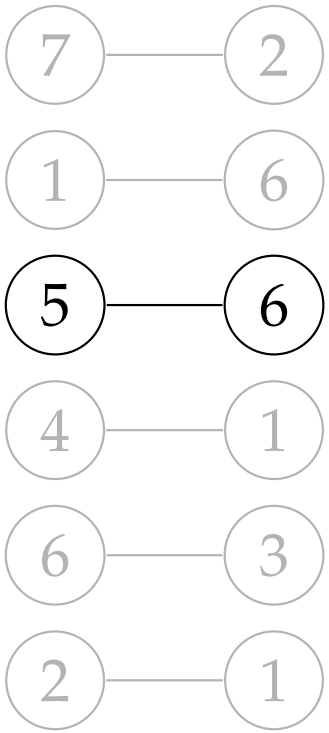
A Streaming Algorithm for the Undirected Longest Path Problem

Lasse Kliemann, Christian Schielke, Anand Srivastav

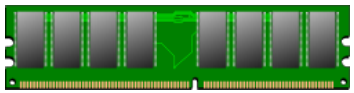
Christian-Albrechts-Universität zu Kiel
Institut für Informatik
Christian-Albrechts-Platz 4
24118 Kiel, Germany
contact: lasse@lassekliemann.de

Based on our ESA 2016 publication.

Graph Streaming Model



$\longrightarrow$ Algorithm $\longleftrightarrow$



$\mathcal{O}(n \cdot \text{poly}\log(n))$ edges

pass $\stackrel{\text{def}}{=}$ each edge is presented to the algorithm exactly once

goal: number of passes $\mathcal{O}(1)$

Graph Streaming Model

- undirected, simple graph $G = (V, E)$ with n vertices
- RAM for just $\Theta(n \cdot \operatorname{poly}\log(n))$ edges at a time
- only sequential access to graph, a pass means seeing each edge once

relatively easy tasks to do in one pass:

- determine degree sequence $(\deg(v))_{v \in V}$
- spanning tree
- minimum spanning tree
- compute a 1/2 approximation of a maximum matching
  that is, an inclusion-maximal matching

Obvious Limitations

what we do not know how to do in $\mathcal{O}(1)$ passes:

- full BFS (proven lower bound, Guruswami and Onak, 2013)
- DFS up to depth $\mathcal{O}(1)$

Shortest Paths

a $\rho := \frac{\log(n)}{\log\log(n)}$ approximation for shortest paths:

1) start with empty graph $S := K_0$
2) for each edge e in the stream:
   $S := S + e$ if that does not create a cycle shorter than ρ

analysis:

- extremal graph theory: S has only $\mathcal{O}(n \cdot \log(n))$ edges
- S is a ρ spanner,
  i. e., shortest paths in S are at most factor ρ longer than in G

Algorithms for Undirected Longest Paths

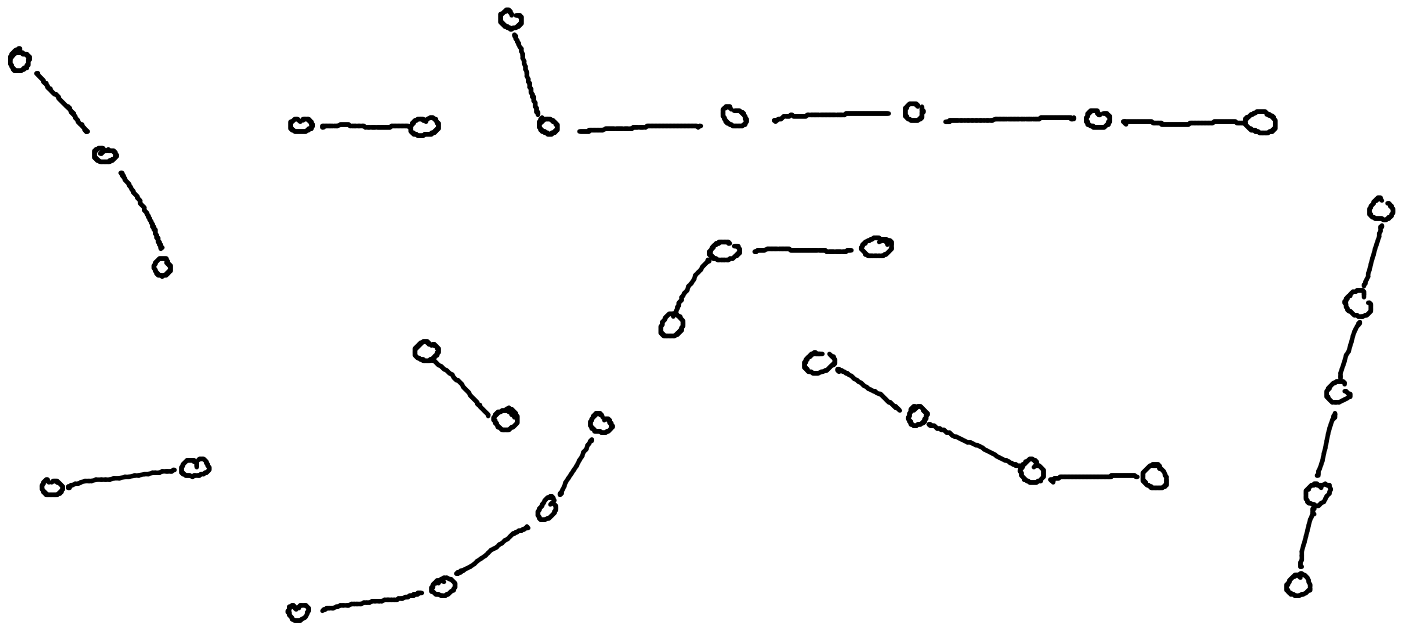
- Warnsdorf (1823), Pohl-Warnsdorf (1967):
  DFS prioritizing neighbors with few unvisited neighbors
- Pongrácz (2012):
  DFS prioritizing neighbors with large distance to fixed vertex
- Color coding (Alon et al. 1995)
- Björklund-Husfeldt (2003):
  decomposition into long cycles connected by paths
- Gabow-Nie (2008):
  extension of Björklund-Husfeldt technique

Streaming Algorithm for Undirected Longest Path

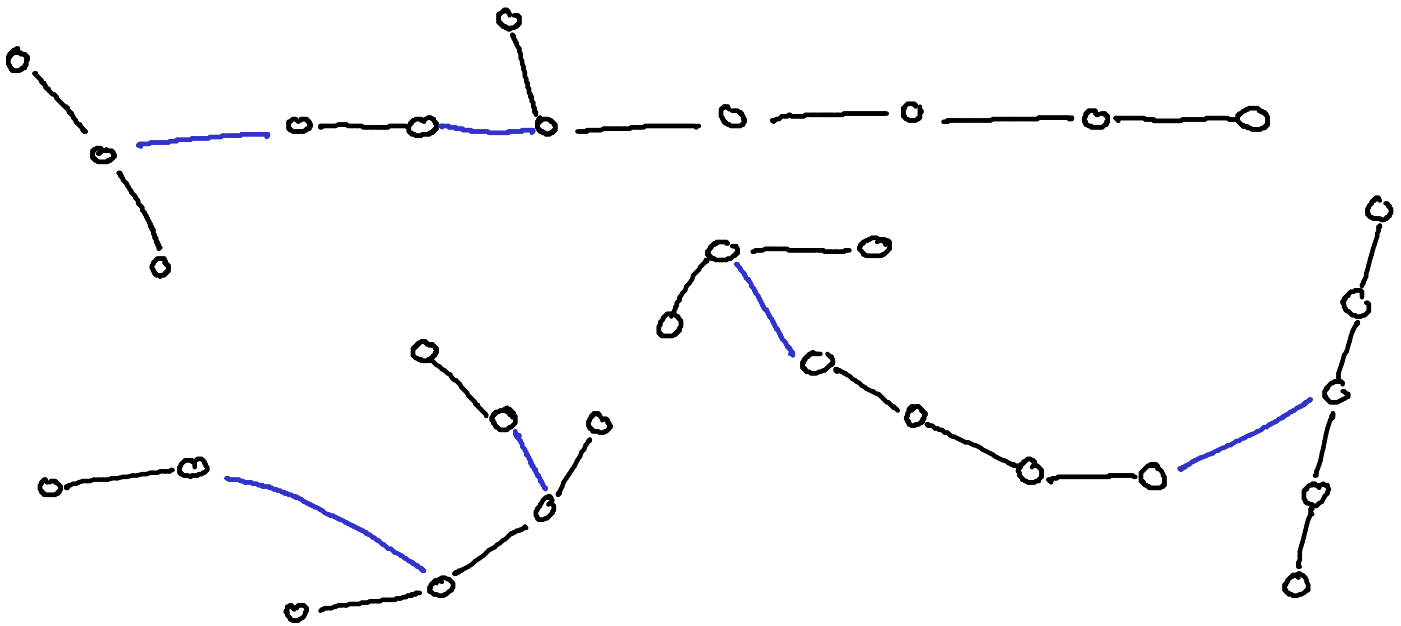
1.) construct τ spanning trees $T_1, \ldots, T_\tau$ with degree limiting
2.) find long path P in $U = \bigcup_{i=1}^{\tau} T_i$ with Warnsdorf's rule in RAM
3.) extend P to a spanning tree T
4.) improve diameter of T by a number of passes; for each edge e:
    i. consider $T' := T + e$, which contains one cycle C
    ii. find $e' \in C$ such that diameter of $T' - e'$ is maximum

Lemma: step ii can be implemented in $\mathcal{O}(n)$ time.

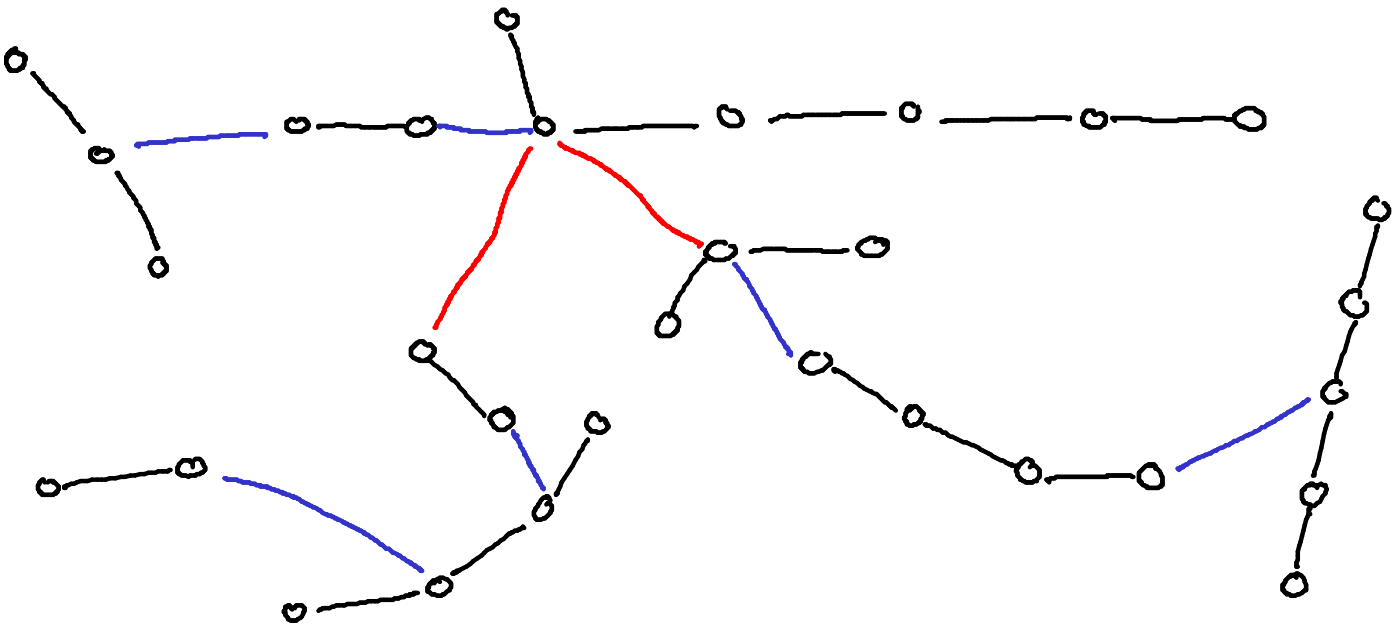
Degree Limit $D = 2$



Degree Limit $D = 3$



Degrees Unlimited



Streaming Algorithm for Undirected Longest Path

1.) construct τ spanning trees $T_1, \ldots, T_\tau$ with degree limiting

2.) find long path P in $U = \bigcup_{i=1}^{\tau} T_i$ with Warnsdorf's rule in RAM

3.) extend P to a spanning tree T

4.) improve diameter of T by a number of passes; for each edge e:

   i. consider $T' := T + e$, which contains one cycle C

   ii. find $e' \in C$ such that diameter of $T' - e'$ is maximum

Lemma: step ii can be implemented in $\mathcal{O}(n)$ time.

- in phase 1, start the first pass at a random position in the stream
- in phase 4, skip edges with both endpoints on longest path

Experiments

Random graphs with various structure:

- power law degree distribution:
  Barabási-Albert (preferential attachment), hyperbolic geometric
- small world (degrees distributed evenly)
- chains

Results for $\tau = 2$, degree limits $(2, 2, 3, \infty)$, and 3 improvement passes:

- excluding preferential attachment graphs,
  obtain 71% of best solution
- without restriction, after removing 10% worst cases,
  still obtain 71% of best solution
- without restriction, after removing 50% worst cases,
  obtain 81% of best solution
- on chain graphs, we are always the best

Challenges

- make it faster:
  - currently about 1 second per 100 edges (for $n = 1 \times 10^5$ vertices)
  - 10 days for 1×10^9 edges
- analyze it?