Randomized Group Testing Both Query-Optimal And Minimal Adaptive

Peter Damaschke and Azam Sheikh Muhammad

Department of Computer Science and Engineering
Chalmers University, 41296 Göteborg, Sweden
[ptr,azams]@chalmers.se

Abstract. The classical group testing problem asks to determine at most d defective elements in a set of n elements, by queries to subsets that return Yes if the subset contains some defective, and No if the subset is free of defectives. By the entropy lower bound, $\log_2 \sum_{i=0}^{d} \binom{n}{i}$ tests, which is essentially $d \log_2 n$, are needed at least. We devise group testing strategies that combine two features: They achieve this optimal query bound asymptotically, with a factor $1 + o(1)$ as n grows, and they work in a fixed number of stages of parallel queries. Our strategies are randomized and have a controlled failure probability, i.e., constant but arbitrarily small. We consider different settings (known or unknown d, probably correct or verified outcome), and we aim at the smallest possible number of stages. In particular, 2 stages are sufficient if d grows slowly enough with n, and 4 stages are sufficient if $d = o(n)$.

Keywords: algorithmic learning theory, randomized algorithm, parallel queries, bioinformatics, group testing

1 Introduction

Suppose that a set of n elements contains an unknown subset of defective elements ("defectives" for brevity). A group test takes any subset, called a pool, and returns a binary answer: The pool is positive if it contains at least one defective, and otherwise negative. The group testing problem is the problem of identifying the defectives using a minimum number of group tests, also called queries. An upper bound d on the number of defectives may be known in advance, or the number d of defectives may be unknown. However, we assume that $d \ll n$.

Group testing is a classical combinatorial search problem [10] and an important example of the "exact learning by queries" model. It has applications in biological and chemical testing and diagnosis [10, 11, 18], communication networks [4, 9, 13], and streaming algorithms [5], to mention only a few domains.

A group testing strategy works in stages, where the pools for every stage are prepared prior to the stage, and then queried in parallel. The pools for the next stage, however, may depend on all previous answers. A strategy with one query per stage is called adaptive. In most group testing applications, highly parallel strategies working in a few stages are preferable because, on the one hand, the tests are time-consuming, and on the other hand, many pools can be tested simultaneously.

As a notational remark, we omit ceiling brackets in expressions, in order to avoid bulky notation. Logarithms are always base 2.

Due to the entropy lower bound, also known as the information-theoretic lower bound, at least $\log_2 \sum_{i=0}^{d} \binom{n}{i} \approx d\log(n/d)$ queries are needed even by adaptive strategies. If defectives are rare, this expression simplifies to $d\log n$, subject to negligible terms. If d is known in advance, essentially $d\log n$ queries are also sufficient, and if d is unknown, still $1.5d\log n$ queries are sufficient [24]. There exist strategies using $O(d\log n)$ queries that need only two stages when d is known [8, 12, 3]. The query number in [3] tends to $1.44d\log n$ as d grows. A query number of the form $O(d\log n)$ cannot be achieved by any deterministic strategy in only one stage, as a consequence of known lower bounds for so called $\bar{d}$-separable pooling designs [2], which are exactly the sets of pools that can distinguish between any sets of at most d defectives. As opposed to this, there is a randomized one-stage strategy that succeeds with any prescribed probability, using asymptotically $1.45d\log n$ queries [3]. For the case of unknown d we proved in [6] that no deterministic strategy can manage with $O(d\log n)$ queries in any constant number of stages, but randomized strategies can, in only two stages.

For applications where defectives are rare but tests are expensive, it would be worthwhile to have strategies where the test number is as close as possible to the entropy lower bound, not only within some constant factor. *The principal contribution of the present paper is to show that, in fact, there exist randomized strategies that combine the two desirable features of query-optimality (at least in an asymptotic sense) and minimal adaptivity:* The constant factor in the leading term $d\log n$ of the query complexity tends to 1 as n grows, and the strategies work in a constant number of stages.

Although these results are not particularly hard to obtain, to our best knowledge this is the first paper presenting group testing strategies with this combination of desirable properties, and the way of combining known ingredients seems to be novel. The strategies also look simple enough for real use. Only elementary methods are needed to construct and analyze them.

We highlight the main results briefly, while the technical statements and also variations of the results are deferred to the following sections. If d grows slower than any power function of n, we achieve query-optimality already in 2 stages. Due to a recent negative result [22], this is not possible if d grows like $d = n^{\delta}$ for some constant exponent $\delta < 1$. But in this case we manage with 3 stages. Finally, if $d = o(n)$ we still get an asymptotically optimal query number in 4 stages. More precisely, we consider any fixed defective rate $r = d/n$ and show that our strategy approaches the entropy lower bound if $r \to 0$. This asymptotic behaviour matches known upper bounds for sequential group testing strategies, therefore one may appreciate that a constant number of stages suffices. It remains open whether even 3 stages would be enough.

An earlier negative result [6] implies that our strategies cannot be derandomized, but apparently they can be turned into deterministic strategies for the statistical model of group testing where elements are defective independently and with some fixed probability.

Due to space limitations, some parts of the proofs are only sketched, but in principle we include complete proofs. Some technicalities are omitted. For instance, when we use random subsets in a strategy, we do not always clearly distinguish between their expected and actual sizes, which however does not affect the asymptotic analysis for large n.

2 Minimal Adaptive Group Testing Close to the Entropy Lower Bound

The following observation is folklore; for completeness we give the proof.

Lemma 1. *If only one defective is present, it can be found by* $\log n + 1$ *queries in one stage.*

Proof. We introduce dummy elements if n is not a power of 2; this can at most double the number of elements. Then we index the elements as bit vectors of length $\log n + 1$. For each i we query a pool consisting of all elements that have entry 1 at the ith position. Obviously, the answers localize one defective provided that there is exactly one. □

Remark: This pooling design cannot check whether $d = 1$. A very minor issue is that we cannot see whether the element indexed by the zero vector is defective, if all pools were negative. Obviously we can catch up this case by one additional query (in the same stage) to this element. Much more importantly, if $d > 1$, it is possible that the strategy cannot safely identify any defective.

We also apply Theorem 10 from [3] that we rewrite as follows:

Lemma 2. *With prescribed probability* $1-\epsilon_1$ *one can correctly identify at most d defectives using* $O(d(\log n+\log(1/\epsilon_1)))$ *queries in one stage. The hidden constant factor is at most 1.9 and converges to 1.45 as d grows.* □

Theorem 1. *Using* $d \log n + O(d \log d) + O(d \log(1/\epsilon))$ *queries in two stages we can, with probability* $1 - \epsilon$*, correctly identify all defectives, provided that at most d defectives are present. The hidden constant factors in the lower-order terms are below* 3.8*, and tend to* 2.9 *as d grows.*

Proof. The overall scheme is very simple: In stage 1 we separate the defectives with probability $1 - \epsilon$, that is, we divide the elements into disjoint subsets each containing exactly one defective (plus one subset without defectives). In stage 2 we apply Lemma 1 to every such subset. It remains to discuss stage 1 in detail.

For some q to be specified below, we assign every element one of q labels, each with probability $1/q$. Elements with the same label form one cell. Like pools, a cell is said to be positive if it contains a defective, and otherwise negative. Then we apply Lemma 2 to the set of cells rather than individual elements: We can, with prescribed probability $1 - \epsilon_1$, correctly identify the (at most d) positive cells using $O(d(\log q + \log(1/\epsilon_1)))$ queries in one stage. The positive cells are our disjoint sets to be used in stage 2.

The probability that any two defectives collide, i.e., get into the same cell, is at most $\binom{d}{2}/q < d^2/2q$. In order to keep this probability below some ϵ_2 we choose $q = d^2/2\epsilon_2$, thus $\log q = 2\log d + \log(1/\epsilon_2) - 1$. The strategy gives an incorrect result with probability at most $\epsilon := \epsilon_1 + \epsilon_2$. Now, minimizing the query bound is equivalent to minimizing $\log(1/\epsilon_1) + \log(1/\epsilon_2)$ under the constraint $\epsilon = \epsilon_1 + \epsilon_2$. A standard calculation yields $\epsilon_1 = \epsilon_2 = \epsilon/2$, and obvious further manipulations give the final query bound. The constants follow from Lemma 2. □

Remarks:
(1) For every fixed d, this bound converges to the entropy lower bound as n grows. This asymptotic optimality holds even for d growing slowly with n (e.g., polylogarithmic). It remains open how many randomized queries would be actually needed in one stage. To our best knowledge, the current upper bound is the mentioned $1.45 d \log n$ from [3]. Is it possible to identify d defectives, with fixed probability $1 - \epsilon$, by essentially $d \log n$ queries in only one stage? Or can a non-trivial lower bound $a(\epsilon) d \log n$ for some $a(\epsilon) > 1$ be proved?

(2) The known deterministic two-stage strategies using $O(d \log n)$ queries, however with a constant strictly larger than 1, determine $O(d)$ candidate positives in stage 1, and need only $O(d)$ queries in stage 2 to test them [8, 12, 3]. Amazingly, in our randomized strategy the situation is exactly the opposite: The complexity of stage 1 does not depend on n, and the main work is done in stage 2. An interesting question is whether there exists a query-optimal two-stage strategy where the workload is balanced.

(3) The strategy in Theorem 1, with $q = \Theta(d^2)$, is designed for any constant failure probability. By choosing q as a larger polynomial in d, or even as a slow function of n, we can make the failure probability vanish asymptotically, without destroying the asymptotic query-optimality. Depending on the choice of q, different patterns of asymptotic behaviour can be achieved.

With one additional stage we can improve the query bound:

Theorem 2. *Using $d \log(n/d) + O(d\sqrt{\log d} \log\log d)$ queries in three stages we can, with probability $1 - \epsilon$, correctly identify all defectives, provided that at most d defectives are present.*

Proof. We give a high-level description of the strategy: Partition the elements randomly into d bags[1] of size n/d. Due to well-known load balancing results (see [20]), with high probability all bags contain fewer than $\log d$ defectives. We call a bag sparse/dense if it has fewer/more than $\sqrt{\log d}$ defectives. In stage 1

[1] We call them “bags” because their role is different from the “cells” used earlier.

we distinguish between sparse and dense bags using $L(n/d)$ queries in each bag, where L is any sublogarithmic function. It suffices to query random pools of size around $n/(d\sqrt{\log d})$ and decide sparse or dense based on the fraction of positive answers. We skip the details, since the only crucial point is that the pool number increases with n/d, thus we can make the error probability arbitrarily small. The rest is to apply the strategy from Theorem 1 in parallel to each bag. In the, up to d, sparse bags we may use $q = \Theta((\sqrt{\log d})^3)$, thus $O(d\sqrt{\log d}\log\log d)$ queries are needed in all sparse bags. In the, up to $d/\sqrt{\log d}$, dense bags we may use $q = \Theta((\log d)^3)$, thus $O((d/\sqrt{\log d})\log d\log\log d)$ queries are needed also in all dense bags. Here, exponent 3 in q is chosen to keep the failure probability in each bag $O(1/d)$. In the final stage we search for the separated defectives individually, among at most n/d elements. □

The advantage of Theorem 2 is that this complexity approaches the entropy lower bound for larger d, such as $d = n^\delta$, $\delta < 1$. Interestingly, it is known that two stages are not enough for that, due to a lower bound of $(\log e)^2 d\log(n/d)$ if d grows like $d = n^\delta$ [22]. (Actually, this result was derived for the statistical model of group testing with independent random defectives, but asymptotically the models are equivalent.)

Our next issue is that the outcome in Theorem 1 is correct with some prescribed probability, but in every specific case the searcher cannot be sure that the returned set of defectives is correct. Trivially, any group testing result can be verified in another stage with $d+1$ queries. But can we accomplish a correct *and* verified outcome without the extra stage? When determining the positive cells in stage 1 we may get some false positives as well. However, the subroutine from [3] never yields false negatives, and the false positive cells will be detected in stage 2. The real difficulty is that the separation can fail, too. More than one defective can get into one cell, and then the simple search as in Lemma 1 does not work; remember the remark after Lemma 1. However, with a slight increase of the test number we can also verify the outcome, as we will see below. First we need another search method for single defectives, known from [25]:

Lemma 3. *Using* $\log n + 0.5\log\log n + o(\log\log n)$ *queries in one stage, we can achieve the following: If only one defective is present, we identify it and confirm that it is the only one. If more defectives are present, we recognize this fact (but we do not necessarily identify some of the defectives in this case). Moreover, this query number is optimal for this purpose.* □

In order to make the paper more self-contained we briefly outline the strategy: The design consists of t pools, where each of the n elements is in exactly $t/2$ pools. Choose t even (or round $t/2$), and make t large enough so that $\binom{t}{t/2} = n$.

Along the lines of Theorem 1 it follows immediately:

Theorem 3. *Using* $d(\log n + 0.5\log\log n + o(\log\log n)) + O(d\log d) + O(d\log(1/\epsilon))$ *queries in two stages we can, with probability* $1-\epsilon$, *identify all defectives and verify that we found them all, provided that at most* d *defectives are present.* □

Note that the extra terms are $o(\log n)$, hence this result still matches asymptotically the entropy lower bound. Nevertheless it is interesting to ask if the $\log\log n$ term is avoidable. For two stages we must leave this as an open question. In three stages we can get rid of the $\log\log n$ term, by applying Lemma 1 and an obvious verification step.

In the following we give a side result related to that. It further extends the optimality statement from Lemma 3, in that it shows that one cannot even narrow down the candidates for the defective to a small set, in one stage with fewer queries. Group testing strategies that apply some pooling design in stage 1 and then test the candidates individually in stage 2 are well established as "trivial strategies" (which is perhaps a misleading name). They are of particular practical interest because no pools at all depend on test results and must be created on-the-fly: Stage 1 is prepared in advance, and only trivial testing is done in stage 2.

Theorem 4. *Suppose that actually only one defective is present (but the searcher is not sure about the number of defectives). With fewer than* $\log n + 0.5\log\log n - \Theta(\log c(n))$ *queries in one stage it is impossible to narrow down the candidate set for the defective to size* $c(n)$*. Here,* c *is any function with* $c(n) = o(\log n)$*.*

Proof. Consider any design of t pools, arbitrarily indexed $1,\ldots,t$. We define the indicator of an element to be the t-bit vector x where the ith position x_i is 1 if the element belongs to the ith pool, and $x_i = 0$ else. Imagine that an adversary declares one element defective, chosen randomly with probability $1/n$. For a t-bit vector x, let $p(x)$ denote the probability that the defective has indicator x. In other words, $p(x)\cdot n$ elements have indicator x.

For two t-bit vectors x and y, symbol $y \le x$ means that y is bitwise smaller than x, that is, $y_i = 1$ implies $x_i = 1$. If the defective has indicator x then exactly those elements with indicators $y \le x$ are candidates for being defective: Note that all pools i with $x_i = 1$ answered positively, and elements with indicators $y \le x$ occur in positive pools only, thus the searcher cannot surely recognize them as negative.

We conclude that the (conditional) expected number of candidates is now $n\sum_{y\le x} p(y)$ for any fixed indicator x of the defective, hence the expected number of candidates is $n\sum_{y\le x} p(x)p(y)$, where the sum is now taken over all such pairs (y,x). In order to get a lower bound for this expression, we choose the distribution $p(x)$ so as to minimize $n\sum_{y\le x} p(x)p(y)$ under the constraint $\sum_x p(x) = 1$.

In fact, this optimization problem is not hard to solve. Define the support of a distribution to be the set of all x with $p(x) > 0$. First we claim that, in some optimal solution, the support is an antichain (set of pairwise incomparable vectors) in the set of t-bit vectors partially ordered by $\le$. If the support A is not an antichain, take some minimal vector $y \in A$ that is smaller than some other vectors in A, and move $p(y)$ arbitrarily to these larger members of A. It is easy to see that this cannot increase our double sum (since no further "comparable pairs" are created). Hence we can repeat this manipulation until A is an antichain. But then our double sum simplifies to $n\sum_{x\in A} p(x)^2$. A sum of squares of numbers with fixed sum is minimized if all these numbers are equal. With $a := |A|$ we get $na(1/a)^2 = n/a$ expected candidates. In order to keep this number below c, we need $a \geq n/c$. Due to Sperner's Theorem [26, 21], the largest antichain in the partial order of t-bit vectors has size $a = \binom{t}{t/2}$. Thus, the known lower bound $\log a + 0.5 \log\log a$ for t yields the asserted lower bound in argument n.

To conclude, when the defective is chosen at random, then any deterministic strategy with fewer queries returns a candidate set whose expected size is not bounded by c. Hence there exists an element v such that, if v is the defective, more than c candidates actually remain. With Yao's technique (see, e.g., Section 2.2.2 in [23]), the same lower bound follows for randomized designs. □

The results so far were formulated for the case of a known d, or more realistically, a known upper bound d. With an additional stage using a procedure from [6] we get rid of this restrictive assumption. For this we need a slight adaptation of a result from [6] saying that $O(\log n)$ nonadaptive random queries are sufficient to find, with any fixed success probability, an upper bound $O(d)$ for d. The basic idea is to test random pools of exponentially growing size, and then the cut-off point between negative and positive pools gives an estimate of d. We remark that $\Omega(\log n)$ queries are also necessary, at least for some restricted but very natural type of random pools, as shown in [7].

Theorem 5. *For an arbitrarily small fixed $g > 0$ and for any fixed constant success probability $1 - \epsilon$, using $(d + g)\log n + O(d \log d)$ queries in three stages we can correctly identify all d defectives even without prior knowledge of d.*

Proof. In stage 1 is we use $g \log n$ pools to output an upper bound $O(d)$ for d, where the hidden constant in $O(d)$ depends only on g and on a prescribed failure probability (of underestimating d) [6]. Stage 2 and 3 consist of the strategy from Theorem 1, with the only modification that the number q of cells is chosen based on the upper bound for d returned by stage 1. Since this upper bound is $O(d)$, only the constant factor in the $O(d \log d)$ term is affected. □

Note that also this result gets arbitrarily close to the entropy lower bound as n grows. Moreover, factor $1 + g/d$ of the dominating term $d \log n$ can be bounded arbitrarily close to 1, uniformly for all d (by choosing g small enough), and for every fixed g it converges to 1 for growing d. An open question is whether we can accomplish the same characteristics as in Theorem 5 already in two stages. If d happens to be $o(\log n / \log\log n)$ (but the magnitude of d is still unknown in advance), we can actually manage this task in two stages, by applying the strategy from Theorem 1 or 3 with some $q = o((\log n / \log\log n)^2)$. But we conjecture that this is no longer possible for larger d.

3 Linear versus Sublinear Growth of the Defectives

The previous results hold for cases when d grows slower than n. However, in many practical settings one would rather expect a constant rate of defectives

$r := d/n$. In the following we also address this case. We assume r to be known in advance, otherwise we can first estimate r by $O(\log n)$ randomized nonadaptive group tests [6]. While the hidden constant depends on the accuracy of estimating r, the query number becomes negligible as n grows, since $\log n/(d \log(n/d)) = \log n/(nr \log(1/r))$ tends to 0.

This section consists of two parts. As a benchmark we first discuss adaptive, i.e., sequential testing. Then we show that 4 stages are enough to achieve a similar test complexity.

We call the model of group testing with a specified number d of defectives (which is either a known or a maximum number) the *combinatorial model.* In the *statistical model* of group testing, elements are defective independently and with some fixed probability r. When we have a strategy S for the statistical model and an input with d defectives, we may shuffle the elements and then apply strategy S for $r = d/n$. Since pools being significantly larger than $1/r$ are almost certainly positive and give little information, we can restrict pools to sizes $O(1/r)$ regardless of n. Thus, for large n the statistical model with probability r can be adopted instead of the combinatorial model with exactly d defectives. In the remainder of the section we assume the statistical model.

The entropy lower bound is now $r \log(1/r) + (1-r)\log(1/(1-r))$ queries per element, or equivalently,

$$\log(1/r) + (1-r)\log(1/(1-r))/r$$

queries per defective. This follows easily from the additivity of entropy. For small r this simplifies to $\log(1/r) + \log e$ queries per defective. It might be interesting to notice that this lower bound also holds for any randomized strategy that identifies d defectives in the combinatorial model, although an exact d means some more prior knowledge for the searcher. This follows from a more general fact (not referring especially to the group testing problem):

Proposition 1. *Let H be a set of h hypotheses, and suppose that a searcher can ask binary queries. Then no randomized strategy can guarantee to identify the correct hypothesis using an expected number of less than $\log h$ queries.*

Proof. Suppose that an adversary selects every hypothesis with probability $1/h$ as the true one. Then any deterministic strategy needs an expected number of at least $\log h$ queries, because every strategy can be viewed as a Huffman code with the expected query number as the average path length, and then the claim is easily seen from Huffman's algorithm [16] applied to the equal-probability case. From this, Yao's lower bound technique yields the assertion as follows. Any randomized strategy R can be seen as a probability distribution on the deterministic strategies. Hence the expected query number of R on the randomized input is at least $\log h$. It follows the existence of a specific hypothesis where R needs at least $\log h$ expected queries. □

In our case, this lower bound is $\log \binom{n}{d}$ and amounts to the same bound as before (with $-o(1)$ terms neglected), by routine calculations using Stirling's formula. Recall that we aim at strategies with an expected query number as close as possible to the lower bound. In a special type of sequential strategies, elements are arranged as a sequence, in any fixed linear order, and then they search for the leftmost defective by querying only pools that are prefixes of this sequence. This restriction leads to a well-studied problem from quality control [14, 15, 1]. Known results from there can be rephrased as follows.

Proposition 2. *The group testing problem with fixed rate r of defectives can be solved sequentially with $\log(1/r) + O(1) = (1 + o(1)) \log(1/r)$ queries per defective, where $o(1)$ vanishes for $r \to 0$.* □

The $o(1)$ term cannot be avoided, even in sequential strategies. Therefore it is interesting that this asymptotic behaviour, perhaps with an $o(1)$ term going slower to 0, can be achieved already in a small constant number of stages. For this result we can, in the following, focus attention on "small" r, which also allows us to neglect some technicalities like rounding. We stress that the announced result does not follow from the techniques of Section 2: Observe that we needed $d \log(n/d) + dM(d)$ queries, for some unbounded monotone function M. These are $\log(1/r) + M(d)$ queries per defective, that is, the additive term would grow infinitely with the input size even if r is fixed. In fact, we will need some more stages to avoid that.

Finally, as a preparation we reconsider one of the strategies in [3] and present a version that it guaranteed to find all defectives in two stages. Note that query numbers stated below are meant to be expected numbers.

Lemma 4. *In two stages using $1.9 \log(1/r) + 1$ queries per defective, where $r := d/n$, we can identify all d defectives.*

Proof. Query nonadaptively a sufficient number of random pools of size $1/r$, and discard the elements in negative pools. (The information in positive pools is not used further.) Every non-defective shall be kept with probability at most er, where e denotes Euler's number. If k denotes the number of negative pools, it is sufficient to have $(1 - 1/(rn))^k = er$. For large n this can be transformed into $e^{-k/rn} = er$, hence $k/rn = \ln(1/r) - 1$, which means $\ln(1/r) - 1$ negative pools per defective. Since a pool of size $1/r$ is negative with probability approximately $1/e$, this stage needs $e(\ln(1/r) - 1) = 1.9 \log(1/r) - e$ queries per defective. In a second stage, the $(1 + e)rn$ remaining candidates are tested individually, thus the total number of queries per defective is $1.9 \log(1/r) + 1$. □

Now we are ready for the main result. Basically it says that we can approach the entropy lower bound in 4 stages when $d = o(n)$.

Theorem 6. *Group testing with defective rate r can be solved in four stages using $(1 + o(1)) \log(1/r)$ queries per defective, where $o(1)$ vanishes for $r \to 0$.*

Proof. We split the elements in disjoint cells of x/r elements, where x is a free parameter. We choose x depending on r such that, $\lim_{r \to 0} x = 0$ but

$\lim_{r\to 0} x\log(1/r)=\infty$, which also implies $\lim_{r\to 0} x/r=\infty$. The expected number of defectives in a cell is x. Below we will use well-known inequalities like $1-x<e^{-x}<1-x+x^2/2$ and $e^x<1+x+x^2$ (for small x).

Remember that we are going to prove an asymptotic bound. Since $r\to 0$ but the cell size grows, the number of defectives in a cell follows, in the limit, a Poisson distribution with expectation x. (We omit a detailed technical discussion with error bounds.) In particular, we can assume that a cell has 0, 1, and more than 1 defective with probability e^{-x}, xe^{-x}, and $1-(1+x)e^{-x}$, respectively. We call these cells type 0, 1, and 2, respectively.

In stage 1 we simply query each cell, thus we recognize the type-0 cells, using $1/x$ queries per defective. In stage 2 we apply Lemma 3 to tell apart the type-1 and type-2 cells, and to find the unique defective in the type-1 cells. This needs

$$\log(x/r)+(0.5+o(1))\log\log(x/r)$$

queries in each type-1 or type-2 cell, and identifies an e^{-x} fraction of the defectives. Here, $o(1)$ denotes a term that vanishes for $x/r\to\infty$. For every type-1 cell there exist on average

$$(1-(1+x)e^{-x})/(xe^{-x})=(e^x-1-x)/x$$

type-2 cells, that is, $(e^x-1)/x<1+x$ type-1 *and* type-2 cells per type-1 cell.

Hence we have used fewer than

$$(1+x)\log(x/r)+(0.5+o(1))\log\log(x/r))+1/x$$

queries per recognized defective, in stage 1 and 2. For $x\to 0$ and $x/r\to\infty$ this simplifies to

$$(1+o(1))\log(x/r)+1/x<(1+o(1))\log(1/r)+1/x,$$

since $\log\log$ grows slower than log.

In stage 3 and 4 we merge all type-2 cells and find the remaining defectives using Lemma 4. They make up an $1-e^{-x}$ fraction of all defectives, and the total size of type-2 cells is $1-(1+x)e^{-x}$ times the original number of elements. Hence the defective rate is

$$r'=r(1-e^{-x})/(1-(1+x)e^{-x}).$$

Due to Lemma 4 we need $1.9\log(1/r')+1$ queries per defective from type-2 cells, which are

$$(1-e^{-x})(1.9\log(1/r')+1)<1.9x\log(1/r')+x$$

queries per defective. Furthermore we have

$$1/r'=(1-(1+x)e^{-x})/(1-e^{-x})\cdot(1/r).$$

This expression is smaller than

$$(1-(1+x)(1-x))/(1-(1-x+x^2/2))\cdot(1/r)=x/(1-x/2)\cdot(1/r).$$

Note that for $x \to 0$, the upper bound expression for $1/r'$ tends to x/r. Thus we have used

$$1.9x \log(x/r) + x < 1.9x \log(1/r) + x$$

queries per defective in stage 3 and 4.

The total number of queries per defective from all stages is still described by

$$(1 + o(1)) \log(1/r) + 1/x.$$

Since $\lim_{r \to 0} x \log(1/r) = \infty$, clearly $(1/x)/\log(1/r)$ tends to 0, thus the $1/x$ term is redundant. □

Acknowledgments

This work has been supported by the Swedish Research Council (Vetenskapsrådet), grant no. 2010-4661, "Generalized and fast search strategies for parameterized problems". Part of the work was inspired by the seminar "Search Methodologies II" (2010) organized by R. Ahlswede and F. Cicalese at the Center for Interdisciplinary Research, University of Bielefeld, Germany. We also thank the reviewers for their encouraging remarks and useful hints.

References

1. Ben-Gal, I.: An Upper Bound on the Weight-Balanced Testing Procedure with Multiple Testers. IIE Trans. 36, 481–493 (2004)
2. Chen, H.B., Hwang, F.K.: Exploring the Missing Link Among d-Separable, $\bar{d}$-Separable and d-Disjunct Matrices. Discr. Appl. Math. 155, 662–664 (2007)
3. Cheng, Y., Du, D.Z.: New Constructions of One- and Two-Stage Pooling Designs. J. Comp. Biol. 15, 195–205 (2008)
4. Clementi, A.E.F., Monti, A., Silvestri, R.: Selective Families, Superimposed Codes, and Broadcasting on Unknown Radio Networks. In: SODA 2001. pp. 709–718. ACM/SIAM (2001)
5. Cormode, G., Muthukrishnan, S.: What's Hot and What's Not: Tracking Most Frequent Items Dynamically. ACM Trans. Database Systems 30, 249–278 (2005)
6. Damaschke, P., Sheikh Muhammad, A.: Competitive Group Testing and Learning Hidden Vertex Covers with Minimum Adaptivity. Discr. Math. Algor. Appl. 2, 291–311 (2010)
7. Damaschke, P., Sheikh Muhammad, A.: Bounds for Nonadaptive Group Tests to Estimate the Amount of Defectives. In: Wu, W., Daescu, O. (eds.) COCOA 2010. LNCS, vol. 6509, pp. 117–130, Springer, Heidelberg (2010).
8. De Bonis, A., Gasieniec, L., Vaccaro, U.: Optimal Two-Stage Algorithms for Group Testing Problems. SIAM J. Comp. 34, 1253–1270 (2005)
9. De Bonis, A., Vaccaro, U.: Constructions of Generalized Superimposed Codes with Applications to Group Testing and Conflict Resolution in Multiple Access Channels. Theor. Comp. Sc. 306, 223–243 (2003)
10. Dorfman, R.: The Detection of Defective Members of Large Populations. The Annals of Math. Stat. 14, 436–440 (1943)

11. Du, D.Z., Hwang, F.K.: Pooling Designs and Nonadaptive Group Testing. Series on Appl. Math. vol. 18. World Scientific (2006)
12. Eppstein, D., Goodrich, M.T., Hirschberg, D.S.: Improved Combinatorial Group Testing Algorithms for Real-World Problem Sizes. SIAM J. Comp. 36, 1360–1375 (2007)
13. Goodrich, M.T., Hirschberg, D.S.: Improved Adaptive Group Testing Algorithms with Applications to Multiple Access Channels and Dead Sensor Diagnosis. J. Comb. Optim. 15, 95–121 (2008)
14. Hassin, R.: A Dichotomous Search for a Geometric Random Variable. Oper. Res. 32, 423–439 (1984)
15. He, Q.M., Gerchak, Y., Grosfeld-Nir, A.: Optimal Inspection Order When Process Failure Rate is Constant. Int. J. Reliability, Quality and Safety Eng. 3, 25–41 (1996)
16. Huffman, D.A.: A Method for the Construction of Minimum-Redundancy Codes. Proc. IRE 40, 1098–1101 (1952)
17. Iwen, M.A., Tewfik, A.H.: Adaptive Group Testing Strategies for Target Detection and Localization in Noisy Environments. IMA Preprint Series no. 2311, Univ. of Minnesota (2010)
18. Kahng, A.B., Reda, S.: New and Improved BIST Diagnosis Methods from Combinatorial Group Testing Theory. IEEE Trans. CAD of Integr. Circuits and Systems 25, 533–543 (2006)
19. Kainkaryam, R.M., Bruex, A., Gilbert, A.C., Schiefelbein, J., Woolf, P.J.: poolMC: Smart Pooling of mRNA Samples in Microarray Experiments. BMC Bioinf. 11, 299 (2010)
20. Kleinberg, J., Tardos, E.: Algorithm Design. Addison-Wesley, Boston (2006)
21. Lubell, D.: A Short Proof of Sperner's Lemma. J. Comb. Theory 1, 299 (1966)
22. Mézard, M., Toninelli, C.: Group Testing With Random Pools: Optimal Two-Stage Algorithms, IEEE Trans. Info. Th. 57, 1736–1745 (2011)
23. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge Univ. Press (1995)
24. Schlaghoff, J., Triesch, E.: Improved Results for Competitive Group Testing. Comb. Prob. and Comp. 14, 191–202 (2005)
25. Spencer, J.: Minimal Completely Separating Systems. J. Combin. Theory 8, 446–447 (1970)
26. Sperner, E.: Ein Satz über Untermengen einer endlichen Menge (in German). Math. Zeitschrift 27, 544–548 (1928)