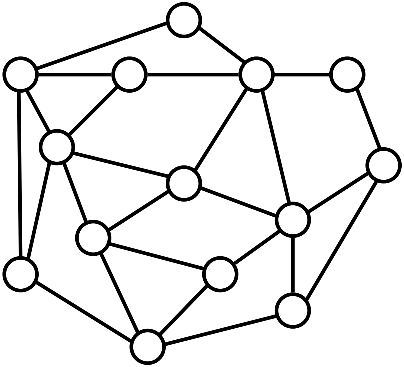
How long it takes for an ordinary node with an ordinary ID to output ?

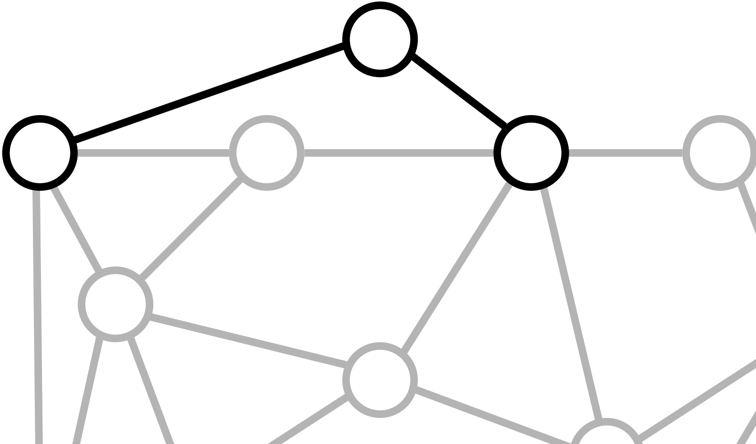
Laurent Feuilloley
Université Paris Diderot

SIROCCO · June 2017

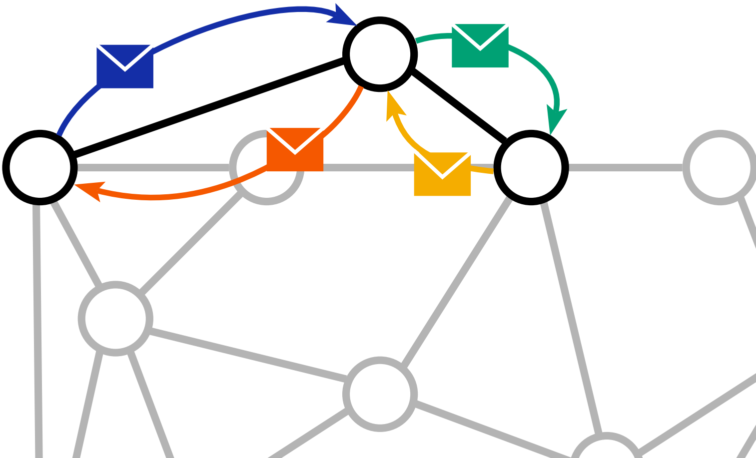
The network



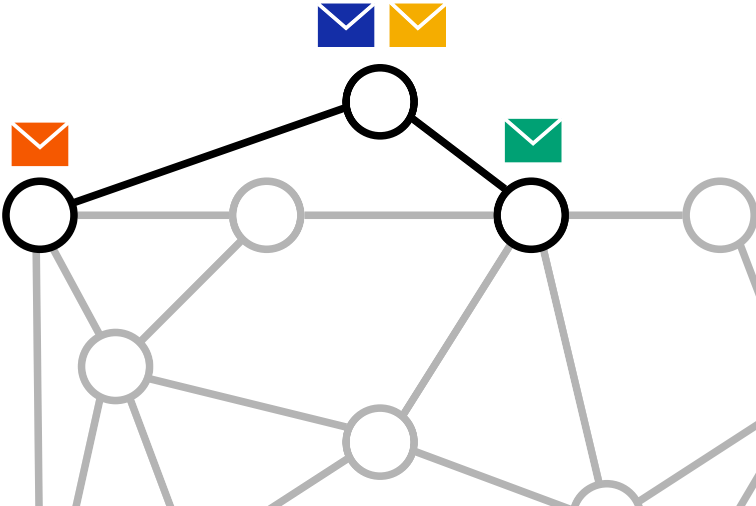
Computation in rounds



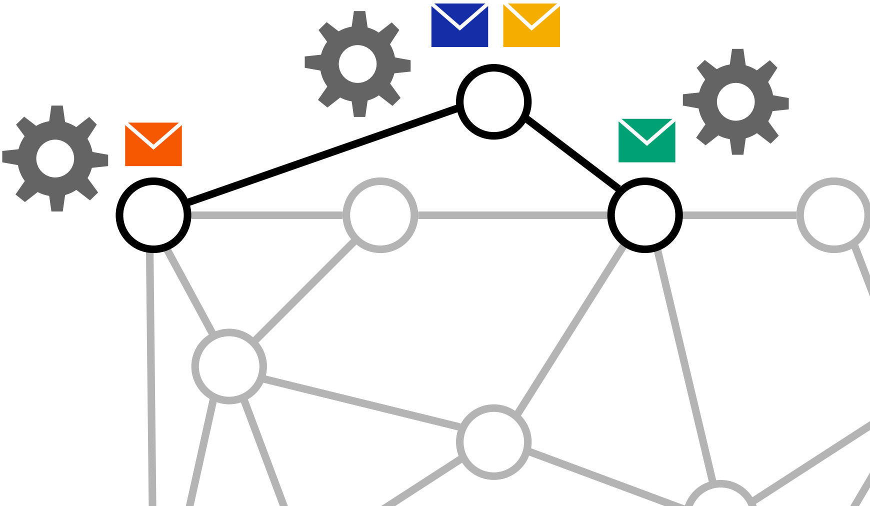
Computation in rounds



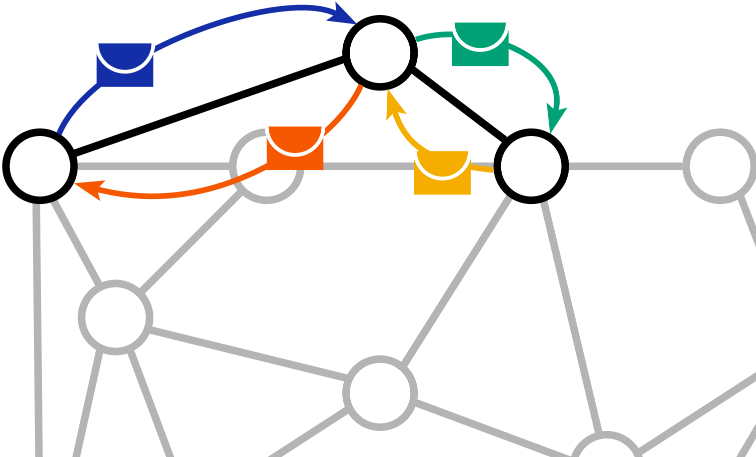
Computation in rounds



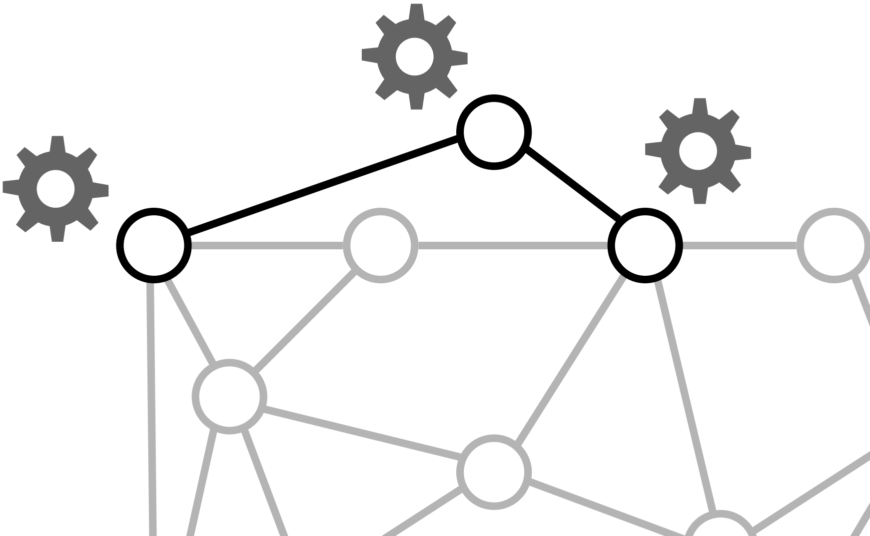
Computation in rounds



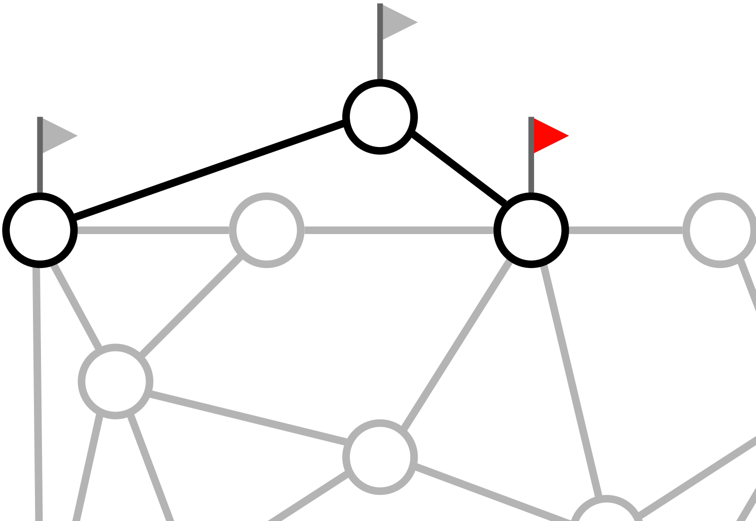
Computation in rounds



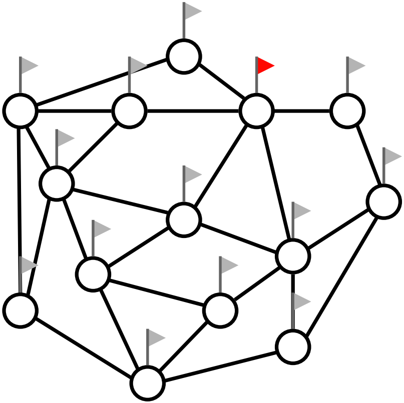
Computation in rounds



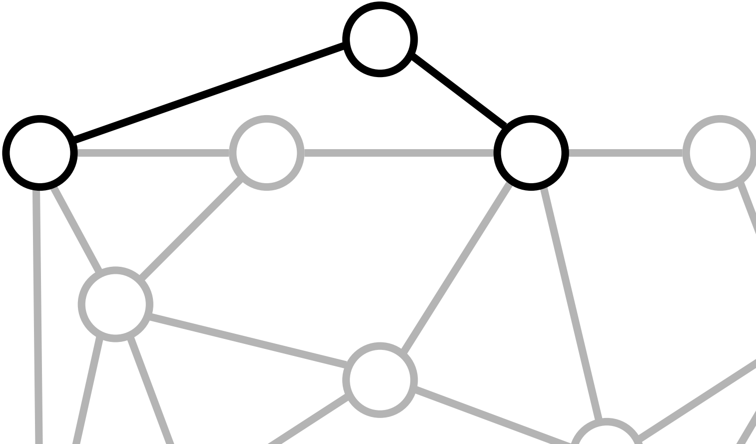
Computation in rounds



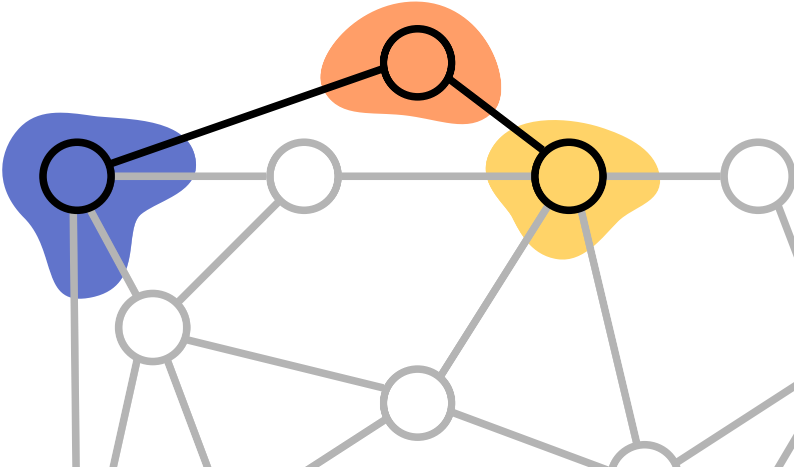
Computation in rounds



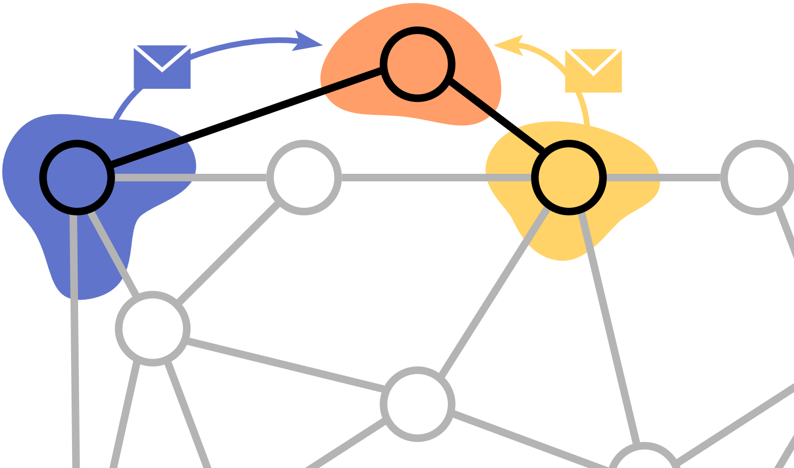
Local knowledge



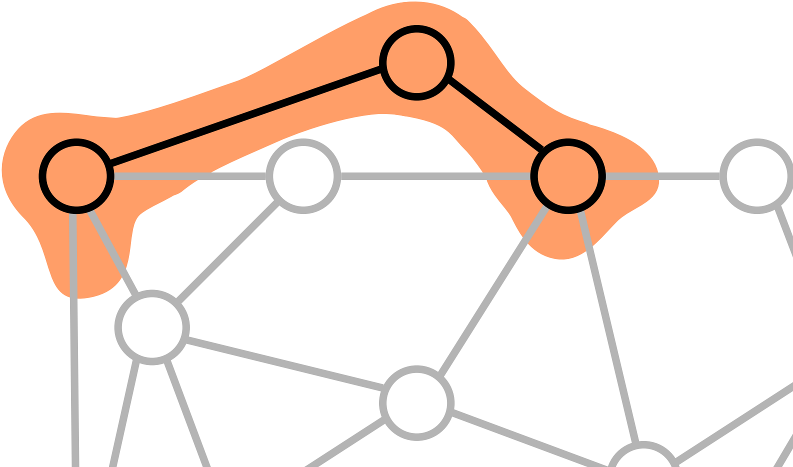
Local knowledge



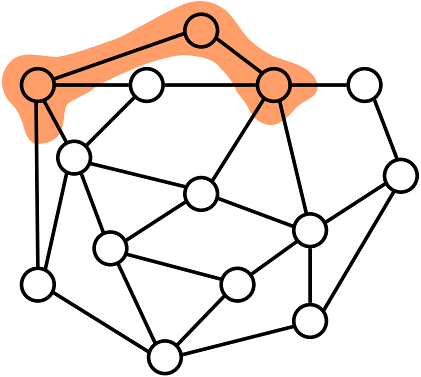
Local knowledge



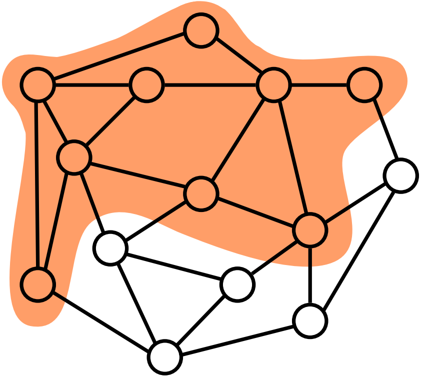
Local knowledge



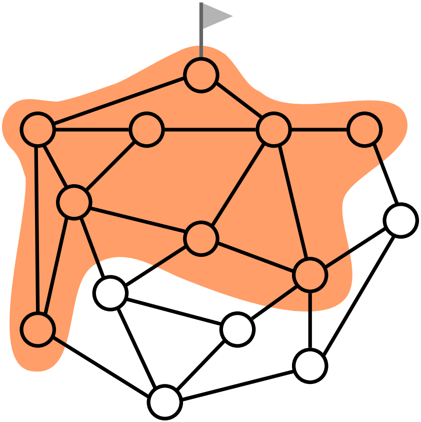
Local knowledge



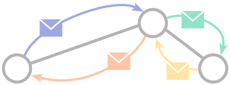
Local knowledge

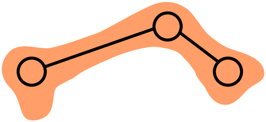


Local knowledge



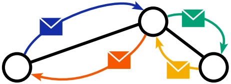
When to output ?

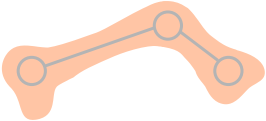




Outputs	Simultaneous	Non-simultaneous
n known	Yes	Not required
Algorithm	Compute $t(n)$, check $t(n)$-view, output.	Increase radius, until enough info, output.

When to output ?





Outputs	Simultaneous	Non-simultaneous
n known	Yes	Not required
Algorithm	Compute $t(n)$, run $t(n)$ rounds, output, stop.	Run until enough info, output, continue to run, stop.

Complexity measures

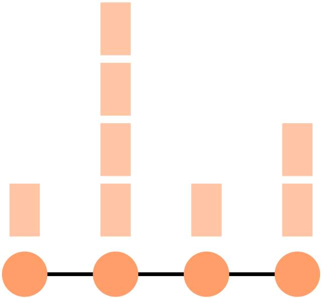
Given a graph G, and an ID assignment :

Slowest node :

$$\max_{v \in G} t(v)$$

Ordinary node :

$$\frac{1}{n} \sum_{v \in G} t(v)$$



Complexity measures

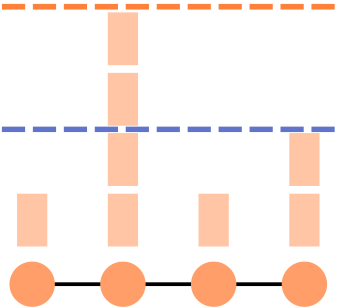
Given a graph G, and an ID assignment :

Slowest node :

$$\max_{v \in G} t(v)$$

Ordinary node :

$$\frac{1}{n} \sum_{v \in G} t(v)$$



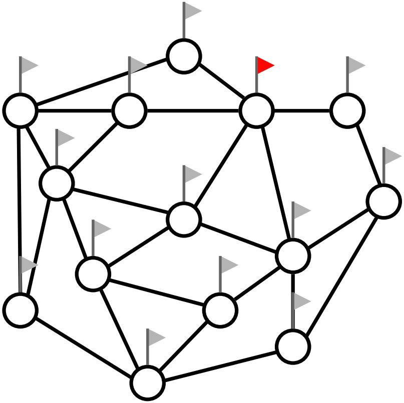
Complexity measures

Complexity of a problem :

$$\max_{G} \max_{IDs} \frac{1}{n} \sum_{v \in G} t(v)$$

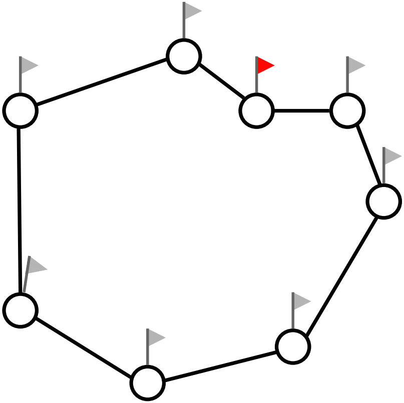
Leader election

Leader election : exactly one node is selected.



Leader election

Leader election : exactly one node is selected.

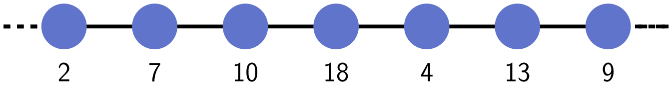


An algorithm for L.E.

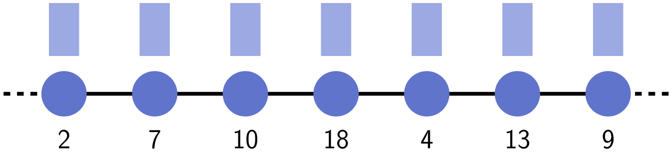
For each node :

- increase the radius until you see a larger ID
- if such an ID exists, then output *non-leader*
- else output *leader*.

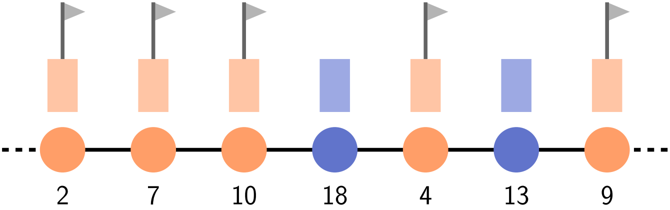
An algorithm for L.E.



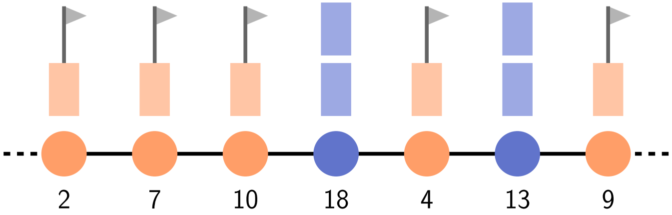
An algorithm for L.E.



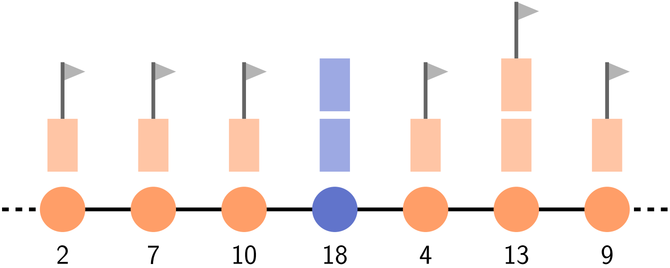
An algorithm for L.E.



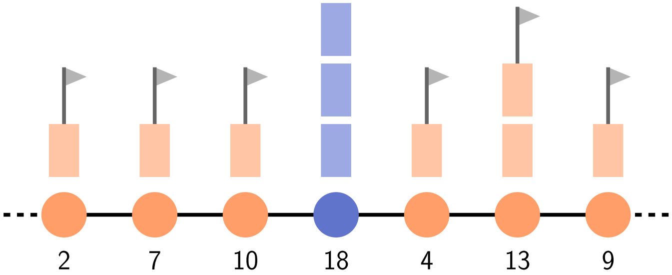
An algorithm for L.E.



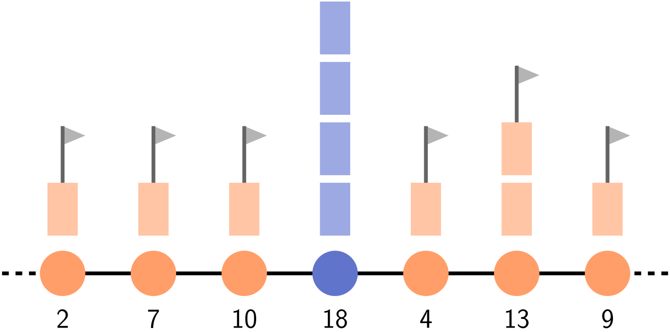
An algorithm for L.E.



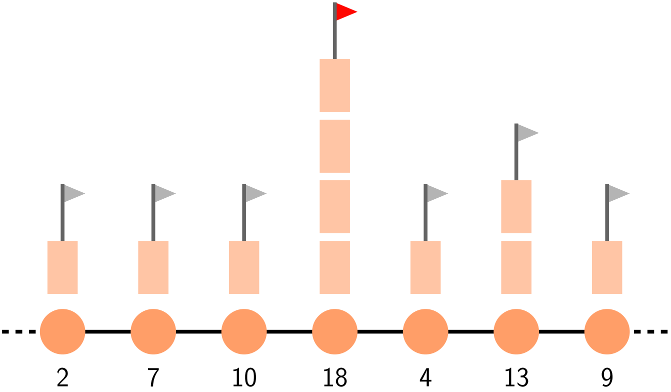
An algorithm for L.E.



An algorithm for L.E.



An algorithm for L.E.



An exponential gap

Ordinary node best ID assign.	Ordinary node worst ID assign.	Slowest node any ID assign.
$O(1)$	$O(\log n)$	$\Omega(n)$

Local problems

Local problem :

Locally correct
everywhere

Local problems

Local problem :

Locally correct
everywhere

Local problems

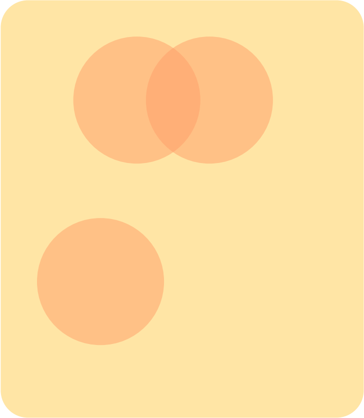
Local problem :

Locally correct everywhere

Local problems

Local problem :

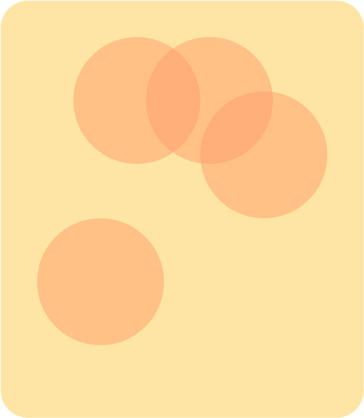
Locally correct
everywhere



Local problems

Local problem :

Locally correct
everywhere



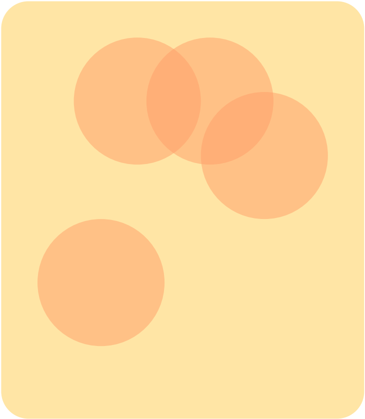
Local problems

Typical local problem :

Colouring

Typical global problem :

Leader election



No gap for local problems

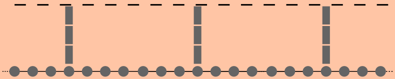
Example of application :

Thm [Linial] : For 3-colouring a cycle, the slowest node requires $\Omega(\log^* n)$ rounds.

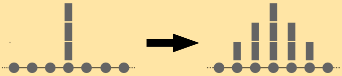
Thm [this paper] : For 3-colouring a cycle, an ordinary node requires $\Omega(\log^* n)$ rounds.

Roadmap of the proof

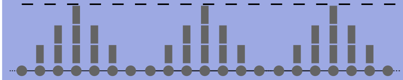
Many high radiuses



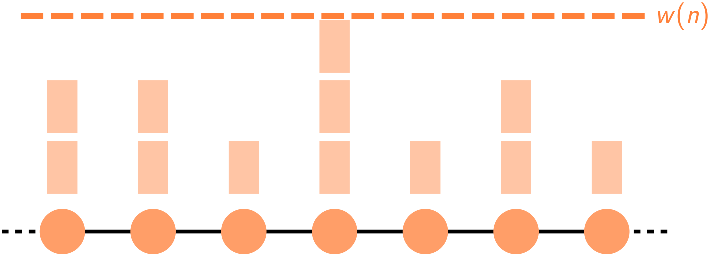
Smoothness



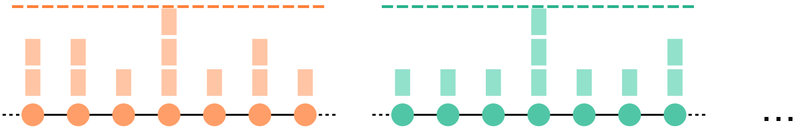
No gap



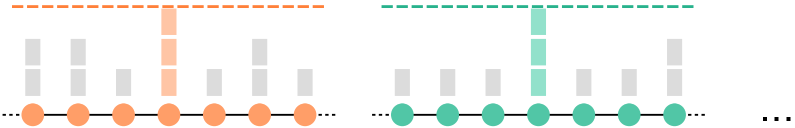
Proof : many high radiuses



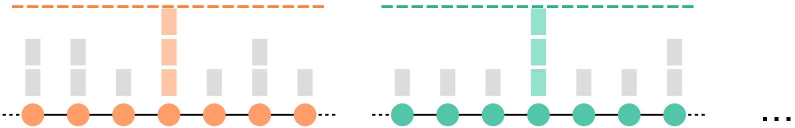
Proof : many high radiuses

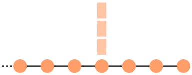


Proof : many high radiuses

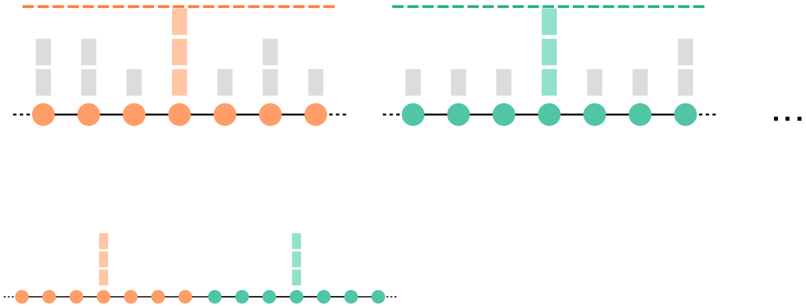


Proof : many high radiuses

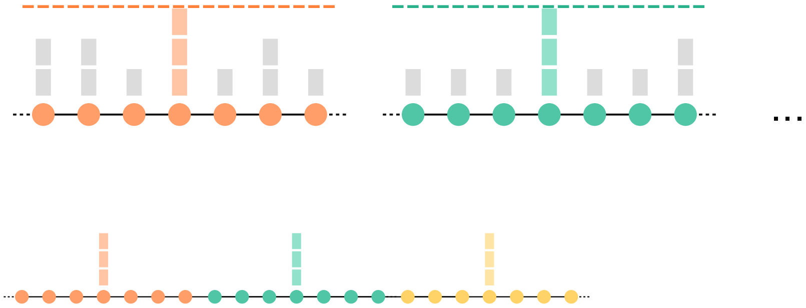




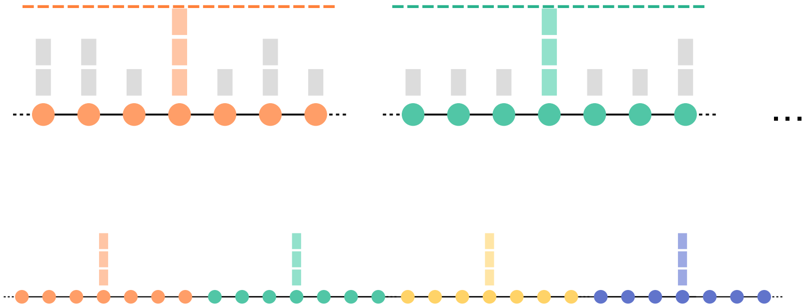
Proof : many high radiuses



Proof : many high radiuses

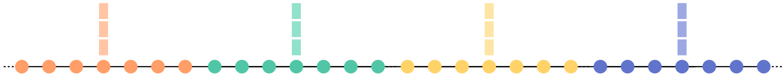


Proof : many high radiuses

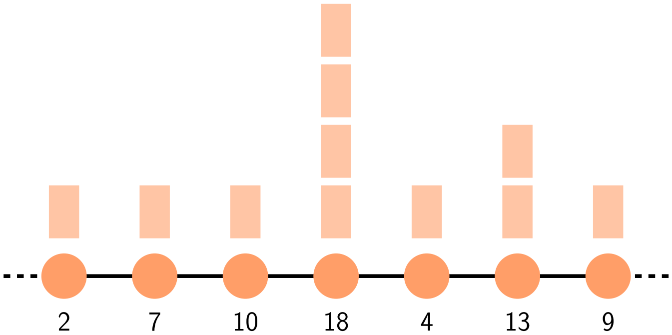


Proof : many high radiuses

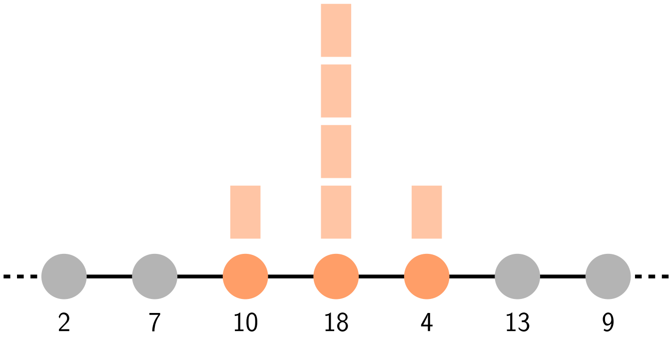
$\rightarrow \frac{n}{w(n)}$ nodes with $w(n)$ radius



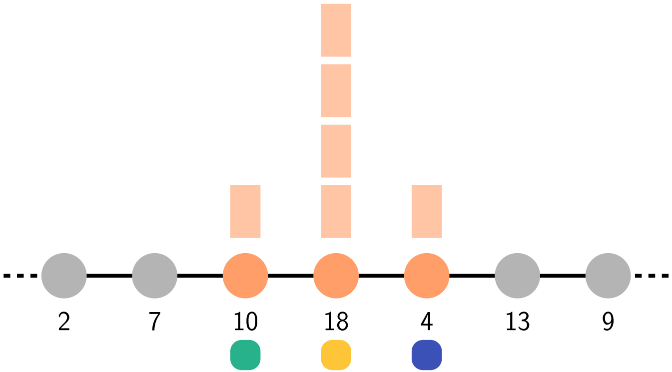
Proof : smoothness



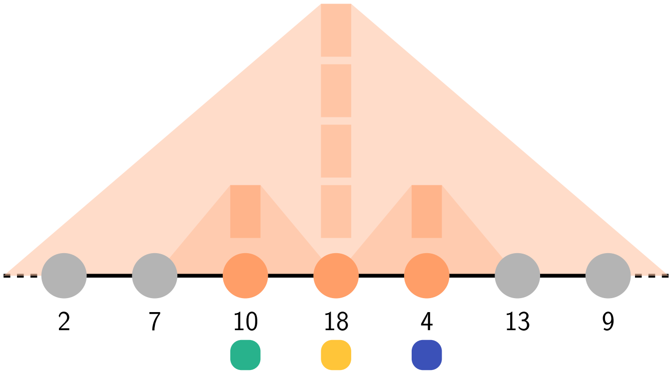
Proof : smoothness



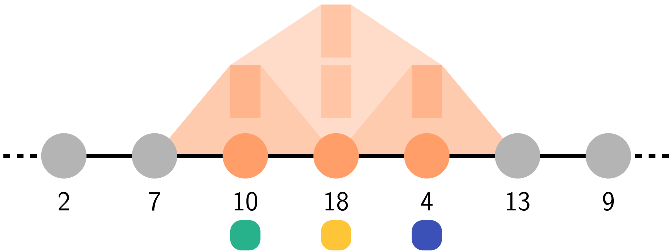
Proof : smoothness



Proof : smoothness



Proof : smoothness



Complexity measures

Complexity of a problem :

$$\max_{G} \max_{IDs} \frac{1}{n} \sum_{v \in G} t(v)$$

Complexity measures

Complexity of a problem :

$$\max_{G} \max_{IDs} \frac{1}{n} \sum_{v \in G} t(v)$$

Random IDs

Complexity of a problem :

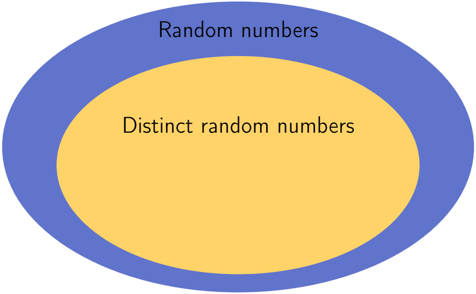
$$\max_G \frac{1}{|IDs|} \sum_{IDs} \max_{v \in G} t(v)$$

Random IDs

Random ID Assignment	Randomized algorithm
Random $O(\log n)$ ID assign.	Random $O(\log n)$ numbers
Uniqueness	Independance

Random IDs

Whp. n random numbers from $[n^4]$ are distinct.



Ordinary node and ID

Complexity of a problem :

$$\max_{G} \frac{1}{|IDs|} \sum_{IDs} \frac{1}{n} \sum_{v \in G} t(v)$$

Ordinary node and ID

3-colouring of a cycle	Worst ID assign.	Random ID assign.
Slowest node	$\Theta(\log^* n)$	$\Theta(\log^* n)$
Ordinary node	$\Theta(\log^* n)$	$\Theta(1)$