of Theorem 3.2 is constructive.

Theorem 3.2. *Suppose $G = (S, E)$ is a hypergraph and $p(e), e \in E$ are (positive) probabilities on its edges. Then there is a group testing algorithm finding the unknown edge e^* by at most $\lceil -\log p(e^*) \rceil + |e^*|$ tests.*

Our strongly 4-competitive algorithm follows from Theorem 3.2 by applying it to the special case where the hypergraph G is the whole power set, $G = (S, 2^S)$, $|S| = n$, and the probability distribution is given by

$$p(\emptyset) := \eta,$$
$$p(e) := \binom{n}{|e|}^{-1} 2^{-|e|} \text{ for } \emptyset \neq e \neq S,$$
$$p(S) := 2^{-n+1} - \eta.$$

Here η is some positive quantity (which may depend on n) such that

$$-\log(1 - \eta 2^{n-1}) < 1.$$

Theorem 3.3. *There exists a group testing algorithm B such that, for all $1 \leqslant d \leqslant n$,*

$$M_B(d \mid n) \leqslant \left\lceil \log \binom{n}{d} \right\rceil + 2d.$$

Algorithm B is strongly 4-competitive.

https://doi.org/10.1017/S0963548304006649 Published online by Cambridge University Press

Proof. Let $|S| = n$ and consider the power set of S with the above probability distribution. Theorem 3.2 guarantees an algorithm B finding the set e^* of defectives by at most $\lceil -\log p(e^*) \rceil + |e^*|$ tests. Note that the probability of a set only depends on its cardinality. For $|e^*| = d$ and $1 \leqslant d \leqslant n-1$, we have

$$\begin{aligned} \lceil -\log p(e^*) \rceil + |e^*| &= \left\lceil \log \binom{n}{d} + d \right\rceil + d \\ &= \left\lceil \log \binom{n}{d} \right\rceil + 2d. \end{aligned}$$

For $d = n$, we estimate

$$\begin{aligned} \lceil -\log p(e^*) \rceil + |e^*| &= \left\lceil -\log \frac{1 - \eta 2^{n-1}}{2^{n-1}} \right\rceil + n \\ &= \lceil n - 1 - \log(1 - \eta 2^{n-1}) \rceil + n \leqslant 2n. \end{aligned}$$

Lemma 3.1 now implies that B is strongly 4-competitive. □

Remarks. **(i)** We also tried to apply Theorem 3.2 in order to get a good competitive algorithm in the first sense. Choose some small $\delta > 0$ and let $2^S \setminus \{\emptyset\}$ be the edge set of our hypergraph. Now consider the following probability distribution:

$$p(e) = \frac{\delta}{n} \quad \text{if } |e| = 1, \quad \text{and}$$

$$p(e) = \frac{1 - \delta}{n - 1} \binom{n}{|e|}^{-1} \quad \text{if } |e| > 1.$$

It can be shown that the algorithm provided by Theorem 3.2 is competitive in the first sense with ratio approximately $\log 3 = 1.58\ldots$. Since this is not as good as the algorithm in Section 2, we omit the details.

(ii) It seems that, until now, nobody has been able to prove nontrivial lower bounds for competitive group testing. This is certainly a central open problem in this field of research.

Acknowledgement

We thank the referees for their careful reading and some helpful suggestions.

References

[1] Ahlswede, R. and Wegener, I. (1979) *Suchprobleme*, Teubner, Stuttgart.
[2] Aigner, M. (1988) *Combinatorial Search*, Wiley–Teubner, Stuttgart and Chichester.
[3] Bar-Noy, A., Hwang, F. K., Kessler, I. and Kutten, S. (1994) A new competitive algorithm for group testing. *Discrete Appl. Math.* **52** 29–38.
[4] Dorfman, R. (1943) The detection of defective members of large populations. *Ann. Math. Statist.* **14** 436–440.
[5] Du, D. Z. and Hwang, F. K. (1982) Minimizing a combinatorial function. *SIAM J. Alg. Discrete Methods* **3** 523–529.
[6] Du, D. Z. and Hwang, F. K. (1993) Competitive group testing. *Discrete Appl. Math.* **45** 221–232.

https://doi.org/10.1017/S0963548304006649 Published online by Cambridge University Press

[7] Du, D. Z. and Hwang, F. K. (2000) *Combinatorial Group Testing and its Applications*, 2nd edn, World Scientific, Singapore.
[8] Du, D. Z. and Park, H. (1994) On competitive group testing. *SIAM J. Comput.* **23** 1019–1025.
[9] Du, D. Z., Xue, G.-L., Sun, S.-Z. and Cheng, S.-W. (1994) Modifications of competitive group testing. *SIAM J. Comput.* **23** 82–96.
[10] Triesch, E. (1996) A group testing problem for hypergraphs of bounded rank. *Discrete Appl. Math.* **66** 185–188.

https://doi.org/10.1017/S0963548304006649 Published online by Cambridge University Press