Vivek Sardeshmukh
5th year PhD Candidate
Distributed Graph Algorithms
Advisor: Sriram Pemmaraju
Distributed Graph Algorithms on the Congested Clique

Vivek B. Sardeshmukh

The University of Iowa

Graduate Student Research Symposium
Outline

1. Graph Algorithms
2. Congested Clique Model
3. Open Problems
Outline

1. Graph Algorithms
2. Congested Clique Model
3. Open Problems
An Example: Minimum Spanning Tree (MST) Problem

Input: A weighted graph clique $G$

Output: A spanning tree with the minimum total weight of its edges
An Example: Minimum Spanning Tree (MST) Problem

**Input:** A weighted graph clique $G$

**Output:** A spanning tree with the minimum total weight of its edges
An Example: Prim’s Algorithm
An Example: Prim’s Algorithm
An Example: Prim’s Algorithm
An Example: Prim’s Algorithm
An Example: Prim’s Algorithm
An Example: Prim’s Algorithm

- One of the greedy algorithms to compute MST in sequential manner
- Running time: $O(n^2)$
Why We Need Distributed Graph Algorithms?

- To do faster computation
  - more machines $\rightarrow$ less time

- Input graph is very large (aka “Big Data”)
  - can’t fit on a single machine

- Input graph is inherently distributed
  - for example, the Internet!

Can we design fast (e.g., sub-logarithmic round) algorithm for minimum spanning tree problem?
Why We Need Distributed Graph Algorithms?

- To do faster computation
  - more machines $\rightarrow$ less time
- Input graph is very large (aka “Big Data”)
  - can’t fit on a single machine
- Input graph is inherently distributed
  - for example, the Internet!

Can we design fast (e.g., sub-logarithmic round) algorithm for minimum spanning tree problem?
Outline

1. Graph Algorithms
2. Congested Clique Model
3. Open Problems
Congested Clique Model

- Nodes in the given clique are processors
- Edges act as communication channels
Nodes in the given clique are processors
Edges act as communication channels
An Example: Minimum Spanning Tree (MST) Problem

- Initially:
  - each node knows only weights of the incident edges

- Finally:
  - each node knows which of its incident edges are in the MST
An Example: Minimum Spanning Tree (MST) Problem

Initially:
- each node knows only weights of the incident edges

Finally:
- each node knows which of its incident edges are in the MST
Communication and Computation

- Time is divided into clock ticks (rounds)
- At each clock tick a node:
  1. Does local computation
  2. Sends messages
  3. Receives messages
Round Complexity

- Time (round) complexity of an algorithm is measured as number of rounds required

Can we design sub-logarithmic round algorithm for the MST problem?
Round Complexity

- Time (round) complexity of an algorithm is measured as number of rounds required

Can we design sub-logarithmic round algorithm for the MST problem?
Congested Clique Model

Bandwidth

- Each edge can carry $O(\log n)$ bits in a single *round*

Message Complexity

- Message complexity of an algorithm is measured as the total number of messages exchanged over the execution of the algorithm
Bandwidth
- Each edge can carry $O(\log n)$ bits in a single round

Message Complexity
- Message complexity of an algorithm is measured as the total number of messages exchanged over the execution of the algorithm
This model captures the essence of cloud computing infrastructures such as:
- MapReduce, Hadoop, ... 
- Pregel, Giraph, ...
Initially each node has a distinct color
Each node \( v \) executes the following steps
A Simple Distributed MST Algorithm

Pick the minimum weight incident edge connecting to a different colored node
A Simple Distributed MST Algorithm

- Send this edge to the leader
If $v$ is leader then

- execute Prim’s algorithm on the received edges and construct a minimum spanning forest
- color the nodes in the same tree with the same unique color
- tell the sender whether the received edge is a part of this forest
- send the color of each node to the respective node
A Simple Distributed MST Algorithm

If \( v \) is leader then

- execute Prim’s algorithm on the received edges and construct a minimum spanning forest
- color the nodes in the same tree with the same unique color
- tell the sender whether the received edge is a part of this forest
- send the color of each node to the respective node
Repeat above steps until there is only one color left
A Simple Distributed MST Algorithm

Pick the minimum weight incident edge connecting to a different colored node
A Simple Distributed MST Algorithm

Send this edge to the leader
A Simple Distributed MST Algorithm

If $v$ is leader then

▸ execute Prim’s algorithm on the received edges and construct a minimum spanning forest

▸ color the nodes in the same tree with the same unique color

▸ tell the sender whether the received edge is a part of this forest

▸ send the color of each node to the respective node
A Simple Distributed MST Algorithm

If \( v \) is leader then

- execute Prim’s algorithm on the received edges and construct a minimum spanning forest
- color the nodes in the same tree with the same unique color
- tell the sender whether the received edge is a part of this forest
- send the color of each node to the respective node
A Simple Distributed MST Algorithm

Repeat above steps until there is only one color left
A vertex $v$ executes:

1. Pick the minimum weight incident edge connecting to a different colored node
2. Send this edge to the leader
3. If $v$ is leader then construct a minimum spanning forest of the received edges and inform the respective tree edges and new color
4. Repeat above steps until there is only one color left
Doubling Growth Lemma

Size of a component after a phase is at least twice the size of the smallest component in the previous phase

\[ 2 \rightarrow 2 \cdot 2 = 2^2 \rightarrow 2 \cdot 4 = 2^3 \rightarrow \ldots \ 2^i \]

Theorem

After \( \log n \) phases, we obtain a minimum spanning tree. Hence the number of communication rounds is \( O(\log n) \).
A Simple Distributed MST Algorithm: Analysis

**Doubling Growth Lemma**
Size of a component after a phase is at least twice the size of the smallest component in the previous phase

\[ 2 \rightarrow 2 \cdot 2 = 2^2 \rightarrow 2 \cdot 4 = 2^3 \rightarrow \ldots \rightarrow 2^i \]

**Theorem**
After \( \log n \) phases, we obtain a minimum spanning tree. Hence the number of communication rounds is \( O(\log n) \).
A Faster Distributed MST Algorithm

[Lotker et al., 2005] Algorithm

- [Lotker et al., 2005] designed $O(\log \log n)$-round deterministic algorithm to solve MST on a clique input graph

Quadratic Growth

After a phase, size of the smallest component is at least square of the smallest component in the previous phase.

$2 \rightarrow 2^2 = 4 \rightarrow 4^2 = 2^{2^2} \rightarrow \ldots \rightarrow 2^{2^i}$

How to get such growth?
A Faster Distributed MST Algorithm

[Lotker et al., 2005] Algorithm

- [Lotker et al., 2005] designed $O(\log \log n)$-round deterministic algorithm to solve MST on a clique input graph

Quadratic Growth

After a phase, size of the smallest component is at least square of the smallest component in the previous phase.

$2 \rightarrow 2^2 = 4 \rightarrow 4^2 = 2^{2^2} \rightarrow \ldots \rightarrow 2^{2^i}$

How to get such growth?
In the beginning of a phase, each component is of size at least $\mu$. After leader of a component selects at least $\mu$ light-weight edges connecting to $\mu$ distinct components and sends to $V^*$.
A Faster Distributed MST Algorithm: Challenges

- In the beginning of a phase, each component is of size at least $\mu$

- Leader of a component selects at least $\mu$ light-weight edges connecting to $\mu$ distinct components and sends to $V^*$
Is $\Theta(\log \log n)$ fastest?

**Lower Bounds?**

- No non-trivial lower bounds have been shown
- Any lower bounds here will imply lower bounds in circuit complexity [Drucker et al., 2014]
Can we solve MST problem in $O(1)$ rounds?
Super-Fast Distributed MST Algorithm

Our Result

A randomized $O(\log \log \log n)$-round algorithm to compute MST [Hegeman et al., 2015]

- Quadratic growth until “enough” progress is made
  - run Lotker et al. Algorithm for $O(\log \log \log)$ rounds
  - size of each component is at least poly log $n$

- Finish remaining part in $O(1)$ rounds
  - non-trivial and challenging
Super-Fast Distributed MST Algorithm: Key Techniques

To finish the remaining part in $O(1)$ rounds, we make use of the following techniques:

▶ Graph Sketches

▶ Karger-Klein-Tarjan sampling

▶ Distributed sorting in $O(1)$ rounds

▶ Distributed routing in $O(1)$ rounds

▶ *

*I’ll be happy to talk with you after the talk for more details*
Is time the only thing important?

- As in classic (sequential) algorithms there are two main metrics:
  1. time complexity
  2. space complexity

- So in distributed algorithms:
  1. round complexity
  2. message complexity
Message Complexity

- Total number of messages sent during the entire execution of the algorithm
- The super-fast algorithm uses all the links and has $\Theta(n^2)$ message complexity [PODC 2015]

Goal
- Improve over message complexity keeping the round complexity same
Congested Clique Model

Message Complexity

- Total number of messages sent during the entire execution of the algorithm
- The super-fast algorithm uses all the links and has $\Theta(n^2)$ message complexity [PODC 2015]

Goal

- Improve over message complexity keeping the round complexity same
Outline

1. Graph Algorithms
2. Congested Clique Model
3. Open Problems
Congested Clique

- High bandwidth availability over the entire network but congestion at individual nodes
- Absence of lower-bounds
  - can we solve MST in $O(1)$ rounds?
- Can we solve MST in $O(\log \log \log n)$ rounds using significantly less messages?
MST: Other Distributed Models?

- MST in Broadcast Congested Clique model?
- MST in MapReduce model?
Thank You
References

On the power of the congested clique model.

Toward Optimal Bounds in the Congested Clique: Graph Connectivity and MST.

Lessons from the congested clique applied to MapReduce.
In *Proceedings of the 21th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 149–164.

Near-constant-time distributed algorithms on a congested clique.
In *Proceedings of the 28th International Symposium on Distributed Computing (DISC)*, pages 514–530.

Minimum-Weight Spanning Tree Construction in $O(\log \log n)$ Communication Rounds.
Vivek Sardeshmukh
5th year PhD Candidate
Distributed Graph Algorithms
Advisor: Sriram Pemmaraju