Algorithms for Streaming Tournaments

Sahil Kuchlous
Mentor: Prantar Ghosh

REU 2023, DIMACS
Supported by NSF grant CNS-2150186
The Streaming Model

- Motivation: What if you have too much data to store in RAM?
- Sequential access to data; available memory: sublinear in the input size $n$
  - Single-pass is ideal, but $O(1)$ passes or more is also interesting
  - We don’t care much about time
- Simple example: Mean in $O(\log n)$ space
- Harder example: # of distinct elements: $(1 + \epsilon)$ approx. in $O(\epsilon^{-2} + \log n)$ space [KNW10]
Graph Streaming

- Motivation: Many graphs are too large to store (e.g., the internet, social networks)
- Input: Stream of edges in an $n$-vertex graph
  Goal: Answer questions about the graph without storing all edges
  - Most commonly $O(n \text{polylog } n)$ memory (semi-streaming); $o(n^2)$ is still interesting
  - More complex models: edge deletions, sliding window
- Simple example: S-T connectivity in undirected graphs – $O(n \log n)$ space
- Harder example: Approximate min-cut – $(1 + \epsilon)$ approx. in $O(n\epsilon^{-2})$ space [AG09]
- A good survey is [McG14]
Tournaments

• Digraphs in which all pairs of vertices have exactly 1 edge between them
  • Imagine a directed clique, or a competition where all pairs of players compete once
• Many streaming digraph problems are much easier on tournaments than on general digraphs [CGMV20]
  • SOTA: upper bounds on tournaments and lower bounds on general digraphs

Feedback Arc Set (FAS)

- Problem: What ordering of vertices minimizes number of back-edges?
- Gap between general digraphs and tournaments

<table>
<thead>
<tr>
<th>FAS</th>
<th>Tournaments</th>
<th>Digraphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper Bound</td>
<td>$(1 + \epsilon)$ approx.: (\tilde{O}(n)) space in 1 pass (exp. time) [CGMV20]</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>(\tilde{O}(n^{1+1/p})) space in (p) passes (poly. time) [BJW21]</td>
<td>?</td>
</tr>
<tr>
<td>Lower Bound</td>
<td>?</td>
<td>Any multiplicative approx.: (\Omega(n^2)) space in 1 pass (\Omega(n^{1+\Omega(1/p)})) space in (p) passes [CGMV20]</td>
</tr>
</tbody>
</table>

Our Main Goal
Our Project

• Goal: Investigate streaming algorithms on tournaments (upper and lower bounds)

• Problems we are interested in investigating:
  • Feedback arc set (FAS)
  • S-T connectivity – Given vertices s and t, is there a path from s to t?
  • Sink finding – Given a tournament with a sink, find one
  • ...


