Distributed Triangle Detection via Graph Partitioning

Yi-Jun Chang (Michigan)

Seth Pettie (Michigan)

Hengjie Zhang (Tsinghua)

Distributed Message-passing Models

- LOCAL: Can only communicate with neighbors. Unbounded message size.
- CONGEST: Can only communicate with neighbors. $O(\log n)$-bit message size.
- Congested Clique: Allow all-to-all communication. $O(\log n)$-bit message size.

Locality

Bandwidth constraint

Effect of Locality

How different are these two models?

- CONGEST: Locality + Bandwidth constraint.
- Congested Clique: Bandwidth constraint only.

Effect of Locality

Minimum Spanning Tree

- CONGEST: $\Theta(D + \sqrt{n})$ (well-known)
- Congested Clique: $\Theta(1)$ (Jurdziński & Nowicki, SODA 2018)

Algorithms in Almost Mixing Time

Mixing time of lazy random walk

Minimum Spanning Tree

- CONGEST: $O(\tau_{\text{mix}}) \cdot n^{o(1)}$ (Ghaffari, Kuhn, and Su, PODC 2017)

- Congested Clique: $\Theta(1)$ (Jurdziński & Nowicki, SODA 2018)

Algorithms in Almost Mixing Time

Mixing time of lazy random walk

Minimum Spanning Tree

- CONGEST:

$O(\tau_{\text{mix}}) \cdot n^{o(1)}$
(Ghaffari, Kuhn, and Su, PODC 2017)

Congested clique algorithms can be simulated efficiently on graphs of low-mixing time.

Beyond Graphs of Low Mixing Time?

[Main question] Can we extend this technique to a broader graph class, or even general graphs?

Congested clique algorithms can be simulated efficiently on graphs of low-mixing time.

Graph Partition

Any graph can be decomposed into connected components of low mixing time after removing a small constant fraction of edges.

(Spielman and Teng, STOC 2004)

(Trevisan, FOCS 2005)

(Arora, Barak, and Steurer, FOCS 2010)

Our Results

- A variant of this graph decomposition can be constructed efficiently in CONGEST.

- Based on this decomposition, we obtain improved upper bounds for some problems in CONGEST.

	Previous state-of-the-art (Izumi & Le Gall, PODC 2017)	Our results
Triangle detection:	$O(n^{2/3})$	$O(n^{1-1/\omega}) < O(n^{0.58})$
Triangle listing:	$O(n^{3/4})$	$O(n^{2/3})$

Thank you!