Bounds for Nonadaptive Group Tests to Estimate the Amount of Defectives

Peter Damaschke and Azam Sheikh Muhammad

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

Abstract. The classical and well-studied group testing problem is to find d defectives in a set of n elements by group tests, which tell us for any chosen subset whether it contains defectives or not. Strategies are preferred that use both a small number of tests close to the information-theoretic lower bound $d \log n$, and a small constant number of stages, where tests in every stage are done in parallel, in order to save time. They should even work if d is completely unknown in advance. An essential ingredient of such competitive and minimal-adaptive group testing strategies is an estimate of d within a constant factor. More precisely, d shall be underestimated only with some given error probability, and overestimated only by a constant factor, called the competitive ratio. The latter problem is also interesting in its own right. It can be solved with $O(\log n)$ randomized group tests of a certain type. In this paper we prove that $\Omega(\log n)$ tests are really needed. The proof is based on an analysis of the influence of tests on the searcher's ability to distinguish between any two candidate numbers with a constant ratio. Once we know this lower bound, the next challenge is to get optimal constant factors in the $O(\log n)$ test number, depending on the desired error probability and competitive ratio. We give a method to derive upper bounds and conjecture that our particular strategy is already optimal.

Keywords: algorithm, learning by queries, competitive group testing, nonadaptive strategy, randomized strategy, lower bound

1 Introduction

Suppose that, in a set of n elements, d unknown elements are defective, and a searcher can do group tests which work as follows. She can take any subset of elements, called a pool, and ask whether the pool contains some defective. That is, the result of a group test is binary: 0 means that no defective is in the pool, and 1 means the presence of at least one defective. The combinatorial group testing problem asks to determine at most d defectives using a minimum number of tests; we also refer to them as queries. Group testing with its variants is a classical problem in combinatorial search, with a history dating back to year 1943 [8], and it has various applications in chemical testing, bioinformatics, communication

networks, information gathering, compression, streaming algorithms, etc., see for instance [3, 4, 7, 9, 11–13].

By the trivial information-theoretic lower bound, essentially $d \log_2 n$ queries are necessary for combinatorial group testing. A group testing strategy using $O(d \log n)$ queries despite ignorance of d before the testing process is called competitive, and the "hidden" constant factor is the competitive ratio. The currently best competitive ratio is 1.5 when queries are asked sequentially [14]. However, group testing strategies with minimal adaptivity are preferable for applications where the tests are time-consuming. Such strategies work in a few stages, where queries in a stage are prepared prior to the stage and then asked in parallel. For 1-stage group testing, at least $\Omega((d^2/\log d)\log n)$ queries are needed even in the case of a known d; see [1]. Clearly, 1-stage competitive group testing is impossible. As opposed to this, already 2 stages are enough to enable an $O(d \log n)$ test strategy, also the competitive ratio has been improved in several steps [6, 10, 2]. Still d must be known in advance or, to say it more accurately, d is some assumed upper bound on the true number of defectives. Apparently we were the first to study group testing strategies that are both minimal adaptive and competitive, i.e., they are suitable even when nothing about the magnitude of d is known beforehand [5]. Unfortunately, any efficient deterministic competitive group testing strategy needs $\Omega(\log d/\log\log d)$ stages (and $O(\log d)$ stages are sufficient). The picture changes when randomization is applied. If we can estimate an upper bound on the unknown d within a constant factor, using a logarithmic number of nonadaptive randomized queries, then we can subsequently apply any 2-stage $O(d \log n)$ strategy for known d, and thus obtain a randomized 3-stage competitive strategy. If we, instead, append a randomized 1-stage strategy with $O(d \log n)$ queries [2], we obtain a competitive group testing strategy that needs only 2 stages. Determining d exactly is as hard as combinatorial group testing itself [5], thus it would require $\Omega((d^2/\log d)\log n)$ nonadaptive queries. But an estimate of d within a constant factor is sufficient (and also necessary) for minimal adaptive competitive group testing. We call the expected ratio of our estimate and the true d a competitive ratio as well; it is always clear from context which competitive ratio is meant.

It is not hard to come up with such a nonadaptive estimator of d [5]. More precisely, using $O(\log n)$ queries we can output a number which is smaller than d only with some prescribed error probability ϵ but has an expectation $O(d)$. (If the alleged d was too small, the subsequent stages will notice the failure, and we try again from scratch, thus solving the combinatorial group testing problem in $O(1)$ expected stages.) To this end we prepare pools as follows. We fix some probability q and put every element in the pool independently with probability $1 - q$. Clearly, the group test gives the result 0 and 1 with probability q^d and $1 - q^d$, respectively. We prepare $O(\log n)$ of these pools such that the values $1/\log_2(1/q)$ form an exponential sequence of numbers between 1 to n. Note that these values are the defective numbers d for which $q^d = 1/2$. Then, the position in the sequence of pools where test results 0 switch to 1 hint to the value of d, subject to some constant factor and with some constant error probability. (Of

course, the details have to be specified and proved [5].) Note that the expected competitive ratio of 2-stage or 3-stage group testing is determined by three quantities: the competitive ratio of the group testing strategy used, and both the query number and competitive ratio of the randomized d estimator. The currently best 2-stage group testing strategy [2] uses $(1.44 + o(1))d \log n$ queries (asymptotically for $n \to \infty$). in this paper we focus on the estimator which requires methods completely different from the combinatorial group testing part that actually finds the defectives. Estimating d is also an interesting problem in its own right, as in some group testing applications we may only be interested in the amount of defectives rather than their identities.

An open question so far was whether $O(\log n)$ tests are really needed to estimate d, in the above sense. Intuitively this should be expected, based on the following heuristic argument. "Remote" queries with $1/\log_2(1/q)$ far from d will almost surely have a fixed result (0 or 1) and contribute little information about the precise location of d. Therefore we must have queries with values $1/\log_2(1/q)$ within some constant ratio of every possible d, which would imply an $\Omega(\log n)$ bound. However, the searcher may use the accumulated information from all queries, and even though "unexpected" results of the remote queries have low probabilities, a few such events might reveal enough useful information about d. Apparently, in order to turn the intuition into a proof we must somehow quantify the influence of remote queries and show that they actually provide too little information. To see the challenge, we first remark that the simple information-theoretic argument falls short. Imagine that we divide the interval from 1 to n into exponentially growing segments. Then the problem of estimating d up to a constant factor is in principle (don't care about technicalities) equivalent to guessing the segment where d is located, or a neighbored segment. The number of possible outcomes is some $\log n$, thus we need $\Omega(\log \log n)$ queries, which is a very weak lower bound. The next idea that comes into mind is to take the very different probabilities of binary answers into account. The entropy of the distribution of result strings is low, however it is not easy to see how to translate entropy into a measure suited to our problem.

A main result of the present paper is a proof of the $\Omega(\log n)$ query bound, for any fixed competitive ratio and any fixed error probability. A key ingredient is a suitable influence measure for queries. The proof is based on a simpler auxiliary problem that may deserve independent interest: deciding on one of two hypotheses about which we got only probabilistic information, thereby respecting a pair of error bounds. It has to be noticed that our result does not yet prove the non-existence of a randomized $o(\log n)$ query strategy in general. The result only refers to randomized pools constructed in the aforementioned simple way: adding every element to a pool independently with some fixed probability $1-q$. However, the result gives strong support to the conjecture that $\Omega(\log n)$ is also a lower bound for any other randomized pooling design. Intuitively, randomized pools that treat all elements symmetrically and make independent decisions destroy all possibilities for a malicious adversary to mislead the searcher by some clever

placement of defectives. Therefore it is hard to imagine that other constructions could have benefits.

The rest of the paper is organized as follows. In Section 2 we give a formal problem statement and some useful notation. In Section 3 we study a probabilistic inference problem on two hypotheses, and we define the influence of the random bit contributed by any query. This is used in Section 4 to prove the logarithmic lower bound for estimating the defective number by group tests. In Section 5 we derive a particular $O(\log n)$ query strategy for estimating the defectives, and we have reason to conjecture that its hidden constant factor is already optimal, for every input parameter. Section 6 concludes the paper.

2 Preliminaries

Motivated by competitive group testing we study the following abstract problem.

Problem 1: Given are positive integers n and L, some positive error probability $\epsilon < 1$, and some $c > 1$ that we call the competitive ratio. Furthermore, an "invisible" number $x \in [1, n]$ is given. A searcher can prepare L nonadaptive queries to an oracle as follows. A query specifies a number $q \in (0, 1)$, and the oracle answers 0 with probability q^x, and 1 with probability $1 - q^x$. Based on the string s of these L binary answers the searcher is supposed to output some number x' such that $\mathbf{Pr}[x' < x] \leq \epsilon$ and $\mathbf{E}[x'/x] \leq c$ holds for every x.

The actual problem is to place the L queries, and to compute an x' from s, in such a way that the demands are fulfilled. The optimization version asks to minimize c given the other input parameters. We will prove that $L = \Omega(\log n)$ queries are needed, for any fixed ϵ and c. Note that randomness is not only in the oracle answers but possibly also in the rule that decides on x' based on s, and even in the choice of queries.

Symbols **Pr** and **E** in the definition refer to the resulting probability distribution of x' given x. Note that no distribution of x is assumed, rather, the conditions shall be fulfilled for any fixed x. We might, of course, define similar problem versions, e.g., with two-sided errors or with worst-case (rather than expected) competitive ratio and tail probabilities. However we stick to the above problem formulation, as it came up in this form in competitive 2-stage and 3-stage group testing, and other conceivable variations would behave similarly. In the group testing context, an oracle query obviously represents a randomized pool where every element is selected independently with probability $1 - q$, and x is the unknown number of defectives. However we will treat x as a real-valued variable. Asymptotically this does not change anything, but it simplifies several technical issues.

It turns out that some coordinate transformations reflect the geometry of the problem better than the variables originating from the group testing application. We will look at x on the logarithmic axis and reserve symbol y for $y = \ln x$. Note that $y \in [0, \ln n]$. Furthermore, we relate every q to that value y which

would make $q^x = q^{e^y}$ some constant "medium" probability, such as $1/e$, the inverse of Euler's constant. (The choice of this constant is arbitrary, but it will simplify some expressions.) We denote this y value by t, in other words, we want $q^{e^t} = 1/e$, which means $q = e^{-e^{-t}}$ and $\ln(1/q) = e^{-t}$. Symbol t is reserved for this transformed q. We refer to t as a query point.

3 Probabilistic Inference of one-out-of-two Hypotheses

Buridan's ass could not decide on either a stack of hay or a pail of water and thus suffered from both hunger and thirst. The following problem demands a decision between two alternatives either of which could be wrong, but it also offers a clear rationale for the decision. As usual in inference problems, the term "target" refers to the true hypothesis.

Problem 2: The following items are given: two hypotheses g and h; two nonnegative real numbers $\epsilon, \delta < 1$; furthermore N possible observations that we simply denote by indices $s = 1, \ldots, N$; probabilities p_s to observe s if g is the target, and similarly, probabilities q_s to observe s if h is the target. Clearly, $\sum_{s=1}^{N} p_s = 1$ and $\sum_{s=1}^{N} q_s = 1$. Based on the observed s, the searcher can infer g with some probability x_s, and h with probability $1 - x_s$. The searcher's goal is to choose her x_s for all s, so as to limit to ϵ the probability of wrongly inferring h when g is the target, and to limit to δ the probability of wrongly inferring g when h is the target.

We rename the observations so that $p_1/q_1 \leq \ldots \leq p_N/q_N$.

In the optimization version of Problem 2, only one error probability, say ϵ, is fixed, and the searcher wants to determine $x_1, \ldots, x_N$ so as to minimize δ. We denote the optimum by $\delta(\epsilon)$. Problem 2 is easily solved in a greedy fashion:

Lemma 1. *A complete scheme of optimal strategies (one for every ϵ) for Problem 2 is described as follows.*[1] *Determine u such that $p_1 + \ldots + p_{u-1} \leq \epsilon < p_1 + \ldots + p_u$, and let $f := (\epsilon - p(1) - p(2) \ldots - p_{u-1})/p_u$. Infer h if $s < u$, infer h with probability f in case $s = u$, and otherwise infer g. Consequently, $\delta(\epsilon) = (1 - f)q_u + q_{u+1} + \ldots + q_N$.*

Proof. We only have to prove optimality. In any given strategy, let us change two consecutive "strategy values" simultaneously by $x_s := x_s - \Delta_s$ and $x_{s+1} := x_{s+1} + \Delta_{s+1}$, for some $\Delta_s, \Delta_{s+1} > 0$. If the target is g, this manipulation changes the probability to wrongly infer h by $p_{s+1}\Delta_{s+1} - p_s\Delta_s$. If the target is h, this manipulation changes the probability to wrongly infer g by $q_{s+1}\Delta_{s+1} - q_s\Delta_s$. We choose our changes so that the first term is zero, that is, $\Delta_s/\Delta_{s+1} = p_{s+1}/p_s$. Now $q_{s+1}/q_s \leq p_{s+1}/p_s = \Delta_s/\Delta_{s+1}$ shows $q_{s+1}\Delta_{s+1} - q_s\Delta_s \leq 0$, hence we only improved the strategy. The manipulation is impossible only if some index u exists with $x_s = 0$ for all $s < u$, and $x_s = 1$ for all $s > u$. Now the lemma follows easily. □

[1] Corrected version. The proceedings version has some variable mismatch here.

Lemma 1 also implies:

Corollary 1. *$\delta(\epsilon)$ is a monotone decreasing and convex (i.e., sub-additive), piecewise linear function with $\delta(0) = 1$ and $\delta(1) = 0$.* □

The following technical lemma shows that certain small additive changes in the probability sequences do not change the error function much (which is quite intuitive). In order to avoid heavy notation we give the proof in a geometric language, referring to a coordinate system with abscissa ϵ and ordinate δ.

Lemma 2. *Consider the following type of rearrangement of a given instance of Problem 2. Replace every p_s with $p_s - \rho_s$, where $\sum_s \rho_s = \rho$. Similarly, replace every q_s with $q_s - \tau_s$, where $\tau_s = \rho_s q_s/p_s$ and $\sum_s \tau_s = \tau$. Then add the removed probability masses, in total ρ and τ, arbitrarily to existing pairs (p_s, q_s) or create new pairs (p_s, q_s), but in such a way that $\sum_s p_s = 1$ and $\sum_s q_s = 1$ are recovered. If such a rearrangement reduces $\delta(\epsilon)$, then the decrease is at most τ.*

Proof. By Corollary 1, the curve of function $\delta(\epsilon)$ is a chain of straight line segments whose slopes $-\delta'(\epsilon)$ get smaller from left to right, and these slopes are the ratios q_s/p_s. The rearrangement has the following effect on the curve: Pieces of the segments are cut out, whose horizontal and vertical projections have total length ρ and τ, respectively. Then their horizontal and vertical lengths may increase again by re-insertions (and all these actions may change the slopes of existing segments), and possibly new segments are created. Finally all segments are assembled to a new chain connecting the points $\delta(0) = 1$ and $\delta(1) = 0$, and having a monotone sequence of slopes again.

Consider a fixed ϵ. Let ρ_0 and τ_0 be the total horizontal and vertical length, respectively, of the pieces cut out to the left of ϵ. Let ρ_1 and τ_1 be defined similarly for the pieces to the right of ϵ. The largest possible reduction of $\delta(\epsilon)$ appears if: (a) some new vertical piece of length τ_1 forms the left end, and (b) some new horizontal piece of length ρ_0 and forms the right end of the modified curve. Note that pieces in (a) were originally located below $\delta(\epsilon)$, and pieces in (b) were originally located to the left of ϵ. This moves the remainder of the original curve (a) down by τ_1 length units, and (b) to the left by ρ_0 length units. The vertical move (a) reduces $\delta(\epsilon)$ by τ_1. The horizontal move (b) causes that the new function value at ϵ is the old function value at $\epsilon + \rho_0$. Since the slopes decrease from left to right, the slope at our fixed ϵ (and to the right of it) can be at most τ_0/ρ_0. Thus, move (b) reduces $\delta(\epsilon)$ by at most $\rho_0\tau_0/\rho_0 = \tau_0$. Finally note that $\tau_1 + \tau_0 = \tau$. □

One should not be confused that ρ does not appear in the decrease bound: As we have chosen to consider δ as a function of ϵ, the setting is not symmetric.

We are particularly interested in the special case of Problem 2 where the $N = 2^L$ observations s are strings of L independent bits.

Problem 3: The following items are given: two hypotheses g and h; two non-negative real numbers $\epsilon, \delta \leq 1$; and 2^L possible observations described by binary

strings $s = s_1 \dots s_L$. Furthermore, for $k = 1, \dots, L$, we are given the probability a_k to observe $s_k = 0$ if the target is g, and the probability b_k to observe $s_k = 0$ if the target is h. The s_k are independent. The rest of the problem specification is as in Problem 2.

Clearly, our p_s and q_s evaluate to $p_s = \prod_{k=1}^{L}((1 - s_k)a_k + s_k(1 - a_k))$ and $q_s = \prod_{k=1}^{L}((1 - s_k)b_k + s_k(1 - b_k))$. Since the greedy algorithm in Lemma 1 applies also to Problem 3, a complete set of optimal strategies is described as follows: Infer h for p_s/q_s below some threshold, infer g for p_s/q_s above that threshold, and infer g or h randomized (with some prescribed probability) for p_s/q_s equal to that threshold.

Remark: Since the p_s and q_s are just products of certain probabilities a_k or $1 - a_k$, and b_k or $1 - b_k$, respectively, taking the logarithm reveals a nice and simple geometric structure of the optimal strategies from Lemma 1: Note that $\log(p_s/q_s) = \sum_{k=1}^{L}((1 - s_k)(\log a_k - \log b_k) + s_k(\log(1 - a_k) - \log(1 - b_k)))$. Since log is a monotone function, comparing the p_s/q_s with some threshold is equivalent to comparing the $\log(p_s/q_s)$ with some threshold. In other words, the decision for g or h is merely a linear threshold predicate. We will not need this remark in our lower-bound proof, still it might be interesting to notice.

In the following we consider any fixed $\epsilon > 0$, and all notations are understood with respect to this fixed error bound. Now think of our L independent bits as $L - 1$ bits plus a distinguished one, say the kth bit. We define the *influence* of this kth bit as the decrease of $\delta(\epsilon)$, that is, the difference to the $\delta(\epsilon)$ value accomplished by an optimal strategy when the kth bit is ignored. Trivially, $\delta(\epsilon)$ can only decrease when more information is available.

Lemma 3. *With the above notations for Problem 3, the influence of the kth bit is at most* $\min(\max(a_k, b_k), \max(1 - a_k, 1 - b_k))$.

Proof. The kth bit splits every old observation s, consisting of the $L - 1$ other bits and generated with probabilities p_s, q_s depending on the target, in two new observations. Their new probability pairs are obviously $(p_s a_k, q_s b_k)$ for $s_k = 0$, and $(p_s(1 - a_k), q_s(1 - b_k))$ for $s_k = 1$. In order to apply Lemma 2 we can view this splitting of observations as cutting out pieces from the segment of slope q_s/p_s of the $\delta(\epsilon)$ curve in the following way. If $q_s/p_s \leq b_k/a_k$, a piece of vertical length $q_s b_k$ is cut out. If $q_s/p_s > b_k/a_k$, a piece of horizontal length $p_s a_k$ is cut out, corresponding to a piece of vertical length $p_s a_k q_s/p_s = q_s a_k$. (Note that we must first "cut out enough length" in both directions, therefore this case distinction is needed.) This is done for all old s. Since, of course, the old q_s sum up to 1, we have $\tau \leq \max(a_k, b_k)$. The same reasoning applies to $1 - a_k, 1 - b_k$, thus we have $\tau \leq \max(1 - a_k, 1 - b_k)$ as well. □

Note that the influence bound in Lemma 3 is expressed only in terms of the probabilities of the respective bit being 0/1, conditional on the hypothesis. Hence we can independently apply Lemma 3 to each of the bits, no matter in

which order they are considered, and simply add the influence bounds of several bits (similarly to a union bound of probabilities).

4 The Logarithmic Lower Bound

We further narrow down our one-out-of-two inference problem to a special case of Problem 3. (Below we reuse symbol q, without risk of confusion.)

Problem 4: The following items are given: two hypotheses r and 1. where $r > 1$ is a fixed real number; two nonnegative real numbers $\epsilon, \delta \leq 1$, furthermore 2^L possible observations described by binary strings $s = s_1 \dots s_L$. For $k = 1, \dots, L$, let q_k^x be the probability to observe $s_k = 0$ if the target is x. We also speak of a "query at q_k". The s_k are independent. The rest of the problem specification is as before. In particular, let ϵ be the probability of wrongly inferring 1 although r is the target, and let δ be the probability of wrongly inferring r although 1 is the target.

Note that the hypothesis $x = r$ generates the string s with probability $\prod_{k=1}^{L}((1 - s_k)q_k^r + s_k(1 - q_k^r))$, and the hypothesis $x = 1$ generates s with probability $\prod_{k=1}^{L}((1 - s_k)q_k + s_k(1 - q_k))$, in other words, $a_k = q_k^r$ and $b_k = q_k$. As earlier we fix some error bound ϵ. From Lemma 3 we get immediately:

Lemma 4. *With the above notations for Problem 4, the influence of a query at q is at most* $\min(q, 1 - q^r)$. □

Problem 4 was stated, without loss of generality, for hypotheses r and 1. Similarly we may formulate it for hypotheses rx and x (for any positive x), which merely involves a coordinate transformation. We speak of the "influence of q on x" when we mean the influence of a query at q, with respect to Problem 4 for hypotheses rx and x. Clearly, the influence of q on x equals the influence of q^x on 1. Therefore Lemma 4 generalizes immediately to:

The influence of q on x is at most $\min(q^x, 1 - q^{rx})$.

Remember $y := \ln x$ from Section 2. By a slight abuse of notation, the phrase "influence of q on y" refers to the logarithmic coordinates, and Lemma 4 gets this form:

The influence of q on y is at most $\min(q^{e^y}, 1 - q^{re^y})$.

While q^{e^y} obviously decreases doubly exponentially with growing $y > 0$, it is also useful to have a simple upper bound for $1 - q^{re^y}$ when $y < 0$. Since $1 - e^{-z} \leq z$ for any variable z, we take z with $e^{-z} = q^{re^y}$ to obtain $1 - q^{re^y} \leq z = -\ln q^{re^y} = \ln(1/q)re^y$. Now we have:

The influence of q on y is at most $\min(q^{e^y}, \ln(1/q)re^y)$.

Finally we also transform q into t as introduced in Section 2, and we speak of the "influence of t on y", denoted $I_t(y)$. With $q = e^{-e^{-t}}$ and $\ln(1/q) = e^{-t}$, our influence lemma is in its final shape:

Lemma 5. $I_t(y) \le \min(e^{-e^{y-t}}, re^{y-t})$. □

From this bound we get:

Lemma 6. *For every fixed t we have $\int_0^{\ln n} I_t(y)\,dy = \Theta(\ln r)$.*

Proof. For simplicity we bound the integral over the entire real axis. (Since $I_t(y)$ decreases rapidly on both sides of t, this is not even too generous.) The advantage is that we can assume $t = 0$ without loss of generality. We split the integral in two parts, at $y = -\ln r$. As $I_t(y)$ is a minimum of two functions, we can take either of them as an upper bound. Specifically we get $\int_{-\infty}^{\infty} I_t(y)\,dy < \int_{-\infty}^{-\ln r} re^y\,dy + \int_{-\ln r}^{\infty} e^{-e^y}\,dy = \int_{\ln r}^{\infty} re^{-y}\,d(-y) + \int_{-\ln r}^{\infty} e^{-e^y}\,dy = re^{-\ln r} + \Theta(\ln r) = 1 + \Theta(\ln r)$. The second integral is $\Theta(\ln r)$ since both $e^{-e^{-\ln r}} = e^{-1/r}$ and (for instance) $e^{-e^0} = e^{-1}$ are between some positive constants, the function is monotone decreasing, and $\int_0^{\infty} e^{-e^y}\,dy = \Theta(1)$. □

The next lemma connects our "bipolar" number guessing problem to the problem we started from.

Lemma 7. *For every $r > 1$ and $0 < \delta < 1$ we have: Any strategy solving Problem 1 with error probability ϵ and competitive ratio $c := 1 + (r-1)\delta$ yields a strategy solving Problem 4 with hypotheses rx and x, for every $x \le n/r$, with error probabilities ϵ and δ.*

Proof. Imagine a searcher wants to solve an instance of Problem 1, and an adversary tells her that the target is either rx or x. Despite this strong help, in case that rx is the target, the searcher must still guess rx subject to an error probability ϵ, due to the definition of Problem 1. In the other case when the target is x, error probability δ means a competitive ratio of $(1-\delta) + r\delta = 1 + (r-1)\delta$. □

We are ready to state the main result of this section:

Theorem 1. *Any strategy for Problem 1, with fixed error probability ϵ and competitive ratio c, needs $\Omega(\ln n / \ln c)$ queries, where the constant factor may depend on ϵ.*

Proof. Fix some $r > c$ and $\delta = (c-1)/(r-1)$, hence $c = 1 + (r-1)\delta$. We choose $r = \Theta(c)$ large enough so that $D := 1 - \epsilon - \delta$ is positive. Due to Lemma 7, the set of queries must be powerful enough to solve Problem 4 with hypotheses rx and x, for every $x \le n/r$, with error probabilities ϵ and δ. In the case of no queries, the error tradeoff at every x would be simply $\delta(\epsilon) = 1 - \epsilon$. Since we need to reduce $\delta(\epsilon)$ down to our fixed δ, all queries together must have an influence at least $1 - \epsilon - \delta$ on x. In transformed coordinates this means $\sum_t I_t(y) \ge D$ for all $0 \le y \le \ln n - \ln r$, where the sum is taken over all t in our query set (multiple

occurrences counted). Hence $\int_0^{\ln n-\ln r} \sum_t I_t(y)\,dy \geq D(\ln n-\ln r)$. Since Lemma 6 states $\int_0^{\ln n-\ln r} I_t(y)\,dy = \Theta(\ln r)$ regardless of t, the number of queries is at least $(\ln n - \ln r)D/\Theta(\ln r) = \Omega(\ln n/\ln r)$. □

Note that this integration argument also applies if the queries themselves are located according to some probability distribution, that is, Theorem 1 also holds for "fully randomized" strategies.

Theorem 1 only shows that the query number is logarithmic, for any fixed parameter values. But the proof method is not suited for deriving also good lower bounds on the hidden constant factor. For instance, this factor should increase to infinity when ϵ tends to 0. To reflect this behaviour in the lower bound, apparently the previous proof must be combined with some reduction between problem instances with different ϵ. We leave this topic here. Anyways, in practice one would apply some reasonable standard value like $\epsilon = 0.05$ rather than trading much more queries for smaller failure probabilities. A more relevant question, addressed in the next section, is which upper bounds we can accomplish.

5 Translation-Invariant Strategies and Upper Bounds

Theorem 1 states that $L/\ln n$ in Problem 1 must be at least some constant, depending on ϵ and c. In order to get upper bounds on $L/\ln n$ we consider the following "infinite extension" of Problem 1. This has merely formal reasons that will be explained below.

Problem 5: Given are some positive error probability $\epsilon < 1$, some $c > 1$ that we call the competitive ratio, and an "invisible" number x which can be any real number. A searcher can prepare countably infinitely many nonadaptive queries to an oracle as follows. A query specifies a number $q \in (0,1)$, and the oracle gives answer 0 with probability q^x and answer 1 with probability $1-q^x$. Based on the infinite string s of the binary answers, the searcher is supposed to output some number x' such that $\mathbf{Pr}[x' < x] \leq \epsilon$ and $\mathbf{E}[x'/x] \leq c$ holds for every x.

For Problem 5 we naturally consider the *density* of queries, i.e., the number of queries per length unit on the logarithmic axis, corresponding to $L/\ln n$ in Problem 1. We withhold a precise formal definition of density, because for our upper bound we will only study a particular strategy for which the notion of density is straightforward:

Remember that $y = \ln x$, and every query, with probability q of responding with 0, is matched to a query point t on the logarithmic axis through $q = e^{-e^{-t}}$. If y is the unknown target value (in logarithmic coordinates), the probability of answer 0 to a query at point t is $q^x = e^{-e^{y-t}}$. The logarithmic lower bound in Theorem 1 and the influence argument in its proof suggests that query points t should be spread evenly over the logarithmic axis. More specifically, we consider strategies where the query points t are placed equidistantly, with space u between neighbored points. We place our queries at points $t = ju + v$, where u is fixed,

j loops over all integers, and v is a random shift being uniformly distributed, with $0 \le v < u$. For every two-sided infinite binary sequence s of answers we also specify an y_s such that the output $y' = \ln x'$ is located y_s length units to the right of the point of the leftmost answer 0 in s (see details below). We call such strategies *translation-invariant* with density u^{-1} because, obviously, all translations of the y-axis are automorphisms. One should not worry about the uncountably infinitely many s; in practice we "cut out a finite segment" of this infinite strategy according to:

Lemma 8. *Any translation-invariant strategy for Problem 5 with bounds ϵ and c and density u^{-1} yields a strategy for the original Problem 1 that has asymptotically, i.e., for $n \to \infty$, the same characteristics as the given strategy: error probability ϵ, competitive ratio c, and $u^{-1} \ln n$ queries.*

Proof. (sketch) We simply take the query points in the interval from 0 to $\ln n$, plus some margins on both sides, whose lengths grow with n but slower than $\ln n$. Since even the total influence of the (infinitely many!) ignored queries on any point y, $0 \le y \le \ln n$, decreases exponentially with the margin length, the resulting finite strategy performs as the original strategy for Problem 5, subject to vanishing terms. □

The reason for replacing Problem 1 with Problem 5 is its greater formal beauty. This way we skip some artificial treatment of the interval ends and obtain "clean" translation-invariance. In particular, in the calculations we can assume without loss of generality that $y = 0$, and the searcher does not know the shift of the coordinates (while in reality the searcher knows the coordinate system but not y). This will simplify the expressions a lot. Furthermore, the random shift v that we used to make our strategy translation-invariant does not sacrifice optimality: If, in any optimal strategy for Problem 5, the query points are first shifted randomly, the strategy remains optimal. To see this, simply note that the resulting strategy still respects the bounds ϵ and c at every y, if the original strategy did.

Next we show how to obtain the optimal values y_s for our specific strategy. For a given error probability and query density we want to minimize the competitive ratio. We need to consider only those two-sided infinite strings s that have a leftmost 0 and a rightmost 1. We call the segment bounded by these positions the significant segment. Clearly, all other response strings appear with total probability 0. We (arbitrarily) index the bits in each s so that $s_0 = 0$ is the leftmost 0, that is, $s_k = 1$ for all $k < 0$. The point on the y-axis where the leftmost query t with answer 0 is located is called the reference point.

The probability density of the event that string s appears, and its reference point is $ju + v$ (j integer, $0 \le v < u$), is given by

$$f_s(ju+v) := u^{-1} \prod_k \left((1-s_k) e^{-e^{-(k+j)u-v}} + s_k (1 - e^{-e^{-(k+j)u-v}}) \right)$$

where k loops over all integers, and the s_k are the bits of s as specified above.

Since, for each s, our strategy returns the point located y_s units to the right of the reference point t, the contribution of string s to the error probability (of having output $y' < 0$) amounts to $\int_{-\infty}^{-y_s} f_s(t)\,dt$. Hence our goal is to minimize $\sum_s \int_{-\infty}^{+\infty} e^{t+y_s} f_s(t)\,dt$ under the constraint $\sum_s \int_{-\infty}^{-y_s} f_s(t)\,dt \le \epsilon$. To summarize:

Proposition 1. *For any fixed u, the solution to the problem of minimizing $\sum_s \int_{-\infty}^{+\infty} e^{t+y_s} f_s(t)\,dt$ under the constraint $\sum_s \int_{-\infty}^{-y_s} f_s(t)\,dt \le \epsilon$ yields an upper bound on the competitive ratio c for Problem 1 when $u^{-1} \ln 2 \cdot \log_2 n$ queries are used.* □

Now these bounds can be calculated by standard nonlinear constraint optimization problem solvers. It suffices to consider some finite set of the most likely strings s whose sum of probabilities is close enough to 1. We implemented the method using the Matlab features `fmincon` for optimization and `quadgk` for numerical integration. As a little illustration, Table 1 displays the competitive ratios for $\epsilon = 0.01, \ldots 0.05$ and $g \log_2 n$ pools, for $g = 0.5$ and $g = 1$.

Table 1. Some competitive ratios c.

g	e 0.01	e 0.02	e 0.03	e 0.04	e 0.05
0.5	11.87	9.83	8.67	7.89	7.28
1.0	5.31	4.56	4.13	3.86	3.61

Of course, the optimizer also outputs the strategy variables y_s, here we do not show them due to limited space. For larger g it becomes harder to run the method in this form on a usual laptop computer. The denser the query points are, the more strings s have non-negligible probabilities, and the resulting large number of variables leads to slow convergence. However, these technical issues can be resolved by more computational power. One should also bear in mind that a strategy needs to be computed only once, for any given pair of input parameters g and ϵ, thus long waiting times might be acceptable. The only thing needed to apply the computed strategy is a look-up table of the y_s. Anyways, some optimality criterion for the problem could enable us to find the optimal strategies more efficiently than by this "naive" direct use of an optimizer.

For the original problem (of estimating the number d of defectives in an n-element set by group tests) we have also found and implemented an LP formulation. Clearly, the competitive ratios grow with n and should tend to the results for Problem 5 when $n \to \infty$. This behaviour is confirmed by our empirical results. Since our methods guarantee optimal competitive ratios for translation-invariant pooling designs, they improve upon the ad-hoc strategies in [5] where the problem was studied for the first time.

6 Open Questions

We studied the problem of estimating the number d of defective elements in a population of size n by randomized nonadaptive group tests, to within a constant factor c, and with a prescribed probability ϵ of underestimating d. A main result is that $\Omega(\log n)$ queries are needed, if the single pools are formed in a natural way by independent random choices. While this bound is intuitive, it has not been proved before, and quite some technical efforts were needed. It remains open how to show this lower bound also for arbitrary pools. A combination of our influence argument with Yao's lower bound technique may lead to an answer. The logarithmic lower bound also suggests that query points should be placed translation-invariant on the logarithmic axis; see details above. We gave such a strategy which allows numerical calculation of the output and competitive ratios, for any given query density and ϵ. One could also think of other translation-invariant strategies, for instance, query points may be chosen by a Poisson process, however this seems worse because then the density of actual query points can accidentally be low around the target value. In summary we conjecture that our strategy in Section 5 is already optimal, with respect to the constant factors and parameters, among all possible randomized strategies. But a proof (if it is true) would apparently require a different mathematical machinery. Disproving the conjecture would give interesting insights as well. Finally, the method proposed in Section 5 is a numerical one. A challenging question is whether the dependency of optimal competitive ratio, error probability and query number can be characterized in a closed analytical form.

Acknowledgments

This work has been supported by the Swedish Research Council (Vetenskapsrådet), grant no. 2007-6437, "Combinatorial inference algorithms – parameterization and clustering". We thank the referees for some helpful editorial remarks.

References

1. 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)
2. Cheng, Y., Du, D.Z.: New Constructions of One- and Two-Stage Pooling Designs. J. Comp. Biol. 15, 195–205 (2008)
3. 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)
4. Cormode, G., Muthukrishnan, S.: What's Hot and What's Not: Tracking Most Frequent Items Dynamically. ACM Trans. Database Systems 30, 249–278 (2005)
5. Damaschke, P., Sheikh Muhammad, A.: Competitive Group Testing and Learning Hidden Vertex Covers with Minimum Adaptivity. In: Kutylowski, M., Gebala, M., Charatonik, W. (eds.) FCT 2009. LNCS, vol. 5699, pp. 84–95. Springer, Heidelberg (2009). Extended version to appear in Discr. Math. Algor. Appl.

6. De Bonis, A., Gasieniec, L., Vaccaro, U.: Optimal Two-Stage Algorithms for Group Testing Problems. SIAM J. Comp. 34, 1253–1270 (2005)
7. 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)
8. Dorfman, R.: The Detection of Defective Members of Large Populations. The Annals of Math. Stat. 14, 436–440 (1943)
9. Du, D.Z., Hwang, F.K.: Pooling Designs and Nonadaptive Group Testing. World Scientific (2006)
10. 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)
11. Gilbert, A.C., Iwen, M.A., Strauss, M.J.: Group Testing and Sparse Signal Recovery. In: Asilomar Conf. on Signals, Systems, and Computers 2008. pp. 1059–1063. (2008)
12. 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)
13. 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)
14. Schlaghoff, J., Triesch, E.: Improved Results for Competitive Group Testing. Comb. Prob. and Comp. 14, 191–202 (2005)