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)

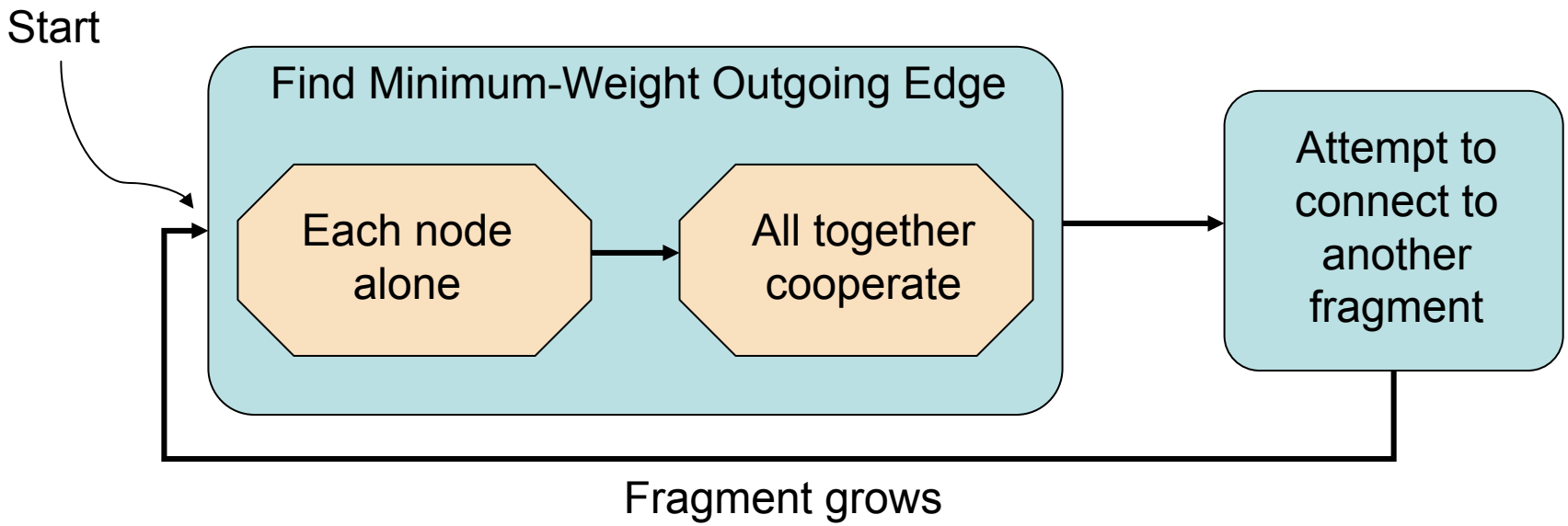
✓ Messages transmitted independently, arrive after finite delay with no errors

✓ Distinct node IDs and edge weights

MST properties

1. Distinct weights → 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,

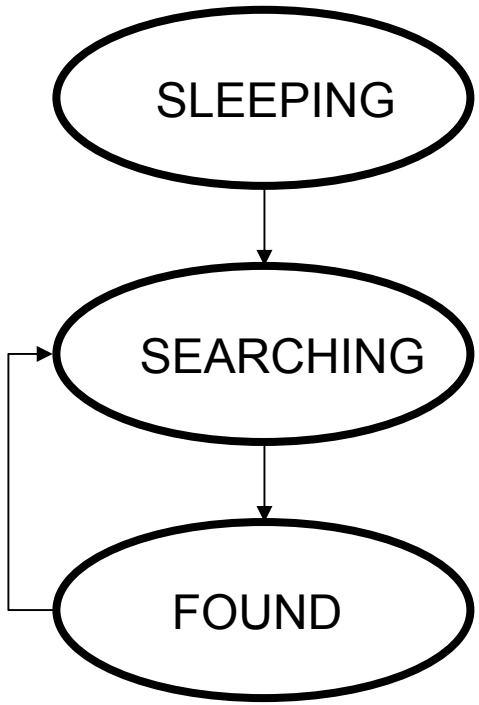
"*A distributed algorithm for minimum-weight spanning trees*,"

ACM TOPLAS,vol.5, no. 1, pp. 66--77, January 1983.

GHS Notation: Fragments

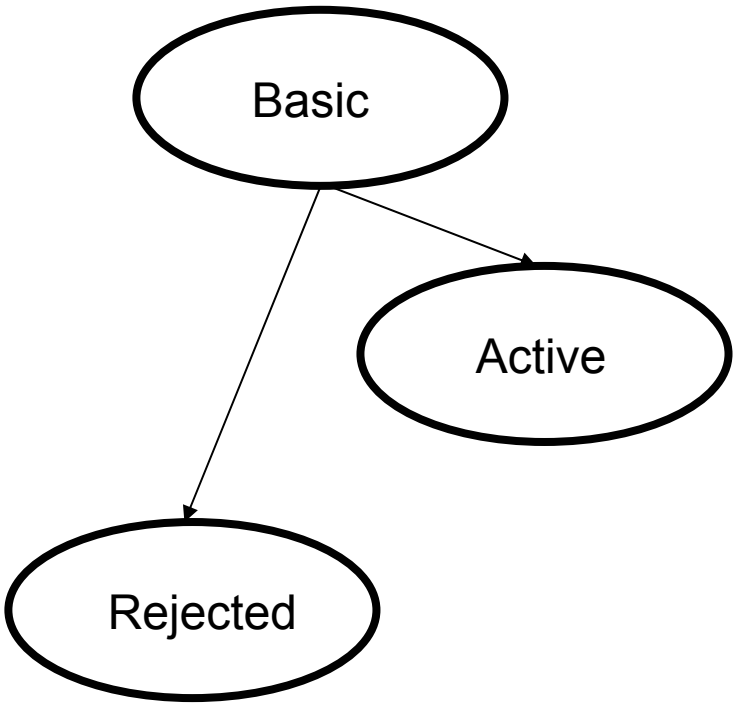
- Fragment = connected component with own MST
- Each Fragment has Level and ID
  - At absorption, small Level fragment inherits Level and ID from larger fragment
  - At merge (L_1=L_2), new Level:= L_1+1 and new ID:= edge connecting the two fragments (core)
  - Initially: each node is a fragment with 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: < fr.ID ; 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 → *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 *Connect(L)* messages:

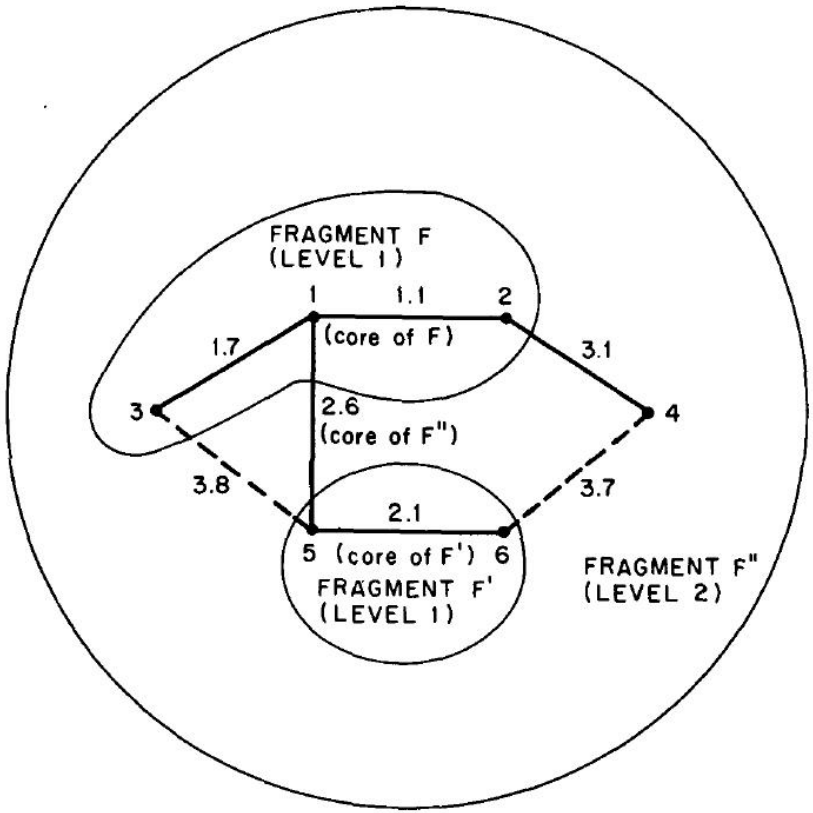
i. $L_{sender} = L_{receiver}$

   ↳ new (L+1)-fragment with new core. *Initiate* messages are broadcasted: < L+1 ; ID >

ii. $L_{sender} < L_{receiver}$

   ↳ immediate absorption. *Initiate* message to small fragment includes *search* command (if *Report* is not already sent)

GHS: Example



$$\left\{ \begin{array}{l} F : \{1\} \rightarrow \{1,2\} \rightarrow \{1,2,3\} \\ F' : \{5\} \rightarrow \{5,6\} \end{array} \right\} \longrightarrow \left\{ F'' : \{1,2,3\} \rightarrow \{1,2,3,5,6\} \rightarrow \{1,2,3,5,6,4\} \right\}$$

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 *Accept*, *Initiate*, *Report*, *Connect* and successful *Test* message
- We have at most 2·E *Reject* messages

⇒ Total messages ≤ $\mathbf{5 \cdot N \cdot log_2 N + 2 \cdot E}$

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 = $\boldsymbol{O(N \cdot log_2 N)}$

Awerbuch algorithm

Baruch Awerbuch,

"*Optimal Distributed Algorithms for Minimum Weight Spanning Tree, Counting, Leader Election, and Related Problems*,"

Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC), New York City, New York, May 1987.

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

- 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 = $\boldsymbol{\Omega(N / \log_2 N)}$
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 → L(v)++; → v restarts procedure
- Why?
  - Token fails ⇒ fragment size ≥ $2^{L(v)+1}$

Awerbuch: *Root-Update*

- When?
  - Fragment-L starts searching for mwoe
- How?
  - If: *Initiate* message relayed for ≥ 2^{L+1} nodes
  - If: Any node counts ≥ 2^{L+1} internal edges
  - Then: increase L, restart mwoe search
- Why?
  - Level L implies 2^{L} nodes, but #nodes ≥ 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(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 = $\boldsymbol{O(N \cdot \log_2 N + E)}$

Awerbuch: Timing Analysis

- 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 $O(2^{logN})$

Other results

Other Results

Juan Garay, Shay Kutten and David Peleg,

"A Sub-Linear Time Distributed Algorithm for Minimum-Weight Spanning Trees (Extended Abstract),"

IEEE Symposium on Foundations of Computer Science, 1993.

- Time $O\,(Diam(G) + N^{\varepsilon} log^{*}N)$, $\varepsilon \approx 0.6$

David Peleg and Vitaly Rubinovich

"A near tight lower bound on the time complexity of Distributed Minimum Spanning Tree Construction,"

SIAM Journal of Computing, 2000, and IEEE Foundations of Computer Science (FOCS) Symposium, 1999.

- D-MST Time lower bound $\Omega\,(N^{1/2} / log_2 N)$