A Note on Randomized Streaming Space Bounds for the Longest Increasing Subsequence Problem

Amit Chakrabarti

Abstract

The deterministic space complexity of approximating the length of the longest increasing subsequence of a stream of \( N \) integers is known to be \( \tilde{\Theta}(\sqrt{N}) \). However, the randomized complexity is wide open. We show that the technique used in earlier work to establish the \( \Omega(\sqrt{N}) \) deterministic lower bound fails strongly under randomization: specifically, we show that the communication problems on which the lower bound is based have very efficient randomized protocols. The purpose of this note is to guide and alert future researchers working on this very interesting problem.

1 Introduction

For a sequence \( \sigma \) of integers, let \( \text{lis}(\sigma) \) denote the length of the longest (strictly) increasing subsequence of \( \sigma \). In the \textsc{Approximate-LIS} problem (abbreviated \textsc{ALIS}\(_{N,\epsilon}^M\)), we are given streaming access to a sequence \( \sigma \) of length \( N \), with entries in \( [M] := \{1, 2, \ldots, M\} \), and must report a \((1 \pm \epsilon)\)-approximation to \( \text{lis}(\sigma) \). The goal, as is usual in data stream algorithms [Mut03], is to minimize the space (i.e., amount of working memory) and the per-item processing time used to do so. We are concerned here primarily with the space complexity of this problem; both deterministic and randomized algorithms are of interest.

The vast majority of sublinear space streaming algorithms use randomization as a key technique and, in fact, provably need to do so [AMS99]. The \textsc{ALIS} problem presents one of the very few instances where (1) a natural problem has a sublinear space deterministic streaming algorithm, (2) randomization is not known to provide any extra space savings, and (3) randomization could provide an exponential improvement, based on current knowledge. We believe that this makes the randomized space complexity of \textsc{ALIS} an extremely interesting theoretical question about data stream algorithms.

A sublinear upper bound for \textsc{ALIS} was given by Gopalan et al. [GJKK07], who showed the following.

**Theorem 1.1.** There is a deterministic \( O(\sqrt{N/\epsilon} \cdot \log M) \)-space streaming algorithm for \textsc{ALIS}\(_{N,\epsilon}^M\).

An essentially matching lower bound was then given by Gál and Gopalan [GG07] and also — independently and via a different argument — by Ergün and Jowhari [EJ08]. These lower bounds applied only to deterministic algorithms and used reductions from certain communication problems on which suitable “direct sum” theorems could be proven. In this note, we show that these techniques do not generalize to give randomized streaming lower bounds, because the underlying communication problems do have randomized protocols with cost exponentially lower than the best deterministic protocol.

2 Preliminaries

For a communication problem \( f \), let \( D^{\text{max}}(f) \) denote the maximum number of bits sent by any single player in a deterministic protocol that computes \( f \), minimized over all such protocols. Let \( R^{\text{max}}(f) \) denote the analogous quantity for constant-error randomized protocols.

*Dartmouth College. ac@cs.dartmouth.edu. Supported in part by NSF Grants CCF-0448277 and IIS-0916565.
In the HIDDEN-INCREASING-SEQUENCE problem (abbreviated HIS\textsuperscript{m}_{t,n,k}), the input is a \( t \times n \) matrix \( X = \{ x_{ij} \}_{i \in [t], j \in [n]} \) with entries in \([m]\), with the promise that each column \( X_j \) of \( X \) satisfies one of the following:

1. The column is non-increasing, i.e., \( \text{lis}(X_j) = 1 \).
2. The column has a “long” increasing subsequence, i.e., \( \text{lis}(X_j) \geq k \).

This input is divided amongst \( t \) players, named \( \text{PLR}_1, \ldots, \text{PLR}_t \), with player \( i \) receiving the \( i \)th row of \( X \). The goal is to compute the predicate \( \text{HIS}^m_{t,n,k}(X) := \bigvee_{i=1}^t (\text{lis}(X_j) \geq k) \), in the following model of communication: \( \text{PLR}_1 \) sends a (private) message to \( \text{PLR}_2 \), who then sends a message to \( \text{PLR}_3 \) and so on, until we reach \( \text{PLR}_t \), who then announces the output.

Let \( \sigma_X \) denote the length-\( (tn) \) sequence, with elements in \([tm]\), obtained by applying the mapping \( x_{ij} \mapsto (j-1)m + x_{ij} \) to the entries of \( X \) and reading off the result in row-major order. It is easy to see that

\[
\text{HIS}^m_{t,n,k}(X) = 0 \implies \text{lis}(\sigma_X) = n, \quad \text{and} \quad \text{HIS}^m_{t,n,k}(X) = 1 \implies \text{lis}(\sigma_X) \geq n + k - 1. \quad (1)
\]

This was first formally observed by Gopalan et al. \[GJKK07\], and it immediately implies the following.

**Lemma 2.1** (Lemma 4.4 of \[GJKK07\]). The deterministic and randomized streaming space complexities of ALIS\textsuperscript{M}_{N,\varepsilon}, with \( M = tm, N = tn \) and \( \varepsilon = (k-1)/n \), are at least \( \text{D}^{\max}(\text{HIS}^m_{t,n,k}) \) and \( \text{R}^{\max}(\text{HIS}^m_{t,n,k}) \), respectively.

Thus, a natural approach to establishing space lower bounds on ALIS is to prove communication complexity lower bounds for HIS. Using this approach, Gád and Gopalan proved a tight deterministic lower bound:

**Theorem 2.2** (Theorems 1.1 and 4.1 of \[GG07\]). Let \( H_b \) denote the binary entropy function. Then, we have

\[
\text{D}^{\max}(\text{HIS}^m_{t,n,k}) \geq n \left( 1 - \frac{k}{t} \right) \log \left( \frac{m}{k-1} \right) - H_b \left( \frac{k}{t} \right) - \log t.
\]

In particular, setting \( k-1 = t/2 = \varepsilon n \), we have \( \text{D}^{\max}(\text{HIS}^m_{t,n,k}) = \Omega(n \log(2m/\varepsilon n)) \). Combining this bound with Lemma 2.1 shows that the deterministic space complexity of ALIS\textsuperscript{M}_{N,\varepsilon} is \( \Omega(\sqrt{N/\varepsilon} \cdot \log(M/\varepsilon N)) \).

In fact, they also generalized this theorem to apply to multi-pass, but still deterministic, streaming algorithms by extending the communication lower bound to multi-round protocols. Our concern here is with a different potential generalization: can one generalize Theorem 2.2 to randomized protocols, and thus, randomized streaming algorithms? Our main result is a negative one, showing that this is not possible.

## 3 A Randomized Communication Upper Bound

**Theorem 3.1** (Main Theorem). We have

\[
\text{R}^{\max}(\text{HIS}^m_{t,n,k}) = O \left( \frac{nt \log m}{k^2} \right).
\]

In particular, for the setting \( k = \Theta(t) = \Theta(\varepsilon n) \), which was used for the lower bound in Theorem 2.2, we have \( \text{R}^{\max}(\text{HIS}^m_{t,n,k}) = O(\varepsilon^{-1} \log m) \).

**Proof.** Let \( r = 2(t-1)/(k-1) \). Consider the following protocol. Each player goes through a receive-and-compute phase (skipped by PLR\textsubscript{1}) followed by a transmit phase (skipped by PLR\textsubscript{t}). In the transmit phase, PLR\textsubscript{i} chooses a subset \( J_i \subseteq [n] \) of size \( |J_i| = \lceil 2n/(k-1) \rceil \) uniformly at random from amongst all such subsets. He then sends to PLR\textsubscript{i+1} the following data:

1. The set \( S_i = \{ x_{ij} : j \in J_i \} \); for each \( j \in J_i \), we say that PLR\textsubscript{i} samples column \( j \).
2. The sets \( S_h \) for \( \max \{ 1, i-r+1 \} \leq h < i \), which he obtains from PLR\textsubscript{i-1}. 

2
In the receive-and-compute phase (which precedes the transmit phase), \( PLR_i \) receives from \( PLR_{i-1} \) the sets \( S_h \), for all \( h \in H_i := \{ h : \max\{1, i - r\} \leq h < i \} \). He checks whether the following condition holds:

\[
\exists h \in H_i \exists j \in J_h : x_{hj} < x_{ij}.
\]

He can do so because \( x_{hj} \) is available to him from the message he receives and \( x_{ij} \) is within the part of the input he knows to begin with. If (2) holds, he terminates the protocol and outputs 1. Notice that he is correct to do so, because he has discovered that \( \text{lis}(X_j) > 1 \), which, by the promise, means that \( \text{lis}(X_j) \geq k \).

Otherwise, if (2) does not hold, he moves on to the transmit phase as described above.

If the protocol reaches \( PLR_i \), and no player (including himself) has output 1 in their receive-and-compute phase, then he outputs 0. This completes the description of the protocol. Clearly, one can tweak it so that the output is always announced by \( PLR_i \), as in our definition.

We now argue that this protocol is correct. Suppose that \( \text{HIS}^m_{r,n,k}(X) = 0 \). Then (2) never holds, and the protocol always reaches \( PLR_i \), who correctly outputs 0.

Suppose that \( \text{HIS}^m_{r,n,k}(X) = 1 \), and suppose that the \( j \)th column of \( X \) contains a “hidden increasing subsequence” \( \langle x_{ij} \rangle_{i \in I} \), where \( I \subseteq [r] \) and \( |I| = k \). Call the player \( PLR_i \) critical if the following condition holds:

\[
i \in I \land (\exists i' \in I : 0 < i' - i \leq r).
\]

Also, in this case, call \( PLR_i \) the follower of \( PLR_{i+1} \), where \( i' \) is the minimum integer such that \( 0 < i' - i \leq r \).

A straightforward estimation argument shows that there exist at least \( (k - 1)/2 \) critical players. Notice that if a critical player samples column \( j \), then his follower correctly announces the output to be 1. Thus, the probability that the protocol fails to output 1 is at most

\[
\Pr[\text{no critical player samples column } j] \leq \left(1 - \frac{2}{k - 1}\right)^{(k-1)/2} \leq e^{-1}.
\]

Finally, note that in order to achieve this constant error probability, each player had to send at most \( r \cdot \lceil 2n/(k-1) \rceil \) entries of \( X \), which required \( O((nr/k) \cdot \log m) = O((nt/k^2) \cdot \log m) \) bits.

\[\square\]

4 Concluding Remarks

We remark that a similar randomized communication upper bound can be shown to hold for the slightly different communication problem used by Ergün and Jowhari [EJ08] in their alternate proof of the deterministic lower bound for \( \text{ALIS} \).

These protocols raise an obvious question: does the idea extend to give a polylogarithmic-space streaming upper bound for \( \text{ALIS} \)? We leave this question open.

References


