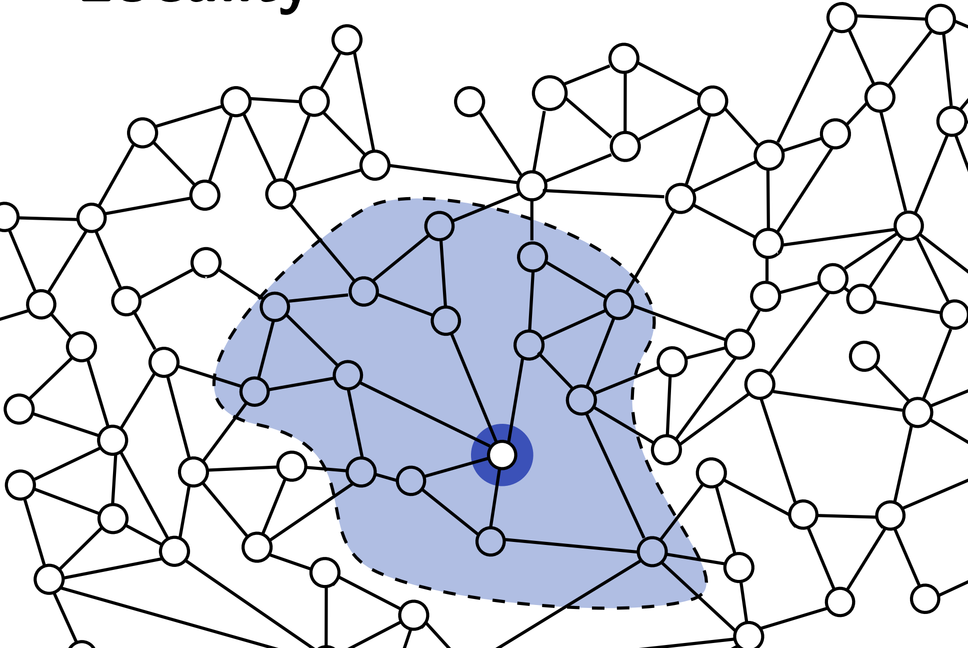
Distributed decision

Laurent Feuilloley
Université Paris Diderot

based on joint works with Pierre Fraigniaud, Juho Hirvonen, Ami Paz, and Mor Perry

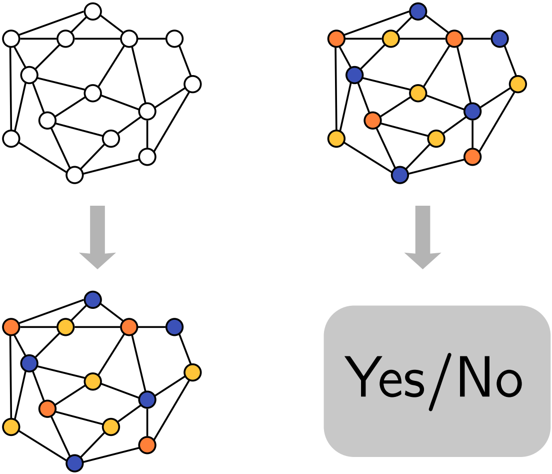
2017

Locality



Distributed decision

Building vs. deciding



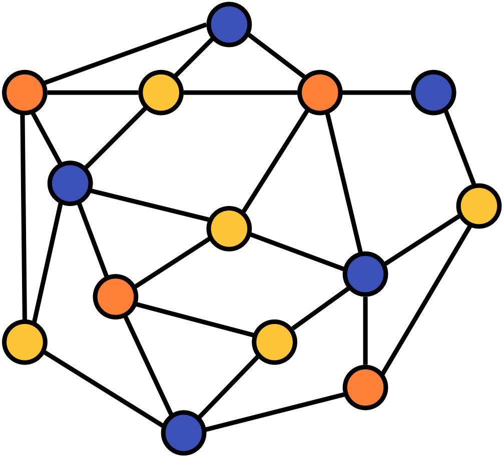
Distributed languages

Context :

- Communication graph G
- Node inputs, $x : v \mapsto x(v)$

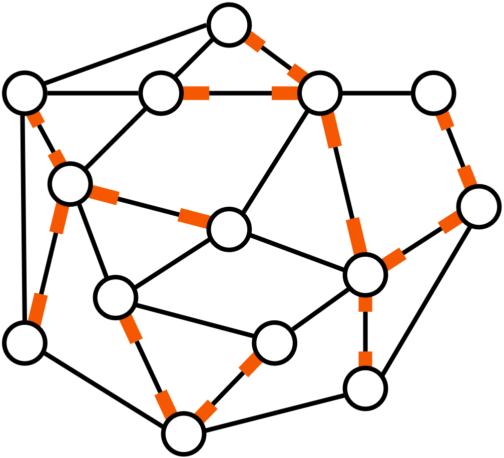
A language is a set $\{(G, x)\}$.

Properly-colored graphs



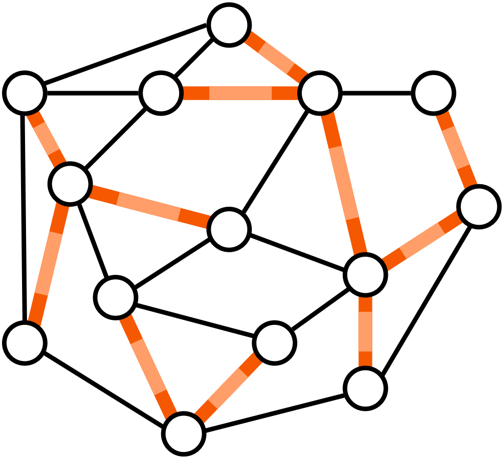
$\mathcal{L} = \{(G, x)$ s.t. x is a proper coloring of $G\}$

Spanning forest



$\mathcal{L} = \{(G, x) \text{ s.t. } x \text{ describes a spanning forest of } G\}$

Spanning forest

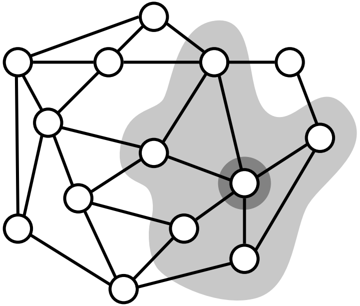


$\mathcal{L} = \{(G, x) \text{ s.t. } x \text{ describes a spanning forest of } G\}$

Decision mechanism

Every node :

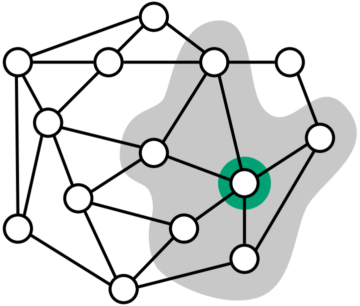
- gathers its 1-neighbourhood
- outputs a local decision accept or reject.



Decision mechanism

Every node :

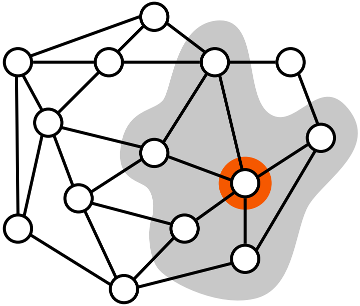
- gathers its 1-neighbourhood
- outputs a local decision accept or reject.



Decision mechanism

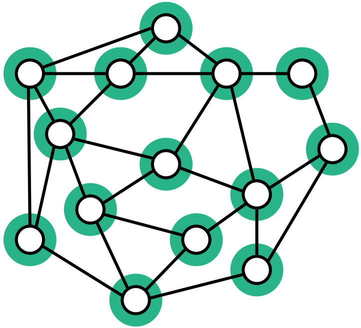
Every node :

- gathers its 1-neighbourhood
- outputs a local decision accept or reject.



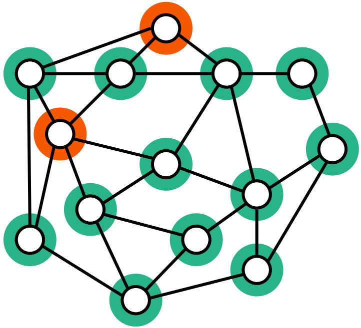
Decision mechanism

(G, x) is accepted
if all node accept.

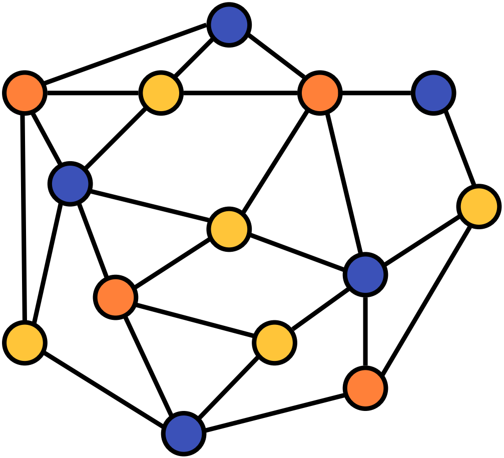


Decision mechanism

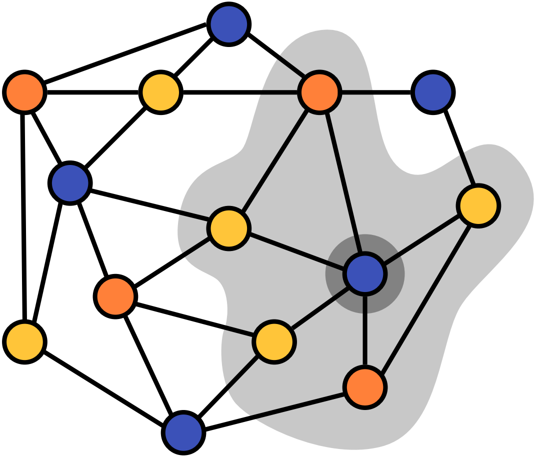
(G, x) is rejected
if at least one node
rejects.



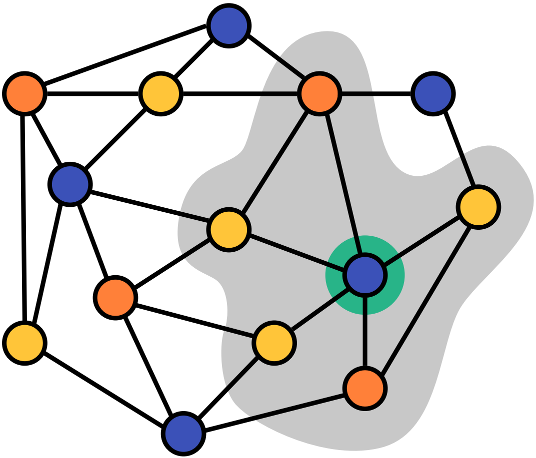
Properly-colored graphs



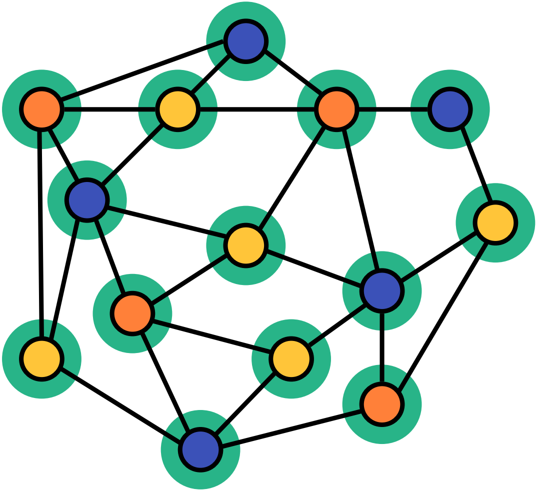
Properly-colored graphs



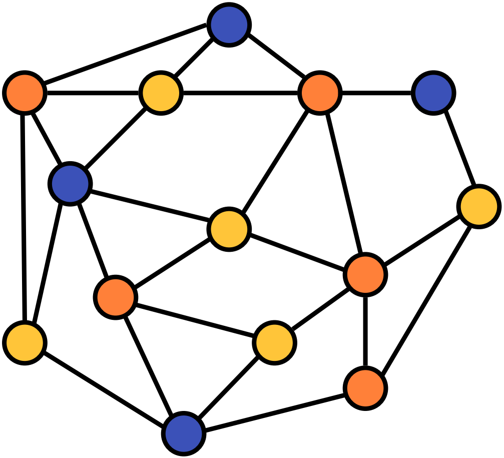
Properly-colored graphs



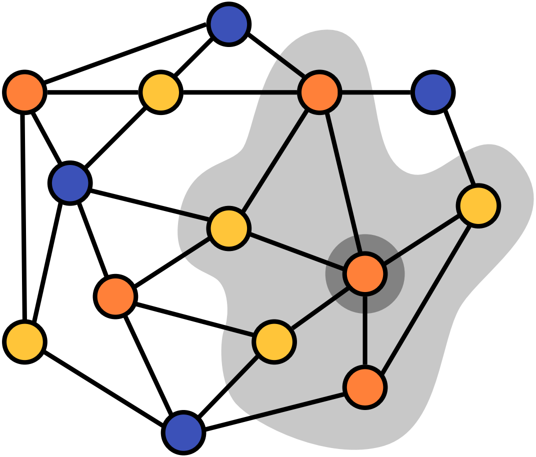
Properly-colored graphs



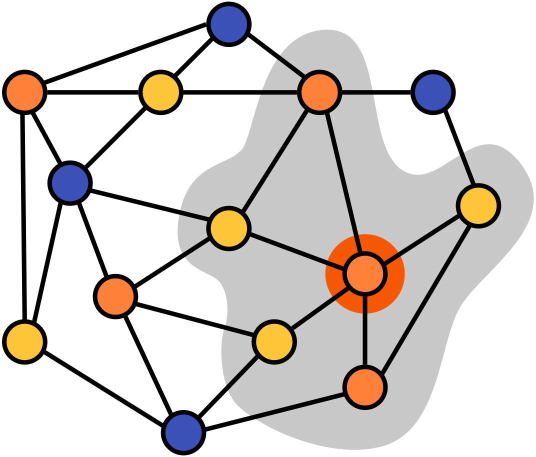
Properly-colored graphs



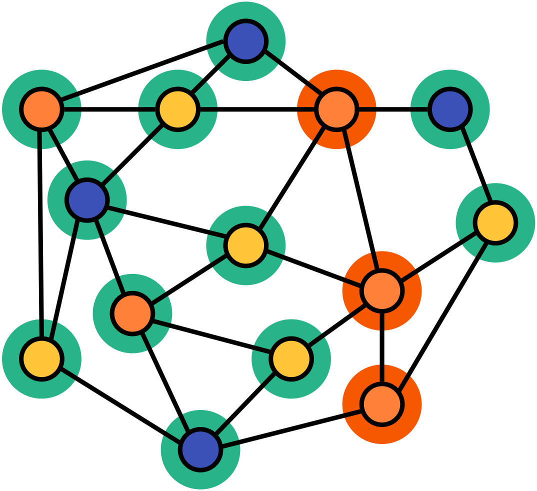
Properly-colored graphs



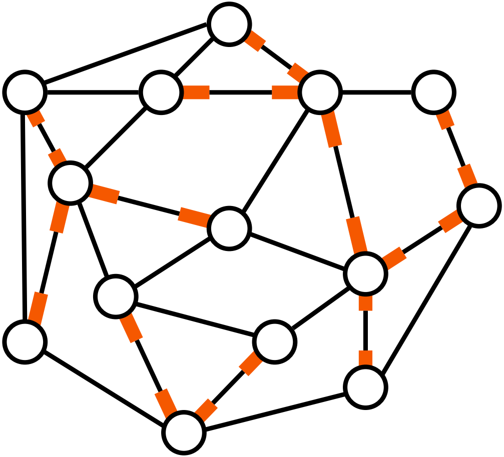
Properly-colored graphs



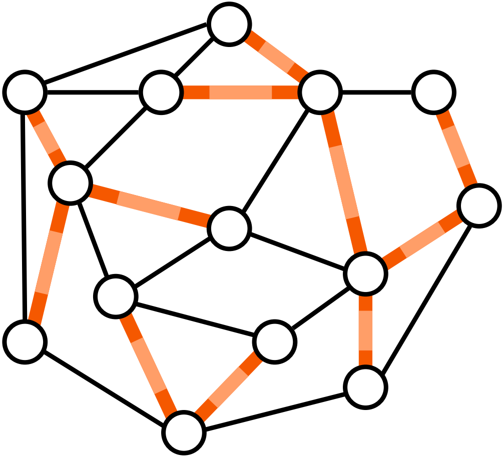
Properly-colored graphs



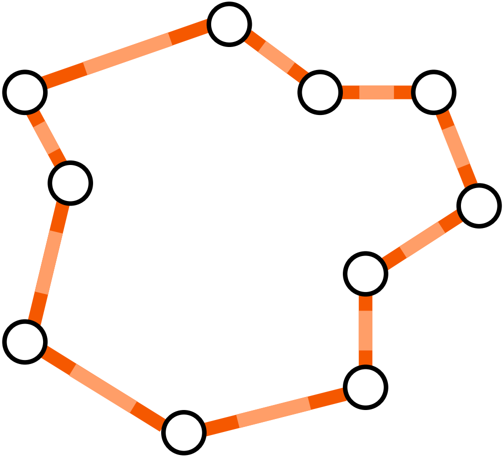
Spanning forest



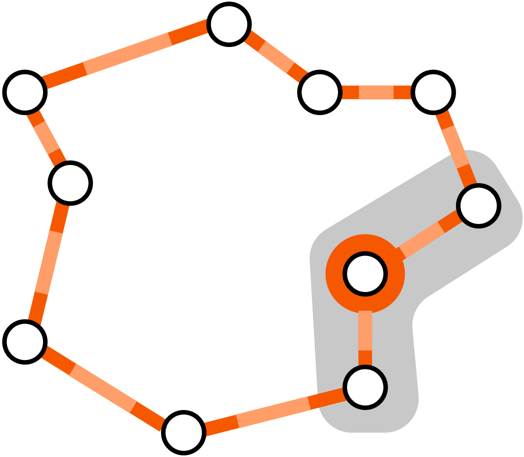
Spanning forest



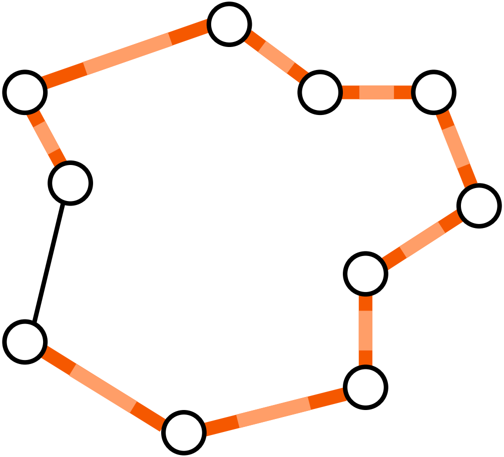
Spanning forest



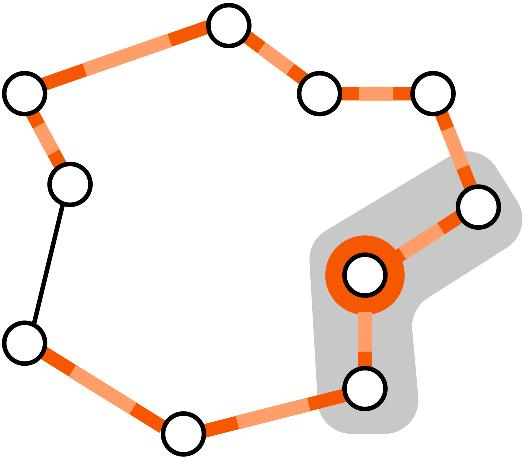
Spanning forest



Spanning forest



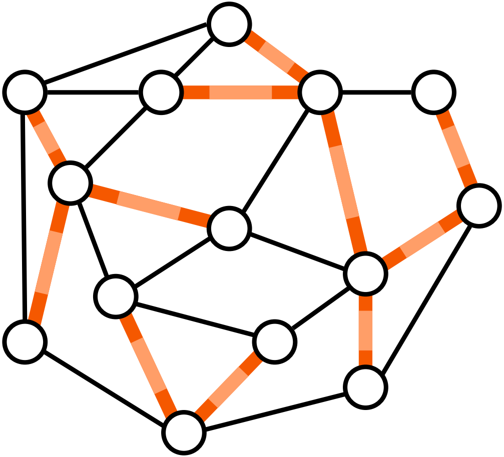
Spanning forest



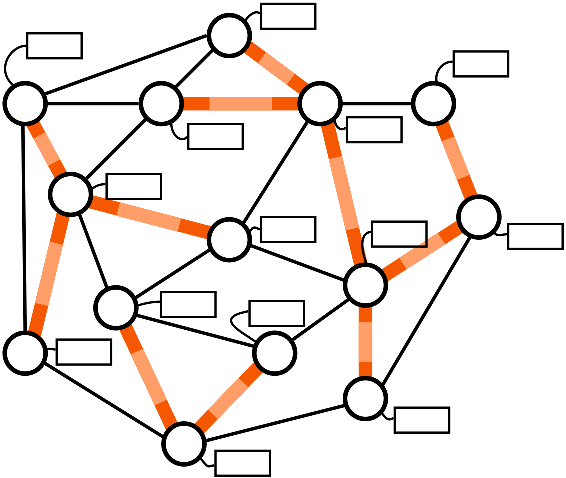
Proof-labeling schemes

Distributed non-determinism

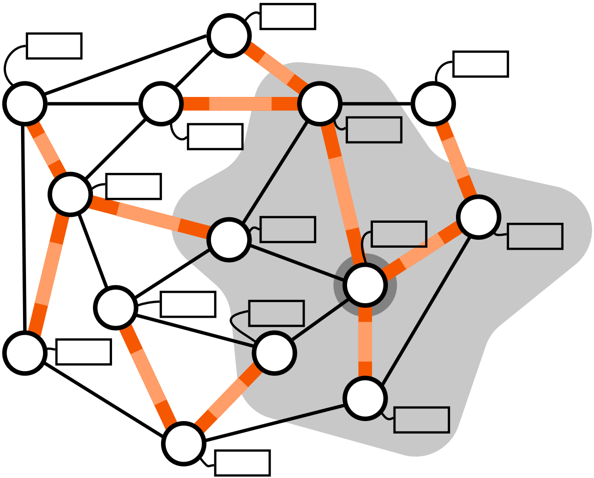
Proof-labeling schemes



Proof-labeling schemes



Proof-labeling schemes



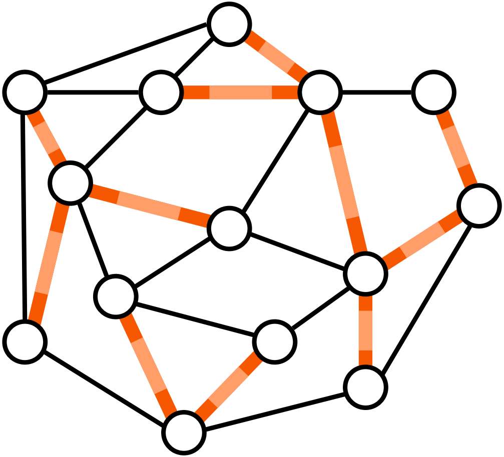
Proof-labeling schemes

Given a proof-labeling scheme for $\mathcal{L}$:

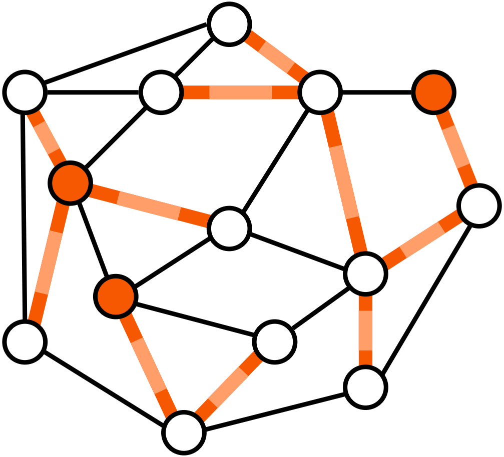
For all (G, x) :

- If $(G, x) \in \mathcal{L}$:
  $\exists c$ s.t. (G, x, c) is accepted.
- If $(G, x) \notin \mathcal{L}$:
  $\forall c$, (G, x, c) is rejected.

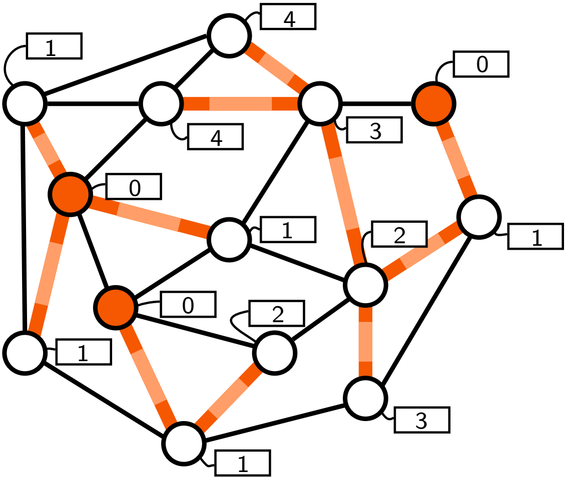
PLS on spanning forest



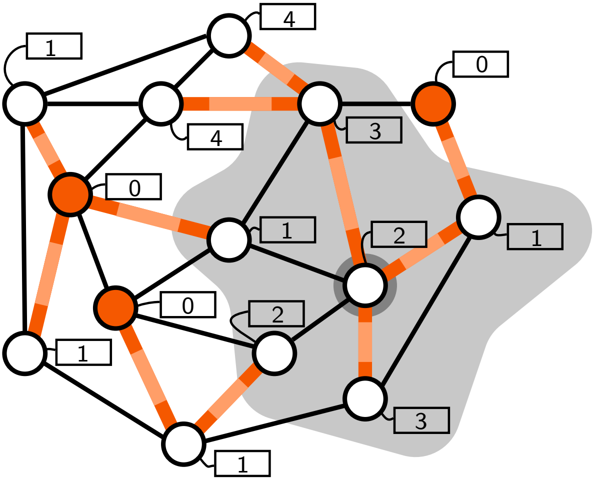
PLS on spanning forest



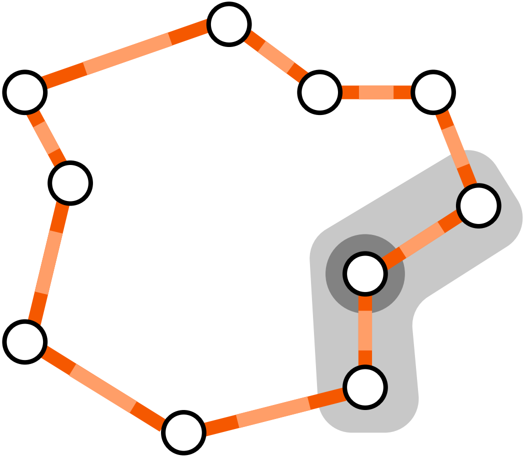
PLS on spanning forest



PLS on spanning forest

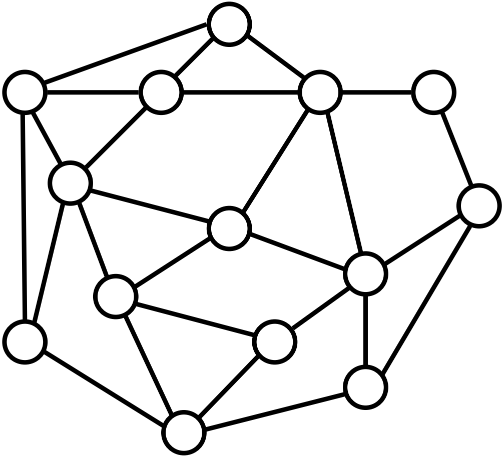


PLS on spanning forest

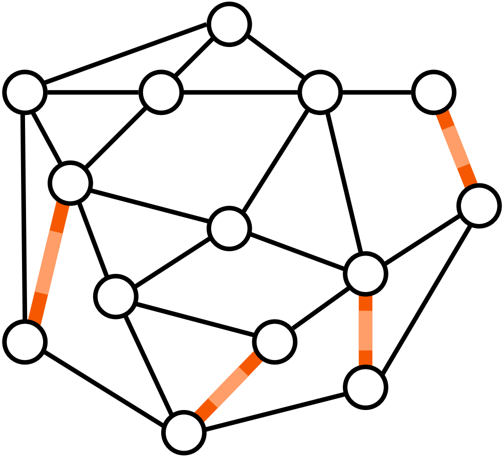


Motivations

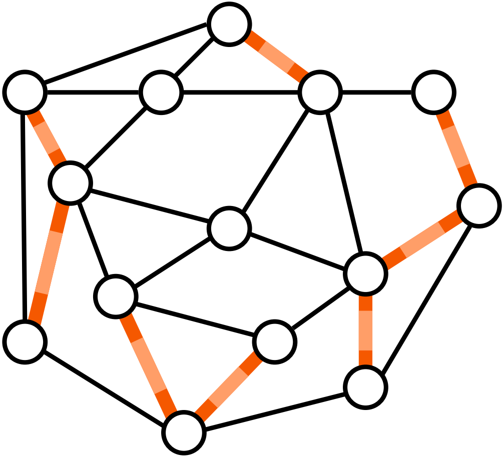
Self-stabilizing algorithms



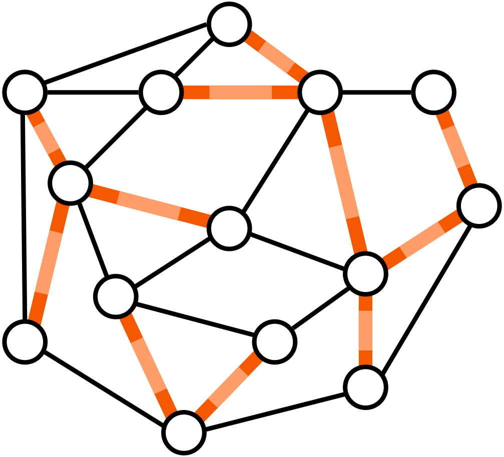
Self-stabilizing algorithms



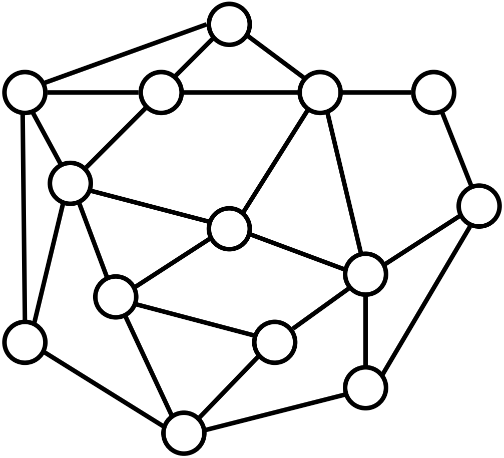
Self-stabilizing algorithms



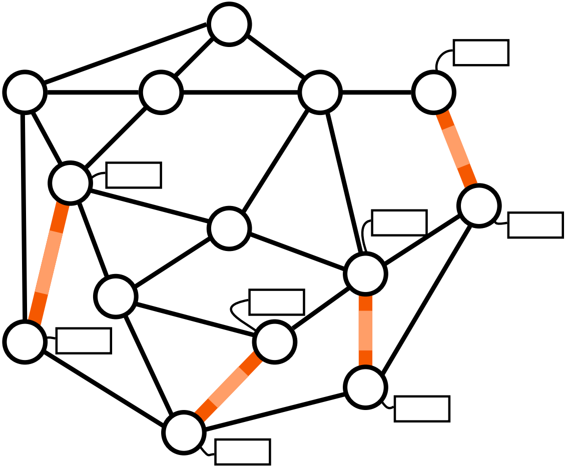
Self-stabilizing algorithms



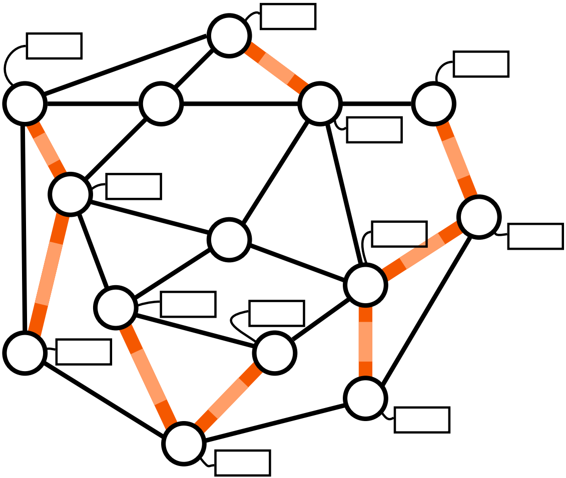
Self-stabilizing algorithms



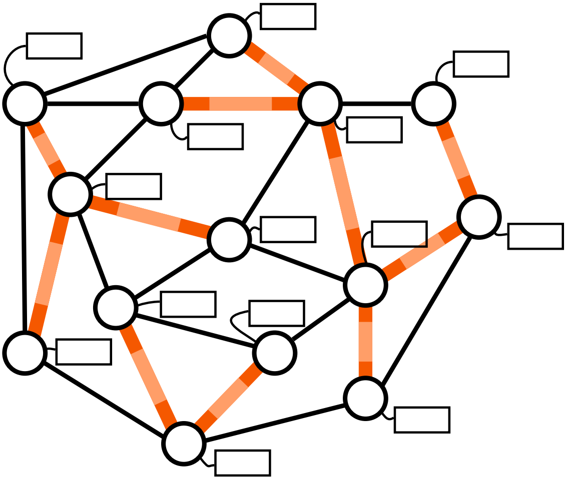
Self-stabilizing algorithms



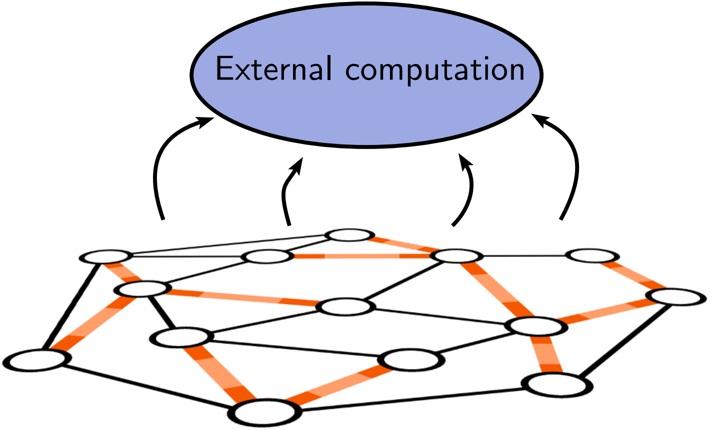
Self-stabilizing algorithms



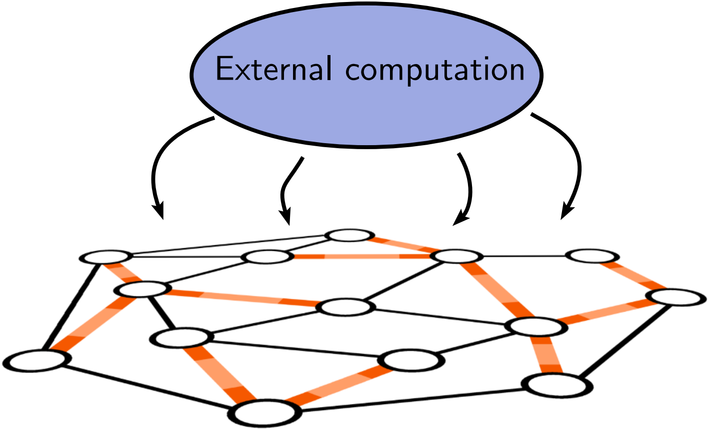
Self-stabilizing algorithms



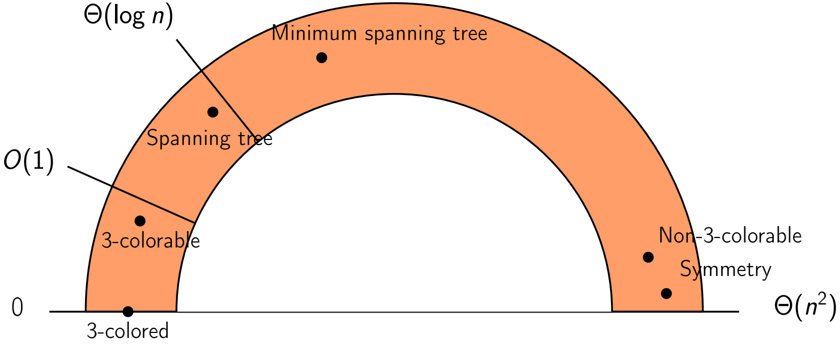
Untrusted oracle



Untrusted oracle



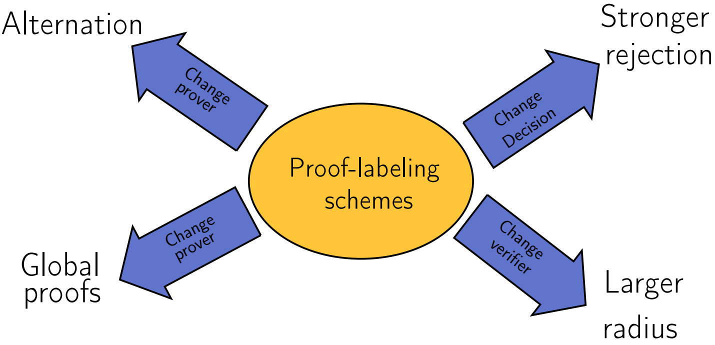
Measuring locality via proof sizes



Unifying models

Unifying models

New works

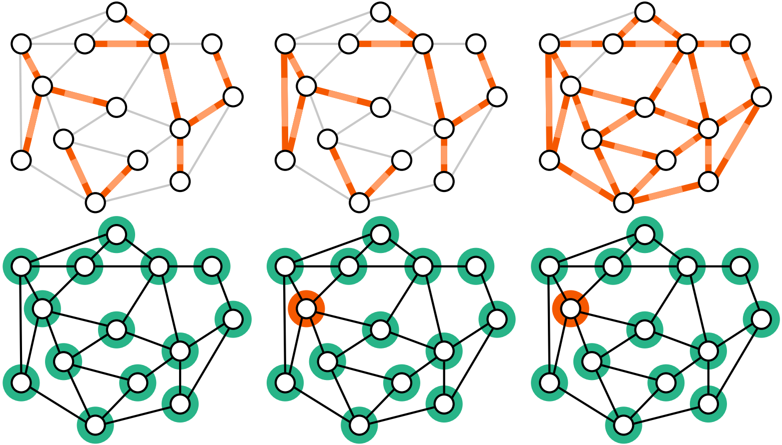

Alternation
Change prover
Stronger rejection
Change Decision
Proof-labeling schemes
Change prover
Global proofs
Change verifier
Larger radius

Stronger rejection

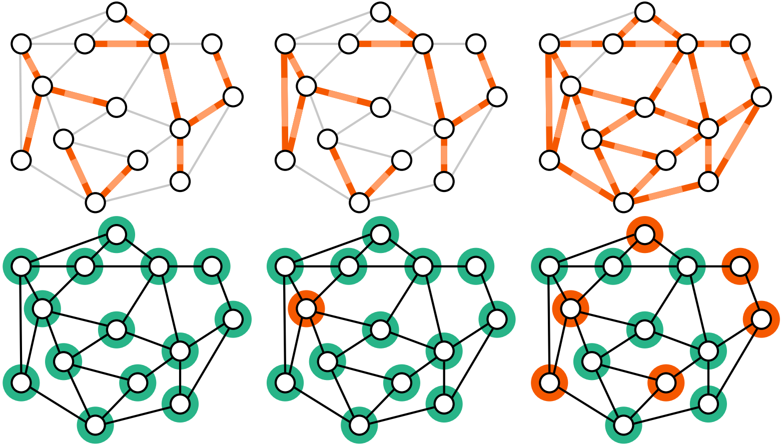
a.k.a.

Error-sensitivity of proof-labeling schemes

One node to reject



More nodes to reject



Error-sensitivity

A PLS is error-sensitive if
the number of rejecting nodes
grows linearly with the distance.

Characterization

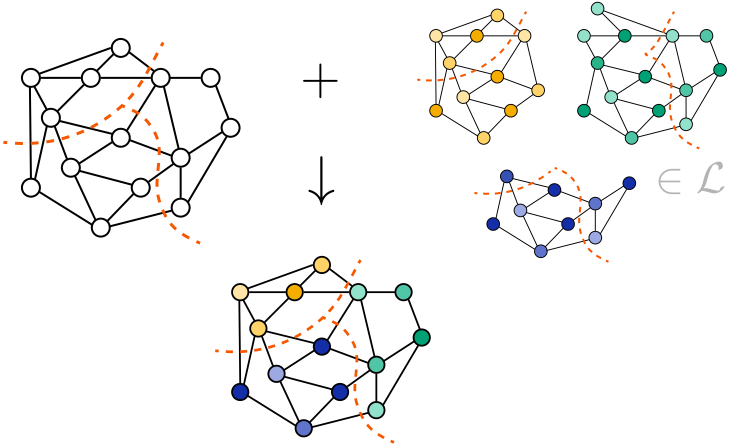
A language $\mathcal{L}$ admits
an error-sensitive PLS

$$\Leftrightarrow$$

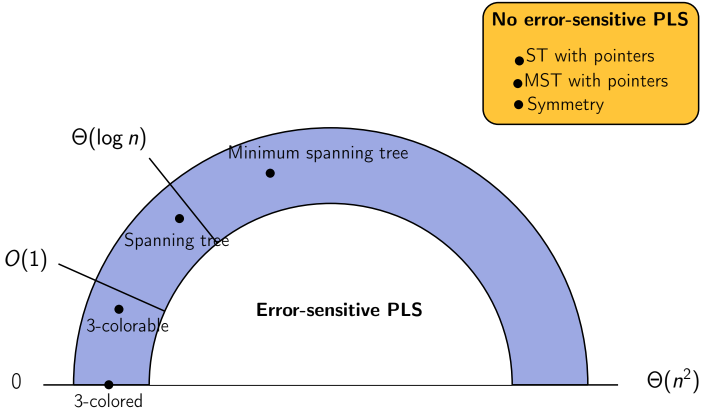
$\mathcal{L}$ is **locally stable**

Local stability

Hybridization



Landscape



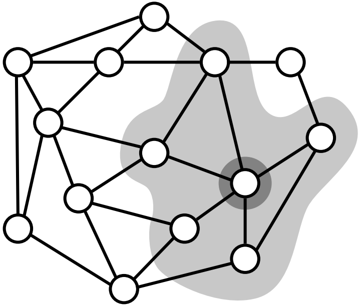
No error-sensitive PLS
ST with pointers
MST with pointers
Symmetry
Θ(log n)
Minimum spanning tree
Spanning tree
O(1)
Error-sensitive PLS
3-colorable
0
3-colored
Θ(n²)

Larger radius

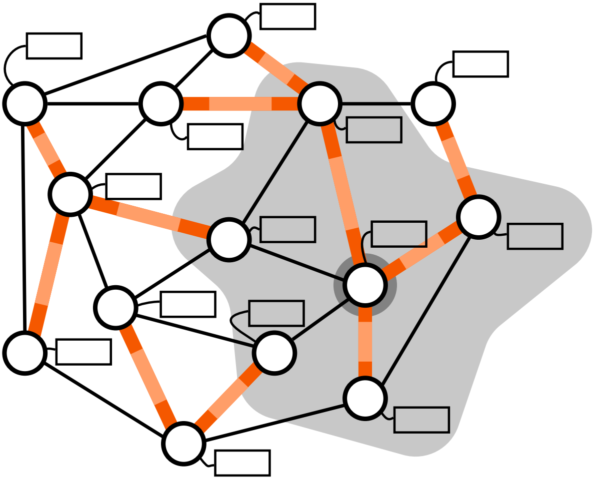
Decision mechanism

Every node :

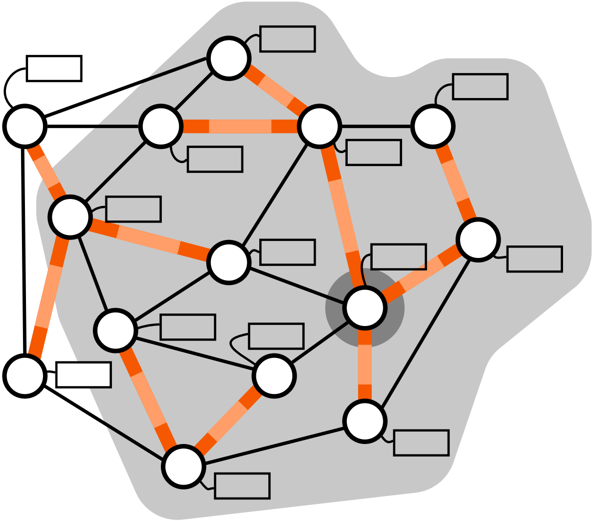
- gathers its 1-neighbourhood
- outputs a local decision accept or reject.



Larger radius



Larger radius

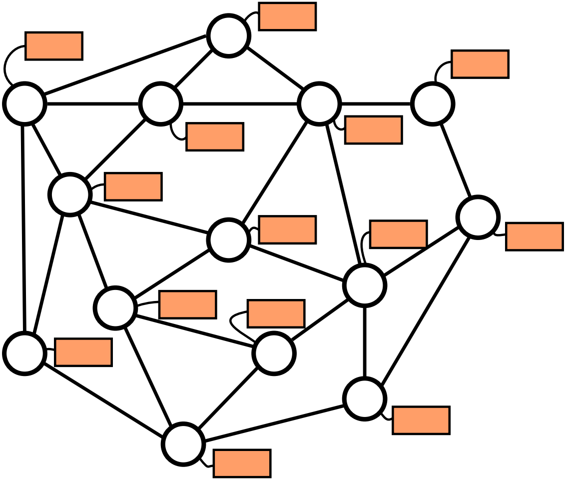


Smaller proofs ?

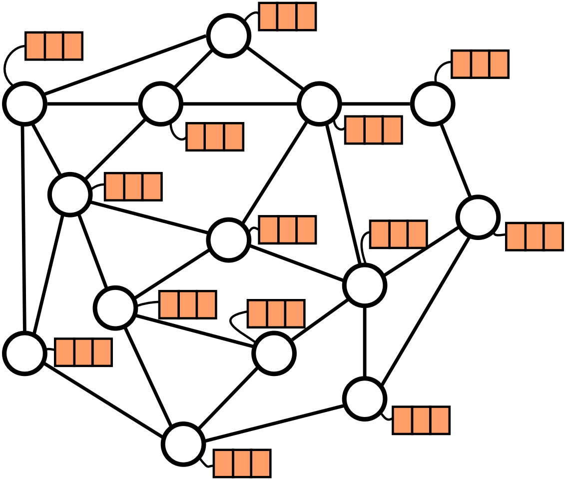
What trade-offs between
the radius t and the certificate size ?

- Can we always get $s_t(n) = s_1(n)/t$?
- When can we get $s_t(n) = s_1(n)/b(t)$?

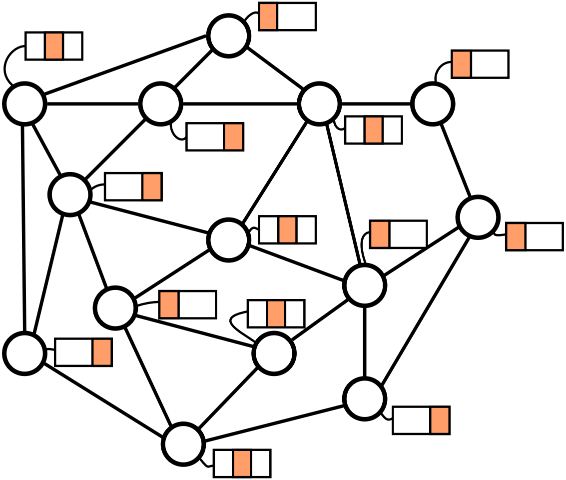
Spreading uniform proofs



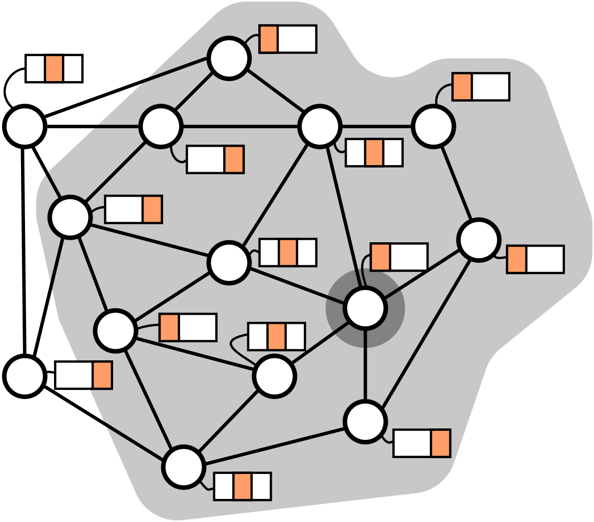
Spreading uniform proofs



Spreading uniform proofs

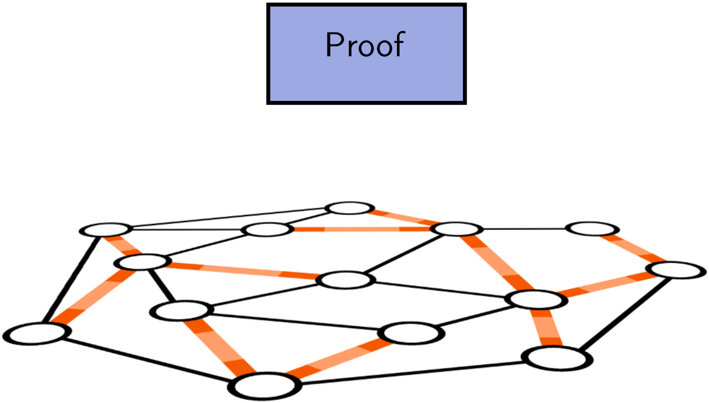


Spreading uniform proofs



Global proofs

A global proof



Basic inequalities

For a fixed language.

- Local(n) : the optimal size for local proofs.
- Global(n) : the optimal size for global proofs.

$$Local(n) \leq Global(n) \leq n \times [Local(n) + \log n]$$

Two selection problems

- AMOS : at most one node is selected
- ALOS : at least one node is selected

$$AMOS \cap ALOS = \text{`Leader elected'}$$

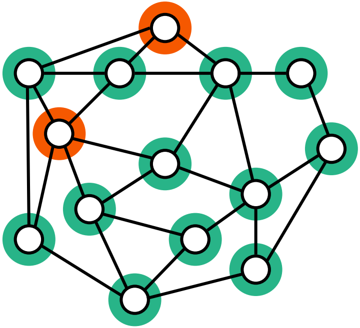
Two selection problems

	Local(n)	Global(n)
AMOS	$\log n$	$\log n$
ALOS	$\log n$	$n \log n$

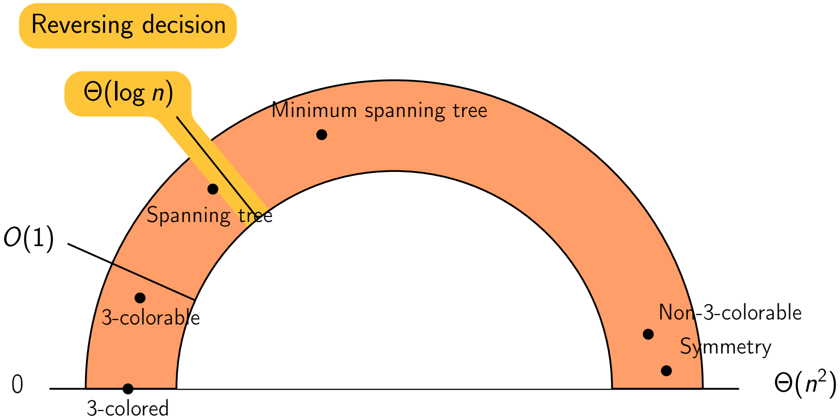
Alternation

Reversing decision

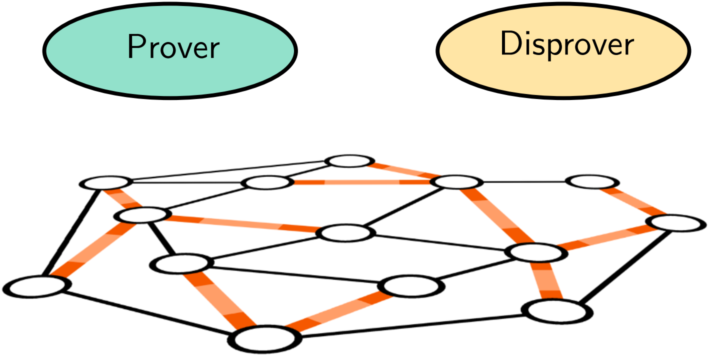
(G, x) is rejected
if at least one node
rejects.



Reversing decision



Prover vs. disprover



Remember PLS

Given a proof-labeling scheme for $\mathcal{L}$:

For all (G, x) :

- If $(G, x) \in \mathcal{L}$:
  $\exists c$ s.t. (G, x, c) is accepted.
- If $(G, x) \notin \mathcal{L}$:
  $\forall c$, (G, x, c) is rejected.

Disprover-prover scheme

Given a disprover-prover scheme for $\mathcal{L}$:

For all (G, x) :

- If $(G, x) \in \mathcal{L}$:
  $\forall c_d \exists c_p$ s.t. (G, x, c_d, c_p) is accepted.
- If $(G, x) \notin \mathcal{L}$:
  $\exists c_d \forall c_p$, (G, x, c_d, c_p) is rejected.

Some conversations

- Non-3-colorable : disprover-prover
- Optimal combinatorial solution : disprover-prover
- Symmetry : prover-disprover-prover
- Some language : talk forever.

Conclusion

Playing with non-determinism is :

- useful, to model to different systems,
- a good way to study locality.