



HAL
open science

Average number of messages for distributed leader-finding in rings of processors

Christian Lavault

► **To cite this version:**

Christian Lavault. Average number of messages for distributed leader-finding in rings of processors. Information Processing Letters, 1989, 30 (4), pp.167-176. hal-00464399

HAL Id: hal-00464399

<https://hal.science/hal-00464399>

Submitted on 5 Apr 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AVERAGE NUMBER OF MESSAGES FOR DISTRIBUTED LEADER-FINDING IN RINGS OF PROCESSORS

Christian LAVAUT

INRIA, Domaine de Voluceau, Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France

Communicated by L. Kott

Received 1 June 1988

Revised 10 September and 30 September 1988

Keywords: Distributed algorithms, average message complexity, election, uni- and bidirectional rings

1. Introduction

In the sequel, we consider the problem of finding a leader in an asynchronous bidirectional ring of processors. Each processor is distinguished by a unique identification number (its “identity”). There is no central controller and every processor has only local information about the network, namely it only knows its direct neighbours in the distributed system. The problem is to design a distributed algorithm that elects a unique processor as the leader (e.g., the largest numbered one) in using a minimum number of messages.

We assume that the processors work fully asynchronously and cannot use clocks or time-outs. Hence, we can assume that the algorithms are message-driven: except for the initialization phase of an election, any processor can only perform actions upon receipt of a message. We also assume the processors and the communications system to be error-free.

As to the terminology, a *message* is any information device which travels around the ring, from one processor to one another, whether they are neighbours or not. An *elementary message*, which is a message between two neighbour sites in the ring will also be called a message. Finally, one *pip* will denote the traversal delay of one elementary message.

Much work has already been completed to obtain good upper, lower and average bounds for different variants of the problem. In [1], tight upper and lower bounds for distributed leader-finding in bidirectional rings are presented. These bounds are exhibited for the probabilistic algorithm given in [8,12] (Algorithm P) and for a derived deterministic version of the latter (Algorithm D). Up to now, this result was the best approximation of the average number of messages required in Algorithm P (an upperbound of $0.7075 \dots nH_n + O(n)$ and a lower bound of $0.7033 \dots nH_n + O(n)$) and Algorithm D (previous upper bound).

In this paper, we derive the exact asymptotic average value of the number of messages required in Algorithm P and Algorithm D. This value is obtained by use of considerations from the theory of permutations and combinatorial enumeration arguments. The main result of this paper asserts that both Algorithm P and Algorithm D have the same asymptotic average message complexity of $2^{-1/2}nH_n + O(n)$. The result leads to a positive answer to the question (first posed by Pachl, Korach and Rotem) whether distributed leader-finding can be solved more efficiently in bidirectional rings than in unidirectional rings by a deterministic algorithm (this actually follows from the results in [1]).

2. Inversion tables (cf. Vuillemin [13])

Let $\pi = (\sigma_1\sigma_2 \cdots \sigma_n) \in \mathfrak{S}_n$ be a permutation of size n . Associated with π , define its *inversion table* $t = t_1t_2 \cdots t_n$ such that t_i is the number of elements (in π) to the left of σ_i that are larger than σ_i for all i ($1 \leq i \leq n$). Hence, $0 \leq t_i \leq i$ for all i ($1 \leq i \leq n$), and the correspondence between a permutation and its inversion table is *one-to-one*.

2.1. Lemma (Vuillemin [13]). *The left-to-right maxima (or upper records) of a permutation $\pi \in \mathfrak{S}_n$ correspond to the occurrences of the value zero in the inversion table of π .*

If one uses variables to denote the values $0, 1, \dots, n - 1$ in the inversion table t of a permutation π (e.g., x_1, x_2, \dots, x_{n-1}), then the set of all inversion tables corresponding to all permutations of size n is fully described by the polynomial

$$\begin{array}{ccccccc}
 x_0 & (x_0 + x_1) & (x_0 + x_1 + x_2) & \cdots & (x_0 + x_1 + \cdots + x_{n-1}) & & \\
 \uparrow & \uparrow & \uparrow & & \uparrow & & \\
 \text{1st column} & \text{2nd column} & \text{3rd column} & & \text{(n-1)st column} & &
 \end{array} \tag{2.1}$$

Notation. (i) The n th harmonic number is given by $H_n = \sum_{i=1}^n 1/i$, with asymptotic expansion $H_n = \ln n + \gamma + \frac{1}{2}n^{-1} + O(n^{-2}) \sim 0.693 \dots \lg n$ (where $\gamma = 0.577 \dots$ is Euler's constant); $\ln n$ is the Neperian logarithm of n and $\lg n$ denotes the logarithm taken to the base 2.

(ii) The notation $[x^k]f(x)$, with $f(x) = \sum_k f_k x^k$, reads as "the coefficient of x^k in $f(x)$ ".

2.2. Definition ([2, p. 48]). The unsigned Stirling numbers of the first kind, $s_{n,k}$, are such that

$$s_{n,k} = [x^k]x(x+1)(x+2) \cdots (x+n-1);$$

thus the enumerating polynomial of $s_{n,k}$ is $\sum_k s_{n,k} x^k = x(x+1)(x+2) \cdots (x+n-1)$.

2.3. Lemma. *The average number of zero elements in an inversion table of size n is H_n . The number of inversion tables of size n having k zeros is the unsigned Stirling number of the first kind $s_{n,k}$.*

Proof. The unsigned Stirling number of the first kind, $s_{n,k}$, is proved in [13] to count both the permutations $\pi \in \mathfrak{S}_n$ with k upper records and the inversion tables of size n such that $|\{i | 1 \leq i \leq n, t_i = 0\}| = k$. \square

Therefore, an immediate consequence of Lemmata 2.1 and 2.3 is the following corollary.

2.4. Corollary. *The average number of left-to-right maxima (upper records) of a permutation $\pi \in \mathfrak{S}_n$ is H_n .*

2.5. Lemma. *Let $\pi = (\sigma_1\sigma_2 \cdots \sigma_n)$. The average distance to the first left-to-right maximum (upper record) of π is $H_n - 1$.*

Proof. This well-known result repeatedly occurs in the theory of permutations; see [13,1] for example; and [9,10] for a more direct proof of this result. \square

2.6. Proposition. *The probability $\Pi_n(j)$ that a permutation $\pi \in \mathfrak{S}_n$ has exactly j upper records ($j \geq 2$), with the leftmost one in position α ($\alpha > 1$) and the rightmost in position β is*

$$\Pi_n(j) = [x^{j-2}] \frac{1}{\beta(\alpha-1)(\beta-1)} \left(1 + \frac{x}{\alpha}\right) \left(1 + \frac{x}{\alpha+1}\right) \cdots \left(1 + \frac{x}{\beta-2}\right). \tag{2.2}$$

Proof. Let $G_n(x)$ denote the generating polynomial of permutations $\pi \in \mathfrak{S}_n$ with exactly j upper records ($j \geq 2$), the leftmost one being in position α and the rightmost one in position β : $G_n(x)$ depends on the values α and β .

In the first place, let consider all the *positions* of the j left-to-right maxima of π ($j \geq 2$) in an inversion table of size n (the leftmost in α , the rightmost in β). Let us then write down all the *monomials*, each corresponding to the possible upper records of π :

Positions:

$$1 \ 2 \ 3 \ \cdots \ (\alpha - 1)\alpha(\alpha + 1) \ \cdots \ (\beta - 1)\beta(\beta + 1) \ \cdots \ n,$$

Monomials:

$$1 \ 1 \ 2 \ \cdots \ (\alpha - 2)x(x + \alpha)(x + \alpha + 1) \ \cdots \ (x + \beta - 2)x(\beta + 1) \ \cdots \ n.$$

The corresponding generating polynomial $G_n(x)$ is derived from the above terms as the product of all the stated *monomials*, divided by the product of all the *positions* in an inversion table of size n . Namely,

$$G_n(x) = \frac{1 \cdots 1 \cdots 2 \cdots (\alpha - 2) \cdot x \cdot (x + \alpha) \cdot (x + \alpha + 1) \cdots (x + \beta - 2) \cdot x \cdot (\beta + 1) \cdots n}{1 \cdot 2 \cdot 3 \cdots (\alpha - 1) \cdot \alpha \cdot (\alpha + 1) \cdots (\beta - 2) \cdot (\beta - 1) \cdot \beta \cdot (\beta + 1) \cdots n},$$

and

$$G_n(x) = \frac{x^2}{\beta(\alpha - 1)(\beta - 1)} \left(\frac{x + \alpha}{\alpha}\right) \left(\frac{x + \alpha + 1}{\alpha + 1}\right) \cdots \left(\frac{x + \beta - 2}{\beta - 2}\right).$$

Secondly, expressing the generating polynomial $G_n(x)$ as the generating function of the probability $\Pi_n(j)$, we get

$$G_n(x) = \sum_{j \geq 2} \Pi_n(j) x^j = x^2 \sum_{j \geq 2} \Pi_n(j) x^{j-2}. \tag{2.3}$$

Thus, $\Pi_n(j)$ is the coefficient of x^{j-2} in $G_n(x)$ for $\frac{1}{2}\beta < \alpha < 2\alpha$ and the value of $\Pi_n(j)$ follows. \square

3. Analysis of Algorithm P

We know from [8,1,12] that Algorithm P requires an expected number of messages of at most $\frac{3}{4}nH_n + O(n)$. This value is only an upper bound, because the possible effect of higher-order upper records was ignored in this evaluation.

Algorithm P (Bodlaender and Van Leeuwen [1])

Each processor P_i keeps the largest identity (identification number) it has seen in a local variable MAX_i ($1 \leq i \leq n$).

Each processor P_i goes through the following stages.

Stage 1 (initialization): $MAX_i := \sigma_i$; choose a direction $d \in \{\text{left, right}\}$ with probability $\frac{1}{2}$; send message $\langle \sigma_i \rangle$ in direction d on the ring;

Stage 2 (election): repeat the following steps, until the end of the election is signaled by receipt of a $\langle ! \rangle$ message: if two messages are received from the left and the right simultaneously, then ignore the smaller message and proceed as if only the larger message is received; if message $\langle \sigma_j \rangle$ is received from a neighbour, then

if $\sigma_j > MAX_i$, then $MAX_i := \sigma_j$; pass message $\langle \sigma_j \rangle$ on

else if $\sigma_j = MAX_i$, then send message $\langle ! \rangle$ on the ring / * P_i has won the election * /

fi;

Stage 3 (inauguration): if a message $\langle ! \rangle$ is received, the election is over and MAX_i holds the identity of the leader; if this processor was elected in Stage 2, then the inauguration is over, otherwise pass message $\langle ! \rangle$ on and stop.

One easily verifies that a processor P_i wins the election if and only if its identity succeeds in making a full round along the ring in a direction chosen in Stage 1. The correctness of the algorithm is proved in [1,11]. In the following analyses, we assume that all processors start the election simultaneously and that the processors work synchronously.

3.1. *Exact evaluation of the expected number of messages*

Consider a ring of n processors P_1, \dots, P_n with identities σ_1 through σ_n . Without loss of generality, we may assume each σ_i to be an integer between 1 and n . Thus, $\pi = (\sigma_1 \sigma_2 \dots \sigma_n)$ is a permutation of \mathfrak{S}_n . Assuming also that the permutations of \mathfrak{S}_n are equally likely, we can make use of the preceding results to analyse Algorithm P.

First, set $i = 1$; the message $\langle \sigma_1 \rangle$ is sent to the right or to the left with probability $\frac{1}{2}$. Hence, the expected number of elements in π visited $\langle \sigma_1 \rangle$ is $\frac{1}{2}(H_n - 1)$ whenever P_1 sends its message $\langle \sigma_1 \rangle$ to the right, and $\frac{1}{2} \lfloor \frac{1}{2} H_n \rfloor$ whenever P_1 sends its message $\langle \sigma_1 \rangle$ to the left since, from Lemma 2.5, $H_n - 1$ is the average distance to the first left-to-right maximum in π . Accumulating the sum of these two quantities for all $\langle \sigma_i \rangle$ -messages ($1 \leq i \leq n$), which are independent random variables, yields the known upper bound of $\frac{3}{4}nH_n + O(n)$ for the average number of messages required by Algorithm P. Now taking also into account the effect of higher-order upper records leads to the exact average value.

3.1. Proposition. *The average number of $\langle \sigma_1 \rangle$ -messages propagated by Algorithm P is exactly*

$$\frac{3}{4}H_n - \sum_{\alpha, \beta} (\alpha - \frac{1}{2}\beta) G_n(\frac{1}{2}) + O(1), \quad \text{for } \frac{1}{2}\beta < \alpha < \beta < 2\alpha,$$

where $G_n(x)$ is the generating polynomial defined in (2.2).

Proof. Let ν_1, ν_2, \dots be random variables denoting the position of the first-, second-, and higher-order upper records. We may consider the ν_i as independent random variables conditioned over the values α and β , and suppose $\langle \sigma_1 \rangle$ is sent to the right without loss of generality. If processors P_α to $P_{\beta-1}$ randomly choose to send their message to the *right* as well, but P_β sends its message to the *left*, then the $\langle \sigma_1 \rangle$ -message is annihilated by the $\langle \sigma_\beta \rangle$ -message if the messages meet before P_α is reached; i.e., at position $1 + \lfloor \frac{1}{2}\beta \rfloor$, provided $\beta < 2\alpha$. Otherwise, the $\langle \sigma_1 \rangle$ -message is simply annihilated at p_α . Hence, let π denote the permutation $(\sigma_1 \dots \sigma_n) \in \mathfrak{S}_n$; the number of positions in π visited by message $\langle \sigma_1 \rangle$ is exactly $\alpha - (1 + \lfloor \frac{1}{2}\beta \rfloor)$, with $\frac{1}{2}\beta < \alpha < \beta < 2\alpha$. Recall now formulae (2.2) and (2.3), and the definition of probability $\Pi_n(j)$ and let us examine the effect of *all* upper records of π . The average number of $\langle \sigma_1 \rangle$ -messages propagated by Algorithm P is

$$\bar{N}_1 = \frac{1}{2}(H_n - 1) + \frac{1}{2} \lfloor \frac{1}{2} H_n \rfloor - \sum_{j \geq 2} 2^{-j} \sum_{\alpha, \beta} (\alpha - \lfloor \frac{1}{2}\beta \rfloor - 1) \Pi_n(j) \tag{3.1}$$

and, since

$$\begin{aligned} \sum_{j \geq 2} 2^{-j} \Pi_n(j) &= \sum_{j \geq 2} 2^{-j} \{ [x^{j-2}] G_n(x) \} = \frac{1}{4} \sum_{j \geq 2} 2^{-j} \{ [x^j] G_n(x) \} \\ &= \frac{1}{4} G_n(\frac{1}{2}), \end{aligned}$$

we have

$$\bar{N}_1 = \frac{3}{4}H_n - \frac{1}{4} \sum_{\alpha, \beta} (\alpha - \frac{1}{2}\beta) G_n(\frac{1}{2}) + O(1) \quad (\frac{1}{2}\beta < \alpha < 2\alpha). \tag{3.2}$$

Note that the identity $\sum_{j \geq 0} 2^{-j} \{[x^j]f(x)\} = f(\frac{1}{2})$ is obvious, since $f(\frac{1}{2}) = \sum_{j \geq 0} f_j 2^{-j}$, whenever $f(x) = \sum_{j \geq 0} f_j x^j$. The coefficient $\frac{1}{4}$ in (3.2) comes precisely from the fact that we have here $[x^{j-2}]G_n(x)$ instead of $[x^j]G_n(x)$. \square

3.2. Corollary *The average number of messages required by Algorithm P is exactly $\frac{3}{4}nH_n - nS_n + O(n)$, where*

$$S_n = \sum_{2 \leq \beta \leq n} \sum_{\beta/2 < \alpha < \beta} \frac{1}{4} \frac{1}{\beta(\alpha-1)(\beta-1)} \left(1 + \frac{1}{2\alpha}\right) \left(1 + \frac{1}{2\alpha+2}\right) \cdots \left(1 + \frac{1}{2\beta-4}\right).$$

Proof. Accumulating in (3.2) the quantity \bar{N}_1 for all the n $\langle \sigma_i \rangle$ -messages ($1 \leq i \leq n$), which are independent random variables, yields the exact average number of messages, namely

$$\bar{N} = \frac{3}{4}nH_n - n \sum_{2 \leq \beta \leq n} \sum_{\beta/2 < \alpha < \beta} \frac{1}{4} (\alpha - \frac{1}{2}\beta) G_n(\frac{1}{2}) + O(n).$$

Now,

$$G_n(\frac{1}{2}) = \frac{1}{\beta(\alpha-1)(\beta-1)} \left(1 + \frac{1}{2\alpha}\right) \left(1 + \frac{1}{2\alpha+2}\right) \cdots \left(1 + \frac{1}{2\beta-4}\right),$$

and summing respectively over β ($2 \leq \beta \leq n$) and α ($\frac{1}{2}\beta < \alpha < \beta$),

$$\bar{N} = \frac{3}{4}nH_n - n \sum_{\beta} \sum_{\alpha} \frac{1}{4} \frac{(\alpha - \beta/2)}{\beta(\alpha-1)(\beta-1)} \left(1 + \frac{1}{2\alpha}\right) \left(1 + \frac{1}{2\alpha+2}\right) \cdots \left(1 + \frac{1}{2\beta-4}\right) + O(n). \tag{3.3}$$

Whence we obtain (3.3) the exact average number of messages: $\frac{3}{4}nH_n - nS_n + O(n)$. \square

The remaining calculations concern the asymptotic behaviour of S_n .

3.2. Asymptotic analysis of S_n (cf. Flajolet [4])

The following asymptotic analysis of S_n makes use of Stirling's formula and the Euler-Maclaurin (one-dimensional) summation formula. For $p \in \mathbb{N}$, define

$$Q_p = \left(1 + \frac{1}{2}\right) \left(1 + \frac{1}{4}\right) \cdots \left(1 + \frac{1}{2p}\right) = \frac{3 \cdot 5 \cdot 7 \cdots (2p+1)}{2 \cdot 4 \cdot 6 \cdots (2p)} = \frac{(2p+1)}{2^{2p}(p!)};$$

and, by means of Stirling's formula for large p , we have $Q_p \sim 2\sqrt{p}/\sqrt{\pi}$ with an error term of $O(1)$. The sum S_n may then be written with the Q_p 's as follows:

$$S_n = \frac{1}{4} \sum_{\alpha, \beta} \frac{Q_{\beta-1}}{Q_{\alpha}} \frac{(\alpha - \frac{1}{2}\beta)}{\beta(\alpha-1)(\beta-1)} = \frac{1}{4} \sum_{2 \leq \beta \leq n} \left\{ \sum_{\beta/2 < \alpha < \beta} \frac{Q_{\beta-1}}{Q_{\alpha}} \frac{(\alpha - \frac{1}{2}\beta)}{\beta(\alpha-1)(\beta-1)} \right\}. \tag{3.4}$$

Denote by $u(\beta)$ the inner sum (with index α). For β large enough, by Stirling's formula,

$$u(\beta) \sim \sum_{\alpha, \beta} \frac{(\alpha - \frac{1}{2}\beta)}{\alpha\beta^2} \sqrt{\frac{\beta}{\alpha}} \quad (\frac{1}{2}\beta < \alpha < \beta).$$

If we now set $\alpha = t\beta$, where t ranges between $\frac{1}{2}$ and 1 by steps of $1/\beta$, then

$$u(\beta) \sim \frac{1}{\beta} \left\{ \sum_t \frac{(t - \frac{1}{2})}{t\sqrt{t}} \cdot \frac{1}{\beta} \right\},$$

and by approximation of the discrete sum by an integral (Euler-Maclaurin summation formula), one gets the asymptotic expression

$$u(\beta) = \frac{1}{\beta} \left\{ \int_{1/2}^1 \frac{(t - \frac{1}{2})}{t^{3/2}} dt + O(1/\beta) \right\}$$

which is uniform in β . And at last,

$$u(\beta) = \frac{1}{\beta} (3 - 2\sqrt{2} + O(1/\beta)) \quad (3.5)$$

which yields the following theorem.

3.3. Theorem. *The asymptotic value of the expected number of messages used in Algorithm P is $nH_n/\sqrt{2} + O(n) = 0.707106\dots n \ln n + O(n)$.*

Proof. From (3.3), $\bar{N} = \frac{3}{4}nH_n - nS_n + O(n)$. The exact computation of S_n from (3.4) and (3.5) gives rise to the expression

$$S_n = \frac{1}{4} \sum_{\beta} \frac{1}{\beta} (3 - 2\sqrt{2}) + O(1);$$

and since $\sum O(1/\beta) = O(1)$ (S_n being uniform in β), we have

$$\bar{N} = \frac{3}{4}nH_n - n \frac{(3 - 2\sqrt{2})}{4} \sum_{\beta} \frac{1}{\beta} + O(n) = \frac{1}{\sqrt{2}}nH_n + O(n). \quad \square$$

4. Analysis of Algorithm D

Algorithm P is probabilistic, and hence does not constitute in itself a proof that distributed leader-finding can be achieved strictly more efficiently for bidirectional rings than for unidirectional rings. To solve the problem, a deterministic version of Algorithm P is described in [1] in which Stage 1 is replaced by a fully deterministic stage. The idea is to let each processor P_i send its $\langle * \sigma_i \rangle$ -message in the direction of the smallest neighbour and thus get rid of all the smaller neighbours from the outset.

Algorithm D (Bodlaender and Van Leeuwen [1])

Similar to Algorithm P except that for each processor P_i , Stages 1 and 2 are replaced as follows:

Stage 1*: send message $\langle * \sigma_i \rangle$ to both neighbours on the ring: wait for the message $\langle * \sigma_{i-1} \rangle$ and

$\langle * \sigma_{i+1} \rangle$ of both neighbours; $MAX_i := \max\{ * \sigma_{i-1}, * \sigma_i, * \sigma_{i+1} \}$;

if $MAX_i = * \sigma_i$ **then if** $* \sigma_{i-1} < * \sigma_{i+1}$ **then** send message $\langle * \sigma_i \rangle$ to the left
else send message $\langle * \sigma_i \rangle$ to the right

fi

fi;

Stage 2* (election): repeat the following steps, until the end of the election is signaled by receipt of a $\langle ! \rangle$ message: if two messages are received from the left and right simultaneously, then ignore

the smaller one and proceed as if the larger message is received; if message $\langle * \sigma_j \rangle$ is received from a neighbour, then

if $* \sigma_j > \text{MAX}_i$ **then** $\text{MAX}_i := * \sigma_j$; pass message $\langle * \sigma_j \rangle$ on
 else if $\sigma_j = \text{MAX}_i$ **and** $* \sigma_i = * \sigma_j$
 then send message $\langle ! \rangle$ on the ring /* P_i has won the election */
 else if $* \sigma_j = \text{MAX}_i$ **and** $* \sigma_i \neq * \sigma_j$ **then** pass message $\langle * \sigma_j \rangle$ on /* the neighbour of P_i will win the election */

fi;

(Stage 3 remains unchanged).

Stage 1* requires exactly $2n$ messages and leaves at most $\lfloor \frac{1}{2}n \rfloor$ processors active or candidate in the election, viz. the peaks of the permutation $\pi = (\sigma_1 \cdots \sigma_n)$, which clearly pass on to the next stage. The other $\frac{1}{2}n$ remaining processors, the “non-peaks” of π , thus stay in the defeated state after Stage 1*. By pairing every permutation of \mathfrak{S}_n with one in which the neighbours of P_i are interchanged, one can see that P_i sends its $\langle * \sigma_i \rangle$ -message to the left or to the right with probability $\frac{1}{2}$ (averaged over all the permutations of \mathfrak{S}_n).

4.1. Average number of peaks, rises and average length of rises and falls

In order to obtain an exact asymptotic estimate of the average number of messages sent by the active processors that pass on to Stage 2* in Algorithm D, we need to know the average number of these active processors (the peaks) and the average distance between two of them. This we obtain thanks to combinatorial average-case results about permutations, namely the expected number of peaks and rises of π , and the expected length of the rises and falls of π .

4.1. Lemma. *The expected number of peaks of $\pi \in \mathfrak{S}_n$ is $\overline{P}_n = \frac{1}{3}(n + 1)$.*

Proof. To derive P_n , we use combinatorial enumeration methods developed by Flajolet (e.g., in [4]). Let \mathcal{T} denote the (binary) tournament tree associated to the permutation π , and let T_g and T_d denote the left and the right subtree of \mathcal{T} respectively. It is easily seen that the number of peaks of π is the number of leaves of \mathcal{T} . Let $\lambda[\mathcal{T}]$ denote the number of leaves of \mathcal{T} ; then

$$\lambda[\mathcal{T}] = \delta_{|\mathcal{T}|,1} + \lambda[T_g]|\mathcal{T}| + \lambda[T_d]|\mathcal{T}| = \delta_{|\mathcal{T}|,1} + 2|\mathcal{T}|\lambda[T].$$

The ordinary generating function of the expectation of λ is thus

$$\Lambda(z) = \sum \Lambda_n z^n = z + 2 \int_0^z \Lambda(x) \frac{dx}{1-x}, \quad \Lambda(0) = 0$$

which leads to the first-order differential equation

$$(1-z) \frac{d}{dz} \Lambda(z) - 2\Lambda(z) = 1-z$$

with solution

$$\Lambda(z) = \frac{1}{3} \frac{z^3}{1-z^2} + \frac{z}{1-z};$$

and, since $\Lambda(z) = \frac{1}{3} \sum_{n \geq 3} (n-2) z^n + \sum_{n \geq 1} z^n$ (ordinary generating function of $\Lambda(z)$ obtained from the above solution),

$$\Lambda_n = [z^n] \Lambda(z) = \frac{1}{3}(n-2) + 1 = \frac{1}{3}(n+1).$$

The expected number of peaks of π is then $\overline{P}_n = \frac{1}{3}(n+1)$. \square

Note that, as a consequence, one obtains a well-known result in the theory of permutations: the average number of peaks and valleys of π is $\frac{1}{2}(n + 1)$.

4.2. Lemma. *The expected number of rises of $\pi \in \mathfrak{S}_n$ is $\overline{R}_n = \frac{1}{2}(n + 1)$.*

Proof. Comtet, for example in [2], shows that the Eulerian numbers, $A(n, k)$ count the number of permutations $\pi \in \mathfrak{S}_n$ with k rises. The exponential generating function of Eulerian numbers $U(t, z) = 1 + \sum_{n,k} A(n, k)(z^n/n!)t^{k-1}$ has the value

$$U(t, z) = \frac{1 - t}{e^{z(t-1)} - 1}. \tag{4.1}$$

Whence, by developing (4.1) with regard to $(t - 1)$ we obtain

$$U(t, z) = \left(\sum_{n \geq 1} z^n \right) + (1 - t) \left(\sum_{n \geq 2} \frac{n-1}{2} z^n \right) + \frac{(1-t)^2}{2} \left(\sum_{n \geq 3} \frac{(n-2)(3n-5)}{12} z^n \right) + \dots$$

Considering $U(t, z)$ and its derivatives in $t = 1$, we obtain

$$[z^n] U(t, z) |_{t=1} = \sum_k A(n, k) = n!, \tag{4.2}$$

$$[z^n] \frac{\partial}{\partial t} U(t, z) |_{t=1} = \sum_k (k-1) A(n, k) = \frac{1}{2}(n-1)n!, \tag{4.3}$$

$$[z^n] \frac{\partial^2}{\partial t^2} U(t, z) |_{t=1} = \sum_k (k-1)(k-2) A(n, k) = \frac{(n-2)(3n-5)}{12} n!.$$

The mean $\overline{R}_n = (4.3)/(4.2) + 1 = \frac{1}{2}(n + 1)$, and the variance $\text{var}(R_n) = (n + 1)/12$ are easily derived from the above identities. In this case, the standard deviation is of order \sqrt{n} . The central limit theorem shows that R_n , when normalized, converges to the normal distribution. \square

4.3. Lemma. *The expected length of rises and falls of $\pi \in \mathfrak{S}_n$ is $\overline{L}_n = 2n/(n + 1)$.*

Proof. Let us first recall the fundamental one-to-one correspondence (due to Françon and Viennot [5]) between permutations of \mathfrak{S}_n and weighted paths [6].

Besides, define the set E_n of "subexceeding functions" on $[1, n]$ to be the set of functions f on $[1, n]$ such that $f(i) \leq i$ for all i in $[1, n]$. Then there exists a one-to-one mapping between the set E_n and \mathfrak{S}_n which may be pictured as the correspondence between the crossed squares of the inversion table of a permutation and the corresponding value $f(i) - 1$ (for all $1 \leq i \leq n$), f being then a subexceeding function (see [10] for a more detailed argumentation). E_n may be described by means of the polynomial

$$F(x_1, \dots, x_n) = x_1(x_1 + x_2) \cdots (x_1 + x_2 + \cdots + x_n),$$

from which we can obtain

$$F_j(t) = (j-1)!(t+j-1) \cdots (t+n-1),$$

the generating polynomial of subexceeding functions on $[1, n]$ according to the number of times when

value j is reached: $F_j(t)$ corresponds to the length of a rise or a fall of π . By summing up $F_j(t)$ on index j from 1 to n , one gets the generating polynomial of the total length of rises and falls of permutations π . Let

$$\Phi(t) = \sum_{j=0}^{n-1} \{j!(t+j) \cdots (t+n-1)\} - n! \frac{n-1}{2} \tag{4.4}$$

be this generating polynomial. From (4.4)

$$\begin{aligned} \Phi(1) &= n \cdot n! - n! \frac{n-1}{2} = \frac{n+1}{2} n!, \quad \text{and} \\ \Phi'(1) &= \sum_{j=0}^{n-1} j! \sum_{p=j+1}^n p^{-1} + n! \sum_{p=1}^n p^{-1} \sum_0^{p-1} = n \cdot n!. \end{aligned}$$

Whence $\bar{L}_n = \Phi'(1)/\Phi(1) = 2n/(n+1)$; the expected length of rises and falls of π . \square

Note that the variance is $2n(n+3)/(n+1)^2 - (4/(n+1))H_n$, and the central limit theorem shows that L_n , when normalized, converges to the normal distribution.

4.2. *Exact asymptotic estimation of the expected number of messages*

At the end of Stage 1*, there remain $\frac{1}{3}(n+1)$ active processors on the average (the peaks of π). We know from Lemma 4.3 that the average length of rises and falls of $\pi \in \mathfrak{S}_n$ is $\bar{L}_n = 2n/(n+1)$. Now, \bar{L}_n also represents the average distance between two consecutive peaks, and thus, the average distance between two consecutive peak-processors in the ring.

4.4. Theorem. *The asymptotic value of the expected number of messages used in Algorithm D is $nH_n/\sqrt{2} + O(n) = 0.707106 \dots n \ln n + O(n)$.*

Proof. Denote by $n^* = \frac{1}{3}(n+1)$ the average number of peak-processors and by \bar{L}_n the average distance between two consecutive peak-processors.

At the end of Stage 1*, the remaining active peak-processors are at least one position apart, and the independence of their choice of direction in sending their messages around the ring (left or right) is a priori not guaranteed for all of them. Indeed, for an arbitrary pair i and j , the random variables for the directions of the peaks' messages ($\langle * \sigma_i \rangle$ and $\langle * \sigma_j \rangle$) are in general not independent. However, as proved in [12, pp. 580–582], these random variables satisfy a condition weaker than independence as they are pairwise uncorrelated. Instead of having independent identically distributed random variables, we may claim that their process is stationary. In other words, the distribution of the messages' directions does not depend on the placement of the peak-processors (the proof is given by Breiman in [2, Chapter 6, p. 105 and p. 120—Problems 14 and 15]).

Hence from [2] and [12], fortunately, every message within Stage 2* travels along the ring in a direction irrespective of the pairwise positions of peak-processors (this “weak” form of dependence is illustrated in [10]). Therefore, the probability that a message is sent to the right or the left is $\frac{1}{2}$, and we are brought back to the average case analysis of Algorithm P.

In this case then (Algorithm P revisited), the asymptotic expected number of messages is $n^*H_{n^*}/\sqrt{2} + n^*\bar{L}_nH_n/\sqrt{2} + O(n)$, when accumulating the average distances \bar{L}_n for all the n^* peak-processors. And since

$$\frac{1}{\sqrt{2}} n^* \left(1 + \frac{2n}{n+1}\right) H_{n^*} + O(n) = \frac{1}{\sqrt{2}} \frac{(n+1)}{3} \left(1 + \frac{2n}{n+1}\right) H_n + O(n),$$

the asymptotic expected number of messages propagated in Algorithm D is

$$\frac{1}{\sqrt{2}} \frac{3n+1}{3} H_n + O(n) = \frac{1}{\sqrt{2}} n H_n + O(n). \quad \square$$

Note that we assumed that all the processors start the election “simultaneously” and work synchronously. The first assumption allows us not to consider the case when there exist $p < n$ initiators, and the second assumption (together with the first one) yields an $O(n)$ “time” complexity for both Algorithm P and Algorithm D: viz. in the best case, $n + \frac{1}{2}n$ and $n + 1 + \frac{1}{2}n$ pips respectively; in the worst case, $3n$ and $3n + 1$ pips respectively; on the average $2n$ and $2n + 1$ pips respectively. In this case, the $\frac{1}{2}n$ pips applies to the delay time elapsed for the inauguration of the leader. As to the worst-case message complexity of Algorithm P and Algorithm D, Bodlaender and Van Leeuwen proved in [1] that the maximum number of messages is $\frac{1}{2}n^2$ and $\frac{1}{4}n^2$ respectively.

5. Conclusions

We have presented a detailed analysis which shows that the probabilistic algorithm Algorithm P as well as the deterministic algorithm Algorithm D have the same asymptotic average message complexity for the election problem in a bidirectional ring. The present result is a confirmation of the tight bounds derived in [1] and of the proof in [1] that distributed leader-finding can be solved strictly more efficiently in bidirectional rings than in unidirectional rings by a deterministic algorithm.

Indeed, combinatorial enumeration and analytic methods (e.g., generating functions) prove powerful and general enough to provide efficient tools and to cope with most average-case analyses of distributed algorithms and distributed data structures.

However, the bidirectional variants of distributed election algorithms seem surprisingly hard to analyse on the average, and it is still an open problem to find an exact expression for the variance of Algorithm P and Algorithm D, as well as to characterize the distribution of asymptotic constants for other bidirectional distributed election algorithms.

References

- [1] H.L. Bodlaender and J. van Leeuwen, New upper bounds for decentralized extrema-finding in a ring of processors, In: B. Monien and G. Vidal-Naquet, eds., *Proc. STACS*, Lecture Notes in Computer Science, Vol. 210 (Springer, Berlin, 1986) 119–129.
- [2] L. Comtet, *Analyse Combinatoire, Tomes 1 et 2* (Presses Universitaires de France, Paris, 1970).
- [3] P. Flajolet, Personal communication, 1987.
- [4] P. Flajolet and J.S. Vitter, *Average-case Analysis of Algorithms and Data Structures*, INRIA Res. Rept. 718, August 1987.
- [5] J. Françon and G. Viennot, Permutations selon les pics, creux, double montées, double descentes, nombres d’Euler et de Genocchi, *Discrete Math.* **28** (1979) 21–35.
- [6] I.P. Goulden and D.M. Jackson, *Combinatorial Enumeration* (Wiley, New York, 1983).
- [7] D. Knuth, *The Art of Computer Programming—Sorting and Searching, Vol. 3* (Addison-Wesley, Reading, MA, 1973).
- [8] E. Korach, D. Rotem and N. Santoro, *A Probabilistic Algorithm for Decentralized Extrema-finding in a Circular Configuration of Processors*, Res. Rept. CS 81-19, Dept. of Computer Science, University of Waterloo, 1981.
- [9] C. Lavault, *Algorithmique et Complexité Distribuées*, Thèse d’Etat, Université de Paris XI-Orsay, December 1987.
- [10] C. Lavault, *Average Number of Messages for Distributed Leader-finding in Rings of Processors*, INRIA Res. Rept. 840, April 1988.
- [11] J. van Leeuwen and R.B. Tan, An improved upperbound for distributed election in a bidirectional ring of processors, *Distributed Comput.* **2** (1987) 149–160.
- [12] D. Rotem, E. Korach and N. Santoro, Analysis of a distributed algorithm for extrema-finding in a ring, *J. Parallel Distributed Comput.* **4** (1987) 575–591.
- [13] J. Vuillemin, A unifying look at data structures, *Comm. ACM* **23** (1980) 229–239.