Analysis of Distributed Randomized Algorithms

Akka Zemmari

LaBRI - Université Bordeaux 1

Séminaire LIF
Apr. 05, 2012
Outline

- The Model
- Distributed (Randomized) Algorithms
- The MIS Problem
- The Colouring Problem
  - The Algorithm
  - A Lower Bound
A distributed system is an interconnected collection of autonomous computers, processes, or processors. The computers, processes, or processors are referred to as the nodes of the distributed system.

- to be qualified as autonomous, the nodes must at least be equipped with their own private control,
- to be qualified as interconnected, the nodes must be able to exchange information.
A distributed algorithm for a collection \( P = \{p_1, p_2, \ldots, p_n\} \) of processes is a collection of local algorithms, one for each process in \( P \).
The Model

Randomized Algorithm [Motwani & Raghavan]
Randomized Algorithm [Motwani & Raghavan]

- In addition to input, algorithm takes a source of random numbers and makes random choices during execution,
- Behavior can vary even on a fixed input.
Design algorithm + analysis to show that this behavior is likely to be good, on every input. (The likelihood is over the random numbers only.)
The Model

Hypothesis
- Anonymous networks
- No global knowledge

(Example of) Impossibility results [Angluin]
- There is no deterministic algorithm that solves the election problem in anonymous networks.
Randomized distributed algorithms

## The Model

### Definitions

- **Time Complexity.** [1] A round (cycle) of each processor is composed of the following three steps: 1. Send messages to (some of) the neighbours, 2. Receive messages from (some of) the neighbours, 3. Perform some local computation. As usual, the time complexity is the maximum possible number of rounds needed until every node has completed its computation.

- **Bit Complexity.** [2] A bit round is a round such that each processor can send/receive at most 1 bit to/from each neighbour. The bit complexity of an algorithm $A$ is the number of bit rounds to complete algorithm $A$.


The Model

Definitions

- **Time Complexity.** [1] A round (cycle) of each processor is composed of the following three steps: 1. Send messages to (some of) the neighbours, 2. Receive messages from (some of) the neighbours, 3. Perform some local computation. As usual, the time complexity is the maximum possible number of rounds needed until every node has completed its computation.

- **Bit Complexity.** [2] A bit round is a round such that each processor can send/receive at most 1 bit to/from each neighbour. The bit complexity of an algorithm $\mathcal{A}$ is the number of bit rounds to complete algorithm $\mathcal{A}$.

References:


Let $G = (V, E)$ be a graph. An independent set is a subset $U$ of $V$ s.t.,

$$\forall u, v \in U, \{u, v\} \notin E,$$

$U$ is maximal if $\forall v \in V \setminus U, U \cup \{v\}$ is not an independent set.
<table>
<thead>
<tr>
<th>Knowledge</th>
<th>Time (on average)</th>
<th>Message size (number of bits)</th>
<th>Bit complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luby (Lynch)</td>
<td>$O(\log n)$</td>
<td>$\log n$</td>
<td>$O(\log^2 n)$</td>
</tr>
<tr>
<td>Luby (Peleg)</td>
<td>$O(\log^2 n)$</td>
<td>$\log n$</td>
<td>$O(\log^3 n)$</td>
</tr>
<tr>
<td>Luby (Wattenhofer)</td>
<td>$O(\log n)$</td>
<td>$\log n$</td>
<td>$O(\log^2 n)$</td>
</tr>
<tr>
<td>Alon, Babai and Itai</td>
<td>$O(\log n)$</td>
<td>$\log n$</td>
<td>$O(\log^2 n)$</td>
</tr>
<tr>
<td>Métivier et al. [1]</td>
<td>no knowledge</td>
<td>$O(\log n)$</td>
<td>$\Omega(\log n)$</td>
</tr>
<tr>
<td>Métivier et al. [2]</td>
<td>no knowledge</td>
<td>1</td>
<td>$\Omega(\sqrt{\log n})$</td>
</tr>
<tr>
<td>Métivier et al. [3]</td>
<td></td>
<td>1</td>
<td>$O(\sqrt{\log n})$ in ring graphs.</td>
</tr>
</tbody>
</table>

Definition.

Let $G = (V, E)$ be a simple connected undirected graph. A proper vertex colouring for $G$ is an assignment of a colour $c(v)$ to each vertex $v$, such that any two adjacent vertices have a different colour, i.e., $c(v) \neq c(u)$ for every $\{u, v\} \in E$. 
### Related Works

<table>
<thead>
<tr>
<th>Time</th>
<th>Bit Complexity</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Johansson [1]</td>
<td>$O(\log n)$</td>
<td>$O(\log^2 n)$</td>
</tr>
<tr>
<td>Kothapalli et al. [2]</td>
<td>$\Omega(\log n)$</td>
<td>no restriction on the number of colours</td>
</tr>
<tr>
<td>Métivier et al. [3]</td>
<td>$O(\log n)$</td>
<td>$O(\log n)$</td>
</tr>
<tr>
<td>Métivier et al. [4]</td>
<td>$\leq 2 \log n + o(\log n)$</td>
<td>$&gt; 2 \log n$</td>
</tr>
</tbody>
</table>

## Related Works

<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Bit Complexity</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Johansson [1]</td>
<td>$O(\log n)$</td>
<td>$O(\log^2 n)$</td>
<td>$\Delta + 1$ colours</td>
</tr>
<tr>
<td>Kothapalli et al. [2]</td>
<td>$\Omega(\log n)$</td>
<td></td>
<td>no restriction on the number of colours</td>
</tr>
<tr>
<td>Métivier et al. [3]</td>
<td>$O(\log n)$</td>
<td>$O(\log n)$</td>
<td>$\Delta + 1$ colours</td>
</tr>
<tr>
<td>Métivier et al. [4]</td>
<td>$\leq 2 \log n + o(\log n)$</td>
<td>$&gt; 2 \log n$</td>
<td>$3$—colouring of ring graphs</td>
</tr>
</tbody>
</table>

## Related Works

<table>
<thead>
<tr>
<th>Time</th>
<th>Bit Complexity</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>O (log n)</td>
<td>O(log² n)</td>
<td>Δ + 1 colours</td>
</tr>
<tr>
<td>Kothapalli et al. [2]</td>
<td>Ω(log n)</td>
<td>no restriction on the number of colours</td>
</tr>
<tr>
<td>Métivier et al. [3]</td>
<td>O(log n)</td>
<td>Δ + 1 colours</td>
</tr>
<tr>
<td>Métivier et al. [4]</td>
<td>≤ 2 log n + o (log n)</td>
<td>3—colouring of ring graphs</td>
</tr>
</tbody>
</table>

## Related Works

<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Bit Complexity</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Johansson [1]</td>
<td>$O(\log n)$</td>
<td>$O(\log^2 n)$</td>
<td>$\Delta + 1$ colours</td>
</tr>
<tr>
<td>Kothapalli et al. [2]</td>
<td>$\Omega(\log n)$</td>
<td></td>
<td>no restriction on the number of colours</td>
</tr>
<tr>
<td>Métivier et al. [3]</td>
<td>$O(\log n)$</td>
<td>$O(\log n)$</td>
<td>$\Delta + 1$ colours</td>
</tr>
<tr>
<td>Métivier et al. [4]</td>
<td>$\leq 2 \log n + o(\log n)$</td>
<td>$&gt; 2 \log n$</td>
<td>3—colouring of ring graphs</td>
</tr>
</tbody>
</table>


---

Akka Zemmari  | Randomized Algorithms

---

- Related Works
- Table of distributed graph colouring algorithms
- References
- Conclusion
<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Bit Complexity</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Johansson [1]</td>
<td>$O(\log n)$</td>
<td>$O(\log^2 n)$</td>
<td>$\Delta + 1$ colours</td>
</tr>
<tr>
<td>Kothapalli et al. [2]</td>
<td>$\Omega(\log n)$</td>
<td>no restriction on the number of colours</td>
<td></td>
</tr>
<tr>
<td>Métivier et al. [3]</td>
<td>$O(\log n)$</td>
<td>$O(\log n)$</td>
<td>$\Delta + 1$ colours</td>
</tr>
<tr>
<td>Métivier et al. [4]</td>
<td>$\leq 2 \log n + o(\log n)$</td>
<td>$&gt; 2 \log n$</td>
<td>3—colouring of ring graphs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Related Works</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time</strong></td>
</tr>
<tr>
<td>Johansson [1]</td>
</tr>
<tr>
<td>Kothapalli et al. [2]</td>
</tr>
<tr>
<td>Métivier et al. [3]</td>
</tr>
<tr>
<td>Métivier et al. [4]</td>
</tr>
</tbody>
</table>

Algorithm *Fast Colour*
Algorithm \textit{Fast\_Colour}

\begin{align*}
\text{0011} & \quad \text{011} \\
\text{010} & \quad \text{110} \\
\text{1011} & \quad \text{000} \\
\text{001} & \quad \text{100} \\
\text{111} & \quad \text{101} \\
\end{align*}
Algorithm *Fast Colour*

![Algorithm diagram](image-url)
Algorithm **Fast\_Colour**

![Graph illustration](image-url)
Algorithm *Fast\_Colour*
Algorithm \textit{Fast\_Colour}
Algorithm *Fast\_Colour*

![Graph Diagram](image-url)
**Lemma.**
In any phase of the algorithm, the expected number of edges removed from the residual graph $G$ is half the number of its edges.

**Corollary.**
There are constants $k_1$ and $K_1$ such that for any graph $G$ of $n$ vertices, the number of phases to remove all edges from $G$ is:
- less than $k_1 \log n$ on average,
- less than $K_1 \log n$ w.h.p.
Lemma.
In any phase of the algorithm, the expected number of edges removed from the residual graph $G$ is half the number of its edges.

Corollary.
There are constants $k_1$ and $K_1$ such that for any graph $G$ of $n$ vertices, the number of phases to remove all edges from $G$ is:

- less than $k_1 \log n$ on average,
- less than $K_1 \log n$ w.h.p.
Analysis of the Algorithm

Theorem.

Algorithm *Fast*\_*Colour* computes a colouring for any arbitrary graph of size $n$ in time $O(\log n)$ w.h.p., each message containing 1 bit.
Reducing the Number of Colors - Algorithm $\mathcal{R}$
Reducing the Number of Colors - Algorithm $\mathcal{R}$
Reducing the Number of Colors - Algorithm \( \mathcal{R} \)
Reducing the Number of Colors - Algorithm $\mathcal{R}$
Reducing the Number of Colors - Algorithm \( \mathcal{R} \)
Reducing the Number of Colors - Algorithm $\mathcal{R}$
Analysis of the Algorithm

Theorem

For any graph $G = (V, E)$ having a maximum degree $\Delta$, Algorithm $R$ achieves a $(\Delta + 1)$-colouring of $G$ in at most $e\Delta + 2 \log n + \log^* n$ rounds w.h.p.
Theorem

Let $G = (V, E)$ be a ring of size $n \geq 3$. Let $T_1$ denote the number of rounds necessary to colour all vertices by the colouring algorithm.

- The expected value of $T_1$ is asymptotically equal to $\log_2 n - \frac{1}{2} + \frac{\gamma}{\log 2} + Q (\log_2 n) + O (n^{-2})$, where $Q (u) = -\frac{1}{\log 2} \sum_{k \in \mathbb{Z} \setminus \{0\}} \Gamma \left( \frac{2ik\pi}{\log 2} \right) e^{-2ik\pi u}$ is a Fourier series with period 1 and with an amplitude which does not exceed $10^{-6}$.

- It is less than $2 \log_2 n$ w.h.p.

Lemma

The time complexity of the decreasing colours algorithm is less than $e + 2 \log_2 n$ w.h.p.
Theorem

The total time for the 3-colouring algorithm is $\log_2 n + o(\log n)$ on average and $2\log_2 n + o(\log n)$ w.h.p.
A Las Vegas algorithm $A$ which computes a proper colouring on rings takes time at least $\log_2 n - o(\log n)$ on average.

No Las Vegas algorithm $A$ which computes a proper colouring on rings runs in time $k \log_2 n$ with high probability for any $k < 2$. 
Formal proof of distributed randomised algorithms (joint work with A. Fontaine and P. Castéran)
- Formal modelisation in Coq.
- Proving correctness of such algorithms.
Thank You