236358
Distributed Graph Algorithms

Spring 2017
Class by Keren Censor-Hillel
MST in CONGEST

• **BFS-based** algorithm in $O(n)$ rounds

• **GHS algorithm** in $O(n \log n)$ rounds

• **GKP algorithm** in $O(\sqrt{n} \log^* n + D)$ rounds

• **Question:** D is necessary. What about $\sqrt{n}$?
Lower Bound for MST in CONGEST

- **Theorem**: Any MST algorithm in CONGEST needs $\Omega(\sqrt{n/\log n})$ rounds.

- **Main tool**: A reduction from the problem of 2-party set-disjointness
Lower Bound for MST in CONGEST

• **First step**: A reduction from the problem of verification of a connected spanning subgraph to MST

• **Input** for node $v$: for every $\{u,v\} \in E$, is $\{u,v\} \in H$?

• **Output** for $v$: is $H$ a connected spanning graph of $G$
  – all-accept-all-reject
  – all-accept-one-reject ($D$ rounds give all-reject)
Reduction from the verification of a connected spanning subgraph to MST

• Lemma (reduction):

If \( A \) is an algorithm for \( \text{MST} \) in CONGEST which completes in \( O(T) \) rounds, then

there is an algorithm \( A' \) for the verification of a connected spanning subgraph in CONGEST which completes in \( O(T+D) \) rounds.
Reduction from the verification of a connected spanning subgraph to MST

• **Proof:**

• The nodes assign the following weights to each edge $e=\{u,v\}$ in $E$:
  - If $e \in H$, assign $w(e) \leftarrow 0$
  - Otherwise, assign $w(e) \leftarrow 1$
Reduction from the verification of a connected spanning subgraph to MST

• **Proof:**
  – If $e \in H$, assign $w(e) \leftarrow 0$
  – Otherwise, assign $w(e) \leftarrow 1$

• $w(\text{MST})=0$ if and only if $H$ is a **connected spanning subgraph**

• Also, each node can locally check whether the MST contains one of its edges that is not in $H$. 
Lower bound for the verification of a connected spanning subgraph

• **Theorem**: Any algorithm for the verification of a connected spanning subgraph in CONGEST needs $\Omega(\sqrt{n}/\log n)$ rounds.

• This implies the same lower bound for the construction of an MST, by the Reduction Lemma.

• **Proof**: By a reduction from the problem of 2-party set-disjointness
2-party communication

• Two players, *Alice* and *Bob*

• Inputs: \( x^A = (x^A_1, \ldots, x^A_k), \ x^B = (x^B_1, \ldots, x^B_k) \) in \( \{0,1\}^k \)

• Players exchange bits according to a protocol \( \pi \)

• Outputs: \( y^A, \ y^B \) in \( \{0,1\} \)
2-party communication

• **Set Disjointness:**

• **Inputs:**
  \[ x^A = (x^A_1, \ldots, x^A_k), \quad x^B = (x^B_1, \ldots, x^B_k) \] in \( \{0,1\}^k \)

• **Outputs:**
  \[ y^A = y^B = 1 \] if and only if
  there is an \( i \in \{1, \ldots, k\} \) such that \( x^A_i = x^B_i = 1 \)
Lower Bound for Set Disjointness

• **Theorem**: The communication complexity of set disjointness is $k+1$
Lower bound for the verification of a connected spanning subgraph

• **Theorem**: Any algorithm for the verification of a connected spanning subgraph in CONGEST needs $\Omega(\sqrt{n} / \log n)$ rounds.

• This implies the same lower bound for the construction of an MST, by the Reduction Lemma.
Base Graph

• Illustration

• The number of nodes is $n = \Theta(kd^q)$
• The diameter is $D = 2q + 2$
Base Graph

**spikes from** \( u^q_i \) **to** \( v^j_i \)

**A**=\( u^q_0 \)

\[
\begin{align*}
A &= u^q_0 \\
v^0_0 & \cdots \\
v^1_0 & \cdots \\
v^{k-1}_0 & \cdots \\
\end{align*}
\]

**B**=\( u^q_{dq-1} \)

\[
\begin{align*}
B &= u^q_{dq-1} \\
v^0_{dq-1} & \cdots \\
v^1_{dq-1} & \cdots \\
v^{k-1}_{dq-1} & \cdots \\
\end{align*}
\]

**complete** \( d \)-ary tree of depth \( q \)

**k paths of length** \( d^q \)
Input-Based Graph

• Given an input $x^A=(x^A_1,\ldots, x^A_k)$, $x^B=(x^B_1,\ldots, x^B_k)$ to Set Disjointness, define $H$ in $G$ as follows:
  
  – All paths are in $H$
  
  – All tree edges are in $H$
  
  – All spikes from $u^{q_i}$ for $i=1,\ldots,d^q\cdot2$ are not in $H$
  
  – For $j=1,\ldots,k$, the spike from $u^{q_0}$ to $v^i_0$ is in $H$ if and only if $x^A_j=0$
  
  – For $j=1,\ldots,k$, the spike from $u^{q_{dq-1}}$ to $v^i_{dq-1}$ is in $H$ if and only if $x^B_j=0$
Input-Based Graph

A = $u^q_0$

B = $u^q_{dq-1}$

$k$ paths of length $d^q$

Complete $d$-ary tree of depth $q$

Spikes from $u^q_i$ to $v^j_i$
Is H Connected?

• **Claim 1:** H is not connected if and only if there is a $j=1,\ldots,k$ such that $x^A_j=x^B_j=1$

• **Proof:** If $x^A_j=x^B_j=1$ then both edges $(u^{q_0}_j,v^{j_0}_j)$ and $(u^{q_{dq-1}}_j,v^{j_{dq-1}}_j)$ are not in H, and hence the $j$-th path is a connected component of H.

• Otherwise, $(u^{q_0}_j,v^{j_0}_j)$ or $(u^{q_{dq-1}}_j,v^{j_{dq-1}}_j)$ is in H for all $j$, and all paths are connected to the tree
Simulation

• **Claim 2**: If ALG is a distributed algorithm for verifying a connected spanning subgraph in $R$ rounds with $R \leq (d^q - 1)/2$ then Alice and Bob can **simulate ALG** by exchanging $O(d^q R \log n)$ bits.
Lower bound for the verification of a connected spanning subgraph

• **Theorem**: Any algorithm for **the verification of a connected spanning subgraph** in CONGEST needs $\Omega(\sqrt{n/logn})$ rounds.

• **Proof**: By **Claim 1**, if A and B simulate ALG then they solve Set Disjointness. Since the CC of Set Disjointness is $\Omega(k)$, by **Claim 2**, $R=\Omega(\min(d^a,k/dqlogn))$
Lower bound for the verification of a connected spanning subgraph

- $R = \Omega(\min(d^q, k/dq \log n))$
- Now choose $k = d^{q+1}q \log n$
- Then $R = \Omega(d^q)$

- Recall $n = \Theta(kd^q)$, then $n = \Theta(d^{2q+1}q \log n)$
- Then $R = \Omega(d^q) = \Omega((n/q \log n)^{q/(2q+1)})$
  $$= \Omega((n/q \log n)^{1/2 - 1/(2(2q+1)))}$$

- Now choose $q = \log n$, then $R = \Omega(\sqrt{n}/\log n)$
Simulation

• Claim 2: If ALG is a distributed algorithm for verifying a connected spanning subgraph in $R$ rounds with $R \leq (d^q-1)/2$ then Alice and Bob can simulate ALG by exchanging $O(dqR\log n)$ bits

• Proof: We show that after exchanging $O(dqR\log n)$ bits, Alice knows the local state of $A = u^q_0$ after $R$ rounds of ALG, and Bob knows the local state of $B = u^q_{dq-1}$ after $R$ rounds of ALG.
Simulation

- \( T(S) = \) tree nodes with descendants in \( S \)

- \( L_0 = V \setminus \{B\} \)
- \( L'_i = \{u^q_j \mid j \leq d^q-1-i\} \)
- \( L_i = \bigcup_{1 \leq t \leq k} \{v^t_j \mid j \leq d^q-1-i\} \cup L'_i \cup T(L'_i) \)

- \( R_0 = V \setminus \{A\} \)
- \( R'_i = \{u^q_j \mid j \geq i\} \)
- \( R_i = \bigcup_{1 \leq t \leq k} \{v^t_j \mid j \geq i\} \cup R'_i \cup T(R'_i) \)
Input-Based Graph

\begin{itemize}
\item \textbf{Spikes from} \( u^q_i \) \textbf{to} \( v^j_i \)
\item \textbf{Complete} \( d \)-ary tree of depth \( q \)
\item \textbf{K paths of length} \( d^q \)
\end{itemize}

\[ A = u^q_0 \quad \ldots \quad B = u^q_{dq-1} \]

\[ v^0_0 \quad \ldots \quad v^{k-1}_0 \quad \ldots \quad v^0_{dq-1} \quad \ldots \quad v^{k-1}_{dq-1} \]
spikes from $u^q_i$ to $v^j_i$
spikes from \( u^q_i \) to \( v^j_i \)

complete \( d \)-ary tree of depth \( q \)

\[ R_0 \]

A = \( u^q_0 \)

B = \( u^q_{dq-1} \)

k paths of length \( d^q \)
k-ary trees of depth \( q \)

Spikes from \( u^{q_i} \) to \( v^{i_1} \)

\( A = u^{q_0} \)

\( B = u^{q_{d^q-1}} \)

\( d^q \) paths of length
$R_2$

complete $d$-ary tree of depth $q$

spikes from $u^q_i$ to $v^j_i$

$k$ paths of length $d^q$
paths of length $d^q$ from $u^q_i$ to $v^j_i$
The state of B after t rounds

• To know the state of nodes in $R_t$ after $t$ rounds, it is enough to know the state of nodes in $N(R_t)$ after $t-1$ rounds

• To know the state of nodes in $R_t$ after $t$ rounds, it is enough to know the state of nodes in $R_{t-1}$ after $t-1$ rounds and the messages sent from $V \setminus R_{t-1}$ to $R_t$ in round $t$
The state of B after t rounds

- By induction on t, there are at most \( dq \) messages from \( V \cap R_{t-1} \) to \( R_t \) in round t, and at most \( dq \) messages from \( V \cap L_{t-1} \) to \( L_t \) in round t
  1. All are on into the leftmost/rightmost tree nodes in \( R_t/L_t \) at every level
  2. \( V \cap R_{t-1} \) is a subset of \( L_{t-1} \)
  3. \( V \cap L_{t-1} \) is a subset of \( R_{t-1} \)
- In total, \( O(dqR \log n) \) bits
- Items 2 and 3 hold only for \( R \leq (d^q - 1)/2 \)
Lower Bound for MST in CONGEST

• **Theorem**: Any MST algorithm in CONGEST needs \( \Omega(\sqrt{n/\log n}) \) rounds.
Local/Global Problems

• We need $\Omega(D)$ rounds for BFS, MST

• Do we need $\Omega(D)$ rounds for all interesting problems?

• No:
  – BFS, MST are global problems
  – some problems are local problems
c-Coloring

• A function $\varphi: V \rightarrow \{1, \ldots, c\}$ is a c-coloring if for every $u, v \in V$ such that $\{u, v\} \in E$ it holds that $\varphi(u) \neq \varphi(v)$

• If $G$ has a c-coloring then $G$ is c-colorable
**Chromatic Number**

- The chromatic number $\chi(G)$ of $G$ is the smallest $c$ for which $G$ is $c$-colorable
  - Finding $\chi(G)$ or a $\chi(G)$-coloring is NP-hard

- Every graph has a $(\Delta+1)$-coloring
  - $\Delta$ is the maximal degree in the graph

- **Proof**: The greedy sequential algorithm
Distributed Coloring

• LOCAL model

• Each node $v$ outputs a color $\varphi(v)$ such that $\varphi$ is a $c$-coloring

• Greedy can be simulated in $n$ rounds in LOCAL
Color Reduction

• Given a $c$-coloring $\varphi$, obtain a $(\Delta+1)$-coloring

1. for $i=c, \ldots, \Delta+2$ do
2. if $\varphi(v)=i$ then
3. $\varphi(v) \leftarrow \min\{x | \varphi(u) \neq x \text{ for all } u \in N(v)\}$
4. send $\varphi(v)$ to all neighbors
5. return $\varphi(v)$
Color Reduction

• **Correctness:**
  - For each $v$, $\varphi(v) \leq \Delta + 1$ because at most $\Delta$ colors are used by neighbors.
  - The coloring is valid, by induction on the round number (starts valid and remains valid at the end of the round).

• **Round complexity:** $c-\Delta-1$
  - This is $O(n-\Delta)$ if we start with IDs as colors