Adaptive Group Testing Algorithms to Estimate the Number of Defectives

Nader H. Bshouty BSHOUTY@CS.TECHNION.AC.IL
Dept. of Computer Science
Technion, Haifa, 32000

Vivian E. Bshouty-Hurani BSHOUTY.VIVIAN@GMAIL.COM
The Arab Orthodox College
Haifa

George Haddad HADDADGEORGE9@GMAIL.COM
The Arab Orthodox College
11th Grade
Haifa

Thomas Hashem HASHEMTHOMAS1@GMAIL.COM
Sister of St. Joseph School
12th Grade
Nazareth

Fadi Khoury FADIKH2000@GMAIL.COM
Sisters of Nazareth High School
11th Grade
P.O.B. 9422, Haifa, 35661

Omar Sharafy OMARSHARAFY109@GMAIL.COM
The Arab Orthodox College
11th Grade
Haifa

Abstract

We study the problem of estimating the number of defective items in adaptive Group testing by using a minimum number of queries. We improve the existing algorithm and prove a lower bound that shows that, for constant estimation, the number of tests in our algorithm is optimal.

1. Introduction

Let X be a set of *items* with some *defective items* $I \subseteq X$. In Group testing, we *test (query)* a subset $Q \subset X$ of items. The answer to the query is 1 if Q contains at least one defective item, i.e., $Q \cap I \neq \emptyset$, and 0 otherwise. Group testing was originally introduced as a potential approach to the economical mass blood testing, (Dorfman, 1943). However it has been proven to be applicable in a variety of problems, including DNA library screening, (Ngo and Du, 1999), quality control in product testing, (Sobel and Groll, 1959), searching files in storage systems, (Kautz and Singleton, 1964), sequential screening of experimental variables, (Li, 1962), efficient contention resolution algorithms for multiple-access communication, (Kautz and Singleton, 1964; Wolf, 1985), data compression, (Hong and Ladner, 2002), and computation in the data stream model, (Cormode and Muthukrishnan, 2005). See a brief history and other applications in (Cicalese, 2013; Du and Hwang, 2000, 2006; Hwang, 1972; Macula and Popyack, 2004; Ngo and Du, 1999) and references therein.

Estimating the number of defective items $|I|$ up to a multiplicative factor of $1 \pm \epsilon$ is studied in (Cheng and Xu, 2014; Damaschke and Muhammad, 2010a,b; Falahatgar et al., 2016; Ron and Tsur, 2014). Estimating the number of defective items is an important problem in biological and medical applications (Chen and Swallow, 1990; Swallow, 1985). It is used for estimating the proportion of organisms capable of transmitting the aster-yellows virus in a natural population of leafhoppers (Thompson, 1962), estimating the infection rate of yellow-fever virus in a mosquito population (Walter et al., 1980) and estimating the prevalence of a rare disease using grouped samples to preserve individual anonymity (L.Gastwirth and A.Hammick, 1989).

In the *adaptive algorithm*, the tests can depend on the answers to the previous ones. In the *non-adaptive algorithm* they are independent of the previous ones and; therefore, one can do all the tests in one parallel step.

In this paper we study the problem of estimating the number of defective items $|I|$ up to a multiplicative factor of $1 \pm \epsilon$ with an adaptive Group testing algorithm. We first give new lower bounds and then give algorithms that improve the results from the literature. Our lower bounds show that our algorithms are optimal.

1.1. Previous and New Results

Let X be a set of n items with a set of defective items I. Estimating the number of defective items $|I| = d$ up to a multiplicative factor of $1 \pm \epsilon$ is studied in (Cheng and Xu, 2014; Damaschke and Muhammad, 2010a,b; Falahatgar et al., 2016; Ron and Tsur, 2014). The best algorithm is the algorithm of Falahatgar et al. (Falahatgar et al., 2016). Falahatgar et al. gave a randomized algorithm that asks $2 \log\log d + O((1/\epsilon^2)\log(1/\delta))$ *queries in expectation* and with probability at least $1 - \delta$ returns an estimation of d up to a multiplicative factor of $1 \pm \epsilon$. They also prove the lower bound $(1-\delta)\log\log d$. We show that by some modifications of their algorithm one can get the same result with $(1-\delta)\log\log d + O((1/\epsilon^2)\log(1/\delta))$ queries in expectation. We then give the lower bound $(1-\delta)\log\log d + (1/\epsilon)\log(1/\delta)$ for the number of queries. This shows that for constant ϵ, our algorithm is optimal.

Those randomized algorithms are not Monte Carlo. They may ask $\log\log n$ queries in the worst case (but with a small probability). We then study deterministic, randomized Las Vegas and randomized Monte Carlo algorithms for this problem. For randomized Monte Carlo algorithms we give the lower bound $\log\log d + (1/\epsilon)\log(1/\delta)$ and then give an algorithm that asks $\log^* n + \log\log d + O((1/\epsilon^2)\log(1/\delta))$ queries in expectation. Here,

$\log^* \alpha = 1$ for $\alpha \le 2$ and $\log^* n = 1 + \log^* \log n$. In particular, when $d > \log\log \overset{k}{\cdots} \log n$ for any constant k (or even $k = o(\log\log d)$), our algorithm asks $\log\log d + O((1/\epsilon^2)\log(1/\delta))$ queries in expectation. This, for constant ϵ, is optimal.

For deterministic and randomized Las Vegas algorithms we prove the lower bound $d\log((1-\epsilon)n/d)$ and then give a deterministic algorithm that asks a number of queries that matches the lower bound.

All the above algorithms run in linear time in n. The following table summarizes our results.

Adaptive Algorithm	Upper Bound	Lower Bound
Deterministic	$d\log\frac{(1-\epsilon)n}{d}$	$d\log\frac{(1-\epsilon)n}{d}$
Randomized Las Vegas	$d\log\frac{(1-\epsilon)n}{d}$	$d\log\frac{(1-\epsilon)n}{d}$
Randomized Monte Carlo	$\log^* n+$ $\log\log d + O\left(\frac{1}{\epsilon^2}\log\frac{1}{\delta}\right)$	$\log\log d + \Omega\left(\frac{1}{\epsilon}\log\frac{1}{\delta}\right)$
Randomized Monte Carlo With Expected #Queries	$(1-\delta)\log\log d+$ $O\left(\frac{1}{\epsilon^2}\log\frac{1}{\delta}\right)$	$(1-\delta)\log\log d+$ $\Omega\left(\frac{1}{\epsilon}\log\frac{1}{\delta}\right)$

All the algorithms in this paper are *adaptive*. That is, the tests can depend on the answers to the previous ones. For non-adaptive algorithms see the results in (Damaschke and Muhammad, 2010a,b). For an algorithm that determines exactly the number defective items see (Cheng, 2011). The best adaptive algorithm for *finding* the defective items asks $d\log(n/d) + O(d)$ queries (Cheng et al., 2014, 2015; Schlaghoff and Triesch, 2005). This query complexity meets the information lower bound for any deterministic or randomized algorithm.

2. Definitions and Preliminary Results

In this section we give some notations, definitions, the type of algorithms that are used in the literature and some preliminary results.

2.1. Notations and Definitions

Let $X = [n] := \{1, 2, 3, , \ldots, n\}$ be a set of *items* with some *defective* items $I \subseteq [n]$. In Group testing, we *query* a subset $Q \subseteq X$ of items and the answer to the query is $Q(I) := 1$ if Q contains at least one defective item, i.e., $Q \cap I \ne \emptyset$, and $Q(I) := 0$, otherwise.

Let $I \subseteq [n]$ be the set of defective items. Let $\mathcal{O}_I$ be an *oracle* that for a query $Q \subseteq [n]$ returns $Q(I)$. Let A be an algorithm that has access to the oracle $\mathcal{O}_I$. The output of the algorithm A for an oracle $\mathcal{O}_I$ is denoted by $A(\mathcal{O}_I)$. When the algorithm is randomized then we add the random seed r as an input to A and then the output of the algorithm is a random variable $A(\mathcal{O}_I, r)$ in $[n]$. Let A be a randomized algorithm and r_0 be a seed. We denote by $A(r_0)$ the deterministic algorithm that is equivalent to the algorithm A with the seed r_0. We denote by $Q(A, \mathcal{O}_I)$ (resp., $Q(A(r), \mathcal{O}_I)$) the set of queries that A asks with oracle $\mathcal{O}_I$ (resp., and a seed r). The algorithms we consider in this paper output $A(\mathcal{O}_I, r) \in [|I|(1-\epsilon), |I|(1+\epsilon)]$ where $[a, b] = \{\lceil a\rceil, \lceil a\rceil + 1, \cdots, \lfloor b\rfloor\}$. Such algorithms are called algorithms *that estimate the number of defective items* $|I|$ *up to a multiplicative factor of* $1 \pm \epsilon$.

2.2. Type of Algorithms

In this paper we consider four types of algorithms whose running time is polynomial in n.

1. The *deterministic* algorithm A with an oracle $\mathcal{O}_I$, $I \subseteq X$. The query complexity of a deterministic algorithm A is the *worst case complexity*, i.e, $\max_{|I|=d} |Q(A, \mathcal{O}_I)|$.

2. The randomized Las Vegas algorithm. We say that a randomized algorithm A is a *randomized Las Vegas algorithm that has expected query complexity* $g(d)$ if for any $I \subseteq X$, algorithm A with an oracle $\mathcal{O}_I$ asks at most $g(|I|)$ queries in expectation and with probability 1 outputs an integer in $[|I|(1-\epsilon), |I|(1+\epsilon)]$.

3. The randomized Monte Carlo algorithm. We say that a randomized algorithm A is a *randomized Monte Carlo algorithm that has query complexity* $g(d, \delta)$ if for any $I \subseteq X$, algorithm A with an oracle $\mathcal{O}_I$ asks at most $g(|I|, \delta)$ queries and with probability at least $1-\delta$ outputs an integer in $[|I|(1-\epsilon), |I|(1+\epsilon)]$.

4. The randomized algorithm. We say that a randomized algorithm A is a *randomized algorithm that has expected query complexity* $g(d, \delta)$ if for any $I \subseteq X$, algorithm A asks $g(|I|, \delta)$ queries in expectation and with probability at least $1-\delta$ outputs an integer in $[|I|(1-\epsilon), |I|(1+\epsilon)]$.

2.3. Preliminary Results

We now prove a few results that will be used throughout the paper

Let $s \in \cup_{i=0}^{\infty}\{0,1\}^i$ be a *string* over $\{0,1\}$ (including the empty string $\lambda \in \{0,1\}^0$). We denote by $|s|$ the *length* of s, i.e., the integer m such that $s \in \{0,1\}^m$. Let $s_1, s_2 \in \cup_{i=0}^{\infty}\{0,1\}^i$ be two strings over $\{0,1\}$ of lengths m_1 and m_2, respectively. We say that s_1 is a (proper) *prefix* of s_2 if $m_1 < m_2$ and $s_{1,i} = s_{2,i}$ for all $i = 1, \ldots, m_1$. We denote by $s_1 \cdot s_2$ the *concatenation* of the two strings s_1 and s_2.

The following result follows from the fact that the weighted path length of Huffman code is at least log the number of symbols. We give the proof for completeness.

Lemma 1 *Let $S = \{s_1, \ldots, s_N\}$ be a set of N distinct strings over $\{0,1\}$ such that no string is a prefix of another. Then, over the uniform distribution,*

$$\max_{s\in S} |s| \geq E(S) := \mathbf{E}_{s\in S}[|s|] \geq \log N.$$

Proof The proof is in Appendix A. ■

Lemma 2 *Let A be a deterministic adaptive algorithm that asks queries and outputs an element in $[n]$. Let $I, J \subseteq X$. If $A(\mathcal{O}_I) \neq A(\mathcal{O}_J)$ then there is $Q_0 \in Q(A, \mathcal{O}_I) \cap Q(A, \mathcal{O}_J)$ such that $Q_0(I) \neq Q_0(J)$.*

Proof Consider the sequence of queries $Q_{1,1}, Q_{1,2}, \cdots$ that A asks with the oracle $\mathcal{O}_I$ and the sequence of queries $Q_{2,1}, Q_{2,2}, \cdots$ that A asks with the oracle $\mathcal{O}_J$. Since A is deterministic, A asks the same queries as long as it gets the same answers to the queries. That is, if

$Q_{1,i}(I) = Q_{2,i}(J)$ for all $i \le \ell$ then $Q_{1,\ell+1} = Q_{2,\ell+1}$. Since $A(\mathcal{O}_I) \neq A(\mathcal{O}_J)$, there must be a query $Q_0 := Q_{1,t} = Q_{2,t}$ for which $Q_0(I) \neq Q_0(J)$. ■

Lemma 3 *Let A be a deterministic adaptive algorithm that asks queries. Let $C \subseteq 2^{[n]} := \{I | I \subseteq [n]\}$. If for every two distinct I_1 and I_2 in C there is a query $Q_0 \in Q(A, \mathcal{O}_{I_1})$ such that $Q_0(I_1) \neq Q_0(I_2)$ then*

$$\max_{I \in C} |Q(A, \mathcal{O}_I)| \ge \mathbf{E}_{I \in C}[|Q(A, \mathcal{O}_I)|] \ge \log |C|.$$

That is, the worst case query complexity and the average-case query complexity of A is at least $\log |C|$.

Proof For $I \in C$, consider the sequence of the queries that A with the oracle $\mathcal{O}_I$ asks and let $s(I) \in \cup_{i=0}^{\infty} \{0,1\}^i$ be the sequence of answers. The worst case query complexity and average-case query complexity of A are $s(C) := \max_{I \in C} |s(I)|$ and $\bar{s}(C) := \mathbf{E}_{I \in C}[|s(I)|]$, respectively, where $|s(I)|$ is the length of $s(I)$. We now show that for every two distinct I_1 and I_2 in C, $s(I_1) \neq s(I_2)$ and $s(I_1)$ is not a prefix of $s(I_2)$. This implies that $\{s(I) \mid I \in C\}$ contains $|C|$ distinct strings such that no string is a prefix of another. Then by Lemma 1, the result follows. Consider two distinct sets $I_1, I_2 \subseteq [n]$. There is a query $Q_0 \in Q(A, \mathcal{O}_{I_1})$ such that $Q_0(I_1) \neq Q_0(I_2)$. Consider the sequence of queries $Q_{1,1}, Q_{1,2}, \cdots$ that A asks with the oracle $\mathcal{O}_{I_1}$ and the sequence of queries $Q_{2,1}, Q_{2,2}, \cdots$ that A asks with the oracle $\mathcal{O}_{I_2}$. Since A is deterministic, A asks the same queries as long as it gets the same answers to the queries. That is, if $Q_{1,i}(I_1) = Q_{2,i}(I_2)$ for all $i \le \ell$ then $Q_{1,\ell+1} = Q_{2,\ell+1}$. Then, either we get in both sequences to the query Q_0 and then $Q_0(I_1) \neq Q_0(I_2)$ or some other query Q' that is asked before Q_0 satisfies $Q'(I_1) \neq Q'(I_2)$. In both cases $s(I_1) \neq s(I_2)$ and $s(I_1)$ is not a prefix of $s(I_2)$. ■

3. Lower Bounds

In this section we prove some lower bounds for the number of queries that are needed in order to estimate the number of defective items.

For deterministic algorithms we prove

Theorem 4 *Let A be a deterministic adaptive algorithm that estimates the number of defective items $|I| = d$ up to a multiplicative factor of $1 \pm \epsilon$. The query complexity of A is at least*

$$d \log \frac{(1-\epsilon)n}{d} - O(d).$$

In particular, for $\epsilon \le 1 - 1/n^{\lambda}$ where $0 < \lambda < 1$ is any constant, the problem of estimating the number of defective items with a deterministic adaptive algorithm is asymptotically equivalent to finding them.

Proof Consider the sequence of queries that A with an oracle $\mathcal{O}_I$ asks and let $s(I) \in \cup_{i=1}^{\infty} \{0,1\}^i$ be the string of answers. Consider the algorithm A with the oracles $\mathcal{O}_{I_1}$ and

$\mathcal{O}_{I_2}$ where I_1 and I_2 are any sets of sizes $|I_1| = d$ and $|I_2| \ge d' := (d+1)(1+\epsilon)/(1-\epsilon)$. For I_1, A outputs an integer D_1 where $(1-\epsilon)d \le D_1 \le (1+\epsilon)d$ and for I_2, A outputs an integer D_2 where $d(1+\epsilon)+(1+\epsilon) \le D_2$. Therefore, $D_1 \ne D_2$ and hence $s(I_1) \ne s(I_2)$. This shows that if $|I_1| = d$ and $s(I_1) = s(I_2)$ then $|I_2| \le d'-1$.

Now let $I' \subseteq X$ be any set of size d. Let $\mathcal{I}$ be the set of all sets $I \subset X$ of size d that have the same sequence of answers, i.e., $s(I) = s(I')$. Let $J = \cup_{I\in\mathcal{I}} I$. We now prove that $s(J) = s(I')$. Suppose for the contrary that this is not true. Then since $I' \subseteq J$ there is a query Q asked by A where $Q(J) = 1$ and $Q(I') = 0$. Therefore there is $j \in J\backslash I'$ such that $Q(j) = 1$ and $Q(I') = 0$. Since $j \in J$ there must be $I'' \in \mathcal{I}$ such that $j \in I''$ and then $Q(I'') = 1$. This is a contradiction to the fact that $s(I') = s(I'')$. Therefore $s(J) = s(I')$ and by the above argument we must have $|J| \le d'-1$. Since $\mathcal{I}$ contains subsets of J of size d, we have

$$|\mathcal{I}| \le L := \binom{d'-1}{d}.$$

This shows that each string in $\{s(I) : |I| = d\}$ corresponds to at most L sets of size d. Therefore $\{s(I) : |I| = d\}$ contains at least

$$M := \frac{\binom{n}{d}}{\binom{d'-1}{d}}$$

distinct strings and since the algorithm is deterministic no string is a prefix of another. By Lemma 1, the longest string is of length at least

$$C := \log M = \log \frac{\binom{n}{d}}{\binom{d'-1}{d}} \ge d\log\frac{n}{d} - d\log\left(\frac{1}{1-\epsilon}\right) - O(d).$$

Since the length of the longest string is the worst case query complexity of the deterministic algorithm the result follows. ■

For randomized Las Vegas algorithms we prove

Theorem 5 *Let A be a randomized Las Vegas adaptive algorithm that estimates the number of defective items $|I| = d$ up to a multiplicative factor of $1 \pm \epsilon$. The expected query complexity of A is at least*

$$d\log\frac{(1-\epsilon)n}{d} - O(d).$$

In particular, for $\epsilon \le 1 - 1/n^{\lambda}$ where $0 < \lambda < 1$ is any constant, the problem of estimating the number of defective items with a randomized Las Vegas adaptive algorithm is asymptotically equivalent to finding them.

Proof Let $X(I,r) = |Q(A(r), \mathcal{O}_I)|$ be a random variable of the number of queries that A asks with oracle $\mathcal{O}_I$ and let $g(d) = \max_{|I|=d} \mathbf{E}_r[X(I,r)]$ be the expected number of queries. Notice that for a fixed r, $A(r)$ is a deterministic algorithm. Consider $S_r = \{s_r(I) : |I| = d\}$ where $s_r(I)$ is the string of answers of the deterministic algorithm $A(r)$ with an oracle $\mathcal{O}_I$. Suppose $S_r = \{w_1, \ldots, w_t\}$ and $|w_1| \le |w_2| \le \cdots \le |w_t|$. Consider a partition $W_1 \cup W_2 \cup \cdots \cup W_t$ of the set of all sets of size d, where $W_i = \{I : |I| = d, s_r(I) = w_i\}$. As in the proof of Theorem 4, there are at least $t \ge M$ distinct strings in S_r. Also, no string is a prefix of other string because the algorithm is deterministic. Also, as in the proof of Theorem 4, for all i,

$$|W_i| \le \binom{d'-1}{d}.$$

Then, since $|w_1| \le |w_2| \le \cdots \le |w_t|$ and by Lemma 1,

$$\begin{aligned} \mathbf{E}_I[X(I,r)|r] &= \frac{\sum_{i=1}^{t} |W_i| \cdot |w_i|}{\binom{n}{d}} \\ &\ge \frac{\sum_{i=1}^{M} \binom{d'-1}{d} \cdot |w_i|}{\binom{n}{d}} \\ &= \frac{\sum_{i=1}^{M} |w_i|}{M} \ge \log M. \end{aligned}$$

Thus

$$\mathbf{E}_I[\mathbf{E}_r[X(I,r)]] = \mathbf{E}_r[\mathbf{E}_I[X(I,r)|r]] \ge \log M.$$

Therefore, there is I_0 such that $g(d) \ge \mathbf{E}_r[X(I_0,r)] \ge \log M$. ■

We now give two lower bounds for randomized Monte Carlo adaptive algorithms.

Theorem 6 *Let $0 < \epsilon < 1/2$ and $\min(\epsilon^\lambda, 1/2) \ge \delta \ge 1/(2(n - 1/\epsilon + 1))$ where $\lambda < 1$ is any constant. Let A be a randomized Monte Carlo adaptive algorithm that estimates the number of defective items up to a multiplicative factor of $1 \pm \epsilon$. Algorithm A must ask at least*

$$\Omega\left(\frac{1}{\epsilon}\log\frac{1}{\delta}\right)$$

queries.

Proof Let $A(r)$ be a randomized Monte Carlo adaptive algorithm that estimates the number of defective items $|I|$ up to a multiplicative factor of $1 \pm \epsilon$ where r is the random seed of the algorithm. Then for $|I| \in \{d, d+1\}$ where $d = \max(\lfloor 1/\epsilon \rfloor - 2, 1)$, it determines exactly $|I|$ with probability at least $1 - \delta$. Let $X(I,r)$ be a random variable that is equal to 1 if $A(\mathcal{O}_I, r) \ne |I|$ and 0 otherwise. Then for any $I \subseteq [n]$, $\mathbf{E}_r[X(I,r)] \le \delta$. Let $m = \lfloor 1/(2\delta) \rfloor + d - 1 \le n$. Consider any $J \subseteq [m]$, $|J| = d$. For any such J let

$$Y_J(r) = X(J,r) + \sum_{i \in [m] \backslash J} X(J \cup \{i\}, r).$$

Then for every $J \subseteq [m]$ of size d, $\mathbf{E}_r\left[Y_J(r)\right] \le (m - d + 1)\delta \le \frac{1}{2}$. Therefore for a random uniform $J \subseteq [m]$ of size d we have $\mathbf{E}_r[\mathbf{E}_J[Y_J(r)]] = \mathbf{E}_J[\mathbf{E}_r[Y_J(r)]] \le 1/2$. Thus, there is r_0 such that for at least half of the sets $J \subseteq [m]$, of size d, $Y_J(r_0) = 0$. Let C be the set of all $J \subseteq [m]$, of size d, such that $Y_J(r_0) = 0$. Then

$$|C| \ge \frac{1}{2}\binom{m}{d} = \frac{1}{2}\binom{\lfloor 1/(2\delta) \rfloor + d - 1}{d}.$$

Consider the deterministic algorithm $A(r_0)$. We claim that for every two distinct $J_1, J_2 \in C$, there is a query $Q \in Q(A(r_0), \mathcal{O}_{J_1})$ such that $Q(J_1) \neq Q(J_2)$. If this is true then, by Lemma 3, the query complexity of $A(r_0)$ is at least

$$\log|C| \geq \log\frac{1}{2}\binom{\lfloor 1/(2\delta)\rfloor + d - 1}{d} \geq d\log\frac{1}{2d\delta} - 1 = \Omega\left(\frac{1}{\epsilon}\log\frac{1}{\delta}\right).$$

We now prove the claim. Consider two distinct $J_1, J_2 \in C$. There is w.l.o.g $j \in J_2 \backslash J_1$. Since $Y_{J_1}(r_0) = 0$ we have $X(J_1, r_0) = 0$ and $X(J_1 \cup \{j\}, r_0) = 0$ and therefore $A(\mathcal{O}_{J_1}, r_0) = d$ and $A(\mathcal{O}_{J_1 \cup \{j\}}, r_0) = d + 1$. Thus, by Lemma 2, there is a query $Q_0 \in Q(A(r_0), \mathcal{O}_{J_1}) \cap Q(A(r_0), \mathcal{O}_{J_1 \cup \{j\}})$ for which $Q_0(J_1) = 0$ and $Q_0(J_1 \cup \{j\}) = 1$. Therefore $Q_0(\{j\}) = 1$ and then $Q_0(J_1) = 0$ and $Q_0(J_2) = 1$. ■

The following is the second lower bound for randomized Monte Carlo adaptive algorithms.

Theorem 7 *Let A be a randomized Monte Carlo adaptive algorithm that estimates the number of defective items $|I| = d$ up to a multiplicative factor of $1/4$ with probability at least $1 - \delta > 1/2$. The query complexity of A is at least*

$$\log\log d - 1$$

Proof Let A be a randomized Monte Carlo algorithm that estimates $|I| = d$ up to a multiplicative factor of $1/4$ with probability at least $1 - \delta$. Let $X(I, r)$ be a random variable where $X(I, r) = 0$ if $A(\mathcal{O}_I, r) \in [(3/4)|I|, (5/4)|I|]$ and 1 otherwise. Then $\mathbf{E}_r[X(I, r)] \leq \delta$. Now for a random uniform integer $2^j \in [d]$ we have $\mathbf{E}_r[\mathbf{E}_j[X([2^j], r)]] = \mathbf{E}_j[\mathbf{E}_r[X([2^j], r)]] \leq \delta$. Therefore, there is a seed r_0 such that $\mathbf{E}_j[X([2^j], r_0)] \leq \delta$. This implies that for at least $t := (1 - \delta)(\log d)$ integers $J := \{2^{j_1}, \ldots, 2^{j_t}\} \subseteq [d]$ the deterministic algorithm $A(r_0)$ determines exactly $|I|$ provided that $|I| \in J$. Therefore, as in the above proofs, $A(r_0)$ asks at least

$$\log t = \log\log d + \log(1 - \delta) \geq \log\log d - 1 \tag{1}$$

queries. ■

We now consider randomized algorithms with success probability at least $1 - \delta$ and $g(|I|, \delta)$ expected queries. In (Falahatgar et al., 2016), Falahatgar et al. gave the following lower bound for $g(d, \delta)$. We give another simple proof in the Appendix A for a slightly weaker lower bound.

Theorem 8 *Let A be a randomized adaptive algorithm that estimates the number of defective items $|I| = d$ up to a multiplicative factor of $1/2$ with probability at least $1 - \delta$. The expected number of queries of A is at least*

$$(1 - \delta)\log\log d$$

Similar to the above techniques we prove in the Appendix A

Theorem 9 *Let $\epsilon^\lambda \ge \delta \ge 1/(2(n-1/\epsilon+1))$ where $\lambda < 1$ is any constant. Let A be a randomized adaptive algorithm that estimates the number of defective items up to a multiplicative factor of $1 \pm \epsilon$. The expected number of queries of A is at least*

$$\Omega\left(\frac{1}{\epsilon}\log\frac{1}{\delta}\right).$$

4. Upper Bounds

In this section we prove some upper bounds.

The following result will be used in this section.

Lemma 10 *(Cheng et al., 2014, 2015; Schlaghoff and Triesch, 2005) There is a deterministic adaptive algorithm,* **Find -Defectives***, that without knowing d, asks $d\log(n/d)+O(d)$ queries and finds the defective items.*

In the next Theorem we give a deterministic algorithm with a query complexity that matches the lower bound in Theorem 4. The time complexity of this algorithm is $O(qn)$ where q is the number of queries.

Theorem 11 *There is a deterministic adaptive algorithm that estimates the number of defective items $|I| = d$ up to a multiplicative factor of $1 \pm \epsilon$ and asks*

$$d\log\frac{(1-\epsilon)n}{d}+O(d)$$

queries.

Proof The algorithm divides the set of items $X=[n]$ into $N=(1-\epsilon)n$ disjoint sets $X_1,\ldots,X_N$ where each set X_i contains $1/(1-\epsilon)$ items. It then runs the algorithm **Find-Defectives** in Lemma 10 with N items. For each query $Q \subseteq [N]$ in **Find-Defectives**, the algorithm asks the query $Q' = \cup_{i\in Q}X_i$. By Lemma 10, the number of queries is

$$d\log\frac{N}{d}+O(d) = d\log\frac{(1-\epsilon)n}{d}+O(d).$$

Now since the d defective items can appear in at most d sets X_i and at least $(1-\epsilon)d$ sets, the output of the algorithm is D that satisfies $(1-\epsilon)d \le D \le d$. ■

We now give a randomized algorithm such that, for any constant ϵ, its expected number of queries matches the lower bound in Theorem 8 and 6.

Theorem 12 *For any constant $c>1$, there is a randomized algorithm that asks*

$$q = (1-\delta+\delta^c)\log\log d + O(\sqrt{\log\log d}) + O\left(\frac{1}{\epsilon^2}\log\frac{1}{\delta}\right)$$

expected number of queries and with probability at least $1-\delta$ estimates the number of defective items d up to a multiplicative factor of $1 \pm \epsilon$.

Proof We first give an algorithm A that asks

$$q'(\delta) := \log\log d + O(\sqrt{\log\log d}) + O\left(\frac{1}{\epsilon^2}\log\frac{1}{\delta}\right)$$

queries in expectation. We then define the following algorithm B: With probability $\delta - \delta^c$ output 0 and with probability $1-(\delta-\delta^c)$ run algorithm A with success probability of $1-\delta^c$.

The expected number of queries that B asks is $(1-\delta+\delta^c)q'(\delta^c) = q$ and the success probability is $1-\delta$.

We now give algorithm A. Algorithm A is the same as the algorithm of Falahatgar et al. (Falahatgar et al., 2016) but with different parameters. Their algorithm runs in 4 stages. In the first stage they give a procedure $\text{A}_{\text{FACTOR}-d}$ that finds an integer D_1 that with probability at least $1-\delta$ satisfies $d \le D_1 \le 2d^2\frac{1}{\delta^2}\log\frac{1}{\delta}$. Procedure $\text{A}_{\text{FACTOR}-d}$ for $i = 1, 2, \cdots$, generates random queries Q_i where each $j \in [n]$ is in Q_i with probability $1-2^{-1/\Delta_i}$ and is not in Q_i with probability $2^{-1/\Delta_i}$ where $\Delta_i = 2^{2^i}$. It then asks the queries Q_i for $i = 1, 2, \cdots$ and halts on the first query Q_{i_0} that gets the answer 0. Then, it outputs $D_1 = 2\Delta_{i_0}\log\frac{1}{\delta}$.

Our procedure $\text{IMPROVEDA}_{\text{FACTOR}-d}$ finds an integer D_1' that with probability at least $1-\delta$ satisfies

$$d \le D_1' \le 2\left(\frac{2d}{\delta}\right)^{2^{2\sqrt{\log\log\frac{2d}{\delta}}+1}}\log\frac{1}{\delta}.$$

Procedure $\text{IMPROVEDA}_{\text{FACTOR}-d}$ for $i = 1, 2, \cdots$, generates random queries Q_i' where each $j \in [n]$ is in Q_i' with probability $1-2^{1/\Delta_i'}$ where $\Delta_i' = 2^{2^{i^2}}$, asks the queries Q_i' and halts on the first query Q_{i_0}' that gets answer 0. Then, it outputs $D_1' = 2\Delta_{i_0}\log\frac{1}{\delta}$. The expected number of queries in $\text{IMPROVEDA}_{\text{FACTOR}-d}$ is

$$\sqrt{\log\log D_1'} = O\left(\sqrt{\log\log\frac{d}{\delta}}\right). \qquad (2)$$

The proof of correctness and the query complexity analysis is the same as in (Falahatgar et al., 2016) and is sketched in the next subsection for completeness.

The second stage of the algorithm by Falahatgar et al. is the procedure $\text{A}_{\text{FACTOR}-1/\delta^2}$. The procedure $\text{A}_{\text{FACTOR}-1/\delta^2}$ is a binary search for $\log d$ in the logarithmic scale of the interval $[1, D_1]$ - that is, in $[0, \log D_1]$. The procedure with probability at least $1-\delta$ returns D_2 such that $\delta^2 d \le D_2 \le d/\delta^2$. The expected number of queries is $\log\log D_1 = \log\log\frac{d}{\delta} + O(\log\log\log(1/\delta))$. The same procedure with the same analysis and proof of correctness works as well in our algorithm for the interval $[0, \log D_1']$. The procedure $\text{A}_{\text{FACTOR}-1/\delta^2}$, with probability at least $1-\delta$, returns D_2' such that $\delta^2 d \le D_2' \le d/\delta^2$. The expected number of queries is

$$\log\log D_1' = \log\log\frac{d}{\delta} + O\left(\sqrt{\log\log\frac{d}{\delta}}\right). \qquad (3)$$

The third and fourth stage in (Falahatgar et al., 2016) (and here), are two procedures that with an input D_2', with probability at least $1-\delta$, estimates the number of defective items d up to a multiplicative factor of $1 \pm \epsilon$ with $O((1/\epsilon^2)\log(1/\delta))$ expected number of queries.

The expected number of queries is the sum of expressions in (2), (3) and $O((1/\epsilon^2)\log(1/\delta))$ which is equal to $q'(\delta)$. ■

We note here that the best constant in the $O(\sqrt{\log\log d})$ is $2\sqrt{2}=2.828$ and can be obtained by the sequence $\Delta_i = 2^{2^{i^2/2}}$.

4.1. Analysis of the Algorithm

The following result is immediate. We omitted the proof.

Lemma 13 *Let Q_Δ be a random query where each $j\in[n]$ is in Q_Δ with probability $1-2^{-1/\Delta}$ and is not in Q_Δ with probability $2^{-1/\Delta}$. Let $I\subseteq[n]$ be a set of defective items of size d. Then for any Δ we have*

$$\Pr[Q_\Delta(I)=0]=2^{-\frac{d}{\Delta}}$$

and for $\Delta>d$,

$$\Pr[Q_\Delta(I)=1]\le\frac{d}{\Delta}.$$

Now, let $\{\Delta_i\}_{i=1}^\infty$ be any sequence of numbers such that, $\Delta_1\ge 1$ and $\Delta_{i+1}/\Delta_i\ge 2$. Consider the algorithm that asks the query Q_{Δ_i} for $i=1,2,3,\ldots$ and stops on the first query $Q_{\Delta_{i_0}}$ that gets answer 0. Let

$$D=2\Delta_{i_0}\log\frac{2}{\delta}.$$

Since $\Delta_{i-1}\le\Delta_i/2$ and by Lemma 13,

$$\begin{aligned}\Pr[D<d] &= \Pr\left[\Delta_{i_0}<\frac{d}{2\log(2/\delta)}\right]\\ &\le \sum_{i:\Delta_i<d/(2\log(2/\delta))}\Pr[Q_{\Delta_i}(I)=0]\\ &= \sum_{i:\Delta_i<d/(2\log(2/\delta))}2^{-d/\Delta_i}\le\delta/2.\end{aligned}$$

Let i_1 be such that $\Delta_{i_1-1}\le 2d/\delta<\Delta_{i_1}$. Then, by Lemma 13,

$$\begin{aligned}\Pr\left[D>2\Delta_{i_1}\log\frac{2}{\delta}\right] &= \Pr[\Delta_{i_0}>\Delta_{i_1}]\\ &\le \Pr[Q_{\Delta_{i_1}}(I)=1]\\ &\le \frac{d}{\Delta_{i_1}}\le\delta/2.\end{aligned}$$

Since, $\Delta_{i+1}/\Delta_i\ge 2$, we have

$$\Pr[\Delta_{i_0}>\Delta_{i_1+k}]\le\frac{d}{\Delta_{i_1+k}}\le\frac{\delta}{2^{k+1}},$$

and therefore the expected number of queries is at most i_1+2.

This proves

Lemma 14 *Let $\{\Delta_i\}_{i=1}^{\infty}$ be any sequence of numbers such that, $\Delta_1 \geq 1$ and $\Delta_{i+1}/\Delta_i \geq 2$. Let i_1 be such that $\Delta_{i_1-1} \leq 2d/\delta < \Delta_{i_1}$. The above algorithm asks at most $i_1 + 2$ queries in expectation and with probability at least $1-\delta$ outputs D that satisfies $D \geq d$ and $D \leq 2\Delta_{i_1}\log(2/\delta)$.*

Suppose we know some upper bound D^* on d. Let i_2 be such that $\Delta_{i_2} > D^*$. The algorithm is also a Monte Carlo algorithm that asks at most i_2 queries.

Now if we take $\Delta_i = 2^{2^{i^2}}$ then $i_1 \leq \sqrt{\log\log(2d/\delta)} + 1$ and

$$\Delta_{i_1} \leq \left(\frac{2d}{\delta}\right)^{2^{2\sqrt{\log\log\frac{2d}{\delta}}+1}}.$$

Therefore

$$d \leq D \leq 2\left(\frac{2d}{\delta}\right)^{2^{2\sqrt{\log\log\frac{2d}{\delta}}+1}} \log\frac{2}{\delta}.$$

This gives the result in Theorem 12.

The randomized Monte Carlo algorithm is in Appendix B

5. Open Problems

The results in the table in Subsection 1.1 suggest the following open problems:

1. Prove a lower bound $\Omega((1/\epsilon^2)\log(1/\delta))$ or find an randomized algorithm that asks $(1-\delta)\log\log d + O((1/\epsilon)\log(1/\delta))$ queries in expectation. We note here that the lower bound obtained by the KL-divergence of an ϵ-bias coin doesn't seem to work for this problem. First, because the answers of the oracle depend on the queries that is known to the algorithm and second because the algorithms we consider here are adaptive so the distribution of the answers can somehow be controlled by the algorithm.

2. Prove the lower bound $\Omega(d)$ for number of queries in any randomized Monte Carlo algorithm when $n \to \infty$. A randomized Monte Carlo algorithm that asks $O(d\log d + d\log(1/\delta))$ queries follows from (Cheng, 2011).

References

Chao L. Chen and William H. Swallow. Using group testing to estimate a proportion, and to test the binomial model. *Biometrics.*, 46(4):1035–1046, 1990.

Yongxi Cheng. An efficient randomized group testing procedure to determine the number of defectives. *Oper. Res. Lett.*, 39(5):352–354, 2011. doi: 10.1016/j.orl.2011.07.001. URL `https://doi.org/10.1016/j.orl.2011.07.001`.

Yongxi Cheng and Yinfeng Xu. An efficient FPRAS type group testing procedure to approximate the number of defectives. *J. Comb. Optim.*, 27(2):302–314, 2014. doi: 10.1007/s10878-012-9516-5. URL `https://doi.org/10.1007/s10878-012-9516-5`.

Yongxi Cheng, Ding-Zhu Du, and Yinfeng Xu. A zig-zag approach for competitive group testing. *INFORMS Journal on Computing*, 26(4):677–689, 2014. doi: 10.1287/ijoc.2014.0591. URL https://doi.org/10.1287/ijoc.2014.0591.

Yongxi Cheng, Ding-Zhu Du, and Feifeng Zheng. A new strongly competitive group testing algorithm with small sequentiality. *Annals OR*, 229(1):265–286, 2015. doi: 10.1007/s10479-014-1766-4. URL https://doi.org/10.1007/s10479-014-1766-4.

Ferdinando Cicalese. *Fault-Tolerant Search Algorithms - Reliable Computation with Unreliable Information.* Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2013. ISBN 978-3-642-17326-4. doi: 10.1007/978-3-642-17327-1. URL https://doi.org/10.1007/978-3-642-17327-1.

Graham Cormode and S. Muthukrishnan. What's hot and what's not: tracking most frequent items dynamically. *ACM Trans. Database Syst.*, 30(1):249–278, 2005. doi: 10.1145/1061318.1061325. URL http://doi.acm.org/10.1145/1061318.1061325.

Peter Damaschke and Azam Sheikh Muhammad. Bounds for nonadaptive group tests to estimate the amount of defectives. In *Combinatorial Optimization and Applications - 4th International Conference, COCOA 2010, Kailua-Kona, HI, USA, December 18-20, 2010, Proceedings, Part II*, pages 117–130, 2010a. doi: 10.1007/978-3-642-17461-2_10. URL https://doi.org/10.1007/978-3-642-17461-2_10.

Peter Damaschke and Azam Sheikh Muhammad. Competitive group testing and learning hidden vertex covers with minimum adaptivity. *Discrete Math., Alg. and Appl.*, 2(3):291–312, 2010b. doi: 10.1142/S179383091000067X. URL https://doi.org/10.1142/S179383091000067X.

R. Dorfman. The detection of defective members of large populations. *Ann. Math. Statist.*, pages 436–440, 1943.

D. Du and F. K Hwang. Combinatorial group testing and its applications. *World Scientific Publishing Company.*, 2000.

D. Du and F. K Hwang. Pooling design and nonadaptive group testing: important tools for dna sequencing. *World Scientific Publishing Company.*, 2006.

Moein Falahatgar, Ashkan Jafarpour, Alon Orlitsky, Venkatadheeraj Pichapati, and Ananda Theertha Suresh. Estimating the number of defectives with group testing. In *IEEE International Symposium on Information Theory, ISIT 2016, Barcelona, Spain, July 10-15, 2016*, pages 1376–1380, 2016. doi: 10.1109/ISIT.2016.7541524. URL https://doi.org/10.1109/ISIT.2016.7541524.

Edwin S. Hong and Richard E. Ladner. Group testing for image compression. *IEEE Trans. Image Processing*, 11(8):901–911, 2002. doi: 10.1109/TIP.2002.801124. URL https://doi.org/10.1109/TIP.2002.801124.

F. K. Hwang. A method for detecting all defective members in a population by group testing. *Journal of the American Statistical Association*, 67:605–608, 1972.

William H. Kautz and Richard C. Singleton. Nonrandom binary superimposed codes. *IEEE Trans. Information Theory*, 10(4):363–377, 1964. doi: 10.1109/TIT.1964.1053689. URL https://doi.org/10.1109/TIT.1964.1053689.

Joseph L.Gastwirth and Patricia A.Hammick. Estimation of the prevalence of a rare disease, preserving the anonymity of the subjects by group testing: application to estimating the prevalence of aids antibodies in blood donors. *Journal of Statistical Planning and Inference.*, 22(1):15–27, 1989.

C. H. Li. A sequential method for screening experimental variables. *J. Amer. Statist. Assoc.*, 57:455–477, 1962.

Anthony J. Macula and Leonard J. Popyack. A group testing method for finding patterns in data. *Discrete Applied Mathematics*, 144(1-2):149–157, 2004. doi: 10.1016/j.dam.2003.07.009. URL https://doi.org/10.1016/j.dam.2003.07.009.

Hung Q. Ngo and Ding-Zhu Du. A survey on combinatorial group testing algorithms with applications to DNA library screening. In *Discrete Mathematical Problems with Medical Applications, Proceedings of a DIMACS Workshop, December 8-10, 1999*, pages 171–182, 1999.

Dana Ron and Gilad Tsur. The power of an example: Hidden set size approximation using group queries and conditional sampling. *CoRR*, abs/1404.5568, 2014. URL http://arxiv.org/abs/1404.5568.

Jens Schlaghoff and Eberhard Triesch. Improved results for competitive group testing. *Combinatorics, Probability & Computing*, 14(1-2):191–202, 2005. doi: 10.1017/S0963548304006649. URL https://doi.org/10.1017/S0963548304006649.

M. Sobel and P. A. Groll. Group testing to eliminate efficiently all defectives in a binomial sample. *Bell System Tech. J.*, 38:1179–1252, 1959.

William H. Swallow. Group testing for estimating infection rates and probabilities of disease transmission. *Phytopathology*, 1985.

Keith H. Thompson. Estimation of the proportion of vectors in a natural population of insects. *Biometrics*, 18(4):568–578, 1962.

S. D. Walter, S. W. Hildreth, and B. J. Beaty. Estimation of infection rates in population of organisms using pools of variable size. *Am J Epidemiol.*, 112(1):124–128, 1980.

Jack K. Wolf. Born again group testing: Multiaccess communications. *IEEE Trans. Information Theory*, 31(2):185–191, 1985. doi: 10.1109/TIT.1985.1057026. URL https://doi.org/10.1109/TIT.1985.1057026.

6. Appendix A

In this Appendix we give the proof of Lemma 1, Theorem 6 and a simple proof of Theorem 8.
Lemma 1 *Let $S = \{s_1, \ldots, s_N\}$ be a set of N distinct strings over $\{0,1\}$ such that no string is a prefix of another. Then, over the uniform distribution,*

$$\max_{s \in S} |s| \geq E(S) := \mathbf{E}_{s \in S}[|s|] \geq \log N.$$

Proof The proof is by induction on N. For $N = 1$ the set S with the smallest $E(S)$ is when $S = \{\lambda\}$ and $E(S) = 0 = \log N$. For $N = 2$ the smallest $E(S)$ is when $S = \{0,1\}$ and $E(S) = 1 = \log N$. Therefore, the statement of the lemma is true for $N = 1, 2$.

Consider a set S of size $N > 2$. Obviously, $\lambda \notin S$. Let $w \in \cup_{i=0}^{\infty}\{0,1\}^i$ be the longest string that is a prefix of all the strings in S. For $\sigma \in \{0,1\}$, let $S_\sigma = \{u \mid w \cdot \sigma \cdot u \in S\}$. Let $N_\sigma = |S_\sigma|$ for $\sigma \in \{0,1\}$. Obviously, $N_0 + N_1 = N$ and for each $\sigma \in \{0,1\}$, no string in S_σ is a prefix of another (in S_σ). Also, $N_0, N_1 > 0$, because otherwise, either w is not the longest common prefix of all the strings in S or $w \in S$ is a prefix of another string in S. Let $p = N_0/N$. By the definition of $E(S)$ and the induction hypothesis

$$\begin{aligned} E(S) &= |w| + 1 + \frac{N_0 E(S_0) + N_1 E(S_1)}{N} \\ &\geq 1 + \frac{N_0 \log(N_0) + N_1 \log(N_1)}{N} \\ &= 1 + \log(N) + p \log p + (1-p)\log(1-p) \geq \log(N). \end{aligned}$$

■

Theorem 6 *Let $\epsilon < 1/2$ and $\min(\epsilon^\lambda, 1/2) \geq \delta \geq 1/(2(n - 1/\epsilon + 1))$ where $\lambda < 1$ is any constant. Let A be a randomized adaptive algorithm that estimates the number of defective items up to a multiplicative factor of $1 \pm \epsilon$. Algorithm A must ask at least*

$$\Omega\left(\frac{1}{\epsilon}\log\frac{1}{\delta}\right)$$

expected number of queries.
Proof Let $A(r)$ be a randomized algorithm that estimates the number of defective items up to a multiplicative factor of $1 \pm \epsilon$ where r is the random seed of the algorithm. Then for $|I| \in \{d, d+1\}$ where $d = \lfloor 1/\epsilon \rfloor - 2$, it determines exactly $|I|$ with probability at least $1 - \delta$. Let $X(I, r)$ be a random variable that is equal to 1 if $A(\mathcal{O}_I, r) \neq |I|$ and 0 otherwise. Then for any $I \subseteq [n]$, $\mathbf{E}_r[X(I,r)] \leq \delta$. Let $m = \lfloor \tau/\delta \rfloor + d - 1 \leq n$ where $\tau > \delta$ is a constant that will be determined later. Consider any $J \subseteq [m]$, $|J| = d$. For any such J let

$$Y_J(r) = X(J, r) + \sum_{i \in [m]\backslash J} X(J \cup \{i\}, r).$$

Then for every $J \subseteq [m]$ of size d, $\mathbf{E}_r\left[Y_J(r)\right] \leq (m - d + 1)\delta \leq \tau$. Therefore for a random uniform $J \subseteq [m]$ of size d we have $\mathbf{E}_r[\mathbf{E}_J[Y_J(r)]] = \mathbf{E}_J[\mathbf{E}_r[Y_J(r)]] \leq \tau$. Let $\eta > \tau$ be a constant that will be determined later. By Markov's inequality, for random r, with probability at least $1-\tau/\eta$, for at least $1-\eta$ fraction of the sets $J\subseteq[m]$, of size d, $Y_J(r)=0$. Let R be the set of such r. Then $\Pr_r[R]\ge 1-\tau/\eta$. Let $r_0\in R$. Let C_{r_0} be the set of all $J\subseteq[m]$, of size d, such that $Y_J(r_0)=0$. Then

$$|C_{r_0}|\ge(1-\eta)\binom{m}{d}=(1-\eta)\binom{\lfloor\tau/\delta\rfloor+d-1}{d}.$$

Consider the deterministic algorithm $A(r_0)$. As in Theorem 6, for every two distinct $J_1,J_2\in C_{r_0}$, there is a query $Q\in Q(A(r_0),\mathcal{O}_{J_1})$ such that $Q(J_1)\ne Q(J_2)$. Then by Lemma 3, the average-case query complexity of $A(r_0)$ is at least

$$\log|C_{r_0}|\ge\log(1-\eta)\binom{\lfloor\tau/\delta\rfloor+d-1}{d}\ge d\log\frac{\tau}{d\delta}-\log\frac{1}{1-\eta}.$$

Let $Z(\mathcal{O}_I,r)=|Q(A(r),\mathcal{O}_I)|$. We have shown that for every $r\in R$,

$$\mathbf{E}_{I\in C_r}[Z(\mathcal{O}_I,r)]\ge d\log\frac{\tau}{d\delta}-\log\frac{1}{1-\eta}.$$

Therefore for every $r\in R$,

$$\begin{aligned}\mathbf{E}_I[Z(\mathcal{O}_I,r)] &\ge \Pr[I\in C_r]\cdot\mathbf{E}_I[Z(\mathcal{O}_I,r)|I\in C_r]\\ &\ge (1-\eta)\left(d\log\frac{\tau}{d\delta}-\log\frac{1}{1-\eta}\right).\end{aligned}$$

Therefore

$$\begin{aligned}\mathbf{E}_I\mathbf{E}_r[Z(\mathcal{O}_I,r)] &= \mathbf{E}_r\mathbf{E}_I[Z(\mathcal{O}_I,r)]\\ &\ge \Pr[r\in R]\cdot\mathbf{E}_r[\mathbf{E}_I[Z(O_I,r)]|r\in R]\\ &\ge \left(1-\frac{\tau}{\eta}\right)(1-\eta)\left(d\log\frac{\tau}{d\delta}-\log\frac{1}{\eta}\right).\end{aligned}$$

Therefore there is I such that

$$\mathbf{E}_r[Z(\mathcal{O}_I,r)]\ge\left(1-\frac{\tau}{\eta}\right)(1-\eta)\left(d\log\frac{\tau}{d\delta}-\log\frac{1}{\eta}\right).$$

Now for $\eta=\sqrt{\tau}=1/16$ we get

$$\mathbf{E}_r[Z(\mathcal{O}_I,r)]=\Omega\left(\frac{1}{\epsilon}\log\frac{1}{\delta}\right).$$

■

We now give a simple proof of Theorem 8.

Theorem 8 *Let A be a randomized adaptive algorithm that estimates d up to multiplicative factor of $1/4$ with probability at least $1-\delta$. The expected number of queries of A is at least*

$$(1-\delta)(\log\log d-\log\log\log d-2)$$

Proof Let $A(r)$ be an adaptive algorithm that estimates d up to a multiplicative factor of $1/4$ with probability at least $1-\delta$. Let $q(d)$ be the expected number of queries of $A(r)$. Define a sequence of sets $I_1=[1], I_2=[2],\ldots,I_t=[2^t]$ where $2^t\le d$ and $2^{t+1}>d$. Then $t=\lfloor \log d\rfloor$. We restrict the inputs of A to be only I_j for some $j=1,\ldots,t$ and force A to halt if it asks more than $q(d)/(1-\delta-\eta)$ queries where $\eta>0$ will be determined later. This new algorithm, denoted by B, is a Monte Carlo algorithm that finds exactly the size of I_j with probability at least $1-(\delta+(1-\delta-\eta))=\eta$ and asks at most $q(d)/(1-\delta-\eta)$ queries. Therefore by Theorem 3 (see (1)), $q(d)/(1-\delta-\eta)\ge \log\log d+\log\eta$ and therefore for $\eta=(\ln 2)(1-\delta)/\log\log d$ we get

$$\begin{aligned} q(d) &\ge (1-\delta-\eta)(\log\log d+\log\eta)\\ &\ge (1-\delta)(\log\log d-\log\log\log d-2). \end{aligned}$$

■

7. Appendix B: A Randomized Monte Carlo Algorithm

In this section we give a randomized Monte Carlo algorithm.

In Lemma 14, if we take the sequence $\Delta_1=1$ and $\Delta_i=2^{\Delta_{i-1}}$ then $\Delta_{i_1}\le 2^{2d/\delta}$, the expected number of queries is $\log^*(d/\delta)$ and the output D satisfies

$$d\le D\le 2^{2d/\delta+1}\log\frac{2}{\delta}.$$

The advantage of this algorithm is that, by Lemma 14, it is also a randomized Monte Carlo algorithm that asks at most $i_2=\log^* n$ queries. Now we can narrow the range and keep the worst case query complexity small by choosing the sequences $\Delta_i=2^{2^{2^{2^i}}}$ then $\Delta_i=2^{2^{2^i}}$ then $\Delta_i=2^{2^{i^2}}$ and then running the last 3 stages of the algorithm by Falahatgar et al. (Falahatgar et al., 2016).

The following table gives the parameters in each stage.

$\Delta_i=$	i_1	D^*	$\Delta_{i_1}=\frac{D}{2\log(2/\delta)}\le$	i_2
$2^{\Delta_{i-1}}$	$\log^*(d/\delta)$	n	$2^{2d/\delta}$	$\log^* n$
$2^{2^{2^{2^i}}}$	$\log^{[4]}\frac{2d}{\delta}+1$	$2^{2d/\delta+1}\log\frac{2}{\delta}$	$2^{2^{(\log^{[3]}\frac{2d}{\delta})^2}}$	$\log^{[3]}\frac{2d}{\delta}$
$2^{2^{2^i}}$	$\log^{[3]}\frac{2d}{\delta}+1$	$2^{2^{(\log^{[3]}\frac{2d}{\delta})^2}+1}\log\frac{2}{\delta}$	$2^{(\log\frac{2d}{\delta})^2}$	$2\log^{[4]}\frac{2d}{\delta}$
$2^{2^{i^2}}$	$\sqrt{\log^{[2]}\frac{2d}{\delta}}+1$	$2^{(\log\frac{2d}{\delta})^2+1}\log\frac{2}{\delta}$	$\left(\frac{2d}{\delta}\right)^{2^{2\sqrt{\log\log\frac{2d}{\delta}}+1}}$	$\log^{[3]}\frac{2d}{\delta}$

Here $\log^{[k]} n=\log\log^{[k-1]} n$ and $\log^{[1]} n=\log n$.

This gives the following result.

Theorem 15 *There is a randomized Monte Carlo algorithm that asks*

$$\log^* n + \log\log d + O(\sqrt{\log\log d}) + O\left(\frac{1}{\epsilon^2}\log\frac{1}{\delta}\right)$$

queries in expectation and with probability at least $1-\delta$ estimates the number of defective items d up to a multiplicative factor of $1\pm\epsilon$.

Note: The above stages can even start from a much slower function. For example $\log^{**} n$ that is defined as $\log^{**}\alpha = 1$ for $\alpha \le 2$ and $\log^{**} n = 1 + \log^{**}(\log^* n)$.