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

C. Lavault

INRIA

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 840

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

Christian LAVAULT

AVRIL 1988

★RR 840★

DRAFT - march 1988

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

by

*Christian LAVAULT**

Abstract

Consider a distributed system of n processors arranged in a ring. All processors are labeled with distinct identity-numbers, but are otherwise identical. In this paper, we prove the exact expression of the average message complexity for the problem of determining a leader in asynchronous bidirectional rings of processors.

To achieve the results, we make use of combinatorial enumeration methods in permutations and derive an exact asymptotic evaluation of the average number of messages in both probabilistic and deterministic versions of the distributed election algorithm of Korach et al. [KOR,ROT,SAN-81], [ROT,KOR,SAN-87]. This settles the result of [BOD,VLE-86] that distributed leader-finding (election) is indeed strictly more efficient in bidirectional rings of processors than in unidirectional ones.

---

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

COMPLEXITE EN MESSAGES DE L'ELECTION SUR DES SYSTEMES DISTRIBUES EN ANNEAU VIRTUEL

*Christian LAVAULT**

Résumé.

Soit un système distribué dont le réseau d'interconnexion est configuré en anneau virtuel de taille n. Tous les processus y sont identiques, mais possèdent des identités distinctes. Dans cet article, nous calculons exactement la complexité moyenne en messages du problème de l'élection sur un réseau asynchrone de processus configuré en anneau virtuel bidirectionnel.

Pour ce faire, nous utilisons des méthodes typiques de la combinatoire des permutations et de l'analyse asymptotique. Nous pouvons ainsi calculer la valeur asymptotique exacte du nombre moyen de messages utilisés par chacune des deux versions, probabiliste et déterministe, de l'algorithme de Korach et al. [KOR,ROT,SAN-81], [ROT,KOR,SAN-87]. Cette démonstration établit donc, en confirmant la preuve obtenue par encadrement de [BOD,VLE-86], que, pour le problème de l'élection, le réseau en anneau virtuel bidirectionnel est strictement plus performant que son équivalent unidirectionnel.

---

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

Université Paris Val de Marne (Paris XII).

1. Introduction

In the sequel, we consider the problem of finding a leader in an asynchronous bidirectional ring of processors. Each site (processsor) 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 (eg the largest numbered one) in using a minimum number of messages.

Note that the problem is equivalent (up to $O(n)$ extra messages) to the problem of determining the largest processor's identity in the ring.

We assume that the processors work fully asynchronous 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 system to be error-free.

As for 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. We will as well denote message an *elementary message*, which is a message between two neighbour-sites in the ring. At last, we denote *pip* the traversal delay of one elementary message (such a traversal time delay in the ring may be measured for example by use of Lamport's method of timestamps).

Much work has already been completed to obtain good upper-, lower- and average bounds for different variants of the problem. An overview of worst-case and average case is given in Table 1 and 2 (see page 2), for unidirectional and bidirectional rings' algorithms respectively.

In [BOD,VLE-86], upperbounds for distributed leader-finding in bidirectional rings are proposed. These bounds are exhibited for the probabilistic algorithm given in [KOR,ROT,SAN-81] and [ROT,KOR,SAN-87] (viz *ALGORITHM-P*) and for a derived deterministic version of the latter (viz *ALGORITHM-D*).

In this paper, we derive the asymptotic average number of messages required both in *ALGORITHM-P* and *ALGORITHM-D*. This value is obtained by use of permutations theory considerations (inversion tables mainly) and combinatorial enumeration arguments involving an asymptotic analysis based on Stirling's formula and Euler-Maclaurin summation formula as well as average analysis by means of various permutations generating functions (eg generating function of Eulerian numbers). Such combinatorial and analytic methods seem powerful and general enough to provide efficient tools and cope with most average-case analysis of distributed algorithms and distributed data structures. Each *ALGORITHM-P* and *ALGORITHM-D* is thus proved to yield the same asymptotic average message complexity of $1/\sqrt{2}\, nH_n + O(n)$.

Simulation tests in a bidirectional ring with size ranging from 1000 up to 50,000 processors (each simulation being composed of 70 tests) may be considered to be in agreement with this value.

The result then leads to a positive answer to the question (first posed by Pachl, Korach & Rotem) of whether distributed leader-finding can be solved more efficiently in bidirectional rings than in unidirectional rings by a deterministic algorithm, thanks to an *exact* asymptotic estimation of the average number of propagated messages.

Algorithm	Average	Worst Case
Le Lann (1977)	n^2	n^2
Chang & Roberts (1979)	nH_n	$\frac{1}{2}n^2$
Peterson (1982)	0.943... nlnn*	1.44...nlgn
Dolev, Klawe & Rodeh (1982)	0.967... nlnn*	1.356...nlgn

Table 1 : Election Algorithms for Unidirectional Ring

*) Result experimentally obtained by Everhardt [BOD-87].

Algorithm	Average	Worst Case
Gallager et al. (1979)		5nlgn
Franklin (1982)		2nlgn
Korach, Rotem & Santoro (1981) *ALGORITHM-P*	[prob.] $\leq \frac{3}{4}nH_n$	[prob.] $\frac{1}{2}n^2$
Korach, Rotem & Santoro (1984)		1.89...nlgn
Bodlaender & van Leeuwen (1986) *ALGORITHM-D*	[det.] $\leq \frac{3}{4}nH_n$	[det.] $\frac{1}{4}n^2$
Van Leeuwen & Tan (1987)		1.44...nlgn
This paper (1988)	[prob.] $1/\sqrt{2}\,nH_n$	[prob.] $\frac{1}{2}n^2$
This paper (1988)	[det.] $1/\sqrt{2}\,nH_n$	[det.] $\frac{1}{4}n^2$
This paper (1988)	$(0.972...)/\sqrt{2}\,nH_n$*	

Table 2 : Election Algorithms for Bidirectional Rings

*) Result experimentally obtained (Section 5 of this paper).

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

For most of the quoted bidirectional algorithms, the assumption of a global sense of direction (ie each processor knows the left and right direction on the ring) is unnecessary.

2. Inversion Tables (see J. Vuillemin [VUI-80])

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

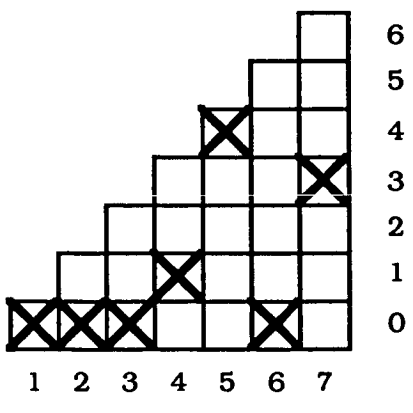
LEMMA 1. *The left-to-right maxima (upper records) of* $\pi \in \mathfrak{S}_n$ *correspond to occurrences of value zero in the inversion tables of size n.*

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

2 3 6 5 1 **7** 4, and the inversion table of π is t such that

t = 0 0 0 1 4 0 3.

An inversion table can thus be pictured as a "staircase" :



Put a cross in each column, the upper records (left-to-right maxima) are corresponding to the lower crosses (the zeros).

Now if one uses variables to denote the values 0,1, ... ,n-1 in the inversion table t (eg $x_1, x_2, \ldots, x_{n-1}$), then the set of all inversion tables, corresponding to all permutations $\pi \in \mathfrak{S}_n$, is fully described by the polynomial

$$x_0(x_0+x_1)(x_0+x_1+x_2)\ \cdots\ (x_0+x_1+\cdots+x_{n-1}). \qquad (1)$$

1st column 2nd column 3rd column (n-1)st column

LEMMA 2. *The average number of zero elements in an inversion table of size n is* $\sum_{1 \le i \le n} 1/i = H_n$, *the* n^{th} *harmonic number. Besides, the number of inversion tables of size n having k zeros is the classical unsigned Stirling number of first kind.*

PROOF - See [VUI-80] , where the unsigned Stirling number of first kind $s_{n,k}$ [COM-70] is proved to count at the same time :

(i) permutations $\pi \in \mathfrak{S}_n$ with k upper records,

(ii) permutations $\pi \in \mathfrak{S}_n$ with k cycles, and

(iii) inversion tables of size n such that $| \{ i/\ 1 \le i \le n, t_i = 0 \} | = k$. □

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)". And, in the present context of permutations counting, the enumerating polynomial of $s_{n,k}$ is

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

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

Therefore, an immediate consequence of Lemma 1. and 2. is

COROLLARY 1. *The average number of left-to-right maxima (upper records) of a permutation* $\pi \in \mathfrak{S}_n$ *is* H_n.

LEMMA 3. *Let* $\pi \in \mathfrak{S}_n$ *and assume that the* σ_i *are equally likely in* π. *The average distance to the first left-to-right maximum (upper record) of* π *is* $H_n - 1$.

PROOF - The average distance to the first upper record of $\pi \in \mathfrak{S}_n$ can be obtained by two different means. On the one hand, it represents nothing indeed but the average number of left-to-right maxima of π whenever its first element σ_1 is fixed as an upper record. Hence, the average distance to the first upper record of π which is different from σ_1 is the average number of occurrences of value zero in the inversion tables of size n, *minus one* : viz $H_n - 1$, since the first upper record of π (σ_1) is not counted in the enumeration.

On the other hand, the average distance to the first upper record in π may be directly derived as the solution of the following recurrent equation, where D_n denotes the distance to the first upper record of π.

$$\begin{cases} D_n = D_{n-1} + (n-1)\, D_{n-1} + (n-1)! \\ D_1 = 0 \end{cases}$$

so

$$\frac{D_n}{n!} = \frac{D_{n-1}}{(n-1)!} + \frac{1}{n} \qquad D_1 = 0$$

Whence the result,

$$\overline{D_n} = \frac{D_n}{n!} = \sum_{i=2}^{n} \frac{1}{i} = H_n - 1 \qquad (2)$$

□

PROPOSITION 1. *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*

$$\Pi_n(j) = [x^{j-2}] \; \frac{1}{\beta(\alpha-1)(\beta-1)} \left(1 + \frac{x}{\alpha}\right)\left(1 + \frac{x}{\alpha+1}\right) \ldots \left(1 + \frac{x}{\beta-2}\right)$$

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 : $G_n(x)$ is conditioned over the values α and β.

In the first place, let us 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 write down all the *monomials*, each corresponding to the possible upper records of π.

Positions 1 2 3 · · · $(\alpha-1)$ α $(\alpha+1)$ · · · · · · $(\beta-1)$ β $(\beta+1)$ · · · n

Monomials 1 1 2 · · · $(\alpha-2)$ x $(x+\alpha)$ $(x+\alpha+1)$ · · · $(x+\beta-2)$ x $(\beta+1)$ · · · 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,

$$G_n(x) = \frac{1.1.2. \ldots (\alpha-2).x.(x+\alpha).(x+\alpha+1) \quad \ldots \quad (x+\beta-2).\,x. \quad (\beta+1). \ldots n}{1.2.3. \ldots (\alpha-1).\alpha.(\alpha+1) \quad \ldots \quad (\beta-2)\,.\,(\beta-1).\,\beta. \quad (\beta+1). \ldots n}\,, \text{ and}$$

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

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

$$G_n(x) = \sum_{j\geq 2} \Pi_n(j)\, x^j = x^2 \sum_{j\geq 2} \Pi_n(j)\, x^{j-2} \qquad (3)$$

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 :

$$\Pi_n(j) = \frac{1}{\beta(\alpha-1)(\beta-1)} [x^{j-2}]\left(1+\frac{x}{\alpha}\right)\left(1+\frac{x}{\alpha+1}\right) \ldots \left(1+\frac{x}{\beta-2}\right) \qquad (4)$$

□

3. Analysis of *ALGORITHM-P*

We know from [KOR,ROT,SAN-81], [BOD,VLE-86] and [ROR,KOR,SAN-87] that *ALGORITHM-P* requires an expected number of messages of at most $\frac{3}{4} n H_n + O(n)$. This value is

only an upperbound because of possible effects of higher order upper records which remain ignored in this evaluation.

ALGORITHM-P [BOD,VLE-86]
Each processor P_i keeps the largest identity (identification number) it has seen in a local variable MAX_i ($1 \le i \le n$).
Each processor P_i goes through the following stages.
Stage 1. (initialization)
$MAX_i := \sigma_i$; choose a direction $d \in \{left, right\}$ with probability $\frac{1}{2}$; send message $<\sigma_i>$ 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 <!> 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 $<\sigma_j>$ is received from a neighbour, then
if $\sigma_j > MAX_i$ **then** $MAX_i := \sigma_j$; pass message $<\sigma_j>$ on
else if $\sigma_j = MAX_i$ **then** send message <!> on the ring
/* P_i has won the election */
fi;
Stage 3. (inauguration)
if a message <!> 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 <!> 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 detailed correctness of the algorithm is proved in [BOD,VLE-86], or [VLE,TAN-87]. In the sequel, we assume that all processors start the election simultaneously. For the analysis, we will also assume that the processors work synchronously.

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, $\pi = (\sigma_1\ \sigma_2 \cdots \sigma_n)$ 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 $<\sigma_1>$ is sent to the right or to the left with probability $\frac{1}{2}$. Thus, the expected number of elements in π visited by $<\sigma_1>$ is

$\frac{1}{2}(H_n - 1)$ whenever P_1 sends its message $<\sigma_1>$ to the right,

and $\frac{1}{2}\lfloor \frac{1}{2} H_n \rfloor$ whenever P_1 sends its message $<\sigma_1>$ to the left, since from (2), $H_n - 1$ is the average distance to the first left-to-right maximum in π. Accumulating the sum of these two quantities for all $<\sigma_i>$-messages ($1 \le i \le n$), which are independant random variables, yields the known upperbound of $\frac{3}{4} n H_n + O(n)$ 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.

PROPOSITION 2. *The average number of* $<\sigma_1>$*-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 (3).*

PROOF - Let $v_1, v_2, \ldots$ be random variables denoting the position of the first, second, and higher order upper records. We may consider the v_i as independant r.v. conditioned over the values α and β, and suppose $<\sigma_1>$ 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 $<\sigma_1>$-message is annihilated by the $<\sigma_\beta>$-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 $<\sigma_1>$-message is simply annihilated at P_α. Hence, let π denote the permutation $(\sigma_1 \cdots \sigma_n) \in \mathfrak{S}_n$, the number of positions in π visited by message $<\sigma_1>$ is exactly $\alpha - (1+\lfloor\frac{1}{2}\beta\rfloor)$, with $\frac{1}{2}\beta<\alpha<\beta<2\alpha$. Recall now formula (3) and the definition of probability $\Pi_n(j)$ and let us examine for the effect of *all* upper records of π. The average number of $<\sigma_1>$-messages propagated by *ALGORITHM-P* is

$$\overline{N}_1 = \frac{1}{2}(H_n - 1) + \frac{1}{2}\lfloor\frac{1}{2}H_n\rfloor - \sum_{j\geq 2} 2^{-j} \sum_{\alpha,\beta} (\alpha - \lfloor\tfrac{1}{2}\beta\rfloor - 1)\, \Pi_n(j) \qquad (5)$$

and, since

$$\begin{aligned}\sum_{j\geq 2} 2^{-j}\Pi_n(j) &= \sum_{j\geq 2} 2^{-j}\{\,[x^{j-2}]G_n(x)\,\} \\ &= \frac{1}{4}\sum_{j\geq 0} 2^{-j}\{\,[x^j]G_n(x)\,\} \\ &= \frac{1}{4}\,G_n(\tfrac{1}{2})\ ,\end{aligned}$$

$$\overline{N}_1 = \frac{3}{4}H_n - \frac{1}{4}\sum_{\alpha,\beta}(\alpha - \tfrac{1}{2}\beta)G_n(\tfrac{1}{2}) + O(1) \qquad (\tfrac{1}{2}\beta<\alpha<\beta<2\alpha) \qquad (6)$$

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 $f(\frac{1}{2}) = \sum_{j\geq 0} f_j 2^{-j}$ if $f(x) = \sum_{j\geq 0} f_j x^j$. The coefficient $\frac{1}{4}$ in (6) comes precisely from the fact that we have here $[x^{j-2}]G_n(x)$ instead of $[x^j]G_n(x)$.

COROLLARY 2. *The average number of messages required by ALGORITHM-P is exactly* $\frac{3}{4}nH_n - nS_n + O(n)$, *where*

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

PROOF- Accumulating in (6) the quantity N_1 for all the n $<\sigma_1>$-messages $(1\leq i\leq n)$ which are independant r.v. yields the exact average number of messages, namely

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

Now,

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

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

$$\overline{N} = \frac{3}{4}nH_n - n\sum_{\beta}\sum_{\alpha} \frac{1}{4}\frac{(\alpha-\beta/2)}{\beta(\alpha-1)(\beta-1)}\left(1+\frac{1}{2\alpha}\right)\ldots\left(1+\frac{1}{2\beta-4}\right) + O(n) \qquad (7)$$

Whence from (7) the exact average number of messages :

$$\frac{3}{4}nH_n - nS_n + O(n)$$ □

The remaining calculations concern the asymptotic behaviour of S_n.

3.2. Asymptotic analysis of S_n [FLA-87]

The following asymptotic analysis of S_n makes use of Stirling's formula and Euler-Maclaurin (one-dimensional) summation formula.

For $p\in\mathbb{N}$, define $Q_p = (1+\frac{1}{2})(1+\frac{1}{4})\cdots(1+1/2p)$

$$= \frac{3.5.7. \ldots (2p+1)}{2.4.6. \ldots (2p)}$$

or

$$Q_p = \frac{(2p+1)}{2^{2p}(p!)}$$

and, by means of Stirling's formula, for large p we have

$$Q_p \sim \frac{2\sqrt{p}}{\sqrt{\pi}}$$

with an error term of $O(1)$. The sum S_n may then be rewritten with the Q_p's as follows :

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

$$= \frac{1}{4} \sum_{2 \le \beta \le n} \left\{ \sum_{\beta/2 < \alpha < \beta} \frac{Q_{\beta-1}}{Q_\alpha} \frac{(\alpha - \beta/2)}{\beta(\alpha-1)(\beta-1)} \right\} \qquad (8)$$

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

$$u(\beta) \sim \sum_{\alpha,\beta} \frac{\alpha^{-\frac{\beta}{2}}}{\alpha\beta^2} \sqrt{\frac{\beta}{\alpha}} \qquad \left(\frac{\beta}{2} < \alpha < \beta\right)$$

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

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

which is uniform in β. And at last

$$u(\beta) \sim \frac{1}{\beta} \left(3 - 2\sqrt{2} + O\left(\frac{1}{\beta}\right)\right) \qquad (9)$$

which yields the

THEOREM 1. *The asymptotic expected number of messages used in ALGORITHM-P is* $1/\sqrt{2}\, nH_n + O(n) = 0.707106\ldots n \ln n + O(n)$ ($\sim 0.49\ldots n \lg n$).

PROOF - From (7),

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

The exact computation of S_n from (8) and (9) gives rise to

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

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

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

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 [BOD,VLE-86] in which Stage 1 is replaced by a fully deterministic stage. The idea is to let each processor P_i send its $<{}^*\sigma_i>$-message in the direction of the smallest neighbour and thus get rid of all the smaller neighbours from scratch.

ALGORITHM-D
Similar to *ALGORITHM-P* except that for each processor P_i, Stages 1 and 2 are replaced as follows :
Stage 1*.
send message $<{}^*\sigma_i>$ to both neighbours on the ring ; wait for the message $<{}^*\sigma_{i-1}>$ and $<{}^*\sigma_{i+1}>$ 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 $<{}^*\sigma_i>$ to the left
    else send message $<{}^*\sigma_i>$ to the right
  fi
fi;
Stage 2*. (election)
repeat the following steps, until the end of the election is signaled by receipt of a <!> 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 $<{}^*\sigma_j>$ is received from a neighbour then
if ${}^*\sigma_j > MAX_i$ **then** $MAX_i := {}^*\sigma_j$; pass message $<{}^*\sigma_j>$ on
  else if ${}^*\sigma_j = MAX_i$ **and** ${}^*\sigma_i = {}^*\sigma_j$ **then** send message <!> on the ring
    /* P_i has won the election */
  else if ${}^*\sigma_j = MAX_i$ **and** ${}^*\sigma_i \neq {}^*\sigma_j$ **then** pass message $<{}^*\sigma_j>$ 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 $\lceil \frac{1}{2}n \rceil$ 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 $<*\sigma_j>$-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 estimation of the average number of messages sent by the active processors that pass on to Stage 2* in *ALGORITHM-D*, we need 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 lenth of the rises and falls of π.

LEMMA 4. *The expected number of peaks of* $\pi \in \mathfrak{S}_n$ *is* $\overline{P}_n = \frac{1}{3}(n+1)$.

PROOF - To derive $\overline{P}_n$, we use combinatorial enumeration methods developped by Philippe Flajolet (see [FLA,VIT-87] for example). $\mathcal{T}$ denotes the (binary) tournament tree associated to the permutation π, T_g and T_d denotes 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

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

Thus, the ordinary generating function of the expectation of λ is

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

which leads to the first order differential equation :

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

whith solution

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

and, since

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

(OGF of $\Lambda(z)$ obtained from the above solution),

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

The expected number of peaks of π is then $\overline{P}_n = \frac{1}{3}(n+1)$. The central limit theorem applies, but the variance being of order n^3 (and thus the standard deviation of order $n^{3/2}$), it yields in the case an expression which is irrelevant to our purpose). □

Note that the average number of valleys of π, $\overline{V}_n$, 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 $\frac{1}{3}(2n-1)$.

LEMMA 5. *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 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

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

has the value ([COM-70], p.63, T.1) :

$$U(t,z) = \frac{1-t}{\exp(z(t-1)) - 1}$$

Whence, developping (10) with regard to (t-1) :

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

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

$$[z^n]U(t,z)\Big|_{t=1} = \sum_k A(n,k) = n! \qquad (11)$$

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

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

The mean $\overline{R}_n = {}^{(12)}/_{(11)} + 1 = \frac{1}{2}(n+1)$, and the variance $\mathrm{var}(R_n) = {}^{(n+1)}/_{12}$ are easily derived from the above identities. In the case, the standard deviation is 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 6. *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* correspondance (due to J. Françon and G. Viennot [FRA,VIE-79] [GOU,JAC-83]) 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 that the initial and final elements of a permutation $\pi \in \mathfrak{S}_n$ are recognized by notionally placing a zero at both ends of π ; whereas in a circular configuration of processors, we also assume that of course the first and last elements of a permutation π are the same : see example in page 17).

Besides, define the set E_n of "subexceeding functions" on [1,n] to be the set of functions f on [1,n] such that $f(i) \le i$ 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 correspondance between the crossed squares of the inversion table of a permutation (viz its associated "staircase") and the corresponding value f(i)-1 (for all $1 \le i \le n$), f being then a subexeeding function.

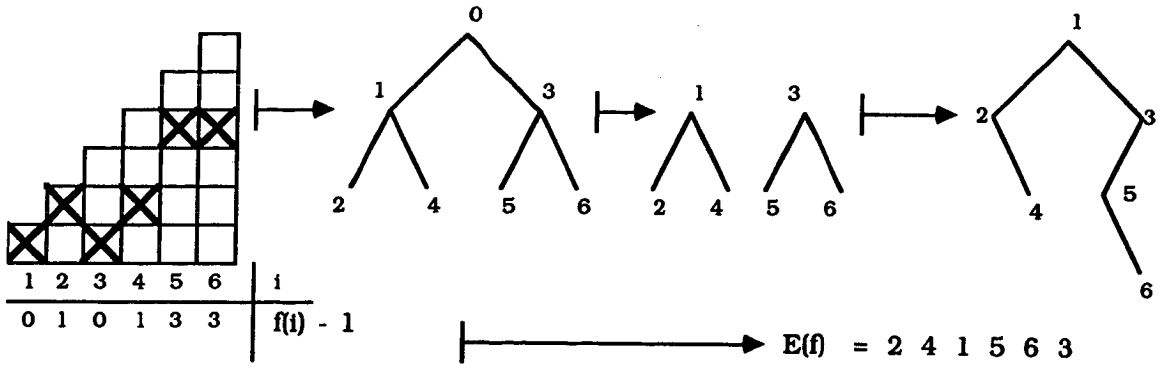


Fig.1. The one to one correspondance between $\mathfrak{S}_n$ and E_n.

E_n may be described by means of the formal polynomial (non-commutative) :

$F(x_1, \ldots, x_n) = x_1(x_1+x_2) \cdots (x_1+x_2+\cdots+x_n)$, 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, provided that the constant term in $F_j(t)$ has value zero.

Let

$$\Phi(t) = \sum_{j=0}^{n-1} \{ j!\,(t+j) \ldots (t+n-1) \} - n!\,\frac{(n-1)}{2} \qquad (13)$$

be this generating polynomial.

From (13), $\Phi(1) = n.n! - n!\,\frac{1}{2}(n-1) = \frac{1}{2}(n+1)n!$

and

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

Whence

$$\overline{L_n} = \frac{\Phi'(1)}{\Phi(1)} = \frac{2n}{n+1}$$

the expected length of rises and falls of π. □

Note that the variance is

$$\frac{2n(n+3)}{(n+1)^2} - \frac{4}{n+1}H_n,$$

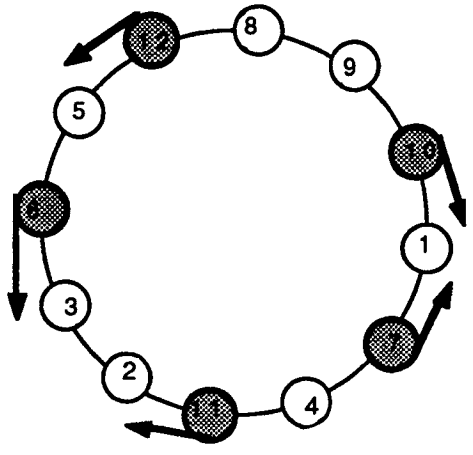
and the central limit theorem shows that L_n, when normalized, converges to the normal distribution.

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), and each of them is at least one position apart. Within Stage 2*, we may claim the independance of their choice of direction in sending their messages around the ring (left or right), with respect to one another. As a matter of fact, the direction where each peak-processor sends its $<^*\sigma_j>$-message ($j \in J$, with $|J| = \frac{1}{3}(n+1)$) is *absolutely independant of its respective position* (regarded as the position of a peak of π) *with respect to the position of any other peak-processor.*

Two consecutive peak-processors may actually send their respective messages in the same direction *or* in opposite directions in the ring, this according only to the position of their smaller neighbour.

As an example, we may consider the permutation π of $\mathfrak{S}_{12}$ with (bold) peaks **10**, **7**, **11**, **6** and **12**: π = (8 9 **10** 1 **7** 4 **11** 2 3 **6** 5 **12**).



The choice of direction of P_{10}, P_7, P_{11}, P_6 and P_{12} will only depend on the respective identities of their (immediate) right and left neighbours : P_{10} sends its $<^*10>$-message clockwise (towards P_1) and so does P_7 (towards P_2) ; but P_7, P_6 and P_{12} will send their respective messages anticlockwise towards P_1, P_3 and P_5 (respectively).

messages anticlockwise towards P_1, P_3 and P_5 (respectively).

This relative independance shows itself 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 peaks which send their messages in the same direction in the ring for the first pair and in opposite directions for the elements of the second couple.

We know from Lemma 6 that the average length of rises and falls of π 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.

THEOREM 2. *The asymptotic expected number of messages used by ALGORITHM-D is* $1/\sqrt{2}\, nH_n + O(n) = 0.707106...\ n\ln n + O(n)$ ($\sim 0.49...\ n\lg n$).

PROOF - Denote $n^* = \frac{1}{3}(n+1)$ the average number of peak-processors which remain at the end of Stage 1* and $\overline{L}_n$ the average distance between two consecutive peak-processors. Within Stage 2*, every message travels along the ring in a direction which is not dependant of the respective 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 analysis of *ALGORITHM-P*.

In the case then (*ALGORITHM-P* revisited), the asymptotic expected number of messages is $1/\sqrt{2}\, n^* H_{n^*} + 1/\sqrt{2}\, n^* \overline{L}_n H_n + O(n)$, when accumulating the average distance $\overline{L}_n$ for all the n^* peak-processors.
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

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

Note that we assumed (Section 3) that all processors start the election "simultaneously" and work synchronously. The first assumption allows us not to consider the case when there exists $p<n$ 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 ; $2n$ and $2n+1$ pips on the average, respectively (A "pip" is a delay used in our terminology, see Introduction in

leader).

As to the worst-case message complexity of *ALGORITHM-P* and *ALGORITHM-D*, note that Bodlaender & Van Leeuwen proved that the maximum number of messages is $\frac{1}{2}n^2$ and $\frac{1}{4}n^2$ respectively [BOD,VLE-86].

5. Simulation tests

In the tests, the basic theorical number of messages is the expected complexity of *ALGORITHM-P*, $1/\sqrt{2}$ nln(n). The number of messages computed by the "SEQUENT" (a 12 processors PRAM machine) is the total number of propagated messages minus 3n. The implemented algorithm is quasi-synchronous, since the maximum and minimum delays of message transmission range between 1 and 2, respectively (some tests performed with transmission delay ranging between 1 and 100 do not apparently change the results). The n processors which participate in the election own pseudo- random identification numbers randomly drawn from a 9-digit decimal generator.

Number of processors in the bidirectional ring A_n : n =	$\overline{N} = \frac{n}{\sqrt{2}} H_n + O(n)$ ($\overline{N} = \frac{1}{\sqrt{2}}$ nlnn)	$\overline{N'}$ = average number for 70 tests (N' = # messages - 3n)	Ratio $\frac{\overline{N'}}{\overline{N}}$
20,000	140,056.464	137,290.6875	0.980
30,000	218,685.91	218,355.7584	0.96648
40,000	299,718.0911	288,955.7	0.964
50,000	382,536.929792	378,747.11 (for 50 tests)	0.994
1,000	4,884.52	4,733.2	0.945
5,000	30,112.825	29,735.5	0.985
10,000	65,126.9413	63,743	0.977 (mean ≃ 0,9725 i.e. different of about 2,8% from 1)

Table 3. Average number of messages, $\overline{N'}$, used by *ALGORITHM-D* (70 tests) ; and ratio $\overline{N'}/\overline{N}$, where $\overline{N} = 1/\sqrt{2}$ nln(n) + *O*(n) (*ALGORITHM-P*).

In the 70 simulation tests performed (with 1000 to 50,000 processors), the ratio $\overline{N'}/\overline{N}$ of the

In the 70 simulation tests performed (with 1000 to 50,000 processors), the ratio $\overline{N'}/\overline{N}$ of the average number of simulated messages to the basic theorical mean ($\overline{N} = {}^{1}/\sqrt{2}\ n\ln(n)$) varies from 0.945 to 0.994. Moreover, these ratios are all accurately close to 1 (up to 2.6% at most) whereas they still remain strictly lower than 1.

On the one hand then, the first fact is a strong argument for the equality of $\overline{N}$ and $\overline{N'}$, when on the other hand, the repartition of ratios is too systematically below 1. Yet, a conclusion may be drawn from the simulations : we have no real control on the factor $O(n)$ and the use (as the number of counted messages) of N' *minus* $3n$ might not be enough to stick to the real process of *ALGORITHM-D*. In a way which is similar to the experimental results obtained by Everhardt, these simulation tests do not enough take into account the small negative value contained in $O(n)$. As a consequence, we may conclude that these simulation tests are in agreement with the theorical result of a same asymptotic average message complexity for both algorithms.

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 while requiring nearly the same amount of "time". Simulation results showed good agreement with constant's value of ${}^{1}/\sqrt{2}$. These algorithms also do not require a global sense of orientation in the ring (ie clockwise does not have to be the same for all processors). It is still an interesting open question to find an *exact* (not only asymptotically exact) expression for the expected message complexity and the variance of *ALGORITHM-P* and *ALGORITHM-D*. This result is a confirmation of the tight bounds derived in [BOD,VLE-86] and settles the proof that distributed leader finding can be solved more efficiently in bidirectional rings than in unidirectional rings by a deterministic algorithm. in general, analysis on the average of bidirectional algorithms seem harder than unidirectional ones ; combinatorial enumeration and generating functions prove suitable and instrumental methods to tackle this kind of problem.

References

[BOD-86] **H.L. BODLAENDER**, Distributed Computing- Structure and Complexity, PHD. Thesis, 1986 (pp. 24-42).

[BOD,VLE-86] **H.L. BODLAENDER & J. Van LEEUWEN**, "New upperbounds for decentralized extrema-finding in a ring of processors", STACS 86, pp.119-129.

[COM-70] **L. COMTET**, Analyse combinatoire, Tome 1 and 2 (PUF, 1970).

[FLA-87] **P. FLAJOLET**, Private Communication, 1987.

[FLA,VIT-87] **P. FLAJOLET & J.S. VITTER**, "Average-case analysis of algorithms and Data Structures", R.R. n°718, INRIA, Aug. 1987.

[FRA,VIE-79] **J. FRANCON & G. VIENNOT**, "Permutations selon les pics, creux, double montées, double descentes, nombre d'Euler et de Genocchi", Discrete Math. 28, 1979.

[GOU,JAC-83] **I.P. GOULDEN & D.M. JACKSON**, Combinatorial enumeration, Wiley-1983.

[KNU-73] **D. KNUTH**, Sorting and Searching, Vol.3 (Addison Wesley, 1973).

[KOR,ROT,SAN-81] **E. KORACH, D. ROTEM & N. SANTORO**,"A probabilistic algorithm for decentralized extrema-finding in a circular configuration of processors", Res. Rep. CS. 81-19, Dep. of Comp. Science, Un. of Waterloo, 1981.

[LA-87] **C. LAVAULT**, Algorithmique et complexité distribuées,Thèse d'Etat (Un. Paris 11-Orsay), Dec. 1987.

[ROT,KOR,SAN-87] **D. ROTEM, E. KORACH & N. SANTORO**, "Analysis of a Distributed Algorithm for Extrema Finfing in a Ring", J. of Parallel and Distributed Computing 4, 1987, pp. 575-591.

[VLE,TAN-87] **J. Van LEEUWEN & R.B. TAN**, "An improved upperbound for distributed election in a bidirectional rings of processors", Distributed Computing 2, 1987, pp. 149-160.

[VUI-80] **J. VUILLEMIN**, "A Unifying look at Data structures", CACM Vol. 23, n°4, April 1980 pp.229-239.