On Randomized Leader Election Algorithms based on Extrema Propagation*

Dominik Bojko[1], Jacek Cichoń[2], and Bogdan Węglorz[3]

1 **Department of Computer Science, Faculty of Fundamental Problems of Technology, Wroclaw University of Science and Technology, Poland**
`dominik.bojko@pwr.edu.pl`

2 **Department of Computer Science, Faculty of Fundamental Problems of Technology, Wroclaw University of Science and Technology, Poland**
`jacek.cichon@pwr.edu.pl`

3 **Faculty of Mathematics and Natural Sciences, School of Exact Sciences, Cardinal Stefan Wyszyński University of Warsaw, Poland**
`b.weglorz@uksw.edu.pl`

Abstract

We consider a general framework for randomized leader election algorithms. The randomization is used only for generation of nodes identifiers and the other part of considered algorithms are deterministic. Its most important part is based on the extreme propagation technique. We consider both multi-hop and single-hop networks model. The correctness of considered algorithm is based on such random generation of nodes identifiers from a linearly (totally) ordered set that there is only one node with the maximal identifiers - the node which generate the maximal identifier will become leader in our algorithms. We will show that this approach cover some previously considered algorithms.

1998 ACM Subject Classification C.2.4 Distributed applications, G.3 Probabilistic algorithms

Keywords and phrases leader election, extrema propagation, randomized algorithm, probability

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2016.23

1 Introduction

Electing a leader is a fundamental problem in distributed systems and it is studied in a variety of contexts and scenario. It is a broadly studied useful building block in distributed systems, whether wired or wireless, especially when failures can occur. For example, if a node failure causes the token to be lost in a mutual exclusion algorithm, then the other nodes can elect a new leader to hold a replacement token.

There exists a huge literature on leader election algorithms. There are many of them which assume that each participating node has a unique identifier. They are used to identify participants during the election process. Node identifiers are used to break ties among nodes which have the same value. For example, H. Garcia-Molina in famous paper [12] introduced Bully Election Algorithm which explicitly assumes that the nodes have unique identifiers. The same assumption may be found in a lot of more recent algorithms (see, for example [19]).

The standard definition of the leader election problem for static networks (see e.g. [2]) is that

* This paper is based on work supported by Polish National Science Center (NCN) grant number 2013/09/B/ST6/02258 and by Wrocław University of Technology grant S50129/K1102

1. eventually there is a leader and
2. there should never be more than one leader.

We shall slightly weaken this requirements into the following ones:

1. eventually there is a leader and
2. with a **very high probability** there should never be more than one leader.

By a very high probability we mean, for example, a probability at least $1 - \frac{1}{10^{20}}$. We claim that for any practical application such probability is sufficient.

We will discuss several variants of leader election algorithms which are based on the extrema propagation techniques for networks, popularized in [3], [4]. One of advantages of this technique is its logical clarity and simplicity for implementation. All algorithms considered in our paper start from random assignment of temporary identifiers to all nodes. The problem of assigning distinct labels to nodes of an unknown anonymous network was considered by other authors. For example, in [11] author show an algorithm for assigning short distinct labels, but their approach cannot be used for leader election, since they assume that there exists one distinguished node in the network. Our approach is different: we are assigning rather long labels, but in a very quick way. We shall examine carefully how long random we should use.

Networks are modeled as simple (unordered) **connected** graphs (V, E), where V represents the set of nodes taking part in the protocol and the set E of edges represent direct links between nodes. In the case of arbitrary connected graph we assume that we know an upper bound on the diameter of the graph. We also assume that we know an upper bound on the number of nodes and that the network is synchronized and the time is divided into discrete rounds.

1.1 Our Contribution

Most of leader election algorithms require existence of unique nodes identifiers. We show that we can use instead random identifiers. We show how to implement this method in multi-hop model and in a single-hop model (the beeping model). We derive a formula connecting the problem of proper generation of identifiers with the classical sum of powers of integers problem (Theorem 4). Finally we show that several known leader election algorithms (among other algorithms from [17], [14]) are covered by the framework discussed in this paper.

1.2 Organization of the Paper

In Section 2 we discuss the general framework of leader election algorithms based on the maximum propagation. In Section 4 we discuss randomized methods for generation of unique identifiers. In Section 5 we compare discussed algorithms with some algorithm from the literature.

2 General framework

The listing Algorithm 1 contains a general framework of analyzed algorithms. They are divided into three parts: choosing identifiers, spreading information about chosen identifiers in the network and informing whole network about final decision.

Part 1 of this framework will be discussed in Section 4. Two scenario for the realization of the part 2 will be discussed in Section 3. The part 3 of this framework will not be discussed in this paper, since it can be realized in a standard way using simple flooding algorithm (as a result we may obtain a spanning tree with root being the leader).

Algorithm 1 General Framework

```
each node generate its temporal identifier
network runs an algorithm which inform the node with the highest identifier that it is
  the leader
(if required) the selected node inform all other nodes
```

3 Selection of leader

In this section we show two different realizations of the framework described in Algorithm 1. The first one is designed for multi-hop environment. The second one is designed for single-hop ones. Both algorithm are designed for each node in the network. They start with a sub-procedure of choosing a random identifiers. We do not specify here how it can be done - this will be explained in Section 4.

3.1 Multi-hop flooding model

Let us consider an arbitrary simple connected graph and let D denotes an upper bound the its diameter. We assume that at the beginning of the round each node can read all messages from all of its neighbors and that at the end of each round each node can send messages to all of its neighbors. We fix a finite linearly ordered set $(\Omega, \preceq)$ and we assume that each node $v \in V$ generates in some way an element $X_v \in \Omega$. We will show that if there exists an unique node u such that $X_u = \preceq - \max\{X_v : v \in V\}$ then the Algorithm 2 correctly selects the leader.

Algorithm 2 Maximum Based Leader Election

```
procedure SELECT(D,Ω)                                        ▷ For each node
    generate randomly an identifier X ∈ Ω
    actMax = X;
    Leader = true
    send actMax to all neighbors
    for I = 1 to D do                       ▷ D is an upper bound on graph diameter
        T = list of all numbers received from all neighbors
        if Max(T) ≻ actMax then
            Leader = false
            actMax = Max(T)
            send actMax to all neighbors
        end if
    end for
end procedure
```

▶ **Theorem 1.** *Is there is a unique node which selects a maximal element from the set $\{X_v : v \in V\}$ then the Algorithm 2 selects this node as the unique leader.*

Proof. Let u be the node with maximal identifier and let $M = X_u$ be the biggest identifier. Let $V_0 = \{u\}$ and let $V_k = \{v \in V : v.actMax = M$ after k-th round$\}$. It is easy to see that $V_k = \{v \in V : d(u,v) \leq k\}$ (where $d(x,y)$ counts the number of hops between nodes $x, y \in V$). Therefore $V_D = V$ (D is un upper bound on the graph diameter). Observe also if $v \in V_K$ and $v \neq u$ then we have $v.Leader =$ false. ◀

Remark 1

Let Δ denotes the graph diameter. Then $\Delta \leq D$. From the above proof we deduce that $V_\Delta = \{u\}$. Moreover, it is easy to see that after Δ-th round no message is send in Algorithm 2. Therefore this algorithm posses self-stabilization property. We do not require to implement any counters controlling the distance traveled by messages.

Remark 2

Algorithm 2 can be optimized in several ways. For example, in line 11 sending a message to a node from which the given node hear about $Max(T)$ value is not necessary. This slight improvement decrease significantly the number of transmissions, specially in linear graphs. Another improvement (or rather extension) is an embedding the process of building the spanning tree rooted at the leader into the main loop.

Remark 3

Algorithms based on extrema propagation technique are quite fault-tolerant both on nodes and links damages. Moreover, the average message complexity in the case of using uniform distribution for generating random identifiers is of order $O(n \log n)$, where $n = |V|$ (see [7]).

3.2 Single-hop "beeping model

We consider now the complete graph topology of the network, i.e. a single-hop situation. We will consider a "beeping - model" (see e.g. [8], [1], [9], [15], [5]). In this model the time is divided into rounds and in each round each station can send a short signal, call a "beep", or may listen. When listening it can recognize a signal send by any other node - the distinction between "SINGLE" and "COLLISSION" (the situation when two or more nodes sends a BEEP) is not necessary. This model, introduced by Cornejo and Kuhn [8] in 2010, is a very weak network communications model. The model is related to the ad-hoc radio network model, and can used, for example, as a surrogate model in results concerning radio networks with collision detection. The beep model is interesting in its own right because of its generality and simplicity.

We assume that the identifiers are from the set $\Omega = \{0, \ldots, L-1\}$. Let $K = \lfloor \log_2(L) \rfloor + 1$. We write each number $X \in \Omega$ in base 2:

$$X = a_0 + a_1 \cdot 2 + a_2 \cdot 2^2 + \ldots + a_K 2^K$$

(where $a_i \in \{0, 1\}$) and we define the string $bin_K(X) = a_K a_{K-1} \cdots a_1 a_0$ (note the reverse order). The sequence $bin(X)$ is used in the Algorithm 3 for selection of the leader.

▶ **Theorem 2.** *Is there is a unique node which selects the maximal number from the set $\{X_v : v \in V\}$ then the Algorithm 3 selects this node as the unique leader.*

Proof. Let us call a node eliminated after the k-th round if its variable `leader` takes value `false` after this round.

We shall proof this by an induction of the number K. Suppose first that $K = 0$. Let u be the node with maximal identifier and let v be an arbitrary other node. Then $s_u = 1$ and $s_v = 0$, so the node v is eliminated after (the only) first round.

Suppose now that $K > 0$ and that the theorem if true for all $K' < K$. Let u be the node with maximal identifier and let $k = \min\{i : s_u(i) = 1\}$. Then for any node v and $i < k$ we have $s_v(i) = 0$. All nodes v such that $s_v(k) = 0$ are eliminated ot the end of k-th round.

Algorithm 3 Leader Election in Beeping Model

```
procedure SELECT(L)                                      ▷ For each node
    generate randomly an identifier X ∈ {0, ..., L − 1}^L
    K = ⌊log_2(L)⌋ + 1
    S = bin_K(X)
    Leader = true
    for I = 1 to K+1 do
        if (S[I] = 1) then
            Send BEEP;
        else
            listen;
            if hear BEEP or COLLISION then
                Leader = false
                exit loop
            end if
        end if
    end for
end procedure
```

Let $V_k = \{v \in V : s_v(k) = 1\}$. This is the set of nodes which are not eliminated after k-th round. Clearly $u \in V_k$. If $V_k = \{u\}$ then the leader is selected. Suppose hence that $V_k \neq \{u\}$ From this moments all nodes from the set V_1 are using sequences $a_{k+1}a_{k+2}\ldots a_1a_0$, which corresponds to sequences $bin(X_v - 2^k)$. Therefore we reduce the problem to sequences of length $K + 1 - k = (K - k) + 1$, hence to $K' = K - k < K$. So, we may use the inductive hypothesis. ◀

4 Generating unique identifiers

In the analyzed in this paper framework of leader election algorithms each node u at the beginning of protocol generate a random number ξ_u using its pseudo-random number generator (PRNG), transform the generated number ξ_u into an element X_u from an linearly ordered set $(\Omega, \preceq)$ and use it as its temporary identifier. If there exists only one node u which selects the element $\preceq -\max\{X_v : v \in V\}$ then the node u with the $\preceq$ - maximal temporary identifier X_u may be selected as the leader. We start with one general result about the probability of this event.

▶ **Theorem 3.** *Let $X_1, \ldots, X_n$ be a sequence of independent random variable with the same distribution and with values in the set $\{1, \ldots, \infty\}$. Let $n > 1$ and let S denotes the event $(\exists i)(\forall j \neq i)(X_j < X_k)$. Then*

$$\Pr[S] = n \sum_{k=2}^{\infty} \Pr[X_1 = k] \Pr[X_1 < k]^{n-1} \tag{1}$$

Proof. Notice that each random variables $X_1, \ldots, X_n$ are equally distributes, so $Pr[(\forall j \neq a)(X_j < X_a)] = Pr[(\forall j \neq b)(X_j < X_b)]$ for all $a, b \in \{1, \ldots, L\}$. Hence $\Pr[S] = n \Pr[(\forall j >$

$1)(X_j < X_1)]$. Therefore

$$\Pr[S] = n \sum_{k=1}^{\infty} \Pr[(\forall j > 1)(X_j < X_1) | X_1 = k] \Pr[X_1 = k] =$$

$$n \sum_{k=1}^{\infty} \Pr[(\forall j > 1)(X_j < k)] \Pr[X_1 = k] = n \sum_{k=2}^{\infty} \Pr[X_1 < k]^{n-1} \Pr[X_1 = k] \ .$$

◀

The general formula (1) from Theorem 3 for some distributions can give specific formulas which can be approximated with required precision. We apply it to two kind of distributions: uniform distribution on finite set $\{1, \ldots, L\}$ and a geometric distributions on the set of positive natural numbers.

4.1 Uniform Distribution

We will apply the Theorem 3 to the uniform distribution on a finite set.

▶ **Theorem 4.** *Let $X_1, \ldots, X_n$ be independent random variable uniformly distributed in the set $\{1, \ldots, L\}$. Let $n > 1$ and let $S_{n,L}$ denotes the event $(\exists k)(\forall j \neq k)(X_j < X_k)$. Then*

$$\Pr[S_{n,L}] = \frac{n}{L^n} \sum_{j=1}^{L-1} j^{n-1} \tag{2}$$

Proof. Let us notice that in the case of uniform distribution we have $\Pr[X_1 = j] = \frac{1}{L}$ and $\Pr[X_1 < j] = \frac{j-1}{L}$. Hence from Theorem 3 we get

$$\Pr[S] = n \sum_{j=2}^{L} \frac{1}{L} \left(\frac{j-1}{L} \right)^{n-1} = \frac{n}{L^n} \sum_{j=1}^{L-1} j^{n-1} \ .$$

◀

▶ **Corollary 5.** *With the same notations and assumptions as in Theorem 4 we have*

$$\Pr[S_{n,L}] = 1 - \frac{n}{2L} + r_n \ ,$$

where $0 \leq r_n \leq \frac{1}{6} \left(\frac{n}{L} \right)^2$

Proof. The sum $\sum_{j=1}^{L-1} j^{n-1}$ can be expressed by the classical Faulhaber's formula. But computationally, Faulhaber's formula become unwieldy as n becomes large. The Bernoulli numbers, which appears in it vanish for odd index ≥ 3, but for even index they increase in magnitude very rapidly and alternate in sign. Instead of this approach we use the Euler summation formula (see e.g. [13]) for the function x^{n-1} and get

$$\sum_{j=0}^{L-1} j^{n-1} = \frac{1}{n} L^n - \frac{1}{2} L^{n-1} + \frac{n-1}{12} L^{n-2} + R_2$$

where $R_2 = -\int_0^L \frac{B_2(\{x\})}{2} (n-1)(n-2) x^{n-3} dx$ and $B_2(t) = t^2 - t + \frac{1}{6}$ is the second Bernoulli polynomial. Since $|B_2(t)| \leq \frac{1}{6}$ for $t \in [0, 1]$ we get $|R_2| \leq \frac{n-1}{12} L^{n-2}$. The final equality follows directly from this upper bound and Theorem 4. ◀

Remark 1

We will use the above result only when $n \ll L$. In this case the approximation $\Pr[S_{n,L}] \approx 1 - \frac{n}{2L}$ is very precise.

Remark 2

Let us compare this result with the probability $b_{n,L}$ that in a random uniform sample of n elements from the universe consisting with L elements (a problem connected with the Birthday Paradox) there are no duplicates. Namely, for $n \ll \sqrt{L}$ we have $b_{n,L} \approx 1 - \frac{n^2}{2L}$ and in our case we have $\Pr[S_{n,L}] \approx 1 - \frac{n}{2L}$.

The next property of the probability of the event $S_{n,L}$ we will use in Section 5.2.

▶ **Theorem 6** (monotonicity)**.** *With the same notations as in Theorem 4 we have* $\Pr[S_{n,L}] > \Pr[S_{n+1,L}]$ *for each* n *and* $L \geq 2$.

Proof. We may assume $n > 1$. Let us observe that $\Pr[S_{n+1,L}] - \Pr[S_{n,L}] = \frac{1}{L^{n+1}}\Delta_{n,L}$, where $\Delta_{n,L} = \sum_{j=1}^{L-1}\left((n+1)j^n - Lnj^{n-1}\right)$. We need to show that $\Delta_{n,L} < 0$. Observe that $\Delta_{n,2} = 1 - n < 0$. Let us fix the number n and let $H_{n,L} = \Delta_{n,L+1} - \Delta_{n,L}$. We shall show that for each $L \geq 2$ we have $H_{n,L} < 0$, which will show that $\Delta_{n,L+1} < \Delta_{n,L}$ for each $L \geq 2$, so the theorem will be proved. It is easy to check that $H_{n,L} = L^n - n\sum_{j=1}^{L} j^{n-1}$. But $\sum_{j=1}^{L} j^{n-1} > \int_0^L x^{n-1}dx = \frac{1}{n}L^n$, hence

$$H_{n,L} < L^n - n\frac{1}{n}L^n = L^n - L^n = 0 \ ,$$

hence the theorem is proved. ◀

4.2 Geometric distribution

Let us fix a number $p \in (0,1)$. Let us recall that a random variable with values in the set of positive natural numbers has a geometric distribution with parameter p ($x \sim Geo(p)$) if $\Pr[X_u = k] = (1-p)^{k-1}p$ for $k \in \{1,2,3,\ldots\}$. Let $X_1, \ldots, X_n$ be independent copy of random variables with $Geo(p)$ distribution and let $M_{n,p} = \max\{X_i : 1 \leq i \leq n\}$ and $W_{n,p} = |\{k : X_k = M_{n,p}\}|$. Then we have

▶ **Lemma 7** ([6])**.** *If* $C > 1$ *then* $\Pr[M_{n,p} > C\frac{\ln n}{\ln \frac{1}{1-p}}] \leq \frac{1}{n^{C-1}}$.

Using an approach similar as in the proof of Theorem 3 one can derive the formula $\Pr[W_{n,p} = a] = \binom{n}{a}p^a \sum_{b=0}^{n-a}\binom{n-a}{b}\frac{(-1)^b}{1-q^{a+b}}$, where $q = 1-p$, and then using the Mellin transform or the Rice method one can derive the following approximation

▶ **Theorem 8** ([16], [6])**.** $\Pr[W_{n,p} = a] \approx \frac{1}{\ln\frac{1}{1-p}}\frac{p^a}{a}$

Lemma 7 may be used for finding such a number L that $\Pr[M_{n,p} > L]$ is very small and Theorem 8 we will used in Section for finding probability of success in leader election algorithms based on geometric distributions.

Remark

It worth to notice that right side of formula from Theorem 8 does not depend on n. In fact the number $\Pr[W_{n,p} = a]$ depends on n, but the influence of n on this probability is very small for all $p < \frac{1}{2}$ (see [6]). Moreover from Theorem 8 we easily deduce that $\Pr[W_{n,p} = 1] = 1 - \frac{p}{2} + O(p^2)$ (as p tends to 0).

5 Applications

We discuss in this section several possible realizations of algorithm described in Section 3.

5.1 Uniform distribution

Let us consider usage of pseudo-random number generator (PRNG's) for selection nodes identifiers. Hence we assume that each node is equipped with a high-quality pseudo-random number generator. Moreover, we assume that this PRNG's are initialized using independent seeds. Let the PRNG's produces sequences of bits of length L in an uniform way, i.e. that each sequence $s \in \{0,1\}^L$ is generated with the same probability. We may transform this sequence of bits into a number from the $\{0,\ldots 2^L - 1\}$. Let $n = |V|$ denotes the number of nodes in the network. Let S denotes the event $|\{u \in V : X_u = \max\{X_v : v \in V\}\}| = 1$. From Corollary 5 we deduce that if $n \ll 2^L$ then

$$\Pr[S] \approx 1 - \frac{n}{2^{L+1}} \ .$$

Notice that

$$\left(1 - \frac{n}{2^{L+1}} > 1 - \frac{1}{10^a}\right) \equiv (L > a \log_2 10 + \log_2 n - 1) \ .$$

Observe that $\log_2 10 = 3.32193\ldots$. Therefore if we wold like to guarantee a probability of success of the order $1 - \frac{1}{10^{20}}$ then we need $66 + \log_2 n$ bits. Surely 64 bits do not suffice. But if we use PRNG's to generate random sequences of length 128 then the event S holds with probability at least $1 - \frac{1}{10^{20}}$ and any $n < 2^{62} \approx 4.6 \cdot 10^{19}$. Hence we derive the following result:

▶ **Corollary 9.** *If we use the uniform distribution on the set $\{0,\ldots,2^{128}-1\}$ then both Algorithms 2 and 3 select an unique leader with probability at least $1 - \frac{1}{10^{20}}$ for any number of nodes $n \le 10^{19}$.*

How to select a loser

The precise probability of a success in the case of uniform distribution on the set $\{0,\ldots,2^L-1\}$, according to Theorem 4, is given by formula $\Pr[S] = \frac{n}{2^L}\sum_{j=1}^{2^L-1} j^{n-1}$. This formula may be found in [10] in a slightly different context. In fact, this paper is devoted to a detailed analysis of Leader Election Algorithm published in [17] (*How to select a loser*, Discrete Mathematics, 1993). Prodinger's algorithm from [17] may be directly transformed into Algorithm 3, and as a by-product of our considerations may be also easily implemented in the multi-hop environment.

5.2 Best-node-based leader election algorithms

Best node based leader election algorithms try to select as a leader the node which is the best one according to some nodes capabilities. An overview of such algorithms may be found in [18]. Most of such algorithms require unique nodes identifiers to resolve collisions. We can combined random node identifiers developed in this paper with this kind of algorithms. Namely, suppose that nodes capabilities are from a finite linearly ordered set $(C, \preceq_C)$. Let $(\Omega, \le)$ be also a finite linearly ordered set. We consider the lexical ordering on the set $C \times \Omega$ defined by

$$(x_1,y_1) \prec (x_2,y_2) \equiv (x_1 \prec_C x_2) \vee ((x_1 = x_2) \wedge (y_1 < y_2) \ .$$

Since $\prec$ is an linear order, we may use it in both algorithm discussed in this paper. At the beginning of this algorithms each node calculates its capability $C_v \in C$, select a random identifier X_v and uses the element (C_v, X_v) in further computations. It is clear the the randomly generated parts of identifiers are used to resolve collisions between nodes from the set $\{u \in V : C_u = \max\{C_v : v \in V\}\}$, hence the results from the previews subsection are applicable in this setting (see Theorem 6).

5.3 Leader Green Election

Leader Green Election (LGE) Algorithm was introduced by P. Jacquet at al. in [14]. This is the main idea of this algorithm: we fix a small number $p \in (0, 1)$ (say p = 10^{-2}); each node u generates a random number X_u from the geometric distribution with parameter p (i.e. $\Pr[X_u = k] = (1 - p)^k p$ for natural numbers k). A winner (leader) is the node with maximal identifier X_u. In original paper LGE was based on beeping model.

At first glimpse this solution are not in the framework discussed in this paper - the values of X_u are not bounded. But from Lemma 7 we may conclude that $\Pr[Y > (\ln 10^{20} + \ln n)/\ln \frac{1}{1-p}] < 10^{-20}$, and hence from a practical point of view it is negligible. Putting into this formula $p = 0.01$ and $n = 10^{20}$ we get $\Pr[Y > 10^4] < 10^{-20}$. Notice also that $\lfloor \log_2 10^4 \rfloor + 1 = 14$, hence only 14 bits are required to run LGE algorithm in the framework of Algorithm 3 (the "beep-model"). And we also see that we can easily implement LGE algorithm in multi-hop environment.

From Theorem 8 we get $\Pr[W_{n,p} = 1] \approx 1 - \frac{p}{2}$, hence is quite small. Authors of [14] propose using this method several times, i.e. they propose to use it k times, where k is such that $(\frac{p}{2})^k$ is sufficiently small. However, there is a better solution. Namely, from Theorem 8 we may deduce (see [6] for details) that

$$\Pr[W_{n,\frac{1}{100}} > 10] < \frac{1}{10^{21}}$$

for any $n > 1$. So we may treat LGE as a method for quick reduction of an arbitrary collection of nodes to small subgroup - using parameter $p = \frac{1}{100}$ with probability at least $1 - \frac{1}{10^{21}}$ this subgroup has cardinality at most 10. From discussion in Section 5.1 we deduce that we need 73 bits to select from any group of size at most 10 nodes a leader using uniform distribution. Therefore we need 87 bits (14+73=87) for a successful leader election from any group of nodes of cardinality less that 10^{20} with probability at least $1 - \frac{1}{10^{20}}$ using a mix of two methods: first we use LGE algorithm and next we use uniform distribution. The mixture can be realized, for example, using the lexicographical product of ordering, which was discussed in previous section.

6 Conclusions

The initial assumption of many leader election algorithms based on assumption that nodes in a network have distinct identifiers is not necessary: nodes temporary identifiers may be generated randomly and we may control the account of necessary randomness by a careful choice of distribution. We claim that such solution improve flexibility of many leader election algorithms.

References

1 Yehuda Afek, Noga Alon, Ziv Bar-Joseph, Alejandro Cornejo, Bernhard Haeupler, and Fabian Kuhn. Beeping a maximal independent set. In *Proceedings of the 25th International*

Conference on Distributed Computing, DISC'11, pages 32–50, Berlin, Heidelberg, 2011. Springer-Verlag.

2 Hagit Attiya and Jennifer Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics.* John Wiley & Sons, 2004.

3 Carlos Baquero, Paulo Sérgio Almeida, and Raquel Menezes. Fast estimation of aggregates in unstructured networks. In *ICAS*, pages 88–93, 2009.

4 Carlos Baquero, Paulo Sérgio Almeida, Raquel Menezes, and Paulo Jesus. Extrema propagation: Fast distributed estimation of sums and network sizes. *IEEE Trans. Parallel Distrib. Syst.*, 23:668–675, 2012.

5 Philipp Brandes, Marcin Kardas, Marek Klonowski, Dominik Pająk, and Roger Wattenhofer. Approximating the size of a radio network in beeping model. In *23rd International Colloquium on Structural Information and Communication Complexity*, Helsinki, Finland, Jul 2016.

6 Jacek Cichoń, Rafal Kapelko, and Dominik Markiewicz. On leader green election. In *Proceedings of the 27th International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms, AofA'16*, Kraków, Poland, July 2016. URL: http://arxiv.org/abs/1605.00137.

7 Jacek Cichon, Jakub Lemiesz, and Marcin Zawada. On message complexity of extrema propagation techniques. In Xiang-Yang Li, Symeon Papavassiliou, and Stefan Rührup, editors, *Ad-hoc, Mobile, and Wireless Networks - 11th International Conference, ADHOC-NOW 2012, Belgrade, Serbia, July 9-11, 2012. Proceedings*, volume 7363 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2012.

8 Alejandro Cornejo and Fabian Kuhn. Deploying wireless networks with beeps. In *Proceedings of the 24th International Conference on Distributed Computing*, DISC'10, pages 148–162, Berlin, Heidelberg, 2010. Springer-Verlag.

9 Samir Elouasbi and Andrzej Pelc. Deterministic rendezvous with detection using beeps. In *Revised Selected Papers of the 11th International Symposium on Algorithms for Sensor Systems - Volume 9536*, ALGOSENSORS 2015, pages 85–97, New York, NY, USA, 2015. Springer-Verlag New York, Inc.

10 James Allen Fill, Hosam M. Mahmoud, and Wojciech Szpankowski. On the distribution for the duration of a randomized leader election algorithm. *Ann. Appl. Probab*, 6:1260–1283, 1996.

11 Pierre Fraigniaud, Andrzej Pelc, David Peleg, and Stéphane Pérennes. Assigning labels in unknown anonymous networks (extended abstract). In *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '00, pages 101–111, New York, NY, USA, 2000. ACM. URL: http://doi.acm.org/10.1145/343477.343527, doi:10.1145/343477.343527.

12 Hector Garcia-Molina. Elections in a distributed computing system. *IEEE Trans. Computers*, 31(1):48–59, 1982. URL: http://dx.doi.org/10.1109/TC.1982.1675885, doi:10.1109/TC.1982.1675885.

13 Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete mathematics - a foundation for computer science (2. ed.).* Addison-Wesley, 1994.

14 Philippe Jacquet, Dimitris Milioris, and Paul Mühlethaler. A novel energy efficient broadcast leader election. In *MASCOTS*, pages 495–504. IEEE, 2013.

15 Marcin Kardas, Mirosław Kutyłowski, and Jakub Lemiesz. On distributed cardinality estimation: Random arcs recycled. In *Proceedings of the Meeting on Analytic Algorithmics and Combinatorics*, pages 129–137, Philadelphia, PA, USA, 2015. Society for Industrial and Applied Mathematics.

16 Peter Kirschenhofer and Helmut Prodinger. The number of winners in a discrete geometrically distributed sample. *The Annals of Applied Probability*, 6(2):687–694, 05 1996. URL: `http://dx.doi.org/10.1214/aoap/1034968150`.

17 Helmut Prodinger. How to select a loser. *Discrete Mathematics*, 120(1-3):149–159, 1993. URL: `http://dx.doi.org/10.1016/0012-365X(93)90572-B`, `doi:10.1016/0012-365X(93)90572-B`.

18 Shantanu Sharma and Awadhesh Kumar Singh. An election algorithm to ensure the high availability of leader in large mobile ad hoc networks. *International Journal of Parallel, Emergent and Distributed Systems*, 0(0):1–25, 0. URL: `http://dx.doi.org/10.1080/17445760.2016.1191077`, `arXiv:http://dx.doi.org/10.1080/17445760.2016.1191077`, `doi:10.1080/17445760.2016.1191077`.

19 Sudarshan Vasudevan, Jim Kurose, and Don Towsley. Design and analysis of a leader election algorithm for mobile ad hoc networks. In *In IEEE International Conference on Network Protocols*, pages 350–360. IEEE Computer Society, 2004.