CS697A-Lower bounds for Graph Streaming Algorithms

Devansh Shringi(17807239)

Advisor: Prof. Raghunath Tewari

October 6, 2021

Contents

Chapter 1

Introduction

Graph streaming algorithms process the input graph with n known vertices by making one or a few passes over the sequence of its unknown edges (given in an arbitrary order) and using a limited memory. This memory should definitely be much smaller than $O(n^2)$ otherwise we can simply store the edges. Our main focus will be on the problem of st-reachability in directed graphs.

There are several problems which are proved to be difficult in the streaming settings. A lot of these lower bounds on the space were developed recently. For the single-pass algorithm $\Omega(n^2)$ space lower bound is known for a lot of major problem like maximum matching and minimum vertex cover, (directed) reachability and topological sorting, shortest path and diameter , minimum or maximum cut , maximal independent set , dominating set etc.

On mutli-pass algorithms, for several problems a large gap is known between one-pass and multiple pass algorithms. For example, st min-cut and global min-cut in undirected graphs have $\Omega(n^2)$ space lower bound, but can be done in $\tilde{O}(n)$ and $\tilde{O}(n^{5/3})$ space in 2 passes. However, in [AR20], they showed that for st-reachability the lower bound holds for 2-passes as well.

1.1 Previous Lower Bounds

For the problem of st-reachability, the hard case case before [AR20] was the set of random graphs where s can reach $\theta(\sqrt{n})$ vertices S, and t can be independently reached by $\theta(\sqrt{n})$ vertices T. In this distribution of random graph, using Birthday paradox, we can show that constant probability for existing and not existing a path between s and t. Then, they show that we need to calculate S and T explicitly for the graphs. At last using reduction to pointer chasing problems, they show that to calculate S and T in p-passes, we need $O(n^{1+1/2(p+1)}$, which can be bounded by $\tilde{O}(n^{3/2})$.

We will have a close look at the techniques and proof of the $n^{2-o(1)}$ space lower bound for the problem of st-reachability in 2-passes, given in [AR20]. We will try to extend this approach for getting a similar space lower bound for 3-pass streaming algorithm for st-reachability.

Chapter 2

Near-Quadratic lower Bound for 2-pass

In this part we will look at the lower bound of required space given in [AR20] for 2-pass streaming algorithm on the problem of st-reachability. First, we will look at a communication game such that it's communication complexity lower bounds

2.1 Convert Streaming algorithm to communication game

In the paper [AR20], they present a reduction from a 2-pass streaming algorithm to a 2-player communication game whose communication complexity is a lower bound for the space complexity of the streaming algorithm.

Proposition 2.1.1. *[AR20] Any two-pass S-space streaming algorithm A on graphs $G = (V, E_1 \cup E_2 \cup E_3)$ of st-reachability can be simulated exactly by a communication protocol π_A with $CC(\pi_A) = O(S)$. In the communication game E_1 is provided to Alice and E_2 is provided to Bob. After one round of communication between Alice and Bob, E_3 will be revealed to both of them.*

The idea is Alice will run A on E_1 and send the memory at the end to Bob who uses this to run A on E_2. Bob then sends the memory at the end to Alice and E_3 is revealed to both. Alice first runs A on E_3, completing first pass and then runs on E_1, and sends the memory to Bob. Bob runs A on E_2 and E_3 and outputs the answer. Thus, the memory will be atleast the communication cost of the protocol.

$$A(E_1) \xrightarrow[M11]{} B(E_2) \xrightarrow[M13]{} A(E_3, E_1) \xrightarrow[M21]{} B(E_2, E_3)$$

We will now show a near-quadratic lower bound on the communication complexity of this game, which will give us a near-quadratic lower bound for the space required by 2-pass streaming algorithm.For this we start by seeing a lower bound on Information cost for the set intersection problem, which will help us obtain a lower bound for communication complexity of Unique Reach problem, which gives us a distribution that is difficult for $st-$reachability.

2.2 Lower Bound for the Set Intersection Problem

The set intersection problem is the following:

Problem 1. It is a 2-player communication game where 2 sets A and B are given to Alice and Bob respectively having elements from $[n]$ and the promise is that there is a **unique** element in the intersection of A and B, $e^* = A \cap B$. The goal is to find the unique element e^* in their intersection.

Definition 2. [AR20] Let $\mathcal{D}$ be a distribution of inputs (A, B) for set-intersection problem. A protocol π solves set-intersection $\mathcal{D}$ iff at least one of the following hold

$$\mathbb{E}_{\Pi,A}||dist(e^*|\Pi, A) - dist(e^*|A)||_{tvd} \geq \epsilon$$
$$\mathbb{E}_{\Pi,B}||dist(e^*|\Pi, B) - dist(e^*|B)||_{tvd} \geq \epsilon$$

Π denote the messages and public randomness of protocol π. $\mathbb{E}$ denotes expected value, and $dist()$ represents the distribution over $\mathcal{D}$ and private randomness of π.

In the paper they give a lower bound on the information cost of any protocol that ϵ solves the set intersection problem.

Theorem 2.2.1. *[AR20] There is a distribution D_{SI} for set-intersection over the universe $[m]$ such that:*

for any $\epsilon \in (0, 1)$, any protocol π that internal ϵ-solves the set-intersection problem over the distribution D_{SI} has internal information cost $IC_{D_{SI}}(\pi) = \Omega(\epsilon^2 m)$

They give the following distribution which already has known $\Omega(m)$ lower bound for being solved for probability $> 2/3$, which they extend to ϵ-solving by giving a reduction to ϵ-solving.

$\mathbb{D}_{\mathbb{SI}}$ An input distribution (A, B) to set-intersection over the universe $[m]$.

- Sample two disjoint sets A' and B' of size $m/4 - 1$ each uniformly at random from $[m]$
- Sample $e^* \in [m] \setminus (A' \cup B')$ uniformly at random and let $A' := A' \cup \{e^*\}$ and $B' := B' \cup \{e^*\}$

This distribution which is hard for set-intersection problem is used to construct a hard distribution for unique reach problem.

2.3 Unique Reach Problem

The unique reach is a 2-player communication game with a source set and a target set with the promise that there is a unique vertex that is reachable in the target set. In more detail, we have a digraph $G = (V, E)$ on n vertices with $V = \{s\} \cup V_1 \cup V_2 \cup V_3$ and edges between s to V_1 and V_i to V_{i+1}. The promise is that there is a unique vertex s* in V_3 reachable from s.

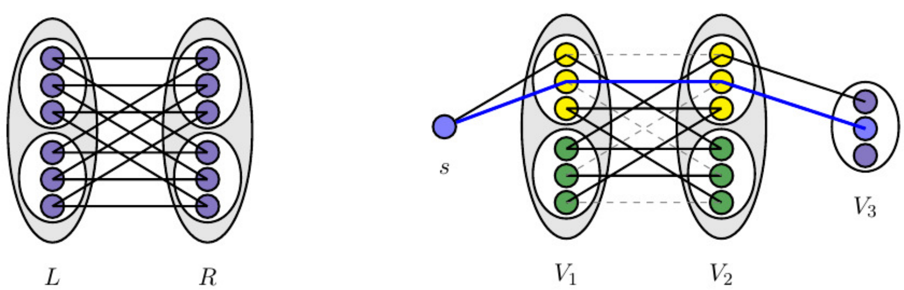
The communication game is as follows. The edges given to Alice, E_A are between V_1 and V_2 and the rest of the edges are given to Bob (E_B). The goal is to find s* using a single communication from Alice to Bob(one way communication model).

2.4 Hard Distribution for Unique Reach

2.4.1 Ruzsa-Szemeredi Graphs

A graph $G = (V, E)$ is a called an (r, t)-Ruzsa-Szemeredi graph (RS graph for short) iff its edgeset E can be partitioned into t induced matchings $M_1^{RS} \dots M_t^{RS}$ each of size r. We can also form an RS digraph by using a bipartite RS graph $G = (L, R, E)$ and directing the edges of G from L to R.

Given M_i^{RS} having edges $e_{i,1} \dots e_{i,r}$ with $e_{i,j} = (u_{i,j}, v_{i,j})$ and let $S \subseteq [r]$, we define $M_i^{RS}|S$ to be the matching in G consisting of the edges $e_{i,j}$ for $j \in S$.



(a) A fixed $(3, 4)$-RS digraph in the distribution.

(b) The graph G of $\mathcal{D}_{\mathsf{UR}}$; dashed edges no longer belong to the graph, and yellow vertices are incident on $M_{i^\star}^{\mathsf{RS}}$.

Figure 1: An illustration of the input distribution $\mathcal{D}_{\mathsf{UR}}$. Here, directions of all edges are from left to right and hence omitted. The marked vertex (blue) in V_3 denotes the unique vertex $s^\star$ in this example along with the path connecting s to $s^\star$.

Figure 2.1: Diagram From the paper [AR20]

Proposition 2.4.1. *[AR20] For infinitely many integers N, there are (r, t)-RS digraphs with N vertices on each side of the bipartition and parameters $r = \frac{N}{2^{\theta(\sqrt{\log N})}}$ and t=N/3.*

2.4.2 Distribution D_{UR}

The distribution is on graphs $G = (\{s\} \cup V_1 \cup V_2 \cup V_3, E_A \cup E_B)$

- We take an (r, t) RS digraph $G = (L, R, E)$ on 2N vertices with induced matchings $M_1^{RS} \dots M_t^{RS}$ with parameters $r = \frac{N}{2^{\theta(\sqrt{\log N})}}$ and t=N/3. This exists due to the above proposition.
- We set V_1=L=$\{u_1 \dots u_N\}$ and V_2=R=$\{v_1 \dots v_N\}$ and V_3 is set to be equal to r new vertices $\{w_1 \dots w_r\}$.
- We sample t independent instances of the set-intersection problem $(S_1, T_1) \dots (S_t, T_t)$ on the universe [r] .
- $E_A = (M_1^{RS}|S_1) \cup (M_2^{RS}|S_2) \dots (M_t^{RS}|S_t)$ is the input to Alice
- We also sample i* $\in [t]$ uniformly at random.
- E_B is the set of edges $(s, u_{i*,j})$ for $j \in T_{i*}$ and $(v_{i*,j}, w_j)$ for $j \in T_{i*}$

2.5 Inverse Unique reach

It is exactly the same as the Unique reach problem with the only difference that the direction of the edges are reversed in this.

2.6 Lower bound for Direct Reachability Communication game

For the distribution defined above and $b = \frac{n}{2^{\theta(\sqrt{\log n})}}$, for any $\epsilon \in (0, 1)$, any protocol π that internal ϵ-solves the unique reach problem over the distribution D_{UR} has communication cost $CC(\pi) = \Omega(\epsilon^2 nb)$ (so almost quadratic).

The main idea for this lower bound is that internally solving unique reach can be reduced to internally solving the set intersection problem for the instance (S_{i*}, T_{i*}) and as we have t instances of the sets S_i which define the edges E_A, the information revealed by π_{UR} about the instance (S_{i*}, T_{i*}) is at least t times smaller than CC(π_{UR}).

The protocol for set intersection π_{SI} using π_{UR} is given:

Protocol $\pi_{\mathbf{SI}}$ Consider the following protocol π_{SI} for instance (A, B) of Set intersection between the two players $Alice_{UR}$ and Bob_{UR}.

- *Alice* and *Bob* sample i* $\in [t]$ using public randomness.
- *Alice* sets A=S_{i*} and samples the remaining $S_i's$ privately which are required for the edge set E_A in D_{UR}
- *Bob* sets B=T_{i*} which is enough to get the edge set E_B in D_{UR}

- Since we have π_{UR} as a subroutine, we can run it on (E_A, E_B) to get the answer to the set intersection problem.

This can be used to show that our protocol π_{SI} can internal ϵ-solve set intersection problem on D_{SI} and that the communication complexity of this protocol is atleast t times the information cost of the set intersection problem which gives us the (near) quadratic lower bound.

2.7 Lower bound for s-t reachability

To make a hard distribution for the st-reachability problem we will combine two independent instances from the unique reach distribution to each other, where the source of second unique reach will act as the target for our st-reachability. The target set(V_3's) in the two unique-reach instances will be connected to each other such that in the distribution vertex between any 2 edges will be present in half of the graphs.

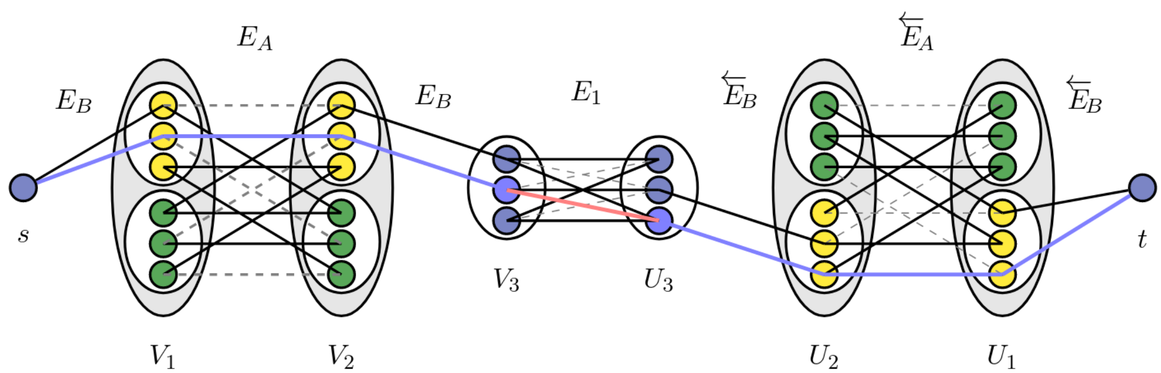


Figure 2: An illustration of the input distribution $\mathcal{D}_{\mathsf{ST}}$. Here, the directions of all edges are from left to right and hence omitted. The vertices $s^\star \in V_3$ and $t^\star \in U_3$ are marked blue and the potential edge $(s^\star, t^\star)$ is marked red–existence or non-existence of this edge uniquely determines whether or not s can reach t in G.

Figure 2.2: Diagram From the paper[AR20]

We give a formal definition of the Distribution as follows:

$\mathbf{D_{ST}}$ A hard input distribution for the st-reachability problem with rules constructed in communication game in section 1.1.

- $V = \{s\} \cup V_1 \cup V_2 \cup V_3 \cup U_3 \cup U_2 \cup U_1 \cup \{t\}$, each set is a layer of graph
- Sample the edges E_1, picking each edge $(u, v) \in V_3 \times U_3$ independently with probability 0.5.

- Sample 2 instances as follows
    1. $H = (\{s\} \cup V_1 \cup V_2 \cup V_3, E_A \cup E_B)$ from D_{UR}
    2. $H = (U_3 \cup U_2 \cup U_1 \cup \{t\}, \overleftarrow{E_A} \cup \overleftarrow{E_B})$ from $\overleftarrow{D}_{UR}$(reversed unique reach)
- The initial input to Alice is E_1, and to Bob is $E_2 = E_A \cup \overleftarrow{E_A}$. The input revealed to both players int the second round is $E_3 = E_B \cup \overleftarrow{E_B}$

In this distribution, a path between s, t exists if there is an edge between s^* and t^*.

Why this distribution works is because for the first communication, if Alice is allowed less than $O(n^2)$ bits, she cannot communicate anything useful about the existence of the edge between s^*, t^* in E_1 as s^* and t^* aren't known to her at that time.

For the communication between Bob and Alice, we use the lower bound from unique-reach problem and where Bob in st problem is Alice of Unique-Reach and Alice of st-problem is Bob of Unique-Reach. From the lower bound we know that with less than near quadratic bit communication, we cannot figure out anything about s^* and t^*. This can be seen through the following reduction where using a protocol that gives *epsilon* advantage in D_{ST}, we can solve D_{UR}

Protocol $\pi_{\mathbf{UR}}$ Consider the following one-way protocol π_{UR} for instances $H := (\{s\} \cup V_1 \cup V_2 \cup V_3, E_A \cup E_B)$ of unique-reach sampled from D_{UR} between the two players $Alice_{UR}$ and Bob_{UR}.

- $Alice_{UR}$ and Bob_{UR} sample $\overleftarrow{E_B} \sim D_{ST}|H$ using public randomness.
- $Alice_{UR}$ samples $\overleftarrow{E_A}) \sim D_{ST}|H, \overleftarrow{E_B})$ and E_1 $D_{ST}|H, \overleftarrow{E_A}), \overleftarrow{E_B})$ using private randomness. These can be done because the set E_1 and the reverse unique-reach graph is independent of out the forward unique-reach graph.
- $Alice_{UR}$ has access to the first-round input of $Alice_{ST}$ and Bob_{ST}, and thus can run π_{ST} for first 2 communication round and send the communication of Bob_{ST} to $Alice_{ST}$ to Bob_{UR}.
- Since Bob_{UR} cannot ϵ-solve the problem without using near quadratic communication bits from previous part. This means $Alice_{ST}$ will not be able to ϵ-solve for s^*.

Similiar arguments on reverse graph, tells that *Alice* cannot figure out t^* also after the first round of communication.

Finally, for last communication, Alice needs to send the information about the edge s^*, t^*, which decides if path exists. Bob at last using E_2 and E_3 can figure out s^*, t^* but he needs to know if there is an edge between them in E_1 which is not known to them. This happens because the edges of E_1 are independent of E_3, so knowing E_3 doesn't help. Also using E_3 only and message from Bob_{ST}, s^*, t^* cannot be figured out with good advantage, otherwise,

Bob_{UR} could solve the problem. So Alice to has send all the edges in E_1 to Bob as s^*, t^* aren't known to her because of the lower bound in unique-reach, which will require near n^2 bits.

At the end, Bob cannot know in any protocol if edge (s^*, t^*) exists that uses less than $n^{2-\Omega(1)}$ communication bits. Since, the edge (s^*, t^*) exists with probability $1/2$, any protocol answering can have a very small advantage over $1/2$. Thus, we have the $n^{2-\Omega(1)}$ lower bound on communication complexity of the game.

Theorem 2.7.1. *[AR20] For any $\epsilon \in (n^{-1/2}, 1/2)$ any communication protocol for $st-$ reachability that succeeds with probability at least $\frac{1}{2} + \epsilon$ requires $\Omega(\epsilon^2 \cdot \frac{n^2}{\Theta(\log(n))}$*

This translates to $n^{2-\Omega(1)}$ lower bound on space used by 2-pass streaming algorithm from section 2.1.

Theorem 2.7.2. *[AR20] Any streaming algorithm that makes two passes over the edges of any n-vertex directed graph $G = (V, E)$ with two designated vertices $s, t \in V$ and outputs whether or not s can reach t in G with probability at least $2/3$ requires $\Omega(\frac{n^2}{2^{\theta(\sqrt{\log n})}})$space.*

2.8 Extensions to Other problems

The lower bound for directed *st*-reachability can be extended to other problems as well. In the paper [AR20] they use well known reductions to get near quadratic space lower bounds for 2-pass algorithms for the problems of Bipartite Perfect Matching and Single-Source Shortest path in undirected graph as well.

2.8.1 Bipartite Perfect Matching

Given $H = (E_H, V_H)$, we create $G = (L \cup R, E)$ such that G has perfect matching iff H has path from s to t

- For $v \in V_H/\{s, t\}$ add 2 vertices $v^l \in L$ $v^r \in R$. Add $s^l \in L$ and $t^r \in R$
- For edge $e = (u, v) \in E_H$ add undirected edge $(u^l, v^r) \in E$. Also add $(v^l, v^r) \in E$

A path between s and t in H will define perfect matching in G, every edge (u, v) corresponds to u^l, v^r being matched.This will begin at s^l and end at t^r. The vertices not on path will have v^l, v^r matching. With same idea any matching in G corresponds to path in H.

This can be done easily in a stream, hence the near quadratic lower bound translates to same lower bound for Bipartite perfect matching.

2.8.2 Single Source Shortest path

We just replace in graphs from D_{ST} the directed edges with undirected edges. Now if there is an edge between s^* and t^*, that is there is a path from s to t, then we can see that the shortest path from s to t is of length 7, one edge going from layer to layer. If there is no path, then definitely, you have to go back atleast one layer, and that increases the path length by 2. Therefore, if there is path in graph $G \in D_{ST}$, then shortest path is 7, otherwise the shortest path is atleast 9. Thus, using same idea as st-reachability we get a near quadratic lower bound for 2-pass streaming algorithm solving shortest path in undirected graphs.

Chapter 3

Possible extensions

Our main focus is to extend this lower bound for 2-pass streaming algorithms to 3-pass streaming algorithms. The reason the above approach naively fails is because in the unique-reach problem because at end of second round, Bob knows s^*, t^* and can send this info to Alice who can check if there is an edge between them or not.

3.1 Unique-reach to Constant-Reach

For converting unique reach to constant reach (say 2 reach), the idea is that in the distribution D_{UR}, we sample i_1 and i_2 instead of just i* and then the edges given to B are a union of the edges corresponding to i_1 and i_2. In unique reach the only vertex reachable from s was the vertex w_{e*} where e* is the unique element in the intersection of (S_{i*}, T_{i*}). Similarly, here there would be 2 elements that are reachable from s, corresponding to i_1 and i_2. The reason that other vertices in V_3 are not reachable from s is because of the fact that the graph that we have is an RS graph in which the only elements that can be reachable are in the intersection of the corresponding instances of Set-intersection problem (there cant be any more elements as the RS graph consists of induced matchings which does not allow path corresponding to a combination of both i_1 and i_2)

3.2 Combining Graphs

We know that 2-pass algorithms cannot find if there is a path in the distribution of D_{ST}, our initial idea was to replace the edge between s^*, t^* with a graph of D_{ST}, such that all vertices in U_3, V_3 act as s, t. As we know finding the path between s^*, t^* is hard in 2 pass and finding s^*, t^* is hard in one pass, intuitively it should be hard for 3-pass.

But first we needed a reduction of the streaming algorithm to a communication game. For the 3-pass streaming algorithm, we propose a new communication game.

3.2.1 3 Player Communication Game

This is inspired by the game in [ACK19]. We will have 3 players A, B, C and the edge stream will be a union of 4 disjoint sets as $\sigma = E = E_1 \cup E_2 \cup E_3 \cup E_4$. E_1, E_2, E_3 will be given to A, B, C respectively at the start of communication game. E_4 will be revealed to all after first round of communication. The streaming algorithm will work as follows

$$A(E_1) \xrightarrow[M11]{} B(E_2) \xrightarrow[M12]{} C(E_3) \xrightarrow[M13]{} A(E_4, E_1) \xrightarrow[M21]{} B(E_2) \xrightarrow[M22]{} C(E_3, E_4) \xrightarrow[M30]{} A(E_1) \xrightarrow[M31]{} B(E_2) \xrightarrow[M32]{} C(E_3, E_4)$$

In this communication game, we have 4 sets of edges and 3 players instead of the 3 sets of edges and 2 players in the 2-pass game. Similar to 2 pass we are able to hide a set of edges in one-round of communication. But, there are 2 rounds where all the information is revealed.

3.2.2 Insert D_{ST} between s^* and t^*

The idea was to modify the hard distribution for the st reachability problem. In that, we have a unique vertex s* in v_3 reachable from s and a similar thing going on for unique t* in U_3 that is connected to t. We tried to increase the complexity(hardness) of our distribution by adding another instance of D_{ST} between s^* and t^*. The idea behind this was that since D_{ST} is difficult for 2 passes, as we are only able to get information about s* and t* atleast after one round gets complete, this distribution could be difficult for the 3 player game for 3 passes. For this modification, as we can't make any distiction between s* and the other vertices in V_3, we considered all the vertices in V_3 to represent a new source s and a similar thing happens for U_3. This modification again ensures that the s-t reachability again depends on only the 1 distinguishing edge of D_{ST} which again appears with probability 0.5 (again only a single path can exist between s and t).
The distribution of edges was similar to that in the s-t reachability problem. Without loss of generality, we can assume that we gave edges inside the D_{ST} instance to B and C respectively and the E_A edges outside the D_{ST} instance to A and the remaining edges E_4 become public after the first round of communication. The problem that occurs in this case is that after the public edges are revealed after the first round, C can use them as well as its own edges to find the unique element in the intersection of its instance of set-intersection (S, T) using which it can find $s*$ and $t*$ inside the D_{ST} instance and communicate it using logarithmic space to B which can determine whether the distinguishing edge is present or not. So as this information can be computed and easily communicated, we dont need information about all the edges between V_3 and U_3(like in st reachability) so we dont have a quadratic lower bound here. Another thing to note is that the same thing will happen for any number of recursively added instances of D_{ST} so our approach would fail at the same point as here.

The idea to prevent C from getting information about the intersection could be to give the edges adjacent to E_A to some other player (instead of public as was done before). But the problem is that the number of such edges are $O(n)$ which can be easily communicated to any other player which does not give us any advantage over making them public. We hope to try to increase the number of such edges from$O(n)$ to something near quadratic using some modification in the graph and then apply this approach to get an appropriate lower bound.

3.2.3 Join mirror image of D_{ST}

Another idea that we thought of working with was to replace the vertex t with a set T_1 and connect a mirror image of this modified graph from $\overleftarrow{D_{ST}}$ by adding 0.5 probability edges between T_1 and T_1'. The connection between V_1 and T_1 will be similar to the one between U_2 and U_3, that is based on the set sampled from D_{SI} for RS graph of V_1, V_2. Thus, we will have 4 sets of edges E_1 will be edges between T_1, T_1', E_2 will be edges between U_3, V_3 and U_3', V_3', E_3 will be edges in the RS-graphs and E_4 will be rest of the edges.

In this the communication game between Alice and Bob of *st*-reachability will work with B, C as they try to figure out the vertex t_1 in T_1 and t_1' in T_1'. They need to get this info to A before his last action, otherwise he will have to send whole E_1 making communication complexity n^2. We thought B, C should not be able to figure out t as their game is almost same(communication of C to B happens through A, which doesn't change anything), and they weren't able to figure out if there is a path to t or not.

But similar to previous approach, once E_4 is revealed at start of second communication round, it can be communicated to the rest of the players easily as it is only $O(n)$, and it can find the intersection of the elements in the 2 sets. If E_4 is known to C, then using E_3, he can figure out t_1, t_1', and communicate them to A, who can communicate if there is an edge between them or not.

It seems from these constructions that the problem of Unique reach on which we try to build the hard cases for 3-pass isn't hard enough, as if Bob_{UR} is allowed to send messages will be easily able to solve the problem since his information is $O(n)$. As we cannot hide some information after the first round of communication, we must try to construct a case which is hard even when all the information is available.

Chapter 4

Future Work

- We can try to look for a hard distribution for the 3 pass 3 player streaming algorithm. We know that if we can find a difficult distribution for reachabiity between s and t for 2 passes in which every edge is distributed between 3 players, then it would correspond to a same lower bound for our 3 payer 3 pass problem. So we can try to work on it.

- The unique reach problem that they use is not difficult for 2 passes that is why it is difficult to directly prove difficulty for more than 2 passes. We could try to look for a more difficult "intermediate" problem to prove difficulty for more number of passes

- We also were thinking of changing the difficult distribution for the s-t reachability problem to one in which we use a different distribution than the RS graphs which could lead to a better lower bound for 3 passes.

- Almost all our approaches failed because of the reason that in the D_{ST},$O(n)$ edges are public and even making them private does not give any advantage. We can try to create a different distribution that satisfies the conditions that atmost 1 path can exist between s and t but is more connected (near quadratic) so that if we make them private, they cant be communicated which can help us in getting a lower bound.

Bibliography

[ACK19] Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Polynomial Pass Lower Bounds for Graph Streaming Algorithms. *arXiv e-prints*, page arXiv:1904.04720, April 2019.

[AR20] Sepehr Assadi and Ran Raz. Near-Quadratic Lower Bounds for Two-Pass Graph Streaming Algorithms. *arXiv e-prints*, page arXiv:2009.01161, September 2020.