A New Approach to the Sensitivity Conjecture

Justin Gilmer
Department of Mathematics
Rutgers University
Piscataway, NJ, USA
jmgilmer@math.rutgers.edu

Michal Koucký
Computer Science Institute
Charles University
Prague, Czech Republic
koucky@iuuk.mff.cuni.cz

Michael Saks
Department of Mathematics
Rutgers University
Piscataway, NJ, USA
saks@math.rutgers.edu

ABSTRACT

One of the major outstanding foundational problems about boolean functions is the *sensitivity conjecture*, which (in one of its many forms) asserts that the degree of a boolean function (i.e. the minimum degree of a real polynomial that interpolates the function) is bounded above by some fixed power of its sensitivity (which is the maximum vertex degree of the graph defined on the inputs where two inputs are adjacent if they differ in exactly one coordinate and their function values are different). We propose an attack on the sensitivity conjecture in terms of a novel two-player communication game. A strong enough lower bound on the cost of this game would imply the sensitivity conjecture.

To investigate the problem of bounding the cost of the game, three natural (stronger) variants of the question are considered. For two of these variants, protocols are presented that show that the hoped for lower bound does not hold. These protocols satisfy a certain monotonicity property, and (in contrast to the situation for the two variants) we show that the cost of any monotone protocol satisfies a strong lower bound.

Categories and Subject Descriptors

F.1.3 [**Computation by Abstract Devices**]: Complexity Measures and Classes—*Relations among complexity measures*

Keywords

Sensitivity conjecture; degree of Boolean functions; sensitivity; decision trees; communication complexity

1. INTRODUCTION

1.1 A Communication Game

The focus of this paper is a somewhat unusual cooperative two player communication game. The game is parameterized by a positive integer n and is denoted G_n. Alice receives a permutation $\sigma = (\sigma_1, \ldots, \sigma_n)$ of $[n] = \{1, \ldots, n\}$ and a bit $b \in \{0, 1\}$ and communicates to Bob in a very restricted way (which will be described momentarily). Bob receives the message from Alice and then outputs a subset J of $[n]$ that is required to include σ_n, the last element of the permutation. The cost to Alice and Bob is the size of the set $|J|$.

The communication from Alice to Bob is constrained as follows: Alice has a memory vector $\mathbf{v}$ consisting of n cells which we will refer to as *locations*, where each location v_ℓ is either empty, denoted by $v_\ell = *$, or is set to 0 or 1. Initially all locations are empty. Alice gets the input as a data stream $\sigma_1, \ldots, \sigma_n, b$ and is required to fill the cells of $\mathbf{v}$ in the order specified by σ. After receiving σ_i for $i < n$, Alice fills location σ_i with 0 or 1. Upon receiving σ_n and b, Alice writes b in location σ_n.

Once $\mathbf{v}$ is filled, Bob inspects $\mathbf{v}$ and outputs the subset J.

Given a protocol Π for this game, the cost of the protocol $c(\Pi)$ is the maximum of the output size $|J|$ over all inputs $\sigma_1, \ldots, \sigma_n, b$.

For example, consider the following protocol. Let $k = \lceil \sqrt{n} \rceil$. Alice and Bob fix a partition of the locations of $\mathbf{v}$ into k blocks each of size at most k. Alice fills $\mathbf{v}$ as follows: When σ_i arrives, if σ_i is the last location of its block to arrive then fill the entry with 1 otherwise fill it with 0.

Notice that if $b = 1$ then the final vector $\mathbf{v}$ will have a single 1 in each block. If $b = 0$ then $\mathbf{v}$ will have a unique all 0 block.

Bob chooses J as follows: if there is an all 0 block, then J is set to be that block, and otherwise J is set to be the set of locations containing 1's. It is clear that $\sigma_n \in J$ and so this is a valid protocol. In all cases the size of J will be at most k and so the cost of the protocol is $\lceil \sqrt{n} \rceil$. We will refer to this protocol as the AND-OR protocol. In Section 2.1 we remark on this protocol's connection to the boolean function

$$\text{AND-OR}(x) = \bigwedge_{i=1}^{\sqrt{n}} \bigvee_{j=1}^{\sqrt{n}} x_{ij}.$$

Let us define $C(n)$ to be the minimum cost of any protocol for G_n. We are interested in the growth rate of $C(n)$ as a function of n. In particular, we propose:

QUESTION 1. *Is there a $\delta > 0$ such that $C(n) = \Omega(n^\delta)$?*

1.2 Connection to the Sensitivity Conjecture

Why consider such a strange game? The motivation is that the game provides a possible approach to the well

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ITCS'15, January 11–13, 2015, Rehovot, Israel.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3333-7/15/01 ...$15.00.
http://dx.doi.org/10.1145/2688073.2688096.

known *sensitivity conjecture* from boolean function complexity.

Recall that the sensitivity of an n-variate boolean function f at an input $\mathbf{x}$, denoted $s_f(\mathbf{x})$, is the number of locations ℓ such that if we flip the bit of $\mathbf{x}$ in location ℓ then the value of the function changes. (Alternatively, this is the number of neighbors of $\mathbf{x}$ in the hamming graph whose f value is different from $f(\mathbf{x})$.) The sensitivity of f, $s(f)$, is the maximum of $s_f(\mathbf{x})$ over all boolean inputs $\mathbf{x}$.

The degree of a function f, $deg(f)$, is the smallest degree of a (real) polynomial p in variables $x_1, \ldots, x_n$ that agrees with f on the boolean cube.

CONJECTURE 2. *(The Sensitivity Conjecture) There is a $\delta > 0$ such that for any boolean function f, $s(f) \geq \Omega(deg(f)^{\delta})$.*

An easy argument (given in Section 2) connects the cost function $C(n)$ of the game G_n to the sensitivity conjecture:

PROPOSITION 3. *For any boolean function on n variables, $s(f) \geq C(deg(f))$.*

In particular, an affirmative answer to Question 1 would imply the sensitivity conjecture.

1.3 Background on the Sensitivity Conjecture

Sensitivity and degree belong to a large class of complexity measures for boolean functions that seek to quantify, for each function f, the amount of knowledge about individual variables needed to evaluate f. Other such measures include decision tree complexity and its randomized and quantum variants, certificate complexity, and block sensitivity. The value of such a measure is at most the number of variables. There is a long line of research aimed at bounding one such measure in terms of another. For measures a and b let us write $a \leq_r b$ if there are constants C_1, C_2 such that for every total boolean function f, $a(f) \leq C_1 b(f)^r + C_2$. For example, the decision tree complexity of f, $D(f)$, is at least its degree $deg(f)$ and thus $deg \leq_1 D$. It is also known [8] that $D \leq_3 deg$. We say that a is *polynomially bounded* by b if $a \leq_r b$ for some $r > 0$ and that a and b are *polynomially equivalent* if each is polynomially bounded by the other.

The measures mentioned above, with the notable exception of sensitivity, are known to be polynomially equivalent. For example, in relating block sensitivity, $bs(f)$, to degree Nisan and Szegedy [9] show that $bs(f) \leq_2 deg(f)$. In the other direction, the bound $deg(f) \leq_3 bs(f)$ follows from a result in [1]. For a survey on many of these results, see [2]. The sensitivity conjecture asserts that $s(f)$ is polynomially equivalent to all of the measures mentioned in this section, and for this, it suffices to show that it is polynomially related to $deg(f)$.

There are a number of equivalent formulations of the sensitivity conjecture. For instance [4] give a graph theoretic formulation by exploring a different relationship between sensitivity and degree than what is presented here. The same graph theoretic question also appeared somewhat earlier in [3], however, sensitivity of boolean functions was only mentioned as a related problem and no direct connection was given. For a good survey of many other variations of the sensitivity conjecture, see [5].

The sensitivity conjecture perhaps more commonly appears as a question on the relationship between sensitivity and block sensitivity. For example, Nisan and Szegedy [9] asked specifically if $bs(f) = O(s^2(f))$ for all functions, and as of this writing no counterexample has been given. The best known bound relating sensitivity to another measure was given by Kenyon and Kutin [6]. They proved that $bs(f) \leq \frac{e}{2\pi} e^{s(f)} \sqrt{s(f)}$ for all boolean functions.

1.4 Outline of the Paper

In Section 2 we prove that a positive answer to Question 1 would imply the sensitivity conjecture. We also describe how protocols relate adverserial methods for proving that boolean functions are evasive (that is have decision tree complexity $D(f) = n$). At the end of the section we prove that it suffices to answer Question 1 for a special subset of protocols called *order oblivious protocols*.

In Section 3 we present three stronger variants of Question 1. We then show that for two of these variants, there are protocols that give negative answers to the questions, and suggest that Question 1 has a negative answer as well. However, these protocols satisfy a property called monotonicity and in Section 4 we prove an $\Omega(n^{1/2})$ lower bound on the cost of any monotone protocol, which shows that any protocol that gives a negative answer to Question 1, must look quite different from the two protocols that refuted the strengthenings. In the same section we prove a rather weak lower bound for a special class of protocols called assignment oblivious protocols. Finally, in Section 5 we give the construction of the lowest cost protocol that we know, whose cost is lower than that of the AND-OR protocol by a constant factor.

2. CONNECTION BETWEEN THE SENSITIVITY CONJECTURE AND THE GAME

In this section we prove Proposition 3, which connects the sensitivity conjecture with the two player game described in the introduction.

We will use $\mathbf{e}_\ell$ to denote the assignment in $\{0,1\}^n$ that is 1 in location ℓ and 0 elsewhere. Given two assignments $\mathbf{v}, \mathbf{w} \in \{0,1\}^n$ we will use $\mathbf{v} \oplus \mathbf{w}$ to denote the assignment for which each coordinate is the mod-2 sum of the corresponding coordinates in $\mathbf{v}$ and $\mathbf{w}$.

Recall that Alice's strategy gives the mapping from the input permutation σ and bit b to a boolean vector $\mathbf{v}$ and Bob's strategy maps the vector $\mathbf{v}$ to a subset of locations in $\mathbf{v}$. We first observe that for each strategy for Alice there is a canonical best strategy for Bob. For a permutation σ, we let $\Pi_A(\sigma)$ denote the vector Alice writes down after receiving $\sigma_1, \cdots, \sigma_{n-1}$ (so the location σ_n is still labeled with a $*$). Thus $\Pi_A(\sigma)$ can be viewed as an edge in the *hamming graph* $\mathbb{H}_n$ whose vertex set is $\{0,1\}^n$, with two vertices adjacent if they differ in one coordinate. The *edge set* $E(\Pi)$ of a protocol Π is the set of edges $\Pi_A(\sigma)$ over all permutations σ. This defines a subgraph of $\mathbb{H}_n$. Given Alice's output $\mathbf{v}$, the possible values for σ_n are precisely those locations ℓ that satisfy $(\mathbf{v}, \mathbf{v} \oplus \mathbf{e}_\ell)$ is an edge in $E(\Pi)$. Thus the best strategy for Bob is to output this set of locations. It follows that $c(\Pi)$ is equal to the maximum vertex degree of the graph $E(\Pi)$.

Proposition 3 will therefore follow by showing the following: Given a boolean function with degree n and sensitivity s, there is a strategy Π for Alice for the game G_n such that the graph $E(\Pi)$ has maximum degree at most s.

We need a few preliminaries. A *subfunction* of a boolean function f is a function g obtained from f by fixing some of the variables of f to 0 or 1. Note it is clear that if g is a subfunction of f then $s(f) \geq s(g)$. We say a function has *full degree* if $deg(f)$ is equal to the number of variables of f. We start by recalling some well known facts.

LEMMA 4. *For any boolean function f there exists a subfunction g on $deg(f)$ variables that has full degree.*

PROOF. If p is the (unique) multilinear real polynomial that agrees with f on the boolean cube, then p contains a monomial $\prod_{\ell \in S} x_\ell$ where $|S| = deg(f)$. Let g be the function obtained by fixing the variables in $[n] \setminus S$ to 0. Then g is a function on $deg(f)$ variables that has full degree. □

LEMMA 5. *Given a function f with full degree and a location ℓ, there exists a bit b such that the function obtained from f by fixing $x_\ell = b$ is also of full degree.*

PROOF. The polynomial (viewed as a function from $\{0,1\}^n \rightarrow \{0,1\}$) for f may be written in the form $p_1(x_1, x_2, \cdots, \not{x_\ell}, \cdots, x_n) + x_\ell p_2(x_1, x_2, \cdots, \not{x_\ell}, \cdots, x_n)$. Here $p_1(x_1, x_2, \cdots, \not{x_\ell}, \cdots, x_n)$ indicates that the variable x_ℓ is not an input to the polynomial. If p_1 has a non zero coefficient on the monomial $\prod_{k \neq \ell} x_k$, then we set $x_\ell = 0$ and the resulting function will have full degree. For the other case, note p_2 must have a non zero coefficient on $\prod_{k \neq \ell} x_k$ because f has full degree. Thus, setting $x_\ell = 1$ will work. □

We remark that the argument in the above lemma is essentially the same as the standard argument that the decision tree complexity of any function f is at least $deg(f)$.

We are now ready to prove Proposition 3.

PROOF. Given the function f, let g be a subfunction on $deg(f)$ variables with full degree. We will construct a protocol Π that satisfies $E(\Pi) \subseteq E(g)$, where $E(g)$ denotes the set of sensitive edges for the function g, i.e. the edges of $\mathbb{H}_n$ whose endpoints are mapped to different values by g. This will imply that $c(\Pi) \leq s(g) \leq s(f)$, and thus prove the proposition. As Alice receives $\sigma_1, \sigma_2, \cdots, \sigma_n$, she fills in $\mathbf{v}$ in such a way so that the function f restricted to the partial assignment written on $\mathbf{v}$ remains a full degree function, which is possible by Lemma 5.

Note that after Alice writes a bit in location σ_{n-1}, the function g restricted to $\mathbf{v}$ is now a non-constant function of one variable, and thus the edge $\Pi_A(\sigma)$ is a sensitive edge for the function g. This implies that $E(\Pi) \subseteq E(g)$.

□

Remark: To summarize, the reduction above shows that a degree n Boolean function having sensitivity s can be converted into a strategy for Alice for the game G_n of cost at most s. We don't know whether this connection goes the other way, i.e., we can't rule out the possibility that the answer to Question 1 is negative (there is a very low cost protocol for G_n) but the sensitivity conjecture is still true.

2.1 Connection to Decision Tree Complexity

We note the connection between protocols Π for the game G_n and boolean functions on n variables for which $D(f) = n$ (sometimes referred to as *evasive* functions). A common method for showing that a function is evasive is to use an *adversary argument*. For example, consider the evasive function

$$\text{AND-OR}(\mathbf{x}) = \bigwedge_{i=1}^{\sqrt{n}} \bigvee_{j=1}^{\sqrt{n}} x_{ij}.$$

To show this function is evasive we simulate the computation of some decision tree on an input $\mathbf{x}$, except when the tree queries a variable x_{ij} the adversary will respond either 0 or 1 in such a way as to keep the value of the function on the input $\mathbf{x}$ unknown until all variables are queried. For the AND-OR function, take the adversary that always answers 0 as long as some other variable in the corresponding OR block remains undetermined, otherwise it answers 1. This adversary is exactly Alice's part of the AND-OR protocol described in the introduction. For more examples of adversary arguments see [7].

Every evasive function by definition admits an adversary argument which in turn defines a protocol Π. In fact a function f is evasive if and only if there exists a protocol Π for which $E(\Pi) \subseteq E(f)$ (recall $E(f)$ is the set of sensitive edges of the function f). This work explores the question, can we use the inherent structure of an arbitrary adversary (or protocol) to exhibit a lower bound on sensitivity? We provide some limited evidence that this may be possible by proving lower bounds for restricted classes of protocols Π (see Section 4).

2.2 Order Oblivious Protocols

In the game G_n, at each step $i < n$, the value written by Alice at location σ_i may depend on her knowledge up to that step, which includes both the sequence $\sigma_1, \cdots, \sigma_i$ and the partial assignment already made to v at locations $\sigma_1, \ldots, \sigma_{i-1}$. A natural way to restrict Alice's strategy is to require that the bit she writes in location σ_i depend only on σ_i and the current partial assignment to v but not on the order in which $\sigma_1, \ldots, \sigma_{i-1}$ arrived. A protocol satisfying this restriction is said to be *order oblivious*. The following easy proposition shows that it suffices to answer Question 1 for order oblivious protocols.

PROPOSITION 6. *Given any protocol Π there exists an order oblivious protocol Π' such that $E(\Pi') \subseteq E(\Pi)$. In particular, $c(\Pi') \leq c(\Pi)$.*

PROOF. First some notation. Given a permutation σ let $\sigma_{\leq k}$ denote the prefix of the first k elements of σ. We let $\Pi_A(\sigma_{\leq k})$ denote the partial assignment written on $\mathbf{v}$ after Alice has been streamed $\sigma_1, \cdots, \sigma_k$.

We give a canonical way of obtaining an order oblivious protocol Π' from Π. We define Π' in steps, where step k refers to what Alice does when she is streamed σ_k. For step 1, when σ_1 arrives, she writes according to what Π does for that value of σ_1. In order to define step $k+1$, assume Π' is defined for the first k steps. Assume as well that it satisfies for every permutation σ, there is a permutation τ of $\sigma_1, \cdots, \sigma_k$ so that $\Pi_A(\tau) = \Pi'_A(\sigma_{\leq k})$.

Suppose σ_{k+1} arrives and the current state of the vector is $\mathbf{v} := \Pi'(\sigma_{\leq k})$. Note from $\mathbf{v}$ Alice can deduce the set of the first k elements of σ (it is the set of locations not labeled with a *). Alice then considers all permutations τ of $\sigma_1, \cdots, \sigma_k$ such that $\Pi_A(\tau) = \Pi'_A(\sigma_{\leq k})$ and picks the lexicographically smallest permutation (call it τ^*) in that set and writes on

location σ_{k+1} according to what Π does after τ^*. Note that the bit written on location σ_{k+1} does not depend on the relative order of $\sigma_1, \sigma_2, \cdots, \sigma_k$. Using this strategy, Alice maintains the invariant that for every permutation σ, there is a permutation τ of $\sigma_1, \cdots, \sigma_k$ so that $\Pi(\tau) = \Pi'(\sigma_{\leq k})$.

Thus, by construction, Π' is assignment oblivious. Also for any permutation σ there is a permutation τ for which $\Pi_A(\tau) = \Pi'_A(\sigma)$. This implies that $E(\Pi') \subseteq E(\Pi)$. □

3. STRONGER VARIANTS OF QUESTION 1

In this section we propose three natural variants of Question 1, and refute two of these variants by exhibiting and analyzing some specific protocols.

The cost function $c(\Pi)$ of a protocol is defined based on the worst case over all choices of $\sigma_1, \ldots, \sigma_n, b$. Alternatively, it is natural to evaluate a protocol based on the average size of the set Bob outputs, where the average is taken over a random permutation $\sigma_1, \ldots, \sigma_n$ and a random bit b. We call this the *expected cost of* Π and denote it by $\tilde{c}(\Pi)$. Let $\tilde{C}(n)$ denote the minimum expected cost of a protocol for G_n.

QUESTION 7. *Is there a $\delta > 0$ such that $\tilde{C}(n) = \Omega(n^\delta)$?*

An affirmative answer to this question would give an affirmative answer to Question 1.

We point out that it is well known that the natural probabilistic version of the sensitivity conjecture, where sensitivity is replaced by average sensitivity (where the average is taken uniformly over $\{0,1\}^n$) is trivially false (for example, for the OR function). For contrast, consider the protocol Π where Alice writes a 0 at each step. This protocol is closely related to the OR function in that Alice's part of this protocol is exactly the adversary argument used to prove that OR is evasive. Note also that $E(\Pi)$ is exactly the set of sensitive edges for the OR function. However, the average cost $\tilde{c}(\Pi)$ is $n/2$ whereas the average sensitivity of the OR function is $o(1)$. We currently know of no protocol Π for which $\tilde{c}(\Pi) = o(\sqrt{n})$.

We also remark that an analog of Proposition 6 holds for the cost function $\tilde{c}(\Pi)$, and therefore it suffices to answer the question for order oblivious protocols. (The proof of the analog is similar to the proof of Proposition 6, except when modifying the protocol τ^* is not selected to be the lexicographically smallest permutation in the indicated set, but rather the permutation in the indicated set that minimizes the expected cost conditioned on the first k steps.)

There is another natural variant of Question 1 based on average case. When we run a fixed protocol Π on a random permutation σ and bit b, we can view the vector $\mathbf{v}$ produced by Alice as a random variable. Let $\tilde{h}(\Pi)$ be the conditional entropy of σ_n given $\mathbf{v}$; intuitively this measures the average number of bits of uncertainty that Bob has about σ_n after seeing $\mathbf{v}$. It is easy to show that this is bounded above by $\log(c(\Pi))$. Let $\tilde{H}(n)$ be the minimum of $\tilde{h}(\Pi)$ over all protocols Π for G_n. The analog of Question 1 in this setting asks whether there is a positive constant δ such that $\tilde{H}(n) = \Omega(\delta \log(n))$? An affirmative answer to this would imply an affirmative answer to Question 1, however it turns out that the answer to this new question is negative.

THEOREM 8. *There is an order oblivious protocol Π for G_n such that $\tilde{h}(\Pi) = O(\log\log(n))$.*

Remark: Earlier we showed one can transform any protocol into an order oblivious protocol with smaller cost. However, it is not clear whether or not this transformation can increase $\tilde{h}$. Instead, we directly provide an example of an order oblivious protocol for which $\tilde{h}(\Pi)$ is small.

PROOF. Before defining the protocol Π we need some setup. Let $k = \lceil \log(n) \rceil$ and associate each integer $\ell \in [n]$ to its binary expansion, viewed as a vector $\mathbf{b}(\ell) \in \mathbb{F}_2^k$. Note that $0 \notin [n]$, and thus each vector $\mathbf{b}(\ell)$ is nonzero. Let $t > k$ be an integer (which we'll choose to be $\log^2(n)$) and for each $S \subseteq [n]$ of size t, let $Z(S)$ be a maximal subset of S such that $\sum_{\ell \in Z(S)} \mathbf{b}(\ell)$ is the 0 vector. Observe that by maximality, $Z(S) \geq |S| - k$ (otherwise $S \setminus Z(S)$ would have a linearly dependent subset which we could add to $Z(S)$). Finally let $\mathcal{H} = \{Z(S) : S \in \binom{[n]}{t}\}$.

Given $T \in \mathcal{H}$ and a partial assignment π, we say T is *compatible* with π if $\pi_i \in \{1, *\}$ for all $i \in T$. The protocol Π is defined as follows. For $i \neq n$ Alice writes a 0 on location σ_i unless doing so makes all $T \in \mathcal{H}$ not compatible with the resulting partial assignment written on $\mathbf{v}$, otherwise she writes a 1.

In an earlier version of this paper, we had defined Π so that Alice writes 1 on location i if and only if $i \in Z(S)$ where S is the set of the last t locations of σ. The cost of this protocol is easier to analyze, but it is not order oblivious. Here we instead analyze the order oblivious protocol you obtain if Alice writes a 0 as long as she remains consistent with some partial assignment in the order sensitive protocol.

We note two properties of Π. First, Alice will write a 0 on the first $n - t$ streamed locations. To see this, let $S(\sigma)$ denote the set of the last t elements of σ. Then $Z(S(\sigma))$ will be compatible with $\mathbf{v}$ for the first $n - t$ steps. We also have:

CLAIM 9. *There is a unique set $F \in \mathcal{H}$ that is compatible with the partial assignment $\Pi_A(\sigma)$.*

PROOF. Recall that $\Pi_A(\sigma)$ will have a $*$ in location σ_n. Suppose that there are two sets F_1, F_2 that are compatible with $\Pi_A(\sigma)$ and let T be their symmetric difference. First suppose $T - \{\sigma_n\}$ is non-empty and pick $i \in T - \{\sigma_n\}$. Then when location i arrived, Alice could have written a 0 since one of F_1 or F_2 would remain compatible. This contradicts the construction of the protocol. Now suppose that $T = \{\sigma_n\}$. In this case, since $\sum_{\ell \in F_1} \mathbf{b}(\ell) = \sum_{\ell \in F_2} \mathbf{b}(\ell) = \vec{0}$, the vector $\mathbf{b}(\sigma_n)$ must be the zero vector. This is also impossible because we defined the protocol to have all $\mathbf{b}(\ell)$ non-zero. □

We will refer to the set promised by Claim 9 as the *final set* and denote it as $F(\sigma)$.

We now obtain an upper bound on the conditional entropy of σ_n given $\mathbf{v}$. Let L be the random variable that is 1 if $\sigma_n \in F(\sigma)$ and 0 otherwise. We have:

$$\begin{aligned} H(\sigma_n|\mathbf{v}) &\leq H(\sigma_n, L|\mathbf{v}) \\ &= H(L|\mathbf{v}) + H(\sigma_n|\mathbf{v}, L) \\ &\leq 1 + H(\sigma_n|\mathbf{v}, L) \\ &= 1 + H(\sigma_n|\mathbf{v}, L=1)\Pr[L=1] \\ &\quad + H(\sigma_n|\mathbf{v}, L=0)\Pr[L=0] \end{aligned}$$

We first bound the second term. Note that given $L = 1$ we have that σ_n is in the final set $F(\sigma)$ and that Bob can deduce

$F(\sigma)$ given the vector $\mathbf{v}$. To see this, let W be the set of locations ℓ for which $\mathbf{v}$ is set to 1 and let $\Gamma = \sum_{\ell \in W} \mathbf{b}(\ell)$. If Γ is $\vec{0}$, then $F(\sigma)$ must be the set of locations that are set to 1. Otherwise Γ will be equal to $\mathbf{b}(\ell^*)$ for some unique ℓ^*, and $F(\sigma)$ is then the set of locations set to 1 union ℓ^*. In either case, the number of possible values for σ_n is no more than t and so the second term is at most $H(\sigma_n|\mathbf{v}, L=1) \leq \log(t)$.

To bound the third term we first show the following:

CLAIM 10. *The probability that $L = 0$ is at most k/t.*

Remark: This claim is very easy to see for the order sensitive version mentioned earlier ($L = 0$ is exactly the event that $\sigma_n \in S - Z(S)$). The fact that it still works for the order oblivious version seems quite intuitive because Alice writing some additional 0's should only help the probability. For completeness, we provide a rigorous proof of this below.

PROOF. Recall that $L = 0$ means that $\sigma_n \in S \setminus F(\sigma)$. As before let $\sigma_{\leq j}$ denote the prefix of the first j elements of σ and let $T(\sigma_{\leq j})$ denote the set of the first j elements of σ. Given a prefix τ of length $n-l$ we let $M(\tau)$ denote $\max_E |T(\tau) - E|$ where the max is over all sets E that are compatible with $\Pi_A(\tau)$. For integers l and m let $f(l,m)$ denote $\min_\tau (\Pr[L = 0|\sigma_{\leq n-t} = \tau])$ where the minimum is over all prefixes τ of length $n-\ell$ for which $M(\tau) = m$. We will show that $f(\ell, m) \leq m/\ell$ for all ℓ, m. In particular, since every $Z(S)$ has size at least $t-k$, showing that $f(t,k) \leq k/t$ will prove the claim. We proceed by induction on $\ell + m$. As a base case, it is easy to see that if $m = 0$ the probability is 0, and if $\ell = m$ then the probability is 1.

Let τ be any prefix of length $n-\ell$ for which $M(\tau) = m$ and suppose that $\sigma_{\leq(n-\ell)} = \tau$. Note that if Alice writes a 0 next, then $M(\sigma_{\leq(n-\ell+1)}) \leq M(\sigma_{\leq(n-\ell)}) - 1$. Also if Alice writes a 1 next, then $M(\sigma_{\leq(n-\ell+1)}) = M(\sigma_{\leq(n-\ell)})$. Let p denote the probability that Alice will write a 0 on location $\sigma_{n-\ell+1}$. Then $p \geq m/\ell$ (if there is exactly one set T that is compatible then $p = m/\ell$ and with additional sets the probability only increases). Thus

$$\begin{aligned} f(\ell,m) &\leq \Pr[L=0|\sigma_{\leq n-\ell} = \tau] \\ &\leq pf(\ell-1,m) + (1-p)f(\ell-1,m-1). \\ &\leq \frac{m}{\ell}\frac{m-1}{\ell-1} + \frac{\ell-m}{\ell}\frac{m}{\ell-1} \qquad \text{(by the I.H.)} \\ &= m/\ell \end{aligned}$$

□

Note that trivially $H(\sigma_n|\mathbf{v}, L=0) \leq \log(n)$, thus the claim implies that the third term is at most $\log(n) \cdot \frac{k}{t}$. By choosing $t = \log^2(n)$ the second term is $O(\log\log(n))$ and the third term is $O(1)$.

For our last variant, suppose Alice can communicate to Bob with a ternary alphabet instead of a binary alphabet. We will show that Question 1 is false in this setting. The setup is the same as before: Alice is streamed a permutation σ, only when σ_i arrives she may write a 0,1, or 2 on location σ_i in $\mathbf{v}$. When $b \in \{0,1,2\}$ arrives she is forced to write b at location σ_n. Bob sees $\mathbf{v}$ and has to output a set J which must contain σ_n. The cost is the maximum size of J for any σ and b.

THEOREM 11. *There is a protocol Π using a ternary alphabet that has cost $O(\log(n))$.*

PROOF. Let $t < n$ be a parameter to be chosen later (we will end up showing that the cost is less than t).

Alice begins by writing 0 on the first $n-t$ locations streamed to her. After this, Alice writes only 1's and 2's (as described below). Clearly if the final input b is not 0, Bob will see exactly t locations that are not labeled a 0 and know the last t elements. Consider then the case that $b = 0$. We'll show that Alice can write the 1's and 2's in such a way that Bob can then determine σ_n exactly. In what follows, a binary string will refer to a string of 1's and 2's.

Consider the graph defined on t element sets where two sets are joined if they have symmetric difference 2. The degree of this graph is trivially less than n^2 so it has a proper coloring with at most n^2 colors.

Now let us encode each of these colors by a binary string of length t. Write $E(c)$ for the encoding of color c. We want our encoding to have the following property: for any two colors c, d if you delete any single bit from the encoding of $E(c)$ (which leaves a $t-1$ bit string) and delete any single bit from the encoding of $E(d)$ then they are still different.

CLAIM 12. *There is such an encoding for $t = 5\log(n)$.*

PROOF. Consider the graph defined on binary strings of length t, where two strings s_1, s_2 are joined if there is a way of deleting a symbol from s_1 and a symbol from s_2 to arrive at the same string of length $t-1$. The degree of this graph is trivially less than $2t^2$, thus there is a proper coloring with at most $2t^2$ colors. Thus there is a color class of size at least $\frac{2^t}{2t^2}$ strings. If $t > 5\log(n)$ then there is a color class of size at least n^2. Picking n^2 strings in this color class will give us the desired encoding $E(c)$. □

After Alice writes the first $(n-t)$ 0's, she knows the final t positions denoted $j_1 < \ldots < j_t$. She determines the color c of that set and the encoding $E(c)$. She then writes the bits of $E(c)$ in the positions $j_1, \ldots, j_t$ (writing the bits in this order and not in the σ order of the last t elements).

If $b = 0$, Bob only sees $t-1$ of the bits. However, by the property of the encoding, this is enough to recover $E(c)$ and therefore c. Furthermore, knowing c and $t-1$ out of the last t elements, the property of the coloring allows Bob to recover the missing element, which is σ_n. This concludes the construction.

4. LOWER BOUNDS FOR RESTRICTED PROTOCOLS

In the previous section we formulated two stronger variants of Question 1 that turned out to be false. This may suggest that the original question is also false. In this section however, we will prove a lower bound which implies that any counterexample to Question 1 will need to look quite different from the two protocols provided in the last section.

An order oblivious protocol can be specified by a sequence of maps $A_1, \cdots, A_n$ where each A_i maps partial assignments on the set $[n]$ to a single bit. When location σ_i arrives, the bit Alice writes is $A_{\sigma_i}(\mathbf{v})$. For partial assignments α and β, we say that β is an *extension* of α, denoted as $\beta \geq \alpha$, if β is obtained by starting from α and possibly fixing more variables. An order oblivious protocol is *monotone* if

each of the maps $A_1, \cdots, A_n$ are monotone with respect to the extension partial order. That is, if $\beta \geq \alpha$ are partial assignments, then $A_i(\beta) \geq A_i(\alpha)$ for each i. As a remark, when running the protocol there may be assignments that are never written on $\mathbf{v}$, however defining each A_i to have domain all partial assignments is still valid and simplifies notation.

Both the AND-OR protocol described in the introduction and the protocol constructed in Theorem 8 are examples of monotone protocols. This definition easily generalizes to protocols on alphabets of size k, in which case the ternary protocol given in the previous section can be seen to be monotone. Our main result in this section is that monotone protocols on binary alphabets have cost $\Omega(\sqrt{n})$. In particular, Question 1 is true for such protocols. For the rest of the paper, all protocols will be on binary alphabets.

Before proving the theorem we'll need some new definitions. Recall that an edge $e \in \mathbb{H}_n$ may be written as a vector in $\{0, 1, *\}^n$ for which $e_\ell = *$ on exactly one location ℓ. We call this location ℓ the *free location* of that edge. We say two edges e, e' *collide* if $e_\ell = e'_\ell$ for all ℓ that is not a free location of either edge. Equivalently, two edges collide if they share at least one vertex (each edge collides with itself). Both of the lower bounds in this section will follow by finding an edge $e \in E(\Pi)$ that collides with m other edges in $E(\Pi)$. This implies at least one of the vertices in e has degree at least $m/2$ in the graph $E(\Pi)$, which in turn lower bounds the cost of the protocol.

Finally, given a permutation σ we will use $\ell <_\sigma k$ to denote that the element ℓ comes before the element k in σ.

THEOREM 13. *All monotone protocols have cost $\Omega(\sqrt{n})$.*

PROOF. Let Π be a monotone protocol.

For a permutation σ denote by $\text{bump}_k(\sigma)$ the permutation obtained from σ by "bumping" the element k to the end of σ and maintaining the same relative order for the rest of σ. For example, $\text{bump}_1(321654) = 326541$.

We let $w(\sigma)$ denote the vector $\Pi_A(\sigma)$ with the entries sorted in σ order. In other words, $w(\sigma)$ is the vector defined by $w(\sigma)_i = (\Pi_A)_{\sigma_i}$. Our proof follows by repeated application of the following:

CLAIM 14. *Let σ be any permutation and let τ be obtained from σ by performing some sequence of bumps on σ. Suppose that τ and $m < n$ satisfies the following:*

- *The elements $\tau_1, \tau_2, \cdots, \tau_m$ were never bumped.*
- *Alice originally wrote a 0 on the locations $\tau_1, \cdots, \tau_m$, that is $\Pi_A(\sigma)_{\tau_i} = 0$ for all $i \leq m$.*

Then $\Pi_A(\tau)_{\tau_i} = 0$ for all $i \leq m$. Equivalently, $w(\tau)$ begins with m 0's.

PROOF. The claim follows easily by induction on i. Suppose we have already shown that $w(\tau)$ begins with $(i-1)$ 0's. Let $\mathbf{v}(\sigma, k)$ denote the partial assignment written on $\mathbf{v}$ just before Alice receives the index k (here the reader should take care to distinguish this from the partial assignment just before Alice receives σ_k). Consider the partial assignment $\mathbf{v}(\tau, \tau_i)$. It follows from the first assumption and the inductive hypothesis that $\mathbf{v}(\sigma, \tau_i)$ is an extension of $\mathbf{v}(\tau, \tau_i)$. Thus, since Alice originally wrote a 0 on location τ_i, by monotonicity she continues to write a 0 on that location when being streamed τ (that is $\Pi_A(\tau)_{\tau_i} = 0$). □

Let σ be the permutation for which $w(\sigma)$ is lexicographically smallest.

CLAIM 15. *$w(\sigma)$ consists of a string of 0's followed by a string of 1's, followed by a single *.*

PROOF. Suppose for contradiction that there is a a 0 that comes after a 1, and let k be the least index such that $w(\sigma)_k = 1$ and $w(\sigma)_{k+1} = 0$. Let τ be obtained from σ by bumping all of the locations ℓ for which $\ell <_\sigma k$ and $\Pi_A(\sigma)_\ell = 1$. Let m denote the number of locations ℓ for which $\ell <_\sigma k$ and $\Pi_A(\sigma)_\ell = 0$. Then by Claim 14, $w(\tau)$ begins with $(m+1)$ 0's. This contradicts the choice of σ □

Let $n-t$ be the number of initial 0's in $w(\sigma)$ and $t-1$ be the number of 1's. For k between 1 and n, let $\tau^{(k)} = \text{bump}_k(\sigma)$. Let x be the assignment obtained from $\Pi_A(\sigma)$ by setting location σ_n (which is a *) to 1.

CLAIM 16. *The edges $\Pi_A(\tau^{(k)})$ and $\Pi_A(\sigma)$ intersect at the input x for all k among the last t elements of σ. In particular x has degree at least t in the graph $E(\Pi)$.*

PROOF. Fix k among the last t elements of σ. Clearly $w(\tau^{(k)})$ has the first $n-t$ bits 0, and so by the choice of σ all other locations in $w(\tau^{(k)})$ must be labeled 1. Thus $w(\tau^{(k)}) = w(\sigma)$. This means that the edges $\Pi_A(\sigma)$ and $\Pi_A(\tau^{(k)})$ agree at all locations except for σ_n and σ_k (which are the free location of the edges respectively). Since $\Pi_A(\sigma)_{\sigma_k} = \Pi_A(\tau^{(k)})_{\sigma_n} = 1$, the two edges meet at x. □

To conclude the proof of the theorem we will find an assignment y that has degree at least $(n-t)/(t+1)$ in the graph $E(\Pi)$.

CLAIM 17. *For k among the first $n-t$ elements of σ, $w(\tau^{(k)})$ has the first $n-t-1$ bits equal to 0, and has at most one 0 among the next t bits (and last bit *).*

PROOF. The fact that the first $n-t-1$ bits of $w(\tau^{(k)})$ are labeled 0 follows by directly by Claim 14.

Suppose for contradiction that there are at least 2 0's among the next t locations and denote the locations of the first and second 0 to be ℓ_1 and ℓ_2 respectively. Take all of the locations that are labeled 1 in $\Pi_A(\tau^{(k)})$ and bump them to the end and let this new permutation be ρ. Once again by applying Claim 14 we have $\Pi_A(\rho)_{\ell_1} = \Pi_A(\rho)_{\ell_2} = 0$. Thus $w(\rho)$ has the first $n-t+1$ locations set to 0 which contradicts the choice of σ. □

Now classify each of the first $n-t$ elements of σ into $t+1$ types $n-t, \ldots, n$. Element k is of type n if $w(\tau^{(k)})$ has t 1's. Otherwise $w(\tau^{(k)})$ has $(t-1)$ 1's, and the type of k is equal to the index j between $n-t$ and $n-1$ such that $w(\tau^{(k)})_j = 0$.

Some type occurs at least $m := (n-t)/(t+1)$ times, call it j^*, and let $k_1, k_2, \cdots, k_m$ be the m elements that are type j^*. For $1 \leq i \leq m$ let $y^{(i)}$ be the assignment obtained by taking the edge $\Pi_A(\tau^{(k_i)})$ and assigning the $*$ to 0.

CLAIM 18. *The assignments $y^{(i)}$ are all equal.*

PROOF. By the definition of the bump operation the permutations $\tau^{(k_i)}$ all have the same elements at positions

$n-t, n-t+1, \cdots, n-1$ (they have the same suffix with the exception of the last element). Since they are all of the same type it follows that the $y^{(i)}$ all agree on locations in the set $\{\tau^{(k_1)}(j) \mid j \in n-t, \cdots, n-1\}$. For all other locations, each $y^{(i)}$ is set to 0, thus they are the same assignment. □

Therefore there are m distinct edges in the graph $E(\Pi)$ that are incident with the assignment $y := y^{(1)}$. Thus y has degree at least $m = (n-t)/(t+1)$. This implies that cost of Π is at least $\max(t, (n-t)/(t+1)) = \Omega(\sqrt{n})$.

As demonstrated by the AND-OR protocol, Theorem 13 is tight up to a constant factor. We remark that the monotone protocols we consider here seem to have no general connection to the class of monotone boolean functions, and our result for monotone protocols seems to be unrelated to the easy and well known fact that the sensitivity conjecture is true for monotone functions.

We conclude this section with a lower bound for a second class of protocols. Although the lower bound is only logarithmic, we point out that proving a logarithmic lower bound for all protocols with a strong enough constant would imply new bounds relating degree and sensitivity.

For a permutation σ let $S_k(\sigma)$ denote the set of elements ℓ that satisfy $\ell <_\sigma k$. For example, if $\sigma = 321654$ then $S_1(\sigma) = \{2, 3\}$. We say a protocol is *assignment oblivious* if the bit written by Alice in location k only depends on the set $S_k(\sigma)$. Such protocols can be described by a collection of n hypergraphs $H_1, H_2, \cdots, H_n$, where each H_ℓ is a hypergraph with vertex set $[n] \setminus \{\ell\}$. When k arrives, Alice writes a 1 if and only if the set $S_k(\sigma)$ is in H_k.

THEOREM 19. *Every assignment oblivious protocol* Π *has* $c(\Pi) \geq \log_2(n)/2$.

PROOF. Let Π be an assignment oblivious protocol.

Given a permutation $\sigma = \sigma_1\sigma_2\cdots\sigma_n$ and $k \in [n]$ we define $\text{swap}_k(\sigma)$ to be the permutation obtained by swapping the positions of the elements k and σ_n within σ and keeping every other element in the same place. For example, $\text{swap}_3(654321) = 654123$. The lemma will follow by constructing a permutation σ such that that $\Pi_A(\sigma)$ and $\Pi_A(\text{swap}_k(\sigma))$ collide for each $k \in \{\sigma_{n-1}, \cdots, \sigma_{n-\lceil \log_2(n) \rceil}\}$

We build up such a σ in a greedy manner. We start with setting $\sigma_{n-1} = 1$. With σ_{n-1} fixed, the bit Alice writes in location 1 is completely determined by σ_n (and does not depend on the values we later choose for $\sigma_1, \cdots, \sigma_{n-2}$). This holds by the assignment oblivious property and because $S_1(\sigma) = \{\ell : \ell \neq 1, \sigma_n\}$. Let R_1 be the locations ℓ for which setting $\sigma_n = \ell$ results in Alice writing a 1 in location 1. At least one of $|R_1|, |R_1^c|$ are bigger than $\lceil (n-1)/2 \rceil$, let T_1 be that set. Now we fix σ_{n-2} to be any element in T_1.

Having fixed σ_{n-1} and σ_{n-2}, the bit Alice writes on location σ_{n-2} also only depends on the value of σ_n. Now let R_2 be the subset of indices j in T_1 such that setting $\sigma_n = j$ would cause Alice to write a 1 in location σ_{n-2}. At least one of $|R_2|, |R_2^c|$ are bigger than $\lceil (|T_1| - 1)/2 \rceil$, let $T_2 \subseteq T_1$ be that set. This process is iteratively repeated. At step i we set σ_{n-i} to be an arbitrary element of T_{i-1}. With $\sigma_{n-1}, \cdots, \sigma_{n-i}$ now fixed, the value written in location σ_{n-i} depends only on the value of σ_n. The set R_i is defined to be all such values of σ_n that result in Alice writing a 1 in location σ_{n-i} and $T_i \subseteq T_{i-1}$ is defined to be the larger of $|R_i|$ and $|R_i^c|$. We proceed until the set T_i has only one element in it, in this case we assign σ_n to be that element. This process will take at least $\lceil \log_2(n) \rceil$ steps. We then assign the remaining elements to $\sigma_1, \cdots, \sigma_{n-i-1}$ in an arbitrary order.

We now claim that $\Pi_A(\sigma)$ and $\Pi_A(\text{swap}_k(\sigma))$ collide for $k = \sigma_n, \sigma_{n-1}, \cdots, \sigma_{n-\lceil \log_2(n) \rceil}$.

CLAIM 20. *Let* $i < \lceil \log_2(n) \rceil$, *and let* $k = \sigma_{n-i}$. *Then* $\Pi_A(\sigma)_\ell = \Pi_A(\textit{swap}_k(\sigma))_\ell$ *for all* $\ell \neq k, \sigma_n$.

PROOF. Let $\sigma' = \text{swap}_k(\sigma)$. If $\ell <_\sigma k$ then $S_\ell(\sigma) = S_\ell(\sigma')$ and so Alice writes the same bit to location ℓ under both permutations.

Suppose that $\ell >_\sigma k$. Let j be such that $\sigma_{n-j} = \ell$. Note that $\sigma_{n-1} = \sigma'_{n-1}, \cdots, \sigma_{n-j} = \sigma'_{n-j}$. Recall that holding $\sigma_{n-1}, \cdots, \sigma_{n-j}$ fixed, the bit Alice writes at location ℓ depends only on the value of σ_n, and furthermore that bit is the same as for all settings of $\sigma_n \in T_j$. Since both σ_n and $\sigma'_n = k$ are in the set T_j, it follows that $\Pi_A(\sigma)_\ell = \Pi_A(\sigma')_\ell$. □

By the above claim, σ collides with $\text{swap}_k(\sigma)$ for at least $\lceil \log_2(n) \rceil$ values of k. Furthermore, at least one of the vertices in $\Pi_A(\sigma)$ has degree more than $\lceil \log_2(n)/2 \rceil$. This concludes the proof.

5. A PROTOCOL WITH LOWER COST THAN THE AND-OR PROTOCOL

In this section we present a construction of a protocol with $c(\Pi) \leq \sqrt{\frac{999}{1000}}\sqrt{n}$ which is the lowest cost protocol we know. The construction is a variant of the AND-OR protocol defined in the introduction.

Assume n and k are integers where $n-k$ is a perfect square. A set of assignments $\{\mathbf{x}_S \in \{0,1\}^n | S \in \binom{[n]}{k}\}$ is an (n,k)-*proper code* if the hamming distance between any $\mathbf{x}_S, \mathbf{x}_{S'}$ is at least $2\sqrt{n}$ and each $\mathbf{x}_S$ is 0 on the locations $i \in S$. Let $\{\mathbf{x}_S | S \in \binom{[n]}{k}\}$ be an (n,k)-proper code. We construct a protocol Π as follows: Alice writes 0 at locations $\sigma_1, \cdots, \sigma_k$. Alice then takes the set $S = \{\sigma_1, \cdots, \sigma_k\}$ and splits $[n] \setminus S$ into $\sqrt{n-k}$ disjoint blocks of size $\sqrt{n-k}$. When Alice continues and receives σ_j (for $k < j < n$) she writes the mod-2 sum of the bit b_j and the bit in location σ_j of $\mathbf{x}_S$, where b_j is 1 if σ_j is the last element in its block, and 0 otherwise.

We claim that upon receiving vector $\mathbf{v}$, Bob knows that the value of σ_n is one of $\sqrt{n-k}$ possible locations. First note that the vector $\mathbf{v}$ is within distance $\sqrt{n-k}$ of the vector $\mathbf{x}_S$, and thus Bob may decode $\mathbf{v}$ to learn the assignment $\mathbf{x}_S$ (and thus the set S as well). Consider the assignment $\mathbf{v} \oplus \mathbf{x}_S$ restricted to the locations outside of S. If the final bit b is 0, then exactly one of the $\sqrt{n-k}$ blocks will be all 0's. Bob can output J to be that block. If the final bit b is 1, then every block will have exactly a single 1 in it. Bob can output J to be the set of locations that are set to 1. In each case $|J| = \sqrt{n-k}$.

To conclude the construction of this protocol we prove the existence of an $(n, n/1000)$-proper code. Consider the following random code indexed by the sets $S \in \binom{[n]}{k}$: Each $\mathbf{x}_S$ is set to 0 on locations in S, and set to an independently and uniformly chosen random bit on locations outside of S. We claim that with nonzero probability this set is a proper code. The second property holds by definition, it only remains to check the pairwise distances of the code words.

Given sets S, S' let $E_{S,S'}$ be the event that $d(\mathbf{x}_S, \mathbf{x}_{S'}) < 2\sqrt{n}$. This may be upper bounded by the probability that $\mathbf{x}_S, \mathbf{x}_{S'}$ differ on less than $2\sqrt{n}$ locations in the set $[n] \setminus (S \cup S')$. This probability is exactly the probability that two random $n-|S\cup S'|$ bit strings are within distance $2\sqrt{n}$. Since $n-|S\cup S'| \geq n/2$ this probability is at most $\exp(-n/32)$ by a standard Chernoff bound. By a union bound the probability of any event $E_{S,S'}$ occurring is at most

$$\binom{n}{n/1000}^2 \exp(-n/32) < 1.$$

Thus with nonzero probability this is a proper code.

COROLLARY 21. *There is an $\epsilon > 0$ and a protocol Π for which $c(\Pi) \leq (1-\epsilon)\sqrt{n}$.*

6. ACKNOWLEDGEMENTS

We thank Ran Raz for helpful discussions. The first author was supported by NSF grant CCF 083727. The second author was supported in part by (FP7/2007-2013)/ERC Consolidator grant LBCAD no. 616787, a grant from Neuron Fund for Support of Science, and the project 14-10003S of GA ČR. The third author was supported by NSF grants CCF-083727 and CCF-1218711, and the Simons Foundation under award 332622.

7. REFERENCES

[1] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. De Wolf. Quantum lower bounds by polynomials. *Journal of the ACM (JACM)*, 48(4):778–797, 2001.

[2] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: a survey. *Theor. Comput. Sci.*, 288(1):21–43, 2002.

[3] F. R. Chung, Z. Füredi, R. L. Graham, and P. Seymour. On induced subgraphs of the cube. *Journal of Combinatorial Theory, Series A*, 49(1):180–187, 1988.

[4] C. Gotsman and N. Linial. The equivalence of two problems on the cube. *Journal of Combinatorial Theory, Series A*, 61(1):142–146, 1992.

[5] P. Hatami, R. Kulkarni, and D. Pankratov. *Variations on the Sensitivity Conjecture*. Number 4 in Graduate Surveys. Theory of Computing Library, 2011.

[6] C. Kenyon and S. Kutin. Sensitivity, block sensitivity, and ℓ-block sensitivity of boolean functions. *Information and Computation*, 189(1):43–53, 2004.

[7] L. Lovasz and N. E. Young. Lecture notes on evasiveness of graph properties. *arXiv preprint cs/0205031*, 2002.

[8] G. Midrijanis. Exact quantum query complexity for total boolean functions. *arXiv preprint quant-ph/0403168*, 2004.

[9] N. Nisan and M. Szegedy. On the degree of boolean functions as real polynomials. *Computational Complexity*, 4:301–313, 1994.