Two Monte-Carlo Election Algorithms for Anonymous Graphs without any Initial Knowledge

Y. Métivier, J.M. Robson, A. Zemmari

LaBRI - Université Bordeaux 1

GT Algo. Dist.
Jan. 14, 2013
Outline

1. The Election Problem
2. The Model
3. Related Works/Our Contribution
4. The MC Algorithms
5. Conclusion and Perspectives
Outline

1. The Election Problem
2. The Model
3. Related Works/Our Contribution
4. The MC Algorithms
5. Conclusion and Perspectives
The Election Problem

Y. Métivier, J.M. Robson, A. Zemmari

MC Election Algorithms
The Election Problem
The Election Problem

Y. Métivier, J.M. Robson, A. Zemmari

MC Election Algorithms
The Election Problem

Y. Métivier, J.M. Robson, A. Zemmari
MC Election Algorithms
The Election Problem
The Election Problem

Definition [LeLann, Tel]

An election algorithm is an algorithm that satisfies the following properties

1. Each process has the same local algorithm.
2. The algorithm is decentralized.
3. The algorithm reaches a terminal configuration in which there is exactly one process in the state leader and all the other processes are in the state lost.
Outline

1. The Election Problem
2. The Model
3. Related Works/Our Contribution
4. The MC Algorithms
5. Conclusion and Perspectives
Anonymous networks.

Point to point communication using messages passing.

No global knowledge (size, diameter, ...)

Nodes can distinguish between neighbors using port numbers.

The system is asynchronous.

⇒ the network is modeled by a graph $G = (V, E)$.

$n$ denotes the size of the graph and $m$ the number of edges.
The Model

Definitions [Tel]:

- An algorithm *process terminates* if in every execution, all processors reach a special *halting state*.

- An algorithm *message terminates* if in every execution, the network reaches a quiescent state where there are no pending messages on the links.

Notation:

- An algorithm is correct with high probability (w.h.p. for short) if it is correct with probability $1 - o(1/n)$.

- An algorithm is correct with very high probability (w.v.h.p. for short) if it is correct with probability $1 - o(1/n^c)$ for any $c \geq 1$. 
Outline

1. The Election Problem
2. The Model
3. Related Works/Our Contribution
4. The MC Algorithms
5. Conclusion and Perspectives
Related Works

Negative results:

- No deterministic algorithm can elect in anonymous networks (even if the size of the network is known) [Angluin].
- With no knowledge on the (anonymous) network, there exists no Las Vegas election algorithm [Itai and Rodeh].
- There exists no process terminating election algorithm that is correct with probability $p > 0$ [Itai and Rodeh].
Some Known Monte Carlo algorithms:

- A spanning tree based algorithm correct with probability $1 - \varepsilon$ where $\varepsilon$ is fixed and known by all the nodes [Schieber and Snir].
  The expected size of messages is $O(\log(n/\varepsilon))$.

- An MC algorithm also correct with probability $1 - \varepsilon$ where $\varepsilon$ is fixed and known by all the nodes [Afek and Matias].
  The expected size of each message is $O(\log \log n + \log \varepsilon^{-1})$. 
Our Contribution

Two Monte Carlo Algorithms:

- The first is correct w.h.p. and the second is correct w.v.h.p.
- Both algorithms terminate in time $O(D)$.
- The first algorithm uses messages of size $O(\log n)$ (w.h.p. and on average).
- The second one uses messages of size $O((\log n)(\log^* n)^2)$ w.v.h.p. their expected size is $O((\log n)(\log^* n))$. 
Algorithm Elect

**Initialization:**

- \( v \) draws u.a.r. a bit \( b(v) \) until \( b(v) = 1 \);
- Let \( t_v \) be the number of draws on the node \( v \);
- \( v \) uses \( t_v \) to draw at random a number \( id_v \);
- \( \text{max}_v = (t_v, id_v) \);
- \( \text{leader} = \text{true} \);
- \( v \) sends \( (t_v, id_v) \) to all neighbors;

**Upon receiving \((t, id)\) at node \(v\) over link \(l\):**

- if \( \text{max}_v \) is not defined then call Initialization;
- if \( (t, id) \leq \text{max}_v \) then ignore
- else
  - \( \text{max}_v = (t, id) \);
  - \( \text{leader} := \text{false} \);
  - send \( (t, id) \) to all neighbors;
The label of each node $v$ is a couple $(t_v, id_v)$ where:
- $t_v$ is the number of draws,
- $id_v$ is a number chosen u.a.r. in the set $\{0, \ldots, 2^{t_v+3 \log_2(t_v)} - 1\}$.

We define the order, denoted $<$, on couples by:

$$(t_v, id_v) < (t'_v, id'_v)$$

if:

- either $t_v < t'_v$
- or $t_v = t'_v$ and $id_v < id'_v$. 
Analysis of the Algorithm

Claim

Let \((t_{\text{max}}, id_{\text{max}})\) be the maximal couple among labels of the vertices. For any run of \(\text{Elect}\), eventually, the network reaches a terminal configuration in which there is no message in transition and no action can happen. The time complexity is \(O(D)\), where \(D\) is the diameter of the graph.

If there is a unique vertex \(v\) such that \((t_v, id_v) = (t_{\text{max}}, id_{\text{max}})\) then \(v\) is the elected vertex.
Analysis of the Algorithm

To analyze the algorithm, we study:

1. the maximal number of bits drawn by each vertex,
2. the size of the messages,
3. the number of vertices that share the maximal lifetime,
4. the number of vertices that share the maximal lifetime and the same maximal number.
The maximal number of bits drawn by each vertex:

**Theorem 1.**

Let $G = (V, E)$ be a graph with $n > 0$ vertices. Let $T$ denote the maximal value of the set $\{t_v | v \in V\}$.

1. $T < 2 \log_2 n + \log^* n$ w.h.p.
2. $T > \log_2 n - \log_2 (2 \log n)$ w.h.p.

**Corollary 2.**

The expected value of $T$ is equal to $\Theta(\log n)$. The messages exchanged are of size at most $2 \log_2 n + O(\log_2 \log_2 n)$ w.h.p. The expected value of the size of the messages is $O(\log n)$. 
The number of vertices that share the maximal lifetime:

**Theorem 3.**

The number of vertices which have the same maximum lifetime is, with high probability, at most $2 \log_2 n$. 
The number of vertices that share the maximal lifetime and the same maximal number:

**Lemma 4.**

With high probability, there exists a unique vertex $v$ with a couple $(t_v, id_v)$ such that for any $w \in V \setminus \{v\}$, $t_v \geq t_w$ and $id_v > id_w$. 
The Second MC Algorithm

In Algorithm Elect

The label of each node $v$ is a couple $(t_v, id_v)$ where:
- $t_v$ is the number of draws,
- $id_v$ is a number chosen u.a.r. in the set $\{0, \ldots, 2^{t_v \log^* t_v} - 1\}$.
Analysis of the Algorithm

**Lemma 5.**
Let $G = (V, E)$ be a graph with $n > 0$ vertices. Let $T$ denote the maximal value of the set $\{t_v \mid v \in V\}$. With very high probability, $T$ satisfies:

$$\frac{1}{2} \log_2 n < T < (\log_2 n) \log^* n.$$ 

**Theorem 6.**
There exists a Monte Carlo election algorithm that succeeds w.v.h.p.

**Proposition 7.**
The size of messages is w.v.h.p. $O \left( (\log n)(\log^* n)^2 \right)$. Its expected value is $O \left( (\log n)(\log^* n) \right)$.
Outline

1. The Election Problem
2. The Model
3. Related Works/Our Contribution
4. The MC Algorithms
5. Conclusion and Perspectives
We obtain time efficient MC algorithms to solve the election problem.

Can we consider other applications of the same technique?

Can a lower/upper bound on the size of the network help?
Thank You
Proof of Theorem 1.

After $t$ rounds the expected number of vertices still alive is $\frac{n}{2^t}$. In particular after $2\log_2 n + \log^* n$ rounds it is $\frac{1}{n^{2\log^* n}}$ so that the probability that any vertex remains is less than $\frac{1}{n^{2\log^* n}}$. This proves the first claim.

On the other hand, for any $t > 0$, we have the following equality:

$$\Pr(T > t) = \Pr(\exists v \in V \text{ s.t. } t_v > t) = 1 - \left(1 - \frac{1}{2^t}\right)^n.$$

We also have the exponential approximation $(1 - a)^n \sim e^{-an}$ as $n \to \infty$, thus: $\Pr(T > t) \sim 1 - e^{-\frac{n}{2^t}}$, as $n \to \infty$.

Taking $t = \log_2 n - \log_2 (2 \log n)$, we obtain: $\Pr(T > t) \geq 1 - o\left(\frac{1}{n}\right)$, this proves the second claim which ends the proof. \qed
Proof of Corollary 2.

At the end of each round, half of the vertices still active become inactive. Thus $\mathbb{E}(T)$, the expected value of $T$, is $O(\log n)$. On the other hand, using the Markov inequality, we have:

$$\mathbb{E}(T) \geq (\log_2 n - \log_2 (2 \log n)) \times \Pr(T > \log_2 n - \log_2 (2 \log n)).$$

(1)

Then, using the second claim of Theorem 22, this proves that $\mathbb{E}(T) = \Omega(\log n)$. Which ends the proof.
Proof of Theorem 4.

We consider the probability $C_t$ that at time $t$, there are still more than $2 \log n$ vertices alive and that they all vanish at the next round. The sum of all $C_t$ is the probability that there are more than $2 \log_2 n$ vertices sharing the same maximum lifetime. We have:

\[
C_t = \Pr[\text{alive vertices at time } t > 2 \log_2 n] \\
\times \Pr[\text{all alive vertices finish } | \text{ there are } > 2 \log_2 n] \\
\leq \Pr[\text{all alive vertices finish } | \text{ there are } > 2 \log_2 n].
\]

Then, using (??), we obtain:

\[
C_t \leq 2^{-2 \log_2 n} = n^{-2}.
\] (2)

On the other hand, we have:

\[
\Pr(X_T \geq 2 \log_2 n) \leq \Pr(T > 2 \log_2 n) + \sum_{t \leq 2 \log_2 n} C_t = o\left(\frac{1}{n}\right). \tag{3}
\]

Which ends the proof.
Proof of Lemma 5.

Let $S = \{v_1, \cdots, v_k\}$ be the set of vertices that share the maximum lifetime. By Claim 2. of Theorem 22, w.h.p., any vertex $v$ in $S$ is such that $t_v > \log_2 n - \log_2(2 \log n)$. Thus, each vertex $v$ in $S$ will choose u.a.r. an identifier $id_v$ from a set which contains w.h.p. the set: $\left\{0, \cdots, \frac{n}{2 \log n} \times \left(\log_2 \left(\frac{n}{2 \log n}\right)\right)^3 - 1\right\}$.
If we denote \( f(n) = \frac{n}{2 \log n} \times \left( \log_2 \left( \frac{n}{2 \log n} \right) \right)^3 \), then, the probability for \( v \) to be the unique vertex with the highest identifier is given by:
\[
\Pr(id_v > id_w, \forall w \in S \setminus \{v\}) \geq \sum_{i=1}^{f(n)-1} \left( \frac{1}{f(n)} \right)^k (i - 1)^{k-1} \sim \frac{1}{k} \left( 1 - \frac{1}{f(n)} \right)^k \text{ as } n \to \infty.
\]
Thus, the probability that a unique vertex in \( S \) has the highest identifier is given by:
\[
\Pr(\exists v \text{ s.t. } id_v > id_w, \forall w \in S \setminus \{v\}) \sim \left( 1 - \frac{1}{f(n)} \right)^k \text{ as } n \to \infty.
\]
Now, by Theorem 23, we have $k < 2 \log_2 n$, w.h.p., thus:

$$\Pr(\exists v \text{ s.t. } id_v > id_w, \forall w \in S \setminus \{v\}) \geq \left(1 - \frac{1}{f(n)}\right)^{2 \log_2 n} \sim e^{-2 \frac{\log_2 n}{f(n)}} = 1 - o\left(\frac{1}{n}\right),$$

which ends the proof.
Proof of Lemma 6.

Taking $t = \frac{1}{2} \log_2 n$ in (??), we obtain:

$$\mathbb{P} \left( T > \frac{1}{2} \log_2 n \right) \sim 1 - e^{-\sqrt{n}}, \text{ as } n \to \infty$$

$$\sim 1 - o \left( \frac{1}{n^c} \right) \text{ for any } c \geq 1.$$

On the other hand, after $(\log^* n) \log_2 n$ rounds, the expected number of vertices still alive is $\frac{n}{n^{\log^* n}}$ so that the probability that any vertex remains is less than $\frac{1}{n^{\log^* n - 1}}$ which is $o \left( \frac{1}{n^c} \right)$ for any $c \geq 1$. 

Y. Métivier, J.M. Robson, A. Zemmari

MC Election Algorithms
Proof of Proposition 8.

The first part is a direct consequence of Lemma 2 and of the choice of $id_v$ in the set:

$$\{0, \ldots, 2^T \log^* T - 1\}.$$ 

For the second part, let $S(id_v)$ denote the size of $id_v$, and let $S_{max} = \max_{v \in V} S(id_v) = T \log^* T$.

Then:

$$\mathbb{E}(S_{max}) = \sum_{x \geq 1} \mathbb{P}(S_{max} \geq x)$$

$$= \frac{(\log n)(\log^* n)}{1} \sum_{x=1} \mathbb{P}(S_{max} \geq x)$$

$$+ \sum_{x > (\log n)(\log^* n)} \mathbb{P}(S_{max} \geq x),$$
Proof of Proposition 8.

yielding:

\[ \mathbb{E}(S_{\text{max}}) \leq (\log n)(\log^* n) \]
\[ + \sum_{x \geq (\log n)(\log^* (\log n))} \mathbb{P}(S_{\text{max}} \geq x). \]  

(4)

Then the second part of the sum in expression (4) satisfies:

\[ \sum_{x \geq (\log n)(\log^* (\log n))} \mathbb{P}(S_{\text{max}} \geq x) \]
\[ = \sum_{t \geq \log n} \sum_{y = t \log^* t} (t+1) \log^* (t+1) - 1 \]
\[ \quad \mathbb{P}(S_{\text{max}} \geq y). \]  

(5)
Proof of Proposition 8.

On the other hand, for any $y$ in the interval $[t \log^* t, (t + 1) \log^* (t + 1) - 1]$:

$$
\Pr(S_{\text{max}} \geq y) = \Pr(S_{\text{max}} \geq t \log^* t) = \Pr(T \geq t) \leq \frac{n}{2^t},
$$

(6)

yielding:

$$
\sum_{x \geq (\log n)(\log^* (\log n))} \Pr(S_{\text{max}} \geq x) \leq \sum_{t \geq \log n} \frac{n((t + 1) \log^* (t + 1) - t \log^* t)}{2^t}.
$$

(7)

Since $\log^* (t + 1) \leq \log^* t + 1$, this gives:

$$
\sum_{x \geq (\log n)(\log^* (\log n))} \Pr(S_{\text{max}} \geq x)
$$
Proof of Proposition 8.

Hence:

\[\mathbb{E}(S_{max}) \leq (\log n)(\log^* n) + O(\log n)\]

\[= O((\log n)(\log^* n)), \quad (9)\]

which ends the proof.