Distributed Property Testing

Pierre Fraigniaud
CNRS and University Paris Diderot
IRIF
INSTITUT
DE RECHERCHE
EN INFORMATIQUE
FONDAMENTALE

ESTATE/VERIMAG Workshop on Distributed Algorithms
Grenoble, France, August 31st, 2017

Grenoble INP
ensimag

Property Testing
(for graphs)

Objective: distinguish between graphs satisfying a given property P from graphs that are far from satisfying P.

ε-farness:

- Dense model: add/remove $\geq \varepsilon n^2$ edges to satisfy P
- Bounded-degree model: add/remove $\geq \varepsilon dn$ edges to satisfy P
- **Sparse model:** add/remove $\geq \varepsilon m$ edges to satisfy P

Sequential Tester

Performs queries to nodes (labeled from 1 to n)

- what is the degree of node v?
- what is the ID of the i^{th} neighbor of node v?

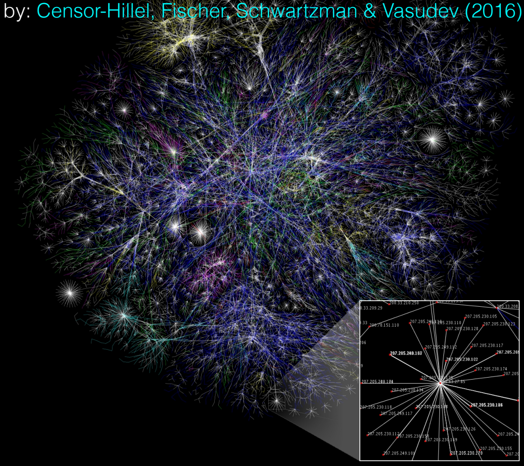
Objective: After o(n) queries, decide whether G satisfies P or not, in poly(n) time.

Typical Decision Rule

- If G satisfies P then Pr[accept] ≥ ⅔
- If G is ε-far from satisfying P then Pr[reject] ≥ ⅔

Distributed Property Testing

Introduced by: Brakerski & Patt-Shamir (2011)
Specified by: Censor-Hillel, Fischer, Schwartzman & Vasudev (2016)



(1) from wikipedia

Distributed Decision Rule

- If G satisfies P then

  Pr[all nodes accept] ≥ ⅔

- If G is ε-far from satisfying P then

  Pr[at least one node rejects] ≥ ⅔

CONGEST Model

- Nodes have IDs in a range $[1,n^c]$
- All nodes start simultaneously
- They perform is synchronous rounds
- Each round consists, for every node:
  - sending a message to each neighbor
  - receiving the message from each neighbor
  - computing, i.e., performing individual computation

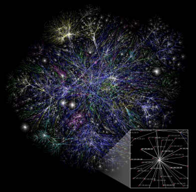
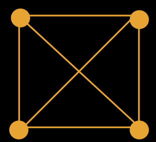
messages of O(log n) bits

Objective

Test whether G satisfies P in the least number of rounds, ideally O(1) rounds.

Example:

H-freeness: does G contains H as a subgraph?

G  H 

Distributed Decision
(Lower Bound)

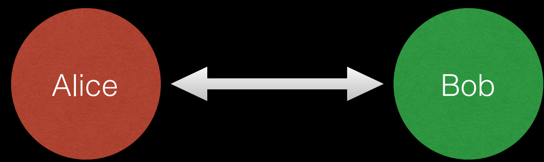
Why ε-farness?
Why randomization?

Theorem [Drucker, Kuhn, Oshman (2014)]
Deciding C_4-freeness requires $\Omega(\sqrt{n})$ rounds, even using randomization.

Proof Reduction from set-disjointness in the context of 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?

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 [Håstad & Wigderson (2007)]

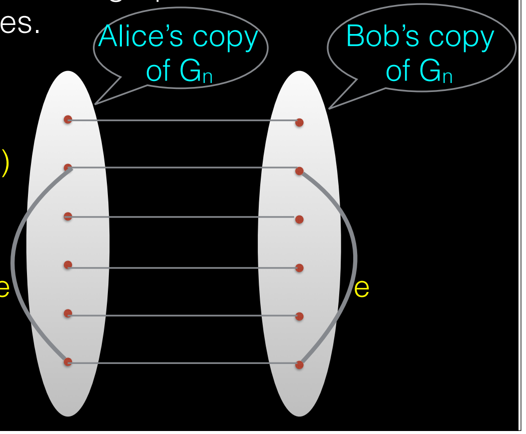
$CC(f) = \Omega(N)$, even using randomization.

Reduction from Set-Disjointness

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

Let A and B as in set-disjointness (N=m)

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

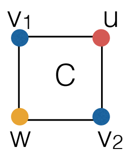


$\Omega(n^{3/2})/n = \Omega(\sqrt{n})$

The bound is tight

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'

Distributed Decision
(Upper Bound)

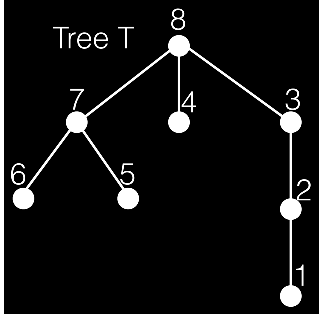
Deciding Tree-Freeness

Theorem [F., Montealegre, Olivetti, Rapaport, Todinca (2017)]
Let T be a tree. There is a deterministic algorithm deciding T-freeness in O(1) rounds.

Remarks

- ✓ no need of the ε-farness assumption.
- ✓ no need of randomization
- ✓ the big-O depends on k=|T| ☞ k^k rounds

A Simple Randomized Algorithm (color-coding technique)

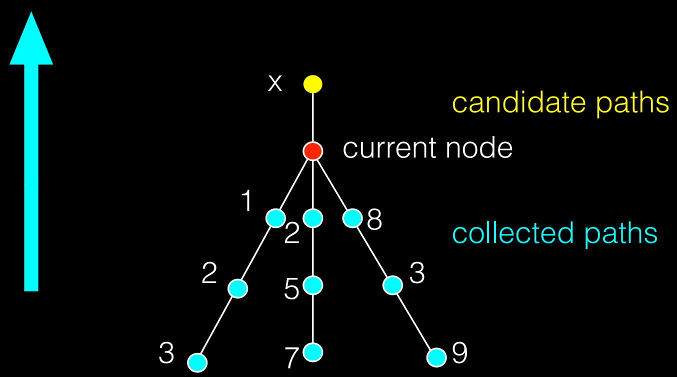


Algorithm

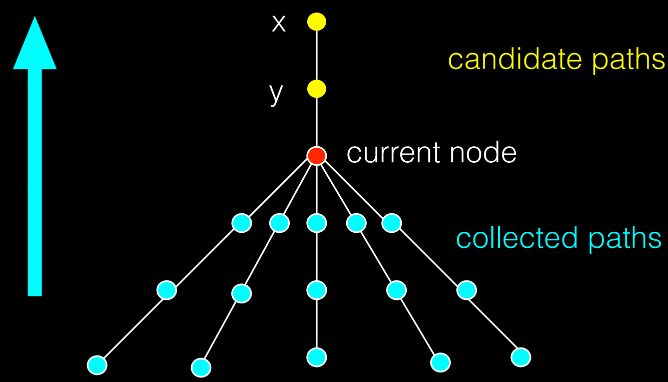
pick col(v)∊{1,2,…,k} u.a.r.
active←false
for k=1 to |T| do
  if col(v)=k and exist well colored set of active neighbors then
    active←true
if col(v)=k and active then reject
else accept

Pr[tree T is detected] ≥ $1/k^k$

Deterministic Algorithm Example: path (1)



Deterministic Algorithm Example: path (2)



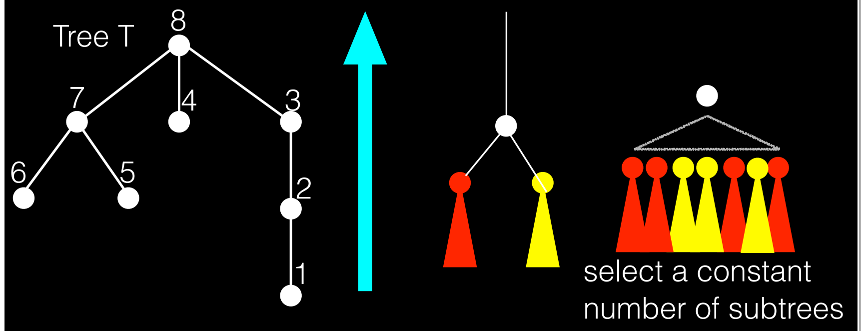
Pruning technique

Definition Let $n>k>t$. Let V be a set of size n, and F a collection of subsets of V with cardinality $\leq t$. A witness of F is a set $F' \subseteq F$ such that, for any $X \subseteq V$ with $|X| \leq k-t$, the following holds:

$$\exists\, Y \in F : X \cap Y = \varnothing \;\Rightarrow\; \exists\, Y' \in F' : X \cap Y' = \varnothing$$

Lemma [Erdős, Hajnal, Moon] There exists a compact witness of F, i.e., a witness of F with cardinality independent of n.

The Deterministic "Pruning Algorithm"



- Same type of pruning as for paths
- Must take unto account the shapes of the subtrees

Remark: individual time-complexity exponential in $|T|$.

Application to distributed property testing

Testing H-freeness for a large class of graphs H

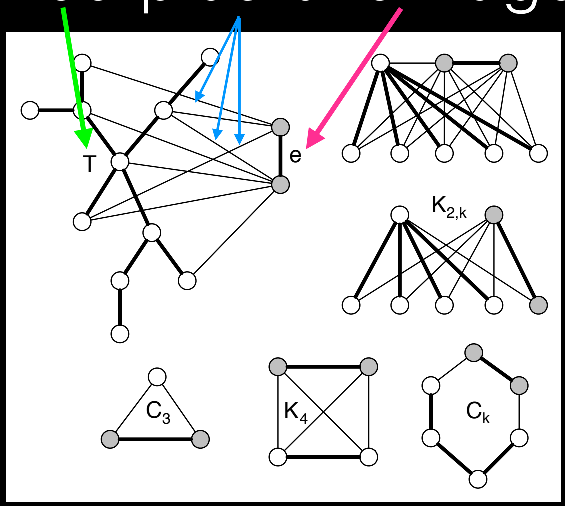
Sparse model: add/remove $\geq \varepsilon m$ edges to satisfy P

Theorem [F., Montealegre, Olivetti, Rapaport, Todinca (2017)] Let H be a tree-plus-one-edge. There is a distributed tester for H-freeness running in $O(1)$ rounds.

Remark the big-O depends on $k=|H|$ and ε

☞ k^k/ε rounds

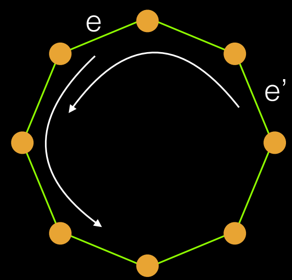
Tree-plus-one-Edge



Generalizes to Forest-plus-one-edge

Algorithm

- Each edge picks a rank in [1,m^2] u.a.r.
- The edge with minimum rank is used as an 'anchor' for the search for T
- Discard competing searches from high rank edges



Corollaries

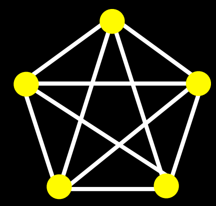
- C_k is a tree-plus-one-edge, for any $k \geq 3$.
  - ➡ C_3 [Censor-Hillel, Fischer, Schwartzman & Vasudev (2016)]
  - ➡ C_4 [F., Rapaport, Salo & Todinca (2016)]
  - ➡ C_k [F. & Olivetti (2017)]
- K_k is a tree-plus-one-edge, for any $k \leq 4$.
  - ➡ K_4 [F., Rapaport, Salo & Todinca (2016)]

DISC 2017 Notice

- Orr Fischer, Tzlil Gonen and Rotem Oshman. Distributed Property Testing for Subgraph-Freeness Revisited
- Pierre Fraigniaud, Pedro Montealegre, Dennis Olivetti, Ivan Rapaport and Ioan Todinca. Distributed Subgraph Detection
- Guy Even, Reut Levi and Moti Medina. Faster and Simpler Distributed Algorithms for Testing and Correcting Graph Properties in the CONGEST Model

Conclusion

Open problems



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

(2) Characterization of graph patterns H for which H-freeness can be tested in $O(1)$ rounds?