A Simple Semi-Streaming Algorithm for Global Minimum Cuts

Sepehr Assadi
Rutgers University

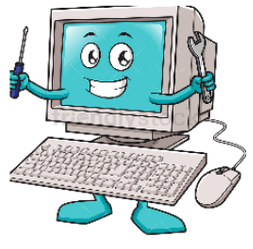
Aditi Dudeja
Rutgers University

Semi-streaming Model

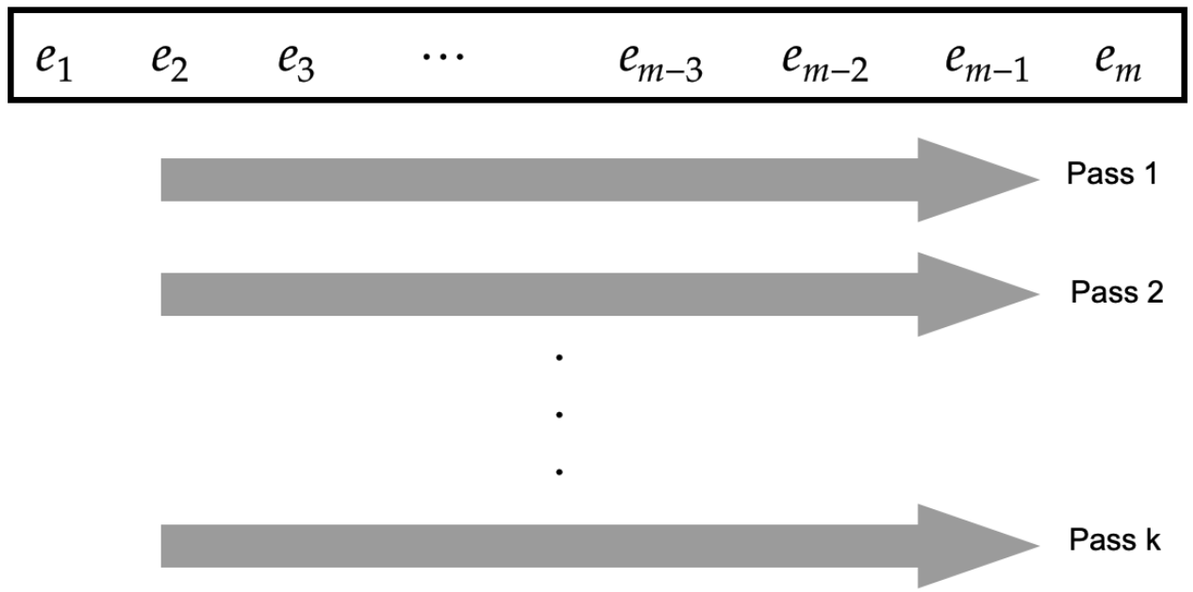
- **Limited Memory:** Memory of the algorithm is smaller than the input $O(n \cdot \text{poly}(\log n))$.
- **Sequential access:** No random access to the input.

Graph Streaming

- **Graph** $G = (V, E)$:



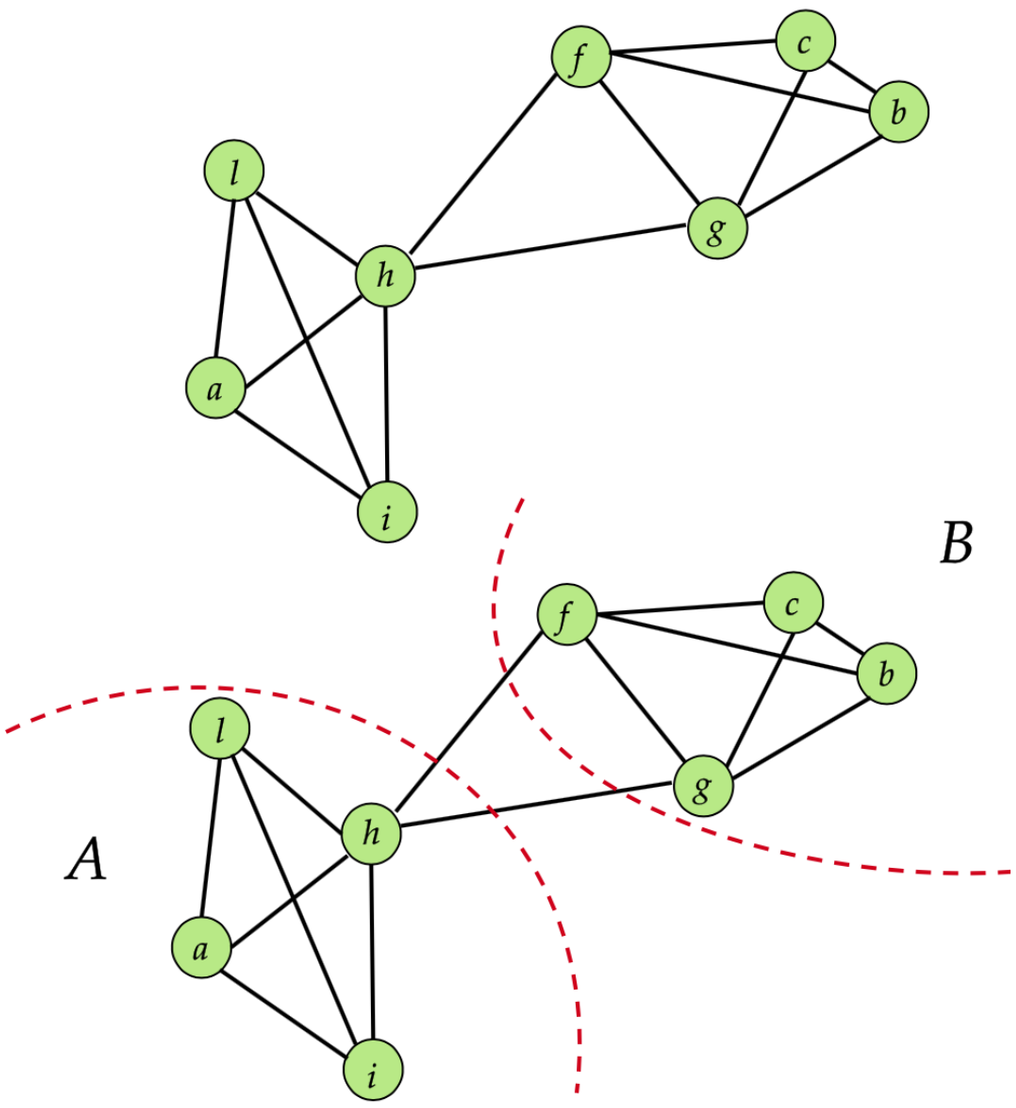
  - Known vertices:
    $V = \{1,2,3,\cdots,n\}$.



  - Unknown edges:
    $E = \langle e_1, e_2, \cdots, e_m \rangle$.

Global Minimum Cut

- Graph $G = (V, E)$:
  - Partition of V into two subsets (A, B) such that the number of edges between A and B is minimized.



Known Results

- An $\Omega(n^2)$ lower bound on space for any one-pass semi-streaming algorithm due to Zelke11.

- An $O(n \cdot \text{poly}(\log n))$ space two-pass streaming algorithm due to RSW18. Observed by ACK19.

- One pass semi-streaming algorithm for $(1+\epsilon)-$approximation.

Our Results

- A simplified two-pass streaming algorithm that outputs a minimum cut of the graph in $O(n \log n)$ space.

- Any streaming algorithm that computes the minimum cut value of an n-vertex graph in constant $p > 1$ passes needs $\Omega(n \cdot (\log n)^{1/2p-1})$ space.

Edge-Out Contractions

- Given graph $G = (V, E)$, for every vertex $v \in V$, sample two edges incident on v independently and with repetition.

- Let $G^{(2)}$ be the graph consisting of only these edges, and let H be the graph obtained by contracting the connected components of $G^{(2)}$. Think of $G^{(2)}$ as a directed graph.

- Then, number of components in $G^{(2)}$ are at most $O\left(\frac{n}{d_{\min}}\right)$.

- Let $C = \{e_1, e_2, \cdots, e_\lambda\}$ be a fixed minimum cut of G. No edge of C is contracted in H with a constant probability.

Edge-Out Contractions

- Given graph $G = (V, E)$, for every vertex $v \in V$, sample two edges incident on v independently and with repetition.

- Let $G^{(2)}$ be the graph consisting of onl
  let H be the graph obtained by cont
  components of $G^{(2)}$. Think of $G^{(2)}$ as

Due to GNT20.

- Then, number of components in $G^{(2)}$ are at most $O\left(\frac{n}{d_{\min}}\right)$.

- Let $C = \{e_1, e_2, \cdots, e_\lambda\}$ be a fixed minimum cut of G. No edge of C is contracted in H with a constant probability.

Edge-Out Contractions

- Given graph $G = (V, E)$, for every vertex $v \in V$, sample two edges incident on v independently and with repetition.

- Let $G^{(2)}$ be the graph consisting of onl
  let H be the graph obtained by cont
  components of $G^{(2)}$.

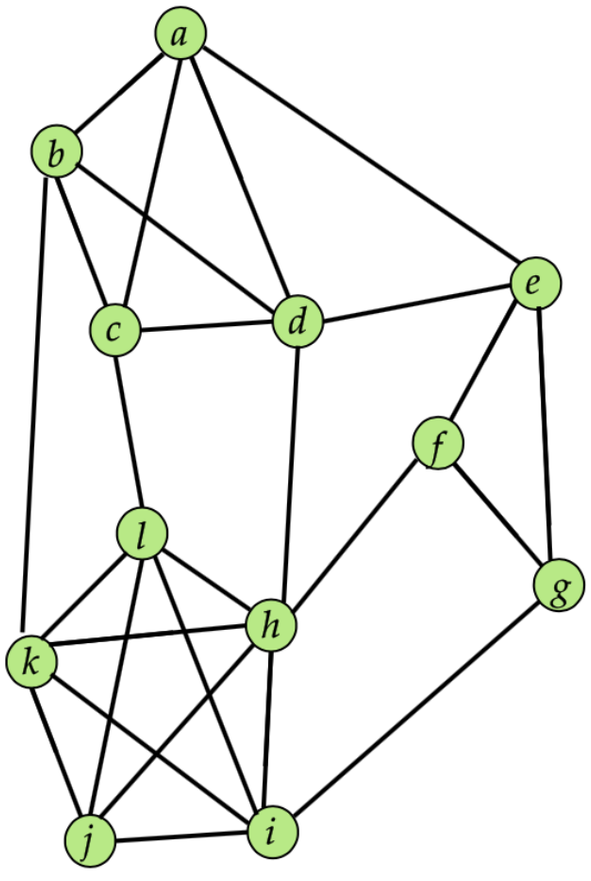
- Then, number of components in $G^{(2)}$ are at most $O\left(\frac{n}{}\right)$.

- Let $C = \{e_1, e_2, \cdots, e_\lambda\}$ be a fixed minimu
  of C is contracted in H with a consta

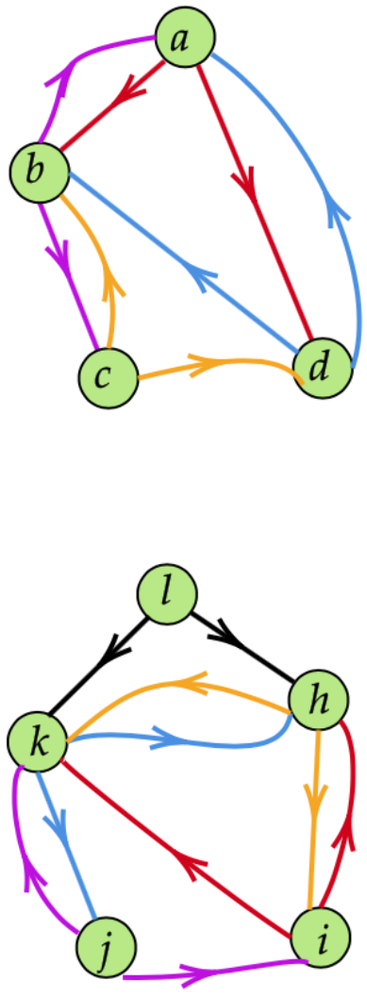
Due to GNT20.

Allows us to get a sparse graph that preserves the minimum cut.

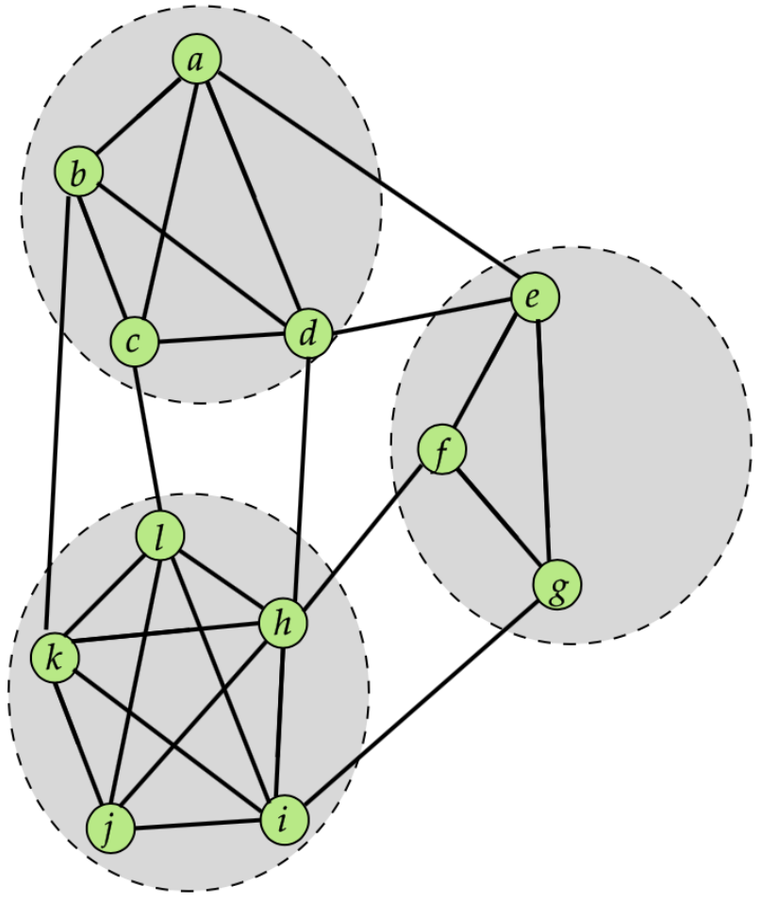
Edge-Out Contractions



a) Graph G



b) Graph $G^{(2)}$



c) Graph H

Algorithm: First Pass

- Essentially, implementing the edge-out contraction in streaming setting:
  - For each $v \in V$ sample two edges incident on it independently and with repetition, and store in memory. Let $G^{(2)}$ be the resulting graph and $V_1, V_2, \cdots, V_t$ be the connected components.
  - Compute $d_{\min}$ in parallel, and if $t > \frac{100 \cdot n}{d_{\min}}$ then abort.

Algorithm: Second Pass

- Let H be the graph obtained by contracting vertices in each V_i into a single component.

- Let $F_1, F_2, \cdots, F_{d_{\min}}$, be initially empty. For each $e \in H$, include e in F_i where i is the smallest index such that $F_i \cup \{e\}$ does not contain a cycle with respect to H. Compute min cut of $F_1 \cup F_2 \cup \cdots \cup F_{d_{\min}}$.

Space Analysis

- First pass: We store only two edges per vertex. This takes space $O(n \log n)$.
- Second pass: Note that $t = O\left(\frac{n}{d_{\min}}\right)$ with constant probability. In each F_i we store at most $t - 1$ edges and the number of F_i's are $d_{\min}$ many. So overall $O(n \log n)$ space needed.

Correctness

- Correctness follows from the following observations:
  - Let $C = \{e_1, e_2, \cdots, e_\lambda\}$ be a fixed minimum cut of G. No edge of C is contracted in H with a constant probability.
  - Any cut of H is a cut of G as well.
  - Finally, if the minimum cut of H has at least k edges then every cut of $F_1 \cup F_2 \cup \cdots \cup F_{d_{\min}}$ has at least $\min\{d_{\min}, k\}$ edges.

Correctness

- Correctness follows from the following

  - Let $C = \{e_1, e_2, \cdots, e_\lambda\}$ be a fixed minimum c[illegible] edge of C is contracted in H with a constant probability.

  - Any cut of H is a cut of G as well.

  - Finally, if the minimum cut of H has at least k edges then every cut of $F_1 \cup F_2 \cup \cdots \cup F_{d_{\min}}$ has at least $\min\{d_{\min}, k\}$ edges.

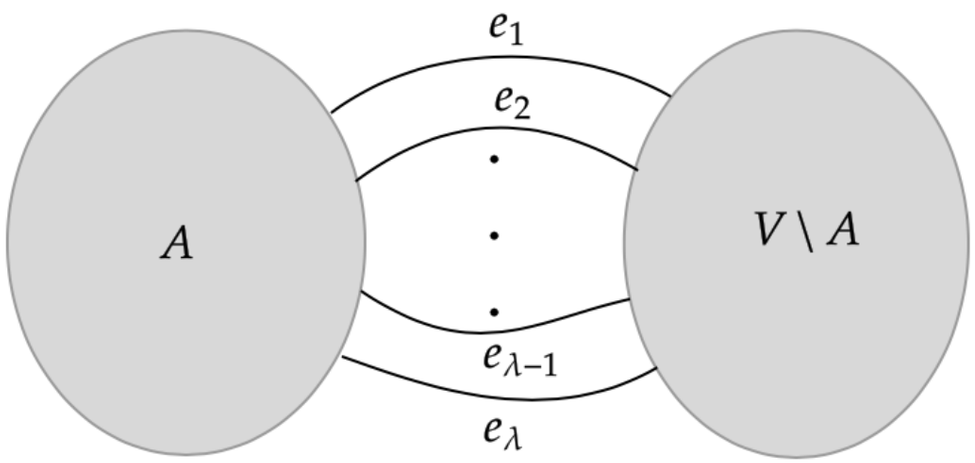
Due to GNT20.

Simplified Proof of GNT20

Recall…

- The number of components in $G^{(2)}$ are at most $O\left(\frac{n}{d_{\min}}\right)$.

- Let $C = \{e_1, e_2, \cdots, e_\lambda\}$ be a fixed minimum cut of G. No edge of C is contracted in H with a constant probability.



Preserving min-cut

- Let $c(v)$ be the set of edges incident on v participating in C.

$$\textbf{Pr}\left(v \textbf{ doesn't sample } c(v)\right) = \left(1 - \frac{c(v)}{deg(v)}\right)^2$$

- **Fact:** $c(v) \leq deg(v)/2.$

Let $N(C)$ be the vertices that are endpoints of C,

$$\textbf{Pr}\left(e_1, e_2, \cdots, e_\lambda \textbf{ are not sampled.}\right) = \prod_{v \in N(C)} \left(1 - \frac{c(v)}{deg(v)}\right)^2$$

$$\geq \textbf{exp}\left(-\frac{4}{\lambda} \sum_{v \in N(C)} c(v)\right) = e^{-8}$$

Bounding number of components

- We want to upper bound the probability that the number of components in $G^{(2)}$ are at least $\frac{100 \cdot n}{d_{\min}}$.

$$\Downarrow$$

- At least $k = \frac{50 \cdot n}{d_{\min}}$ components of size at most $\frac{d_{\min}}{50}$.

Bounding number of components

- We want to upper bound the probability that the number of components in $G^{(2)}$ are at least $\frac{100 \cdot n}{d_{\min}}$.

$\Downarrow$

- At least $k = \frac{50 \cdot n}{d_{\min}}$ components of size at most $\frac{d_{\min}}{50}$.

For each of these components, doing a BFS from one of the vertices, we reach at most $d_{\min}/50$ vertices.

Recall: $G^{(2)}$ is directed.

Bounding number of components

- Let s_i be the size of the ith component. Let $s = s_1 + s_2 + \cdots + s_k$.

- We have $s \geq \dfrac{50 \cdot n}{d_{\min}}$, **and** $s_i \leq \dfrac{d_{\min}}{50}$.

Bounding number of components

- Each component can be associated with a “root” vertex, v_i and an unlabelled BFS tree.

- In turn, for the ith component, the BFS tree can be associated with a vector, $(x_1, x_2, \cdots, x_{s_i})$, where $0 \leq x_j \leq 2$, and $\sum x_j = s_i - 1$.

Bounding number of components

Total number of possible BFS trees for a fixed s_i.

$\leq$

Total number of possible vectors for a fixed $s_i = 3^{s_i}$.

Probability of getting a component for a fixed choice of v_i and BFS tree: $\left(\frac{s_i}{d_{\min}}\right)^{s_i+1} \leq \left(\frac{1}{50}\right)^{s_i+1}$

Probability of getting components for a fixed choice of v_i's and BFS tree: $\left(\frac{3}{50}\right)^{s}$

Bounding number of components

Total number of choices of s_i's and v_i's for fixed s and k.

$$\binom{s+k}{k} \cdot \binom{n}{k}$$

Summing over all choices of s:

$$\sum_{s \geq \frac{50 \cdot n}{d_{\min}}} \binom{s+k}{k} \cdot \binom{n}{k} \cdot \left(\frac{3}{50}\right)^s \leq 10^{-5}$$

Space Lower Bound

- Theorem: Any streaming algorithm that computes the minimum cut value of an n-vertex graph in constant $p > 1$ passes needs $\Omega(n \cdot (\log n)^{1/2p-1})$ space.

Space Lower Bound

- Theorem: For any streaming algorithm that computes the min cut value of an n-vertex graph in constant $p > 1$ passes needs $\Omega(n \cdot (\log n)^{1/2p-1})$ space.

Proof by reducing a communication complexity problem to the finding min-cut.

Space Lower Bound

- **M-Fold-GreaterThan:** Alice and Bob are each given M separate N-bit number, $X = (x^1, x^2, \cdots, x^M)$ and $Y = (y^1, y^2, \cdots, y^M)$ respectively. Goal is to determine the value of

$$GT_N^M(X, Y) : \begin{cases} 1 & \textbf{if } \exists\ x_i, y_i \textbf{ such that } x_i < y_i \\ 0 & \textbf{otherwise} \end{cases}$$

Space Lower Bound

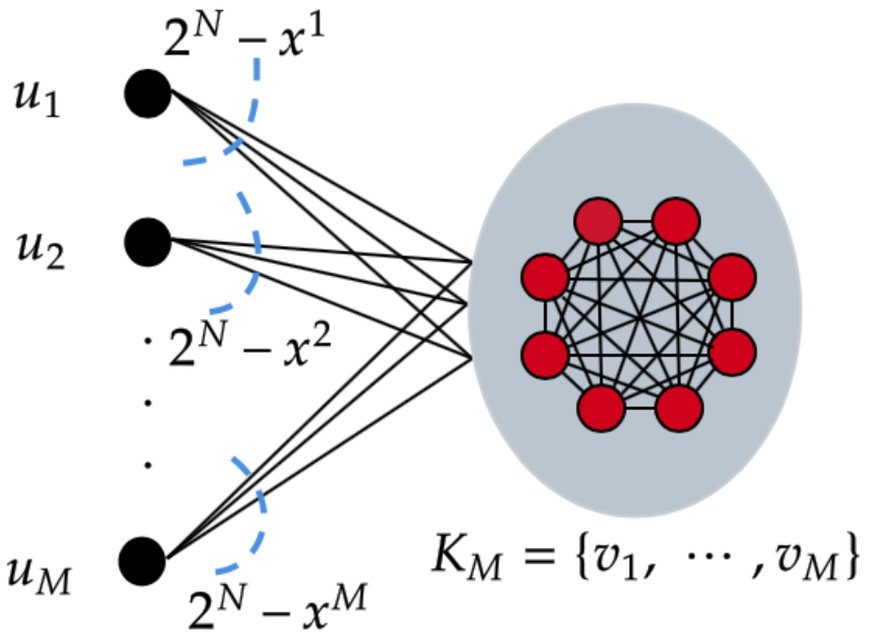
- **M-Fold-GreaterThan:** Alice and Bob are each given M separate N-bit number, $X = (x^1, x^2, \cdots, x^M)$ and $Y = (y^1, y^2, \cdots, y^M)$ respectively. Goal is to determine the value of

$$GT_N^M(X, Y) : \begin{cases} 1 & \textbf{if } \exists\ x_i, y_i \textbf{ such that } x_i < y_i \\ 0 & \textbf{otherwise} \end{cases}$$

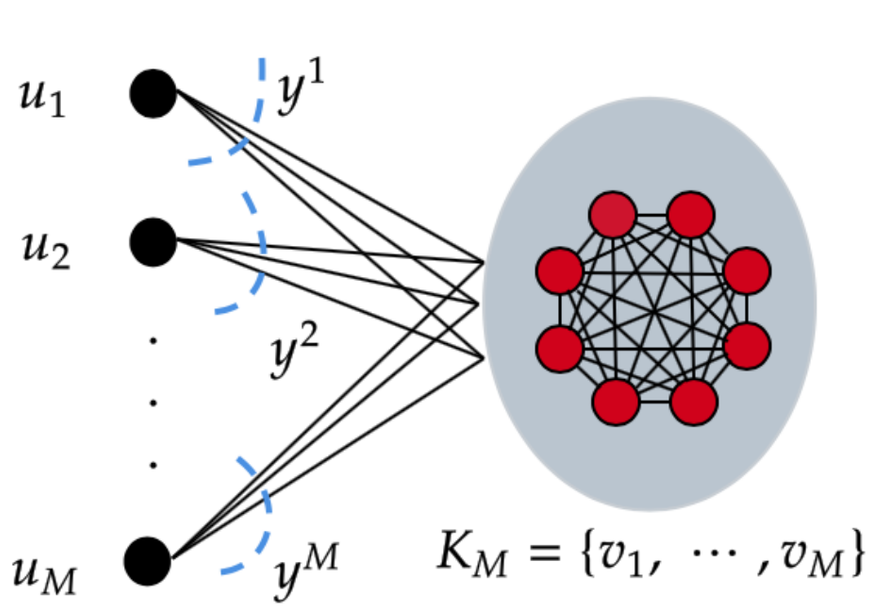
r – round communication lower bound is $\Omega_r(M \cdot N^{\frac{1}{r}})$. [JRS03]

Reduction

- Given instance $X = (x^1, x^2, \cdots, x^M)$ and $Y = (y^1, y^2, \cdots, y^M)$ of GT_N^M.

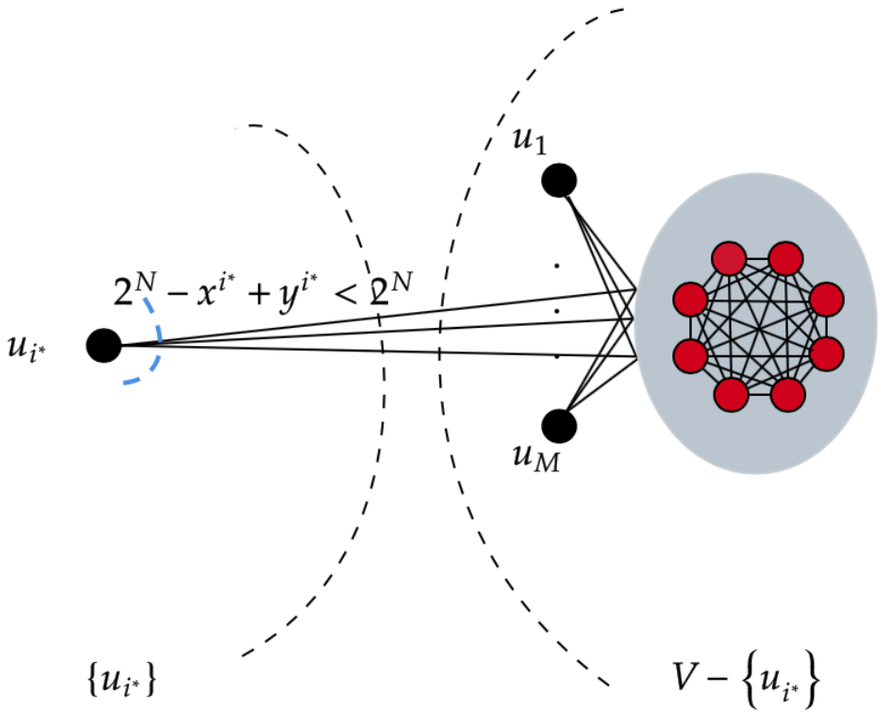


a) Edges E_A added by Alice.



b) Edges E_B added by Alice.

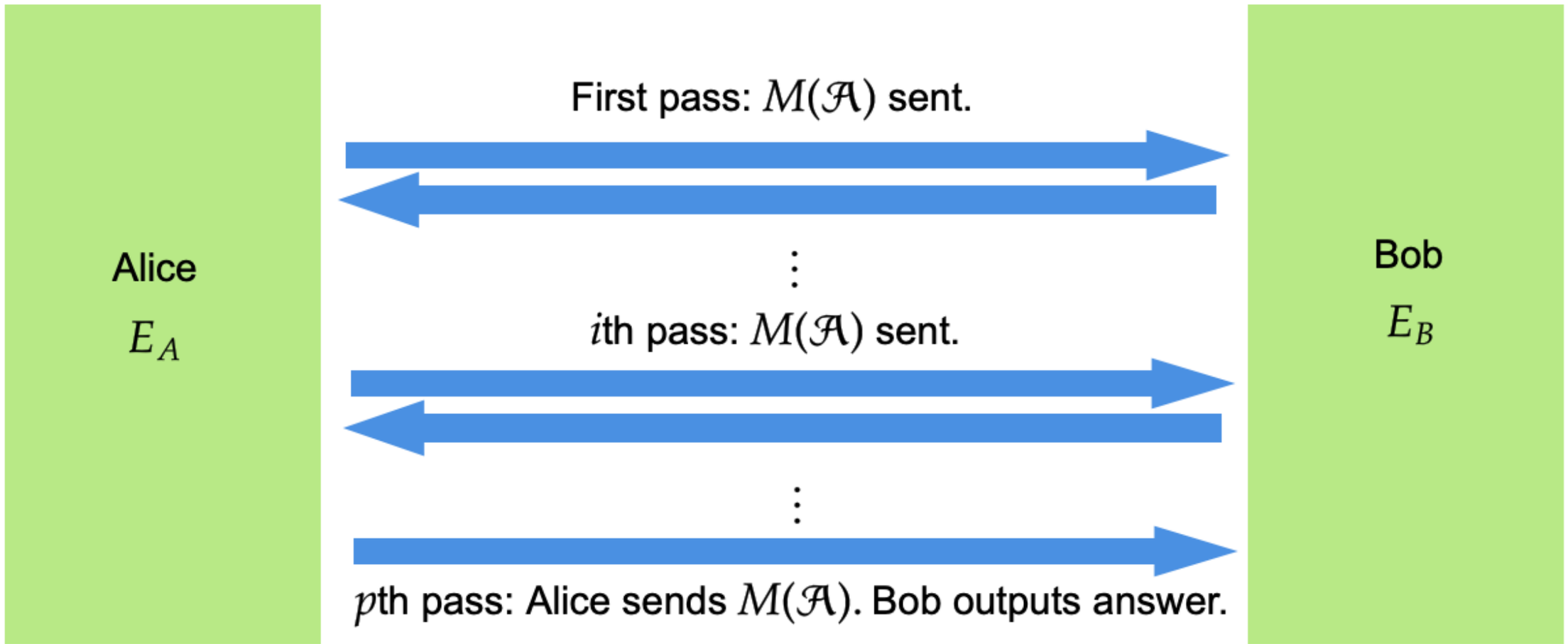
Reduction



Min-cut in this graph is $\left(\{u_{i^*}\}, V - \{u_{i^*}\}\right)$, $i^* = \arg\min_{i\in[M]} 2^N - x^{i^*} + y^{i^*}$

- If min-cut of G is less than 2^N, then for all $i \in [M], y^i < x^i$. This implies that $GT_N^M(X, Y) = 0$.
- If min-cut of G is more than 2^N, then $\exists i \in [M], y^i \geq x^i$. This implies that $GT_N^M(X, Y) = 1$.

Reduction



Algorithm $\mathcal{A}$ a p – pass algorithm to compute global min-cut

Conclusion

- A simplified two-pass streaming algorithm that outputs a minimum cut of the graph in $O(n \log n)$ space.

- Any streaming algorithm that computes the minimum cut value of an n-vertex graph in constant $p > 1$ passes needs $\Omega(n \cdot (\log n)^{1/2p-1})$ space.

Conclusion

- A simplified two-pass streaming algorithm that outputs a minimum cut of the graph in $O(n \log n)$ space.

- Any streaming algorithm that computes the minimum cut value of an n-vertex graph in constant $p > 1$ passes needs $\Omega(n \cdot (\log n)^{1/2p-1})$ space.

Open Question: Close gap between upper and lower bound.