Lower Bounds on Streaming Algorithms for Approximating the Length of the Longest Increasing Subsequence*

Anna Gál†
UT Austin.
panni@cs.utexas.edu

Parikshit Gopalan‡
Microsoft Research - Silicon Valley.
parik@microsoft.com

September 8, 2009

Abstract

We show that any deterministic data-stream algorithm that makes a constant number of passes over the input and gives a constant factor approximation of the length of the longest increasing subsequence in a sequence of length n must use space $\Omega(\sqrt{n})$. This proves a conjecture made by Gopalan, Jayram, Krauthgamer and Kumar [GJKK07] who proved a matching upper bound. Our results yield asymptotically tight lower bounds for all approximation factors, thus resolving the main open problem from their paper. Our proof is based on analyzing a related communication problem and proving a direct sum type property for it.

1 Introduction

The data-stream model of computation has been studied intensively in recent years, motivated by the design of better algorithms for massive data sets arising from settings such as the Internet, social and biological networks (see the surveys by Muthukrishnan [Mut05] and Babcock et al. [BBD+02] for an overview of this area). Much of this work on the algorithmic side has focused on designing highly efficient algorithms for simple problems such as estimating some statistics about the data. The seminal work of Alon, Matias and Szegedy demonstrated the surprising computational power of this model, by giving simple and efficient randomized algorithms to approximate various frequency moments [AMS96]. They also noted that techniques from communication complexity could be applied to prove unconditional lower bounds on the space complexity of such algorithms.

*A preliminary version of this paper appeared in FOCS'07
†Supported in part by NSF Grants CCF-0430695 and CCF-0830756
‡Work done while at UT Austin and the University of Washington

An area that has been extensively studied is that of estimating the sortedness of a data-stream. This is a natural problem that arises in several scenarios [GJKK07], and since efficient sorting is not possible in the data stream model, it is of interest to design highly efficient algorithms that are able to estimate sortedness of a large data set on the fly. Several possible measures of sortedness have been considered in the literature including inversions [AJKS02, GZ03], transpositions [CMS01], Edit distance a.k.a. Ulam distance [GJKK07] and the length of the longest increasing subsequence [LNVZ06, GJKK07, SW07].

Given a sequence σ of length n over an ordered alphabet $[m]$, an *increasing subsequence* in σ is a subsequence $i_1 < \cdots < i_k$ such that $\sigma(i_1) < \cdots < \sigma(i_k)$. Let $\mathrm{lis}(\sigma)$ denote the length of the *longest increasing subsequence* (LIS) in σ. A closely related quantity is the edit-distance from sortedness, denoted $\mathrm{ed}(\sigma)$ which is the number of edit operations needed to sort a sequence. It can be shown that $\mathrm{ed}(\sigma) = n - \mathrm{lis}(\sigma)$, since the best way to sort the input is to identify an LIS and insert all the other elements into this subsequence. The LIS is an important and well-studied object in its own right; see the books by Gusfield [Gus97] and Pevzner [Pev03] for applications in bioinformatics and the survey by Aldous and Diaconis [AD99]. There is a classical algorithm for computing the length of the LIS, known as *Patience Sorting*, which was first discovered in the context of card games [AD99]. Patience Sorting can naturally be viewed as a one-pass $O(n \log m)$ space data stream algorithm (though its discovery far predates the advent of the data-stream model). The problem of finding more space-efficient streaming algorithms for this problem has been well-studied.

Liben-Nowell, Vee, and Zhu [LNVZ06] showed an $\Omega(\sqrt{n})$ space lower-bound for computing $\mathrm{lis}(\sigma)$ exactly. Gopalan, Jayram, Kumar and Krauthgamer (hereafter referred to as GJKK) [GJKK07] and Sun and Woodruff [SW07] independently improved this lower-bound to $\Omega(n)$ even for randomized algorithms. This shows that Patience Sorting is optimal for exact computation of $\mathrm{lis}(\sigma)$ and to get a better space bound, one needs to settle for approximation. We say that an integer ℓ is a $1+\varepsilon$-approximation of $\mathrm{lis}(\sigma)$, if $\mathrm{lis}(\sigma) \leq \ell \leq (1+\varepsilon)\,\mathrm{lis}(\sigma)$. Such an algorithm was given by GJKK, who present a deterministic one-pass streaming algorithm which delivers a $1+\varepsilon$-approximation of $\mathrm{lis}(\sigma)$ using space $O(\sqrt{\frac{n}{\varepsilon}} \log m)$ for any $\varepsilon > 0$. They conjectured a $\Omega(\sqrt{n})$ lower bound for getting a $1+\varepsilon$ factor approximation for some constant $\varepsilon > 0$.

The algorithm of GJKK is obtained by first showing an upper bound of $t \log m$ on the maximum communication by any player in a t-player one-way communication protocol for estimating the length of the LIS, when the string is broken into t blocks, and one block is given to each player. Their algorithm simulates the protocol for $t = \sqrt{n}$ where n is the length of the input sequence.

The best lower-bound known before our results for the space requirement of streaming algorithms for approximating the length of the LIS was $\Omega(\frac{1}{\varepsilon} \log m)$ for $(1+\varepsilon)$ approximation, which comes from analyzing the above communication problem with two players, and holds for randomized algorithms with a constant number of passes [SW07]. GJKK were able to prove their conjecture for a restricted class of algorithms, in a model where the bit-sizes of the input are not taken into account. However, their proof only works when $m \geq 2^{\sqrt{n}}$, thus in the standard model, their bound is not better than the $\Omega(\log m)$ lower bound.

1.1 A communication problem

We obtain our bounds by analyzing a related communication problem that was proposed by GJKK. The problem involves t-players $P_1, \ldots, P_t$, where player P_i holds a number $x_i \in [m]$. Their goal is to decide whether the sequence $\mathbf{x} = (x_1, \ldots, x_t)$ has $\mathrm{lis}(\mathbf{x}) = 1$ or if it has $\mathrm{lis}(\mathbf{x}) \geq \varepsilon t$. We refer to this as the *primitive* problem. We then consider the OR of ℓ independent instances of this primitive problem, and the goal is to decide if any of the ℓ instances has $\mathrm{lis} \geq \varepsilon t$ or all have $\mathrm{lis} = 1$.

GJKK showed (see Lemma 4.4 in [GJKK07]) that a lower bound on the the maximum communication by any player in a t-player one-way communication protocol for the above problem gives a lower bound on the space used by any one pass deterministic streaming algorithm that gives a $(1 + \varepsilon)$ approximation of the length of the LIS.

A straightforward upper bound for the above problem is to run the protocol for each instance of the primitive problem separately. The hope for the conjecture is that the problem has some kind of direct sum property, and one cannot do much better than the above straightforward solution.

Direct sum problems in communication complexity have been studied in many scenarios, and they can be very useful [KN97]. However, in many cases it is hard to prove that a direct sum type property holds. Direct sum properties of problems that consist of taking the OR of independent copies of some primitive problem have been studied in several contexts, and the set disjointness problem is an important example of a problem of this type. [SS02] and [BYJKS04] prove direct sum properties for these type of problems of certain measures related to information complexity, introduced in [CSWY01]. These direct sum results yield strong lower bounds on the randomized multi-party communication complexity of set disjointness [SS02, BYJKS04, CKS03] and on the space complexity of randomized streaming algorithms for several related problems.

1.2 Our results

We prove the conjecture made by GJKK, and obtain asymptotically tight lower bounds for all approximation factors, thus resolving the main open problem from their paper.

Theorem 1.1. *For any $\varepsilon > 0$, any deterministic data-stream algorithm that makes R passes over the data and computes a $1 + \varepsilon$-approximation of $\mathrm{lis}(\sigma)$ in a sequence σ of length n over an alphabet $[M]$ requires space $\Omega\left(\frac{1}{R}\sqrt{\frac{n}{\varepsilon}}\log(\frac{M}{\varepsilon n})\right)$.*

Theorem 1.1 essentially matches the upper bound of $O(\sqrt{\frac{n}{\varepsilon}}\log M)$ proved by GJKK for one-pass algorithms for all $\varepsilon > 0$. It shows that no substantial improvement in their upper bound is possible even if the algorithm were allowed to make a constant number of passes. We should note that in our lower bound ε can take any value greater than 0, it can be a large constant or even $\omega(1)$. If we take $\varepsilon = \frac{1}{n}$ which corresponds to exactly computing the length of the LIS, we recover the linear lower bound.

Our proof is based on analyzing the above communication problem. In communication complexity, the standard techniques usually give bounds on the *total communication*, e.g.

the total length of the messages sent by all the players. For obtaining space lower bounds for streaming algorithms, one needs to estimate the *maximum communication*, that is the maximum (over all players) length of messages sent by the individual players. The standard approach to showing lower bounds for maximum communication is to show a lower bound on total communication and then divide by the number of players [BYJKS04, CKS03]. However, we use tools that directly address the maximum communication.

As mentioned above, the hope for the conjecture of GJKK was that the problem should have a direct sum property. We indeed show such a direct sum type property for the maximum communication. We are not aware of previous results that prove direct sum results for maximum communication. One of the difficulties in obtaining such results is that a direct sum result on total communication does not necessarily imply direct sum results on maximum communication. In fact we show that for a slight variant of the primitive problem, a direct sum type property holds for the total communication, but not for the maximum communication.

Finally, the kind of communication model considered plays a crucial role in this problem. In this paper we only consider the "Number In Hand" (NIH) model, where the players inputs are disjoint. Within the NIH model, there are two models of multi-party one-way communication protocols studied in the literature. In both models, the players send messages only once, in the order $P_1, \ldots, P_t$. In the first model, called the *private messages* model, player P_i sends a message to player P_{i+1} which is visible only to P_{i+1}. In the *blackboard* model, the players write their messages on a blackboard that is visible to all players. In order to show lower bounds for streaming algorithms, it suffices to consider the private messages model. Many known lower bounds hold for the blackboard model, which is clearly stronger [BYJKS04, CKS03]. Our lower bound comes by considering a promise version of the primitive problem for which a direct sum type property holds in the private messages model, but not in the blackboard model. In this paper we apply lower-bound techniques that can exploit this difference.

Our communication complexity results give further insights into direct sum type properties and why such theorems are hard to prove. Our results highlight the fact that having a strong direct sum property is sensitive to issues such as the type of communication (total versus maximum) and the type of model (blackboard versus private messages). Along the way, we obtain a separation between the private messages and blackboard models of deterministic one-way communication, this is the first such result to our knowledge.

1.3 Related Work

Independently of our work, Ergun and Jowhari [EJ08] subsequently gave a different proof of the conjecture of GJKK. Their proof is based on analyzing a different communication problem, where estimating total communication is sufficient to prove the conjecture. Their lower bound is proved in the blackboard model. Note however that while our methods give tight bounds for all approximation factors, this is not the case in [EJ08].

2 Preliminaries and Notation

We will consider the one-way multi-party communication model with t players $P_1, \ldots, P_t$ who are given inputs $x_1, \ldots, x_t$ respectively. Their goal is to compute the function $f(x_1, \ldots, x_t)$. The players communicate in the order $P_1, \ldots, P_t$. We will consider two versions of the model, the private messages model and the blackboard model. Most of our results will be for the former.

We first define single round one-way protocols in the private messages model, At step 1, player P_1 sends a message $M_1 = M_1(x_1)$ to player P_2, which is a function of its input x_1. At step i, player P_i sends a message $M_i = M_i(x_i, M_{i-1})$ to player $i+1$, which depends on its input x_i and the message M_{i-1} received from player $i-1$. At step t, player P_t must output $M_t(x_t, M_{t-1}) = f(x_1, \ldots, x_t)$. Thus a protocol $\mathcal{P}$ for f must specify the functions $M_1, \ldots, M_t$ for each player. The total and maximum communication complexity of the protocol $\mathcal{P}$ are defined as

$$\mathrm{CC}_t^{\mathrm{tot}}(\mathcal{P}) = \max_{x_1,\ldots,x_t} \sum_{i=1}^{t} |M_i|,$$
$$\mathrm{CC}_t^{\max}(\mathcal{P}) = \max_{x_1,\ldots,x_t} \max_{i=1}^{t} |M_i|$$

The total communication complexity of f denoted $\mathrm{CC}_t^{\mathrm{tot}}(f)$ is defined as the minimum of $\mathrm{CC}_t^{\mathrm{tot}}(\mathcal{P})$ over all protocols $\mathcal{P}$ for computing f. The maximum communication complexity of f denoted $\mathrm{CC}_t^{\max}(f)$ is defined similarly.

Next we define single round protocols in the blackboard model. In the blackboard one-way model, every player writes his messages on a blackboard where it is seen by all the players. Thus we now have $M_i = M_i(x_i, M_{i-1}, \ldots, M_1)$. In this model we denote total and maximum one-way communication complexity by $\mathrm{BB}_t^{\mathrm{tot}}(f)$ and $\mathrm{BB}_t^{\max}(f)$ respectively. Note that in both one-way models, the message M_i is completely determined by the prefix $x_1, \ldots, x_i$ of the input. Hence $M_i(x_1, \ldots, x_i)$ is well defined.

We will consider multi-round protocols in the private messages model with the goal of obtaining lower bounds for multi-pass data stream algorithms. During each round $r \leq R$, player P_i sends a message to P_{i+1} for $i \leq t-1$. At the end of round $r < R$, player P_t sends a message to P_1. In the last round R, player P_t is required to output the outcome of the protocol. The main difference from 1-round protocols is that in round $r \geq 2$, each players message can depend on the entire input - through the messages the player received - and not just its prefix. We use $M_i^r(x_1, \ldots, x_t)$ to denote the message sent by P_i in round r. Note that

$$M_1^r(x_1, \ldots, x_t) = M_1^r(x_1, M_t^{r-1}(x_1, \ldots, x_t), \ldots, M_t^1(x_1, \ldots, x_t)),$$

and for $i \geq 2$

$$M_i^r(x_1, \ldots, x_t) = M_i^r(x_i, M_{i-1}^r(x_1, \ldots, x_t), \ldots, M_{i-1}^1(x_1, \ldots, x_t)).$$

We use $\mathrm{CC}^{\mathrm{tot}}_{t,R}(f)$ and $\mathrm{CC}^{\max}_{t,R}(f)$ respectively to denote the total and maximum communication complexity of R round protocols.

Finally, we consider the unrestricted blackboard model, where players may communicate in any order, any number of times. We use the standard notation $\mathrm{D}_t(f)$ and $\mathrm{N}_t(f)$ to denote the (total) deterministic and non-deterministic t-party communication complexity of f.

We will use the extension of the notion of combinatorial rectangles [KN97] to multi-party NIH communication. A combinatorial rectangle is a set of the form $S_1 \times \cdots \times S_t$ where S_i is a subset of inputs to player P_i. As in the two party case, the sets of inputs that have the same communication transcript under the protocol form combinatorial rectangles.

A Data Stream Algorithm receives an input of size n, where n is thought of as very large. We think of this input as either written on an external memory device or obtained on the fly from some source such as a sensor network. The algorithm is allowed to make only one pass over its input or a few passes, while using very little storage space and update time per element. We would like both these quantities to be sub-linear in n, ideally we would like them to be of the order $(\log n)^{O(1)}$. Typically, in order to solve problems in this highly restricted model, we need to settle for randomization and/or approximation.

We use $\mathbf{x} \in [m]^t$ to denote a vector $\mathbf{x} = (x_1, \ldots, x_t)$. Given $\mathbf{x} \in [m]^{t\ell}$, we will also view it as a matrix of dimension $t \times \ell$ in $[m]^{t\times\ell}$. With this matrix view, we let $R_i(\mathbf{x})$ and $C_j(\mathbf{x})$ denote the i^{th} row and j^{th} column of the matrix respectively. We use $\mathbf{x} \circ \mathbf{y}$ to denote string concatenation. We use $\mathbf{x}^\ell$ to denote $\mathbf{x} \circ \mathbf{x} \cdots \circ \mathbf{x}$ ℓ times. For $\mathbf{x} \in [m]^t$, let $\mathrm{lis}(\mathbf{x})$ denote the length of the longest increasing subsequence in $\mathbf{x}$. Let $\mathrm{lis}_a(\mathbf{x})$ to denote the length of the longest increasing subsequence in $\mathbf{x}$ that ends with a value at most a. Thus $\mathrm{lis}(\mathbf{x}) = \mathrm{lis}_m(\mathbf{x})$. In some places, we will omit floors and ceilings to improve readability.

3 Lower Bounds for a Special Case

We start by analyzing the communication problem with the following primitive problem.

Definition 1. *Given* $\mathbf{x} \in [m]^t$, *let*

$$h(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathrm{lis}(\mathbf{x}) = 1 \\ 1 & \text{if } \mathrm{lis}(\mathbf{x}) = t. \end{cases}$$

Thus $h(\mathbf{x})$ is a promise problem which is 0 if $\mathbf{x}$ is a non-increasing sequence, 1 if it is an increasing sequence, and can be arbitrary otherwise.

First we give a lower bound on the communication complexity of this problem that holds even in the unrestricted blackboard model, and even for nondeterministic communication complexity.

Lemma 3.1. *For the function* h, $\mathrm{N}_t(h) \geq \log(\frac{m}{t-1})$.

Proof: The proof is via a fooling set argument. For each $a \in [m]$, let a^i be a repeated i times. Clearly $h(a^t) = 0$. We claim that in any protocol for f, at most $t-1$ such inputs can

lie in a single combinatorial rectangle. Assume for contradiction that $a_1^t, \ldots, a_t^t$ share the same rectangle where $a_1 < \ldots < a_t$. By the definition of combinatorial rectangles, the input $\mathbf{a} = (a_1, \ldots, a_t)$ also lies in the same rectangle, however $h(\mathbf{a}) = 1$. This shows that there are at least $\frac{m}{t-1}$ distinct 0-rectangles which implies $\mathrm{N}_t(h) \geq \log(\frac{m}{t-1})$. ∎

There is a matching upper bound that holds for $\mathrm{BB}_t^{\mathrm{tot}}(h)$ in the blackboard one-way model.

Claim 3.2. *For the function h, $\mathrm{BB}_t^{\mathrm{tot}}(h) \leq \lceil \log(\frac{m}{t-1}) \rceil$.*

Proof: Let $y_i = \lceil \frac{x_i}{t-1} \rceil \in [1, \lceil \frac{m}{t-1} \rceil]$. If $h(\mathbf{x}) = 1$, then $x_1 < \cdots < x_t$, so $x_1 + t - 1 \leq x_t$, so $y_1 < y_t$. If $h(\mathbf{x}) = 0$ then $x_1 \geq x_t$ so $y_1 \geq y_t$. So P_1 writes y_1 on the board, from which P_t can compute h. This takes $\lceil \log(\frac{m}{t-1}) \rceil$ bits. ∎

While it is clear that $\mathrm{CC}_t^{\max}(h) \geq \frac{\mathrm{N}_t(h)}{t} \geq \frac{1}{t} \log(\frac{m}{t-1})$, this bound is not as strong as we would like. In fact, we can improve the lower bound by a factor of $t - 1$, yielding a tight bound. This improvement is important for obtaining the desired bounds on streaming algorithms. We also obtain a slight improvement for $\mathrm{CC}_t^{\mathrm{tot}}(h)$ over the bound that follows from Lemma 3.1. Our proof technique will crucially use the fact that the messages are private.

Lemma 3.3. *For the function h, $\mathrm{CC}_t^{\max}(h) \geq \log(\frac{m}{t-1})$ and $\mathrm{CC}_t^{\mathrm{tot}}(h) \geq \log(m - t + 1)$.*

Proof: Consider the inputs a^t as before. Let us define the set $S(a, i)$ to consist of all increasing sequences $x_1, \ldots, x_i$ where $x_i \leq a$. We say that a^t is *done* at player P_i if the message P_i sends for a^i is different from any message P_i sends on vectors in $S(a, i)$: $M_i(a^i) \neq M_i(\mathbf{x})$ for $\mathbf{x} \in S(a, i)$.

In a protocol for h, every input a^t is done at player P_t, since for any $\mathbf{x} \in S(a, t)$, $h(\mathbf{x}) = 1$ whereas $h(a^t) = 0$. On the other hand, no input is done at player P_1 since $a^1 = a \in S(a, 1)$. However, it may happen that an input a^t is done at many players before P_t. So let us consider the first time when a particular input is done. By the pigeonhole principle, there must be some $i \geq 2$ so that $\frac{m}{t-1}$ of the inputs a^t are done for the first time at Player P_i, fix this i. We will show that player P_{i-1} must send many distinct messages to player P_i.

Pick two inputs a^t and b^t which are both done for the first time at player P_i. We claim that $M_{i-1}(a^{i-1}) \neq M_{i-1}(b^{i-1})$. Assume that this is not the case. Since both a^t and b^t are done for the first time at Player P_i, they cannot be done at Player P_{i-1}. Thus, there are vectors $\mathbf{x} \in S(a, i-1)$ and $\mathbf{y} \in S(b, i-1)$ so that $M_{i-1}(\mathbf{x}) = M_{i-1}(a^{i-1})$ and $M_{i-1}(\mathbf{y}) = M_{i-1}(b^{i-1})$. Hence M_{i-1} is the same for $\mathbf{x}, \mathbf{y}, a^{i-1}$ and b^{i-1}. Assume $a < b$ and consider the sequence $\mathbf{z} = \mathbf{x} \circ b$ of length i obtained by concatenating $\mathbf{x}$ with b. Since $\mathbf{x}$ is an increasing sequence with $x_{i-1} \leq a < b$, it follows that $\mathbf{z}$ is an increasing sequence with $z_i \leq b$. Hence $\mathbf{z} \in S(b, i)$. However we have

$$M_i(b^i) \ = M_i(b, M_{i-1}(b^{i-1})) \ = M_i(b, M_{i-1}(\mathbf{x})) \ = M_i(\mathbf{x} \circ b).$$

This contradicts the assumption that b^t is done at player P_i. This shows that P_{i-1} has to send $\frac{m}{t-1}$ distinct messages, which implies that $\mathrm{CC}_t^{\max}(h) \geq \log(\frac{m}{t-1})$.

To get the bound for $\mathrm{CC}_t^{\mathrm{tot}}(h)$, consider the inputs $\mathbf{a}^t$ as before. We showed that if $m_i \geq 1$ of these inputs are done for the first time at player P_i where $2 \leq i \leq t$, then P_{i-1} must send $\log(m_i)$ distinct messages. Overall $\sum_i m_i = m$. Further, if we let S denote the set of indices where $m_i \geq 2$, then $\sum_{i \in S} m_i \geq m - (t-1)$. Thus we have

$$\mathrm{CC}_t^{\mathrm{tot}}(h) \geq \sum_{i \in S} \log(m_i) \ = \log(\prod_{i \in S} m_i) \ \geq \log(\sum_{i \in S} m_i) \ \geq \log(m - (t-1)).$$

where we use $\prod_{i \in S} m_i \geq \sum_{i \in S} m_i$ since $m_i \geq 2$.

■

This bound on $\mathrm{CC}_t^{\max}(h)$ is tight: player P_1 can send y_1 (defined in Claim 3.2) to P_2 who sends it to P_3 and so on, so $\mathrm{CC}_t^{\max}(h) \leq \lceil \log(\frac{m}{t-1}) \rceil$. The bound on $\mathrm{CC}_t^{\mathrm{tot}}(h)$ is also essentially tight, since one possible protocol is where player P_{t-1} sends x_{t-1} to player P_t, who compares it to x_t and accordingly outputs 0 or 1. This shows that $\mathrm{CC}_t^{\mathrm{tot}}(h) \leq \log m$.

Next we consider the communication problem taking the OR of ℓ disjoint instances of h.

Definition 2. *For* $\mathbf{x} \in [m]^{t\ell}$, *let* $f(\mathbf{x}) = \vee_{j=1}^{\ell} h(C_j(\mathbf{x}))$.

Thus $f(\mathbf{x})$ is 1 if some column is increasing, 0 if every column is non-increasing and undefined otherwise. We consider the t player communication complexity when player P_i is given $R_i(\mathbf{x})$ as input. We would hope that solving f essentially requires solving ℓ independent copies of h. However, it turns out that this is true for $\mathrm{CC}_t^{\mathrm{tot}}(f)$ but not for $\mathrm{CC}_t^{\max}(f)$.

We show that the total communication complexity does satisfy a direct sum property, even for the unrestricted blackboard model. We give a lower bound on $\mathrm{N}_t(f)$, which implies lower bounds on $\mathrm{D}_t(f)$ and $\mathrm{CC}_t^{\mathrm{tot}}(f)$.

Lemma 3.4. *For the function* f, $\mathrm{N}_t(f) \geq \ell \log(\frac{m}{t-1})$.

Proof: Consider any ℓ-tuple $\mathbf{a} = (a_1, \ldots, a_\ell)$ where $a_j \in [m]$. We view $\mathbf{a}^t \in [m]^{t\ell}$ as a matrix with the row $\mathbf{a}$ repeated t times. Note that there are m^ℓ choices for $\mathbf{a}$, and that $f(\mathbf{a}^t) = 0$. We claim that at most $(t-1)^\ell$ such input matrices can share the same combinatorial rectangle in a protocol. Assume for contradiction that there are more of them. Each such input is specified by a distinct ℓ-tuple $\mathbf{a}$. Since there are at least $(t-1)^\ell + 1$ distinct tuples, there must be some index j where these tuples take t distinct values. Denote these tuples by $\mathbf{a1}, \ldots, \mathbf{at}$ and assume that $a1_j < \ldots < at_j$. The inputs $\mathbf{a1}^t, \ldots, \mathbf{at}^t$ lie in the same combinatorial rectangle. Hence the input $\mathbf{x}$ where $R_i(\mathbf{x}) = R_i(\mathbf{ai}^t) = \mathbf{ai}$ also lies in this rectangle. But then $C_j(\mathbf{x}) = (a1_j, \ldots, at_j)$ is an increasing sequence, hence $f(\mathbf{x}) = 1$.

This shows that no more than $(t-1)^\ell$ of the inputs $\mathbf{a}^t$ can lie in the same combinatorial rectangle and hence there are at least $(\frac{m}{t-1})^\ell$ rectangles in total. This shows that $\mathrm{N}_t(f) \geq \ell \log(\frac{m}{t-1})$. ■

Note that this gives a lower bound of $\mathrm{CC}_t^{\mathrm{tot}}(f) \geq \ell \log(\frac{m}{t-1})$ and hence $\mathrm{CC}_t^{\max}(f) \geq \frac{\ell}{t-1} \log(\frac{m}{t-1})$.

While this lower bound on $\mathrm{CC}_t^{\max}(f)$ directly implies lower bounds on streaming algorithms, the bounds are not strong enough for the range of parameters we are interested

in, where we would like a bound of the form $\ell \log(\frac{m}{t-1})$. Proving a direct sum result on maximum communication complexity would give strong enough bounds for our purposes. Unfortunately the direct sum property proved above does not hold for the maximum communication complexity, and the above lower bound for $\mathrm{CC}_t^{\max}(f)$ is nearly tight.

Lemma 3.5. *For the function f, $\mathrm{CC}_t^{\max}(f) \leq \lceil \frac{\ell}{t-1} \rceil \log(m)$.*

Proof: Divide the columns into $t-1$ groups of $b = \lceil \frac{\ell}{t-1} \rceil$ columns each. Let $G_i = \{(i-1)b+1, \ldots, ib\}$. We give a protocol where P_{i+1} will compute h for all columns in group G_i.

For each $j \in G_i$, player P_i sends x_{ij} to P_{i+1}. If $h(C_j(\mathbf{x})) = 1$, then $x_{i+1,j} > x_{i,j}$, else $x_{i+1,j} \leq x_{i,j}$. Thus P_{i+1} can compute h for each of these columns. If $h(C_j(\mathbf{x})) = 1$ for some column, he sends a special message to P_{i+2}. Else, he sends $x_{i+1,j}$ for $j \in G_{i+1}$ to P_{i+2}. The maximum message size is $b \log(m) = \lceil \frac{\ell}{t-1} \rceil \log(m)$. ∎

The above lemma shows that for the case $\ell = O(t)$ (the important setting of the parameters for our purposes), the maximum communication complexity of solving the OR of ℓ instances of the problem is within constant factor of the maximum communication complexity of solving just one instance, which is $\Omega(\log m)$ by Lemma 3.3.

4 Lower Bounds for the General Problem

We are able to prove the desired direct sum type property for maximum communication complexity of the problem defined using a weaker primitive problem.

Definition 3. *Given $\mathbf{x} \in [m]^t$, for $k \leq t$, let*

$$h_k(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathrm{lis}(\mathbf{x}) = 1 \\ 1 & \text{if } \mathrm{lis}(\mathbf{x}) \geq k. \end{cases}$$

For $\mathbf{x} \in [m]^{t\ell}$ and $k \leq t$, let $f_k(\mathbf{x}) = \vee_{j=1}^{\ell} h_k(C_j(\mathbf{x}))$.

We consider the t-player communication problem where player P_i is given the ith row $R_i(\mathbf{x})$ as input, and the players want to compute $f_k(\mathbf{x})$. The problem considered in the previous section is a special case of this problem taking $k = t$.

4.1 One-Round Protocols

We first analyze single round protocols, which give lower bounds for single pass algorithms. The following theorem and its proof are just a special case of our arguments for multi-round protocols. We present it separately first, because this case is somewhat simpler.

Theorem 4.1. *For the function f_k,*

$$\mathrm{CC}_t^{\max}(f_k) \geq \ell \left((1 - \frac{k}{t}) \log(\frac{m}{k-1}) - H(\frac{k}{t}) \right) - \log(t).$$

Proof: As in Lemma 3.4, we will consider inputs of the form $\mathbf{a}^t$ where $C_j(\mathbf{a}^t) = a_j^t$, thus the same row $\mathbf{a} = (a_1, \ldots, a_\ell)$ is repeated t times. For each such input, we define the set $S_i(\mathbf{a})$ of vectors that are confused with $\mathbf{a^i}$ by P_i. Formally, let $S_i(\mathbf{a}) = \{\mathbf{x} \in [m]^{i\ell} | M_i(\mathbf{x}) = M_i(\mathbf{a}^i)\}$. For each $\mathbf{x} \in S_i(\mathbf{a})$, we consider the length of the longest increasing subsequence in $C_j(\mathbf{x})$ ending with a value at most a_j and take the maximum over $\mathbf{x} \in S_i(\mathbf{a})$. Formally, let $q_{i,j}(\mathbf{a}) = \max_{\mathbf{x} \in S_i(\mathbf{a})} \mathrm{lis}_{a_j}(C_j(\mathbf{x}))$.

Claim 4.2. *For every $j \in [\ell]$,*

$$1 = q_{1,j}(\mathbf{a}) \leq q_{2,j}(\mathbf{a}) \leq \ldots \leq q_{t,j}(\mathbf{a}) \leq k - 1 \tag{1}$$

Proof: Since $\mathbf{a} \in S_1(\mathbf{a})$, it is clear that $q_{1,j}(\mathbf{a}) = 1$. Now consider an $\mathbf{x} \in S_i(\mathbf{a})$ such that $C_j(\mathbf{x})$ contains an increasing subsequence of length $q_{i,j}(\mathbf{a})$ ending with a value at most a_j. Consider the vector $\mathbf{y} = \mathbf{x} \circ \mathbf{a} \in m^{(i+1)\ell}$. It is easy to see that $\mathbf{y} \in S_{i+1}(\mathbf{a})$, and since $C_j(\mathbf{y}) = C_j(\mathbf{x}) \circ a_j$, we get $q_{i+1,j}(\mathbf{a}) \geq q_{i,j}(\mathbf{a})$. Finally, in a protocol for f_k, $q_{t,j}(\mathbf{a}) < k$, else some vector $\mathbf{x}$ such that $f_k(\mathbf{x}) = 1$ is not distinguished from $\mathbf{a}^t$ while $f_k(\mathbf{a}^t) = 0$. ∎

Note that at most $k - 2$ of the inequalities $q_{i-1,j}(\mathbf{a}) \leq q_{i,j}(\mathbf{a})$ can be strict. We say that player P_i is bad for $\mathbf{a}$ on column j if $q_{i-1,j}(\mathbf{a}) < q_{i,j}(\mathbf{a})$, else we say P_i is good on j. For a vector $\mathbf{a}$, on each column, there are at most $k - 2$ bad players. By averaging, for each $\mathbf{a}$, some player is good for $\mathbf{a}$ on at least $(1 - \frac{k}{t})\ell$ columns. There are at most $t - 1$ choices for the player, and at most $2^{H(\frac{k}{t})\ell}$ possibilities for choosing a set of $(1 - \frac{k}{t})\ell$ columns. There are at most $(k-1)^{\ell(1-\frac{k}{t})}$ possibilities for the $|S|$-tuples $q_{i,j}(\mathbf{a})$ for $j \in S$ where S is of size $\ell(1 - \frac{k}{t})$. Hence, there is a player P_i, a set S of size $\ell(1 - \frac{k}{t})$, and a set of vectors $T' \subseteq [m]^\ell$ with

$$|T'| \geq \frac{m^\ell}{(t-1)2^{H(\frac{k}{t})\ell}(k-1)^{\ell(1-\frac{k}{t})}}$$

such that player P_i is good for all vectors in T' on all columns in S, and for any $j \in S$ and $\mathbf{a}, \mathbf{b} \in T'$ we have $q_{i,j}(\mathbf{a}) = q_{i,j}(\mathbf{b})$.

Since at most $m^{\frac{k}{t}\ell}$ vectors can agree at the coordinates in S, we can pick a subset T of T' of size $\frac{|T'|}{m^{\frac{k}{t}\ell}}$ such that any two vectors in T differ on some coordinate in S. We now come to the crucial claim.

Claim 4.3. *For $\mathbf{a} \neq \mathbf{b} \in T$, $M_{i-1}(\mathbf{a}^{i-1}) \neq M_{i-1}(\mathbf{b}^{i-1})$.*

Proof: Assume for contradiction that $M_{i-1}(\mathbf{a}^{i-1}) = M_{i-1}(\mathbf{b}^{i-1})$. The vectors $\mathbf{a}$ and $\mathbf{b}$ differ at some coordinate $j \in S$, assume that $a_j < b_j$. Fix $\mathbf{x} \in S_{i-1}(\mathbf{a})$ such that $\mathrm{lis}_{a_j}(C_j(\mathbf{x})) = q_{i-1,j}(\mathbf{a})$. Let $\mathbf{y} = \mathbf{x} \circ \mathbf{b} \in [m]^{i\ell}$. Then

$$M_i(\mathbf{b}^i) \ = M_i(\mathbf{b}, M_{i-1}(\mathbf{b}^{i-1})) \ = M_i(\mathbf{b}, M_{i-1}(\mathbf{a}^{i-1})) \ = M_i(\mathbf{b}, M_{i-1}(\mathbf{x})) \ = M_i(\mathbf{y}).$$

Thus $\mathbf{y} \in S_i(\mathbf{b})$. But $C_j(\mathbf{y}) = C_j(\mathbf{x}) \circ b_j$, hence $\mathrm{lis}_{b_j}(C_j(\mathbf{y})) = q_{i-1,j}(\mathbf{a}) + 1$ since we can take the increasing subsequence of length $q_{i-1,j}(\mathbf{a})$ in $C_j(\mathbf{x})$ ending with a value at most a_j, and append b_j to it. This contradicts $q_{i,j}(\mathbf{b}) = q_{i-1,j}(\mathbf{b}) = q_{i-1,j}(\mathbf{a})$. ∎

The theorem follows from this claim, since

$$|T| \geq \frac{m^{\ell(1-\frac{k}{t})}}{(t-1)2^{H(\frac{k}{t})\ell}(k-1)^{\ell(1-\frac{k}{t})}}.$$

■

This lower bound is dominated by the term $\ell(1-\frac{k}{t})\log(\frac{m}{k-1})$. Lemma 5.2 will show that this is essentially tight.

4.2 Multi-Round Protocols

We now consider protocols that involve R rounds, which will give lower bounds for multi-pass algorithms. Recall that we use $M_i^r(\mathbf{x})$ to denote the message sent by P_i on input $\mathbf{x}$ in round r, and that $R_i(\mathbf{x})$ denotes the i-th row of $\mathbf{x}$, that is the input of P_i. Note that

$$M_1^r(\mathbf{x}) \ = M_1^r(R_1(\mathbf{x}), M_t^{r-1}(\mathbf{x}), \ldots, M_t^1(\mathbf{x})),$$

and for $i \geq 2$

$$M_i^r(\mathbf{x}) \ = M_i^r(R_i(\mathbf{x}), M_{i-1}^r(\mathbf{x}), \ldots, M_{i-1}^1(\mathbf{x})).$$

We define the set $S_i(\mathbf{a})$ as

$$S_i(\mathbf{a}) = \{\mathbf{x} \in [m]^{i\ell} |\ \forall r, M_i^r(\mathbf{x} \circ \mathbf{a}^{t-i}) = M_i^r(\mathbf{a}^t)\}.$$

Note that the definition of $S_i(\mathbf{a})$ in the one round case is just a special case of this definition. The following properties of these sets will be useful.

Claim 4.4. *If $\mathbf{x} \in S_i(\mathbf{a})$ then $\mathbf{x} \circ \mathbf{a}^{j-i} \in S_j(\mathbf{a})$ for all $j > i$.*

Proof: Assume that $\mathbf{x} \in S_i(\mathbf{a})$. By definition, for every round r, $M_i^r(\mathbf{x} \circ \mathbf{a}^{t-i}) = M_i^r(\mathbf{a}^t)$. But then

$$\begin{aligned} M_{i+1}^r(\mathbf{x} \circ \mathbf{a}^{t-i}) &= M_{i+1}^r(\mathbf{a}, M_i^r(\mathbf{x} \circ \mathbf{a}^{t-i}), \ldots, M_i^1(\mathbf{x} \circ \mathbf{a}^{t-i})) \\ &= M_{i+1}^r(\mathbf{a}, M_i^r(\mathbf{a^t}), \ldots, M_i^1(\mathbf{a^t})) \\ &= M_{i+1}^r(\mathbf{a}^t). \end{aligned}$$

This shows that $\mathbf{x} \in S_i(\mathbf{a}) \Rightarrow \mathbf{x} \circ \mathbf{a} \in S_{i+1}(\mathbf{a})$. The claim follows by repeatedly applying this last observation. ■

Claim 4.5. *Assume that for every $r \leq R$ $M_{i-1}^r(\mathbf{a}^t) = M_{i-1}^r(\mathbf{b}^t)$ and for every $r < R$ $M_t^r(\mathbf{a}^t) = M_t^r(\mathbf{b}^t)$. Then for any $\mathbf{x} \in S_{i-1}(\mathbf{a})$ we also have $\mathbf{x} \in S_{i-1}(\mathbf{b})$.*

Proof: We show that under the above assumptions, for every round $r \leq R$

$$M_{i-1}^r(\mathbf{x} \circ \mathbf{a}^{t-i+1}) = M_{i-1}^r(\mathbf{x} \circ \mathbf{b}^{t-i+1}),$$

and

$$M_t^r(\mathbf{x} \circ \mathbf{a}^{t-i+1}) = M_t^r(\mathbf{x} \circ \mathbf{b}^{t-i+1}).$$

Then, to conclude the proof of the claim note that $\mathbf{x} \in S_{i-1}(\mathbf{b})$ follows from the first statement, since the assumptions of the claim give $M_{i-1}^r(\mathbf{x} \circ \mathbf{a}^{t-i+1}) = M_{i-1}^r(\mathbf{a}^t) = M_{i-1}^r(\mathbf{b}^t)$ for every $r \leq R$.

We prove these two statements by induction on r.

In the first round of any protocol, the players $P_1, \ldots, P_{i-1}$ have no information about the inputs of the players P_j for $j \geq i$. Thus, $M_{i-1}^1(\mathbf{x} \circ \mathbf{a}^{t-i+1}) = M_{i-1}^1(\mathbf{x} \circ \mathbf{b}^{t-i+1})$ must hold. This in turn, together with the assumptions of the claim, implies that $M_{i-1}^1(\mathbf{x} \circ \mathbf{b}^{t-i+1}) = M_{i-1}^1(\mathbf{b}^t)$. Thus, in the first round the players P_j for $j \geq i$ cannot distinguish the vectors $\mathbf{x} \circ \mathbf{b}^{t-i+1}$ and $\mathbf{b}^t$. Note also that by Claim 4.4 the players P_j for $j \geq i$ cannot distinguish the vectors $\mathbf{x} \circ \mathbf{a}^{t-i+1}$ and $\mathbf{a}^t$ in any round. Since $M_t^1(\mathbf{a}^t) = M_t^1(\mathbf{b}^t)$ by assumption, we get $M_t^1(\mathbf{x} \circ \mathbf{a}^{t-i+1}) = M_t^1(\mathbf{x} \circ \mathbf{b}^{t-i+1})$.

To finish the proof it is enough to note that if $M_t^s(\mathbf{x} \circ \mathbf{a}^{t-i+1}) = M_t^s(\mathbf{x} \circ \mathbf{b}^{t-i+1})$ in every round $s \leq r-1$, then in round r the players $P_1, \ldots, P_{i-1}$ cannot distinguish the vectors $\mathbf{x} \circ \mathbf{a}^{t-i+1}$ and $\mathbf{x} \circ \mathbf{b}^{t-i+1}$. The rest of the above argument now can be repeated for round r. ∎

We prove the following theorem.

Theorem 4.6. *For the function f_k,*

$$\mathrm{CC}_{t,R}^{\max}(f_k) \geq \frac{\ell}{2R-1}\left((1-\frac{k}{t})\log(\frac{m}{k-1}) - H(\frac{k}{t})\right) - \frac{\log(t)}{2R-1}.$$

Proof: As before, we consider the quantity

$$q_{i,j}(\mathbf{a}) = \max_{\mathbf{x} \in S_i(\mathbf{a})} \mathrm{lis}_{a_j}(C_j(\mathbf{x})).$$

Since $\mathbf{a} \in S_1(\mathbf{a})$, we have $q_{1,j}(\mathbf{a}) = 1$. The bound $q_{t,j}(\mathbf{a}) \leq k-1$ is implied by the correctness of the protocol. From Claim 4.4, it follows that $q_{i,j}(\mathbf{a}) \leq q_{i+1,j}(\mathbf{a})$. Hence

$$1 = q_{1,j}(\mathbf{a}) \leq q_{2,j}(\mathbf{a}) \leq \ldots \leq q_{t,j}(\mathbf{a}) \leq k-1.$$

Now by the same argument as in the proof of Theorem 4.1, we get a set of vectors $T \subseteq [m]^\ell$, a set $S \subseteq [\ell]$ of columns and a player P_i such that:

- $q_{i,j}(\mathbf{a}) = q_{i,j}(\mathbf{b})$ for $\mathbf{a}, \mathbf{b} \in T$ and $j \in S$.
- $q_{i,j}(\mathbf{a}) = q_{i-1,j}(\mathbf{a})$ for all $\mathbf{a} \in T$ and $j \in S$.
- Any two vectors in T differ at some coordinate in S.

We now come to the crucial claim.

Claim 4.7. *For $\mathbf{a} \neq \mathbf{b} \in T$, either P_{i-1} sends different messages on $\mathbf{a}^t$ and $\mathbf{b}^t$ in some round $r \leq R$, or P_t sends different messages on $\mathbf{a}^t$ and $\mathbf{b}^t$ in some round $r < R$.*

Proof: Assume for contradiction that for every round $r \leq R$, we have $M^r_{i-1}(\mathbf{a}^t) = M^r_{i-1}(\mathbf{b}^t)$ and for every $r < R$ $M^r_t(\mathbf{a}^t) = M^r_t(\mathbf{b}^t)$. As before assume that $\mathbf{a}$ and $\mathbf{b}$ differ on column j and that $a_j < b_j$.

Fix $\mathbf{x} \in S_{i-1}(\mathbf{a})$ such that $\mathrm{lis}_{a_j}(C_j(\mathbf{x})) = q_{i-1,j}(\mathbf{a})$. By Claim 4.5 $\mathbf{x} \in S_{i-1}(\mathbf{b})$. Then, Claim 4.4 implies $\mathbf{x} \circ \mathbf{b} \in S_i(\mathbf{b})$. Since $a_j < b_j$, the longest increasing sequence in $C_j(\mathbf{x} \circ \mathbf{b})$ has length $q_{i-1,j}(\mathbf{a}) + 1$, this gives a contradiction. This proves Claim 4.7. ∎

To finish the proof of the theorem, consider the sequence of $2R-1$ messages $M^r_{i-1}(\mathbf{a})$ for $r \leq R$ and $M^r_t(\mathbf{a})$ for $r \leq R-1$. By Claim 4.7, this sequence of messages is different for $\mathbf{a} \neq \mathbf{b} \in T$. Thus if the maximum message size is μ, then $2^{(2R-1)\mu} \geq |T|$ and the statement of the theorem follows. ∎

4.3 The Reduction to LIS

We apply the reduction from computing f_k to the LIS problem from GJKK, and show that Theorem 4.6 yields a tight lower bound on the space complexity of deterministic streaming algorithms for approximating the length of the LIS that make a constant number of passes over the data. For completeness, we present the proof from GJKK of the following lemma in the Appendix.

Lemma 4.8. *(Lemma 4.4, [GJKK07])* $\mathrm{CC}^{\max}_{t,R}(f_k)$ *is a lower bound on the space required for any R-pass deterministic streaming algorithm that computes a $1 + \frac{k-1}{\ell}$ approximation to the LIS on sequences of length $t \cdot \ell$ over an alphabet of size $m \cdot \ell$.*

We now conclude the proof of our main result.

Proof of Theorem 1.1 We set $k - 1 = t/2$. Now let $n = t\ell$, $\varepsilon = \frac{k-1}{\ell}$ and $M = mt$. This gives $t = \sqrt{2\varepsilon n}$, $\ell = \sqrt{\frac{n}{2\varepsilon}}$. Now plugging this into the lower bound of Theorem 4.6, and ignoring the lower order terms, we get

$$S \geq \frac{\ell}{2R-1}(1 - \frac{k}{t}) \log(\frac{m}{k-1}) = \frac{1}{2R-1}\sqrt{\frac{n}{8\varepsilon}} \log(\frac{M}{\varepsilon n}).$$ ∎

When $R = 1$, this matches the $O(\sqrt{\frac{n}{\varepsilon}} \log M)$ upper bound of the GJKK algorithm up to constant factors provided we take M to be $(\varepsilon n)^{1+\gamma}$ for some $\gamma > 0$. In particular, it gives a tight lower bound even when ε is $o(1)$. If we take $\varepsilon = \frac{1}{n}$, then this corresponds to exact computation of the length of the LIS, for which we get a tight linear lower bound. In addition, it shows that as long as the number of passes is constant, the space required is $\Omega(\sqrt{n})$.

4.4 (No) Direct Sum in the One-Way Blackboard Model

It is interesting to note that the lower bounds of Theorems 4.6 and 5.1 hold even for a version of the function f_k with a stronger promise, where in the NO case, the input is guaranteed to be of the form $\mathbf{a}^t$ for $\mathbf{a} \in [m]^\ell$. We show that in the one-way blackboard model the direct sum property for max communication does not hold for this version of f_k. To the best of our knowledge, this is the first separation between the private messages and blackboard models of one-way communication.

Definition 4. *For $\mathbf{x} \in [m]^t$ and $k \leq t$, define the function $\hat{h}_k$ as*

$$\hat{h}_k(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathrm{lis}(\mathbf{x}) \geq k \\ 0 & \text{if } \mathbf{x} = a^t, \quad a \in [m] \end{cases}$$

For $\mathbf{x} \in [m]^{t\ell}$, define the function $\hat{f}_k$ as $\hat{f}_k(\mathbf{x}) = \vee_{j=1}^{\ell} \hat{h}_k(C_j(\mathbf{x}))$.

Lemma 4.9. *For the function $\hat{f}_k$, $\mathrm{BB}_t^{\max}(\hat{f}_k) \leq \lceil \frac{\ell}{k-1} \rceil \log m$ and $\mathrm{BB}_t^{\mathrm{tot}}(\hat{f}_k) \leq \ell \log m$.*

Proof: We divide the columns into $k-1$ groups $G_1, \ldots, G_{k-1}$ of size $\lceil \frac{\ell}{k-1} \rceil$. Player P_i writes his inputs for the columns in G_i on the blackboard. The other players check if their inputs agree with his on G_i, else they know that it is a YES instance. Suppose for all i, the players $P_k, \ldots, P_t$ all have inputs that agree with P_i on the columns in G_i. In this case, every column contains the same letter repeated at least $t-k+2$ times. Hence, none of these columns have an increasing subsequence of length more than $k-1$, so is a NO instance. The max communication of this protocol is $\lceil \frac{\ell}{k-1} \rceil \log m$, the total communication is $\ell \log m$. ∎

By the same argument as Lemma 3.4, one can show that $\mathrm{N}_t(\hat{f}_k) \geq \ell \log(\frac{m}{k-1})$, so both bounds are nearly tight.

In the setting where k is a constant fraction of ℓ, $m = \ell^{O(1)}$ which we are interested in, $\mathrm{BB}_t^{\max}(\hat{f}_k) = O(\log m)$. In the case when $\ell = 1$, trivially $\mathrm{BB}_t^{\max}(\hat{f}_k) \geq 1$. Thus the max communication does not increase by a factor of ℓ as in a direct sum result.

5 Tight Bounds on One-Round Communication in the Private Messages Model

As noted in the previous section, the total communication for the version of f_k that we analyze does not satisfy a strong enough direct sum property in the blackboard model. In this section, we give tight bounds on the total and maximum one-round, communication complexity of this problem in the private messages model.

We give a lower bound on the one-round total communication complexity of the problem $\hat{f}_k$, which also implies a lower bound for $\mathrm{CC}_t^{\mathrm{tot}}(f_k)$.

Theorem 5.1. *For the function $\hat{f}_k$,*

$$\mathrm{CC}_t^{\mathrm{tot}}(\hat{f}_k) \geq (t-3k)\frac{\ell}{2}\left(\log(\frac{m}{k-1}) - 2\right).$$

Proof: For $\mathbf{a} \in [m]^\ell$ define the numbers $q_{i,j}(\mathbf{a})$ as before and say that player P_i is bad for $\mathbf{a}$ on column j if $q_{i-1,j}(\mathbf{a}) < q_{i,j}(\mathbf{a})$, else we say P_i is good on column j. For a vector $\mathbf{a}$, on each column $j \in [\ell]$, there are at least $t-k+1$ good players. Suppose that player P_i is good for $\mathbf{a}$ on $\ell_i(\mathbf{a})$ columns. Then $\sum_i \ell_i(\mathbf{a}) \geq (t-k+1)\ell$. By averaging, for at least $t-2k$ players, we must have $\ell_i(\mathbf{a}) \geq \ell/2$, else

$$\sum_i \ell_i(\mathbf{a}) \;<\; (t-2k)\ell + 2k\frac{\ell}{2} \;\leq\; \ell(t-k).$$

Call these the good players for $\mathbf{a}$. Suppose player P_i is good for a set T_i'' of vectors in $[m]^\ell$ and let $|T_i''| = g_i$. Then we have $\sum_i g_i \geq (t-2k)m^\ell$. Again, by averaging, there are at least $t-3k$ players where $g_i \geq \frac{m^\ell}{3}$, else

$$\sum_i g_i < (t-3k)m^\ell + 3k\frac{m^\ell}{3} = (t-2k)m^\ell.$$

Now fix a player P_i where $g_i \geq \frac{m^\ell}{3}$ and note that for each $\mathbf{a} \in T_i''$, P_i is good on at least $\ell/2$ columns.

There are at most 2^ℓ choices for choosing a set S of $\ell/2$ columns, and $(k-1)^{\ell/2}$ choices for the $\ell/2$-tuples $q_{i,j}(\mathbf{a})$ for $j \in S$. Thus, there is a set S of $\ell/2$ columns, and $T_i' \subset T_i''$ of size at least $\frac{g_i}{2^\ell(k-1)^{\ell/2}}$ such that P_i is good for each $\mathbf{a} \in T_i'$ on every column $j \in S$ and $q_{i,j}(\mathbf{a}) = q_{i,j}(\mathbf{b})$ for $j \in S$ and $\mathbf{a}, \mathbf{b} \in T_i'$. Further we can find $T_i \subset T_i'$ of size at least $\frac{|T_i'|}{m^{\ell/2}}$ so that any two vectors in T_i' differ on some coordinate in S. As in Claim 4.3, we can argue that P_i has to send different messages on any two vectors in T_i. Hence P_i sends messages of size at least $\log(|T_i|) \geq \frac{\ell}{2}\log(\frac{m}{k-1}) - \ell - \log 3$. Thus the total communication is at least

$$\sum_i \log(|T_i|) \geq (t-3k)\left(\frac{\ell}{2}\log(\frac{m}{k-1}) - \ell - \log 3\right).$$

■

We note that this is dominated by $\frac{\ell(t-3k)}{2}\log(\frac{m}{k-1})$. If we take $k = \frac{t}{4}$, this gives $\mathrm{CC}_t^{\mathrm{tot}}(\hat{f}_k) = \Omega(t\ell \log \frac{m}{k-1})$. In contrast, Lemma 4.9 shows that $\mathrm{BB}_t^{\mathrm{tot}}(\hat{f}_k) \leq \ell \log m$. Similarly, by Theorem 4.1 and Lemma 4.9 the max communication can also differ by a factor of $k = \frac{t}{4}$. This shows a separation between the two models of one-way communication. One can improve on the $t-3k$ term in the lower bound, but we omit the details.

Finally we show that our lower bounds on the maximum and total communication complexity of f_k are essentially tight. We will use the simple fact that in a sequence $\mathbf{x} = (x_1, \ldots, x_n)$ of n numbers, there exists $i \in [n]$ such that $x_{i+1} > x_i$ if and only if $\mathrm{lis}(\mathbf{x}) \geq 2$.

Lemma 5.2. *The following upper bounds hold for f_k:*

$$\mathrm{CC}_t^{\max}(f_k) \leq \ell\log(\tfrac{m}{k-1}), \quad \mathrm{CC}_t^{\mathrm{tot}}(f_k) \leq t\ell\log(\tfrac{m}{k-1}),$$
$$\mathrm{CC}_t^{\max}(f_k) \leq \lceil \ell(1-\tfrac{k-2}{t})\rceil \log(m), \quad \mathrm{CC}_t^{\mathrm{tot}}(f_k) \leq t\lceil \ell(1-\tfrac{k-2}{t})\rceil \log(m).$$

Proof: Let us define

$$y_{i,j} = \lceil \frac{x_{i,j}}{k-1} \rceil \in [1, \lceil \frac{m}{k-1} \rceil].$$

The players will try to tell if $C_j(\mathbf{y})$ is non-increasing for every j or whether some column does contain a non-trivial increasing subsequence. For every column j, P_i sends $y_{i,j}$ to P_{i+1}. If $y_{i+1,j} > y_{i,j}$, then P_{i+1} concludes that $f_k(\mathbf{x}) = 1$ and ends the protocol, else it sends $y_{i+1,j}$ for all j to the next player. The maximum message size is bounded by $\ell \log(\frac{m}{k-1})$.

Assume that $h_k(C_j(\mathbf{x})) = 1$. Let the first and last elements of the increasing subsequence in $C_j(\mathbf{x})$ be $x_{r,j}$ and $x_{s,j}$ respectively. Then $x_{s,j} \geq x_{r,j} + k - 1$, hence $y_{s,j} \geq y_{r,j} + 1$, so $y_{1,j}, \ldots, y_{t,j}$ contains an increasing subsequence of length at least 2. So $y_{i+1,j} > y_{i,j}$ for some i. On the other hand, if $h_k(C_j(\mathbf{x})) = 0$, then $x_{s,j} \leq x_{r,j}$, and so $y_{s,j} \leq y_{r,j}$ for all $r < s$. Hence the sequence $y_{i,j}$ is non-increasing.

For the second upper bound, consider the protocol where we divide the ℓ columns into $\lceil \frac{t}{t-k+2} \rceil$ groups of size at most $\lceil \ell(1 - \frac{k-2}{t}) \rceil$ each. The first $t - k + 2$ players will compute $h_k(C_j(\mathbf{x}))$ for columns C_j where $j \in [1, \lceil \ell(1 - \frac{k-2}{t}) \rceil]$. Each player P_i for $i \in [1, \ldots, t-k+2]$ sends $x_{i,j}$ where $j \in [1, \lceil \ell(1 - \frac{k-2}{t}) \rceil]$ to P_{i+1}. If P_{i+1} finds $x_{i,j} < x_{i+1,j}$ for some j, then $h_k(C_j(\mathbf{x})) = 1$ and the protocol terminates. Clearly, the maximum message size is bounded by $\lceil \ell(1 - \frac{k-2}{t}) \rceil \log(m)$.

Observe that if $h_k(C_j(\mathbf{x})) = 1$, then any subsequence of $t-k+2$ numbers from C_j cannot be non-increasing, as it will contain two numbers from the increasing subsequence, which are in increasing order. So it must contain two consecutive numbers in increasing order. This will be detected by our protocol. On the other hand, when $f_k(\mathbf{x}) = 0$, each column is non-increasing, so every pair of consecutive numbers is also non-increasing. ∎

6 Open Problems

We have proved $\Omega(\sqrt{n})$ lower bounds on the space requirement of deterministic data-stream algorithms with constant number of passes that give constant factor approximation of the length of the longest increasing subsequence in a sequence of length n. It is interesting to ask if such a lower bound holds even for randomized algorithms. However, the direct sum based approach employed here for the problem f_k will not yield a randomized lower bound. Amit Chakrabarti has pointed out that the randomized communication complexity of the problem f_k is bounded by $O(\frac{1}{\varepsilon} \log m)$, for the setting of parameters that yields the $\Omega(\sqrt{n})$ lower bound for deterministic algorithms in Theorem 1.1.

Let $\mathrm{R}_t^{\max}(f_k)$ denote the maximum communication complexity of f_k in the randomized one-way private messages model.

Theorem 6.1. *[Cha07] For the function f_k,*

$$\mathrm{R}_t^{\max}(f_k) \leq \frac{2\ell t}{(k-1)^2} \log m.$$

Note that if we set $k-1 = t/2$, $n = t\ell$, $\varepsilon = \frac{k-1}{\ell}$ as in the proof of Theorem 1.1, we get that $\mathrm{R}_t^{\max}(f_k) \leq \frac{4}{\varepsilon} \log m$. It is interesting to ask if indeed a better randomized upper bound is possible for the general problem of approximating the LIS, and the following question raised in GJKK remains open.

Problem 1. *Is there a randomized streaming algorithm to approximate* $\mathrm{lis}(\sigma)$ *within* $(1+\varepsilon)$ *for constant* $\varepsilon > 0$ *using space* $o(\sqrt{n})$*?*

Acknowledgments

The second author would like to thank T.S Jayram, Robert Krauthgamer and Ravi Kumar for many interesting discussions about this problem. We would also like to thank Amit Chakrabarti for enlightening discussions about direct sum problems, and allowing us to include the statement of Theorem 6.1.

References

[AD99] D. Aldous and P. Diaconis. Longest increasing subsequences: From patience sorting to the Baik–Deift–Johansson theorem. *Bulletin of the American Mathematical Society*, 36:413–432, 1999.

[AJKS02] M. Ajtai, T.S. Jayram, R. Kumar, and D. Sivakumar. Approximate counting of inversions in a data stream. In *Proc.* 34^{th} *Ann. ACM Symposium on Theory of Computing (STOC'02)*, pages 370–379, 2002.

[AMS96] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *Proc. 28th ACM Symp. on Theory of Computing*, pages 20–29, 1996.

[BBD+02] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *Proc.* 21^{st} *ACM Symposium on Principles of Databases Systems*, pages 1–16, 2002.

[BYJKS04] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004.

[Cha07] A. Chakrabarti. Personal communication, 2007.

[CKS03] A. Chakrabarti, S. Khot, and X. Sun. Near-optimal lower bounds on the multi-party communication complexity of set-disjointness. In *Proceedings of the 18th Annual IEEE Conference on Computational Complexity (CCC'03)*, pages 107–117, 2003.

[CMS01] G. Cormode, S. Muthukrishnan, and S. C. Sahinalp. Permutation editing and matching via embeddings. In *Proc.* 28^{th} *International Colloquium on Automata, Languages and Programming (ICALP'01)*, pages 481–492, 2001.

[CSWY01] A. Chakrabarti, Y. Shi, A. Wirth, and A. C. Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *Proc.* 42^{nd} *IEEE Symp. on Foundations of Computer Science (FOCS'01)*, pages 270–278, 2001.

[EJ08] F. Ergün and H. Jowhari. On distance to monotonicity and longest increasing subsequence of a data stream. In *Proc.* 19^{th} *ACM-SIAM Symposium on Discrete Algorithms (SODA'08)*, 2008.

[GJKK07] P. Gopalan, T.S. Jayram, R. Krauthgamer, and R. Kumar. Estimating the sortedness of a data stream. In *Proc.* 18^{th} *ACM-SIAM Symposium on Discrete Algorithms (SODA'07)*, 2007.

[Gus97] D. Gusfield. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, 1997.

[GZ03] A. Gupta and F. Zane. Counting inversions in lists. In *Proc.* 14^{th} *ACM-SIAM Symposium on Discrete Algorithms (SODA'03)*, pages 253–254, 2003.

[KN97] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[LNVZ06] D. Liben-Nowell, E. Vee, and A. Zhu. Finding longest increasing and common subsequences in streaming data. *Journal of Combinatorial Optimization*, 11(2):155–175, 2006.

[Mut05] S. Muthukrishnan. *Data Streams: Algorithms and Applications*. Now Publishers Inc., 2005.

[Pev03] P. Pevzner. *Computational Molecular Biology*. Elsevier Science Ltd., 2003.

[SS02] M. Saks and X. Sun. Space lower bounds for distance approximation in the data stream model. In *Proc.* 34^{th} *Ann. ACM symposium on Theory of computing (STOC'02)*, pages 360–369, 2002.

[SW07] X. Sun and D. P. Woodruff. The communication and streaming complexity of computing the longest common and increasing subsequences. In *Proc.* 18^{th} *ACM-SIAM Annual Symposium on Discrete Algorithms (SODA'07)*, 2007.

A Proof of Lemma 4.8

Proof: [GJKK07] The players reduce computing $f_k(\mathbf{x})$ to approximating the length of the LIS for a string σ of length $t \cdot \ell$ defined as $\sigma_{(i-1)\ell+j} = (j-1)m + x_{i,j}$. Note that the block of ℓ consecutive numbers $\sigma_{(i-1)\ell+1}, \ldots, \sigma_{i\ell}$ of σ can be computed by player P_i, and that these numbers are in increasing order. Thus if we view σ as a $t \times \ell$ matrix, $R_i(\sigma)$ is an increasing sequence. Further, $R_i(\sigma)$ depends only on $R_i(\mathbf{x})$. Now let us consider the columns. The column $C_j(\sigma)$ consists of the numbers $(j-1)m + x_{i,j}$ for $i = 1, \ldots, t$, thus it is essentially $C_j(\mathbf{x})$ with each number shifted by $(j-1)m$. Thus, all the numbers in $C_j(\sigma)$ lie in the range $[(j-1)m+1, jm]$. Further, $\mathrm{lis}(C_j(\sigma)) = \mathrm{lis}(C_j(\mathbf{x}))$.

We claim that if $f_k(\mathbf{x}) = 0$, then $\mathrm{lis}(\sigma) \leq \ell$. Since every column is non-increasing, an increasing subsequence in σ can contain at most one number per column, so $\mathrm{lis}(\sigma) \leq \ell$. In fact $\mathrm{lis}(\sigma) = \ell$, since every row is an increasing sequence.

On the other hand, if $f_k(\mathbf{x}) = 1$, then $\mathrm{lis}(\sigma) \geq \ell + k - 1$. Assume that column j has an increasing subsequence of length k. Then by taking $\sigma_1, \ldots, \sigma_{j-1}$ followed by the increasing subsequence in column j, followed by $\sigma_{(t-1)\ell+j+1}, \ldots, \sigma_{t\ell}$, we get an increasing subsequence of length $\ell + k - 1$.

Now assume that there is an R-pass streaming algorithm that can compute a $1+\varepsilon$ approximation to the length of the LIS using space S where $\varepsilon = \frac{k-1}{\ell}$. By a standard simulation, by running this algorithm on σ, we get an R-round, one-way communication protocol for f_k in the private messages model with $\mathrm{CC}_t^{\max}(f) \leq S$. ∎