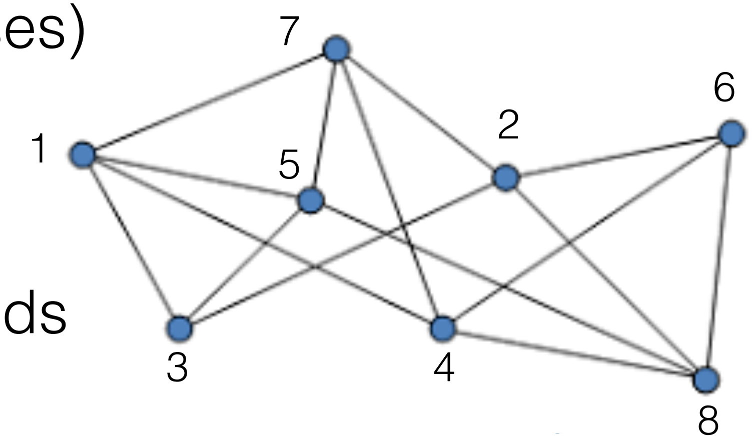
ADFOCS Lectures

- ☐ Asynchronous Crash-Prone Distributed Computing
- ☐ Locality in Distributed Network Computing
- ☑ **Congestion-Prone Distributed Network Computing**
- ☐ Other Aspects of Distributed Computing

CONGEST Model

- Each process is located at a node of a network modeled as an n-node graph (n = #processes)



- Each process has a unique ID in {1,…,n}

- Computation proceeds in synchronous rounds during which every process:

    1. Sends a message to each neighbor

    2. Receives a message from each neighbor

    3. Performs individual computation (same algorithm for all nodes)

Typically, B = O(log n)

Non-local problems

All-Pairs Shortest Paths

Specification: Every node v aims at computing $dist_G(v,u)$ for every other node

General idea (for unweighted graphs):

- Every node u launches a signal performing BFS(u)
- Whenever v receives signal of BFS(u), it sets $dist_G(v,u)$ = #hops performed by the signal from u

Issue: Several signals may traverse the same edge at the same round.

☞ The signals must be scheduled carefully.

Linear time algorithm

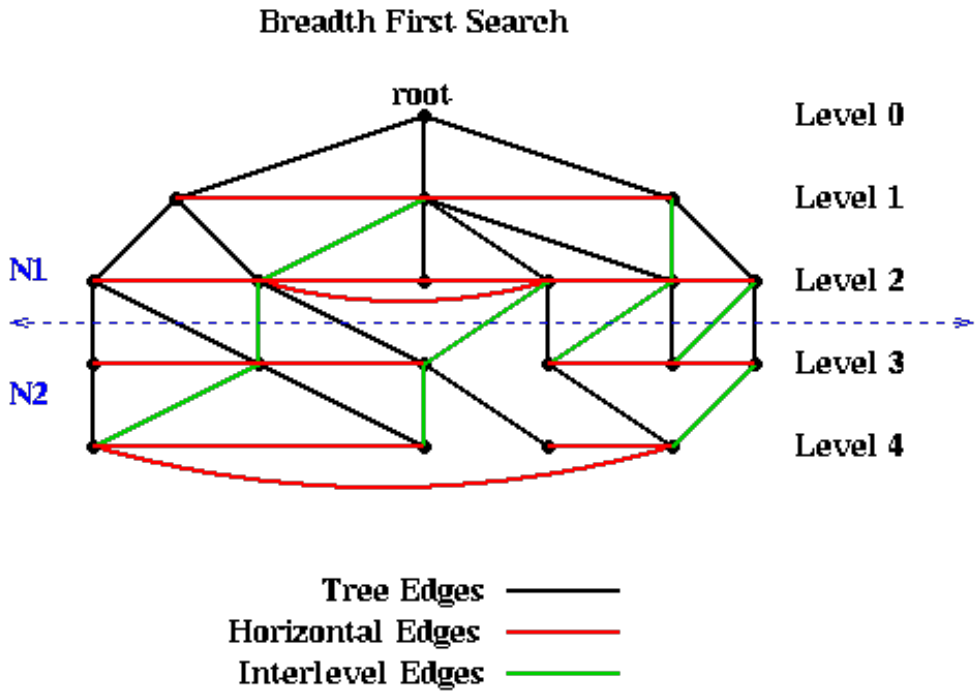
Theorem [S. Holzer, R. Wattenhofer (2012)]
APSP can be solved in $O(n)$ rounds in the CONGEST model.

Proof. Scheduling of the signals:

- Construct a BFS tree rooted a node with smallest ID
- Perform a DFS traversal of the tree where, whenever reaching a node u for the first time:
  (1) wait 1 round,
  (2) launch the BFS signal of u
  (3) move to next DFS node.

See https://users.ics.aalto.fi/suomela/apsp/ ❑

BFS Construction



Lemma BFS construction requires $O(D)$ rounds in the CONGEST model

Algorithm of node u

$\mathtt{id}_{min} \leftarrow ID(u)$

repeat

send $\mathtt{id}_{min}$ to neighbors, and receive IDs from neighbors

if $\exists\ \mathtt{id} \in$ {IDs sent by neighbors} : $\mathtt{id} < \mathtt{id}_{min}$ **then**

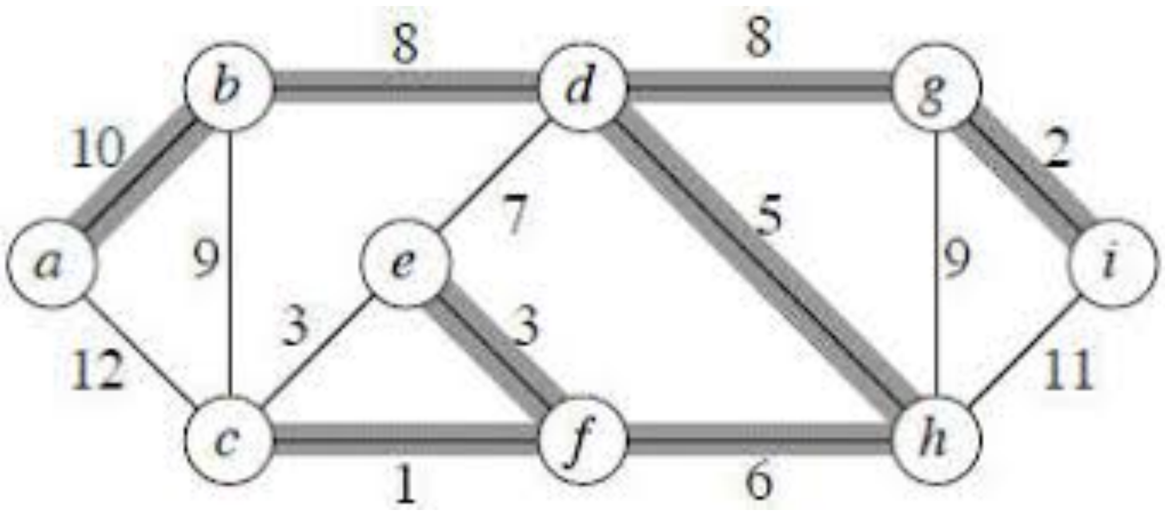
$\mathtt{id}_{min} \leftarrow \mathtt{id}$

parent(u) ← ID(v) where v is the neighbor which sent `id`

Weighted Graphs

Cf. Cristoph Lenzen's lecture!

Minimum Spanning Tree (MST)



Input of node u : ID(u), w(e) for every e ∈ E(u)

Output of node u : list of edges e ∈ E(u) belonging to MST

Facts about MST

Let $G = (V,E)$ be a connected weighted graph

- Without loss of generality, all weights can be assumed distinct ➟ for every $e = \{u,v\}$ with $ID(u) > ID(v)$, replace $w(e)$ by $(w(e),ID(u),ID(v))$.
- For every **cut** $(S,V \setminus S)$ in G, the edge of **minimum** weight in the cut belongs to the MST.
- For every **cycle** C in G, the edge of **maximum** weight in C does not belong to the MST

MST is a non-local problem

$$I_1 = (1, 3) \quad I_2 = (3, 2) \quad I_3 = (1, 2)$$

Remark MST requires at least D rounds in the cycle.

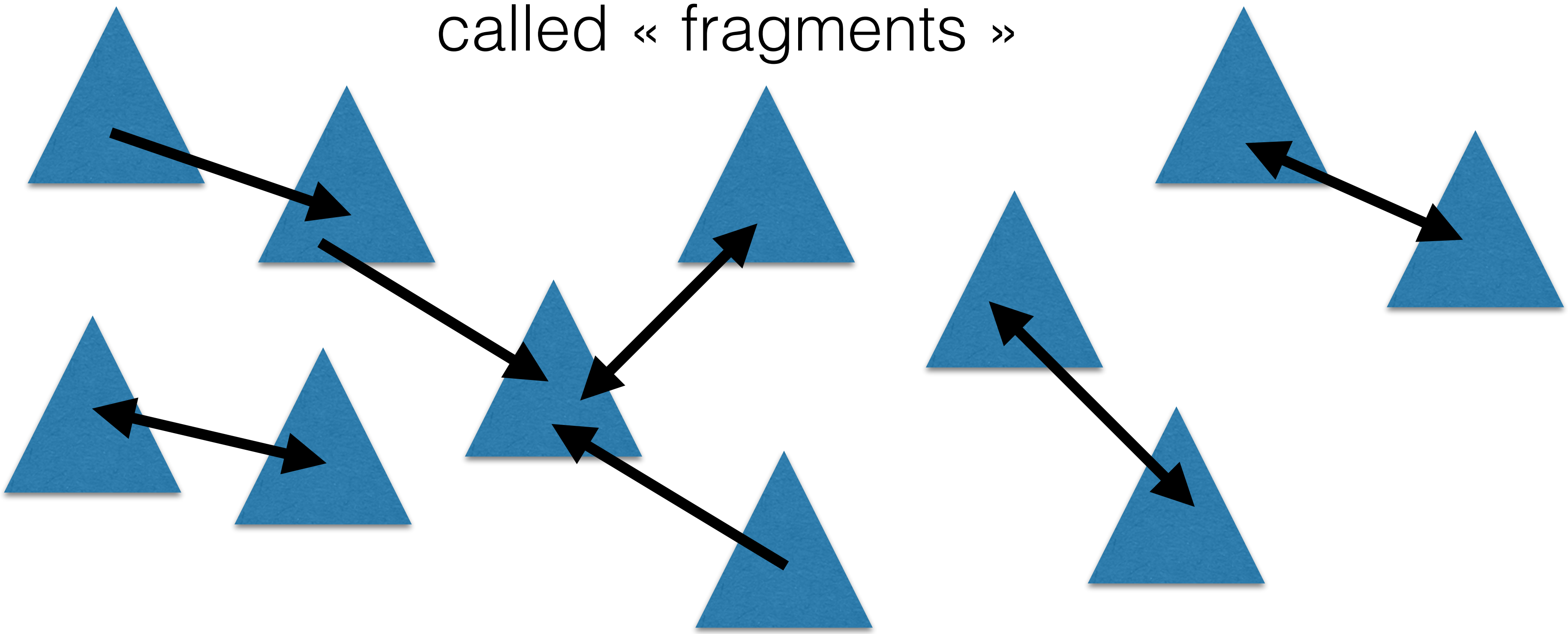
Algorithms with round-complexity O(f(n)+D)
in n-node graphs of diameter D.

Objective: minimizing f(n)

Borůvska's algorithm (1926)

distributed version

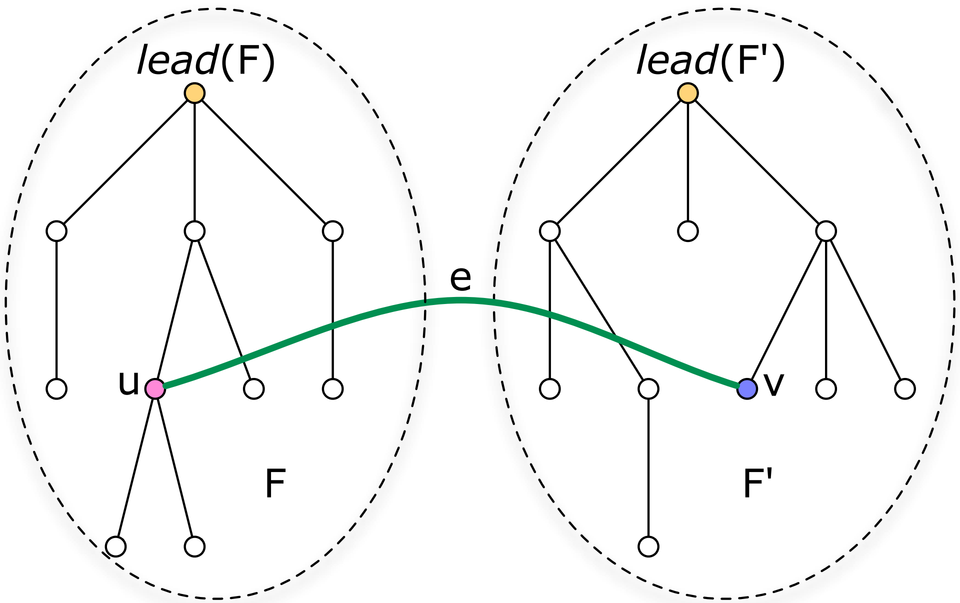
Collection of subtrees
called « fragments »

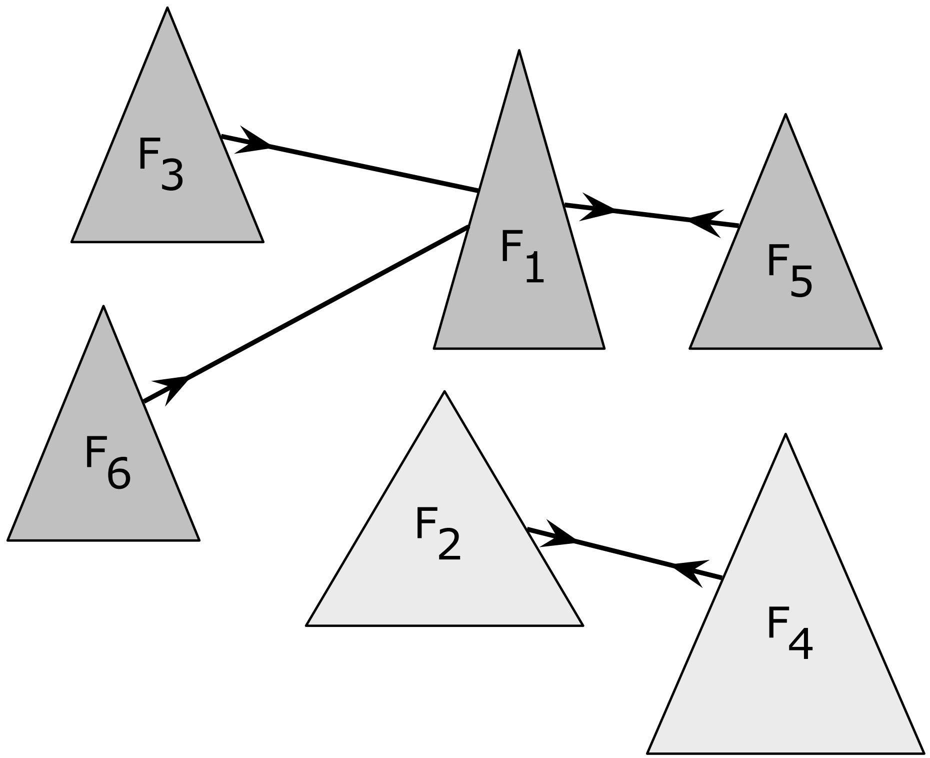


A phase = fragments
are merged

Merges use the edge of minimum
weight going out of each fragment

Fragments & Merging





$e = (ID(u), ID(v), w(e))$

$N(t)$ = #fragments after t rounds

$N(0) = n$
$N(t+1) \leq N(t)/2$

$\Rightarrow$ at most $\lceil \log_2 n \rceil$ phases

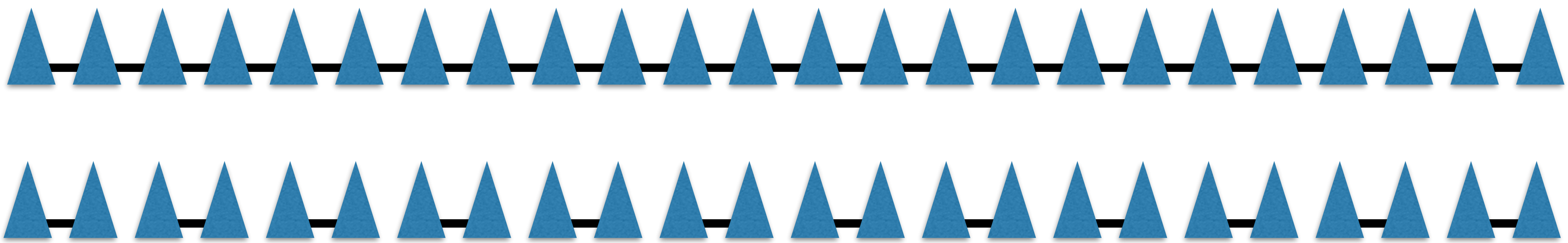
Round complexity

complexity of a phase = $O(\max_F \text{diam}(F))$

$\text{diam}(F) \leq n-1$

Theorem The distributed version of Borůvska's algorithm can be implemented in $O(n \log n)$ rounds in the CONGEST model.

The bound is tight:



Matroid Algorithm (1)

Algorithm for a node u

$K \leftarrow E(u)$ edges incident to node u

wait until having received an edge from each child

repeat

know $K \leftarrow K \cup \{\text{received edges}\}$

up $U \leftarrow \{\text{edges previously sent to parent}(u)\}$

remove $R \leftarrow \{e \in K \setminus U : U \cup \{e\} \text{ contains a cycle}\}$

candidate $C \leftarrow K \setminus (U \cup R)$

if $C \neq \varnothing$ **then**

send $e \in C$ with minimum weight to parent

receive edges from children

else terminate

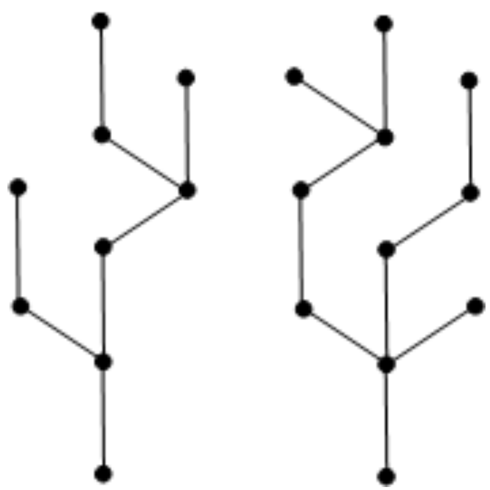
Proof of correctness

Theorem The Matroid algorithm performs in $O(n + D)$ rounds in the CONGEST model, and enables the root of the tree to construct a MST.

Lemma 0 Let A and B be acyclic subsets of edges. If $|A|>|B|$ then there exists $e \in A \setminus B$ such that $B \cup \{e\}$ is acyclic.

This is a matroid axiom

Proof   B is a forest $\{T_1,\ldots,T_k\}$. Let $n_i = |V(T_i)|$. We have $|E(T_i)|=n_i-1$.



For every i, there are at most n_i-1 edges of A connecting nodes in T_i.

➡ There is an edge in A whose extremities do not belong to a same tree T_i. ❑

A node u is said **active** at phase t if it has not terminated at phase $t-1$.

Let $h(u)$ = height of u = length of longest path to a leaf of the subtree T_u rooted at u.

Lemma 1 For every active child v of a node u, the set C of candidates for u at time t contains at least one edge sent by v to u before time t. ➪ **no premature termination**

Proof Induction on $h(u)$. Lemma holds for $h(u)=0$.

Assume lemma hold for all nodes at height $\leq k$.

Let u with $h(u)=k+1$, and v active child of u. Note $h(v) \leq k$.

E_u and E_v be edges sent by u to $p(u)$, and by v to $u=p(v)$ before phase t.

Since $h(v)<h(u)$ we have $|E_v| > |E_u|$.

By Lemma 0, $\exists\ e \in E_v \setminus E_u$ such that $E_u \cup \{e\}$ is acyclic ➡ $e \in C$. ❑

Lemma 2

(a) If u sends e to $p(u)$ at phase t then

1. all edges received by u at phase $t-1$ from its active children were of weight $\geq w(e)$, and
2. all edges to be received by u at phases $\geq t$ will be of weight $\geq w(e)$.

(b) The weights of the edges sent by u to its parent are ↗

→ it is legitimate to remove edges creating cycles with previously sent edges.

Proof True for height 0. Assume holds for height k.

(a.1) Let u with $h(u) = k+1$.

Let e' be edge sent by child v at phase t-1.

Let e'' ∈ C whose existence follows from Lemma 1.

By induction, property (b) implies $w(e'') \leq w(e')$.

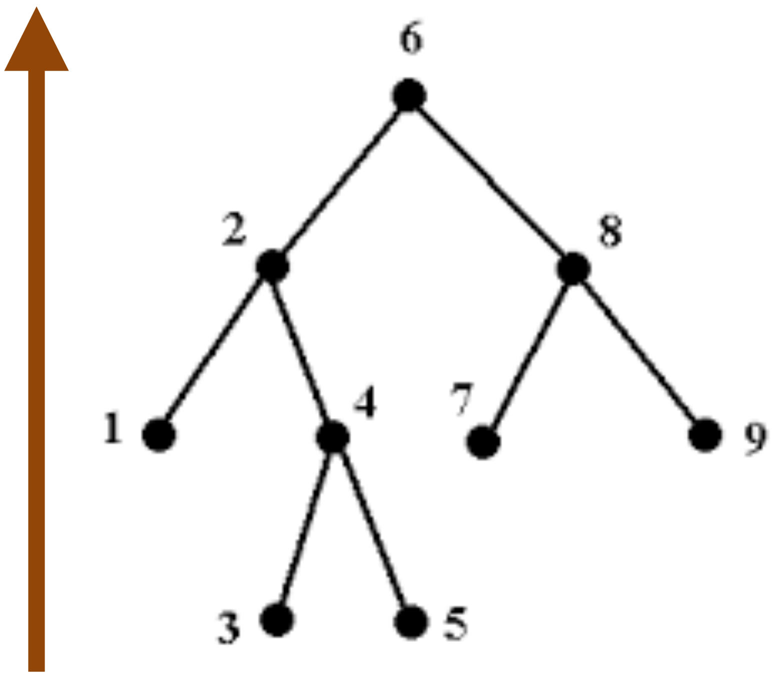
By the choice of the edge in C, we have $w(e) \leq w(e'')$.

➥ $w(e') \geq w(e)$.

(a.2) follows from (a.1) and by induction from (b).

(b) follows from (a.2) by the choice of the edge in C. ❑

Complexity

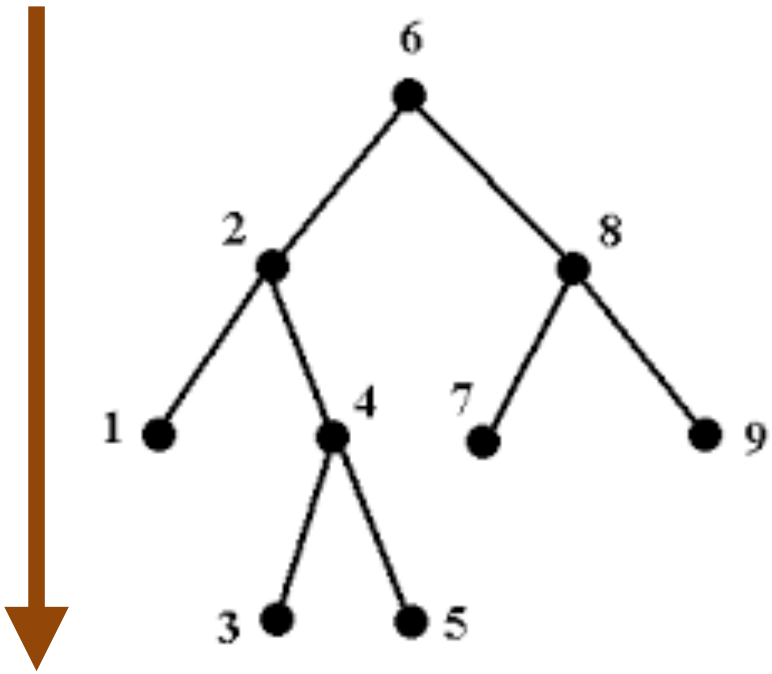


In n-node graphs, any set of n edges includes a cycle

➥ every node sends ≤ n-1 edges

➥ #rounds ≤ D + n - 1

Broadcasting the MST from the root to all nodes



Pipelining the edges of $T = \{e_1, e_2, \ldots, e_{n-1}\}$ down the BFS tree

➥ #rounds $\leq D + n - 1$

Wrap Up

- **Borůvska:** O(n log n) rounds — this is because fragments can have arbitrarily large diameter

- **Matroid:** O(D+n) rounds — this is because too many edges are gathered at a single node.

- **Combining Borůvska and Matroid:**

  - control the diameter of the fragment, and stops when fragments have too large diameter

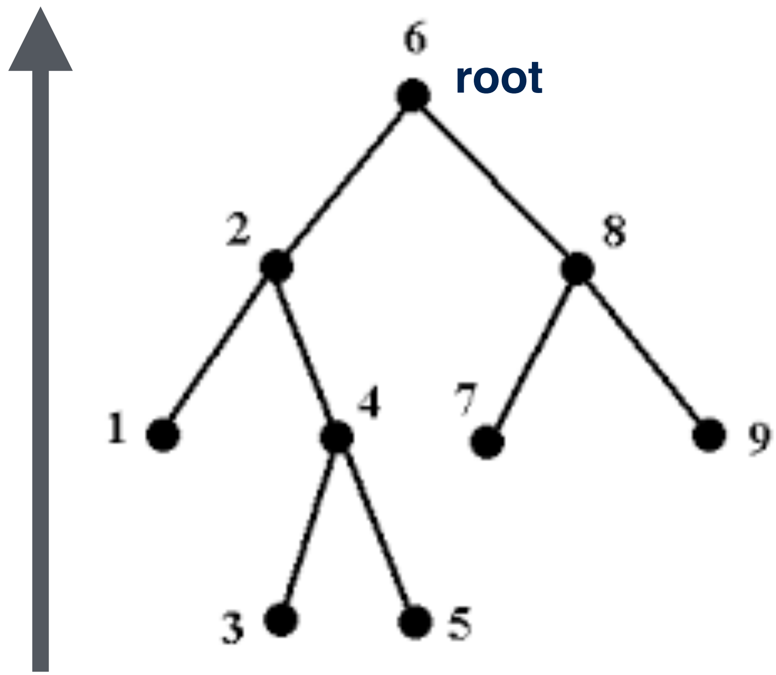
  - carry on with matroid for computing the (few) edges connecting the fragments already computed by Borůvska

Tool

- $D \subseteq V$ is a dominating set if every $u \notin D$ has a neighbor in D.
- Remarks:
  - Every maximal independent set is a dominating set.
  - Every tree has a dominating set of size $\leq n/2$
- **Objective:** Distributed computing of a dominating set of size $\leq n/2$ in consistently oriented trees.

MIS in Rooted Trees

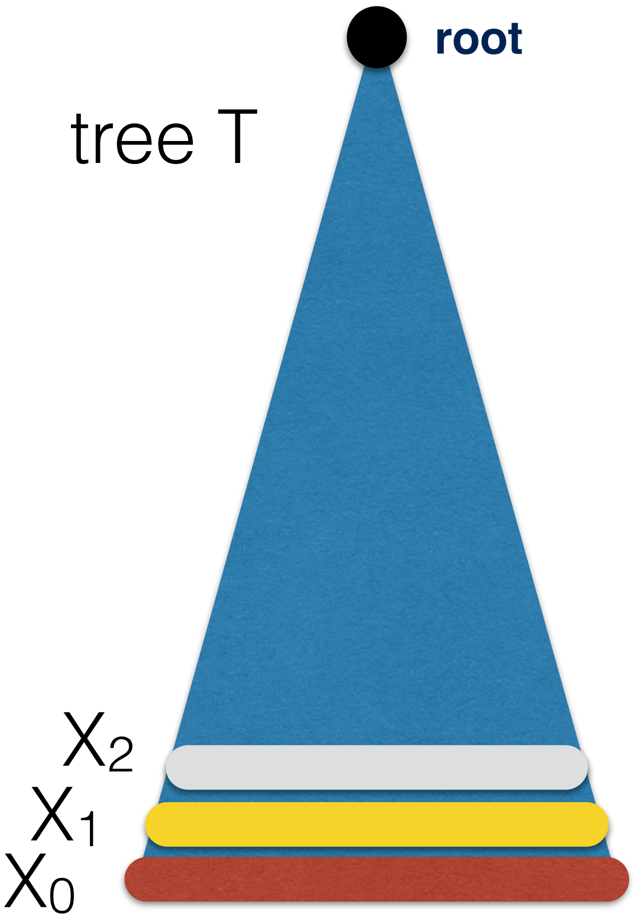
Every node has pointer to its parent



- Perform Cole and Vishkin algorithm with parent
- When colors are on 3 bits, every node pushes down its color
- Performs 5 rounds to get all colors in {1,2,3}.

Complexity : O(log*n) rounds

Computing small dominating sets in rooted trees

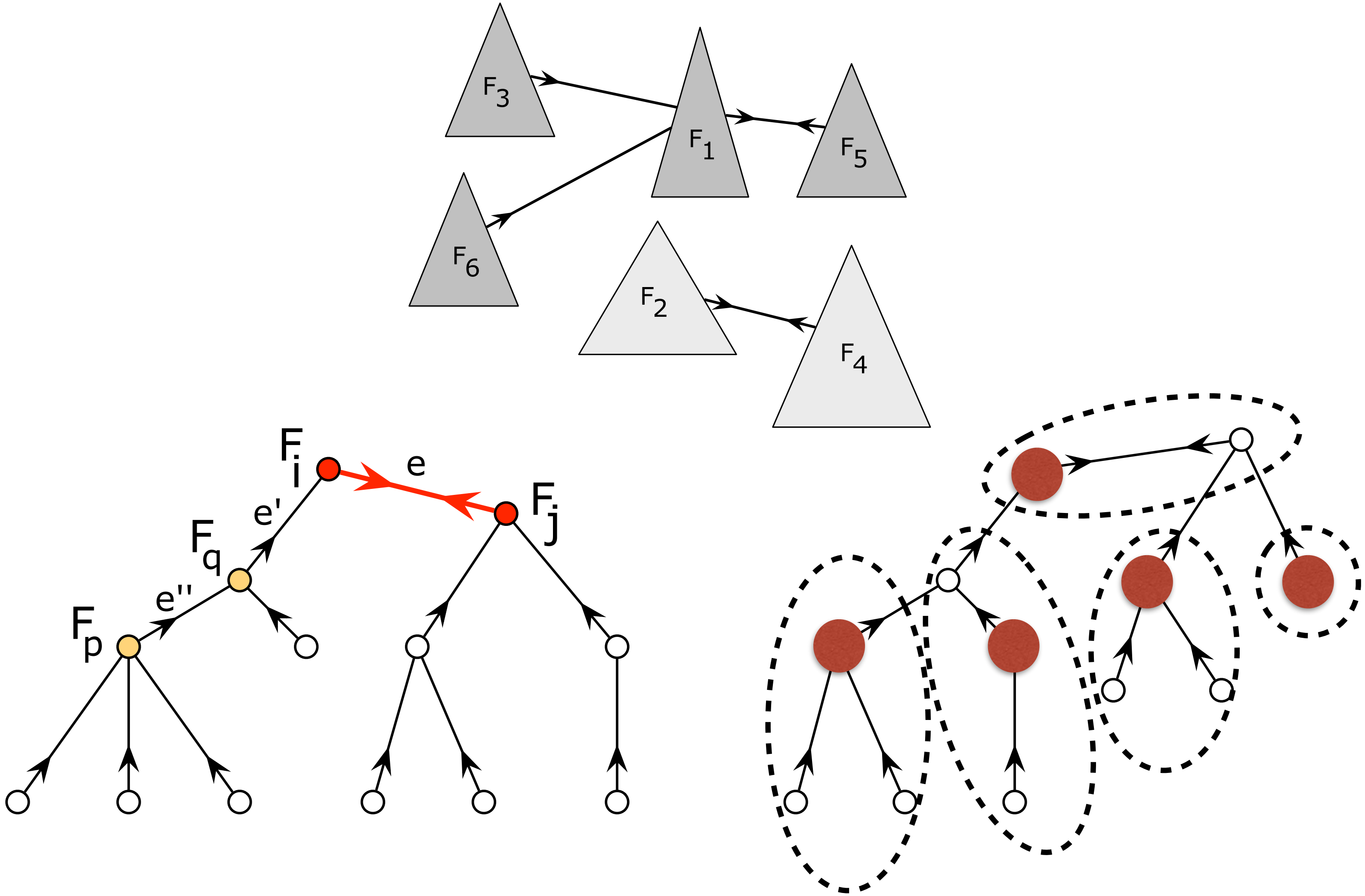


- X_d = {nodes at distance d from a leaf}
- $Y = V(T) \setminus (X_0 \cup X_1 \cup X_2)$
- Let J be MIS in Y (comput. in O(log*n) rounds)
- Let $D = J \cup X_1$

- D is a dominating set
- $|X_1| \leq |X_0| \Rightarrow |X_1| \leq ½ |X_0 \cup X_1|$
- $|J| \leq |(Y \cup X_2) \setminus J| \Rightarrow |J| \leq ½ |Y \cup X_2|$

➥ $|D| \leq n/2$

Bounding the diameter of fragments



Fast MST algorithm

Two stages:

1. Few phases of Borůvska
2. Completed by Matroid

$N(t) \leq N(t-1)/2$

$\Longrightarrow N(t) \leq n/2^t$

$\mathrm{diam}(t) \leq 3\ \mathrm{diam}(t-1) + 2$

$\Longrightarrow \mathrm{diam}(t) \leq 3^t - 1$

$N(t)$ = #frags after t phases

$\mathrm{diam}(t)$ = max diameter frags

Phase t costs

$O(\mathrm{diam}(t) \log^* n)$ rounds

τ phases Borůvska costs

$\tilde{O}(3^\tau)$ rounds

Matroid completes in

$O(D+N(\tau))$ rounds

$$3^\tau = n/2^\tau \Longrightarrow \#\text{rounds} = \tilde{O}(D + n^{0.6131})$$

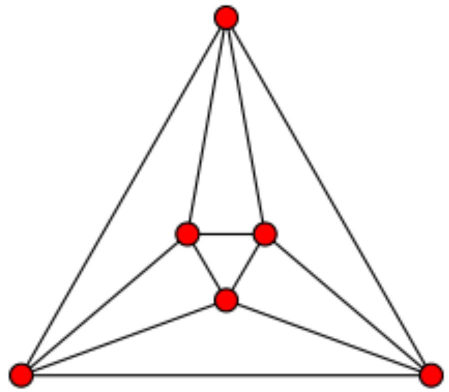
Theorem MST construction can be achieved in $\tilde{O}(D + \sqrt{n})$ rounds in the CONGEST model.

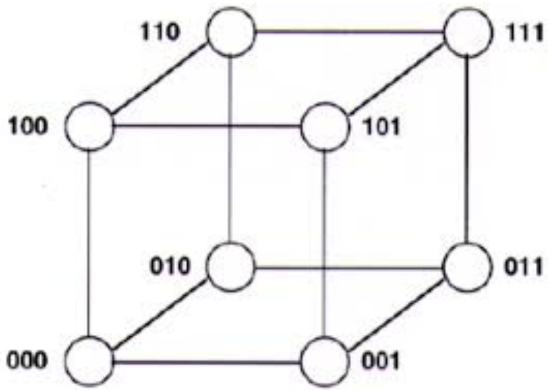
Local problems

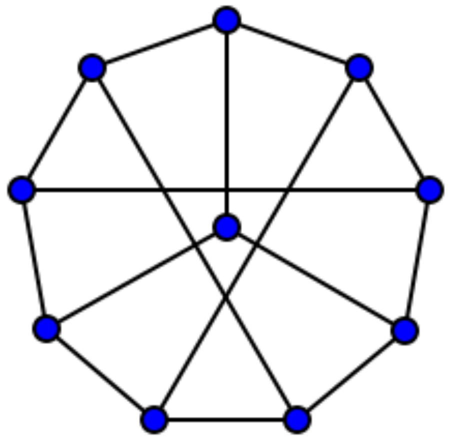
C_4-detection

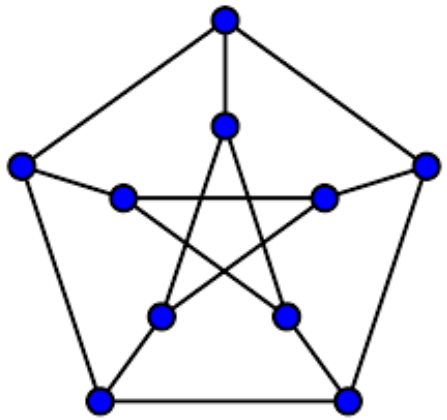
H is a subgraph of G if V(H) ⊆ V(G) and E(H) ⊆ E(G)

G is H-free if G does not contain H as a subgraph.









Distributed decision

A distributed algorithm A *decides* ϕ if and only if:

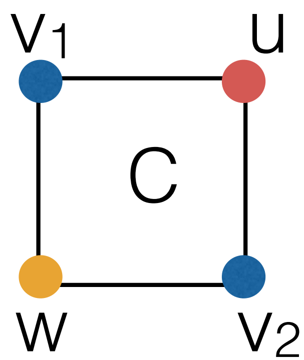
- $G \models \phi$ ⇒ all nodes output *accept*
- $G \not\models \phi$ ⇒ at least one node output *reject*

Theorem (Drucker, Kuhn & Oshman, 2014)
Deciding C_4-freeness can be done in $O(\sqrt{n})$ rounds.

Algorithm

Algorithm 3 C_4-detection executed by node u.

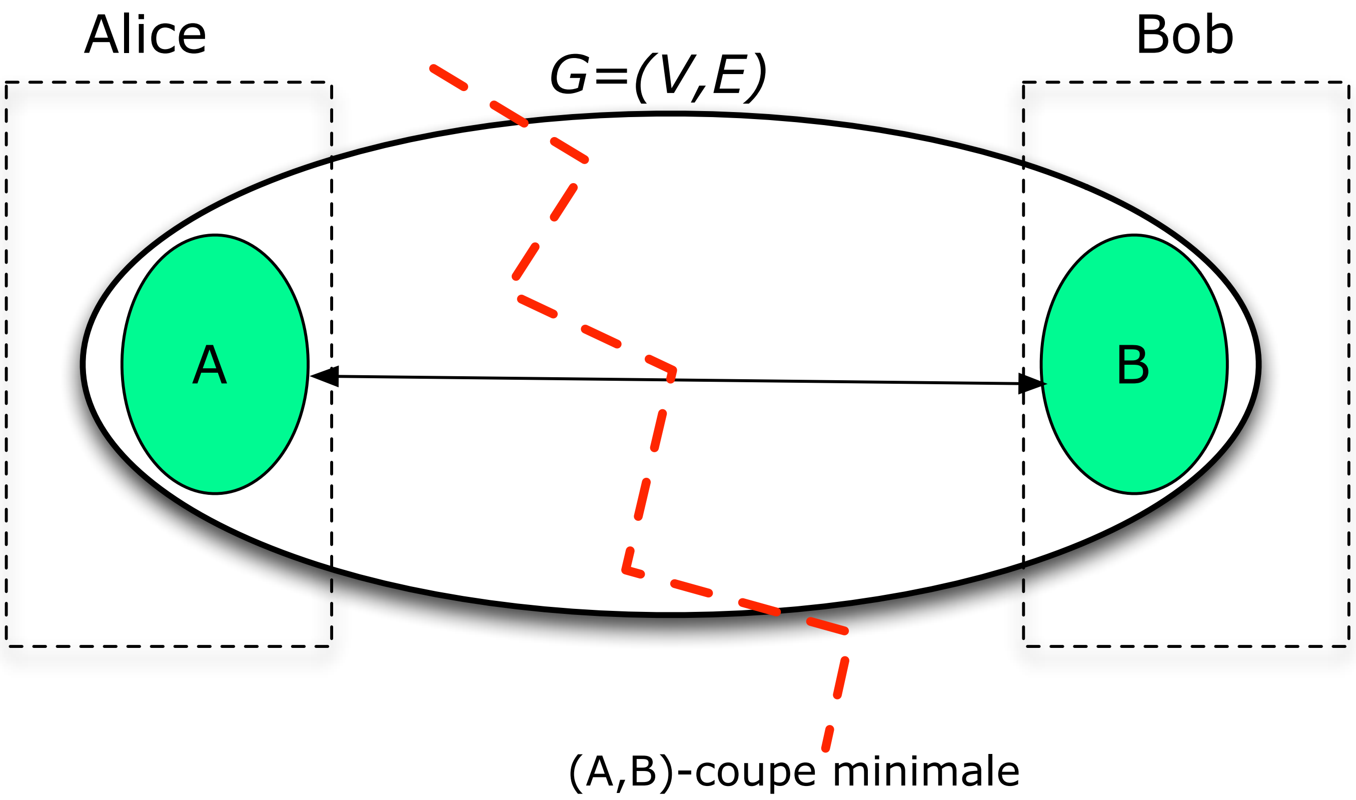
send $\mathrm{ID}(u)$ to all neighbors, and receive $\mathrm{ID}(v)$ from every neighbor v
send $\deg(u)$ to all neighbors, and receive $\deg(v)$ from every neighbor v
$S(u) \leftarrow \{$IDs of the $\min\{\sqrt{2n}, \deg(u)\}$ neighbors with largest degrees$\}$
send $S(u)$ to all neighbors, and receive $S(v)$ from every neighbor v
if $\sum_{v \in N(u)} \deg(v) \geq 2n + 1$ **then**
    output reject
else
    if $\exists v_1, v_2 \in N(u), \exists w \in S(v_1) \cap S(v_2) : w \neq u$ and $v_1 \neq v_2$ **then**
        output reject
    else
        output accept
    end if
end if



Case 1: there exists a 'large' node w in C
Case 2: all nodes of C are 'small'

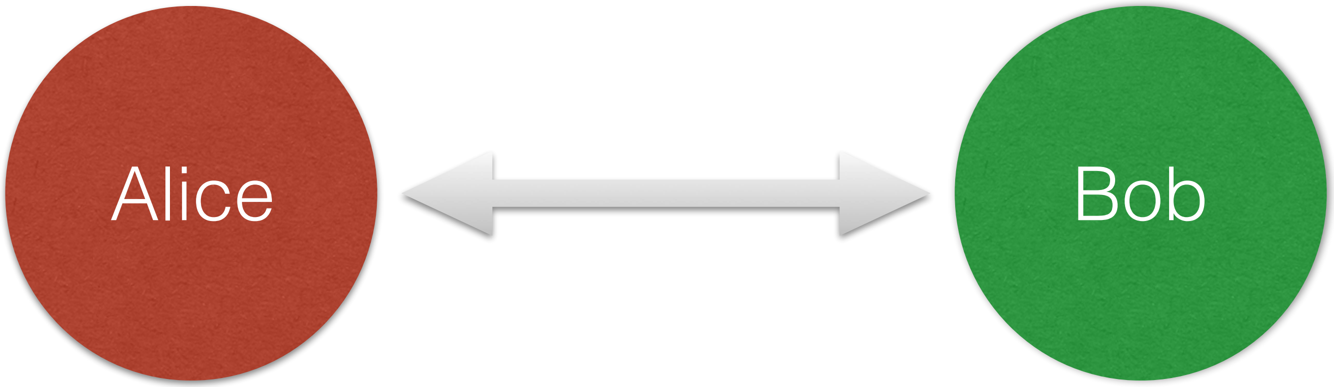
Lower bound techniques

Reduction to communication complexity



Communication complexity

$f : \{0,1\}^N \times \{0,1\}^N \rightarrow \{0,1\}$



$a \in \{0,1\}^N$ $b \in \{0,1\}^N$

Alice & Bob must compute $f(a,b)$

How many bits need to be exchanged between them?

Equality

- Alice gets $a \in \{0,1\}^N$, and Bob gets $a \in \{0,1\}^N$

$$f(a,b) = 1 \iff a = b$$

Theorem $CC(EQ) = \Omega(N)$.

Set-disjointness

- Ground set S of size N
- Alice gets $A \subseteq S$, and Bob gets $B \subseteq S$

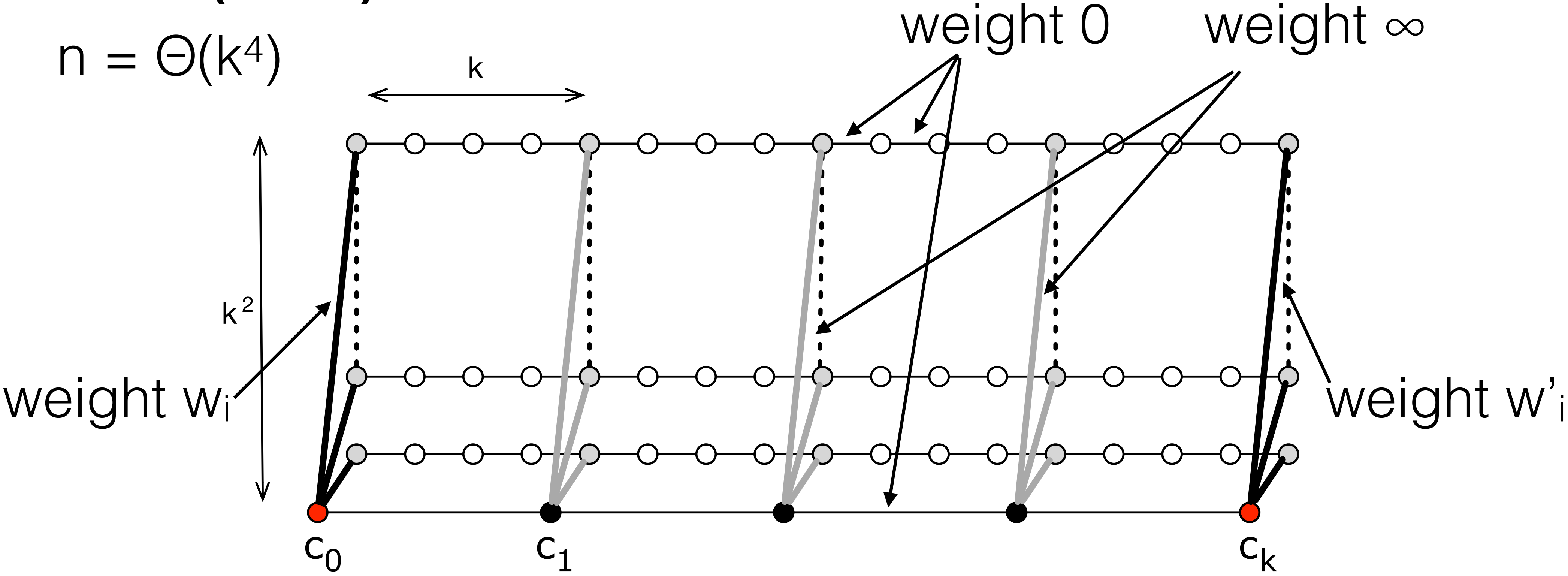
$$f(A,B) = 1 \iff A \cap B = \emptyset$$

Theorem $CC(DISJ) = \Omega(N)$, even using randomization (i.e., even if Alice and Bob have access to sources of random bits).

Application 1

$\Omega(\sqrt{n})$ lower bound for MST

$n = \Theta(k^4)$



$w_i = 2$ $\quad$ $w'_i = 2\,x_i + 1$ with $x_i \in \{0,1\}$

Lemma Transmitting k^2 bits from c_k to c_1 takes $\Omega(k^2)$ rounds

Proof (simplified: no recombination)

- $\exists\, i$, x_i uses $\leq k/2$ of highway ➪ $\Omega(k \cdot k/2)$ rounds
- $\forall\, i$, x_i uses $> k/2$ of highway ➪ $\Omega((k^2 \cdot k/2)/(k \log n))$ rounds ❑

Application 2

Deciding C_4-freeness

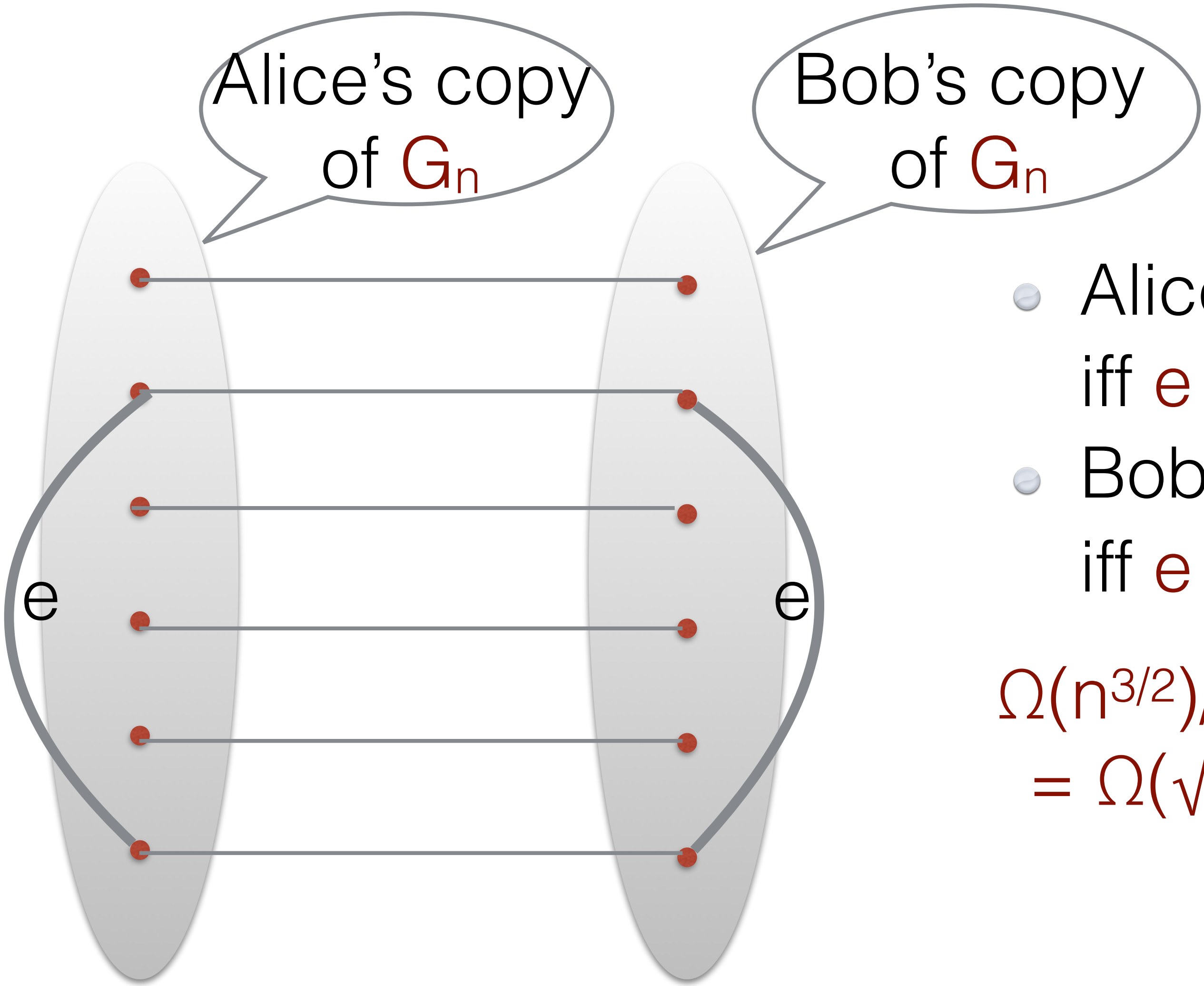
Theorem (Drucker, Kuhn & Oshman, 2014)
Deciding C_4-freeness required sending $\Omega(\sqrt{n}/\log n)$ bits between some neighbors

Reduction from Set-Disjointness.

Lemma There are C_4-free graphs G_n with n nodes and $m=\Omega(n^{3/2})$ edges.

Reduction

Let A and B as in set-disjointness with $N = m = \Omega(n^{3/2})$

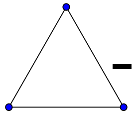


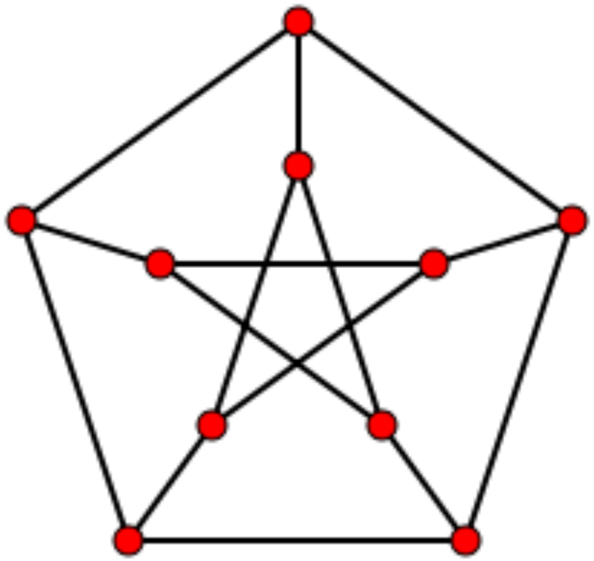
- Alice keeps $e \in E(G_n)$ iff $e \in A$
- Bob keeps $e \in E(G_n)$ iff $e \in B$

$\Omega(n^{3/2})/(n \log n)$
$= \Omega(\sqrt{n}/\log n)$

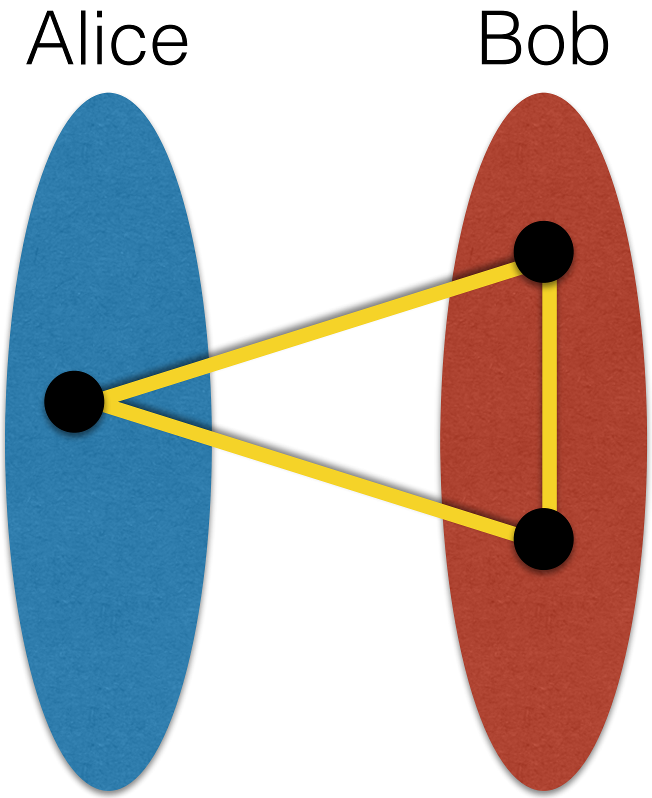
□

Open problem

deciding  -freeness



C_3-free graph



communication complexity fails

Distributed Property Testing

- **Property testing:** checking correctness of large data structure, by performing small (sub-linear) amount of queries.

- Graph queries (with nodes labeled from 1 to n):
  - what is degree of node x?
  - what is the i^{th} neighbor of node x?

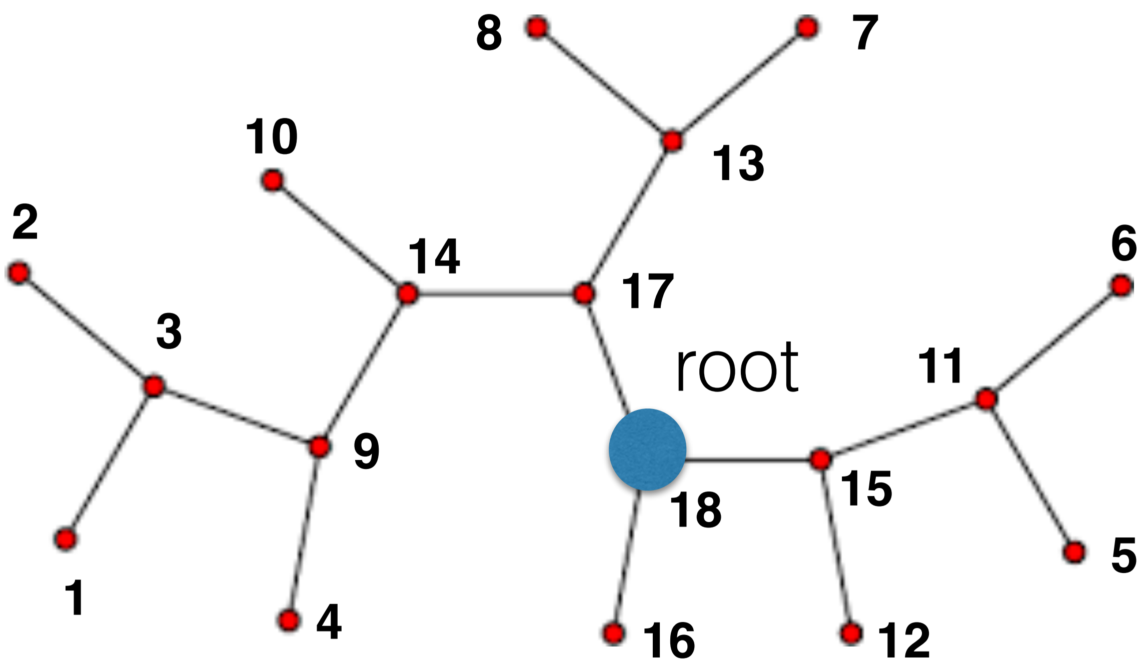
- Two relaxations:
  - G is ε-far from satisfying ϕ if removing/adding up to εm edges to/from G results in a graph which does not satisfy ϕ.
  - algorithm A tests ϕ if and only if:
    - $G \models \phi \Rightarrow$ Pr[all nodes output accept] $\geq$ ⅔
    - $G \not\models \phi \Rightarrow$ Pr[at least one node outputs reject] $\geq$ ⅔

Testing T-freeness

Theorem For every tree T, there exists a 1-sided error randomized algorithm performing in O(1) rounds in the CONGEST model, which correctly detects if the given input network contains T as a subgraph, with probability at least ⅔.

Color coding:

Let T be a k-node tree. Label the node from k down to 1, using BFS from arbitrary root.



Algorithm 1 Randomized tree-detection, for a given tree T. Algorithm executed by node u.

send $\mathrm{ID}(u)$ to all neighbors, and receive $\mathrm{ID}(v)$ from every neighbor v
let $k = |V(T)|$, and pick $\mathrm{color}(u) \in [k]$ uniformly at random
send $\mathrm{color}(u)$ to all neighbors, and receive $\mathrm{color}(v)$ from every neighbor v
for every $c \in [1,k]$, let $N_c(u) = \{v \in N(u) \mid \mathrm{color}(v) = c\}$
$\mathrm{active}(u) \leftarrow \mathrm{false}$
for $c = 1$ **to** k **do**
    send $\mathrm{active}(u)$ to all neighbors, and receive $\mathrm{active}(v)$ from every neighbor v
    compute $A(u) = \{v \in N(u) \mid \mathrm{active}(v) = \mathrm{true}\}$
    if $\mathrm{color}(u) = c$ **and** $\left(\forall c' \in \mathrm{child}(c),\ N_{c'}(u) \cap A(u) \neq \emptyset\right)$ **then**
        $\mathrm{active}(u) \leftarrow \mathrm{true}$
    end if
end for
if $\mathrm{color}(u) = k$ **and** $\mathrm{active}(u) = \mathrm{true}$ **then**
    output reject
else
    output accept
end if

Remark: does not use ε-farness.

Pr[detecting T] ≥ $(1/k)^k$

Perform $O(k^k)$ repetitions of Algorithm 1 to get prob[detecting T] ≥ ⅔

Testing C_3-freeness

Algorithm of node u

Exchange IDs with neighbors
for every neighbor v do
    pick a received ID u.a.r.
    send that ID to v
if u receives ID(w) from $v \in N(u)$ with $w \in N(u)$ and $v \neq w$
then output reject
else output accept

Lemma 1 For any triangle Δ, $\Pr[\Delta \text{ is detected}] \geq 1/n$

Analysis

Theorem Let ε ∈]0,1[. If G is ε-far from being C_3-free, then the algorithm detects a cycle with prob ≥ $1-(1/e)^{\varepsilon/3}$

Lemma 2 If G is ε-far from being C_3-free, then G contains at least εm/3 edge-disjoint triangles.

Proof Let $S=\{e_1,e_2,\ldots,e_k\}$ be min #edges to remove for making G triangle-free ($k \geq \varepsilon m$).

Repeat removing e from S, as well as all edges of a triangle Δ_e containing e ➡ at least k/3 steps.

All triangles Δ_e are edge-disjoint. ❏

Analysis (continued)

Proof (of theorem)

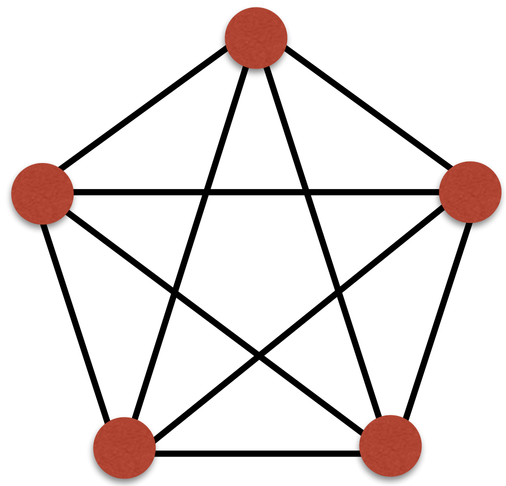
- Pr[no Δ detected] $\leq (1-1/n)^{\varepsilon m/3} \leq (1-1/n)^{\varepsilon n/3}$
- $(1-1/n)^n \approx 1/e$
- Pr[no Δ detected] $\leq (1/e)^{\varepsilon/3}$ ❏

Repeat k times with k such that $(1/e)^{\varepsilon k/3} \leq 1/3$

That is $k \geq 3 \ln(3) / \varepsilon$ ➡ #rounds = $O(1/\varepsilon)$.

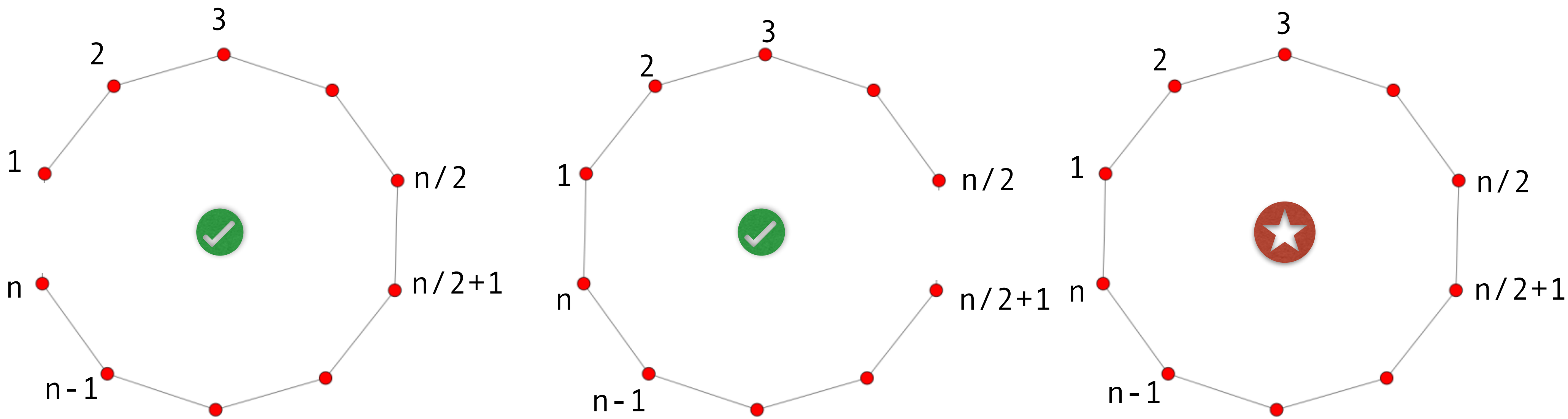
Open problem

Is there a distributed tester for K_5-freeness running in O(1) rounds in the CONGEST model?



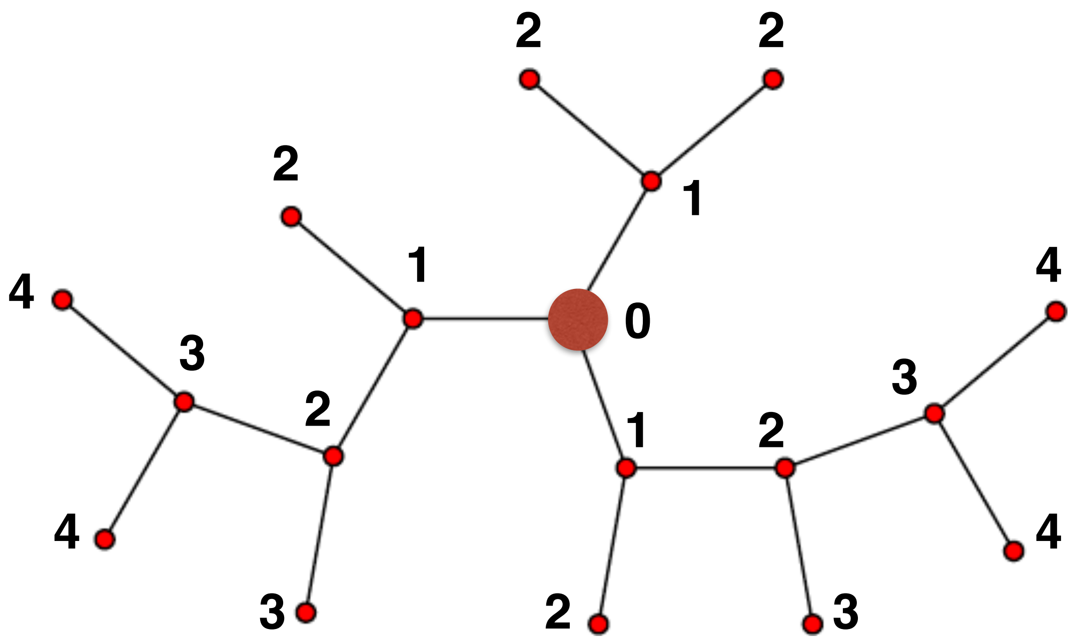
Distributed Verification

Acyclicness



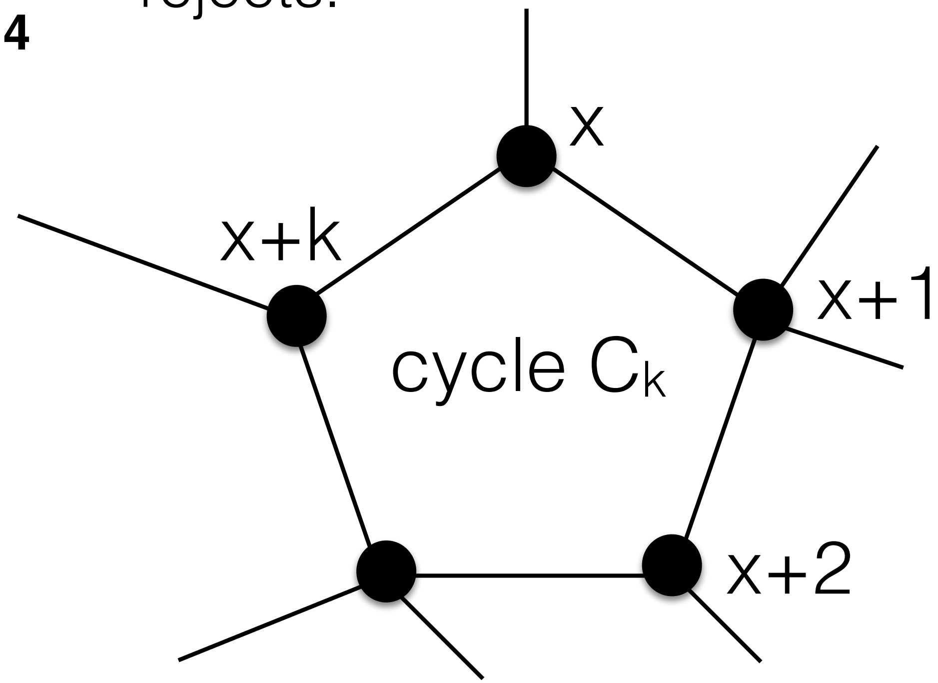
Non locally decidable!

Acyclicness



if G is acyclic, then there is an assignment of the counter resulting in all nodes accept.

if G is has a cycle, then for every assignment of the counters, at least one node rejects.



Algorithm of node u

exchange counters with neighbors
if ∃! v∈N(u) : cpt(v)=cpt(u)-1 and
∀ w∈N(u)∖{v}, cpt(w)=cpt(u)+1
then accept
else reject

Proof-Labeling Scheme

A distributed algorithm A *verifies* ϕ if and only if:

- $G \models \phi \Rightarrow \exists\, c: V(G) \to \{0,1\}^*$: all nodes accept (G,c)
- $G \not\models \phi \Rightarrow \forall\, c: V(G) \to \{0,1\}^*$ at least one node rejects (G,c)

The bit-string c(u) is called the *certificate* for u (cf. class NP)

Objective: Algorithms in O(1) rounds (ideally, just 1 round in LOCAL)

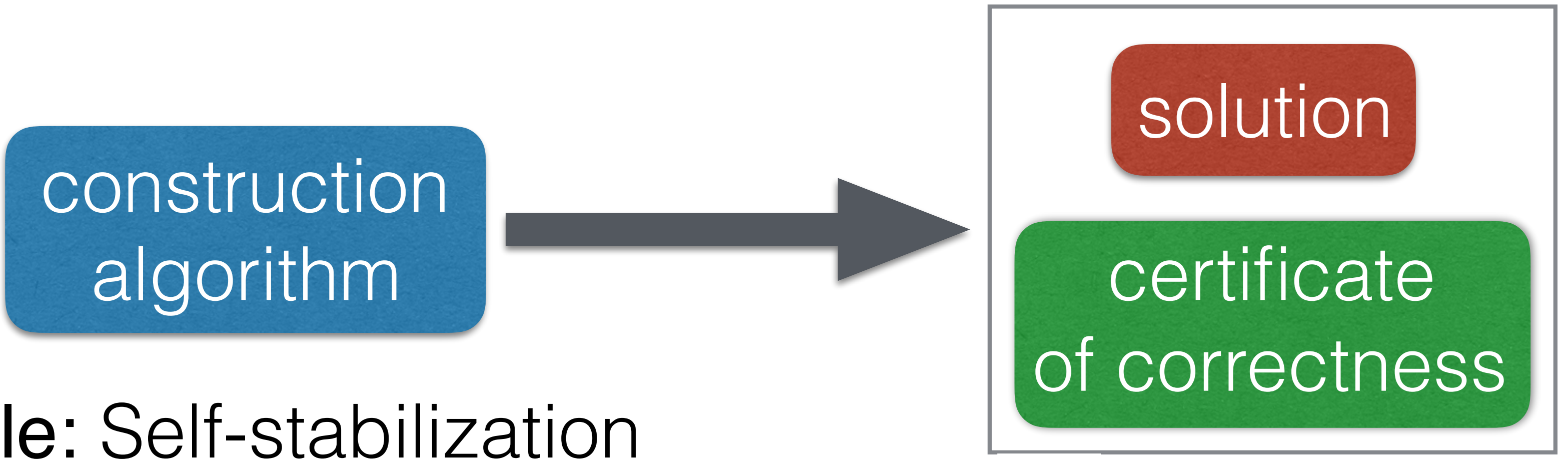
Examples:

O(log n) bits

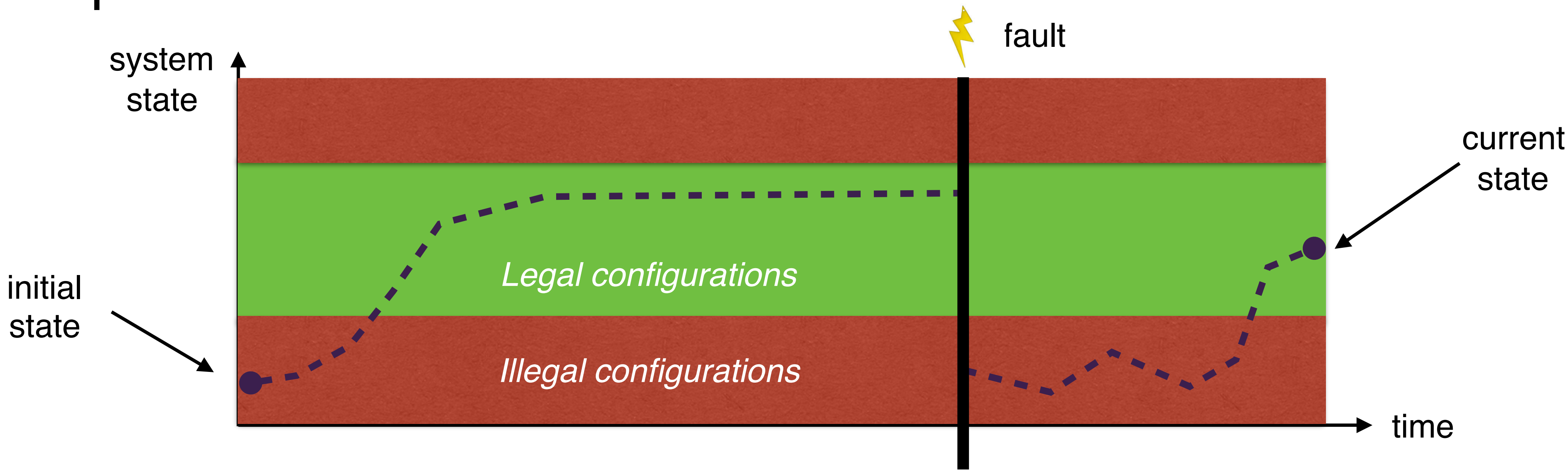
- Acyclicness: $c(u) = dist_G(u,r)$
- Spanning tree: $c(u) = (dist_G(u,r), ID(r))$

Measure of complexity: $\max_{u \in V(G)} |c(u)|$

Application: Fault-Tolerance



Example: Self-stabilization



Universal PLS

Theorem For any (decidable) graph property ϕ, there exists a PLS for ϕ, with certificates of size $O(n^2)$ bits in n-node graphs.

Proof c(u) = (M,x) where

- M = adjacency matrix of G
- x = table[1..n] with x(i) = ID(node with index i)

Verification algorithm:

1. check local consistency of M using x
2. if no inconsistencies, check whether M satisfies ϕ

G satisfies $\overset{\text{exercice}}{\Longleftrightarrow}$ both tests are passed ❏

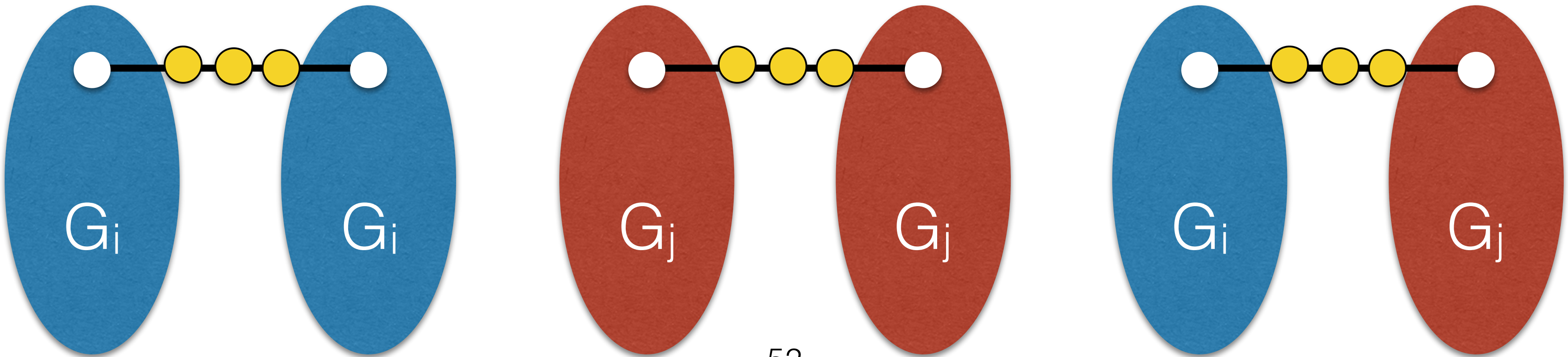
Lower bound

Theorem There exists a graph property for which any PLS has certificates of size $\Omega(n^2)$ bits.

Proof Graph automorphism = bijection $f:V(G) \to V(G)$ such that $\{u,v\} \in E(G) \iff \{f(u),f(v)\} \in E(G)$

Fact There are $\geq 2^{\varepsilon\, n^2}$ graphs with no non-trivial auto.

If certificates on $< \varepsilon n^2/3$ bits, then $\exists$ $i \neq j$ such that the three nodes ●●● have same certificates on G_i-G_i and G_j-G_j.



❑

Local hierarchy

- Equivalent of, e.g., polynomial hierarchy in complexity theory
- {locally decidable properties} = $\Sigma_0 = \Pi_0$
- {locally verifiable properties (with PLS)} = Σ_1

Deciding graph property ϕ is in Σ_1 if and only if:

- $G \models \phi \Rightarrow \exists\, c$ all nodes accept (G,c)
- $G \not\models \phi \Rightarrow \forall\, c$ at least one node rejects (G,c)

Deciding graph property ϕ is in Π_1 if and only if:

- $G \models \phi \Rightarrow \forall\, c$ all nodes accept (G,c)
- $G \not\models \phi \Rightarrow \exists\, c$ at least one node rejects (G,c)

The hierarchy $(\Sigma_k, \Pi_k)_{k \geq 0}$

Deciding graph property ϕ is in Σ_2 if and only if:

- $G \models \phi \;\Rightarrow\; \exists\, c_1\; \forall\, c_2$ all nodes accept (G, c_1, c_2)
- $G \not\models \phi \;\Rightarrow\; \forall\, c_1\; \exists\, c_2$ at least one node rejects (G, c_1, c_2)

Deciding graph property ϕ is in Π_2 if and only if:

- $G \models \phi \;\Rightarrow\; \forall\, c_1\; \exists\, c_2$ all nodes accept (G, c_1, c_2)
- $G \not\models \phi \;\Rightarrow\; \exists\, c_1\; \forall\, c_2$ at least one node rejects (G, c_1, c_2)

Deciding graph property ϕ is in Σ_k if and only if:

- $G \models \phi \;\Rightarrow\; \exists\, c_1\; \forall\, c_2\; \exists\, c_3 \ldots Q\, c_k$ all nodes accept $(G, c_1, \ldots, c_k)$
- $G \not\models \phi \;\Rightarrow\; \forall\, c_1\; \exists\, c_2\; \forall\, c_1 \ldots \neg Q\, c_k$ at least one node rejects $(G, c_1, \ldots, c_k)$

Deciding graph property ϕ is in Π_k if and only if:

- $G \models \phi \;\Rightarrow\; \forall\, c_1\; \exists\, c_2\; \forall\, c_3 \ldots Q\, c_k$ all nodes accept $(G, c_1, \ldots, c_k)$
- $G \not\models \phi \;\Rightarrow\; \exists\, c_1\; \forall\, c_2\; \exists\, c_3 \ldots \neg Q\, c_k$ at least one node rejects $(G, c_1, \ldots, c_k)$

Example: Minimum Dominating Set

Decision problem MinDS:

- input = dominating set $\mathcal{D}$ (i.e., $\mathcal{D}(u)\in\{0,1\}$)
- output = accept if $|\mathcal{D}| = \min_{\text{dom } D} |D|$

Theorem MinDS $\in \Pi_2$

Proof

c_1 encodes a dominating set, i.e., $c_1(u)\in\{0,1\}$

c_2 encodes:

- a spanning tree T_{err} pointing to node u with error in c_1 if any
- a spanning tree T_0 for counting $|\mathcal{D}|$ (w/ same root)
- a spanning tree T_1 for counting $|c_1|$ (w/ same root)

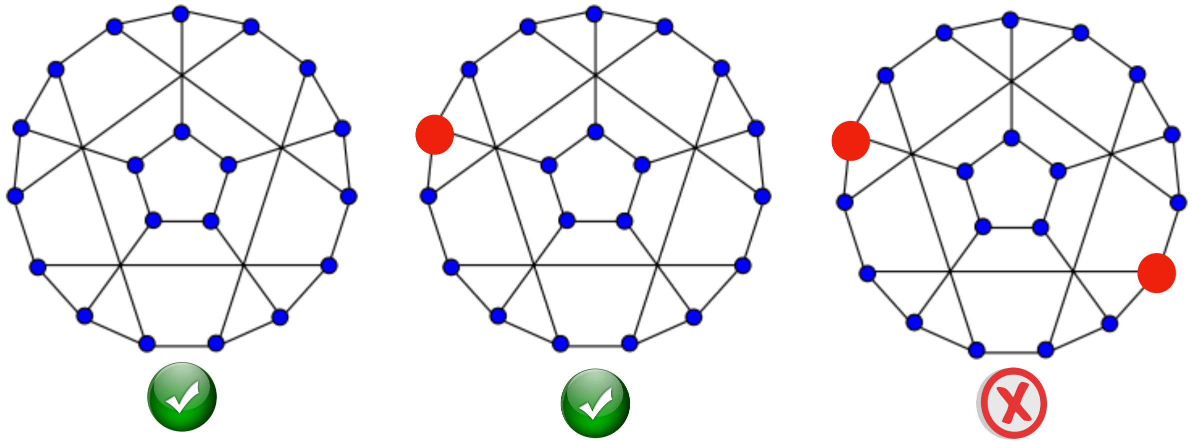
Algorithm:

- If root u ses $|c_1| < |\mathcal{D}|$ with no error, it rejects, otherwise it accepts
- If any node detects inconsistencies in T_0, T_1 or T_{err} it rejects, otherwise it accepts. ❑

Randomized Protocols

[FKP, 2013]

- At most one selected (AMOS)

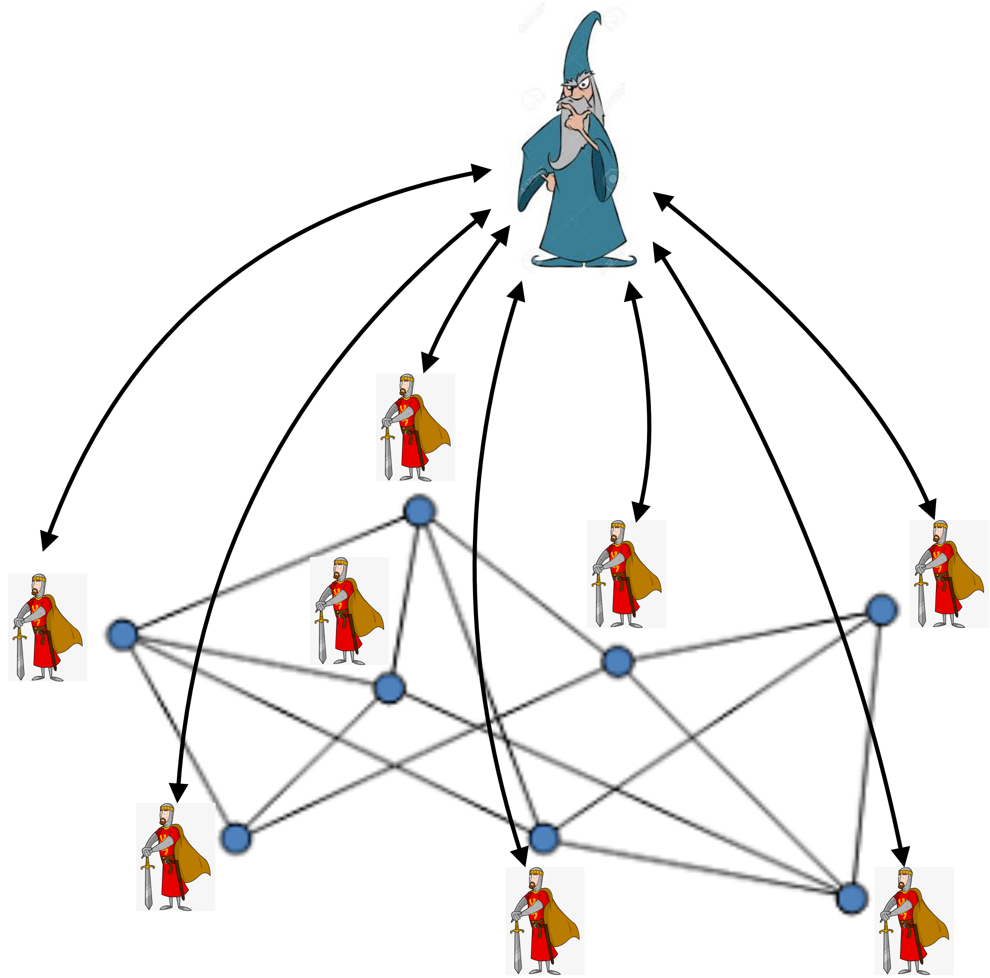


- Decision algorithm (2-sided):
  - let $p = (\sqrt{5}-1)/2 = 0.61\ldots$
  - If not selected then accept
  - If selected then accept w/ prob p, and reject w/ prob 1-p

- Issue with boosting! — But OK for 1-sided error

Distributed Interactive Protocols

[KOS, 2018]

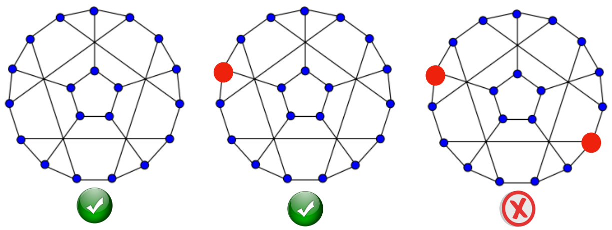


- Arthur-Merlin Phase *(no communication, only interactions)*
- Verification Phase *(only communications)*

- Merlin has infinite communication power
- Arthur is randomized

- k = #interactions
- dAM[k] or dMA[k]

Example: AMOS



- In BPLD with success prob $(\sqrt{5}-1)/2 = 0.61\ldots$
- In Σ_1LD(O(log n)) — Not in Σ_1LD(o(log n))
- Not in dMA(o(log n)) for success prob > 4/5
- In dAM(k) with k random bits, and success prob $1\text{-}1/2^k$
  - Arthur independently picks a k-bit index at each node u.a.r.
  - Merlin answer $\perp$ if no nodes selected, or the index of the selected node

Sequential setting

- For every $k \geq 2$, $AM[k] = AM$
- $MA \subseteq AM$ because $MA \subseteq MAM = AM[3] = AM$
- $MA \in \Sigma_2 P \cap \Pi_2 P$
- $AM \in \Pi_2 P$
- $AM[\mathit{poly}(n)] = IP = PSPACE$

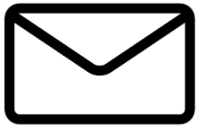
Known results

[KOS 2018, NPY 2018]

- Sym $\in$ dAM($n \log n$)
- Sym $\in$ dMAM($\log n$)
- Any dAM protocol for Sym requires $\Omega(\log\log n)$-bit certificates
- $\neg$Sym $\in$ dAMAM($\log n$)
- Other results on graph non-isomorphism

Parameters

- Number of interactions between  and 
- Size of 

- Size of 
- Number of random 
- Shared vs distributed 

Tradeoffs

[CFP, 2019]

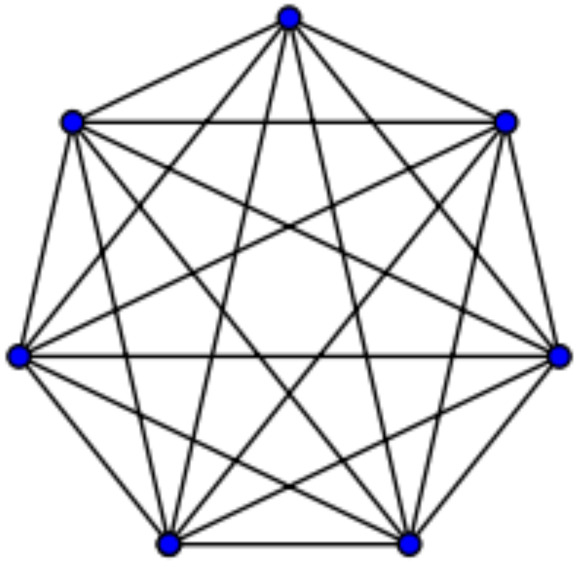
- **Theorem 1** For every c, there exists a Merlin-Arthur (dMA) protocol for *triangle-freeness*, using $O(\log n)$ bits of shared randomness, with $\tilde{O}(n/c)$-bit certificates and $\tilde{O}(c)$-bit messages between nodes.

- **Theorem 2** There exists a graph property admitting a proof-labeling scheme with certificates and messages on $O(n)$ bits, that cannot be solved by an Arthur-Merlin (dAM) protocol with certificates on $o(n)$ bits, for any fixed number $k \geq 0$ of interactions between Arthur and Merlin, even using shared randomness, and even with messages of unbounded size.

Congested Clique

Definition

CONGEST model, but on a clique!



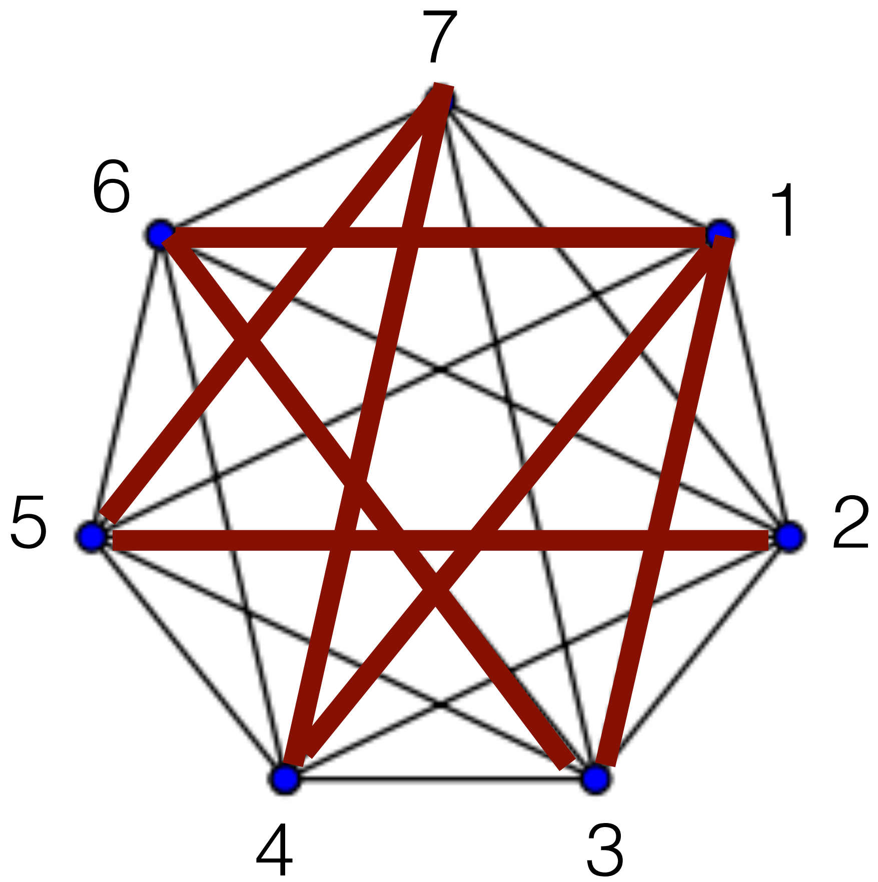
Unicast variant

potentially
different
messages

Broadcast variant

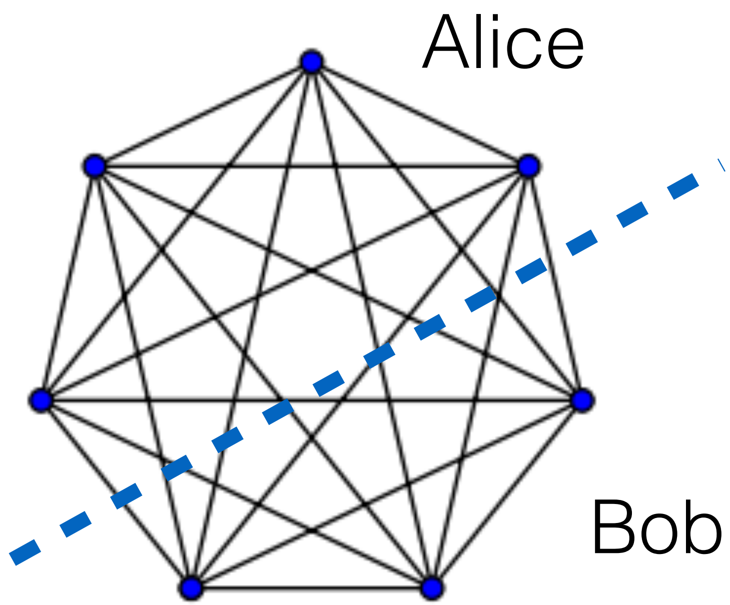
same
messages

Graph Problems in the Congested Clique



Lower bound in the Broadcast Congested Clique

Theorem (Drucker, Kuhn & Oshman, 2014)
Deciding C_4-freeness required sending $\Omega(\sqrt{n})$ bits between some neighbors in the Broadcast Congested Clique.



$\Theta(n^2)$ links but bandwidth $\Theta(n \log n)$

Lower bound in the Unicast Congested Clique

To date, no lower bounds for this model are known…

Theorem (*informal* - Drucker, Kuhn & Oshman, 2014))
The unicast congested clique can « *simulate* »
« *powerful* » classes of bounded-depth circuits.

It follows that even slightly super-constant lower bounds for the unicast congested clique would give new lower bounds in circuit complexity.

Concluding remarks

Open problems

- **Lower bound** for the congested clique (hard!) — first step: broadcast congested clique.
- Ability to **solve local problems** (e.g., triangle detection) in the CONGEST model.
- Practical approach: how do the known results **scale** with the bandwidth B of the link?
- Congest algorithms for **dynamic networks**.

Time vs. Space