Theoretical Computer Science 73 (1990) 61-79
North-Holland

EXACT AVERAGE MESSAGE COMPLEXITY VALUES FOR DISTRIBUTED ELECTION ON BIDIRECTIONAL RINGS OF PROCESSORS *

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

Communicated by M. Nivat
Received May 1988
Revised July 1988

Abstract. Consider a distributed system of n processors arranged on a ring. All processors are labeled with distinct identity-numbers, but are otherwise identical. In this paper, we make use of combinatorial enumeration methods in permutations and derive the one and the same exact asymptotic value, $\frac{1}{2}\sqrt{2}nH_n + O(n)$, of the expected number of messages in both probabilistic and deterministic bidirectional variants of Chang-Roberts distributed election algorithm. This confirms the result of Bodlaender and van Leeuwen (1986) that distributed leader finding is indeed strictly more efficient on bidirectional rings of processors than on unidirectional ones.

1. Introduction

We consider the problem of finding a leader on an asynchronous bidirectional ring of processors. Each site (processor) is distinguished by a unique identification number (its "identity"). There is no central controller and every processor only has local information about the network topology, 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) using a minimum number of messages. Note that the problem is equivalent (up to $O(n)$ extra messages) to the problem of determining the identity of the largest processor on the ring.

We assume that the processors work fully asynchronously and cannot use clocks or timeouts. 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 communication subsystem to be error-free and that links operate in a FIFO-manner.

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 on the ring will also be called a message. Finally, *pip* will denote the traversal delay of one elementary message (we assume here that all message delay times are equal).

* This work was supported by C-cube, CNRS GRECO 65 ("COPARADIS" Group).

Much work has already been done to obtain good upper and lower bounds for different variants of the problem, both in the worst and in the average case. Tight upper and lower bounds for bidirectional variants of Chang–Roberts decentralized extrema-finding algorithm were presented in [3]. These bounds were established for the probabilistic algorithm given in [12] and [18] (Algorithm P) and for a deterministic version of the same algorithm (Algorithm D). Up until now, these bounds were the best approximation of the average number of messages required by Algorithm P and Algorithm D.

In this paper, we derive the exact asymptotic value of the average number of messages required both in Algorithm P and Algorithm D: $\frac{1}{2}\sqrt{2}nH_n + O(n)$. This value is obtained by using techniques and results from theory of permutations (inversion tables mainly), average-case analyses involving generating functions (e.g. generating function of Eulerian numbers), and asymptotic techniques (e.g. Stirling formula and Euler–Maclaurin summation formula).

The result confirms the positive answer (given in [3]) to the question (first posed by Pachl, Korach and Rotem) of whether distributed leader finding can be solved more efficiently on bidirectional rings than on unidirectional rings.

The paper is organized as follows. In Section 2, some definitions and preliminary results from the theory of permutations are given. Sections 3 and 4 are devoted to the analysis of the exact asymptotic value of the expected number of messages used in Algorithm P and Algorithm D, respectively: $\frac{1}{2}\sqrt{2}nH_n + O(n)$. In Section 5, we present the results of experimental tests, achieved on a bidirectional circular configuration of 1000 up to 50 000 processors. The experimental values thus obtained may be considered to be in good agreement with the preceding theoretical average value.

Tables 1 and 2 present an overview of the existing known upper bounds for the leader finding problem, in rings where the size n is unknown to all the processors, and (a priori) without sense of direction on the ring, in the bidirectional variant of the problem.

Table 1
Distributed election algorithms for unidirectional ring

Algorithm	Average	Worst case
Le Lann (1977)	n^2	n^2
Chang and Roberts (1979)	nH_n	$\frac{1}{2}n^2$
Peterson (1982)		$1.44\ldots n \lg n$
Dolev, Klawe and Rodeh (1982)		$1.356\ldots n \lg n$

For most of the quoted bidirectional algorithms, the existence of a global sense of direction (i.e. each processor has the consistent global knowledge of the left and right direction on the ring) is unnecessary, although this is of course not the case for the algorithm of Dolev et al. Thus, the average message complexity of Algorithm P or Algorithm D is the same on a ring with or without a global sense of direction.

Table 2
Distributed election algorithms for bidirectional rings

Algorithm	Average	Worst case
Gallager et al. (1979)		$5n \lg n$
Franklin (1982)		$2n \lg n$
Korach, Rotem and Santoro (1981) Algorithm P	[prob] $\leq \frac{3}{4} nH_n$	[prob] $\frac{1}{2}n^2$
Korach, Rotem and Santoro (1984)		$1.89\ldots n \lg n$
Bodlaender and van Leeuwen (1986) Algorithm P	$0.7033\ldots nH_n \leq$ [prob] $\leq 0.7075\ldots nH_n$	[prob] $\frac{1}{2}n^2$
(1986) Algorithm D	[det] $\leq 0.7075\ldots nH_n$	[det] $\frac{1}{4}n^2$
Van Leeuwen and Tan (1985)		$1.44\ldots n \lg n$
Moran, Shalom and Zaks (1985)		$1.44\ldots n \lg n$
Dolev, Klawe and Rodeh (1982) with sense of direction		$1.356\ldots n \lg n$
This paper (1988) Algorithm P	[prob] $\frac{1}{2}\sqrt{2}nH_n$	[prob] $\frac{1}{2}n^2$
This paper (1988) Algorithm D	[det] $\frac{1}{2}\sqrt{2}nH_n$	[det] $\frac{1}{4}n^2$
This paper (1988) Algorithm D	$(0.972\ldots/\sqrt{2})nH_n$ [a]	

[a] Result experimentally obtained (Section 5).

The existence of a sense of direction on a bidirectional ring does not actually shrink the *average* message complexity of distributed *extrema-finding* algorithms, although this is a priori not the case for *any* bidirectional distributed election algorithm.

The average lower bound of $\frac{1}{2}nH_n$ on bidirectional rings (with sense of direction, n unknown), derived by Bodlander in [2], displays the existing gap between the coefficients $\frac{1}{2}$ and $\frac{1}{2}\sqrt{2}$ for the average message complexity of the bidirectional distributed leader finding problem (with or without sense of direction on the ring).

2. Inversion tables

Let $\pi = (\sigma_1 \sigma_2 \ldots \sigma_n) \in \mathfrak{S}_n$ be a permutation of size n. Associated with π, define its *inversion table* $t = t_1 t_2 \ldots t_n$ such that t_i is the number of elements (in π) to the left of σ_i larger than σ_i ($1 \leq i \leq n$). Hence, $0 \leq t_i < i$ for all i ($1 \leq i \leq n$), and the correspondence between a permutation and its inversion table is *one to one.*

Lemma 2.1 (Vuillemin [19]). *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 tables of π.*

Example 2.2. Let $\pi = (2\,3\,6\,5\,1\,7\,4)(\pi \in \mathfrak{S}_7)$. The (bold) upper records of π are

$$\mathbf{2}\quad \mathbf{3}\quad \mathbf{6}\quad 5\quad 1\quad \mathbf{7}\quad 4,$$

and the inversion table of π is such that

$$t = 0 \quad 0 \quad 0 \quad 1 \quad 4 \quad 0 \quad 3.$$

An inversion table can thus be pictured as a "staircase": put a cross in each column, the left-to-right maxima are corresponding to the lower crosses (the zeros). Now using variables to denote the values $0, 1, \ldots, n-1$ in the inversion table t of π (e.g. $x_1, x_2, \ldots, x_{n-1}$), the set of all inversion tables corresponding to all permutations in $\mathfrak{S}_n$ is fully described by the polynomial

$$\underset{\text{1st column}}{\underset{\uparrow}{x_0}}\underset{\text{2nd column}}{\underset{\uparrow}{(x_0+x_1)}}\underset{\text{3rd column}}{\underset{\uparrow}{(x_0+x_1+x_2)}}\ldots\underset{(n-1)\text{st column}}{\underset{\uparrow}{(x_0+x_1+\cdots+x_{n-1})}} \tag{2.1}$$

Notation

(i) The nth harmonic number is denoted by $H_n = \sum_{i=1}^{n} 1/i$ with asymptotic expansion $H_n = \ln n + \gamma + \frac{1}{2}n^{-1} + O(n^{-2}) \sim 0.693 \ldots \lg n$ (where $\gamma = 0.577\ldots$ is Euler's constant); $\ln n$ is the neperian logarithm of n and $\lg n$ denotes the logarithm in 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 $f(x)$".

Definition 2.3 (Feller [5, p. 48]). The unsigned Stirling number of first kind, $s_{n,k}$, are such that $s_{n,k} = [x^k]x(x+1)(x+2)\ldots(x+n-1)$; and the enumerating polynomial of $s_{n,k}$ is $\sum_k s_{n,k}x^k = x(x+1)(x+2)\ldots(x+n-1)$.

Lemma 2.4 (Vuillemin [19]). *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 first kind $s_{n,k}$.*

The unsigned Stirling number of first kind, $s_{n,k}$, is proved to count at the same time: the permutations $\pi \in \mathfrak{S}_n$ with k upper records, the permutations $\pi \in \mathfrak{S}_n$ with k cycles, and the inversion tables of size n such that $|\{i \mid 1 \leq i \leq n, t_i = 0\}| = k$ (see [19, p. 231]).

Therefore, an immediate consequence of Lemmas 2.1 and 2.4 is the following.

Corollary 2.5. *The average number of left-to-right maxima (upper records) of a permutation $\pi \in \mathfrak{S}_n$ is H_n.*

Lemma 2.6. *Let $\pi = (\sigma_1\sigma_2\ldots\sigma_n)$. The average distance to the first left-to-right maximum (upper record) of π is $H_n - 1$.*

This well-known result occurs repeatedly in the theory of permutations [5, 17, 11, 19, 3]. However, this result may be obtained with very direct proofs, using two arguments involving different properties of $\pi \in \mathfrak{S}_n$ [13, p. 142].

Proof. On the one hand, it represents nothing indeed but the average number of left-to-right maxima of π, whenever we assume its first element σ_1 to be an upper record. Hence, since the first upper record of $\pi(\sigma_1)$ is not counted in the enumeration, the average distance to the first upper record, different from σ_1, is the average number of occurrences of the value zero in the inversion tables of size n, minus 1: i.e. $H_n - 1$.

On the other hand, the average distance to the first upper record of π may simply be derived as the direct solution of a linear recurrent equation. Let D_n be the distance to the first upper record of π, then D_n satisfies the recurrence

$$D_n = D_{n-1} + (n-1)D_{n-1} + (n-1)!, \quad \text{subject to } D_1 = 0$$

(since σ_1 is the first left-to-right maximum). Now the average value of D_n is

$$D_n/n! = D_{n-1}/(n-1)! + 1/n, \quad \text{subject to } D_1 = 0;$$

from which we obtain the solution

$$\overline{D_n} = \frac{D_n}{n!} = \sum_{2 \le i \le n} 1/i = H_n - 1. \qquad \square$$

Proposition 2.7. *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*

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

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, so that $G_n(x)$ is conditioned over the values α and β. Let us first 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* corresponding to the possible upper records of π.

Positions	$1\ 2\ 3 \ldots \ldots (\alpha-1)\alpha(\alpha+1) \ldots \quad \ldots (\beta-1)\beta(\beta+1) \quad \ldots n$
Monomials	$1\ 1\ 2 \ldots (\alpha-2)x(x+\alpha)(x+\alpha+1) \ldots (x+\beta-2)x(\beta+1) \ldots n.$

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,

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

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

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

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

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

3. Analysis of Algorithm P

We know from [12] that Algorithm P (Fig. 1) requires an expected number of messages of at most $\frac{3}{4}nH_n + O(n)$. This value is only an upper bound, because of possible effects of higher order upper records which remain ignored in this evaluation. The detailed proof of correctness of the algorithm may be found in [3], or [16].

In the following, we assume that all processors start the algorithms simultaneously (at time 0), otherwise the first message a processor receives serves to wake it up and trigger its Stage 1, before it actually processes the message. For the analyses, we also assume throughout the paper that the processors operate synchronously, and that the algorithms can deal with the case when a processor receives two messages (from both neighbours) at the same moment. Furthermore, we ignore the

---

Algorithm P [3]

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)
$\mathrm{MAX}_i := \sigma_i$;
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 $\sigma_j > \mathrm{MAX}_i$ **then** $\mathrm{MAX}_i := \sigma_j$; pass message $\langle \sigma_j \rangle$ on
else if $\sigma_j = \mathrm{MAX}_i$ **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.

Fig. 1.

time necessary for possible internal computations within the processors. These assumptions enable us to have asynchronous, message-driven algorithms, running on a synchronous ring with a fully deterministic behaviour [3]; however, similar results can be proved when weaker assumptions hold.

3.1. Exact evaluation of the expected number of messages

Consider a ring of n processors $P_1, \ldots, P_n$ 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. And thus, $\pi = (\sigma_1\sigma_2 \ldots \sigma_n)$ 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 $\frac{1}{2}$. Thus, the expected number of elements in π visited by $\langle\sigma_1\rangle$ 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 2.6, $H_n - 1$ 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 upperbound 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, the exact average value can be determined.

Proposition 3.1. *The average number of* $\langle\sigma_1\rangle$*-messages propagated by Algorithm* P *is exactly* $\frac{3}{4}H_n - \sum_{\alpha,\beta} (\alpha - \frac{1}{2}\beta)G_n(\frac{1}{2}) + O(1)$, *for* $\frac{1}{2}\beta < \alpha < \beta < 2\alpha$, *where* $G_n(x)$ *is the generating polynomial defined in* (2.2).

Proof. Let $\nu_1, \nu_2, \ldots,$ be random variables denoting the position of the first, second, and higher order upper records. We may consider the ν_is as independent random variables conditioned over the values α and β; without loss of generality let $\langle\sigma_1\rangle$ be sent to the right. If processors P_α to $P_{\beta-1}$ randomly choose to send their message to the *right* while 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. Let π denote the permutation $(\sigma_1 \ldots \sigma_n) \in \mathfrak{S}_n$; then 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 consider the effect of *all* upper records of π. The average number of $\langle\sigma_1\rangle$-messages propagated by Algorithm P is

$$\overline{N_1} = \tfrac{1}{2}(H_n - 1) + \tfrac{1}{2}\lfloor\tfrac{1}{2}H_n\rfloor - \sum_{j\geq 2} 2^{-j} \sum_{\alpha,\beta} (\alpha - \lfloor\tfrac{1}{2}\beta\rfloor - 1)\Pi_n(j) \tag{3.1}$$

and, since

$$\sum_{j\geq 2} 2^{-j}\Pi_n(j) = \sum_{j\geq 2} 2^{-j}\{[x^{j-2}]G_n(x)\} = \tfrac{1}{4} \sum_{j\geq 2} 2^{-j}\{[x^j]G_n(x)\} = \tfrac{1}{4}G_n(\tfrac{1}{2}),$$

then

$$\overline{N_1} = \tfrac{3}{4}H_n - \tfrac{1}{4}\sum_{\alpha,\beta}(\alpha - \tfrac{1}{2}\beta)G_n(\tfrac{1}{2}) + O(1), \quad \tfrac{1}{2}\beta < \alpha < 2\alpha. \tag{3.2}$$

Shifting from $\alpha - \lfloor \frac{1}{2}\beta \rfloor - 1$ to $\alpha - \frac{1}{2}\beta$ produces indeed an O(1) error term; besides, α ranges from $\frac{1}{2}\beta + 1$ to $\beta - 1$, and β ranges from 2 to n. □

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 $[x^j]G_n(x)$.

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

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

Proof. Accumulating in (3.2) the quantity $\overline{N_1}$ 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

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

Now,

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

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

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

which gives the exact average number of messages:

$\frac{3}{4}nH_n - nS_n + O(n)$. □

3.2. Asymptotic analysis of S_n

The following asymptotic analysis of S_n in [6], makes use of Stirling's formula and Euler–Maclaurin (one-dimensional) summation formula (see [7] for similar techniques). For $p \in \mathbb{N}$ define

$$Q_p = (1+\tfrac{1}{2})(1+\tfrac{1}{4})\cdot\ldots\cdot\left(1+\frac{1}{2p}\right) = \frac{3\cdot5\cdot7\cdot\ldots\cdot(2p+1)}{2\cdot4\cdot6\cdot\ldots\cdot(2p)} = \frac{(2p+1)!}{2^{2p}(p!)^2};$$

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 $O(1/p)$.

The sum S_n may then be rewritten with the Q_ps as follows:

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

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

$$u(\beta) = \sum_\alpha \frac{(\alpha - \frac{1}{2}\beta)}{\alpha\beta^2}\sqrt{\frac{\beta}{\alpha}}\left(1 + \mathrm{O}\left(\frac{1}{\beta}\right)\right), \quad \tfrac{1}{2}\beta < \alpha < \beta.$$

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

$$u(\beta) = \frac{1}{\beta}\left\{\sum_t \frac{(t-\frac{1}{2})}{t\sqrt{t}}\frac{1}{\beta}\right\}\left(1 + \mathrm{O}\left(\frac{1}{\beta}\right)\right);$$

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

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

which is uniform in β. And at last,

$$u(\beta) = \frac{1}{\beta}\left\{3 - 2\sqrt{2} + \mathrm{O}\left(\frac{1}{\beta}\right)\right\} \tag{3.5}$$

which yields the following.

Theorem 3.3. *The asymptotic value of the expected number of messages used in Algorithm* P *is* $\frac{1}{2}\sqrt{2}nH_n + \mathrm{O}(n) = 0.707106\ldots n \ln n + \mathrm{O}(n)$.

Proof. From (3.3),

$$\overline{N} = \tfrac{3}{4}nH_n - nS_n + \mathrm{O}(n).$$

The exact computation of S_n from (3.4) and (3.5) gives rise to the expression

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

and since

$$\sum_\beta \mathrm{O}\left(\frac{1}{\beta^2}\right) = \mathrm{O}(1), \quad S_n \text{ being uniform in } \beta,$$

$$\overline{N} = \tfrac{3}{4}nH_n - n\tfrac{1}{4}(3 - 2\sqrt{2})\sum_\beta \frac{1}{\beta} + \mathrm{O}(n)$$

$$= \tfrac{1}{2}\sqrt{2}nH_n + \mathrm{O}(n). \qquad \square$$

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 in bidirectional rings than in unidirectional rings. To solve the problem, a deterministic version of Algorithm P is described in [3] 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 (Fig. 2).

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 $\pi = (\alpha_1 \ldots \sigma_n)$, which clearly pass on to the next stage. The other $\lceil \frac{1}{2}n \rceil$ remaining processors, the "non-peaks" of π, thus stay in the state defeated 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 $\mathfrak{S}_n$).

Algorithm D [3]
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 (with the indices "$i-1$" and "$i+1$" interpreted in the usual circular sense as indices of the left and right neighbour, respectively);
$\text{MAX}_i := \max\{*\sigma_{i-1}, *\sigma_i, *\sigma_{i+1}\}$;
if $\text{MAX}_i = *\sigma_i$ **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 the 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)

Fig. 2.

For the analysis of Algorithm D, our assumptions (emphasized in Section 3) still hold in the following analysis.

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

In order to obtain an exact asymptotic estimation 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.

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

Following is a simple proof of Lemma 4.1, originally proved by Bienaymé in [1].

Proof. 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 $\mathcal{T}$ [7]. Let $\lambda[\mathcal{T}]$ denote the number of leaves of $\mathcal{T}$, then

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

Thus, the ordinary generating function (ogf) of the expectation of λ is

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

which leads to the first-order differential equation

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

with the solution

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

and, since

$$A(z) = \frac{1}{3}\sum_{n\geq 3}(n-2)z^n + \sum_{n\geq 1} z^n$$

(ogf of $A(z)$ obtained from the above solution),

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

The expected number of peaks of π is then $\overline{P_n} = \frac{1}{3}(n+1)$. The central limit theorem applies; however, since the variance is of order n^3 (and whence the standard deviation of order $n^{3/2}$), it cannot be used to derive the distribution of P_n. □

Note that the average number $\overline{V_n}$ of valleys of π is such that $\overline{V_n}=\overline{P_n}-1=\frac{1}{3}(n-2)$, and that the average number of peaks and valleys of π is then $\frac{1}{3}(2n-1)$, a well-known result in the theory of permutations.

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

Proof. Louis Comtet, for example in [4], shows that the eulerian numbers $A(n,k)$ count the number of permutations $\pi\in\mathfrak{S}_n$ with k rises. The bivariate exponential generating function (bgf) of eulerian numbers,

$$U(t,z)=1+\sum_{n,k}A(n,k)\frac{z^n}{n!}t^{k-1}$$

has the value (see [4, p. 63, T.1])

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

Hence, developing (4.1) with regard to $(t-1)$,

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

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

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

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

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

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

Note that the expected number of falls of π is then $\frac{1}{2}(n-1)$. □

Lemma 4.3. *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 J. Françon and G. Viennot [9, 10]) between permutations of $\mathfrak{S}_n$ and weighted paths, with $n-1$ steps, from altitude 0 to altitude 0 with possibility functions: $\mathrm{pos}_1(k)=k+1$, $\mathrm{pos}_0(k)=2(k+1)$ and $\mathrm{pos}_{-1}(k)=k+1$ (recall that the initial and final elements of a permutation $\pi\in\mathfrak{S}_n$ are recognized by notationally placing a zero at both ends of π; whereas in a circular configuration of processors, we assume that of course the "first" and "last" elements of π are the same).

Define the set E_n of "subexceeding functions" on $[1, n]$ [8], 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 seen as the correspondence between the crossed squares of the inversion table of a permutation (recall Lemma 2.1) and the corresponding value $f(i)-1$ (for all $1 \leq i \leq n$), f being then a subexceeding function (see [8, 14] for a more detailed argumentation).

E_n may be described by means of the formal (non-commutative) polynomial $F(x_1, \ldots, x_n) = x_1(x_1+x_2)\ldots(x_1+x_2+\cdots+x_n)$, from which we can obtain the generating polynomial $F_j(t) = (j-1)!(t+j-1)\ldots(t+n-1)$, 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 $F_j(t)$, j ranging from 1 to n, one obtains the generating polynomial of the total length of rises and falls of permutations π, provided that the constant term in $F_j(t)$ has value zero. Let

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

be this generating polynomial. From (4.4),

$$\Phi(1) = n \cdot n! - n!\frac{n-1}{2} = \frac{n+1}{2}n!,$$

$$\Phi'(1) = \sum_{j=0}^{n-1} j! \sum_{p=j+1}^{n} p^{-1} + n! \sum_{p=1}^{n} p^{-1} \sum_{0}^{p-1} 1 = n \cdot n!.$$

Hence, the expected length of rises and falls of π is $\overline{L_n} = \Phi'(1)/\Phi(1) = 2n/(n+1)$. □

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 average (the peaks of π). The remaining active peak-processors are at least one position apart, and *the independence of their choice of direction* for sending 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 the following lemma, these random variables satisfy a condition weaker than independence as they are *pairwise independent*, in the sense of Feller [5, p. 127 and p. 220].

The permutations of $\mathfrak{S}_n$ are assumed equally likely, and also the order type of the resulting configuration of the peaks of π is again assumed to be random at the end of Stage 1*.

Let $d_1, \ldots, d_p$ ($1 \leq p \leq \lfloor n/2 \rfloor$) be the sequence of p random variables denoting the directions towards which the peak-processors' messages $\langle {}^*\sigma_1 \rangle, \ldots, \langle {}^*\sigma_p \rangle$ are sent on the ring, and let this sequence of variables be such that d_i equals the left or right direction on the ring for the message $\langle{}^*\sigma_i\rangle$. From [5, p. 127 and p. 220], we have the following.

Definition 4.4. The p random variables $d_1, \ldots, d_p$ are pairwise independent if they are not mutually independent though their distributions and joint distributions verify the identity

$$\Pr\{d_i = l, d_j = r\} = \Pr\{d_i = l\} \cdot \Pr\{d_j = r\}, \tag{4.5}$$

for an arbitrary pair i and j $(1 \leq i, j \leq p)$.

Lemma 4.5. (i) *In the case where two consecutive peak-processors are at least two positions apart on the ring, we may claim the independence of their choice of direction at the end of Stage* 1*.

(ii) *In the case where two consecutive peak-processors are exactly one position apart on the ring, the directions* $d_1, \ldots, d_p$ *at the end of Stage* 1* *are not mutually independent random variables, but satisfy the weaker (and sufficient) property of being pairwise independent, according to identity* (4.5) *of Definition* 4.4.

Proof. The first claim of Lemma 4.5 is straightforward, since the direction of messages at the end of Stage 1* cannot depend on the respective placement of consecutive peak-processors which are at least two positions apart around the ring [3].

The proof of the claim (ii) of Lemma 4.5 is a little more involved. Let us consider two consecutive peak-processors P_i and P_j which are *exactly* one position apart around the ring and the directions d_i and d_j, respectively $(1 \leq i \leq p)$. Suppose $d_i = r$ and $d_j = l$, for example; the following equalities hold:

$$\Pr\{d_i = r\} = \Pr\{{}^*\sigma_{i-1} < {}^*\sigma_{i+1}\} = \tfrac{1}{2},$$

$$\Pr\{d_j = l\} = \Pr\{{}^*\sigma_{j+1} < {}^*\sigma_{j-1}\} = \tfrac{1}{2}$$

$(i-1, i+1$, etc. being taken modulo n).

Now,

$$\begin{aligned}\Pr\{d_i = r, d_j = l\} &= \Pr\{{}^*\sigma_{i-1} < {}^*\sigma_{i+1}, {}^*\sigma_{j+1} < {}^*\sigma_{i-1}\} = \tfrac{1}{2} \cdot \tfrac{1}{2} \\ &= \Pr\{d_i = r\} \cdot \Pr\{d_j = l\},\end{aligned} \tag{4.6}$$

and the above identities (4.6) still hold for $d_i = l$, $d_j = r$, or $d_i = d_j = l$, etc. But the p identically distributed random variables $d_1, \ldots, d_p$ are not mutually independent whenever *at least two of them* correspond to peak-processors which are *exactly* one position apart on the ring. In such a case indeed, the distributions and joint distributions verify the inequality

$$\Pr\{d_1 = D, \ldots, d_p = D\} \neq \Pr\{d_1 = D\} \cdot \ldots \cdot \Pr\{d_p = D\}, \quad D \in \{l, r\}. \qquad \square$$

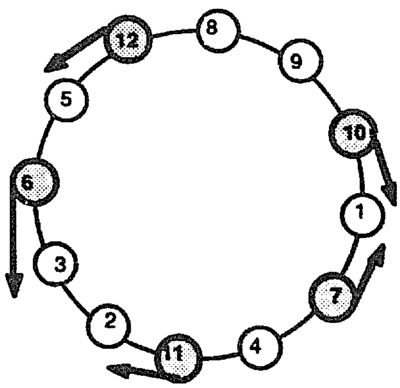

Fig. 3.

Example 4.6. As an example, we may consider the permutation $\pi \in \mathfrak{S}_{12}$ with (bold) peaks **10, 7, 11, 6,** and **12**: $\pi = (8 \quad 9 \quad \mathbf{10} \quad 1 \quad \mathbf{7} \quad 4 \quad \mathbf{11} \quad 2 \quad 3 \quad \mathbf{6} \quad 5 \quad \mathbf{12})$. According to the Stage 2* in Algorithm D, the choice of direction of processors P_{10}, P_7, P_{11}, P_6 and P_{12} only depends on the respective identities of their (immediate) right and left neighbours on the ring (Fig. 3). P_{10} sends its ⟨*10⟩-message clockwise (towards P_1), and so does P_{11} (towards P_2); whereas P_7, P_6 and P_{12} send their respective messages anticlockwise (towards P_1, P_3 and P_5, respectively) around the ring.

This "weak" form of dependence is illustrated in the fact that P_6 and P_{12} on the one hand, P_{10} and P_7 on the other hand, are pairwise consecutive peak-processors which send their messages in the same direction around the ring for the first pair, whereas the peak-processors of the second pair send their messages in opposite directions.

In other words, this is an example in which *no three consecutive* random variables of the sequence $d_1, \ldots, d_p$ are independent. Hence, the d_is are pairwise independent ((4.5) or (4.6) are verified) without being mutually independent, and the distribution of the messages' directions does not depend on the placement of pairwise peak-processors, even in the case where the latter are exactly one position apart on the ring (P_{10} and P_7, or P_6 and P_{12}).

Theorem 4.7. *The asymptotic value of the expected number of messages used in Algorithm* D *is* $\frac{1}{2}\sqrt{2}nH_n + O(n) = 0.707106\ldots n \ln n + O(n)$.

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

We know from Lemma 4.3 that the average length of rises and falls of $\pi \in \mathfrak{S}_n$ is $\overline{L_n} = 2n/(n+1)$. Now, $\overline{L_n}$ also represents the average distance between two consecutive peaks and thus, the average distance between two consecutive peak-processors in the ring. Moreover, it follows from Lemma 4.5, that fortunately, every message within Stage 2* travels along the ring in a direction which is irrespective of the pairwise positions of peak-processors.

Therefore, the probability that a message is sent to the right or to the left is $\frac{1}{2}$ and we are brought back to the average-case analysis of Algorithm P. In this case, Algorithm P revisited, the asymptotic expected number of messages is

$$\tfrac{1}{2}\sqrt{2}n^* H_{n^*} + \tfrac{1}{2}\sqrt{2}n^* \overline{L_n} H_n + O(n),$$

when accumulating the average distance $\overline{L_n}$ for all the n^* peak-processors. And since

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

the asymptotic expected number of messages propagated in Algorithm D is

$$\tfrac{1}{2}\sqrt{2}\tfrac{1}{3}(3n+1)H_n + O(n) = \tfrac{1}{2}\sqrt{2}nH_n + O(n) \qquad \square$$

In the case when $s < n$ starters/initiators start Algorithm P and Algorithm D, the previous results remain basically valid; namely, the expected number of messages propagated in both algorithms is then $\frac{1}{2}\sqrt{2}nH_s + O(n)$.

Note however that we assumed (Section 3) that all the processors start the election "simultaneously" and work synchronously. The first assumption allows us not to consider the case when there exists $s < n$ 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; $2n$ and $2n+1$ pips on average, respectively ("pip" is the elementary delay time defined in the Introduction). 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, note that Bodlaender and van Leeuwen proved in [3] that the maximum number of messages is $\frac{1}{2}n^2$ and $\frac{1}{4}n^2$, respectively.

5. Experimental tests

In the following tests, the basic theoretical number of messages which is considered is the expected message complexity, $\frac{1}{2}\sqrt{2}n \ln n$ of Algorithm P. The number of messages computed by the sequent machine (a 12 processors PRAM) is the total number of messages propagated during the processing simulations of Algorithm D minus $3n$. The implemented algorithm is actually quasi-synchronous, since the minimum and maximum message delay times range between 1 and 2, respectively (some tests performed with message transmission delays ranging from 1 to 100 do not apparently alter the preceding results). All the n processors which participate in the election have distinct (pseudo-random) identification numbers, randomly drawn from a 9-digits decimal generator.

In the 70 experimental tests performed (with 1000 to 50 000 processors), the ratio $\overline{N'}/\overline{N}$ of the average number of simulated messages to the theoretical average value

$\overline{N}$, only varies from 0.945 to 0.994. Moreover, the ratios are all sharply close to 1 (up to 0.6% at best), and the mean ratio of 0.9725... differs less than 2.75% from 1. However, all ratios remain strictly smaller than 1.

The remarks which arise from this sample of tests, though it is certainly too limited an experiment, are twofold. First, the tight experimental values obtained seem to confirm the equality $\overline{N} = \overline{N'}$. Secondly, the ratios distribute too systematically below 1.

Yet, a conclusion may be drawn from these limited simulations and the apparent contradiction arising from the latter remarks. Indeed, we know that the simulations performed do not bring enough control over the factor $O(n)$ (in spite of the subtraction of $3n$ from the enumeration), and that the variations in the message transmission delays are not really taken into account in the above tests. This, together with the help of the very full and accurate empirical analysis completed by Mattern in [15], actually enables us to explain the fact that the ratios so systematically distribute below 1: Mattern's experimental results show that the asymptotic value $\frac{1}{2}\sqrt{2}$ of the coefficient of nH_n is approximated very slowly.

Thus, both the simulation results in Table 3 and Mattern's own empirical analysis of Algorithm P and Algorithm D confirm very strongly the mathematical analysis of the present paper. Therefore, we may conclude that the experimental analyses in [15] and the simulation tests results in Table 3 are in good agreement with the theoretical result obtained in Theorems 3.3 and Theorem 4.7.

Table 3
Average number of messages $\overline{N'}$ used in Algorithm D, and ratio $\overline{N'}/\overline{N}$

Number n of processors	$\overline{N} = \frac{1}{2}\sqrt{2}nH_n + O(n)$	$\overline{N'}$ = average number for 70 tests	Ratio $\overline{N'}/\overline{N}$[a]
20 000	140 056.46	137 290.687	0.98
30 000	218 685.9	218 355.758	0.966
40 000	299 718.09	288 955.7	0.964
50 000	382 536.929	378 747.1 (for 50 tests)	0.994
1000	4884.5	4733.2	0.945
5000	30 112.825	29 735.5	0.985
10 000	65 126.94	6374	0.977

[a] The mean ratio, 0.9725..., differs about 2.75% from 1.

6. 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 extrema-finding problem on a bidirectional ring, while requiring nearly the same amount of "time". Besides, the simulation results obtained in the experimental tests show good agreement with the asymptotic constant's value $\frac{1}{2}\sqrt{2}$.

Furthermore, this result is a confirmation of the fact (already proved in [3]) that distributed leader finding can be solved more efficiently on bidirectional rings than on 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 cope with most average-case analyses of distributed algorithms and distributed data structures. However, the bidirectional variants of distributed election algorithms seem surprisingly harder to analyse on 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 [15].

In [2], Bodlaender has shown that any bidirectional leader finding algorithm on rings (with sense of direction, n unknown) requires at least $\frac{1}{2}nH_n + O(n)$ on average (see Section 1). The gap between the coefficients $\frac{1}{2}$ and $\frac{1}{2}\sqrt{2}$ raises the question whether the bidirectional variant of the Chang-Roberts algorithm is average-case optimal. This result also raises the question of determining the average message complexity of other bidirectional distributed election algorithms, and possibly finding average-case optimal election algorithms (with respect to Bodlaender's lowerbound of $\frac{1}{2}nH_n$).

Acknowledgment

This work could not have been completed without Philippe Flajolet's first impulse and general contribution; it benefited also from discussions with François Lassner (especially for the proof of Lemma 4.5).

References

[1] J. Bienaymé, Sur une question de probabilités, *Bull. Soc. Math. France* **2** (1874) 153-154.

[2] H.L. Bodlaender, A better lower bound for distributed leader finding in bidirectional asynchronous rings of processors, *Inform. Process. Lett.* **27** (1988) 287-290.

[3] H.L. Bodlaender and J. van Leeuwen, New upperbounds for decentralized extrema-finding in a ring of processors, *STACS '86*, Lecture Notes in Computer Science (Springer, Berlin, 1986) 119-129.

[4] L. Comtet, *Analyse Combinatoire*, 2 Tomes (Presses Universitaires de France, 1970).

[5] W. Feller, *An Introduction to Probability Theory and its Applications, Vol. 1* (Wiley, New York, 1968).

[6] P. Flajolet, Personal communication, 1987.

[7] P. Flajolet and J.S. Vitter, Average-case analysis of algorithms and data structures, INRIA Res. Rep. 718, August 1987.

[8] D. Foata and M.P. Schützenberger, *Théorie Géométrique des Polynômes Eulériens*, Lecture Notes in Mathematics **138** (Springer, Berlin, 1970).

[9] J. Françon and G. Viennot, Permutations selon les pics, creux, double montées, double descentes, nombre d'Euler et de Genocchi, *Discrete Math.* **28** (1979) 21-35.

[10] I.P. Goulden and D.M. Jackson, *Combinatorial Enumeration* (Wiley, New York, 1983).

[11] D. Knuth, *The Art of Computer Programming, Vol. 3* (Addison-Wesley, Reading, MA, 1973).

[12] E. Korach, D. Rotem and N. Santoro, A probabilistic algorithm for decentralized extrema-finding in a circular configuration of processors, Res. Rep. CS. 81-19, Department of Computer Science, University of Waterloo, 1981.

[13] C. Lavault, Algorithmique et complexité distribuées, Thèse d'Etat, Univ. Paris XI-Orsay, December 1987.

[14] C. Lavault, Average number of messages for distributed leader finding in rings of processors, *Inform. Process. Lett.* **30** (1989) 167-176.

[15] F. Mattern, Message complexity of simple ring-based election algorithms—an empirical analysis, Draft version of Report SFB 124-36/88 (21.9.88), Department of Computer Science, University of Kaiserslautern, 1988.

[16] J. van Leeuwen and R.B. Tan, An improved upperbound for distributed election in bidirectional rings of processors, *Distributed Comput.* **2** (1987) 149-160.

[17] A. Rényi, Egy megfigyeléssorozat kiemelkedö elemeiröl, MTA III. *Oszt. Közl.* **12** (1962) 105-121.

[18] D. Rotem, E. Korach and N. Santoro, Analysis of a distributed algorithm for extrema-finding in a ring, *J. Parallel and Distributed Comput.* **4** (1987) 575-591.

[19] J. Vuillemin, A unifying look at data structures, *Comm. ACM* **23** (1980) 229-239.