CS 671: Graph Streaming Algorithms and Algorithms **Rutgers: Fall 2020**

Lecture 12

November 24, 2020

Instructor: Sepehr Assadi *Scribe: Vishvajeet Nagargoje*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

In this lecture, we will discuss lower bounds for streaming algorithms via the pointer chasing problem.

1 Introduction to Pointer Chasing

We will consider the pointer chasing problem on layered graphs which we define as follows,

Definition 1 ((m, k)-layered graphs)**.** *For any integers $m, k \geq 1$, we define a (m, k)-layered graph $G(V, E)$, as any graph with the following properties:*

- *Vertex-set V consists of $k + 1$ layers of vertices $V_1, \ldots, V_{k+1}$, each of size m.*
- *Edge-set E consists of edges that go only between consecutive layers. For $i \in [k]$, E_i connects vertices of V_i to those of V_{i+1}, such that each vertex in V_i is connected to exactly one vertex in V_{i+1}.*

  For any vertex $v \in V_1$, $P_1(v)$ denotes the unique vertex reachable from v in V_2. Note that $P_1(v) = E_1(v)$. Similarly, $P_2(v) = E_2(E_1(v))$ is the (unique) vertex reachable from v in V_2, and continuining in this fashion, $P_{k+1}(v) = E_k(E_{k-1} \ldots (E_1(v)))$ is the vertex reachable in V_{k+1}.

We define pointer chasing on layered graphs as follows :

Definition 2. *Pointer Chasing (**PC**) on layered graphs*

Let $m, k \in \mathbb{N}^+$. In $\boldsymbol{PC}_k$, we have a (m, k)-layered graph on vertex layers $V_1, \ldots, V_{k+1}$, the goal is to find out the unique vertex $P_{k+1}(v)$ reachable from v in V_{k+1}.

We will also need the following version of **PC**in the standard 2-player communication model to prove our lower bounds, in fact, is the main focus of this lecture.

Definition 3. *Pointer Chasing (**PC**) in 2-party communication*

Let Alice and Bob have functions $f_A : [m] \to [m]$ and $f_B : [m] \to [m]$. We define

$$\boldsymbol{PC}_i(f_A, f_B) = \boldsymbol{PC}_i(f_A, f_B) = \begin{cases} 1 & \text{if i=0} \\ f_A(\boldsymbol{PC}_{i-1}(f_A, f_B)) & \text{if i is odd} \\ f_B(\boldsymbol{PC}_{i-1}(f_A, f_B)) & \text{if i is even, i>0} \end{cases}$$

*In the k-round **PC** in the communication complexity model, the goal is to compute $\boldsymbol{PC}_k(f_A, f_B)$.*

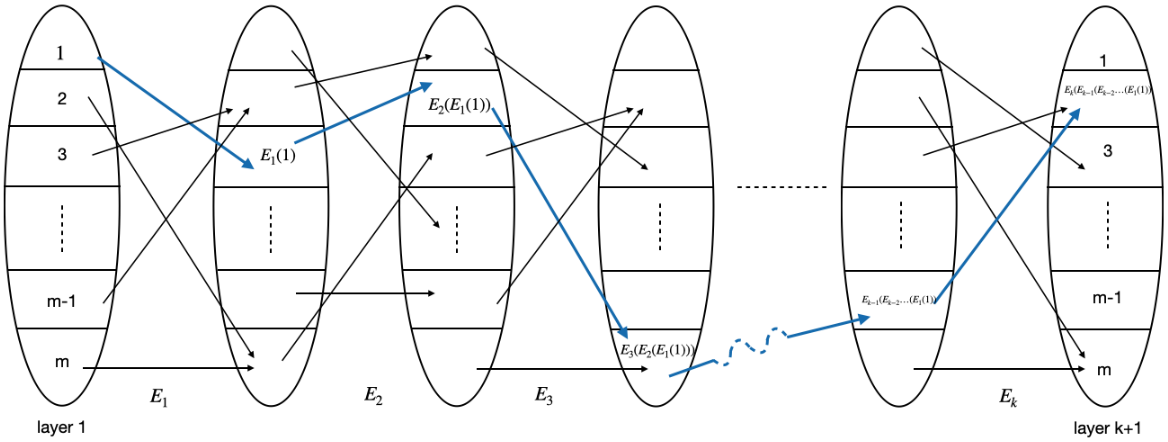


Figure 1: An instance of **PC**on a (m, k)-layered graph reaching vertex 2 in the final layer.

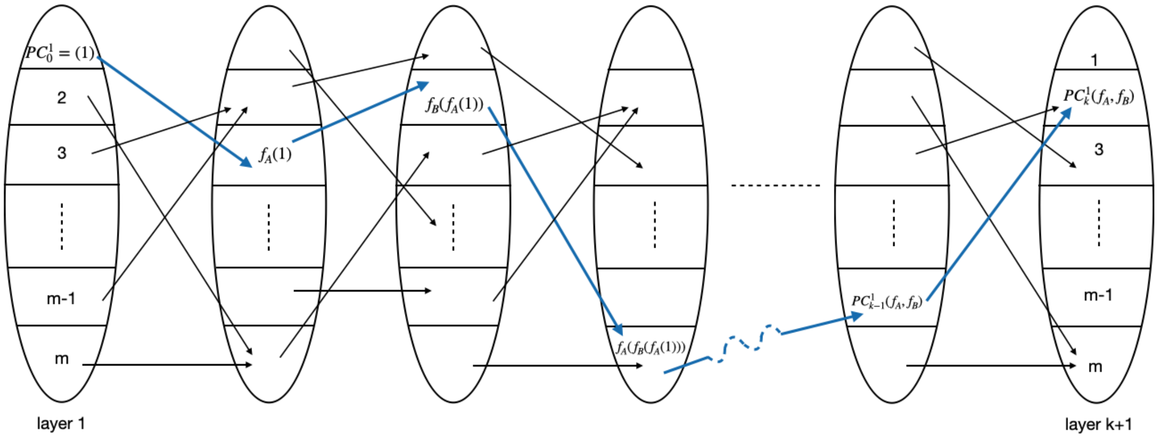


Figure 2: An instance of k-round communication pointer chasing with $PC_k^1(f_A, f_B) = (2)$ viewed as a graph.

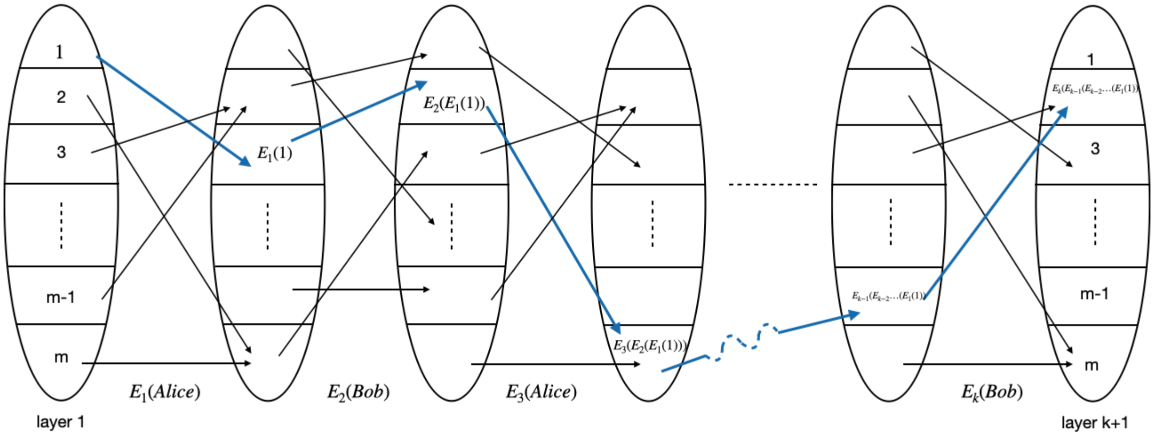


Figure 3: An instance of **PC**on a (m, k)-layered graph reaching vertex 2 in the final layer can be viewed as a communication problem with Alice receiving all edge sets E_i with odd i, and Bob receiving all edge-sets E_i with odd i. Note that in the figure k is even, so Bob gets the last edge-set.

Note that we are interested in "*chasing the pointer starting at the vertex* 1" in the definition written above. In general, we assume that the start vertex is known to both parties, if it is different than 1. Thus, WLOG we can assume the start vertex is 1.

The above problems are closely related. Specifically, given an instance of pointer chasing as a graph, we can define an equivalent instance in the communication complexity model, and vice-versa.

Given a k- round instance of **PC** in the communication model, we construct a (m, k)-layered graph. In order to define the edges, we note that in the graph version of the problem edge-sets E_i were functions from $[m] \to [m]$. We set $E_i \leftarrow f_A$ when i is odd, and $E_i \leftarrow f_B$ when i is even.

On the other hand, given an instance of **PC** in the graph model, we "give" all E_i with odd i to Alice and E_i with even i to Bob and their goal is to chase the pointer starting at 1, with Alice speaking in the first round. Observe that in this construction we require that all the odd-numbered edge-sets are the same, and all the even-numbered edge-sets are the same. However, we note that redefining the communication model version of the problem slighly helps us overcome this restriction if need be.

We wil use both definitions interchangeably from now on. We also need the direct-sum version of the question,

Definition 4. *ℓ-fold Direct-Sum of k-round Pointer Chasing*

Alice's input is ℓ functions $f_{A,1}, f_{A,2}, \ldots f_{A,\ell} : [m] \to [m]$ and Bob's input is $f_{B,1}, f_{B,2}, \ldots f_{B,\ell} : [m] \to [m]$. Their goal is to compute
$\boldsymbol{PC}_k^{\ell}(< f_{A,1}, f_{A,2}, \ldots, f_{A,\ell} >, < f_{B,1}, f_{B,2}, \ldots, f_{B,\ell} >) =< \boldsymbol{PC}_k^{1}(f_{A,1}, f_{B,1}), \boldsymbol{PC}_k^{1}(f_{A,2}, f_{B,2}), \ldots, \boldsymbol{PC}_k^{1}(f_{A,\ell}, f_{B,\ell}) >.$

In the above problems, our functions were "single-valued", in that, for any v in any vertex layer for which $P(v)$ is well-defined, $P(v)$ takes a single value. We will need its generalization to multiple values. However, for the function to be well-defined, we need the functions $f_{A,i}$ and $f_{B,i}$ to be defined on power sets. We define the d-valued version of **PC** as follows:

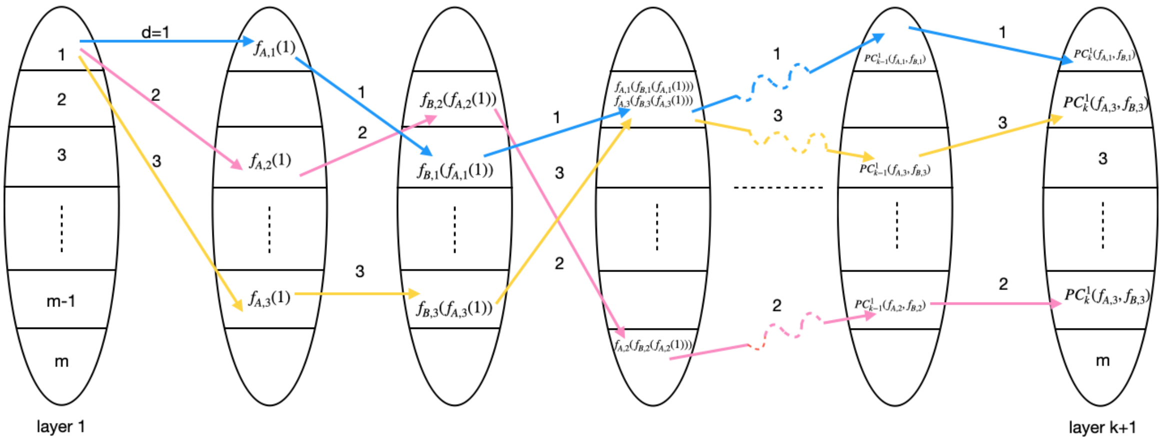

d=1
1
2
3
m-1
m
layer 1
$f_{A,1}(1)$
$f_{A,2}(1)$
$f_{A,3}(1)$
$f_{B,2}(f_{A,2}(1))$
$f_{B,1}(f_{A,1}(1))$
$f_{B,3}(f_{A,3}(1))$
$f_{A,1}(f_{B,1}(f_{A,1}(1)))$
$f_{A,3}(f_{B,3}(f_{A,3}(1)))$
$f_{A,2}(f_{B,2}(f_{A,2}(1)))$
$PC^1_{k-1}(f_{A,1}, f_{B,1})$
$PC^1_{k-1}(f_{A,3}, f_{B,3})$
$PC^1_{k-1}(f_{A,2}, f_{B,2})$
$PC^1_k(f_{A,1}, f_{B,1})$
$PC^1_k(f_{A,3}, f_{B,3})$
3
$PC^1_k(f_{A,3}, f_{B,3})$
m
layer k+1

Figure 4: An instance of k-round 3-fold direct-sum of (single-valued) pointer chasing with $PC^3_k =< 1,\ m - 1,\ 2 >$

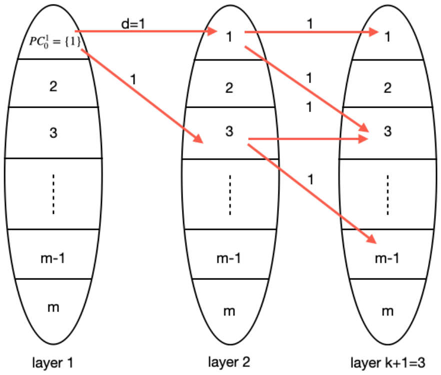


Figure 5: An instance of 2-round 1-fold direct-sum of 2-valued pointer chasing with $\mathbf{PC}^1_3 =< 1, 3, m-1 >$. Note that here $f_A(\{1\}) = \{1, 3\}$, $f_B(\{1\}) = \{1, 3\}$ and $f_B(\{3\}) = \{3, m-1\}$

Definition 5. *d-valued Pointer Chasing*

Let Alice and Bob have functions $f_A : \mathcal{P}([m]) \to \mathcal{P}([m])$ and $f_B : \mathcal{P}([m]) \to \mathcal{P}([m])$ such that $\forall i \in [m], \mid f(i) \mid \leq d$ and $\forall C \subset [m], f(C) = \cup_{i \in C} f(i)$. We define

$$\boldsymbol{PC}_i(f_A, f_B) = \boldsymbol{PC}_i(f_A, f_B) = \begin{cases} \{1\} & \text{if i=0} \\ f_A(\boldsymbol{PC}_{i-1}(f_A, f_B)) & \text{if i is odd} \\ f_B(\boldsymbol{PC}_{i-1}(f_A, f_B)) & \text{if i is even, i>0} \end{cases}$$

In the k-round d-valued $\boldsymbol{PC}$ in the communication complexity model, the goal is to compute $\boldsymbol{PC}_k(f_A, f_B)$ as in this definition, i.e. when the functions are d-valued.

We also define the d-valued version of the direct-sum question similarly.

The neighbourhood-PC problem, which we define below, can be thought of as though we forget the 'labels' of each of the folds of the direct-sum version of the problem, and chase 'neighbourhods' starting with the

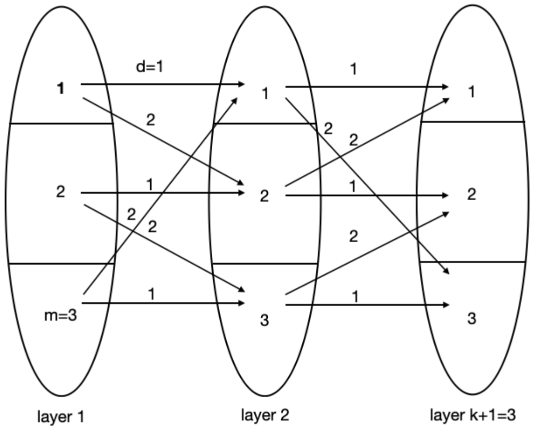


Figure 6: An instance of $\mathbf{PC}_2^2$ and $\mathbf{NPC}_2^2$ with $\mathbf{PC}_2^2(< f_{A,1}, f_{A,2} >, < f_{B,1}, f_{B,2} >) =< 1, 1 >$ and $\mathbf{NPC}_2^2(f_A, f_B) = \{1, 2, 3\}$ for single-valued $f_{A,1}, f_{A,2}, f_{B,1}, f_{B,2}$.

start vertex in the first layer. Goal is to find the nieghbourhood chased in the final layer.

Definition 6. *k-round ℓ Neighbourhood-Pointer Chasing ($\boldsymbol{NPC}_k^\ell$)*

Alice's input is d-valued functions $f_{A,1}, f_{A,2}, \ldots f_{A,\ell} : \mathcal{P}([m]) \to \mathcal{P}([m])$ and Bob's input is d-valued functions $f_{B,1}, f_{B,2}, \ldots f_{B,\ell} : \mathcal{P}([m]) \to \mathcal{P}([m])$. We define $\ell \cdot d$-valued functions $f_A(j) := f_{A,1} \cup f_{A,2} \cup \ldots \cup f_{A,\ell}$ and $f_B(j) := f_{B,1} \cup f_{B,2} \cup \ldots \cup f_{A,\ell}$. We also define,

$$\boldsymbol{NPC}_i^\ell(< f_{A,1}, f_{A,2}, \ldots, f_{A,\ell} >, < f_{B,1}, f_{B,2}, \ldots, f_{B,\ell} >)$$
$$= \begin{cases} \{1\} & \textit{if i=0} \\ f_A(\boldsymbol{NPC}_{i-1}^\ell(< f_{A,1}, f_{A,2}, \ldots, f_{A,\ell} >, < f_{B,1}, f_{B,2}, \ldots, f_{B,\ell} >)) & \textit{if } i \textit{ is odd} \\ f_B(\boldsymbol{NPC}_{i-1}^\ell(< f_{A,1}, f_{A,2}, \ldots, f_{A,\ell} >, < f_{B,1}, f_{B,2}, \ldots, f_{B,\ell} >)) & \textit{if } i \textit{ is even, } i>0 \end{cases}$$

In the k-round ℓ-fold neighbourhood chasing problem, the goal is to compute $\boldsymbol{NPC}_k^\ell$.

We will consider $\ell = d$ in the above definitions for our reductions.

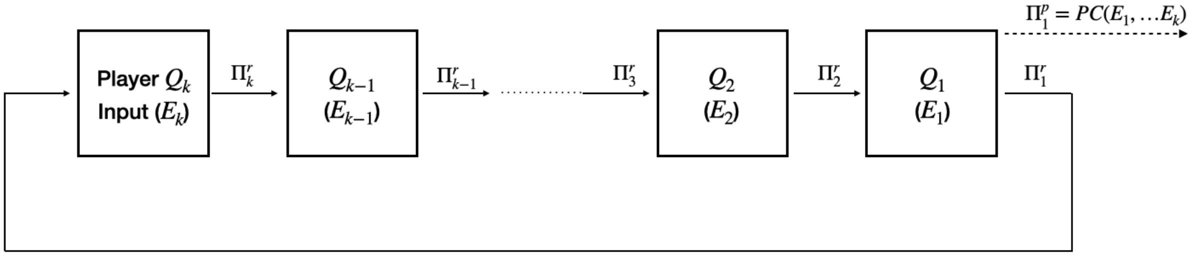
2 A Lower Bound on Pointer Chasing

In this section, we prove a lower bound on an instance of pointer chasing using information theoretic tools. We will consider **PC** instances with edge sets between different vertex layers chosen uniformly at random and independently. Our goal is to prove lower bound on streaming algorithms on an adversarial stream $E_k|| \ldots E_1$. In this inverted stream, we expect the protocol to always "lag" behind if we do not store many edges or make many passes (in other words communicate a lot).

Theorem 7. *Consider a p-pass s-space streaming algorithm $\mathcal{A}$ for $\boldsymbol{PC}$ over random (m, k)-layered graphs with edge sets $E_1, \ldots, E_k$ given in the stream $E_k|| \ldots ||E_1$. For $\gamma \in (0, 1)$, if $\mathcal{A}$ succeeeds with probabiliity at least $\frac{1}{m} + \gamma$ then either $p > k - 1$ or $s = \Omega((\frac{\gamma}{k})^2 \cdot m)$.*

2.1 A Multiplayer Communication Game

We consider the following k-player p-round communication game, which we define in the context of the **PC** problem.



Complexity measure: *maximum* size of message sent by *any* player in *any* round.

Figure 7: Multi-player communication game starting with Q_k who sends a message to Q_{k-1} in the first round, based on the their input E_k. In round $r \in [p]$, (player Q_i sends a message Π_i^r to Q_{i-1}) mod b. At the end of$p$ rounds, player Q_1 outputs Π_1^p, the answer to **PC** on the (m, k)-layered graph consisting of edge-sets $(E_1, \ldots, E_k)$.

1. We have k players $Q_1, Q_2, \ldots, Q_k$ and the game consists of p rounds.
2. For $i \in [k]$, Q_i has the input E_i.
3. In the first round, the player Q_k speaks first and sends a message Π_k^1 to player Q_{k-1}. Based on their input E_{k-1}, and the message Π_k^1 received from Q_k, Q_{k-1} sends a message Π_{k-1}^1 to Q_{k-2}. This process continues all the way to player Q_1 who sends a message Π_1^1 to player Q_k. Round 1 is now complete.
4. Continuing in a similar fashion, in the r-th round, player Q_k speaks first, and based on all previous messages received and their input, sends a message Π_k^r to player Q_{k-1}. Based on their input E_{k-1}, and all previous messages received, Q_{k-1} sends a message Π_{k-1}^r to Q_{k-2}. This process continues all the way to player Q_1 who sends a message Π_1^r to player Q_k. Round r is now complete.
5. At the end of the p-th round, the player Q_1 outputs the answer to the **PC** problem on (m, k)-layered graphs with vertex layers $V_1, V_2, \ldots, V_{k+1}$ with the edge set E_i connecting layer V_i to V_{i+1}, and the goal of the game is for Q_1 to be able to output this answer.

The measure of communication complexity in this game is the size of the message sent by *any* player in *any* round of the game. We first show that a good streaming algorithm for **PC** on the stream $E_k || \ldots || E_1$ implies a low communication protocol in the above multi-player game.

Lemma 8. *If there exists a p-pass s-space streaming algorithm $\mathcal{A}$ solving **PC** on (m, k)-layered graphs given in the stream $E_k || \ldots || E_1$, there exists a p-round protocol such that any player in any round does not send a message of more than s bits.*

Proof. For $i \in [k], r \in [p]$:

Let m_i^r be the memory state of the algorithm after scanning $E_k || \ldots || E_i$ in pass r. In the communication game, player Q_i sends m_i^r to Q_{i-1}. In other words, $m_i^r = \Pi_i^r$. Assuming player Q_{i-1} sends m_{i-1}^r, Q_i can compute m_i^r because:

- They have access to m_{i-1}^r sent by Q_{i-1} in message Π_{i-1}^r
- They have access to E_i

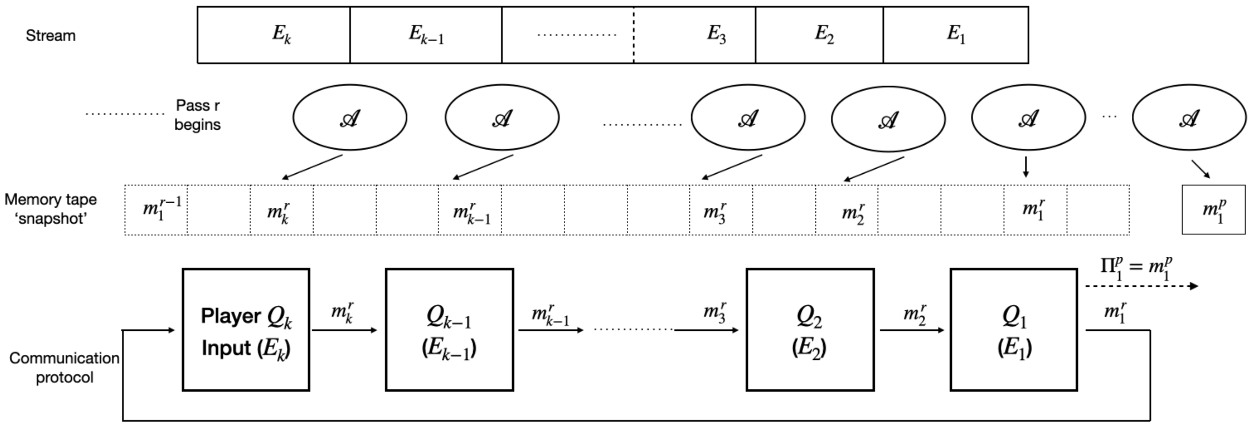


Figure 8: A streaming algorithm for the input stream induces a communication protocol in the multi-party communication game.

- They have access to $\mathcal{A}$

Thus, player Q_i can run $\mathcal{A}$ starting from the state m_{i-1}^r on their stream E_i and compute m_i^r.

At the end of p rounds of the communication game, thus, player Q_1 is able to output m_1^p, which is the output to **PC** on the stream $E_k||\dots||E_1$ - by the correctness of algorithm $\mathcal{A}$. Also, since the messages sent by the players are memory states of the algorithm, no player sends more than s bits in any round. □

Now, we show a lower bound on the communication complexity of any protocol computing **PC** .

We define the random variable $Z_r = (P_1(s), \cdots P_{r+1}(s), \Pi^1, \cdots, \Pi^r)$ which is the information all players have after round r. Initially, the inputs of the players are independent, our goal is to show that low amounts of communication do not help, and the distribution of output conditioned on messages sent, is close to being uniform (the original distribution, without any communication). We do this by removing the conditioned message one by one, in some order, by a series of claims which follow. Notation is self-explanatory.

Formally, we prove the following lemma:

Lemma 9. *(Similar to the result in [5] but for total variational distance instead of triangular discrimination)*

In the multiplayer communication game, if no player sends a message of size more than $\frac{1}{500r} \cdot (\frac{\gamma}{k})^2 \cdot m$, *then* $\mathbb{E}_{Z_r}||P_{r+2}(s) - (P_{r+2}(s)|Z_r)||_{tvd} < \gamma \cdot \frac{r}{k}$

In particular, for $p = r = k-1$, $\mathbb{E}_{Z_p}||P_{k+1}(s) - (P_{k+1}(s)|Z_p)||_{tvd} < \gamma$.

First, let us see why proving Lemma 9 proves Theorem 7.

Lemma 9 implies Theorem 7. By definition of correctness of the protocol,

$\Pr\ (\Pi \text{ is correct}) = \mathbb{E}_{Z_p}\ \Pr_{P_{k+1}|Z_p}(P_{k+1}(s) \text{ equals the fixed vertex v output by } Z_p)$

$\leq \mathbb{E}_{Z_p}[\ \Pr_{P_{k+1}}(P_{k+1}(s) \text{ equals the fixed vertex v}) + ||P_{k+1}(s) - (P_{k+1}(s)|Z_p)||_{tvd}]$

$= \frac{1}{m} + \mathbb{E}_{Z_p}||P_{k+1}(s) - (P_{k+1}(s)|Z_p)||_{tvd}$

$\leq \frac{1}{m} + \gamma$

where the first inequality is using Property (19). The second equality is because the E_i were chosen uniformly, and the last inequality is due to Lemma 9.

The mutual information is zero, thus the random variables must be independent, by Property (18) proving the claim. □

We now prove Lemma 9 using a series of claims. We will use induction. Note that the base case is true since $Z_0 = P_1(s)$ is fixed and Π^0 is empty. Thus we assume $\mathbb{E}_{Z_{r-1}}||P_{r+1}(s) - (P_{r+1}(s)|Z_{r-1})||_{tvd} < \gamma \cdot \frac{r-1}{k}$.

We first prove that " in round r, we can focus on the messages sent and input of player $r+1$" in the following claim.

Claim 10. *For any choice of $Z_r = (Z_{r-1}, \Pi^r_{-(r+1)}, \Pi^r_{r+1}, P_{r+1}(s))$,*

$P_{r+2}(s) \perp \Pi^r_{-(r+1)} \mid Z_{r-1}, \Pi^r_{r+1}, P_{r+1}(s)$

Proof. We analyse

$\mathbb{I}(P_{r+1}(v); \Pi^r_{-(r+1)} \mid Z_{r-1}\Pi^r_{r+1}, P_{r+1}(s))$

which is

$\leq \mathbb{I}(E_{r+1}; E_{-(r+1)} \mid Z_{r-1}\Pi^r_{r+1}, P_{r+1}(s))$

By Property (17) because $P_{r+2}(v)$ is a function of E_{r+1} and $P_{r+1(s)}$ and $\Pi^r_{-(r+1)}$ is a function of $E_{-(r+1)}$, Z_{r-1} and Π^r_{r+1}

$\leq \mathbb{I}(E_{r+1}; E_{-(r+1)}) = 0$

This is a repeated application of Property (15) since each conditioned RV is a function of exactly one side of the mutual information term, contiioned on the remaining ones (we start removing variables with r=0 first, in appropriate order). The last equality to zero is because the edge-sets were chosen independently, at random.

□

Thus,

$\mathbb{E}_{Z_r}||P_{r+2}(s) - (P_{r+2}(s)|Z_r)||_{tvd}$
$= \mathbb{E}_{Z_{r-1},\Pi^r_{r+1},P_{r+1}(s)}||P_{r+2}(s) - (P_{r+2}(s)|Z_{r-1}, \Pi^r_{(r+1)}, P_{r+1}(s))||_{tvd}$ using the preceding Claim 10
$= \mathbb{E}_{Z_{r-1},\Pi^r_{r+1}}\mathbb{E}_{v\sim P_{r+1}(s)|Z_{r-1},\Pi^r_{r+1}}||P_{r+2}(s) - (E_{r+1}(v)|Z_{r-1}, \Pi^r_{(r+1)}, P_{r+1}(s) = v)||_{tvd}$ by definition of expectation

Claim 11. *For any choice of $Z_{r-1}, \Pi^r_{(r+1)}$ and $v \in V_{r+1}$,*

$E_{r+2}(v) \perp (P_{r+1}(s) = v) \mid Z_{r-1}, \Pi^r_{r+1}$

Proof. Similar to that of Claim 10. □

Thus,

$\mathbb{E}_{Z_{r-1},\Pi^r_{r+1}}\mathbb{E}_{v\sim P_{r+1}(s)|Z_{r-1},\Pi^r_{r+1}}||P_{r+2}(s) - (E_{r+1}(v)|Z_{r-1}, \Pi^r_{(r+1)}, P_{r+1}(s) = v)||_{tvd}$

equals

$= \mathbb{E}_{Z_{r-1},\Pi^r_{r+1}}\mathbb{E}_{v\sim P_{r+1}(s)|Z_{r-1},\Pi^r_{r+1}}||P_{r+2}(s) - (E_{r+1}(v)|Z_{r-1}, \Pi^r_{(r+1)})||_{tvd}$

using the preceding Claim 11

Claim 12. *For any choice of Z_{r-1},*

$P_{r+1}(s) \perp \Pi^r_{r+1} \mid Z_{r-1}$

Proof. Similar to that of Claim 10. □

Thus,

$\mathbb{E}_{Z_{r-1},\Pi^r_{r+1}} \mathbb{E}_{v \sim P_{r+1}(s)|Z_{r-1},\Pi^r_{r+1}} ||P_{r+2}(s) - (E_{r+1}(v)|Z_{r-1}, \Pi^r_{(r+1)})||_{tvd}$

equals

$= \mathbb{E}_{Z_{r-1},\Pi^r_{r+1}} \mathbb{E}_{v \sim P_{r+1}(s)|Z_{r-1}} ||P_{r+2}(s) - (E_{r+1}(v)|Z_{r-1}, \Pi^r_{(r+1)})||_{tvd}$

now using the preceding Claim 12 implies

$\leq \mathbb{E}_{Z_{r-1},\Pi^r_{r+1}} [\mathbb{E}_{v \sim unif(V_{r+1})} ||P_{r+2}(s) - (E_{r+1}(v)|Z_{r-1}, \Pi^r_{(r+1)})||_{tvd} + ||unif(V_{r+1}) - (P_{r+1}(s)|Z_{r-1})||_{tvd}]$

which using Property (19) implies

$= \mathbb{E}_{Z_{r-1},\Pi^r_{r+1}} [\mathbb{E}_{v \sim unif(V_{r+1})} ||P_{r+2}(s) - (E_{r+1}(v)|Z_{r-1}, \Pi^r_{(r+1)})||_{tvd} + ||E_{r+1}(s) - (P_{r+1}(s)|Z_{r-1})||_{tvd}]$

now using the induction hypothesis implies,

$\leq \mathbb{E}_{Z_{r-1},\Pi^r_{r+1}} [\mathbb{E}_{v \sim unif(V_{r+1})} ||P_{r+2}(s) - (E_{r+1}(v)|Z_{r-1}, \Pi^r_{(r+1)})||_{tvd}] + \gamma \cdot \frac{r-1}{k}$

Claim 13. *For any choice of $(\Pi^1_{r+1}, \ldots, \Pi^r_{r+1})$ and any $v \in V_{r+1}$,*

$E_{r+1}(v) \perp Z_{r-1} | (\Pi^1_{r+1}, \ldots, \Pi^r_{r+1})$

Proof. We consider

$\mathbb{I}(E_{r+1}(v); Z_{r-1} | (\Pi^1_{r+1}, \ldots, \Pi^r_{r+1}) = \mathbb{I}(E_{r+1}(v); \Pi^1_{-(r+1)}, \ldots, \Pi^r_{-(r+1)} | (\Pi^1_{r+1}, \ldots, \Pi^r_{r+1})$

after restructuring the random variable Z_{r-1} using the conditioned ones.

which, by Property (17) (since $E_{-(r+1)}$ determines $\Pi^1_{-(r+1)}, \ldots, \Pi^r_{-(r+1)}$ conditioned on $(\Pi^1_{r+1}, \ldots, \Pi^r_{r+1}))$ equals

$= \mathbb{I}(E_{r+1}; E_{-(r+1)} | \Pi^1_{r+1}, \ldots, \Pi^r_{r+1}))$

now, $M_{r+1} \perp \Pi^1_{>r+1}$ because the players' inputs are independent and applying Property (15) implies

$\leq \mathbb{I}(E_{r+1}; E_{-(r+1)} | \Pi^1_{>r+1}, \Pi^1_{r+1}, \ldots, \Pi^r_{r+1})$

again, $M_{r+1} \perp \Pi^1_{<r+1}, \Pi^1_{r+1}$ because the players' inputs are independent and applying Property (15) implies

$\leq \mathbb{I}(E_{r+1}; E_{-(r+1)} | \Pi^1_{<r+1} \Pi^1_{>r+1}, \Pi^1_{r+1}, \ldots, \Pi^r_{r+1})$

which, by restructuring using notation $\Pi^1_{<r+1} \Pi^1_{>r+1}, \Pi^1_{r+1} = \Pi^1$ equals

$\leq \mathbb{I}(E_{r+1}; E_{-(r+1)} | \Pi^1, \Pi^2_{r+1} \ldots, \Pi^r_{r+1})$

now we can repeat the same argument and "bring in" other random variables in the conditioning one by one changing them to Π^2, then Π^3 and so on. Thus,

$\leq \mathbb{I}(E_{r+1}; E_{-(r+1)} | \Pi^1, \Pi^2 \ldots, \Pi^r)$

which is

$\leq \mathbb{I}(E_{r+1}; E_{-(r+1)}) = 0$

where the using the independence of the players' inputs.

□

Thus,

$\mathbb{E}_{Z_{r-1},\Pi^r_{r+1}}\mathbb{E}_{v\sim unif(V_{r+1})}||P_{r+2}(s) - (E_{r+1}(v)|Z_{r-1},\Pi^r_{(r+1)})||_{tvd} + \gamma\cdot\frac{r-1}{k}$

equals

$= \mathbb{E}_{Z_{r-1},\Pi^r_{r+1}}\mathbb{E}_{v\sim unif(V_{r+1})}||P_{r+2}(s) - (E_{r+1}(v)|\Pi^1_{r+1},\ldots,\Pi^r_{r+1})||_{tvd} + \gamma\cdot\frac{r-1}{k}$

using the preceding Claim 13. Now, dropping Z_{r-1} since in the expectation,

$= \mathbb{E}_{\Pi^r_{r+1}}\mathbb{E}_{v\sim unif(V_{r+1})}||P_{r+2}(s) - (E_{r+1}(v)|\Pi^1_{r+1},\ldots,\Pi^r_{r+1})||_{tvd} + \gamma\cdot\frac{r-1}{k}$.

We now analyse this last remaining term in the expectation. Looking closely at the right hand side term inside the total variational distance it is a function of the player Q_{r+1} and the question is whether the distribution of a random edge in their input can be changed by the previous messages they sent in the first r rounds. We expect this to be small is the size of the messages is small, which the following claim proves.

Claim 14. *For any* $s < \frac{1}{500r}\cdot(\frac{\gamma}{k})^2\cdot m$,

$\mathbb{E}_{\Pi^1_{r+1},\ldots,\Pi^r_{r+1}}\mathbb{E}_{v\sim unif(V_{r+1})}||P_{r+2}(s) - (E_{r+1}(v)|\Pi^1_{r+1},\ldots,\Pi^r_{r+1})||_{tvd} < \gamma\cdot\frac{1}{k}$

Proof. We first start by redefining the right hand side for ease of notation. Define $\Pi = \Pi^1_{r+1},\ldots,\Pi^r_{r+1}$,$\mathcal{D}$ is V_{r+1} viewed as a domain of size $|\mathcal{D}|$, from which v is chosen uniformly. Also, $E = E_{r+1}$:

$\mathbb{E}_{\Pi^1_{r+1},\ldots,\Pi^r_{r+1}}\mathbb{E}_{v\sim unif(V_{r+1})}||P_{r+2}(s) - (E_{r+1}(v)|\Pi^1_{r+1},\ldots,\Pi^r_{r+1})||_{tvd}$

$=$

$\mathbb{E}_{\Pi}\mathbb{E}_{v\sim unif(\mathcal{D})}||E(v) - (E(v)|\Pi)||_{tvd}$

which, by symmetry of tvd equals

$\mathbb{E}_{\Pi}\mathbb{E}_{v\sim unif(\mathcal{D})}||(E(v)|\Pi) - E(v)||_{tvd}$

which, using Pinsker's inequality Property (20) is upper bounded by

$\leq \mathbb{E}_{\Pi}\mathbb{E}_{v\sim unif(\mathcal{D})}\sqrt{\frac{1}{2}\ \mathbb{D}\ (E(v)|\Pi\ ||\ E(v))}$

now, $\sqrt{.}$ is a concave function, using Jensen's inequality Property (21)

$\leq \sqrt{\frac{1}{2}\mathbb{E}_{\Pi}\mathbb{E}_{v\sim unif(\mathcal{D})}\ \mathbb{D}\ (E(v)|\Pi\ ||\ E(v))}$

since v is uniform over $\mathcal{D}$,

$= \sqrt{\frac{1}{2}\cdot\frac{1}{|\mathcal{D}|}\sum_{v\in\mathcal{D}}\mathbb{E}_{\Pi}\ \mathbb{D}\ (E(v)|\Pi\ ||\ E(v))}$

which equals, by Property (22)

$= \sqrt{\frac{1}{2}\cdot\frac{1}{|\mathcal{D}|}\sum_{v\in\mathcal{D}}\ \mathbb{I}(E(v);\Pi)}$

Now, for any v,

$\mathbb{I}(E(v);\Pi)$
$= \mathbb{H}(E(v)) - \mathbb{H}(E(v)|\Pi)$
$= \log|\mathcal{D}| - \mathbb{H}(E(v)|\Pi)$

Consider, $\sum_{v\in\mathcal{D}}\mathbb{I}(E(v);\Pi) = |\mathcal{D}|\log|\mathcal{D}| - \sum_v \mathbb{H}(E(v)|\Pi)$

now, $\sum_v \mathbb{H}(E(v)|\Pi) \geq \mathbb{H}(E|\Pi) \geq \mathbb{H}(E) - \mathbb{H}(\Pi) = |\mathcal{D}| \log |\mathcal{D}| - r \cdot s$

because Π consists of r messages, each consisting of s bits.

Thus, $\sum_{v \in \mathcal{D}} \mathbb{I}(E(v); \Pi) \leq r \cdot s$.

$\leq \sqrt{\frac{1}{2} \cdot \frac{1}{m} \cdot r \cdot s} \leq \frac{\gamma}{k}$ as $s < \frac{1}{500r} \cdot (\frac{\gamma}{k})^2 \cdot m$

□

We have thus completed the proof of Lemma 9, thus implying Theorem 7.

2.2 Some Facts from Information Theory

In our proofs, we use the following properties, many of which can be found in [1]. Each property is followed by a short description about it.

Property 15. *For random variables A, B, C, D, $\mathbb{I}(A; B|C, D) \leq \mathbb{I}(A; B|C)$ if $A \perp D|B, C$*

This property says that even after revealing random variable D, which is such that the random variable A is independent of D conditioned on B, it can never reveal more information about A than in the case when we weren't revealed D.

Property 16. *For random variables A, B, C, D, $\mathbb{I}(A; B \mid C) \leq \mathbb{I}(A; B \mid C, D)$ if $A \perp D|C$*

This property says that conditioning on a random variable D which is such that it is (conditionally) independent of the RV of interest A, can only increase the information B gives about A.

Property 17. *(Data-Processing Inequality) For a deterministic function $f(A)$, $\mathbb{I}(f(A); B|C) \leq \mathbb{I}(A; B|C)$*

This property says that applying a deterministic function on one of the random variables we are interested in can never increase the mutual information.

Property 18. $\mathbb{I}(A; B|C) \geq 0$ *with equality if and only if $A \perp B|C$*

This property says that the mutual information can never be less than zero, and is zero if and only if the random variables are (conditionally) independent of each other. It is proved using the non-negativity of KL-divergence. Thus, if we can prove that the (conditional) mutual information between two random variables is zero, we prove that the random variables are (conditionally) independent of each other.

Property 19. *If μ and ν are distributions for an event $\mathcal{E}$, $\Pr_\mu(\mathcal{E}) \leq \Pr_\nu(\mathcal{E}) + ||\mu - \nu||_{tvd}$.*

This property says that if we mistake distribution ν for distribution μ, on any event $\mathcal{E}$, we perform worse by an additive factor of *tvd* between the two distributions.

Property 20. *Pinsker's inequality: $||\mu - \nu||_{tvd} \leq \sqrt{\frac{1}{2} \cdot \mathcal{D}(\mu||\nu)}$*

Pinsker's inequality helps us use information-theoretic tools on distributions by upper bounding the *tvd* by the square-root of KL divergence, when we need to bound their *tvd*, and is tight up to constant factors.

Property 21. *Jensen's inequality: For any random variable X and any convex function Φ, $\Phi[\mathcal{E}(X)] \leq \mathcal{E}[\Phi(X)]$.*

Jensen's inequality is stated here in the context of probability theory.

Property 22. *For random variables A, B, C,*

$$\mathbb{I}(A; B|C) = \mathbb{E}_{(b,c)\sim(B,C)}[\mathbb{D}((A|B = b, C = c)||(A|C = c))]$$

This property can be proved by definition of KL-divergence and expectation, and helps us analyse the KL-divergence instead of mutual information.

3 A Lower Bound on Source-Reachability in Directed Graphs via Pointer Chasing

In this section, we prove lower bounds for the Source-Reachability problem in directed graphs which we define as follows,

Definition 23. *Given a directed graph $G(V, E)$, and a source vertex $s \in V$, the source-reachability problem is to identify all nodes reachable from s in G.*

We prove lower bounds for this problem using a lower bound on the communication complexity of the **NPC** problem. In particular, we will use the following theorem for our lower bound:

Theorem 24. *(Theorem 4.5 in [2]) Any k round protocol for $\boldsymbol{NPC}_k^d$ that succeeds with probability at least $2/3$ must communicate at least $\mathbf{R}_{1/3}^k(NPC_k^d) = \Omega(\frac{dm}{(k+1)^3} - 6d^{k+1}\log m - d(k+1)\log m - 2d)$ bits.*

We will prove this using a lower bound for the direct sum of **PC**, i.e. $\mathbf{PC}_k^d$ as in the following theorem,

Towards proving this, we will use result 2 from [3] which is stated here as follows,

Theorem 25. *(Result 2 from [3]) Let d, k be positive integers, and $\epsilon, \delta > 0$. Let $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ be a function. Then, $R_\delta^k(f^d) \geq \frac{d}{k} \cdot (\frac{\epsilon^2}{2} \cdot C_{[\,],\delta+2\epsilon}^k(f) - 2)$.*
Here $R_\delta^k(f)$ denotes the k-round private communication complexity of f where the protocol is allowed to err with probability at most δ on any input. Also, $C_{[\,],\delta}^k(f)$ denotes the maximum, over all product distributions μ on the inputs, of the deterministic k-round communication complexity of f, where the protocol is allowed to err with probability at most δ.

In the proof of Theorem 7 (specifically Claim 14) we proved that $\mathbf{C}_{1/3}^k(\mathbf{PC}_k^1) = r \cdot s = \Omega(\frac{m}{k^2})$. Applying the preceding Theorem 25, we get that $\mathbf{R}_{1/3}^k(\mathbf{PC}_k^d) = \Omega(\frac{d \cdot m}{k^3})$, which we state in the following theorem, in a form that is used to prove our lower bounds,

Theorem 26. *(Theorem 4.4 in [2]) Any k round protocol for $\boldsymbol{PC}_k^d$ that succeeds with probability at least $2/3$ must communicate at least $\mathbf{R}_{1/3}^k(\boldsymbol{PC}_k^d) = \Omega(\frac{dm}{k^3} - dk\log m - 2d)$ bits.*

Now, we are ready to prove Theorem 24,

Proof of Theorem 24. For sake of contradiction, assume there exists a k-round protocol Π for $\mathbf{NPC}_k^d$ that succeeds with probability at least $2/3$ communicates $o(dm/(k+1)^3 - d(k+1)\log m - 2d - 12d^{k+1}\log m)$ bits. Using Π, we construct a $(k+1)$-round protocol Π' for PC_{k+1}^d which violates Theorem 26.

We first define $f_A^*(j) = \{f_{A,i}(j) : i \in [d]\}$ and $f_B^*(j) = \{f_{B,i}(j) : i \in [d]\}$

Π' simulates Π, and additionally, on the j-th message, sends the triples :

$T_{j-1} = \{< a, b, f_{C,a}(b) >: b \in NPC_{j-2}^d(f_A^*, f_B^*)\}$, where $C = \begin{cases} A & \text{if j is even,} \\ B & \text{if j is odd} \end{cases}.$

Proof of correctness is inductive.

We need to bound the amount of extra communication needed using the following observations:

- Counting the number of triplets: $a \in [d]$, $b \in g^{j-2,d}(f_A^*, f_B^*)$ and $|g^{j-2,d}(f_A^*, f_B^*)| \leq d^{j-2}$
- $3\log m$ bits are required to describe each triple

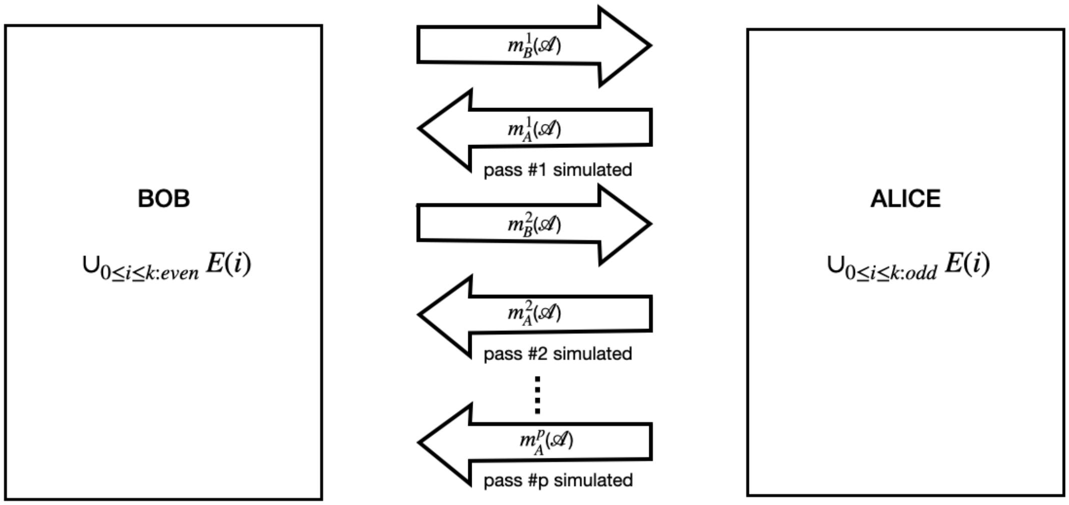


Figure 9: A p-pass s-space streaming algorithm $\mathcal{A}$ for source-reachability on stream $\mathcal{S}$ induces a $(2p \cdot s)$ bit communication protocol for $\mathbf{NPC}_k^d$.

Thus, the total amount of extra communication needed is $\sum_{i\in[k+1]} 3d^i \log m \leq 6d^{k+1} \log m$ bits, after $k+1$ rounds.

Thus, Π' solves $\mathbf{PC}_k^d$ using communication $o(dm/(k+1)^3 - d(k+1)\log m - 2d)$ contradicting Theorem 26. □

We are now ready to prove a space-pass trade-off for the source-reachability problem,

Theorem 27. *(Extension of Theorem 4.1 in [2]) For any constant k, any k-pass algorithm that identifies all vertices reachable from a source with probability at least 2/3 requires at least $\Omega(n^{1+1/k})$ space.*

Proof. On an instance (f_A, f_B) $\mathbf{NPC}_k^d$, we construct the underlying layered graph $G(V, E)$. Formally, $V = \cup_{0\leq i\leq k}\{v_1^i, \ldots, v_m^i\}$ and define the set of directed edges from $\{v_1^{i-1}, \ldots, v_m^{i-1}\}$ to $\{v_1^i, \ldots, v_m^i\}$ by

$$E_i = \begin{cases} \{(v_j^{i-1}, v_\ell^i) : \ell \in f_A(j)\} & \text{if i is odd,} \\ \{(v_j^{i-1}, v_\ell^i) : \ell \in f_B(j)\} & \text{if i is even} \end{cases}.$$

Consider the stream $\mathcal{S} = \cup_{0\leq i\leq k:even} E_i || \cup_{0\leq i\leq k:odd} E_i$. Consider the following communication game - Alice has $\cup_{0\leq i\leq k:even} E_i$ and Bob has $\cup_{0\leq i\leq k:odd} E_i$, and their goal is solve $\mathbf{NPC}_k^d$. We show that a good streaming algorithm for source-reachability induces a low-communication protocol in this game.

Our simulation consists of 2 rounds for each pass of the streaming algorithm.

Alice speaks first - simulates the streaming algorithm till the end of her input and sends the memory state of the streaming algorithm to Bob who starts in this state and continues simulating the streaming algorithm till the end of his stream. He then sends the state the streaming algorithm is in, back to Alice. The simulation of the first pass of the streaming algorithm is now complete. Now, Alice starts in the memory state sent by Bob and then simulates the streaming algorithm till the end of her input stream. She then sends the state of the algorithm to Bob, who can now start in this state and continue simulating the streaming algorithm till the end of his input stream, send the memory state to Alice. The simulation of the second pass of the streaming algorithm is now complete. Continuing in this fashion, Alice and Bob take 2 rounds to simulate each pass of the streaming algorithm, and at the end of $2p$-rounds, have the answer to the source-reachability problem. Note that since the streaming algorithm was a s-space algorithm, each message the players send contains s bits. Thus, the total amount of communication used is $2p \cdot s$.

Now, we use the fact that vertices reachable from the source vertex and in the k-th layer of this directed graph is the answer to the $\mathbf{NPC}_k^d$ problem we started with.

Thus, using Theorem 24, for $m = n/(k+1)$, $d = c(m/\log m)^{1/k}$ for constant c, either $2p > k$ or $s = \Omega(n^{1+1/k})$ proving the space-pass tradeoff we desired.

□

References

[1] T. M. Cover and J. A. Thomas. *Elements of information theory (2. ed.).* Wiley, 2006. 11

[2] J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, and J. Zhang. Graph distances in the data-stream model. *SIAM J. Comput.*, 38(5):1709–1727, 2008. 12, 13

[3] R. Jain, J. Radhakrishnan, and P. Sen. A direct sum theorem in communication complexity via message compression. In *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, June 30 - July 4, 2003. Proceedings*, pages 300–315, 2003. 12

[4] E. Kushilevitz and N. Nisan. *Communication complexity.* Cambridge University Press, 1997.

[5] A. Yehudayoff. Pointer chasing via triangular discrimination. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:151, 2016. 7