Topics in Communication Complexity (working title)

May 13, 2006

Contents

1 Introduction

Provide a brief introduction to the study of general complexity.

1.1 Communication Complexity

Provide an introduction to the study of communication complexity

1.2 Players in Communication Games

Discuss the impact of the number of players on the complexity of a communication game. Begin with Yao's two-player model and then discuss the k player scenario. Mention input models for problems involving more than two players (in particular, address the number on the forehead model). Also address models for how players communicate (in particular, the ”‘blackboard”’ / ”‘broadcast”’ model).

1.3 Rounds

Define what a round is in a commmunication game, and talk about the importance of which player speaks first, and the order in which players speak.

1.4 One-way multiparty protocols and ACC circuits

Talk about [2].

2 The Pointer Jumping Problem

Define the pointer jumping problem (k players, 1 round). Explain why the problem is easier to solve with more players (and thus harder to prove lower bounds).

2.1 Bounds for Multiparty Pointer Jumping in the Conservative Model

Give the relationship between pointer jumping and tree labeling given by [3] on page 53 (using a reduction on T_k to prove a lower bound on the randomized complexity for k-round 2-party pointer jumping).

2.2 Communication Complexity with Help

Here introduce Babai et al.'s notion of communication with help and the Lemma they use in proving their results on BHV and 3party pointer jumping.

2.3 Lower Bounds on the Communication Complexity of the Bits of Hash Value Function

Give the result proved by Nisan and Wigderson [4] that is restated by Babai et al. [1] on the bits of hash value function and explain how it is related to pointer jumping.

2.4 Lower Bounds on the Communication Complexity of the Three-party Pointer Jumping Problem

Give the result proved by Babai et al. [1] on three-party pointer jumping function. Explain it using all the ideas we introduced in discussing the four-party problem like "‘near"’ and "‘far"’ points. Eliminate the discussion of the near-Hamming ball.

3 Lower Bounds on the Communication Complexity of the Four-party Pointer Jumping Problem

Many of the definitions we introduce in the discussion of 3-party pointer jumping can be reused here. Since I don't have that written yet, I will write them all here.

I will assume that we will have already proved the result from Babai et al. [1] for communication complexity with help. I will refer to this result as Lemma 2.x.

Define the four-party pointer jumping function.

We can construct the four-party pointer jumping problem as follows. There are five layers of nodes in a directed graph containing $m_0, m_1, \ldots, m_4$ nodes respectively. As in k-party pointer jumping, the first layer in the graph contains only one node, and the last only two. Thus, $m_0 = 1$ and $m_4 = 2$. The four-party pointer jumping function takes as input the four-tuple (f_1, f_2, f_3, f_4), which are the edges in the directed graph described above, where $f_i : [m_{i-1}] \to [m_i]$ and outputs their composition $f_4 \circ f_3 \circ f_2 \circ f_1$. Computing the function is equivalent to following the pointers from the first layer in the graph to the last layer. We are interested in computing the one-way communication complexity of the four-party pointer jumping function.

(Is this relation to the graph helpful?)

There are four players in the four-party pointer jumping function. The input model used is the number on the forehead model in which players see all inputs except their own. Thus, for example, Player 2 will receive as input the tuple (f_1, f_3, f_4). Players communicate according to the blackboard model in which all players can see all of the messages that are sent. Players communicate in a one-way fashion. Player 1 speaks first, followed by Players 2,3, and 4 in that order.

Limit the input model.

In our consideration of the communication complexity of the four-party pointer jumping problem, we will limit the input model by restricting players to see only the inputs of players ahead of them in the speaking order in the "forward looking" input model. Player 1 sees (f_2, f_3, f_4), Player 2 sees (f_3, f_4), Player 3 sees f_4, and Player 4 does not see any inputs. This makes the pointer jumping function slightly harder for the players to compute, and thus it is slightly easier to lower bound its communication complexity. Though the bound for four-party pointer jumping with the number on the forehead model is more interesting than with the foreward looking model, we believe techniques our techniques may be an important step toward finding a lower bound with the number on the forehead model.

(It seems like this foreward model isn't what we really want. Really the only limitation is that Player 3 does not see the input f_2. The foreward model eliminates the relevance of shifting and kills the construction relating four-party pointer jumping to three-party composition.)

State the theorem.

Theorem 1 *For the 4-party pointer jumping function in which Player 3 does not see* f_2*, where* $n = m_3/1000 = m_2 = (m_1)^2$*, the one-way communication complexity is* $\Omega(\sqrt{n})$*.*

Sketch the structure of proof.

Proof:

Building upon the techniques presented in [1], our proof proceeds by relating a three-party composition function with help to the four-party pointer jumping function, and then showing that in order to solve the problem with fewer than some amount of communication, at least one of several conditions, each of which would lead to a contradiction, must be true.

The three-party composition function we wish to consider takes as input the three tuple (f_2, f_3, f_4), and outputs their composition: $f_4 \circ f_3 \circ f_2$. Note that because f_2 takes m_1 inputs and f_4 has two possible output values, it takes m_1 bits to describe a composition of this sort. There is no longer an input f_1. Communication occurs with Player 1 acting as Helper, speaking first by sending the help message, and then communication proceeds in a one-way fashion.

Suppose we are given a protocol of cost less than $\sqrt{n}/k_1$ (I will plug in an actual constant for k_1 later) for the four-party pointer jumping problem. We can use this protocol to construct a new protocol for the three-party composition function with help that uses at most n/k_1 communication bits. We can do this as follows. Player 1 sends as a help message the same message he would have sent in the protocol for four-party pointer jumping. Then, for each $1 \leq i \leq m_1$, Player 2 sends the message he would have sent if $f_1(1) = i$ given Player 1's message in the original protocol. Likewise, Player 3 then sends for each $1 \leq i \leq m_1$ the message she would have sent if $f_1(1) = i$ given Player 1 and 2's messages in the original protocol. Player 4 now has enough information to compute $f_4 \circ f_3 \circ f_2(i)$ for each $i \in [m_1]$, and thus can output the desired composition (I feel like this construction is broken because in the forward looking input model, Players 2 and 3 wouldn't see Player 1's input in the pointer jumping problem!).

Give the shifting argument.

(What does shifting buy us if we don't allow Player 4 to see f_2, f_3 anyway? I will write this up assuming we will need it and that Player 4 does receive inputs f_2, f_3.)

We now present a shifting argument to reduce the size of the set of Player 4's output. This technique is similar to that presented in [1] for a similar purpose.

Recall that in the three-party composition function, Player 4 receives inputs (f_2, f_3) and outputs a function from $[m_1] \to [m_4]$. Thus, we can describe Player 4's output with the function $b(f_2, f_3) : [m_2]^{[m_1]} \times [m_3]^{[m_2]} \to [m_4]^{[m_1]}$. The number of such functions b is $\left([m_4]^{[m_1]}\right)^{[m_2]^{[m_1]}[m_3]^{[m_2]}}$. The shifting will reduce the size of this output space to $[m_4]^{[m_3]}$.

Before presenting the shifting argument, we first introduce some notation.

Notation 1 *m_1, m_2, m_3, m_4 are positive integers. f, γ, and g, g_0, g' are functions: $f \in [m_2]^{[m_1]}$, $\gamma \in [m_3]^{[m_2]}$, and $g, g_0, g' \in [m_4]^{[m_3]}$. $S, S_0, S' \subseteq [m_4]^{[m_3]}$. When we draw f, γ, or g at random, f is drawn with uniform probability from $[m_2]^{[m_1]}$, γ is drawn with uniform probability from $[m_3]^{[m_2]}$, and g is drawn with uniform probability from S. Finally, b is a function such that $b : [m_2]^{[m_1]} \times [m_3]^{[m_2]} \to [m_4]^{[m_1]}$. $p(b, S, f, \gamma) = \Pr_g[b(f, \gamma) = g \circ \gamma \circ f]$.*

Lemma 1 (Shifting, due to the shifting argument in [1]) *For all pairs (b, S), for all g_0, there exists S_0 such that $|S_0| = |S|$ and for all f, γ, $p(b, S, f, \gamma) \leq p(b_0, S_0, f, \gamma)$ where b_0 denotes the function $g_0 \circ \gamma \circ f$.*

We give a construction modified from that in [1] to prove the Lemma. The outer loop of the construction obeys the following invariant: at the end of the loop, for every f, γ, $p(b, S, f, \gamma)$ either increases or remains the same.

(How do I get the verbatim section to display with proper indenting? How do I get the math notation to work? This could be challenging.)

```
for j = 1 to m_3
S' = \emptyset
for g \in S
Define the function g' by g'(i) =
\cases{g_0(j) if i = j \cr
g(i) otherwise}
If g' \notin S, add g' to S'
else add g to S'
endfor
S = S'
Redefine b so that, for all f,\gamma
\cases{b(f,\gamma)(i) = g_0(j) when \gamma(f(i)) = j \cr
b(f,\gamma)(i) is unchanged otherwise}
endfor
```

Observe that in the end of each iteration of the outer loop, for each g that was in S, either a g or a g' is added to S'. Thus, every time the construction replaces S with S', the size of S remains the same. To see that at the end of each iteration of the outer loop, for every f, γ, $p(b, S, f, \gamma)$ either increases or remains the same, note the following for each iteration of the loop: **1.** For a given $j \in [m_3]$, we modified b so that the function it outputs behaves the same as the the composition $g_0 \circ \gamma \circ f$ does for all f, γ when the composition $\gamma \circ f$ would output j and **2.** For a given $j \in [m_3]$, if the inner loop changed the number of functions $g \in S$ that behave as g_0 does on j, that number could only have increased. This proves the Lemma. Note that this construction would still behave as desired if, for all f, γ, we set $b(f,\gamma)(i) = g_0(f(i))$, but we have written the construction as above so that it obeys the invariant used in proving the Lemma.

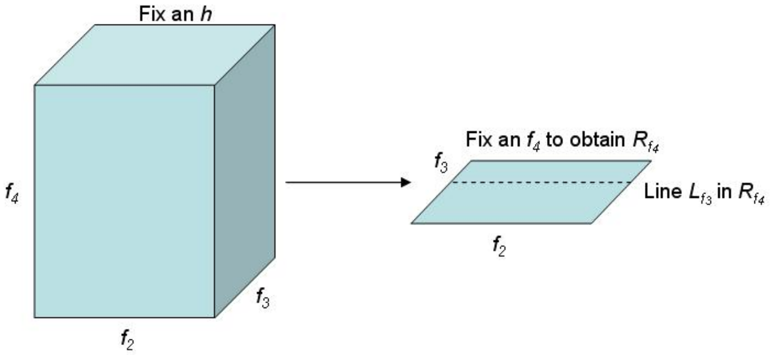
Outline the ideas of near and far points, good and bad lines, and define them mathematically.

We now return to consider the three-party composition function with help. In the protocol, denote the message h as the help message sent by Player 1, s as the message sent by Player 2, and t as the message sent by Player 3. Define the sets:

$S_{h,s} = \{(f_3, f_4)$: Player 2 sends s given $h\}$.

$T_{h,s,t} = \{f_4$: Player 3 sends t given $(h,s)\}$.

$Y_{h,s,t} = \{(f_3, f_4)$: Player 2 sends s given h, and Player 3 sends t given $(h,s)\}$.



(Maybe you should include cylinders S, T, and Y in diagram as well.)

The inputs to the three-party composition function can be thought of as composing a rectangular prism in three dimensions with the f_4 axis pointing towards the top of the page and the f_3 axis pointing into the page. (maybe a diagram would help here). The sets $Y_{h,s,t}$ form cylinder intersections within this prism and form a partition of the prism. If we shift these cylinder intersections according to the shifting construction given in the proof for (figure out how to use LaTeX reference to lemma) all with respect to the same function g (is this confusing? reader may not understand why we use the same function if it doesn't seem necessary), the function output by Player 4 will be constant within the shifted cylinder intersection. Define the function $b_{h,s,t}$ to be the function Player 4 outputs within the given cylinder intersection. Observe that due to the shifting argument, $b_{h,s,t}$ will be of the form $f_4 \circ f_3 \circ f_2$. For a given f_2, f_3, we define $g_{h,s,t} \in [m_4]^{[m_3]}$ to be the function that Player 4 composes with $f_3 \circ f_2$ to form $b_{h,s,t}$. Thus, for a given f_2, f_3, stating $g_{h,s,t}$ is sufficient to describe Player 4's output $b_{h,s,t}$.

The Hamming distance of functions a, b in the same family is the number of inputs on which a and b produce different outputs. We denote the Hamming distance of a, b as $d(a, b)$.

Consider a function $f_4 \in [m_4]^{[m_3]}$. Certain $g_{h,s,t}$ will have a large Hamming distance with respect to f_4, and certain $g_{h,s,t}$ will have a small Hamming distance with respect to f_4. Define $g_{h,s,t}$ to be *far* from f_4 if $d(f_4, g_{h,s,t}) \geq \delta m_3$ for $0 \leq \delta \leq 1$. $g_{h,s,t}$ is *near* to f_4 if $d(f_4, g_{h,s,t}) \geq \delta m_3$. That is, $g_{h,s,t}$ is far from f_4 if it disagree with f_4 on at least a δ fraction of the inputs, and it is near otherwise.

For a given h, if we fix an f_4, visually, this amounts to slicing the input prism horizontally to give a rectangle of dimensions $m_2 \times m_3$ (diagram would again be nice). Consider a cylinder intersection $Y_{h,s,t}$ sliced by this rectangle. Denote a point (f_3, f_2) in this rectangle (maybe helpful to name the rectangle R_{f_4} or something), as *correct* with respect to $Y_{h,s,t}$ if $g_{h,s,t} \circ f_3 \circ f_2 = f_4 \circ f_3 \circ f_2$ at (f_3, f_2).

Observe that because Player 3 does not see f_2, all cylinder intersections $Y_{h,s,t}$ extend fully across the f_2 axis of rectangles that slice them. Because shifting can only shift $Y_{h,s,t}$ vertically along the f_4 axis, this is true even after we shift the $Y_{h,s,t}$. Thus, if we consider an f_3 in this rectangle, all points along the line corresponding to f_3 must lie within $Y_{h,s,t}$ and therefore the Player 4's output along this line remains constant. To lower bound the communication complexity of the composition function, we wish to make it hard for players to compute the output, forcing them to use as much communication as possible. From the prover's perspective, a line in a rectangle R_{f_4} is *bad* if many points along the line are correct with respect to $Y_{h,s,t}$. Suppose f_4 has Hamming distance δm_3 from $g_{h,s,t}$ where $g_{h,s,t}$ is Player 4's output in $Y_{h,s,t}$. Observe that for a function f_3 that maps few of its inputs to the δm_3 points on which f_4 disagrees with $g_{h,s,t}$, the probability that a point (f_2, f_3) is correct on the the line $L_{f_3} \in R_{f_4}$ is high. More formally, we call a line $L_{f_3} \in R_{f_4}$ bad with respect to $Y_{h,s,t}$ if f_3 maps no more than $\frac{\delta m_2}{2}$ inputs to the δm_3 points on which f_4 disagrees with $g_{h,s,t}$. (Maybe this definition is still not formal enough).

Now we shift all cylinder intersections $Y_{h,s,t}$ as described in the proof for (cite shifting Lemma) with respect to some function $\hat{g} \in [m_4]^{[m_3]}$. We will arrive at a lower bound on the communication complexity of the four-party pointer jumping problem in which Player 3 does not see Player 2's input by giving upper bounds on the following quantities:

1. The number of functions f_4 near to $\hat{g}$.
2. For fixed f_4 far from $\hat{g}$, the number of correct points on all bad lines in R_{f_4} with respect to all cylinders $Y_{h,s,t}$.
3. For all f_4 far from $\hat{g}$, the number of correct points on all good lines with respect to all cylinders $Y_{h,s,t}$.

Prove near result.

Lemma 1 *With $\delta = 1/4$, the number of functions f_4 near to $\hat{g} < 2^{5m_3/6}(m_4 - 1)^{m_3/4}$.*

Proof: Observe that there are exactly $\binom{m_3}{i}(m_4-1)^i$ functions at Hamming Distance i from $\hat{g}$. This is because there are $\binom{m_3}{i}$ ways to choose the inputs that differ from $\hat{g}$, and for each of these choices, there are $(m_4 - 1)$ ways to differ on an input from $\hat{g}$, and i points that must differ. There is only one way to choose how the other inputs map to m_4, namely, the way that $\hat{g}$ maps them. Consequently, the number of near functions to $\hat{g}$ is less than $\sum_{i=0}^{m_3/4}\binom{m_3}{i}(m_4-1)^i$. From binary entropy, we have that $\sum_{i=0}^{m_3/4}\binom{m_3}{i} < 2^{nH(1/4)}$ where H is the binary entropy function. From [1], we have that $H(1/4) < 5/6$. Thus, the number of functions f_4 near to $\hat{g} < 2^{5m_3/6}(m_4-1)^{m_3/4}$.

Split far into good and bad.

Observe that there are no more than $m_4^{m_3}$ functions far from $\hat{g}$ because there are only $m_4^{m_3}$ functions $f_4 \in [m_4]^{[m_3]}$. Note that if we fix a function f_4 far from $\hat{g}$ and a cylinder intersection $Y_{h,s,t}$, we can partition the lines in R_{f_4} into two sets, one of which contains only lines that are bad with respect to $Y_{h,s,t}$, and one of which contains only lines that are good with respect to $Y_{h,s,t}$. We can denote these sets as $B_{f_4,h,s,t}$ and $G_{f_4,h,s,t}$ respectively. More formally, $B_{f_4,h,s,t} = \{f_3 : L_{f_3} \in Rf_4 \text{ is a bad line with respect to } Y_{h,s,t}\}$ and $G_{f_4,h,s,t} = \{f_3 : L_{f_3} \in Rf_4 \text{is a good line with respect to } Y_{h,s,t}\}$.

Use Chernoff to prove the upper bound on bad lines.

Lemma 1 *(I think I have to redefine far with the constant 1/4 replacing δ). For fixed f_4 far from $\hat{g}$ and fixed h, either the number of correct points in all bad lines in R_{f_4} with respect to all $Y_{h,s,t}$ is less than or equal to $2^{m_3}m_3^{m_2}m_2^{m_1}e^{-m_2/32}$ or the cost of the four-party pointer jumping function where Player 3... is $\Omega(\sqrt(n)$.*

Proof:

First we will apply the Chernoff bound to upper bound the probability that a line is bad. For a fixed f_4 and $g_{h,s,t}$ far from f_4, for $i \in [m_2]$, let $p_i = \Pr_{f_3} f_4(f_3(i)) \neq g_{h,s,t}(f_3(i))$. Note that because f_4 is far from g_h, s, t, for all i, $p_i \geq 1/4$. Let X_i be an indicator random variable for whether a randomly selected f_3 maps i to one of the points on which $g_{h,s,t}$ and f_4 disagree. Then $\Pr[X_i] = p_i$, and $\Pr[X_i = 0] = 1 - p_i$. $X = \sum_{i=1}^{m_2} X_i$. $E(X) = m_2/4$.

Applying the Chernoff bound, we see that:

$$\Pr[X < (1-1/2)\frac{m_2}{4}] \leq e^{-m_2(1/4)\frac{(1/2)^2}{2}} = e^{-m_2/32}$$

This bounds the probability that any randomly selected f_3 is bad in R_{f_4} with respect to $g_{h,s,t}$. Since we have a total of $m_2^{m_1}$ points on the line L_{f_3}, and a total of $m_3^{m_2}$ lines in the rectangle R_{f_4}, we can bound the number of correct points on bad lines in R_{f_4} with respect to $g_{h,s,t}$ by $m_3^{m_2}m_2^{m_1}e^{-m_2/32}$. For fixed h, there are at most $|(s,t)|$ different cylinder intersections $Y_{h,s,t}$ where $|(s,t)|$ denotes the number of possible message pairs Players 2 and 3 can send. Suppose that $|(s,t)| > 2^{m_3}$. By Lemma 2.x (cite the $f, \tilde{f}$ lemma), this would imply that the cost of the four-party pointer jumping function in which Player 3 does not see Player 2's input (find a shorter way of describing this) is $\Omega(m_3/m_1) = \Omega(\sqrt(n))$, proving the desired lower bound. If $|(s,t)| \leq 2^{m_3}$, we can bound above the number of correct points in all bad lines in R_{f_4} with respect to all $Y_{h,s,t}$ with $2^{m_3} m_3^{m_2} m_2^{m_1} e^{-m_2/32}$.

Bound the number of good lines with the total number of lines.

Lemma 1 *The number of correct points on all good lines for all* f_4 *far from* $\hat{g}$*, with respect to all cylinders* $Y_{h,s,t}$ *with fixed* h *is less than or equal to* $m_4^{m_3} m_3^{m_2} m_2^{m_1} e^{-m_1/8}$.

Proof:

Suppose that prior to shifting the cylinder intersections $Y_{h,s,t}$, all lines in all R_{f_4} are good lines with respect to some $Y_{h,s,t}$. Note that prior to shifting, the cylinder intersections cannot overlap and so a line in any R_{f_4} can only be inside a single cylinder. Consider any such line on fixed. Since the line is a good line, on at least $m_2/8$ inputs i to f_3, $f_4(f_3(i)) \neq g_h, s, t(f_3(i))$. Thus, for $j \in [m_1]$, $\Pr_{f_2}[f_4(f_3(f_2(j))) = g_h, s, t(f_3(f_2(j)))] \leq (1 - \frac{1/8m_2}{m_2} \leq e^{-m_2/8}$. To see this, note that each input to f_2 has a $frac1/8m_2m_2$ chance of mapping to a point that the composition $f_4 \circ f_3$ will carry to a different output than $g_{h,s,t} \circ f_3$, and that there are m_1 such outputs. Since there are $m_2^{m_1}$ points on a line and $m_4^{m_3} m_3^{m_2}$ lines over all R_{f_4}, prior to shifting the the cylinder intersections $Y_{h,s,t}$, the number of correct points on all good lines for all f_4 far from $\hat{g}$, with respect to all cylinders $Y_{h,s,t}$ is less than or equal to $m_4^{m_3} m_3^{m_2} m_2^{m_1} e^{-m_1/8}$. Observe that because the shifting can only shift lines with respect to f_4, and all f_4 we are considering are far from $\hat{g}$ this quantity cannot increase as a result of the shift. (This may be true, but explain why in greater detail).

We will now prove our main result using the bounds given above.

Restate the theorem and add everything up to get a lower bound to prove it.

Theorem 1 (Restatement of the main theorem) *For the 4-party pointer jumping function when Player 3 does not observe Player 2's input, where* $n = m_3/1000 = m_2 = (m_1)^2$*, the one-way communication complexity is* $\Omega(\sqrt{n})$.

Proof: Define the sets:

$X_(h,s,t) = \{(f_2, f_3, f_4)$: Player 1 sends help message h, Player 2 sends s, and Player 3 sends $t\}$. $Z_(h,s,t) = \{(f_2, f_3, f_4) : b_{h,s,t}(f_2, f_3) = f_4 \circ f_3 \circ f_2\} \cap Y_{h,s,t}$

Observe that $X_(h,s,t) \subseteq Z_(h,s,t)$. To see why this is the case, note that because the protocol is deterministic, on all inputs that case (h,s,t) to be sent, Player 4 must output the correct answer $b_{h,s,t}(f_2, f_3) = f_4 \circ f_3 \circ f_2$. However, if we fix a help message h, it is not required by the protocol that an input that causes the players to output $(s, t, b_{h,s,t}(f_2, f_3) = f_4 \circ f_3 \circ f_2)$ given h would actually cause the Helper to send h.

Consequently, $|X_(h,s,t)| \leq |Z_(h,s,t)|$. Observe that the sets $X_(h,s,t)$ partition the input space $[m_2]^{[m_1]} \times [m_3]^{[m_2]} \times [m_4]^{[m_3]}$. Thus, $\sum_{h,s,t} |X_(h,s,t)| = m_2^{m_1} m_3^{m_2} m_4^{m_3}$.

Observe that we can partition the sets $Z_{h,s,t}$ into sets $Z^{near}_{h,s,t}$ and $Z^{far}_{h,s,t}$ that contain those points in $Z_{h,s,t}$ that lie on planes near to and far from $\hat{g}$ respectively. Further note that we can partition the sets $Z^{far}_{h,s,t}$ into sets $Z^{far,bad}_{h,s,t}$ and $Z^{far,good}_{h,s,t}$ that contain those points in $Z^{far}_{h,s,t}$ on bad and good lines respectively. Then:

$$\begin{aligned}
m_2^{m_1} m_3^{m_2} m_4^{m_3} &\leq \sum_{h,s,t} |Z_(h,s,t)| \\
m_2^{m_1} m_3^{m_2} m_4^{m_3} &\leq \sum_{h,s,t} |Z^{near}_{h,s,t}| + \sum_{h,s,t} |Z^{far,bad}_{h,s,t}| + \sum_{h,s,t} |Z^{far,good}_{h,s,t}|
\end{aligned}$$

By Lemmas lem2, lem3, and lem4

$$\begin{aligned}
m_2^{m_1} m_3^{m_2} m_4^{m_3} \leq \sum_{h,s,t} 2^{5m_3/6}(m_4-1)^{m_3/4} m_3^{m_2} m_2^{m_1} &+ \sum_{h} (m_4^{m_3}) 2^{m_3} m_3^{m_2} m_2^{m_1} e^{-m_2/32} \\
&+ \sum_{h} m_4^{m_3} m_3^{m_2} m_2^{m_1} e^{-m_1/8}
\end{aligned}$$

Plugging in $m_1 = \sqrt{n}, m_2 = n, m_3 = n/1000$,

$$\begin{aligned}
2^{n/1000} \leq \sum_{h,s,t} 2^{n/1200} &+ \sum_{h} (2^{(n/1000)+1}) 2^{(-n/32)\log_2 e} \\
&+ \sum_{h} 2^{n/1000} 2^{-\sqrt{n}/8 \log_2 e}
\end{aligned}$$

In order for the inequality to hold, either the number of message tuples (h,s,t) is at least $2^{6n/5}/3$, the number of help messages h is at least $\frac{2^{n/32 \log_2 e}}{3 \times 2^{n/1000}}$, or the number of help messages h is at least $2^{\sqrt{n}/8 \log_2 e}/3$. This completes the proof.

Do I state anywhere here that after shifting, cylinder intersections can overlap? I am not sure that all of the $\hat{g}$ stuff is right. You use the word “points” too loosely. Be more formal in the statement of your lemmas. You probably want to state somewhere that most of these arguments rely on fixed h. Why do all the Lemmas and theorems appear as Lemma/Theorem 1? Get the terms in the sums and probabilities to be under the symbols rather than on in front of them. Is it clear from the quantities in the last paragraph that all of the situations lead to $\Omega(\sqrt{n})$ bound?

4 Future work

Here we discuss what remains to be studied with respect to the pointer-jumping problem.

References

[1] Laszlo Babai, Thomas P. Hayes, and Peter G. Kimmel. The cost of the missing bit: Communication complexity with help. *Combinatorica*, 21:455–488, 10 2001.

[2] Carsten Damm, Stasys Jukna, and Jiri Sgall. Some bounds on multiparty communication complexity of pointer jumping. *Comput. Complex.*, 7(2):109–127, 1998.

[3] Eyal Kushilevitz and Noam Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[4] Noam Nisan and Avi Wigderson. Rounds in communication complexity revised. *SIAM Journal on Computing*, 22:211–219, 1 1993.