Electronic Colloquium on Computational Complexity, Report No. 78 (2003)

Finding Favorites

Fan Chung [*,†] Ron Graham[*] Jia Mao[*] Andrew Yao[‡]

Abstract

We investigate a new type of information-theoretic identification problem, suggested to us by Alan Taylor. In this problem we are given a set of items, more than half of which share a common "good" value. The other items have various other values which are called "bad". The only method we have for gaining information about the items' values is to ask whether two items share the same value. We can assume there is an oracle which always answers each such query truthfully. Our goal is to identify at least one good item, i.e., an item which is guaranteed to have a good value, using a minimum possible number of queries. We will establish upper and lower bounds for the number of queries needed for both adaptive as well as oblivious strategies.

The practical context in which this problem arose was in connection with trying to identify a good sensor from a set of sensors in which some are non-operational or corrupted, for example, where it was desired to minimize the amount of intercommunication used in doing so.

1 Introduction

We imagine a situation in which we are initially given some set S of n elements. Each element $s \in S$ has been assigned some value $\phi(s) \in \{0, 1, \ldots, R\}$, with the restriction that most of the elements of S have been assigned the value 0, i.e., $|\phi^{-1}(0)| > n/2$. These are called "good" elements; the others are called "bad". Our goal is to identify at least one good element. However, the only way we have of gaining information about the values of elements is to ask whether two elements have the *same* value, i.e., for $s, t \in S$, we can ask:

[*]Department of Computer Science & Engineering, University of California, San Diego
[†]Research supported in part by NSF Grant DMS 0100472 and ITR 0205061
[‡]Department of Computer Science, Princeton University; research supported in part by NSF Grant CCR-0310466

ISSN 1433-8092

"Is $\phi(s) = \phi(t)$?". We assume each such query will be answered truthfully by an oracle. In particular, we would like to use the smallest possible number of queries for which we can be guaranteed of learning the identity of a good element, no matter how the values $\phi(s), s \in S$, happen to be chosen (subject only to the requirement that $|\phi^{-1}(0)| > n/2$).

We can view our problem as a game played betweeen two players: **Q**, the Questioner, and **A**, the adversary. **Q**'s role is to ask a sequence of queries $Q(s,t) :=$ "Is $\phi(s) = \phi(t)$?". **A** must answer each such query truthfully, but is free to change the current values $\phi(s), s \in S$, as long as all previous answers remain true, and of course, so that $|\phi^{-1}(0)| > n/2$. In this way, **A** would like to extend the game as long as possible before **Q** can finally identify a good element.

We will consider two types of strategies for **Q**. These are *adaptive* strategies in which each query can depend on the answers given to all previous queries, and *oblivious* (or non-adaptive) strategies in which all the queries must be specified before **A** is required to answer any of them. Clearly, in the oblivious case, **A** has more opportunity to be evasive. We will let $f(n)$ denote the minimum number of queries needed by **Q** to guarantee finding a good element with some adaptive strategy, and we will let $g(n)$ denote the corresponding minimum over all oblivious strategies. We will also examine the special case in which there is only one bad value, i.e., $\phi : S \to \{0, 1\}$. In this case, we denote the corresponding functions by $f_1(n)$ and $g_1(n)$, respectively. Clearly, $f_1(n) \le f(n)$ and $g_1(n) \le g(n)$ for all n.

2 Adaptive strategies

Theorem 1 *Suppose n is odd. Then*

$$n - 2\sqrt{n} \le f_1(n) \le f(n) \le n - w_2(n) \tag{1}$$

where $w_2(n)$ denotes the binary weight of n, i.e., the number of 1's in the binary expansion of n.

Proof: (Lower bound for $f_1(n)$)

We assume that the Adversary maintains at any time the information revealed so far as a graph H on n vertices $S = \{s_1, s_2, \ldots, s_n\}$ with an edge $\{s_i, s_j\}$ corresponding to each query $Q(s_i, s_j)$, i.e., " Is $\phi(s_i) = \phi(s_j)$?". The edge is colored *blue* if the answer is "$\phi(s_i) = \phi(s_j)$", and is colored *red*, otherwise. At any time in this process the graph H has a certain number of connected components, say $C_1, C_2, \ldots, C_m$. Each component C_r is uniquely 2-colorable in the sense that its vertices can be uniquely partitioned into two clusters (one of which may be empty) so that blue edges exist only between vertices in the same cluster, and red edges exist only between vertices belonging to different clusters. For each component C_r, let $\delta(C_r) = |a - b|$ where a and b are the sizes of the two clusters. Also, let $\Delta(H) = \sum_{r=1}^{m} \delta(C_r)$.

Lemma 1 *The number of queries asked so far is at least as large as $n - m$, where m is the number of connected components of H.*

Proof: The number of queries asked is exactly the number of edges in G, which is at least as large as

$$\sum_{r=1}^{m} (|C_r| - 1) = n - m.$$

This proves Lemma 1. □

Note that any valid assignment of values $\phi(s_i)$ to the s_i consistent with H is obtainable in the following way. For each component C_r, assign 0 to all vertices in one cluster, and assign 1 to all vertices in the other cluster. Of course, an assignment is valid only if the total number of 0's is greater than $n/2$.

The next lemma gives a characterization for when the current information is sufficient to identify some s_i as having $\phi(s_i) = 0$.

Lemma 2 *Let $i \in \{1, 2, \ldots, n\}$. Every assignment consistent with H must assign 0 to s_i (i.e., $\phi(s_i) = 0$), if and only if $\delta(C_r) \geq \frac{1}{2}\Delta(H)$, where s_i is in the larger cluster of the component C_r.*

Proof: First, suppose that $\delta(C_r) < \frac{1}{2}\Delta(H)$. For every component C_s where $s \neq r$, assign the values of vertices in C_s to be biased in favor of 0. Then no matter which of the two ways one assigns values to the vertices in C_r, the total number of 0's minus the total number of 1's is at least as large as

$$\sum_{s \neq r} \delta(C_s) - \delta(C_r) \geq \Delta(H) - 2\delta(C_r) > 0,$$

and will therefore be an acceptable assignment. This contradicts the assumption that $\phi(s_i)$ must be 0. On the other hand, if $\delta(C_r) \geq \frac{1}{2}\Delta(H)$ and $\phi(s_i) = 1$ then every element in the larger cluster (together with s_i) of C_r must also have the value 1. Thus

$$|\phi^{-1}(0)| - |\phi^{-1}(1)| \leq -\delta(C_r) + \sum_{s \neq r} \delta(C_s) \leq 0$$

which contradicts the requirement that $|\phi^{-1}(0)| > n/2$. This proves Lemma 2. □

Let us now postulate a strategy for the Adversary **A** which will produce the desired lower bound for $f_1(n)$. Suppose $Q(s_i, s_j)$ is the next query. The adversary answers by consulting the current auxiliary graph H. Without loss of generality, we can assume that s_i and s_j come from two different components C_k and C_l. No matter which way this query is answered, the two components C_k and C_l will be merged into one component C_r. Depending on the answers, one has either $\delta(C_r) = \delta(C_k) + \delta(C_l)$, or $\delta(C_r) = |\delta(C_k) - \delta(C_l)|$. Here is how the Adversary answers the query:

If $\delta(C_k) \leq \sqrt{n}/2$ and $\delta(C_l) \leq \sqrt{n}/2$ then it is answered so that $\delta(C_r) = \delta(C_k) + \delta(C_l)$; otherwise it is answered so that $\delta(C_r) = |\delta(C_k) - \delta(C_l)|$.

Note that with this strategy, all components C_r must satisfy $0 \leq \delta(C_r) \leq \sqrt{n}$ at all times. Furthermore, $\delta(C_r)$ can be zero only when it is obtained from a query with $\delta(C_k) = \delta(C_l) > \sqrt{n}/2$, and hence $|C_r| > \sqrt{n}$.

To analyze this Adversary strategy, consider any algorithm for **Q**, and let G be the auxiliary graph when this algorithm halts under the specified Adversary strategy. The next two lemmas show that G has at most $2\sqrt{n}$ connected components. By Lemma 1, this immediately proves the lower bound for $f_1(n)$ in Theorem 1.

Lemma 3 *G has fewer than $\sqrt{n}$ components C_r with $\delta(C_r) = 0$.*

Proof: As noted previously, any C_r with $\delta(C_r) = 0$ must contain more than $\sqrt{n}$ vertices. There has to be fewer than $n/\sqrt{n} = \sqrt{n}$ such components. This proves Lemma 3. □

Lemma 4 *G has at most $\sqrt{n}+1$ components C_k with $\delta(C_k) > 0$.*

Proof: By Lemma 2, we must have some component C_r such that $\delta(C_r) \geq \frac{1}{2}\Delta(G)$. Let I be the set of indices i such that $\delta(C_i) > 0$. Then

$$2\delta(C_r) \geq \sum_{i \in I} \delta(C_i)$$

and hence, $\delta(C_r) \geq |I| - 1$. But $\delta(C_r) \leq \sqrt{n}$ by the preceding remarks. This implies that $|I| \leq \sqrt{n} + 1$. This completes the proof of Lemma 4 and the proof for the lower bound of Theorem 1. □

We now establish the upper bound for $f(n)$ in (1). We will do this by providing a specific strategy for **Q**. As in the preceding proof of the lower bound for $f_1(n)$, the current state of the information revealed so far will be expressed by a graph H, with edges colored blue (if the corresponding vertices have the same value), or red (if they have different values). Let us call a connected component C_r of H *mixed* if it has at least one red edge. Otherwise, we say that C_r is *pure*. Note that with this definition, an isolated point is a pure component. **Q**'s strategy is simply this: The next query will be any question $Q(s_i, s_j)$: "Is $\phi(s_i) = \phi(s_j)$?" where s_i and s_j belong to two different pure components having the same size. Since H initially consists of n pure components of size 1, then in general, any connected component C_r of H will have $|C_r| = 2^t$ for some $t \geq 0$.

Let G be the final graph when the process stops, i.e., when all pure components $C_1, \ldots, C_m$ of G have distinct sizes, say $|C_i| = 2^{t_i}$ with $t_1 > t_2 > \ldots > t_k \geq 0$. We first observe that

$k \geq 1$, i.e., G must have at least one pure component. This is because any mixed component C of G must have been formed from two equal-sized pure components having different values, and so can have at most half of its points good. Since by hypothesis more than half of the points are good, then at least one pure component is required to account for the remaining good points. We now claim that any point $s_i \in C_1$ must be good. For if not, then

$$n = \sum_{i=1}^{k} 2^{t_i} + \sum_{C \text{ mixed}} |C| \tag{2}$$

and

$$\# \text{ of bad points} \geq 2^{t_1} + \frac{1}{2} \sum_{C \text{ mixed}} |C| > n/2$$

(since $2^{t_1} > \sum_{i \geq 2} 2^{t_i}$) so that $|\phi^{-1}(0)| < n/2$, which is a contradiction.

Finally, if G has m components altogether (pure and mixed), then by (2), n can be represented as a sum of m powers of 2. If $n = \sum_{j=1}^{p} 2^{a_j}$ is such a representation with the minimum possible value of p, then we must have all the a_j distinct, since otherwise we can replace $2^{a_j} + 2^{a_j}$ by 2^{a_j+1}. Therefore, $p = w_2(n)$, the number of 1's in the binary expansion of n. Since by the definition of **Q**'s strategy, all the components of G are trees, then Lemma 2 implies that **Q** asks at most $n - w_2(n)$ queries. This proves Theorem 1. □

We remark that the upper bound here was already obtained by Taylor and Zwicker [9]. Computation has verified that $f_1(n) = n - w_2(n)$ for n odd, $1 \leq n \leq 31$. For even values of the argument, we have the following:

Lemma 5 *$f(n+1) \leq f(n)$ if n is odd.*

Proof: Let $S = \{s_1, \ldots, s_{n+1}\}$ be our initial set of items. By hypothesis, $|\phi^{-1}(0)| > \frac{1}{2}|S| = \frac{n+1}{2}$. Thus, $|\phi^{-1}(0)| \geq \frac{n+3}{2}$. Hence, if we remove s_{n+1} from S to form $S^- = \{s_1, \ldots, s_n\}$, then for the elements of S^- we have $|\phi^{-1}(0)| \geq \frac{n+1}{2} > \frac{n}{2}$. Therefore, **Q** can apply any (optimal) strategy for identifying a good element of S^-, and this requires (by definition) at most $f(n)$ queries. This proves Lemma 5. □

In fact, we believe equality should always hold in Lemma 5.

3 Oblivious strategies with one bad value

We now consider the situation in which all of **Q**'s queries must be specified in advance before the Adversary is required to answer any of them. In principle, this could give the Adversary a significant advantage compared to the adaptive case. We first deal with the case in which there is only one bad value, i.e., with the value of $g_1(n)$.

Theorem 2 *For $n \geq 3$,*

$$g_1(n) \;=\; \begin{cases} n-2 & \text{if } n \text{ is odd,} \\ n-3 & \text{if } n \text{ is even.} \end{cases}$$

Proof: Suppose the oblivious Adversary still uses the auxiliary graph H to maintain information. At any time in the game, H consists of a number of connected components $C_1, C_2, ..., C_m$. We say that C_i is *odd* if the number of vertices in C_i is odd. Otherwise we say C_i is *even*.

It is easy to see (by induction on $|C_i|$) that the Adversary can always assign values of 0 and 1 to the vertices of a component C_i so that $\delta(C_i) = 1$ if C_i is odd, and $\delta(C_i) = 0$ (or $\delta(C_i) = 2$) if C_i is even.

First, suppose n is odd. Then H must have an odd number t of odd components. However, if $t \geq 3$ then the Adversary could force t values of $\delta(C_i)$ to be 1, which contradicts Lemma 2. Hence we must have $t = 1$ when n is odd. We now claim that H can have at most two components. For suppose to the contrary that H has $m \geq 3$ components $C_i, 1 \leq i \leq m$. Then the Adversary can make $0, 1$ assignments so that $\delta(C_i) = 1$ for each odd C_i, and $\delta(C_i) = 2$ for each even C_i. However, this contradicts the conclusion of Lemma 2. Thus, by Lemma 1, $g_1(n) \geq n-2$ if n is odd.

A very similar argument applies to the case that n is even to show that $m \leq 3$. Thus, we can conclude that $g_1(n) \geq n-3$ when n is even.

To establish the upper bounds, **Q** chooses components as follows:

If n is odd, then $H = \{C_1, C_2\}$ with $|C_1| = n-1, |C_2| = 1$;

If n is even, then $H = \{C_1, C_2, C_3\}$ with $|C_1| = n-2, |C_2| = |C_3| = 1$.
Since $\delta(C) = 1$ if $|C| = 1$ and $\delta(C)$ is even if $|C|$ is even, then in each case we can always identify a good element once the Adversary answers all the queries. This proves Theorem 2.
□

4 Oblivious strategies with many bad values

We next consider the case in which element s in S can be assigned some value $\phi(s) \in \{1, 2, \ldots, R\}$ where R is unrestricted. In principle, this is a more challenging situation for **Q**. At least the upper bounds we have on $g(n)$ in this case are weaker than those for $g_1(n)$.

Theorem 3 *For all n,*

$$g(n) \leq (1 + o(1))27n.$$

Proof: Let B denote the set $\{1, 2, \ldots, R\}$. Thus, s is bad if and only if $\phi(s) \in B$. We will specify the queries of **Q** by a graph H on the vertex set S, where an edge $\{s_i, s_j\}$ in H corresponds to the query $Q(s_i, s_j) :=$ "Is $\phi(s_i) = \phi(s_j)$?". As usual, the edge is colored *blue* if they are equal, and *red* if they are not equal. By a *valid assignment* ϕ on S we mean a mapping $\phi : S \to \{0, 1, \ldots, R\}$ such that:
(i) $\phi(s_i) = \phi(s_j) \Rightarrow \{s_i, s_j\}$ is blue,
(ii) $\phi(s_i) \neq \phi(s_j) \Rightarrow \{s_i, s_j\}$ is red,
(iii) $|\phi^{-1}(0)| > n/2$.

We are going to use certain special graphs $X^{p,q}$, called Ramanujan graphs, which are defined for any primes p and q congruent to 1 modulo 4 (see [7]).

$X^{p,q}$ has the following properties:
(i) $X^{p,q}$ has $n = \frac{1}{2}q(q^2-1)$ vertices;
(ii) $X^{p,q}$ is regular of degree $p+1$;
(iii) The adjacency matrix of $X^{p,q}$ has the large eigenvalue $\lambda_0 = p+1$ and all other eigenvalues λ_i satisfying $|\lambda_i| \leq 2\sqrt{p}$.

We will use the following discrepancy inequality (see [1, 2]) for a d-regular graph $H = H(n)$ with eigenvalues satisfying

$$\max_{i\neq 0} |\lambda_i| \leq \delta.$$

For any disjoint subsets $X, Y \subseteq V(H)$, the vertex set of H, we have

$$| \, e(X,Y) - \frac{d}{n}|X| \; |Y| \, |\leq \frac{\delta}{n}\sqrt{|X|(n-|X|)|Y|(n-|Y|)} \tag{3}$$

where $e(X,Y)$ denotes the number of edges between X and Y.

Applying (3) to $X^{p,q}$, we obtain for all $X, Y \subseteq V(X^{p,q})$,

$$| \, e(X,Y) - \frac{p+1}{n}|X| \; |Y| \, |\leq \frac{2\sqrt{p}}{n}\sqrt{|X|(n-|X|)|Y|(n-|Y|)} \tag{4}$$

where $n = \frac{1}{2}q(q^2-1) = |V(X^{p,q})|$.

We will now set $S = V(X^{p,q}) = \{s_1, \ldots, s_n\}$. Let ϕ be a valid assignment of S to $\{0, 1, \ldots, R\}$ and consider the subgraph G of $X^{p,q}$ induced by $\phi^{-1}(0)$ (the good vertices of $X^{p,q}$ under the mapping ϕ).

Claim: Suppose $p \geq 41$. Then G has a connected component C with size at least $c'n$, where

$$c' > \frac{1}{2} - \frac{8p}{(p-1)^2}$$

Proof: We will use (4) with $X = C$, the largest connected component of G, and $Y = \phi^{-1}(0) \setminus X$. Write $|\phi^{-1}(0)| = \alpha n$ and $|C| = \beta n$. Since $e(X,Y) = 0$ for this choice, then by (4) we have

$$\begin{aligned}
(p+1)^2|X| \; |Y| &\leq 4p(n-|X|)(n-|Y|), \\
(p+1)^2\beta(\alpha-\beta) &\leq 4p(1-\beta)(1-\alpha+\beta), \\
\beta(\alpha-\beta) &\leq \frac{4(1-\alpha)p}{(p-1)^2},
\end{aligned}$$

There are two possibilities:

$$\beta \geq \frac{1}{2}\left(\alpha + \sqrt{\alpha^2 - \frac{16(1-\alpha)p}{(p-1)^2}}\;\right) \quad \text{or} \quad \beta \leq \frac{1}{2}\left(\alpha - \sqrt{\alpha^2 - \frac{16(1-\alpha)p}{(p-1)^2}}\;\right)$$

Subcase (a).

$$\begin{aligned}
\beta &\geq \frac{1}{2}\left(\alpha + \sqrt{\alpha^2 - \frac{16(1-\alpha)p}{(p-1)^2}}\ \right) \\
&> \frac{1}{4}\left(1 + \sqrt{1 - \frac{32p}{(p-1)^2}}\ \right) && \text{since } \alpha \geq 1/2 \\
&\geq \frac{1}{2} - \frac{8p}{(p-1)^2} && \text{since } p \geq 37
\end{aligned}$$

as desired.

Subcase (b).

$$\begin{aligned}
\beta &\leq \frac{1}{2}\left(\alpha - \sqrt{\alpha^2 - \frac{16(1-\alpha)p}{(p-1)^2}}\ \right) \\
&\leq \frac{8(1-\alpha)p}{\alpha(p-1)^2}
\end{aligned}$$

Thus, we can choose a subset F of some of the connected components whose union $\cup F$ has size $xn = |\cup F|$ satisfying

$$\frac{\alpha}{2} - \frac{4(1-\alpha)p}{\alpha(p-1)^2} \leq x < \frac{\alpha}{2} + \frac{4(1-\alpha)p}{\alpha(p-1)^2} \tag{5}$$

Now we apply the discrepancy inequality (4) again by choosing $X = \cup F$ and $Y = \phi^{-1}(0) \setminus X$. We have

$$\begin{aligned}
(p+1)^2 x(\alpha - x) &\leq 4p(1-x)(1-\alpha+x) \\
\text{or}\ \ x(\alpha - x) &\leq \frac{4(1-\alpha)p}{(p-1)^2}.
\end{aligned}$$

However, it is easily checked that because of (5) this is not possible for $\alpha \geq 1/2$ and $p \geq 41$. Hence, subcase (b) cannot occur. This proves the claim.

We now prove Theorem 3. Let $n = \frac{1}{2}q(q^2-1)$. Consider an algorithm $\mathbf{Q}$ specified by a graph $H = X^{p,q}$ where $p \geq 53$. We show that a good element can always be identified after all the queries are answered.

Suppose we have an arbitrary blue/red coloring of the edges of $X^{p,q}$, and $\phi : S \to \{0, 1, \ldots, R\}$ is a valid assignment on $S = V(X^{p,q})$. Consider the connected components formed by the blue edges of $X^{p,q}$. By the Claim there is at least one blue component of size at least $(\frac{1}{2} - \frac{8p}{(p-1)^2})n > \frac{1}{3}n$ (since $p \geq 53$.). Call any such blue component *large*.

If there is only one large component then we are done, i.e., every point in it must be good. Since there cannot be three large blue components, the only remaining case is that we have exactly two large blue components, say S_1 and S_2. Again, if either $S_1 \subseteq \phi^{-1}(0)$ or $S_2 \subseteq \phi^{-1}(0)$ is forced, then we are done. So we can assume there is a valid assignment ϕ_1 with $S_1 \subseteq \phi_1^{-1}(0)$, $S_2 \subseteq \phi_1^{-1}(B)$, and a valid assignment ϕ_2 with $S_2 \subseteq \phi_2^{-1}(0), S_1 \subseteq \phi_2^{-1}(B)$ (where we recall that $B = \{1, 2, \ldots, R\}$).

Let us write $S_i' = \phi_i^{-1}(0) \setminus S_i, i = 1, 2$. Since $\phi_1^{-1}(0)$ and $\phi_2^{-1}(0)$ each has more than $n/2$ elements, we clearly must have $A := S_1' \cap S_2' \neq \emptyset$. Also note that $|A| \leq n - |S_1| - |S_2| < \frac{16p}{(p-1)^2}n$.

Define $B_1 = S_1' \setminus A, B_2 = S_2' \setminus A$. Observe that there can be no edge between A and $S_1 \cup S_2 \cup B_1 \cup B_2$. Now we are going to use (4) again, this time choosing $X = A, Y = S_1 \cup S_2 \cup B_1 \cup B_2$.

Note that

$$n > |Y| = |\phi_1^{-1}(0)| - |A| + |\phi_2^{-1}(0)| - |A| > n - 2|A|.$$

Since $e(X, Y) = 0$, we have by (4),

$$\begin{aligned} (p+1)^2|X|\ |Y| &\leq 4p(n - |X|)(n - |Y|), \\ (p+1)^2|A|(n - 2|A|) &\leq 4p(n - |A|)2|A|. \end{aligned}$$

However, this implies

$$\begin{aligned}(p+1)^2(n-2|A|) &\leq 8p(n-|A|),\\ \text{i.e.,}\qquad n((p+1)^2-8p) &\leq 2|A|((p+1)^2-4p)\\ &\leq 2|A|(p-1)^2\\ &< 32pn\\ (p+1)^2-8p &< 32p\end{aligned}$$

which is impossible for $p \geq 41$.

Setting $p = 53$ (so that $X^{p,q} = X^{53,q}$ is regular of degree $p+1 = 54$), we see that $X^{53,q}$ has $(1+o(1))27n$ edges. This shows that Theorem 3 holds when $n = \frac{1}{2}q(q^2-1)$ for a prime $q \equiv 1 (\text{mod } 4)$.

If $\frac{1}{2}q_i(q_i^2-1) < n < \frac{1}{2}q_{i+1}(q_{i+1}^2-1) = n'$ where q_i and q_{i+1} are consecutive primes congruent to 1 modulo 4, we can simply augment our initial set S to a slightly larger set S' of size n' by adding $n'-n = \delta(n)$ additional good elements. Standard results from number theory show that $\delta(n) = o(n)$, (see [8]). Since the Ramanujan graph query strategy of $\mathbf{Q}$ actually identifies $\Omega(n')$ good elements of S' (for fixed p) then it certainly identifies a good element of our original set S. This proves Theorem 3 for all n. □

We point out that we always have $g_1(n) \leq g(n)$ for all n. We also remark that the constant 27 can be further reduced by using random sparse graphs and applying concentration estimates from probabilistic graph theory. However, such methods can only deduce the existence of a graph with the desired properties whereas we use an explicit construction (Ramanujan graphs) here.

5 Concluding remarks

We mention several unanswered questions here.

(i) Is it true that for all odd $n \geq 1$,

$$f_1(n) = f(n) = n - w_2(n)?$$

Computation shows this is true for $n \leq 45$.

(ii) Is it true that $g_1(n) = g(n)$ for all n?

We currently have no example showing that allowing an arbitrary number of bad values gives the Adversary any advantage over just allowing one bad value.

(iii) What is the true order of growth of $g(n)$?

For example, is it true that $g(n) = (1 + o(1))n$?

(iv) Suppose we consider the more general situation in which we only require, for a valid assignment ϕ, that $|\phi^{-1}(0)| \geq \alpha n$ for some fixed $\alpha > 0$ and $|\phi^{-1}(i)| < |\phi^{-1}(0)|$ for all $i > 0$. What are the corresponding bounds on f and g in this case ? Of course the same questions can be asked in the even more general situation where we just assume $|\phi^{-1}(0)| \geq h(n)$ for some function of n. For example, in the extreme case that we only assume $|\phi^{-1}(0)| = 2$, so that $|\phi^{-1}(i)| = 1$, for $i > 0$, it can require $\binom{n}{2} - 1$ queries in the worst case to determine a good element.

(v) On the opposite side of the coin, so to speak, to our problems are those coming from *group testing*. This involves the problem of identifying a small subset of defective items in a much larger set by means of "group tests", in which various subsets X are tested with the query "*Is there a defective item in X?*" Again, the challenge is to minimize the number of such queries needed to identify all defective items (see [3]) for a comprehensive survey).

(vi) The adaptive version of our problem with only one bad value (i.e., $R = 1$) has the following equivalent number-theoretic formulation, still as a game played between two players **Q** and **A**. At any point during the game a multiset S of non-negative integers is maintained (where the starting multiset consists on n 1's). **Q**'s move is to select two elements from S, say a and b. **A**'s move is then to remove a and b from S, and replace them by the single element s which is either $a+b$ of $|a-b|$ (**A** gets to choose which), forming a new multiset with one fewer element. The game is over as soon as

$$\max_{s \in S} s \geq \frac{1}{2} \sum_{s' \in S} s'.$$

A's goal is to extend the game as long as possible. The elements of S correspond to the discrepancies of the components in the earlier formulation. The maximum length of the game is just $f_1(n)$.

More generally, we could start with some arbitrary multiset S of non-negative integers and ask for the maximum length $f_1(S)$ of the corresponding game in this case. As pointed out by Don Knuth [5], in this form the game has a similar flavor to the problem of finding the second largest element in a partially sorted subset of a linearly ordered set studied by Floyd and Knuth [4, 6]. In that case, the queries we are allowed have the form "Is $x < y$?", and the $s \in S$ represent the sizes of the partially ordered components. Conceivably, a careful examination of the data for this more general problem would suggest an exact expression for $f_1(S)$ which could be proved inductively. Certainly, this represents a promising direction to explore.

References

[1] N. Alon, Eigenvalues and expanders, *Combinatorica* **6** (1986), 86-96.

[2] F. R. K. Chung, Spectral Graph Theory, *CBMS Lecture Notes*, AMS Publications.

[3] D.-Z. Du and F. K. Hwang, *Combinatorial Group Testing and its Applications*, World Scientific Publishing Co., Inc., River Edge, NJ, 1993.

[4] R. W. Floyd and D. E. Knuth, The Bose-Nelson sorting problem, *A survey of combinatorial theory (Proc. Internat. Sympos., Colorado State Univ., Fort Collins, colo.,* 1971), pp. 163-172, North-Holland, Amsterdam, 1973.

[5] D. E. Knuth, personal communication.

[6] D. E. Knuth, *The Art of Computer Programming, Volume 3. Sorting and Searching*, Addison-Wesley Publishing Co., Reading, Mass., 1973.

[7] A. Lubotzky, R. Phillips and P. Sarnak, Ramanujan graphs, *Combinatorica* **8** (1988), 261-277.

[8] K. Prachar, *Primzahlverteilung*, Springer-Verlag, Berlin, 1957, pp. 327.

[9] A. Taylor and W. Zwicker, personal communication.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'