An Improved Distributed Algorithm for Maximal Independent Set

Erfan Abdi

ETH zürich

Introduction to Distributed Computing

Maximal Independent Set

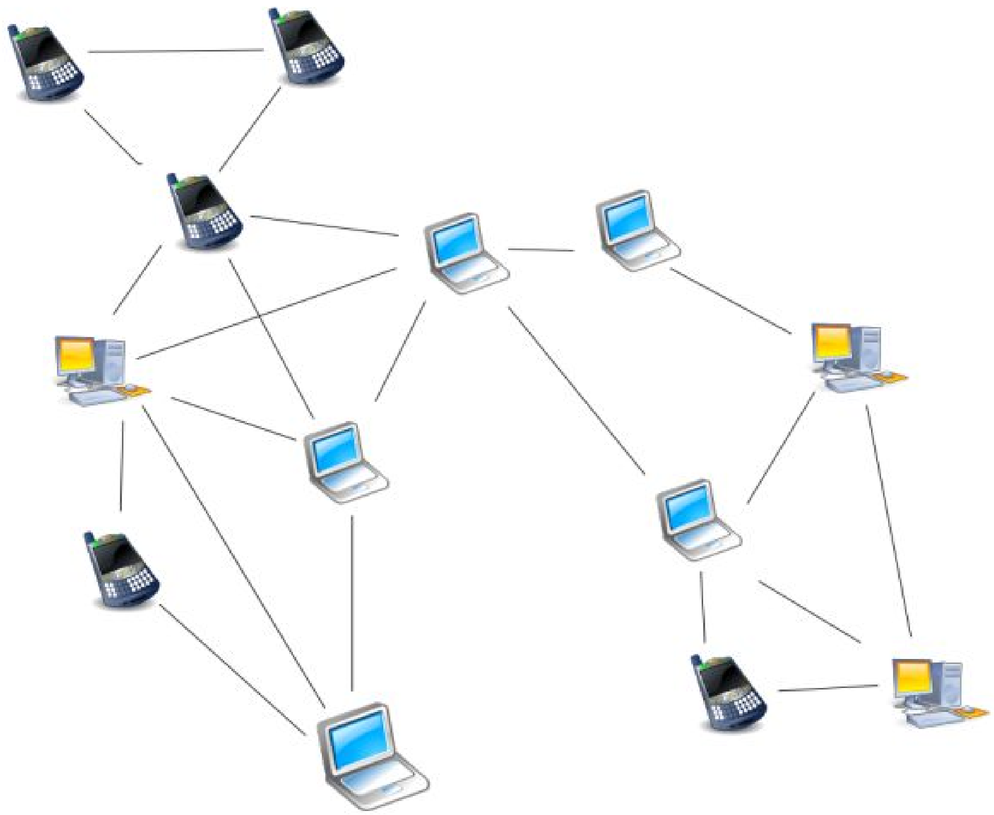
Luby’s Algorithm

Ghaffari’s Algorithm

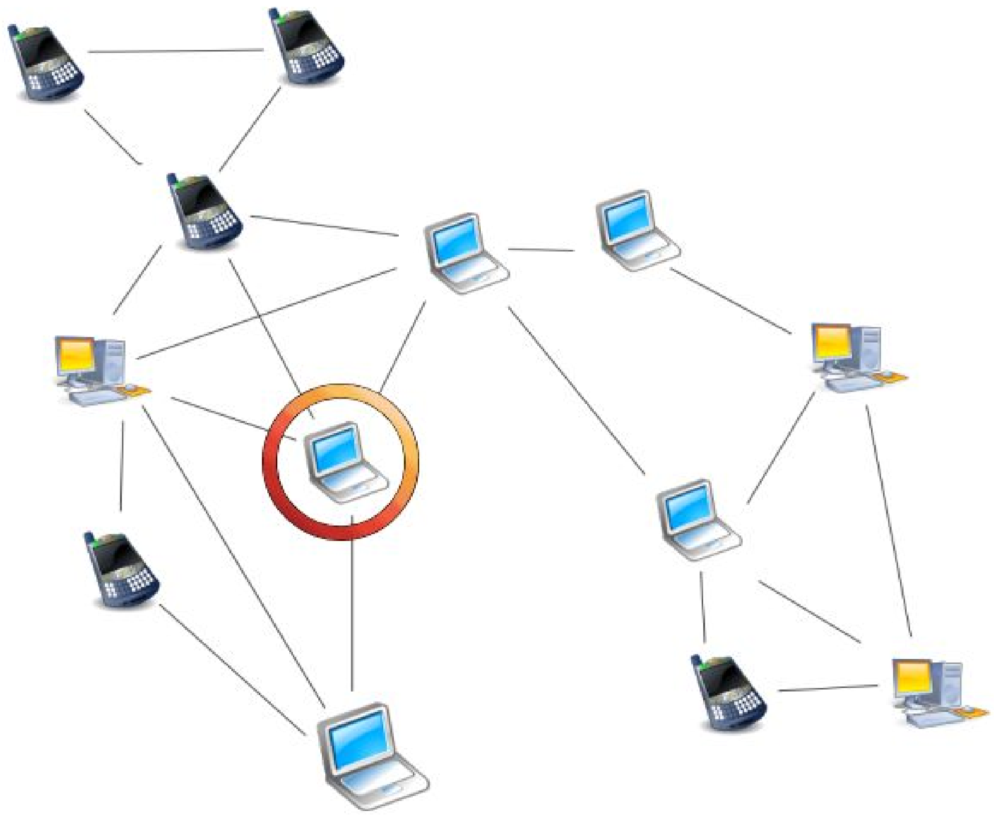
Local Complexity of Ghaffari’s Algorithm

Global complexity of Ghaffari’s Algorithm

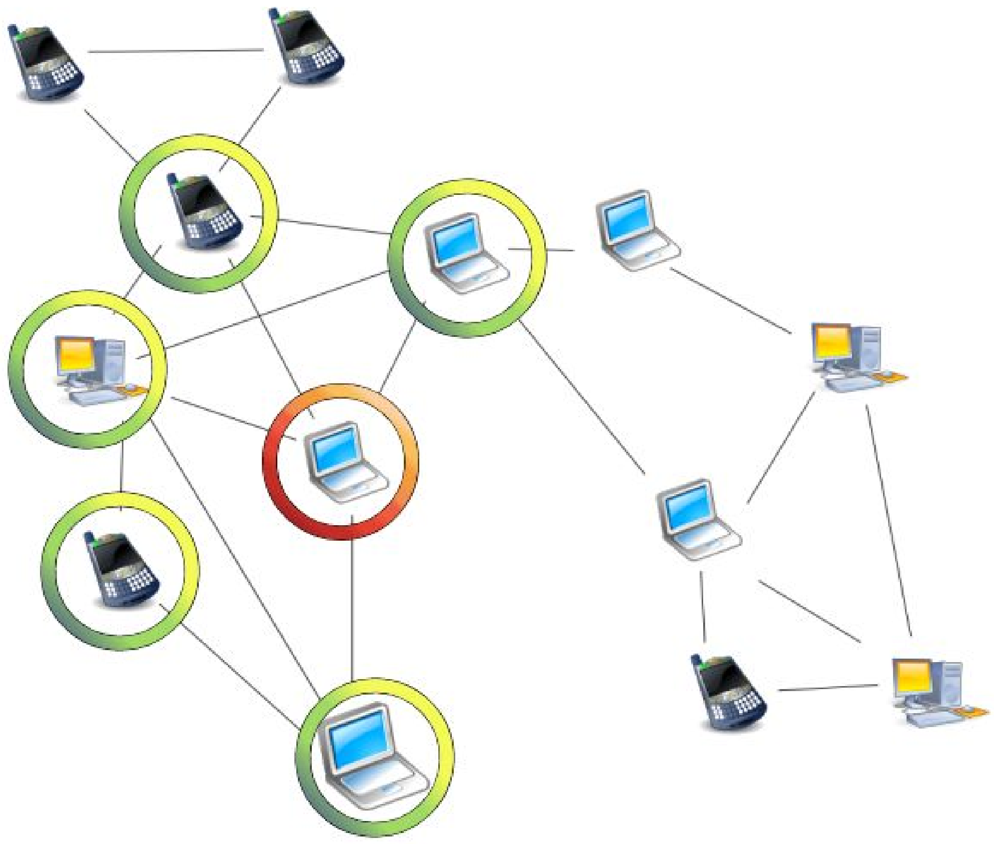
Distributed Computing



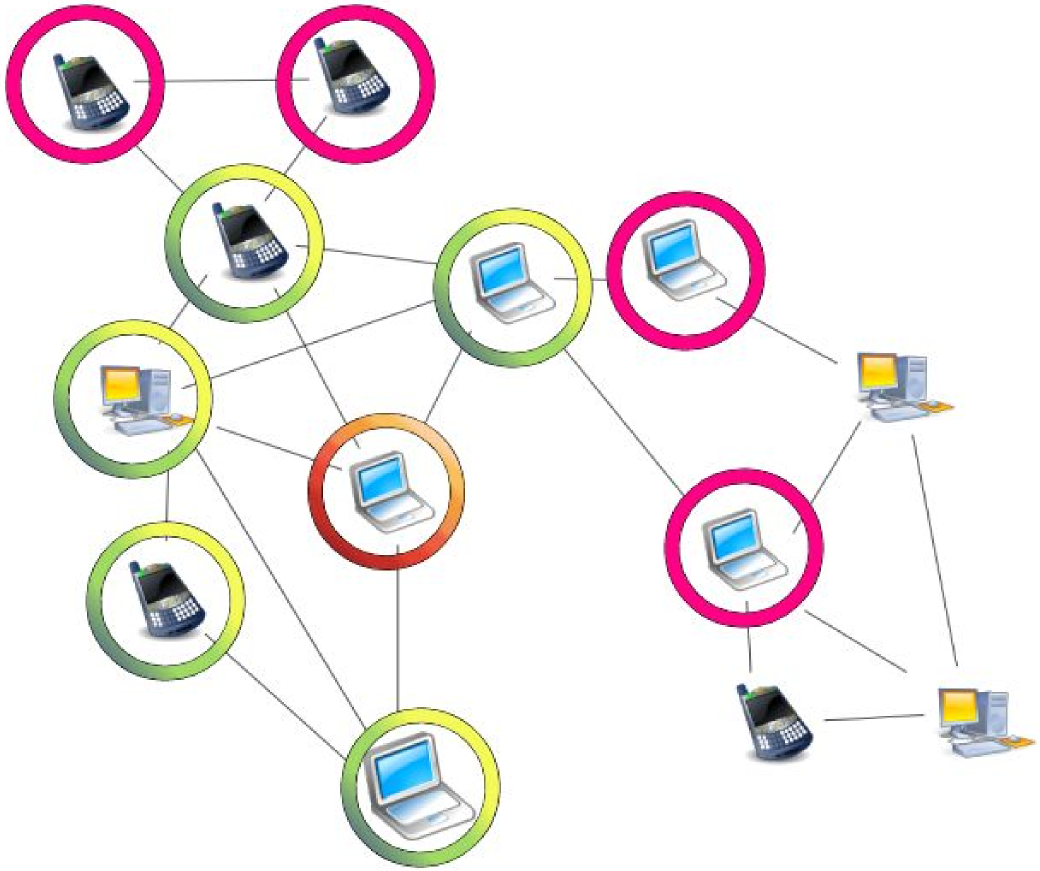
Distributed Computing



Distributed Computing

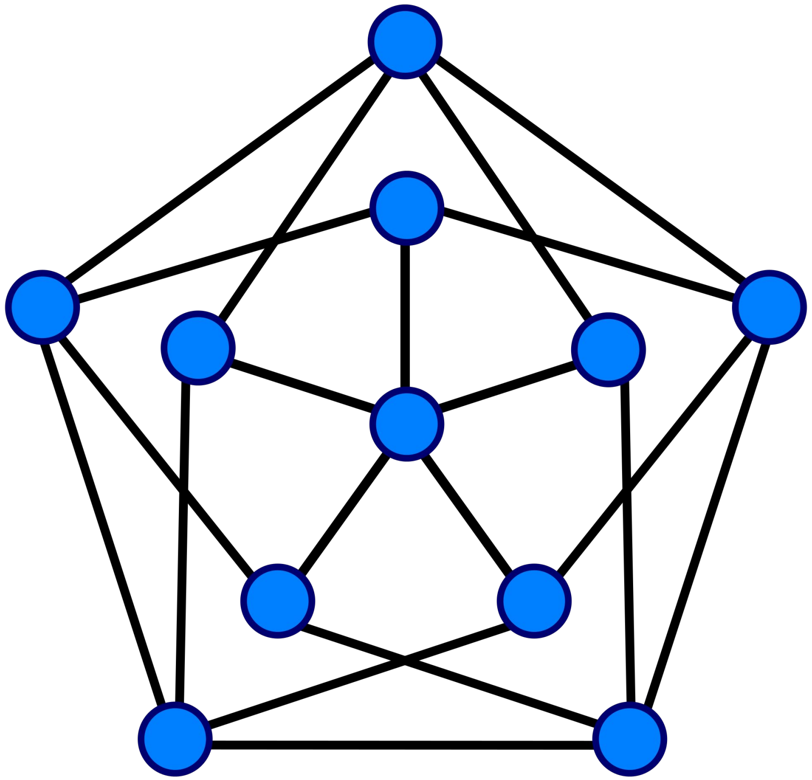


Distributed Computing



Distributed Algorithm

- The same algorithm on all Nodes

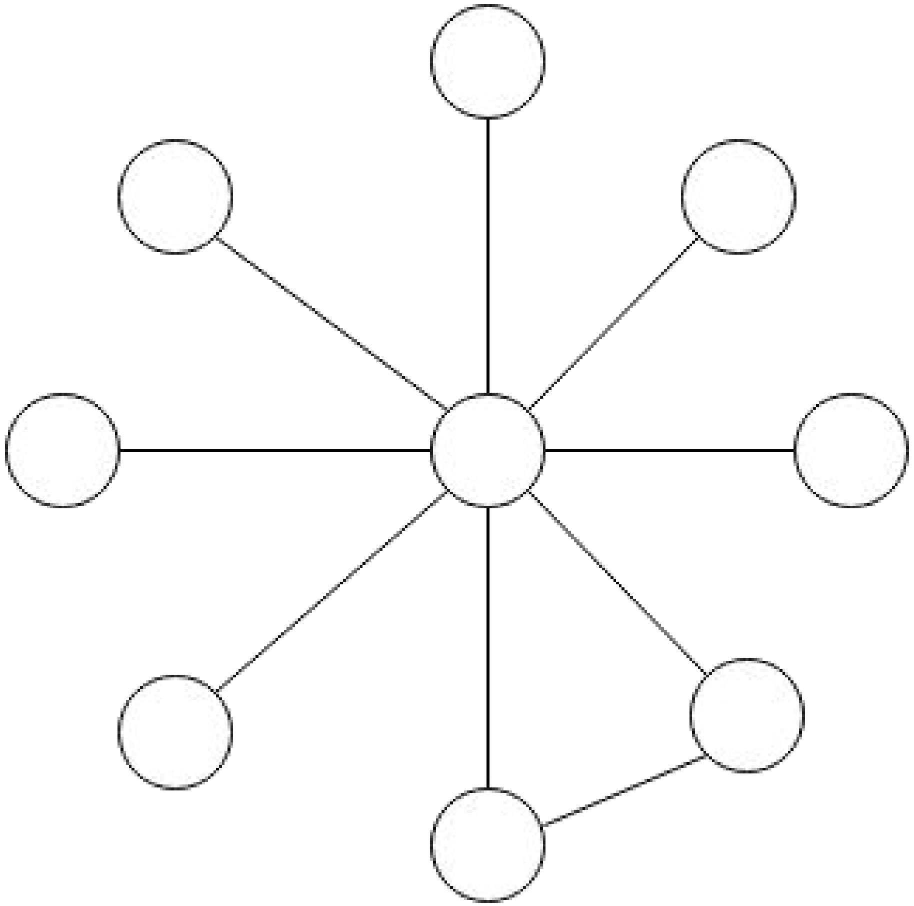


Complexity

- Global Complexity
  - All nodes with high probability (1 - 1/n)

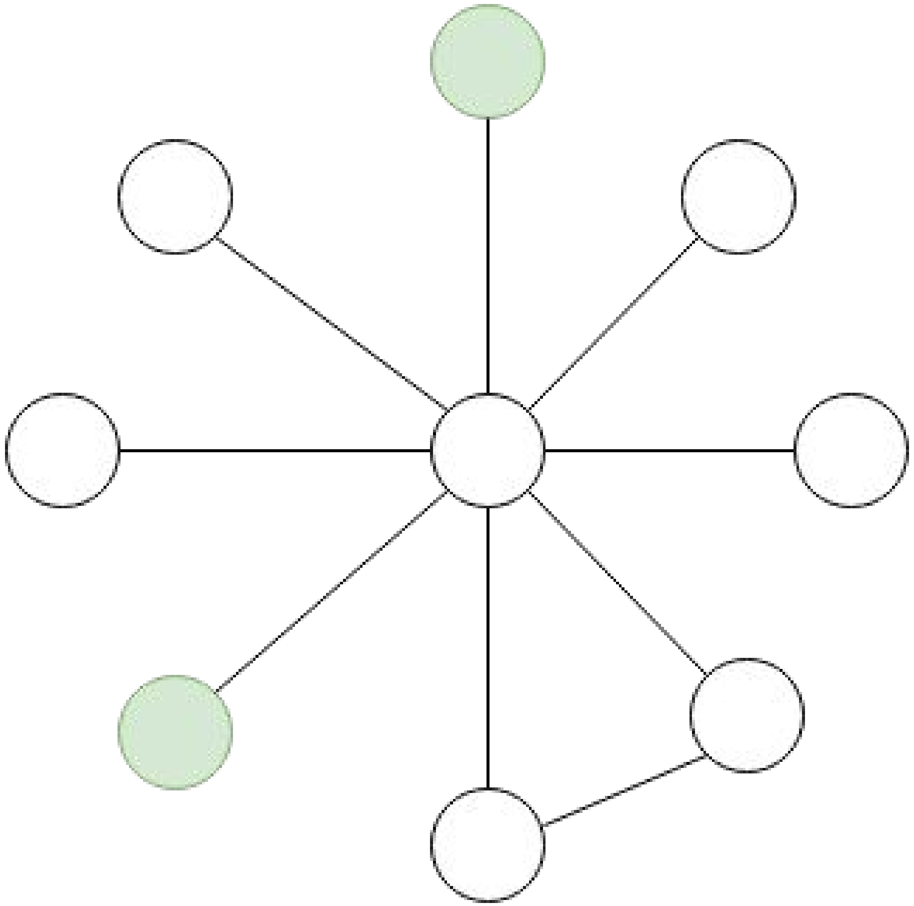
- Local Complexity
  - Node v with probability at least 1-ε

Maximal Independent Set



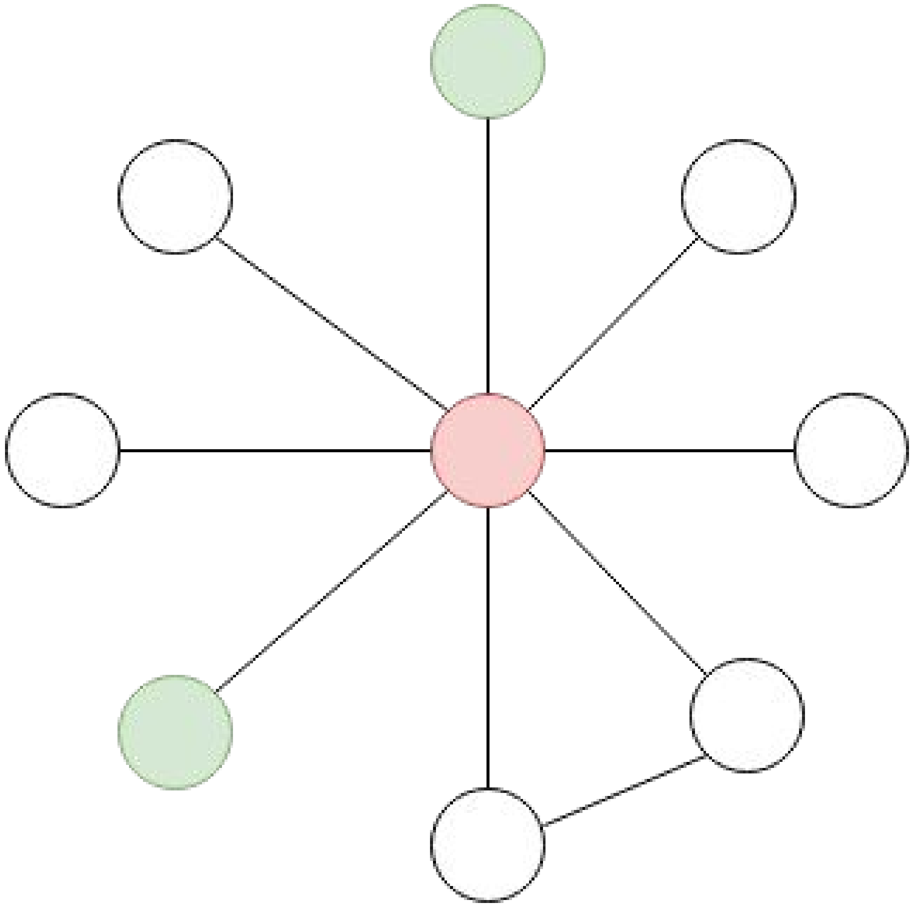
Maximal Independent Set

Independent Set



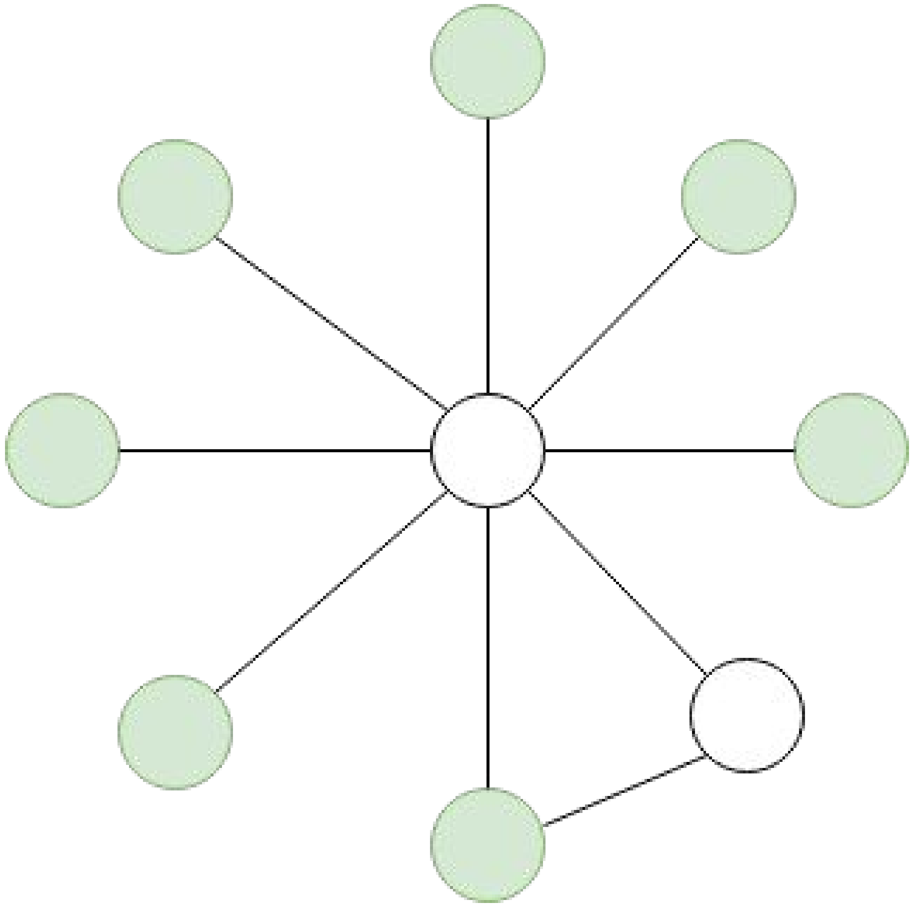
Maximal Independent Set

Not Independent Set



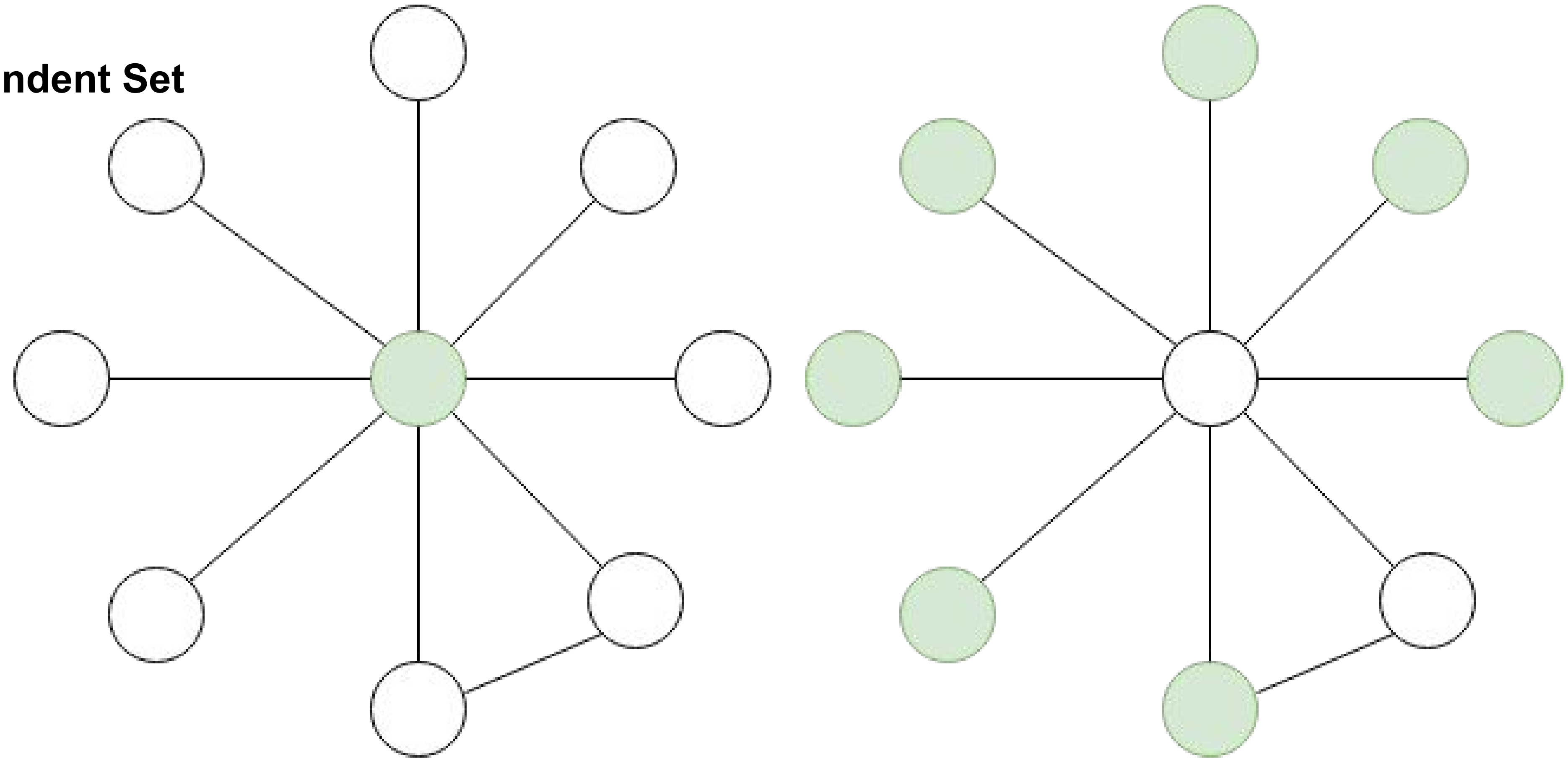
Maximal Independent Set

Maximum Independent Set



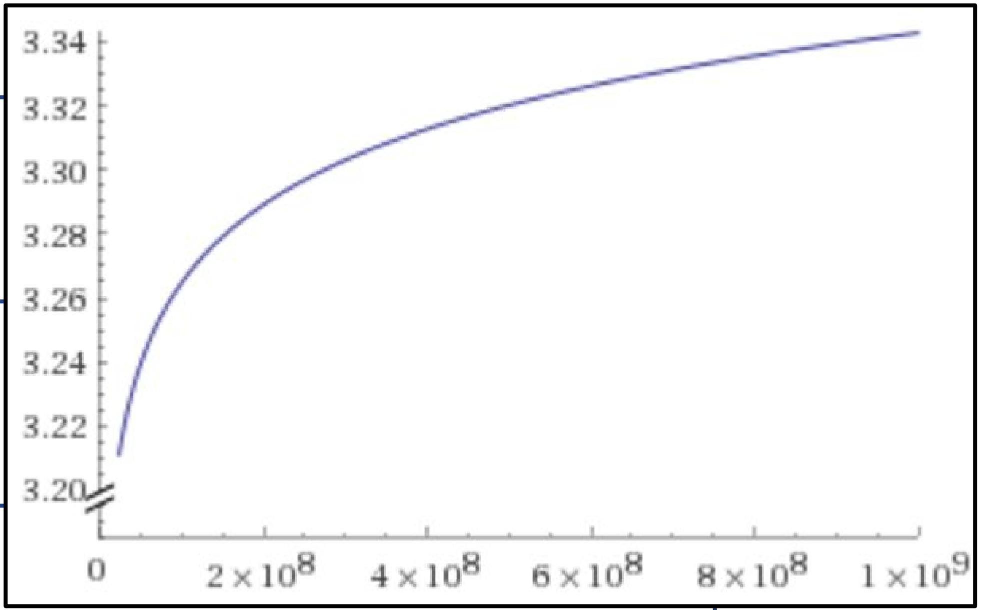
Maximal Independent Set

Maximal Independent Set



Distributed MIS

Kuhn et al's lower bound	
Luby	
Barenboim et al	
Ghaffari	$O(log \triangle) + 2^{O(\sqrt{log\ log\ n})}$



Distributed MIS

Lowerbound: $\Omega\,(min\{log\Delta, \sqrt{log\ n}\})$

If $log\ \Delta \in [2^{O(\sqrt{log\ log\ n})}, \sqrt{log\ n}\ ]$

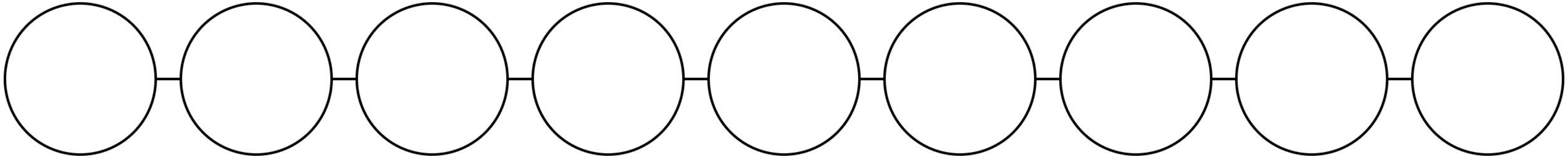
Lowerbound = $\Omega(log\ \Delta\ )$

$O(log\ \triangle) + 2^{O(\sqrt{log\ log\ n})} = O(log\ \Delta\ )$

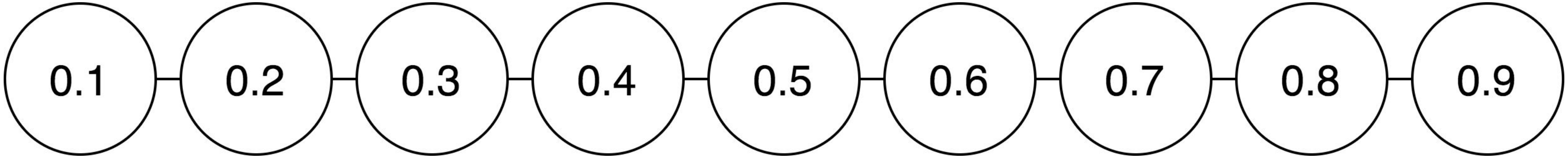
Luby's Algorithm

"In each round, each node picks a random number uniformly from [0,1]; strict local minimas join the MIS, and get removed from the graph along with their neighbours"

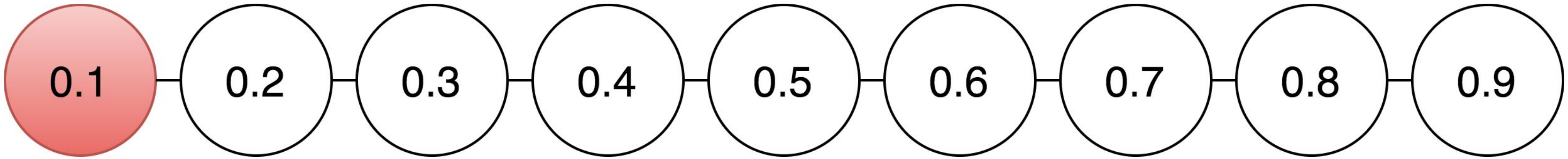
Luby’s Algorithm



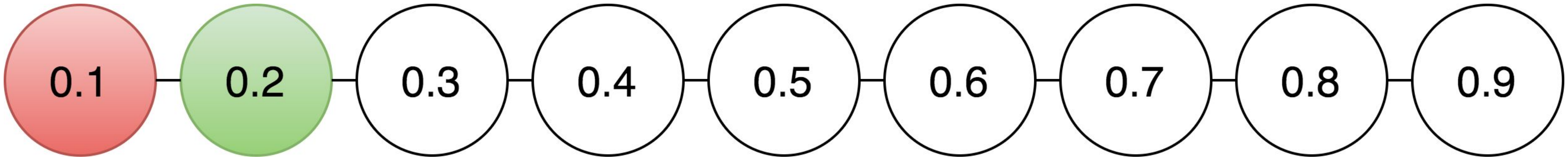
Luby's Algorithm



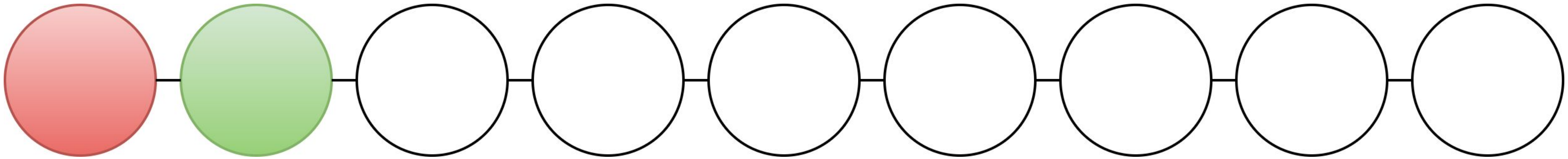
Luby's Algorithm



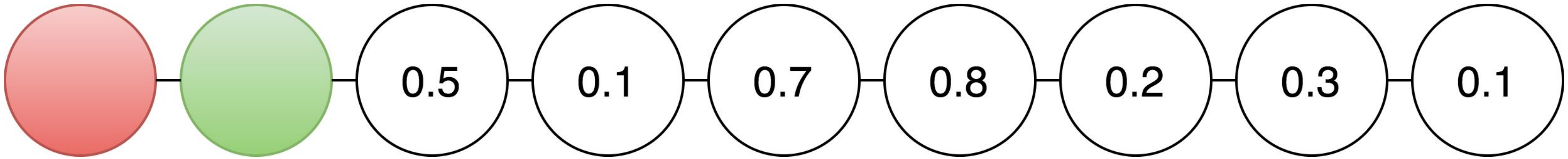
Luby's Algorithm



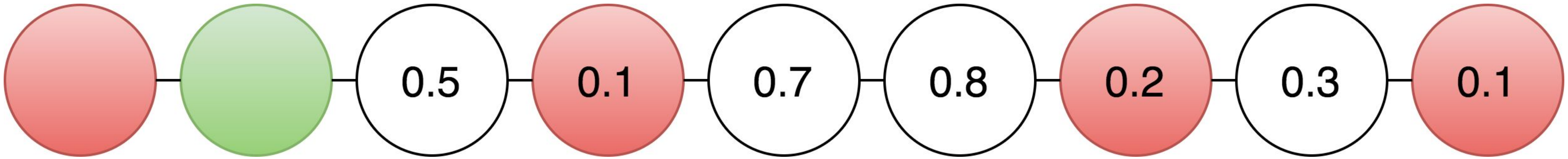
Luby’s Algorithm



Luby's Algorithm

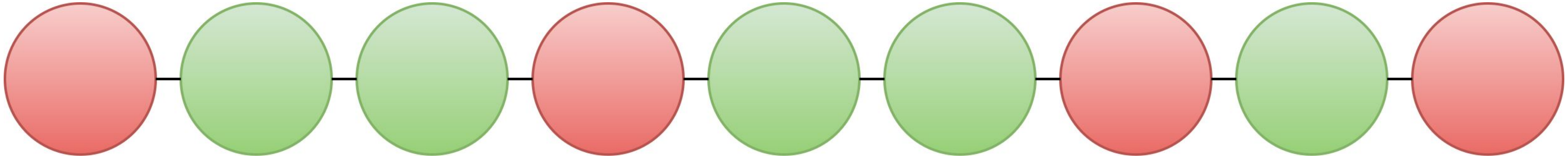


Luby's Algorithm



Luby's Algorithm

Luby’s Algorithm



Analysis

- Global complexity $O(log\ n)$ with probability $1 - \frac{1}{n}$

- Local Complexity $O(log^2 \Delta + log \frac{1}{\varepsilon})$ with probability $1 - \frac{1}{\varepsilon}$

Introduction to Distributed Computing

Maximal Independent Set

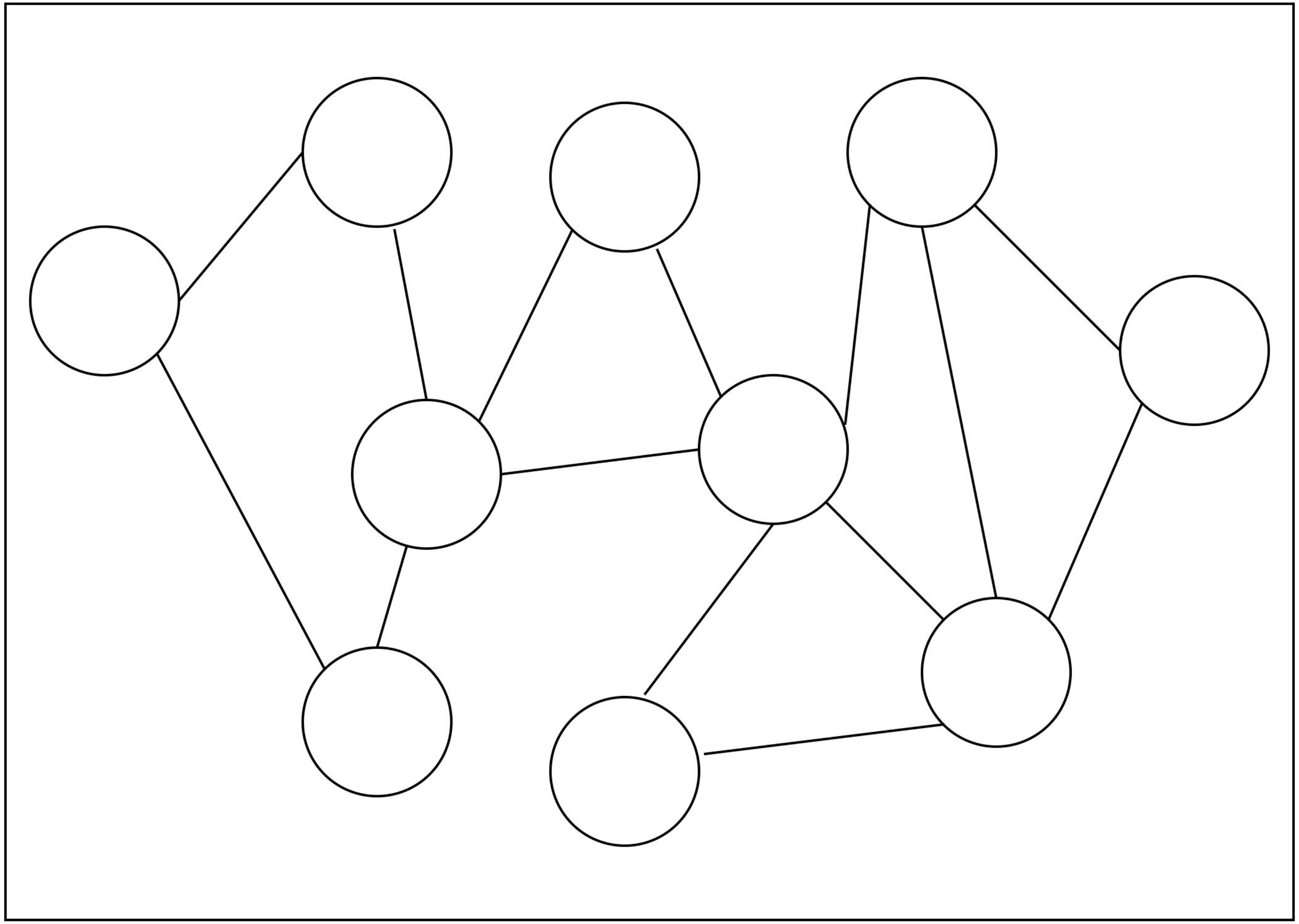
Luby’s Algorithm

Ghaffari’s Algorithm

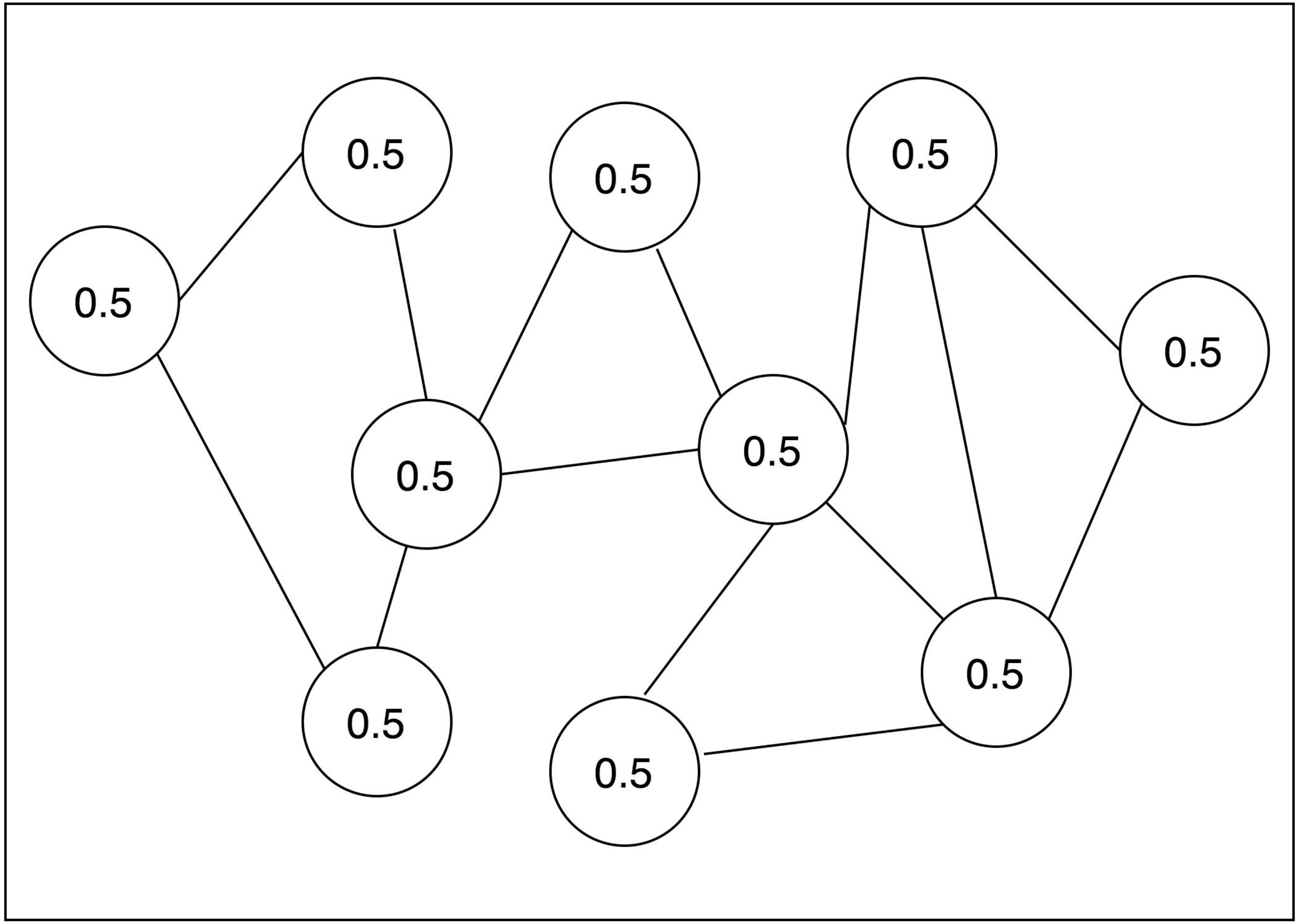
Local Complexity of Ghaffari’s Algorithm

Global complexity of Ghaffari’s Algorithm

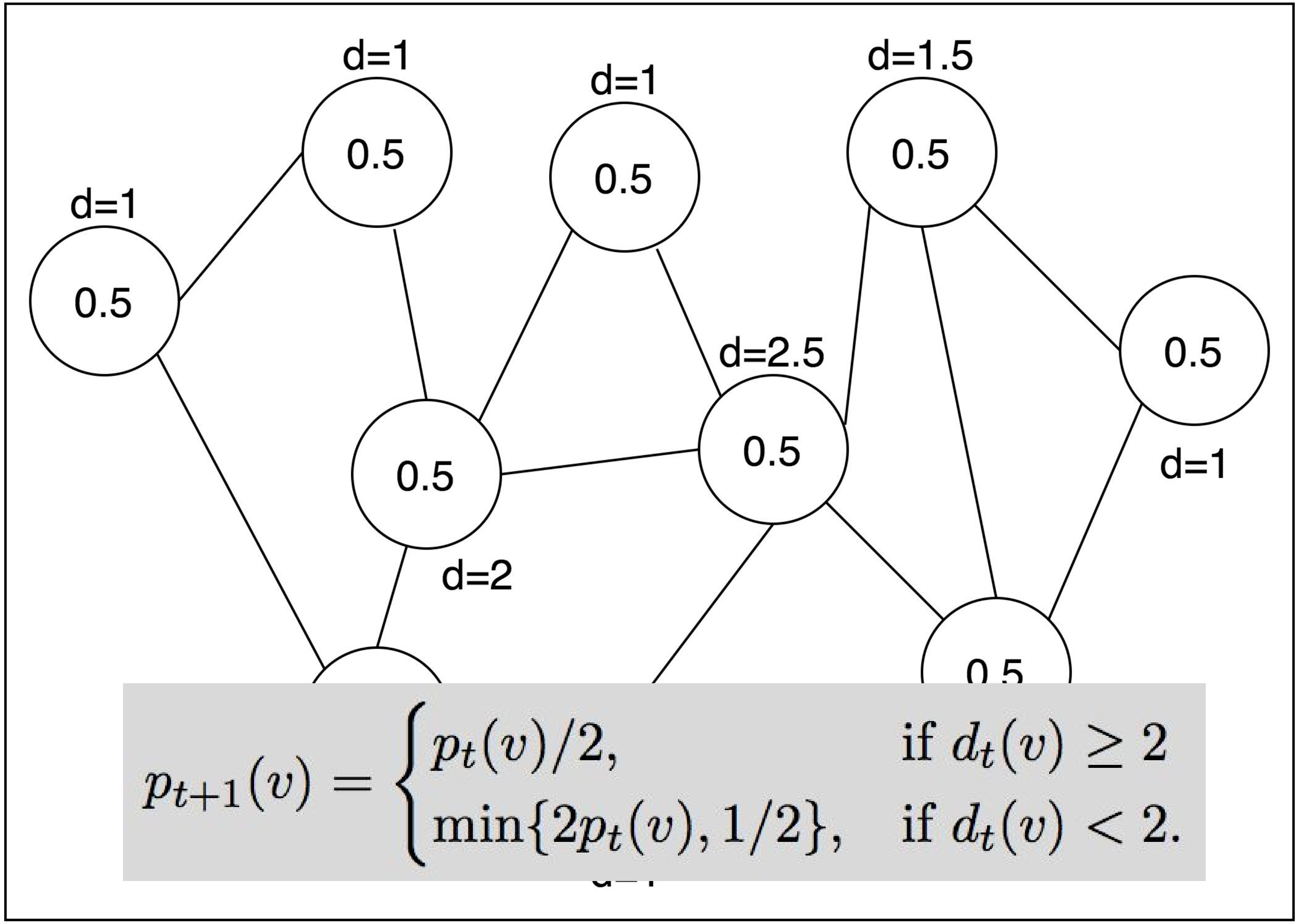
Ghaffari’s Algorithm



Ghaffari's Algorithm

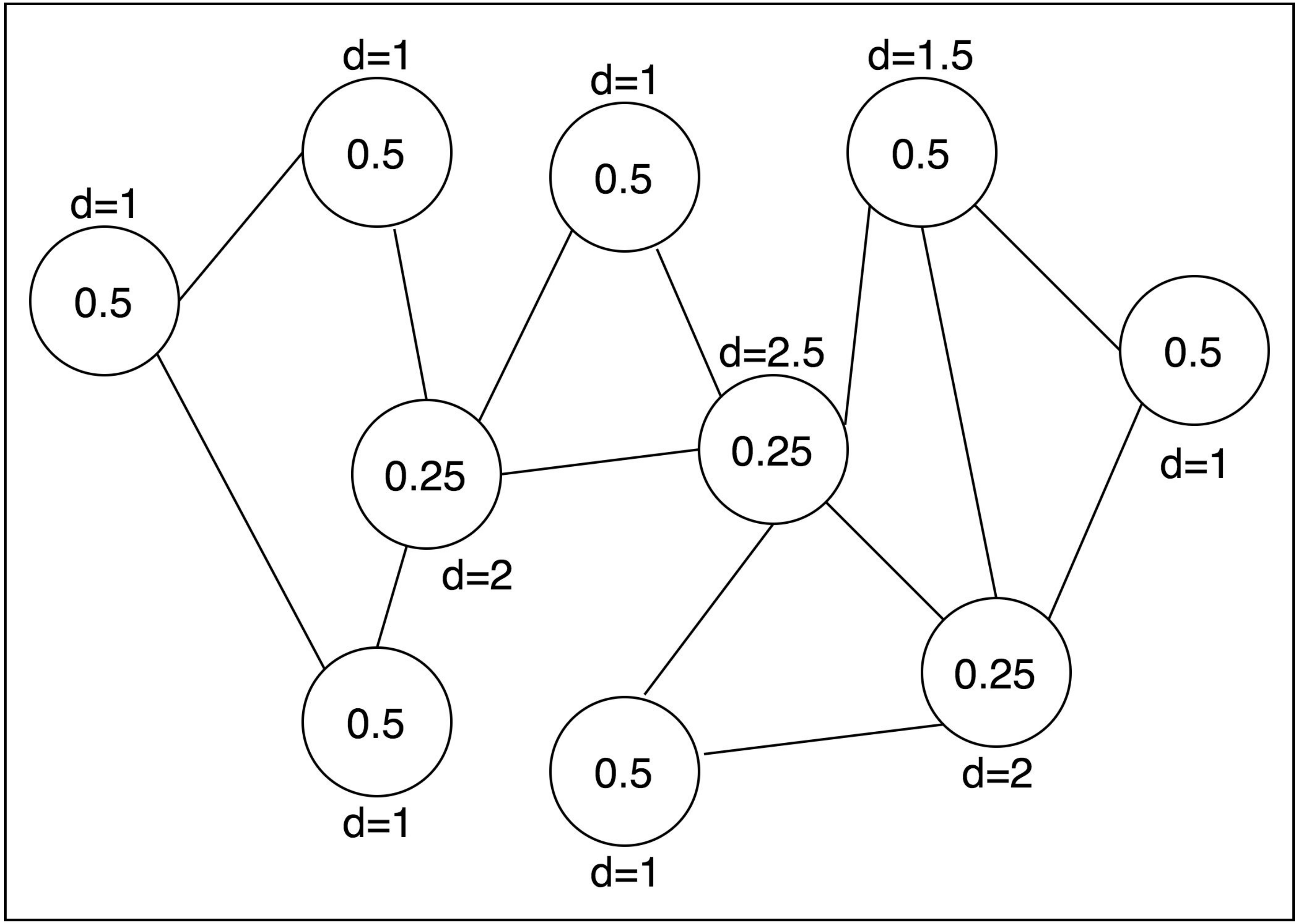


Ghaffari's Algorithm

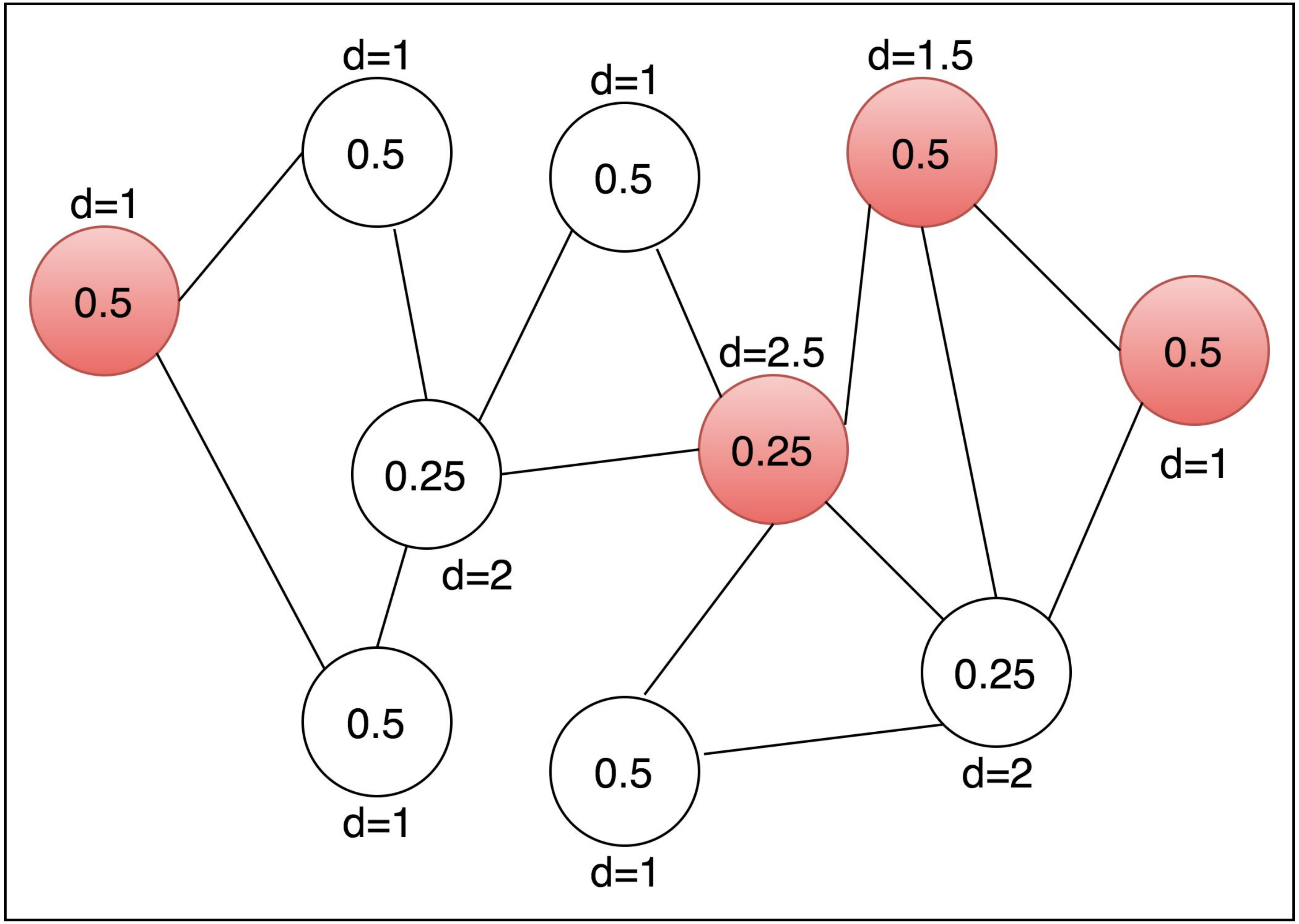


$$p_{t+1}(v) = \begin{cases} p_t(v)/2, & \text{if } d_t(v) \geq 2 \\ \min\{2p_t(v), 1/2\}, & \text{if } d_t(v) < 2. \end{cases}$$

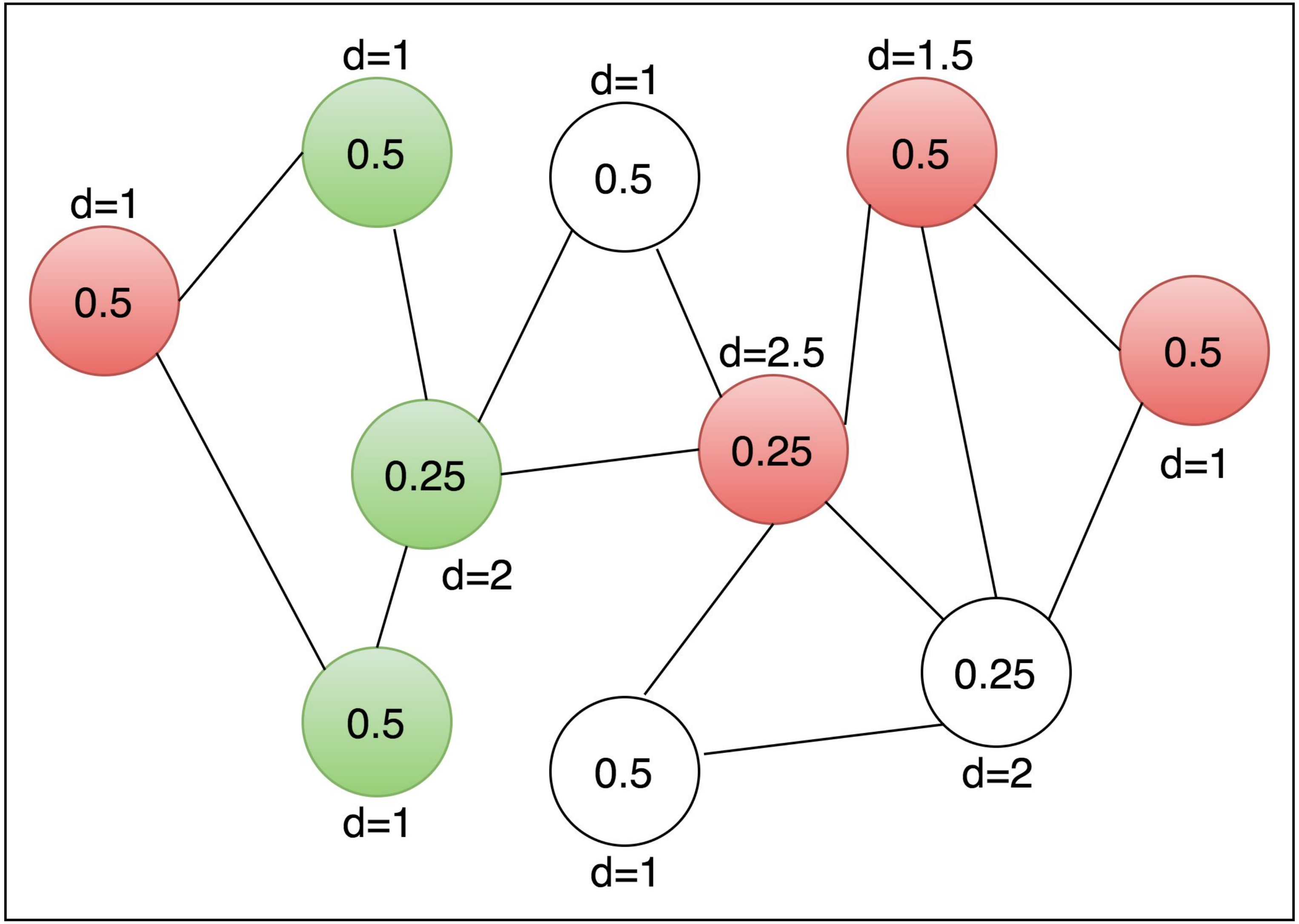
Ghaffari's Algorithm



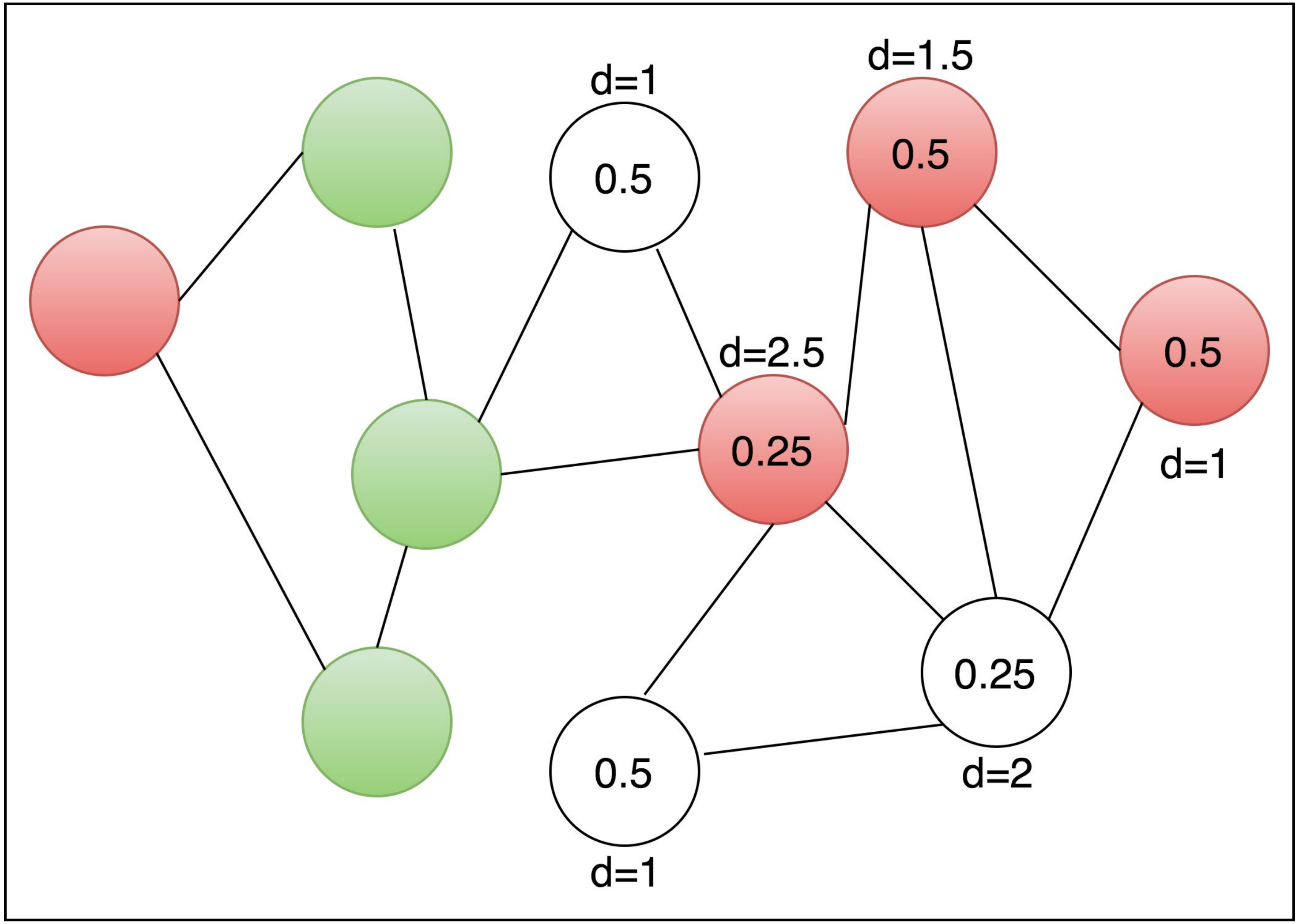
Ghaffari's Algorithm



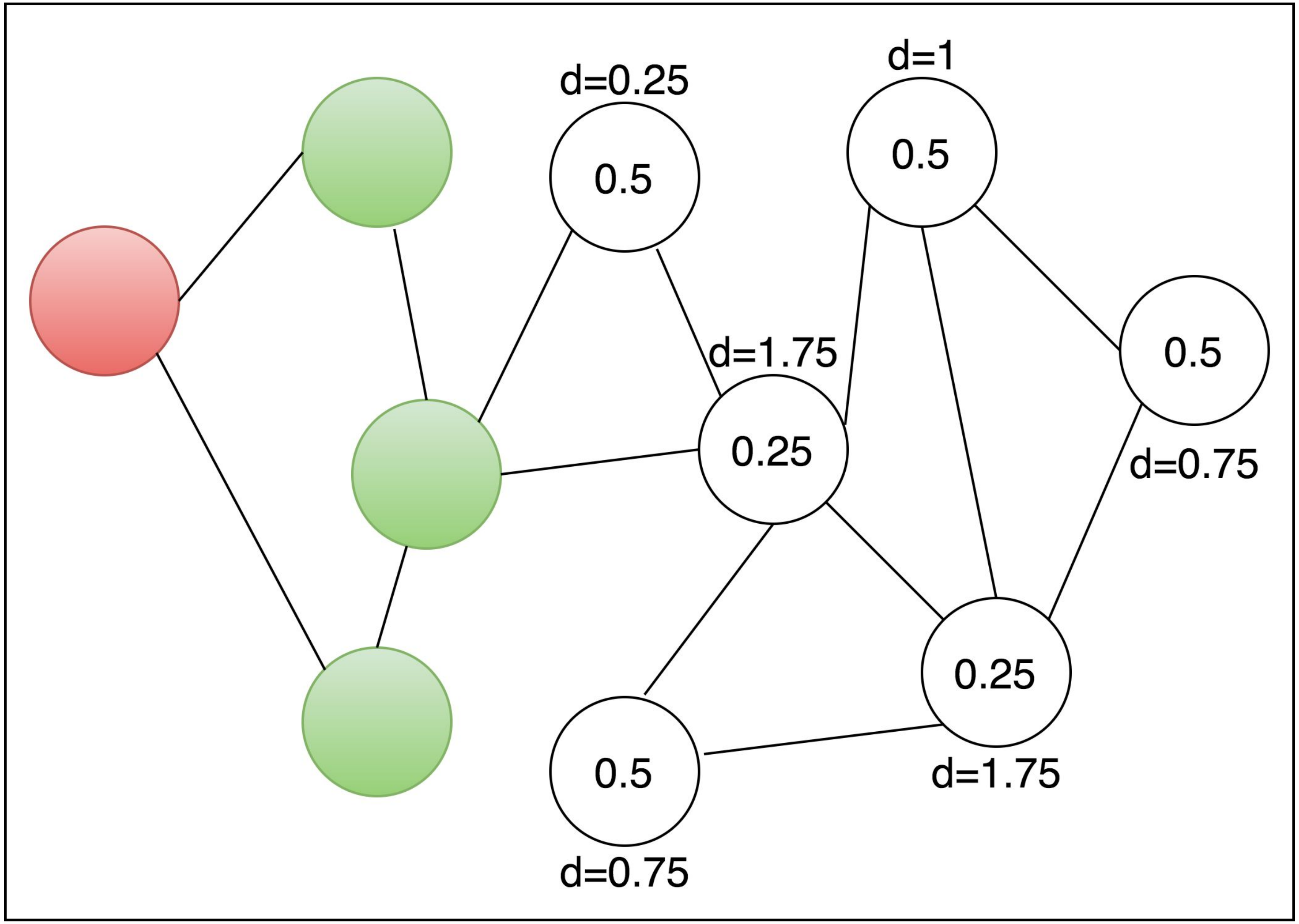
Ghaffari's Algorithm



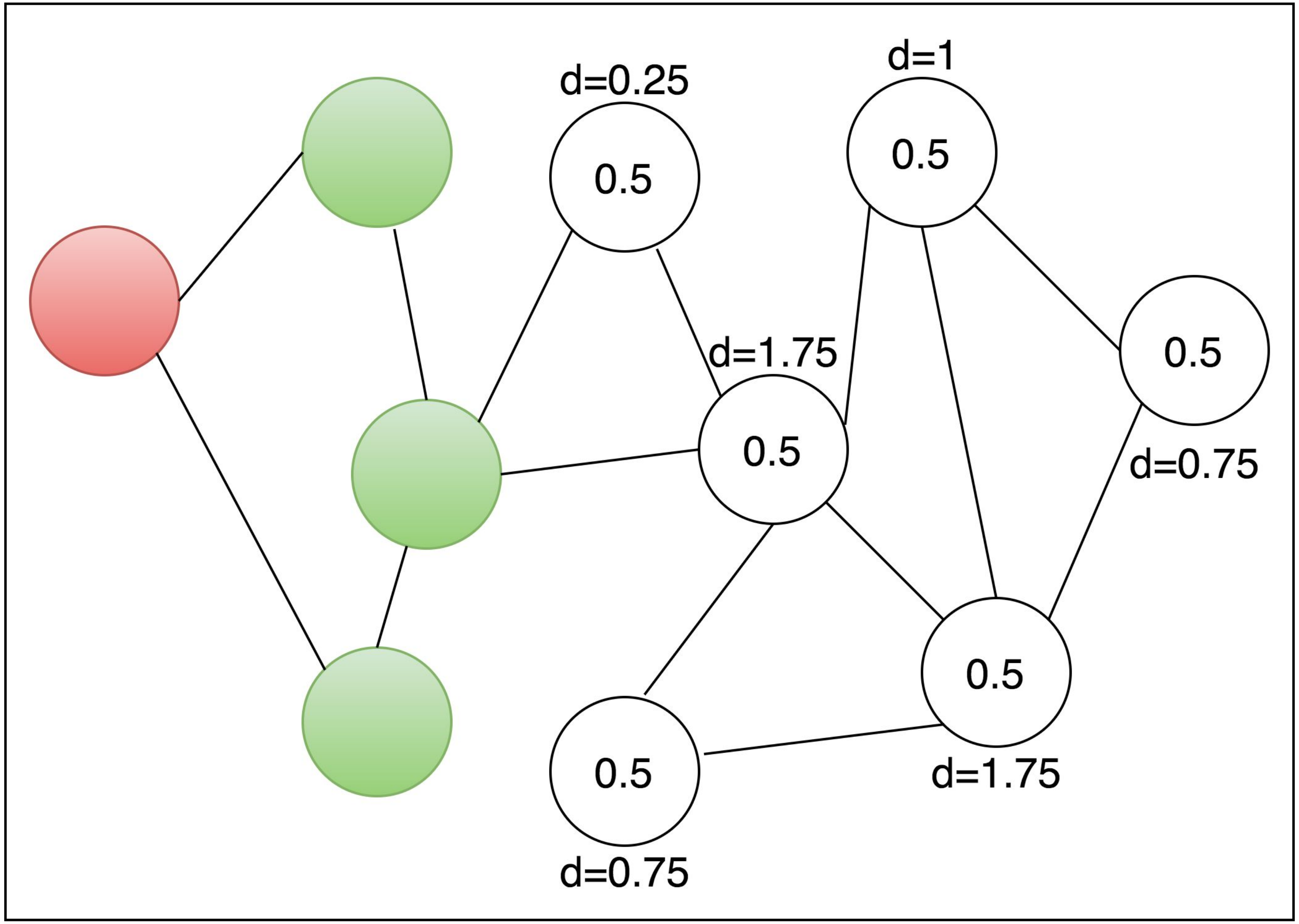
Ghaffari's Algorithm



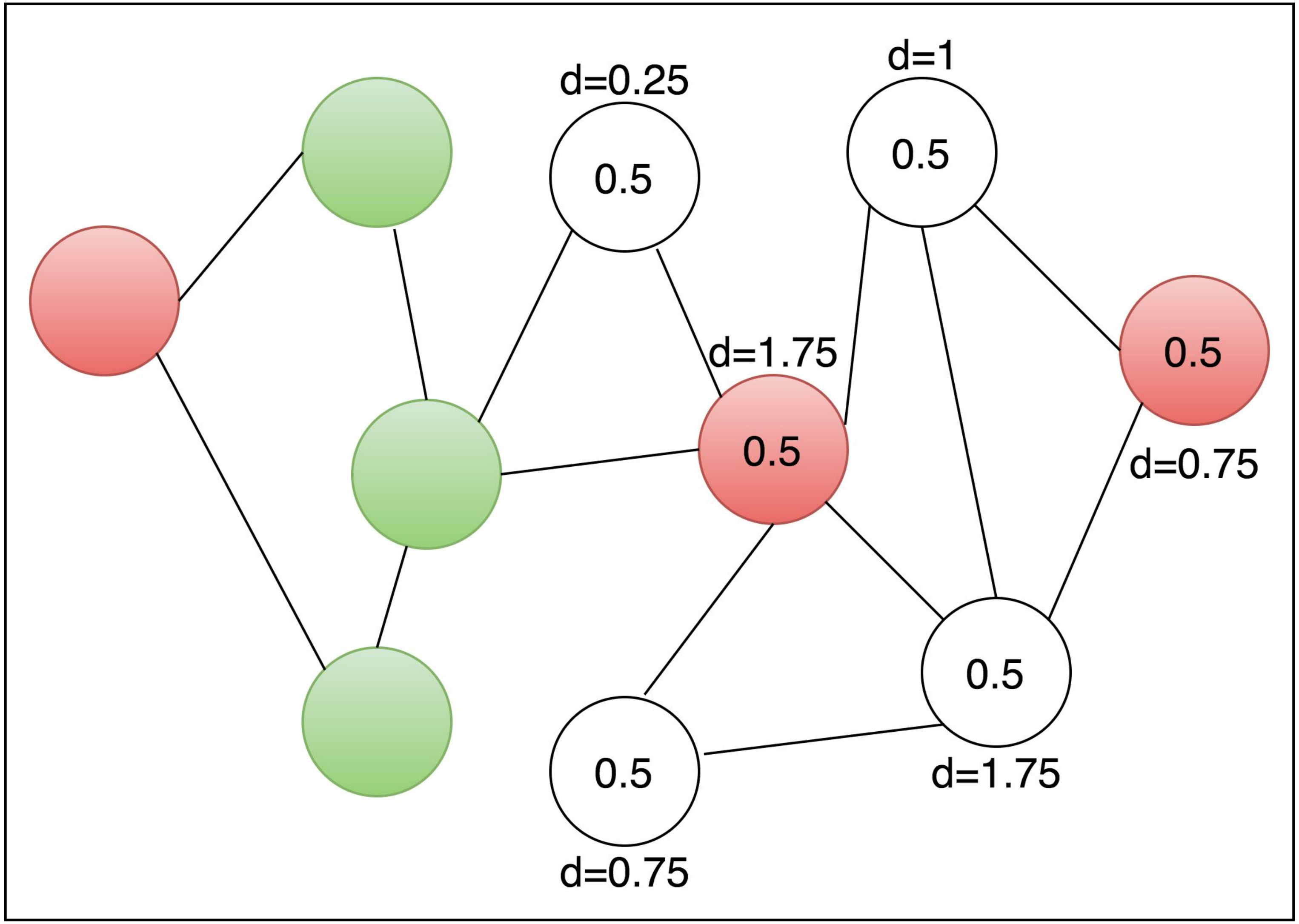
Ghaffari's Algorithm



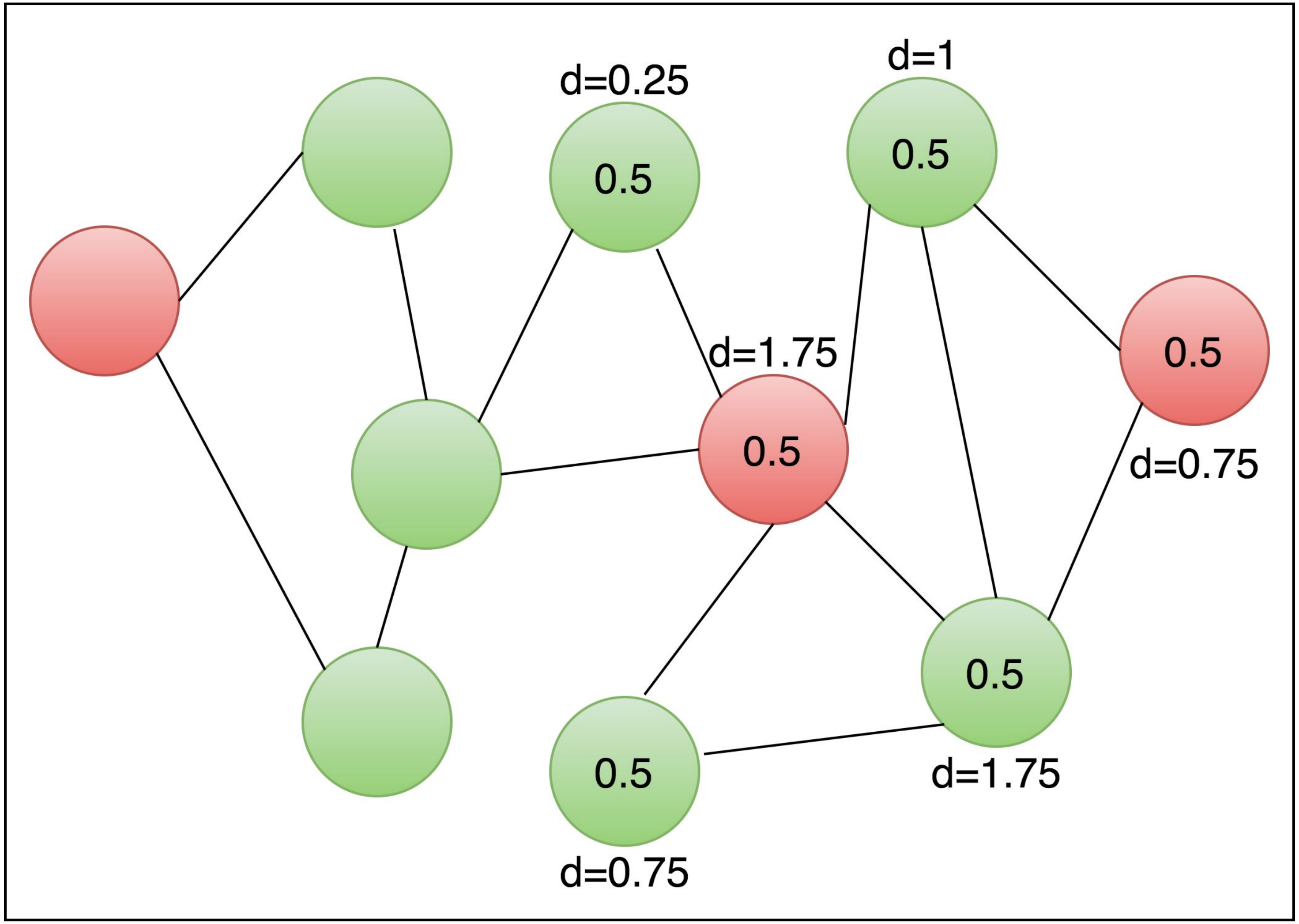
Ghaffari's Algorithm



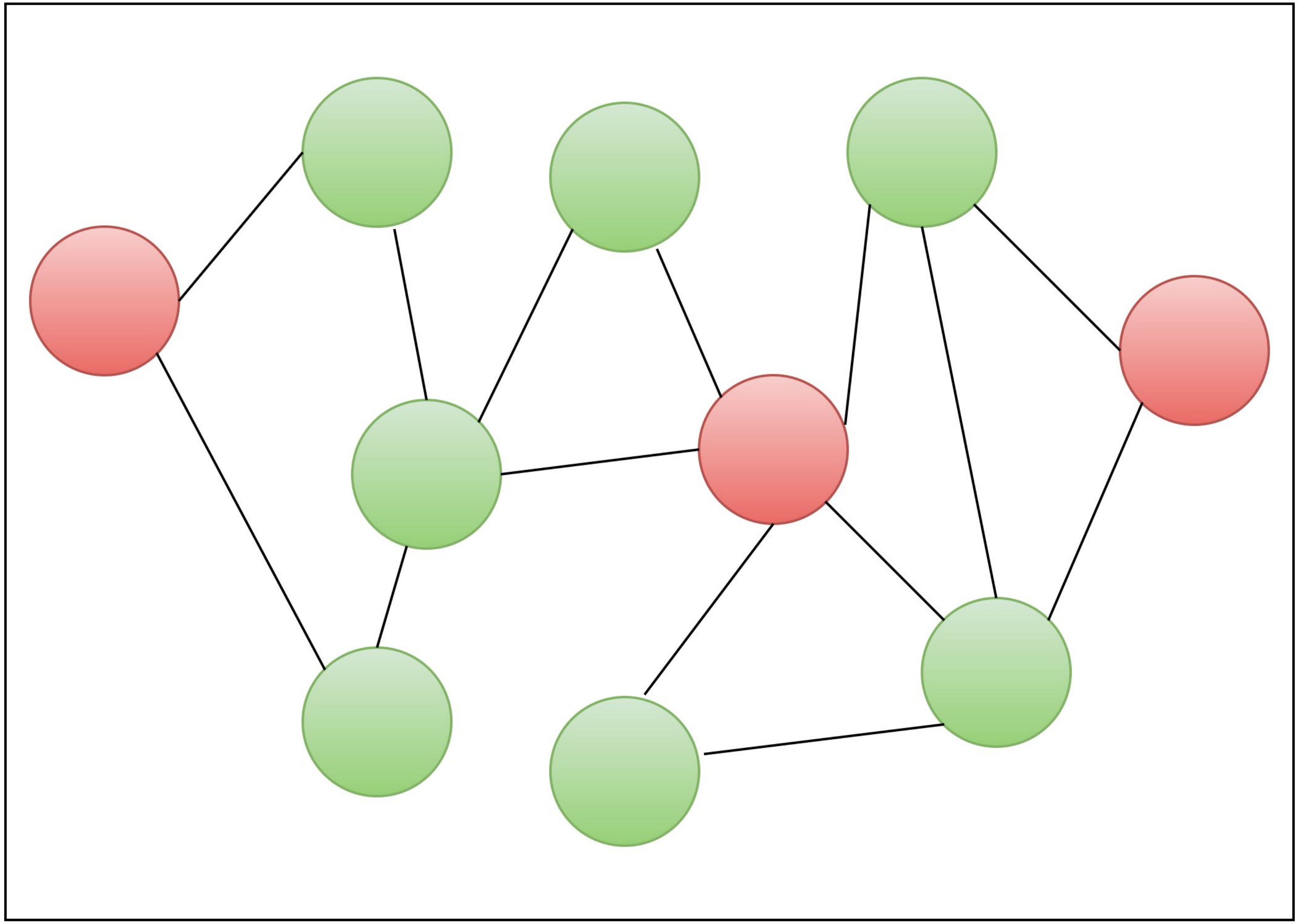
Ghaffari's Algorithm



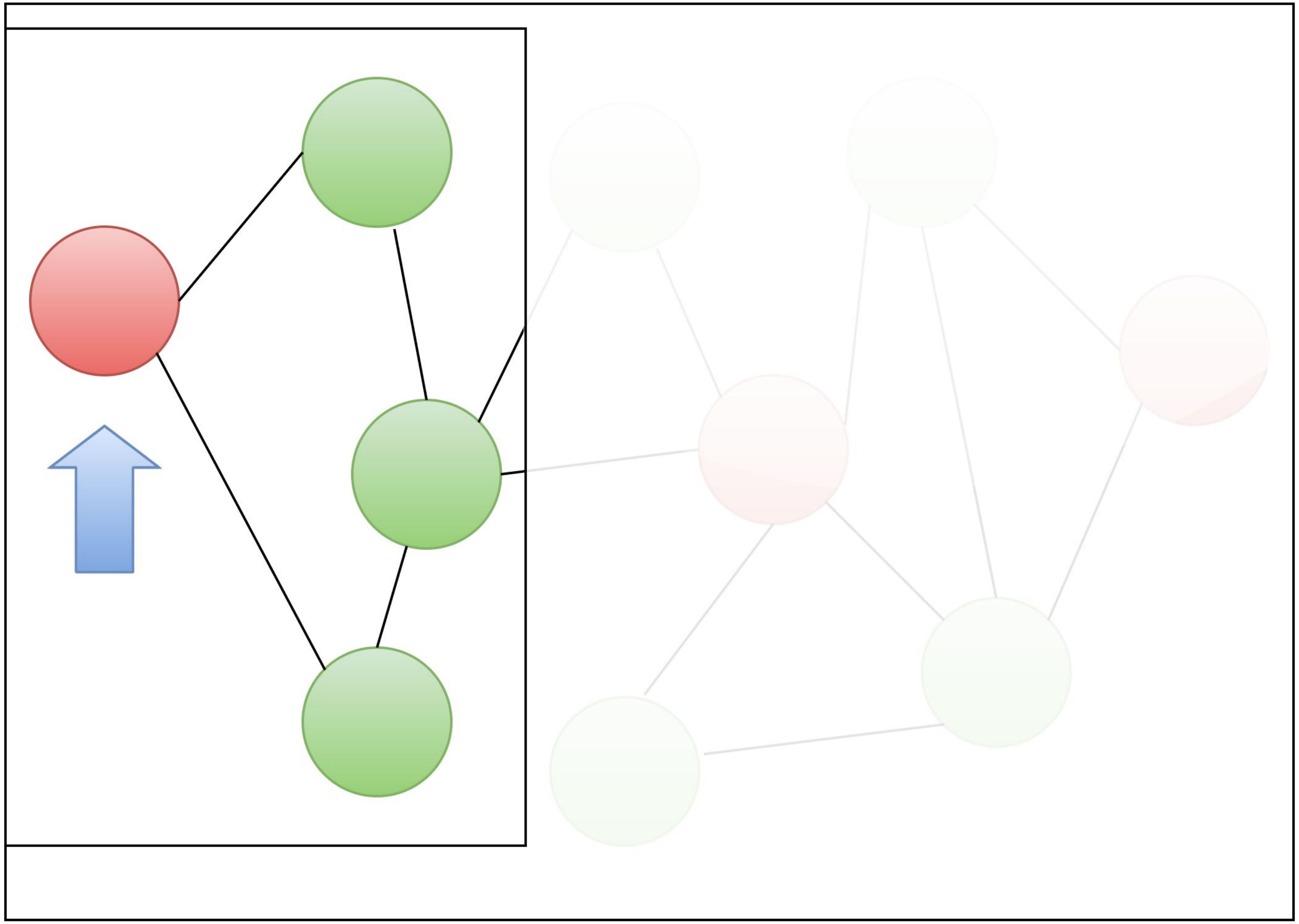
Ghaffari's Algorithm



Ghaffari's Algorithm



Ghaffari's Algorithm



ETH zürich

Introduction to Distributed Computing

Maximal Independent Set

Luby’s Algorithm

Ghaffari’s Algorithm

Local Complexity of Ghaffari’s Algorithm

Global complexity of Ghaffari’s Algorithm

Local Complexity

- For each node v, the probability that v has not made its decision in the first $\beta(log\ deg + log\ \frac{1}{\varepsilon})$ is at most ε

Local Complexity (cont.)

- Golden rounds
  a. rounds in which $d(v) < 2$ and $p(v) = ½$
  b. rounds in which $d(v) \geq 1$ and at least $d(v)/10$ of it is contributed by neighbors who have $d(v)<2$

$$\frac{1}{200}$$

- By round $\beta(log\ deg + log\ \frac{1}{\varepsilon})$, at least one of golden round counts of node v reached $\frac{\beta}{13}(log\ deg + log\ \frac{1}{\varepsilon})$

Local Complexity (cont.)

- Thus the probability that v does not get removed in the first $\beta(\log deg + \log \frac{1}{\varepsilon})$ steps is at most

$$\left(1 - \frac{1}{200}\right)^{\frac{\beta}{13}(\log deg + \log \frac{1}{\varepsilon})}$$

Introduction to Distributed Computing

Maximal Independent Set

Luby's Algorithm

Ghaffari's Algorithm

Local Complexity of Ghaffari's Algorithm

Global complexity of Ghaffari's Algorithm

Global complexity

- After $O(log\ \Delta)$ shattering phenomenon happens
- Deterministic Algorithm in small components

- The overall complexity is

$$O(\log \Delta) + 2^{O(\sqrt{\log \log n})}$$

Thank You!