An Improved Distributed Algorithm for Maximal Independent Set

Erfan Abdi
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
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 \(1-\varepsilon\)
Introduction to Distributed Computing

Maximal Independent Set

Luby’s Algorithm

Ghaffari’s Algorithm

Analysis of Ghaffari’s Algorithm

Global complexity of Ghaffari’s Algorithm
Maximal Independent Set
Maximal Independent Set

Independent Set
Maximal Independent Set

Not Independent Set
Maximal Independent Set

Maximum Independent Set
Maximal Independent Set
# Distributed MIS

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Kuhn et al’s lower bound</td>
<td>Luby</td>
</tr>
<tr>
<td>Barenboim et al</td>
<td>Ghaffari</td>
</tr>
</tbody>
</table>

\[
O(\log \Delta) + 2^{O(\sqrt{\log \log n})}
\]
Distributed MIS

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

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

\[ O(\log \Delta) + 2^{O(\sqrt{\log \log n})} = O(\log \Delta) \]
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
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

0.1 – 0.2 – 0.3 – 0.4 – 0.5 – 0.6 – 0.7 – 0.8 – 0.9
Luby’s Algorithm
Luby’s Algorithm

0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
Luby’s Algorithm
Luby’s Algorithm

0.5 0.1 0.7 0.8 0.2 0.3 0.1
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 \( 1 - \frac{1}{\varepsilon} \)
Ghaffari’s Algorithm
Ghaffari’s Algorithm
Ghaffari’s Algorithm

\[ p_{t+1}(v) = \begin{cases} 
\frac{p_t(v)}{2}, & \text{if } d_t(v) \geq 2 \\
\min\{2p_t(v), 1/2\}, & \text{if } d_t(v) < 2.
\end{cases} \]
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
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 $\varepsilon$
Local Complexity (cont.)

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

- By round $\beta (\log \text{deg} + \log \frac{1}{\epsilon})$, at least one of golden round counts of node $v$ reached $\frac{\beta}{13} (\log \text{deg} + \log \frac{1}{\epsilon})$
Thus the probability that $v$ does not get removed in the first $\beta(\log \text{deg} + \log \frac{1}{\varepsilon})$ steps is at most

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