Distributed Minimum Spanning Tree
Outline

I. Introduction
II. GHS algorithm (1983)
III. Awerbuch algorithm (1987)
IV. Other results
Introduction
Problem and Model

- Undirected $G(N,E)$

);$ asynchronous algorithm for MST of $G$

- Each node executes same local algorithm (send/wait messages and processing)
- Messages transmitted independently, arrive after finite delay with no errors
- Distinct node IDs and edge weights
MST properties

1. Distinct weights $\rightarrow$ unique MST
2. Subset of MST + lighter outgoing edge = bigger subset of MST (greedy approach)
Distributed MST: Intuitively

- Fragments (partial MSTs) grow in parallel until they cover the entire graph
GHS algorithm

Robert G. Gallager, Pierre A. Humblet, and P. M. Spira,
"A distributed algorithm for minimum-weight spanning trees,"
GHS Notation: Fragments

- Fragment = connected component with own MST
- Each Fragment has Level and ID
  - At absorption, small Level fragment inherits Level and ID from larger fragment
  - At merge \((L_1=L_2)\), new Level:= \(L_1+1\) and new ID:= edge connecting the two fragments (core)
  - Initially: each node is a fragment with Level=0
GHS Notation: Node states

- **SLEEPING**
  - Initial state
- **SEARCHING**
  - mwoe search for a fragment
- **FOUND**
  - Other times
GHS Notation: Edge states

- **Basic**
  - Uncharacterized
- **Active**
  - MST branch
- **Rejected**
  - Not in MST
GHS: SEARCHING state

- Each node:
  - Finds minimum-weight *basic* edge
  - Sends *Test* message: \(< \text{fr.ID} ; \text{fr.LEVEL} >\)
    - *Reject* if IDs agree
    - *Accept* if receiver’s fr.LEVEL is greater or equal
    - Else response is delayed
GHS: Search cooperation

- Nodes cooperate to find fragment’s mwoe:
  - *Accept* responses $\rightarrow$ *Report(W)* messages
  - Minimum-weight reports are relayed to the fragment’s core (on-the-fly comparisons).
  - *mwoe* broadcasted to the fragment nodes
  - mwoe node sends *Connect(L)*
GHS: Fragment connection

- On \textit{Connect}(L) messages:
  
  i. \( L_{\text{sender}} = L_{\text{receiver}} \)
      
      ➤ new \((L+1)\)-fragment with new core. \textit{Initiate} messages are broadcasted: \(< L+1 ; \text{ID} >\)

  ii. \( L_{\text{sender}} < L_{\text{receiver}} \)
      
      ➤ immediate absorption. \textit{Initiate} message to small fragment includes \textit{search} command (if \textit{Report} is not already already sent)
GHS: Example

\[
\begin{align*}
F & : \{1\} \rightarrow \{1,2\} \rightarrow \{1,2,3\} \\
F' & : \{5\} \rightarrow \{5,6\} \\
F'' & : \{1,2,3\} \rightarrow \{1,2,3,5,6\} \rightarrow \{1,2,3,5,6,4\}
\end{align*}
\]
GHS: Correctness

- Preserves MST properties 1 and 2 (greedy approach)
- No deadlocks

⇒ Terminates and leads to MST
GHS: Communication cost

- A Fr-(L+1) contains at least two Fr-(L)
  - A Fr-(L) contains at least $2^L$ nodes
  - Upper bound on levels: $\log_2 N$
- At each level, a node can send/receive at most one \textit{Accept}, \textit{Initiate}, \textit{Report}, \textit{Connect}
  and successful \textit{Test} message
- We have at most $2 \cdot E$ \textit{Reject} messages

$\Rightarrow$ Total messages $\leq 5 \cdot N \cdot \log_2 N + 2 \cdot E$
GHS: Timing analysis

- $5LN-3N$ time units for all nodes to reach $L$
  - Proof by induction on $L$ (fragment level).
  - **Key note:** propagation of cooperation signals within a fragment, requires $O(N)$ time units.

⇒ Time complexity $= O(N \cdot \log_2 N)$
Awerbuch algorithm

Baruch Awerbuch,

“Optimal Distributed Algorithms for Minimum Weight Spanning Tree, Counting, Leader Election, and Related Problems,”

GHS: Disadvantage

- Level is not accurate metric for fragm. size
  ⇒ A fragment might wait for its “small” neighbor to grow, even if the neighbor is big enough for the merge
 ’il The nodes of this fragment remain idle!
Awerbuch: Ideas

✓ Update Level more aggressively
  - estimate the fragment size

✗ Size estimating increases message count
  - do not start estimating from the beginning
Awerbuch: size of G(N,E)

- Awerbuch introduces a counting algorithm using $O(N \cdot \log_2 N + E)$ messages running in $O(N)$ time
- We count N before starting the MST alg.
Awerbuch: The phases

1) GHS until fragment sizes = $\Omega(N / \log_2 N)$
2) Modified GHS (estimating fragment sizes)
Awerbuch: Estimating sizes

- Mechanisms:
  - Test-Distance
  - Root-Update

- Special exploration tokens (messages)
Awerbuch: *Test-Distance*

- **When?**
  - Tree(v) connects to Tree(w)
  - Node v tries to find-distance from node w

- **How?**
  - Messages with counter $2^{L(v)+1}$ relayed to father
  - Each father subtracts #sons from counter
  - Token fails $\rightarrow L(v)++; \rightarrow v$ restarts procedure

- **Why?**
  - Token fails $\Rightarrow$ fragment size $\geq 2^{L(v)+1}$
Awerbuch: *Root-Update*

- **When?**
  - Fragment-L starts searching for mwoe

- **How?**
  - If: *Initiate* message relayed for $\geq 2^{L+1}$ nodes
  - If: Any node counts $\geq 2^{L+1}$ internal edges
  - Then: increase L, restart mwoe search

- **Why?**
  - Level L implies $2^L$ nodes, but #nodes $\geq 2^{L+1}$
Awerbuch: Communication Cost

- Phase 1 messages: bounded by GHS cost
- Phase 2 messages:
  - *Root-Update*
    - Constant number per node per level ⇒ $O(N \log_2 N)$
  - *Test-Distance*
    - Total tokens per Test-Distance = $O(\text{final_token}) = O(N)$
    - Any node receives ≤1 final token per sub-tree of ph.1
    - Maximum number of ph.1 sub-trees = $\log_2 N$

- Cost = $O(N \cdot \log_2 N + E)$
Phase 1: $O(N)$
- Intuitively, GHS time for graph sizes $N/\log_2 N$

Phase 2: $O(N)$
- Small fragments increase rapidly their Level: immediate responses, $O(2^{L+1})$ time for mwoe, $O(2^{L+1})$ time for root-updates and test-distances
- The period of time in which SL is the smallest Level in the network is upper-bounded $O(2^{SL})$
- The period sum of $\log_2 N$ Levels is $O(2^{\log N})$
Other results
Other Results

Juan Garay, Shay Kutten and David Peleg,
"A Sub-Linear Time Distributed Algorithm for Minimum-Weight Spanning Trees (Extended Abstract),"

\[ \text{Time } O(Diam(G) + N^{\varepsilon}\log^*N), \quad \varepsilon \approx 0.6 \]

David Peleg and Vitaly Rubinovich
“A near tight lower bound on the time complexity of Distributed Minimum Spanning Tree Construction,”

\[ \text{D-MST Time lower bound } \Omega(N^{1/2} / \log_2 N) \]