NEW LOWER BOUND TECHNIQUES FOR DISTRIBUTED LEADER FINDING AND OTHER PROBLEMS ON RINGS OF PROCESSORS

Hans L. Bodlaender

RUU-CS-88-18
April 1988

Rijksuniversiteit Utrecht

Vakgroep informatica

Padualaan 14 3584 CH Utrecht
Corr. adres: Postbus 80.089, 3508 TB Utrecht
Telefoon 030-531454
The Netherlands

NEW LOWER BOUND TECHNIQUES FOR DISTRIBUTED LEADER FINDING AND OTHER PROBLEMS ON RINGS OF PROCESSORS

Hans L. Bodlaender

Technical Report RUU-CS-88-18
April 1988

Department of Computer Science
University of Utrecht
P.O. Box 80.089, 3508 TB Utrecht
The Netherlands

A former version of this paper appeared as "A new lowerbound technique for distributed extrema finding on rings of processors" , technical report RUU-CS-87-11, Dept. of Comp. Science, University of Utrecht. This version contains several new results.

New Lower Bound Techniques for Distributed Leader Finding and other Problems on Rings of Processors*

Hans L. Bodlaender†
Department of Computer Science, University of Utrecht
P.O.Box 80.089, 3508 TB Utrecht, the Netherlands

Abstract

Several new lower bounds are derived for deterministic and randomized extrema finding and some other problems on asynchronous, non-anonymous rings of processors, where the ring size n is known in advance to the processors.

With a new technique, using results from extremal graph theory, an $\Omega(n \log n)$ lower bound is obtained for the average number of messages for distributed leader finding, on rings where the processors know the ring size n, and processors take identities from a set I with size as small as cn, for any constant $c > 1$. Formerly, this bound was only known for special values of n, and exponential size of I. Also, improvements are made on the constant factor of the $\Omega(n \log n)$ bound.

An elementary, but powerful result shows that the same bounds hold for randomized algorithms. It is shown that $\Omega(n \log n)$ lower bounds can be derived for the expected message complexity for computing AND on an input 1^n, OR on an input 0^n or XOR over all inputs, even when processors have unique identities. This confirms a conjecture of Abrahamson et. al. [2].

1 Introduction.

Consider an asynchronous ring of processors. Each processor is distinguished by a unique identification number, taken from some index set I. In this paper we assume that the size n of the ring is known in advance to the processors. There is no central controller. We consider two (types of) problems(s). The first problem we consider is to design a distributed algorithm that "elects" a unique processor as leader (e.g. the highest numbered processor), using a minimum number of messages. The second type of problem is when each processor has, besides its identification number, an input bit, and some cyclic boolean function must be computed over the string of input-bits.

We assume that the processors work fully asynchronous and cannot use clocks or time-outs. Hence we can assume that the algorithm is message-driven: except for the first message upon initialization, a processor can only send messages as a result of the receipt

*Part of this research was done while the author was visiting the Laboratory of Computer Science of the Massachusetts Institute of Technology, with a grant from the Netherlands Organization for Scientific Research (N.W.O.).

†Electronic mail: mcvax!ruuinf!hansb.

of a message. We also assume that processors and the communication subsystem work error-free and that links work in a FIFO-manner.

There are basically two variants: the ring may be unidirectional (all messages go in one direction) or bidirectional (messages can go in both directions). For bidirectional algorithms, one has the variant where the ring has "a sense of direction", i.e. each processor has the same idea about "left" and "right", and the variant where processors do not have a sense of direction. We will assume the former case, which only strengthens the results.

Much work has been done to obtain good upper and lower bounds for the different variants of both types of problems.

In this paper, we concentrate on the leader finding problem. In section 8 we show what results evolve when the techniques are applied to some other problems.

In table 1, the best known upper bounds for the leader finding problem are summarized. None of these algorithm requires that processors know the ring size. (H_n is the n'th harmonic number, i.e. $H_n = \sum_{i=1}^{n} \frac{1}{i} \approx 0.69n \log n$).

	average	worst-case
Unidirectional	nH_n [10]	$1.356n \log n + \mathcal{O}(n)$ [11]
Bidirectional with sense of direction	$\frac{\sqrt{2}}{2}nH_n$ [7,13]	$1.356n \log n + \mathcal{O}(n)$ [11]
Bidirectional without sense of direction	$\frac{\sqrt{2}}{2}nH_n$ [7,13]	$1.44n \log n + \mathcal{O}(n)$ [16,17]

Table 1: Overview of upper bounds for leader finding problem.

Pachl, Korach and Rotem [20] obtained $\Omega(n \log n)$ lower bounds for the average and worst-case number of messages on unidirectional and bidirectional rings without known ring size, and the worst-case number of messages on rings with known ring size. Similar lower bounds, improving with a constant factor the results in [20], can be found in [5,6] and [14].

It has long been an open problem to determine the average number of messages on rings with a fixed ring size. Recently, Duris and Galil [12] obtained lower bounds of $(\frac{1}{4} - \varepsilon)n \log n - \mathcal{O}(n)$ for the average number of messages on unidirectional rings with fixed ring size, and $(\frac{1}{8} - \varepsilon)n \log n - \mathcal{O}(n)$ for the average number of messages on bidirectional rings with fixed ring size. Their proof assumes that n is a power of 2, and requires that the size of the index set I is exponential in n.

In [19] Pachl gives a lower bound for probabilistic unidirectional algorithms. We only deal with randomized algorithms (i.e. algorithms that succeed with probability 1, and can use randomization).

In this paper we prove $\Omega(n \log n)$ lower bounds for unidirectional and bidirectional rings with *any* fixed ring size n, where the size of the set of identities I may be as small as cn, for any constant $c > 1$. To be precise, for unidirectional rings we have a lower bound of $\frac{1}{2}(\frac{1}{c} - \frac{1}{c^2})n \log n - \mathcal{O}(n)$, for all n, and $(\frac{1}{c} - \frac{1}{c^2})n \log n - \mathcal{O}(n)$ for infinitely many n. If we allow that $|I| \geq n^2$, then we have lower bounds of $\frac{1}{4}n \log n$ and $\frac{1}{2}n \log n$, respectively. The respective coresponding lower bounds for bidirectional rings are $\frac{1}{4}(\frac{1}{c} - \frac{1}{c^2})nH_n - \mathcal{O}(n)$, $\frac{1}{4}(\frac{1}{c} - \frac{1}{c^2})n \log n - \mathcal{O}(n)$, $\frac{1}{8}nH_n - \mathcal{O}(n)$ and $(\frac{1}{4} - \varepsilon)nH_n$.

Note that if $|I| - n$ is very small, then one can design algorithms which use less than $\Omega(n \log n)$ messages. For example, one can turn all processors with an identity, which is one of the $n-1$ smallest in I "inactive", and then run a variant of Petersons $1.44n \log n + \mathcal{O}(n)$ unidirectional algorithm [21]. This gives an algorithm using $\mathcal{O}(n \log(|I| - n))$ messages (worst-case). (This observation was made by Gerard Tel.)

Also, we show, with an elementary but powerful result, that the same lower bounds hold for randomized algorithms.

Next, suppose that each processor has beside its unique identity an input-bit, and some cyclic boolean function (like AND, OR, XOR) of the sequence of input bits must be computed distributedly. Any such function, that is not constant has a worst-case bit complexity of $\Omega(n \log n)$ [18], even if processors have identities taken from a set I with $|I| \geq n^{1+\epsilon}$, for constant $\epsilon > o$ [8].

Abrahamson and al [2] define the expected bit complexity of a (randomized) algorithm to be the maximum over all inputs of the expected number of bits transmitted on an anonymous ring with that input. They show that every non-constant function has expected bit complexity $\Omega(n\sqrt{\log n})$ and give a function that matches this bound. They conjecture that OR and AND have expected bit complexity $\Theta(n \log n)$. We show that the conjecture holds, even if processors have identities. Also, the expected number of messages over all inputs of randomized algorithms computing XOR is shown to be $\Omega(n \log n)$. Note that there exist simple determinstic algorithms for AND and OR with the average bit complexity over all inputs $\mathcal{O}(n)$ [4].

This paper is organized as follows. In section 2 some definitions and preliminary results are given. In section 3 we consider leader finding on unidirectional rings with certain ring sizes. In section 4 the results are extended to arbitrary ring sizes, and in section 5 to bidirectional networks. In section 6 a negative result on randomized algorithms on non-anonymous networks is given, relating the average number of messages for deterministic algorithms, and the expected number of messages for randomized algorithms. In section 7 some other problems on rings of processors are considered.

2 Definitions and preliminary results.

For an index set I, define $D(I)$ to be the set of finite, non-empty sequences of distinct elements of I. The concatenation of two strings $s = s_1 \cdots s_k$ and $t = t_1 \cdots t_l$ is denoted by $s \cdot t = s_1 \cdots s_k t_1 \cdots t_l$. The $l'th$ element of a string s is denoted by s_l. The length of a string $s = s_1 \cdots s_k$ is denoted by length(s) $= k$. The set of finite, non empty sequences of distinct elements of I with length k is denoted by $D_k(I) = \{s \in D(I) \mid \text{length}(s) = k\}$.

For the sake of analysis, we assume a (clockwise) numbering of the processors $1, 2, \ldots, n$. (n is the size of the ring; the numbering is not known to the processors). We say a ring is labelled with $s = s_1 \ldots s_n \in D_n(I)$, if for each $i, 1 \leq i \leq n$, processor i has identity s_i.

Further we denote $X_k(I)$ to be the set of all sets of $\lfloor \frac{|I|}{k} \rfloor$ disjoint strings from $D_k(I)$, i.e. $X_k(I) = \{S \subseteq D_k(I) \mid |S| = \lfloor \frac{|I|}{k} \rfloor$ and $(\forall s, t \in S : s \neq t \Rightarrow \forall i, j \leq k : s_i \neq t_j))\}$.

For $k|n$, we say that a string $s \in D_n(I)$ is *derived* from $S \in X_k(I)$, if s is formed by concatenating $\frac{n}{k}$ different elements from S.

Next we review some results from extremal graph theory. The interested reader is referred to the book of Bollobás [9], for background, proofs, etc.

Define $\alpha(m, l)$ ($\bar{\alpha}(m, l)$) to be the maximum number of edges in a directed (undirected) graph with m vertices, that does not contain a cycle with length l, and let $\beta(m, l) = 1 - \frac{\alpha(m,l)}{m(m-1)}$.

Lemma 2.1
$\forall N, l, 3 \leq l \leq N : \alpha(N,l) \leq \bar{\alpha}(N,l) + \frac{1}{2}N(N-1)$.

Proof.
Let $G = (V,E)$ be a directed graph with $\bar{\alpha}(N,l) + \frac{1}{2}N(N-1) + 1$ edges. It follows that there are at least $\bar{\alpha}(N,l)+1$ pairs of nodes v, w with $(v,w) \in E$ and $(w,v) \in E$. Hence G contains a cycle with length l. □

Theorem 2.2 [3]
Let $G = (V,E)$ be an undirected graph with $|V| = N, |E| = M$, and let $l \in \mathbb{N}^+$ be a positive natural number, such that $l \geq \lfloor \frac{1}{2}(N+3) \rfloor$, and $M > \binom{l-1}{2} + \binom{n-l+2}{2}$. Then G contains a cycle with length r, for every $r, 3 \leq r \leq l$.

Corollary 2.3
Let $N > l > \frac{1}{2}N + 3$ and let $3 \leq r \leq l$. Then

(i) $\bar{\alpha}(N,r) \leq \binom{l-1}{2} + \binom{N-l+2}{2}$.

(ii) $\alpha(N,r) \leq \binom{l-1}{2} + \binom{N-l+2}{2} + \frac{1}{2}N(N-1)$.

(iii) $\beta(N,r) \geq \frac{l}{N} - \frac{l^2}{N^2} \pm \mathcal{O}(\frac{1}{N})$.

Proof.
(i), (ii) follow directly from theorem 2.2. and lemma 2.1.
(iii) can be derived as follows:

$$\begin{aligned}
\beta(N,l) &= 1 - \frac{\frac{1}{2}l(l-1) + \frac{1}{2}(N-l+2)(N-l+1)}{N(N-1)} - \frac{1}{2} \\
&= \frac{1}{2} - \frac{\frac{1}{2}N^2 + \frac{1}{2}l^2 - lN + 3N - 3l + 2 + \frac{1}{2}l^2 - \frac{1}{2}l}{N(N-1)} \\
&= \frac{1}{2} - \frac{\frac{1}{2}N(N-1) + \frac{1}{2}N + l^2 - lN + 3N + 3\frac{1}{2}l + 2}{N(N-1)} \\
&= \frac{lN}{N(N-1)} - \frac{l^2}{N(N-1)} + \mathcal{O}(\frac{1}{N}) \\
&= \frac{l}{N} - \frac{l^2}{N^2} + \mathcal{O}(\frac{1}{N}).
\end{aligned}$$

□

Theorem 2.4 [9]
Let $G = (V,E)$ be an undirected graph with $|V| = N, |E| = M$. Let k be a natural number and let $M > 90kN^{\frac{1+1}{k}}$. Then G contains a cycle of length $2l$ for every integer $l, k \leq l \leq kn^{\frac{1}{k}}$.

Corollary 2.5
Let l be even; $4 \leq l \leq 4\sqrt{N}$. Then

(i) $\bar{\alpha}(N,l) \leq 180N\sqrt{N}$.

(ii) $\alpha(N,l) \leq 180N\sqrt{N} + \frac{1}{2}N(N-1)$.

(iii) $\beta(N,l) \geq \frac{1}{2} - \frac{180\sqrt{N}}{N-1}$.

Proof.
Use Theorem 2.4 with $k = 2$. □

3 Lower bounds for leader finding on unidirectional rings with certain ring sizes.

In this section we consider leader finding on unidirectional rings, and assume that the ring size n is a power of 2. We prove a lower bound of $(\frac{1}{c} - \frac{1}{c^2})n\log n - \mathcal{O}(n)$ messages, for the average case on unidirectional rings, with index set I, with $|I| \geq cn, 1 < c \leq 2$. For $|I| \geq n^2$, we prove a lower bound of $\frac{1}{2}n\log n - \mathcal{O}(n)$ messages, which improves a lower bound of Duris and Galil [12].

For our analysis we first remark that as links operate in a FIFI-manner, the number of messages sent does not depend on the relative time it takes to send messages, in the unidirectional case. So we may as well assume that all processors start simultaneously at time 1, and each message takes unit time. As a consequence, it only depends on the id's of the $t-1$ processors, directly preceding a processor i, and its own id, whether or not processor i will send a message on time t. (This technique is very similar to techniques used in [20]).

Now consider some fixed ring size n and index set I. Let A be an asynchronous unidirectional leader finding algorithm for ring size n and index set I.

We may assume that after completion of the algorithm, every processor knows the identity of the leader. (Other variants differ in $\mathcal{O}(n)$ messages, at most.)

Lemma 3.1
For all $r \in D_n(I)$, and $t < n-1$ there is at least one processor that sends a message at time t, on a ring labelled with r, when executing A, if $|I| \geq n+1$.

Proof.
Suppose not. Suppose processor i becomes the leading processor. At time t, processor $i-1$ (or n, if $i = 1$) cannot distinguish the case that processor i has identity r_i, or processor i has an identity, not in r. Contradiction. □

For all $k \leq \frac{1}{2}n$, we now define for each $S \in X_k(I)$ the following directed graph $G(S) = (S, E(S))$, by $E(S) = \{(s,t) \mid s,t \in S$: processor $2k$ will not send a message between time $k+1$ and $2k$, on a ring labelled with $r \in D_n(I)$, with $s = r_1 \cdots r_k$ and $t = r_{k+1} \cdots r_{2k}\}$.

Lemma 3.2
Let $k|n, k < \frac{1}{2}n$; let $S \in X_k(I)$. Then $G(S)$ does not contain a cycle with length $\frac{n}{k}$.

Proof.
Suppose $G(S)$ contains a cycle with length $\frac{n}{k}$, and let $s^1, \ldots, s^{\frac{n}{k}}$ be the successive nodes on this cycle. Let $r = s^1 \cdot s^2 \cdot \ldots \cdot s^{\frac{n}{k}}$. Now consider an execution of A on a ring labelled with r. (Note that $r \in D_n(I)$.)

It follows from lemma 3.1. that there is at least one processor that sends a message at time $2k$. So suppose processor $(ik+j), 0 \leq j \leq k-1$ sends a message at time $2k$. Then processor ik (or processor n, if $i = 0$), sends a message between time $k+1$ and $2k$. It follows that $(s^{i-1}, s^i) = (r_{(i-2)k+1} \ldots r_{(i-1)k}, r_{(i-1)k+1} \ldots r_{ik}) \notin E(S)$. (Or, if $i = 1, (s^{\frac{n}{k}}, s^1) \notin E$). Contradiction. □

Theorem 3.3
Let $k|n$, $k < \frac{1}{2}n$. Then the number of messages, sent on a unidirectional ring with known ring size n between time $k+1$ and $2k$, averaged over all ring labellings $r \in D_n(I)$, is at least $\beta(\lfloor \frac{|I|}{k} \rfloor, \frac{n}{k}) \cdot n$.

Proof.
Consider some $S \in X_k(I)$. Since a non-edge in $G(S)$ corresponds to a message, sent by processor $2k$, between time $k+1$ and $2k$ the average number of messages sent by processor $2k$ between time $k+1$ and $2k$, over all rings, labelled with $r \in D_n(I)$, which is derived from S, is at least $\beta(\lfloor \frac{|I|}{k} \rfloor, \frac{n}{k})$. Note that each $r \in D_n(I)$ is derived from the same number of $S \in X_k(I)$. It follows that the average number of messages, sent by processor $2k$ between time $k+1$ and $2k$ is at least $\beta(\lfloor \frac{|I|}{k} \rfloor, \frac{n}{k})$. The result now follows by symmetry, because each processor can be taken as processor $2k$. □

We are now ready to prove the main results in this section.

Theorem 3.4
For all $c, 1 < c < 2$, and all leader finding algorithms on unidirectional rings, where processors know the ring size n, the average number of messages sent on a ring of size n, n a power of 2, over all ring labellings $r \in D_n(I)$, with $|I| \geq cn$, is at least $(\frac{1}{c} - \frac{1}{c^2})n \log n - \mathcal{O}(n)$.

Proof.
Denote the average number of messages, sent between time $2^i + 1$ and 2^{i+1}, over all $r \in D_n(I)$, by $av(2^i+1, 2^{i+1})$. We now have the following lower bound for the number of messages, which must be estimated:

$$\begin{aligned}
\sum_{i=1}^{\log n-2} av(2^i+1, 2^{i+1}) &\geq \sum_{c=o}^{\log n-2} n \cdot \beta(\lfloor \frac{|I|}{2^i} \rfloor, \frac{n}{2^i}) \\
&\geq \sum_{i=1}^{\log n-2} n \left(\frac{\frac{n}{2^i}}{\lfloor \frac{|I|}{2^i} \rfloor} - \left(\frac{\frac{n}{2^i}}{\lfloor \frac{|I|}{2^i} \rfloor} \right)^2 - \mathcal{O}(\frac{1}{n}) \right) \\
&\geq n \cdot \sum_{i=1}^{\log n-2} \left(\frac{n}{|I|} - (\frac{n}{|I|})^2 - \mathcal{O}\left(\frac{\frac{n}{2^i}}{(\frac{|I|}{2^i})^2} \right) - \mathcal{O}(\frac{1}{n}) \right)
\end{aligned}$$

$$\geq n\left(\frac{1}{c}\log n - \frac{1}{c^2}\log n - \sum_{i=1}^{\log n-2} \mathcal{O}(\frac{1}{2^i})\right) - \mathcal{O}(\log n)$$
$$= (\frac{1}{c} - \frac{1}{c^2})n\log n - \mathcal{O}(n).$$

□

By taking a somewhat larger index set, one can improve the constant by a factor 2.

Theorem 3.5
For all leader finding algorithms on unidirectional rings, where processors know the ring size n, the average number of messages sent on a ring of size n, n a power of 2, over all ring labellings $r \in D_n(I)$, with $|I| \geq n^2$, is at least $\frac{1}{2}n\log n - \mathcal{O}(n)$.

Proof.
$$\sum_{i=1}^{\log n-2} n\beta(\lfloor\frac{|I|}{2^i}\rfloor, \frac{n}{2^i}) \geq n \cdot \sum_{i=1}^{\log n-2}\left(\frac{1}{2} - \frac{180\sqrt{\frac{|I|}{2^i}}}{(\frac{|I|}{2^i})-1}\right) = \frac{1}{2}n\log n - \mathcal{O}(n).$$ □

4 Lower bounds for leader finding on unidirectional rings for all fixed ring sizes.

In this section we modify the results of section 3 for the case that the ring size n is not fixed, but any arbitrary number $\in \mathbb{N}$. Basically, we lose a factor of 2 in comparison to the results in section 3, where certain assumptions on n could be made.

Let $k < \frac{1}{2}n$, $n = c_1k + c_2(k-1)$. For each $S \in X_k(I)$, we now define a graph $G_S = (S, E_S)$, with $E_S = \{(s,t) \mid s,t \in S, s \neq t$, processor $2k-1$ will not send a message between time k and $2k-1$ on a ring labelled with $r \in D_n(I)$, with $s = r_1 \ldots r_k$ and $t_1 \ldots t_{k-1} = r_{k+1} \ldots r_{2k-1}$ and processor $2k-2$ will not send a message between time k and $2k-2$ on a ring labelled with $r \in D_n(I)$ with $s_1 \ldots s_{k-1} = r_1 \ldots r_{k-1}$ and $t_1 \ldots t_{k-1} = r_k \ldots r_{k-2}\}$.

Lemma 4.1
Let $k < \frac{1}{2}n$, $n = c_1k + c_2(k-1)$, $S \in X_k(I)$. Then $G(S)$ does not contain a simple cycle with length $c_1 + c_2$.

Proof.
Suppose not. Let $s^1 \ldots s^{c_1+c_2}$ form a simple cycle in $G(S)$ with length $c_1 + c_2$. Let $t^i = s^i_1 \ldots s^i_{k-1}$. Then $r = s^1 \ldots s^{c_1} t^{c_1+1} \ldots t^{c_1+c_2} \in D_n(I)$. There is at least one processor that sends a message at time $2k-1$ on a ring labelled with r. Suppose the processor has identity s^i_j. $j \neq k$, else processor s^i_{k-1} sends a message at time $2k-2$, contradiction. We now consider two cases.

Case I. $i-1 \in \{1, \ldots c_1\}$. The processor with identity s^{i-1}_k sends a message at time $2k-1-j$, hence the processor with identity s^{i-1}_{k-1} sends a message at time $2k-2-j$. It follows that $2k-2-j < k \Rightarrow j \geq k-1$, and because $j \neq k, j = k-1$. But $(s^{i-1}, s^i) \in E(S)$, contradiction.

Case II. $i-1 \in \{0, c_1+1, \ldots, c_1+c_2-1\}$. Let $i^1 = i-1$, if $i-1 > 0$, and $i^i = c_1+c_2$, if $i-1=0$. Now the processor with identity $s_{k-1}^{i^1}$ sends a message at time $2k-1-j \in \{k, \ldots, 2k-2\}$, which is a contradiction. □

Lemma 4.2
Let $k \geq \frac{1}{2}n$, $n = c_1k + c_2(k-1)$. Let $m = \lfloor \frac{|I|}{k} \rfloor$. The number of messages, sent on a unidirectional ring with known ring size n between time k and $2k-1$, averaged over all ring labellings $r \in D_n(I)$, is at least $\beta(m, c_1+c_2) \cdot \frac{n}{2}$.

Proof.
For each $S \in X_k(I)$, let $A(S) = \{(s,t) \mid s \in S, t \in S, s \neq t$, processor $2k-1$ sends a message between time k and $2k-1$ on a ring labelled with $r \in D_n(I)$, with $s = r_1 \ldots r_k$ and $t_1 \ldots t_{k-1} = r_{k+1} \ldots r_{2k-1}\}$, and $B(S) = \{(s,t) \mid s \in S, t \in S, s \neq t$, processor $2k-2$ sends a message between time k and time $2k-2$ on a ring labelled $r \in D_n(I)$, with $s_1 \ldots s_{k-1} = r_1 \ldots r_{k-1}$ and $t_1 \ldots t_{k-1} = r_k \ldots r_{2k-2}\}$. Clearly $A(S) \cup B(S) \cup E_S = \{(s,t) \mid s,t \in S, s \neq t\}$. Hence, for each $S \in X_k(I)$, $|A(S)| + |B(S)| \geq \frac{1}{2}m(m-1) - \alpha(m, c_1+c_2)$. We consider two cases.

Case 1. The average of $|A(S)|$ over all $S \in X_k(I)$ is at least $\frac{1}{2}(\frac{1}{2}m(m-1) - \alpha(m, c_1+c_2))$. For $S \in X_k(I)$, let $r(S) = \{r \in D_n(I) \mid r_1 \ldots r_k \in S, r_{k+1} \ldots r_{2k} \in S\}$. For each $S \in X_k(I)$, the average number of messages sent by processor $2k-1$ between time k and $2k-1$ over all rings $r \in r(S)$, is at least $|A(S)|/m(m-1)$. As each ring $r \in D_n(I)$ belongs to the same number of sets $r(S)$, $S \in X_k(I)$, it follows that the average number of messages, sent by processor $2k-1$ between time k and $2k-1$, is at least the average over all $S \in X_k(I)$ of $|A(S)|/m(m-1)$, hence at least $\frac{1}{2}(m(m-1) - \alpha(m, c_1+c_2))/m(m-1) = \frac{1}{2}\beta(m, c_1+c_2)$. By symmetry, the same bound holds for the average number of messages sent by any other processor i between time k and $2k-1$. Hence, the average number of messages, sent between time k and $2k-1$ is at least $\frac{n}{2}\beta(m, c_1+c_2)$.

Case II. The average of $|B(S)|$ is at least $\frac{1}{2}(\frac{1}{2}m(m-1) - \alpha(m, c_1+c_2))$. Similar as in case 1, one derives that the average number of messages, sent between time k and $2k-2$ is at least $\frac{n}{2}\beta(m, c_1+c_2)$. So the result follows. □

Lemma 4.3
Let $t \leq k \leq \frac{1}{2}n$, $n = c_1k + c_2(k-1)$. Let $m = \lfloor \frac{|I|}{k} \rfloor$. The number of messages, sent on a unidirectional ring with known ring size n between time t and $2t-1$, averaged over all ring labellings $r \in D_n(I)$, is at least $\frac{t}{k} \cdot \beta(m, c_1+c_2) \cdot \frac{n}{2}$.

Proof.
The result follows from lemma 4.2, by observing that if $t_1 \geq t_2$, then the number of messages sent at time t_2 is at least the number of messages, sent at time t_1. □

Lemma 4.4
$\forall t \leq \frac{1}{4}n \; \exists k \leq \frac{1}{2}n, k > t$, $n = c_1k + c_2(k-1)$, c_1+c_2 is even, and $\frac{t}{k} \geq 1 - \frac{t}{n} - \frac{1}{t} - \frac{2(t-1)}{n\left(\frac{n}{(t-1)} - 2\right)}$.

Proof.
Let t be given. Let $c_3 = \lfloor \frac{n}{(t-1)} \rfloor$. If c_3 is even, let $c_{12} = c_3, c_4 = n - c_{12} \cdot (t-1)$. Now $0 \leq c_4 \leq t-1$. If c_3 is odd, let $c_{12} = c_3 - 1, c_4 = n - c_{12}(t-1)$. Now $t \leq c_4 < c_{12} + (t-1)$.

So, in both cases, $n = c_{12}(t-1) + c_4, c_{12}$ is even, and $0 \leq c_4 \leq c_{12} + (t-1)$. Write $c_4 = c_5 \cdot c_{12} + c_1$, $0 \leq c_1 < c_{12}$. Let $k = t + c_5$, $c_2 = c_{12} - c_1$. Clearly, $c_1 + c_2 = c_{12}$ is even. We claim that $n = c_1 k + c_2(k-1)$, and $\frac{t}{k} \geq \frac{t}{n} - \frac{1}{t}$. First note that $c_1 k + c_2(k-1) = c_{12}(k-1) + c_1 = c_{12}(t-1) + c_5 c_{12} + c_1 = c_{12}(t-1) + c_4 = n$. Secondly,

$$c_5 = \lfloor \frac{c_4}{c_{12}} \rfloor \leq \frac{t-1}{c_{12}} + 1 \leq \frac{(t-1)}{\frac{n}{(t-1)} - 2} + 1 = \frac{(t-1)\left(\frac{n}{(t-1)} - 2\right)}{\left(\frac{n}{(t-1)}\right)\left(\frac{n}{(t-1)} - 2\right)} + \frac{2(t-1)}{\frac{n}{(t-1)}\left(\frac{n}{(t-1)} - 2\right)} + 1$$

$$\leq \frac{t^2}{n} + \frac{2(t-1)^2}{n\left(\frac{n}{(t-1)} - 2\right)} + 1 \leq \frac{t^2}{n} + 1 + \frac{2(t-1)t}{n\left(\frac{n}{(t-1)} - 2\right)}.$$

Hence, $\frac{t}{k} = \frac{t}{t + c_5} = 1 - \frac{c_5}{t + c_5} \geq 1 - \frac{c_5}{t} \geq 1 - \frac{t}{n} - \frac{1}{t} - \frac{2(t-1)}{n\left(\frac{n}{(t-1)} - 2\right)}$.

Finally, $t \leq \frac{1}{4}n \Rightarrow c_{12} \geq 3 \Rightarrow k - 1 \leq \frac{n}{3} \Rightarrow k \leq \frac{1}{2}n$. □

Theorem 4.5
For all $c, 1 < c < 2$, and all leader finding algorithms on unidirectional rings, where processors know the ring size n, the average number of messages sent on a ring of size n, over all ring labellings $r \in D_n(I)$, with $|I| \geq cn$, is at least $\frac{1}{2}\left(\frac{1}{c} - \frac{1}{c^2}\right) n \log n - \mathcal{O}(n)$.

Proof.
First we estimate the average number of messages, sent between time 2^i and $2^{i+1} - 1$ as follows ($i = 1 \ldots \lfloor \log n \rfloor - 3$). Let $k > 2^i, n = c_1 k + c_2(k-1)$, and $\frac{2^i}{k} \geq 1 - \frac{2^i}{n} - \frac{1}{2^i} - \frac{2(2^i-1)}{n\left(\frac{n}{(2^i-1)} - 2\right)}$ as indicated by lemma 4.4.

Write $\lfloor \frac{|I|}{2^i} \rfloor = m$. The average number of messages, sent between time 2^i and $2^{i+1} - 1$ is at least $\frac{1}{2} \cdot \frac{2^i}{k} \cdot \beta(m, c_1 + c_2)$, by lemma 4.3. As $c_1 + c_2 \leq \frac{n}{(k-1)} \leq \frac{n}{2^i}$, this is at least

$$\frac{1}{2} \cdot \frac{2^i}{k} n \left(\frac{\frac{n}{2^i}}{m} - \left(\frac{\frac{n}{2^i}}{m}\right)^2 \pm \mathcal{O}\left(\frac{1}{m}\right) \right) \geq \frac{1}{2} \cdot \frac{2^i}{k} n \left(\frac{1}{c} - \frac{1}{c^2} \pm \mathcal{O}\left(\frac{1}{m}\right) \right) \geq$$

$$\frac{1}{2}\left(1 - \frac{2^i}{n} - \frac{1}{2^i} - \frac{2(2^i-1)}{n\left(\frac{n}{(2^i-1)-2}\right)}\right) \cdot n \left(\frac{1}{c} - \frac{1}{c^2} - \mathcal{O}\left(\frac{2^i}{cn}\right) \right).$$

So, the total average number of messages is at least

$$\sum_{i=1}^{\lfloor \log n \rfloor - 3} \frac{1}{2}\left(1 - \frac{2^i}{n} - \frac{1}{2^i} - \frac{2(2^i-1)}{n\left(\frac{n}{(2^i-1)-2}\right)}\right) n \left(\frac{1}{c} - \frac{1}{c^2} - \mathcal{O}\left(\frac{2^i}{cn}\right) \right)$$
$$= \frac{1}{2}\left(\frac{1}{c} - \frac{1}{c^2}\right) n \log n - \mathcal{O}(n).$$ □

Theorem 4.6
For all leader finding algorithms on unidirectional rings, where processors know the ring size n, the average number of messages sent on a ring of size n, n a power of 2, over all ring labellings $r \in D_n(I)$, with $|I| \leq n^2$, is at least $\frac{1}{4}n \log n - \mathcal{O}(n)$.

Proof.
Similar as in theorem 3.5 and 4.5. □

5 Lower bounds for leader finding on bidirectional rings.

The lower bounds for problems on bidirectional rings are of the type, where we average over all rings, labelled with strings $\in D_n(I)$, but where the delay times may be chosen in any manner, in order to obtain an as large as possible number of messages. All lower bounds for the average number of messages for leader finding on asynchronous rings we know of, are of this type. Here we assume that all message delay times are equal, i.e. each message takes unit time. Further assume that when a processor receives two messages (from both neighbors) at the same moment, it handles the left one first. In this way we lose the implicit non-determinism, associated with asynchronous, bidirectional rings.

So we may assume that we have an asynchronous, message-driven algorithm, running on a synchronous ring. We again assume that all processors start to send at time 1. Note that it depends only on the id's of the processors with distance at most $t-1$ to processor i, whether or not processor i will send a message at time t or not.

Lemma 5.1
Let $|I| \geq n+1$. Then, for all $r \in D_n(I)$ and $t < \frac{1}{2}n$, there is at least one processor that sends a message at time t on a ring labelled r.

Proof.
Similar to lemma 3.1. □

Now for all $k \leq \frac{1}{2}n$, $2|k$, $l \leq \frac{1}{2}k$ and each $S \in X_k(I)$ we define the following directed graph $H_l(S) = (S, E_l(S))$, by $E_l(S) = \{(s,t) \mid s,t \in S$; when a ring is labelled with $r \in D_n(I)$, with $s = r_1 \ldots r_k, t = r_{k+1} \ldots r_{2k}$, then none of the processors $\frac{1}{2}k+1, \frac{1}{2}k+2, \ldots, 1\frac{1}{2}k-1, 1\frac{1}{2}k$ sends a message at time $l\}$.

Lemma 5.2
Let $k|n$, $2|k$, $l \leq \frac{1}{2}k$, $k \leq \frac{1}{2}n, S \in X_k(I)$. Then $H_l(S)$ does not contain a simple directed cycle with length $\frac{n}{k}$.

Proof.
Suppose $H_l(S)$ contains a cycle with length $\frac{n}{k}$, say $s_1, \ldots, s_{\frac{n}{k}}$. Then on a ring labelled $s_1 \cdot s_2 \cdot \cdots \cdot s_{\frac{n}{k}}$ no processor sends a message at time l. Contradiction. □

Theorem 5.3
Let $k|n$, $2|k$, $l \leq \frac{1}{2}k$, $k \leq \frac{1}{2}n$. Then the average number of messages, sent at time l, over all rings, labelled with $r \in D_n(I)$, is at least $\frac{n}{k} \cdot \beta(\frac{|I|}{k}, \frac{n}{k})$.

Proof.
Consider some $S \in X_k(I)$. Since any non-edge in $H_l(S)$ corresponds to a message, sent at time l, by a processor in $\frac{1}{2}k+1 \ldots 1\frac{1}{2}k$, the average number of messages sent by processors $\frac{1}{2}k+1, \ldots, 1\frac{1}{2}k$ at time l is at least $\beta(\frac{|I|}{k}, \frac{n}{k})$. Again we argue that each $r \in D_n(I)$ is derived from the same number of $S \in X_k(I)$. It follows that the average number of messages, sent by processors $\frac{1}{2}k+1, \ldots, 1\frac{1}{2}k$ at time l, over all $r \in D_n(I)$ is at least $\beta(\frac{|I|}{k}, \frac{N}{k})$. The result now follows by symmetry, as every k successive processors can be taken as processors $\frac{1}{2}k+1, \ldots, 1\frac{1}{2}k$. □

Theorem 5.4
For all $c, 1 < c < 2$, and all leader finding algorithms on bidirectional rings, where processors know the ring size n, the average number of messages sent on a ring of size n, n a power of 2, over all ring labellings $r \in D_n(I)$, with $|I| \leq cn$, is at least $\frac{1}{4}(\frac{1}{c} - \frac{1}{c^2})n \log n - \mathcal{O}(n)$.

Proof.
It follows from theorem 5.3. that between times $\frac{1}{4}k+1$ and $\frac{1}{2}k$, at least $\beta(\frac{|I|}{k}, \frac{n}{k}) \cdot \frac{n}{4}$ messages are sent, on the average over all ring labellings $r \in D_n(I)$. Now the result follows, similar as in theorem 3.4. □

Similar as in theorem 3.5., one can improve the constant by taking $|I| \geq n^2$. In this way one obtains basically the same lower bound as Duris and Galil [12], with the main difference that $|I|$ is here polynomial instead of exponential in n.

Theorem 5.5
For all leader finding algorithms on bidirectional rings, where processors know the ring size n, the average number of messages sent on a ring of size n, n, a power of 2, over all ring labellings $r \in D_n(I)$, with $|I| \geq n^2$, is at least $\frac{1}{8}n \log n - \mathcal{O}(n)$.

For n of the form $2m!$, we can obtain lower bounds with (asymptotically) a better constant factor. Define for n even: $f(l,n) = \min\{k \geq 2l \mid 2|k \text{ and } k|n\}$.

Lemma 5.6
Let $n = 2(m!)$. Then $\sum_{l=1}^{\frac{1}{2}n} \frac{1}{f(l,n)} \geq \frac{1}{2}H_n - \mathcal{O}(m)$.

Proof.
Write

$$\begin{aligned}\sum_{l=1}^{\frac{1}{2}n} \frac{1}{f(l,n)} &= \sum_{i=1}^{m-1} \sum_{j=1}^{i} \sum_{l=j\cdot(\frac{\frac{1}{2}n}{(i+1)!})+1}^{(j+1)(\frac{\frac{1}{2}n}{(i+1)!})} \frac{1}{f(l,n)} \\ &\geq \sum_{i=1}^{m-1} \sum_{j=1}^{i} (\frac{\frac{1}{2}n}{(i+1)!}) \cdot \frac{1}{(j+1)n(i+1)!}\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{m-1} \frac{1}{2}(H_i - 1) \\
&\geq \sum_{i=1}^{m-1} \frac{1}{2} ln(i) - \mathcal{O}(m) \\
&= \frac{1}{2} ln((m-1)!) - \mathcal{O}(m) \\
&= \frac{1}{2} H_n - \mathcal{O}(m).
\end{aligned}
$$

($ln(x)$ denotes the logarithm of x to the base e.) □

Note that $\sum_{l=1}^{\frac{1}{4}n} \frac{1}{f(l,n)} = \sum_{l=1}^{\frac{1}{2}n} \frac{1}{f(l,n)} \cdot \pm\mathcal{O}(1)$.

Theorem 5.7
For every $\varepsilon > o$, there are infinitely many $n \in \mathbf{N}^+$, such that for all leader finding algorithms on bidirectional rings, where processors know the ring size n, the average number of messages sent on a ring with size n, over all ring labellings $r \in D_n(I)$, with $|I| \geq n^2$, is at least $(\frac{1}{4} - \varepsilon) n H_n$.

Proof.
It follows from theorem 5.3. that one can estimate the desired average by

$$
\sum_{l=1}^{\frac{1}{4}n} \frac{n}{f(l,n)} \cdot \beta\left(\frac{|I|}{f(l,n)}, \frac{n}{f(l,n)}\right) \geq \sum_{l=1}^{\frac{1}{4}n} \frac{n}{f(l,n)} \cdot \left(\frac{1}{2} - \frac{180\sqrt{\frac{|I|}{f(l,n)}}}{\frac{|I|}{f(l,n)-1}}\right).
$$

Note that

$$
\sum_{l=1}^{\frac{1}{4}n} \frac{180\sqrt{\frac{|I|}{f(l,n)}}}{\frac{|I|}{f(l,n)-1}} = \mathcal{O}(1).
$$

When we take n of the form $2 \cdot (m!)$, then from lemma 5.6 and the observation that

$$
\sum_{l=1}^{\frac{1}{4}n} \frac{1}{f(l,n)} = \sum_{l=1}^{\frac{1}{2}n} \frac{1}{f(l,n)} - \mathcal{O}(1)
$$

it follows that

$$
\sum_{l=1}^{\frac{1}{4}n} \frac{n}{f(l,n)} \cdot \frac{1}{2} = \frac{1}{4} n H_n - \mathcal{O}(m).
$$

The result follows now easily; by taking m large enough by given $\varepsilon > o$. □

Note that $\frac{1}{4} n H_n \approx 0.173 n \log n$. Combining the techniques of section 4 and of this section one can also obtain the following results:

Theorem 5.8
For all $c, 1 < c < 2$, and all leader finding algorithms on bidirectional rings, where processors know the ring size n, the average number of messages, sent on a ring of size n over all ring labellings $r \in D_n(I)$ with $|I| \geq cn$, is at least $\frac{1}{4}\left(\frac{1}{c} - \frac{1}{c^2}\right) nH_n - \mathcal{O}(n)$.

Theorem 5.9
For all leader finding algorithms on bidirectional rings, where processors know the ring size n, the average number of messages, sent on a ring of size n over all ring labellings $r \in D_n(I)$, with $|I| \geq n^2$, is at least $\frac{1}{8}nH_n - \mathcal{O}(n) \approx 0.081 n \log n - \mathcal{O}(n)$.

6 Randomized versus deterministic algorithms on non-anonymous networks.

In this section we give a negative result on randomization for problems on non-anonymous networks. To ease presentation, we again assume that the algorithm is message-driven, and the average or expected number of messages is counted when each message takes unit time, and all processors become active at the same moment. However, similar results can be proved when weaker assumptions hold.
We can limit ourselves to randomized algorithms that always terminate within a bounded number of steps. The following lemma justifies this assumption.

Lemma 6.1
If there exists a randomized algorithm A that solves problem P on networks of type N, that uses an expected number of messages α on network $G \in N$, and terminates with probability 1, and a deterministic algorithm B that solves P on networks of type N, then for every $\varepsilon > 0$, there exists a randomized algorithm C that solves problem P on networks of type N, that uses an expected numer of messages $\alpha + \varepsilon$ on G, and always terminates in a finite number of steps.

Proof.
Run A until termination, or if a processor has sent $>> \alpha$ messages. In the latter case, stop A, start B, and broadcast to stop A. □

A message-driven algorithm A can be seen as an effectively computable function, which maps each pair, consisting of a state S of a processor and the (non-empty) set m of incoming messages to the set $A((s,m))$ of 3-tupples, consisting of a new state S' of a processor, a (possibly empty) set of messages m' it sends, and the probability $p \in (0,1]$, that a processor with state s' and incoming messages m goes to state s' and sends messages m'. We must have for each pair $(s,m) : \sum_{(s',m',p) \in A((s,m))} p = 1$.

For a deterministic algorithm, $|A((s,m))| = 1$ for all (s,m). The entry "1" is sometimes dropped.

A function A of the form, described above, which is not necessarily computable, is called a pseudo-algorithm. For each (pseudo-) algorithm A we have a set of deterministic pseudo-algorithms $PS(A)$, which are obtained by chosing for each (s,m) a unique possible transition of $A((s,m)) : B \in PS(A) \Leftrightarrow \forall (s,m) \in \text{dom(A)} : \exists (s',m',p) \in A((s,m)) : (s',m',1) \in B((s,m))$.

If A solves a distributed problem P and always terminates in a bounded number of steps, then all $B \in PS(A)$ solve P too.

The set of all input configurations (= the set of all possible labellings of all processors with an identity and an input) is denoted by I. I is assumed to be of finite size for a given network G. We assume the identity of a processor to be part of its state. By giving each processor an extra counter that is increased by one with each state-transition, we may assume that no state is reached more than once in any run of the algorithm. The set of possible runs of (pseudo)- algorithm A on input i, when each message takes unit time and all processors start simultaneously is denoted by $r(i, A)$. The set of all pairs (state, set of incoming messages) reached in run r is denoted by $s(r)$, the set of all actions (state, set of incoming messages) $\rightarrow$ (new state, set of outgoing messages) that occur in run r is denoted by $a(r)$. An action $(s, m) \rightarrow (s', m')$ is also somtimes denoted as $(s, m) \rightarrow (s', m', p)$, if $(s', m', p) \in A((s, m))$. The probability of an action $a = (s, m) \rightarrow (s', m')$ is denoted by $p(a)$, i.e. $(s', m', p((s, m) \rightarrow (s', m'))) \in A((s, m))$. The set of all actions, possible from (s, m) is $\alpha(A(s, m)) = \{(s, m) \rightarrow (s', m', p) \mid (s', m', p) \in A((s, m))\}$. The set of all actions is $\alpha(\mathrm{A}) = \bigcup_{(s,m) \in \mathrm{dom(A)}} \alpha(\mathrm{A},(\mathrm{s},\mathrm{m}))$.

For given A, enumerate the pseudo-algorithms from $PS(A) : B_1, B_2, B_3, \ldots$. In order to handle $PS(A)$ with infinite size, we use the following notations:

$$\alpha(A, q) = \bigcup_{1 \leq i \leq q} \alpha(B_i)$$

$$r(i, A, q) = \{r \in r(i, A) \mid r \text{ uses only actions from } \alpha(\mathrm{A},\mathrm{q})\}$$

$$S_q = \{(s, m) \in S \mid \exists (s', m', q) \text{ with } ((s, m) \rightarrow (s', m', q)) \in \alpha(A, q)\} = \bigcup_{1 \leq i \leq q} \mathrm{dom}(B_i).$$

$$PS(A, q) = \{B_1, \ldots, B_q\}$$

$$\alpha(A, (s, m), q) = \alpha(A, (s, m)) \cap \alpha(A, q).$$

Theorem 6.2
Suppose that for every deterministic pseudo-algorithm for problem P, the average message complexity (bit complexity) over all inputs on network G is at least δ. Then for every randomized (pseudo)- algorithm for problem P, the expected message complexity (bit complexity) over all inputs on network G is at least δ.

Proof.
By lemma 6.1 it is sufficient to prove the result for randomized (pseudo)- algorithms that always terminate in a bounded number of steps.

Denote the complexity of a run r with compl(r). Let $S = \mathrm{dom(A)}$. For a run $r \in r(i, A)$, the probability that run r occurs with input i is $\prod_{a \in a(r)} p(a)$.

We prove the result for the case that $PS(A)$ is infinite. The proof for the finite case is easily derived from this case. We write:

$$\sum_{r \in r(i,A,q)} \mathrm{compl}(r) \cdot \prod_{a \in a(r)} p(a) =$$

$$\geq \sum_{r \in r(i,A,q)} \mathrm{compl}(r) \cdot \prod_{a \in a(r)} p(a) \cdot \prod_{(s,m) \in S_q - s(r)} \sum_{a \in \alpha(A,(s,m),q)} p(a)$$

$$= \sum_{r \in r(i,A,q)} \mathrm{compl}(r) \cdot \prod_{(s,m) \in S_q} \sum_{a \in \alpha(A,(s,m),q)} \begin{cases} p(a) & \text{if } (s,m) \notin s(r) \text{ or } a \in a(r) \\ 0 & \text{if } (s,m) \in s(r) \text{ and } a \notin a(r) \end{cases}$$

$$\geq \sum_{r \in r(i,A,q)} \mathrm{compl}(r) \sum_{B \in PS(A,q)} \prod_{a \in \alpha(B)} \begin{cases} p(a), & \text{if } (s,m) \notin s(r) \text{ or } a \in a(r) \\ 0, & \text{if } (s,m) \in s(r) \text{ and } a \in a(r) \end{cases}$$

$$= \sum_{B \in ps(A,q)} \sum_{r \in r(i,B)} \mathrm{compl}(r) \cdot \prod_{a \in \alpha(B)} p(a).$$

Hence the average complexity of algorithm A is:

$$\lim_{q \to \infty} \left(\sum_{i \in I} \sum_{r \in r(i,A,q)} \mathrm{compl}(r) \cdot \prod_{a \in a(r)} p(a) \right) / |I|$$

$$\geq \lim_{q \to \infty} \left(\sum_{i \in I} \sum_{B \in PS(A,q)} \sum_{r \in r(i,B)} \mathrm{compl}(r) \cdot \prod_{a \in \alpha(B)} p(a) \right) / |I|$$

$$= \lim_{q \to \infty} \sum_{B \in ps(A,q)} \left(\left(\sum_{i \in I} \sum_{r \in r(i,B)} \mathrm{compl}(r)/|I| \right) \cdot \prod_{a \in \alpha(B)} p(a) \right)$$

$$\geq \lim_{q \to \infty} \sum_{B \in ps(A,q)} \prod_{a \in \alpha(B)} p(a) \cdot \delta = \delta. \qquad \square$$

In other words, for any network G, if there exists a randomized algorithm that solves problem P on a class of networks including G, then there exists a deterministic pseudo-algorithm B for P, with the same or better expected bit complexity or message complexity on G. In several cases one can strengthen the result a little such that B is an algorithm and not merely a pseudo-algorithm, e.g. if the network size and the number of possible input-configurations are finite. For instance, we have:

Corollary 6.3
For all n, if there exists a randomized leader finding algorithm on bidirectional (unidirectional) rings with fixed size n with expected message complexity δ, then there exists a deterministic leader finding algorithm on bidirectional (unidirectional) rings with fixed ring size n with average message complexity $\leq \delta$.

An important corollary of this result is that theorems 3.4, 3.5, 4.5, 4.6, 5.5, 5.7, 5.8 and 5.9 hold also for randomized algorithms.

7 Other problems on rings of processors.

The same techniques can be used to prove lower bounds for several other problems on rings of processors. A well-studied problem is the complexity of cyclic functions on rings of processors, i.e. functions $f : \Sigma^n \to \Sigma'$ (Σ, Σ' some given alphabets), such that for all $x, y \in \Sigma^n$ with x is a cyclic shift of y, $f(x) = f(y)$.

We require all processors to decide on the output. Note there is a reduction from the case where at least one processor must decide on the input to this case, using $\mathcal{O}(n)$ extra messages.

For upper and lower bounds on the message and bit complexity of boolean functions of this type, see e.g. [2,3,4,8,18].

Definition.
A cyclic function $\Sigma^n \rightarrow \Sigma'$ is α -global on $\Sigma'' \subseteq \Sigma$, if for all $x \in (\Sigma'')^n$, there is an $y \in \Sigma^n$, with $f(x) \neq f(y)$ and $\exists k, 1 \leq k \leq n : x_{k \bmod n+1} \cdots x_{(k+\alpha-1) \bmod n+1} = y_{k \bmod n+1} \cdots y_{(k+\alpha-1) \bmod n+1}$.

Theorem 7.1
Let $f : \Sigma^n \rightarrow \Sigma'$ be a cyclic function, that is α -global on $\Sigma'' \subseteq \Sigma$. Then for any distributed algorithm that computes f on non-anonymous unidirectional or bidirectional rings, for any labelling of the processors with identities, any input $\in (\Sigma'')^n$, and any possible execution on a ring with this labelling and input, for every $t < \frac{1}{2}\alpha$, there is at least one processor that sends a message at time t.

Proof.
Suppose not, for input $x \in (\Sigma'')^n$. Then every processor must have decided upon time t on the output. But processor $(k + \frac{1}{2}\alpha - 1) \bmod n + 1$ will be in the same state at time t on a ring with input x or y. Contradiction. □

Similar as in section 3, 4 and 5 one can derive a lower bound for the average number of messages, sent on unidirectional or bidirectional rings with identities. With the results of section 6, it follows that the same bound holds for randomized algorithms as well.

Theorem 7.2
Let $f : \Sigma^n \rightarrow \Sigma'$ be a cyclic function, that is α- global on $\Sigma'' \subseteq \Sigma$. For every deterministic, or randomized algorithm that computes f on bidirectional non-anonymous rings with identities taken from a set I, with $|I| \geq cn$, the average or expected number of messages over all inputs $\in (\Sigma'')^n$, and all labellings $\in D_n(I)$ is at least $\frac{1}{4}(\frac{1}{c} - \frac{1}{c^2})nH_\alpha - \mathcal{O}(n) = \Omega(n \log \alpha)$.
Similar, but slightly better bounds follow if the ring is unidirectional, $|I| \geq n^2$, and/or n is of a special type. Basically, replace the factor $\log n$ by a factor $\log \alpha$ in the bounds of theorem 3.4, 3.5, 4.5, 4.6, 5.5, 5.7 and 5.9. We have some new results for the well-studied case of cyclic boolean functions.

Corollary 7.3
Let A be any distributed algorithm that computes XOR on a bidirectional ring of processors, where each processor has a unique identity taken from I, with $|I| \geq cn$ for some constant $c > 1$. Then the average number of messages sent by A over all ring labellings $\in D_n(I)$, and inputs $\in \{0,1\}^n$ is $\Omega(n \log n)$.

Abrahamson et. al. [2] define the expected complexity of an algorithm to be the maximum over all inputs of the expected number of bits sent on a ring with that input. They conjecture that the expected complexity of AND and OR is $\Omega(n \log n)$. We are even able to prove a stronger result. Let the expected message omplexity of an algorithm be the maximum over all inputs of the expected number of messages sent on a ring with that input.

Corollary 7.4
For any distributed algorithm A, that computes AND, OR or solves the orientation problem (see e.g. [3,4]) on a bidirectional ring, the expected message complexity is $\Omega(n \log n)$.

Proof.
Take $\Sigma'' = \{1\}$ (AND), $\Sigma'' = \{0\}$ (OR) or $\Sigma'' = \{\rightarrow\}$ (orientation problem). AND and OR are $(n-1)$ -global on Σ'', the orientation problem is $(\frac{n}{2}-1)$- global on Σ''. □

Corollary 7.4 can be extended to non-anonymous rings. In that case the expected message complexity must be defined as the maximum over all inputs ($\in \{0,1\}^n$) of the average over all ring labellings $\in D_n(I)$ ($|I| \geq cn$, or $|I| \geq n^2$) of the expected number of messages sent on a ring with that input and labelling. Again $\Omega(n \log n)$ lower bounds follow.

Acknowledgements

This work benefitted very much from suggestions of and discussions with Zui Galil, Anneke Schoone, Gerard Tel, Marinus Veldhorst and Manfred Warmuth.

References

[1] Abrahamson, K., A. Adler, R. Gelbart, L. Higham, and D. Kirkpatrick, *The bit complexity of probabilistic leader election on a unidirectional ring*, Techn. Rep. 86-3, Univ. of British Columbia, Vancouver B.C., 1986.

[2] Abrahamson, K., A. Adler, L. Higham, and D. Kirkpatrick, *Randomized function evaluation on a ring*, (preliminary version), in proceedings 2nd Int. Workshop on Distributed Algorithms, 1987.

[3] Attiya, C., M. Snir, and M.K. Warmuth, *Computing on an anonymous ring*, to appear in J.ACM; a preliminary version appeared in Proc. 4th Ann. ACm Symp. on Principles of Distributed Computation, pp. 196-203, 1985.

[4] Attiya, H., and M. Snir, *Bounds for the average message complexity of distributed algorithms*, to appear in proceedings Agean Workshop on Computing, 1988.

[5] Bodlaender, H.L., *Distributed Computing: Structure and Complexity*, (Ph.D.Thesis). CWI Tract 43, CWI, Amsterdam, the Netherlands, 1987.

[6] Bodlaender, H.L., *A better lowerbound for distribured leader finding in bidirectional asynchronous rings*, Techn. Rep. RUU-CS-87-13, Dept. of Comp. Sc., Univ. of Utrecht, Utrecht, 1987, to appear in Inform. Proc. Letters.

[7] Bodlaender, H.L., and J. van Leeuwen, *New upperbounds for decentralized extrema finding in a ring of processors*, in Proc. 3rd Ann. Symp. on Theoretical Aspects of Computer Science, 1986, pp. 119-129, Lect. otes in Comp. Sc. 210, Springer Verlag, Berlin.

[8] Bodlaender, H.L., S. Moran, and M. Warmuth, *The inherent complexity of asynchronous computations on non-anonymous rings*, draft paper.

[9] Bollobas, B., *Extremal Graph Theory*, Academic Press, London, 1978.

[10] Chang, E., and R. Roberts, *An improved algorithm for decentralized extrema-finding in circular configurations of processes*, C.ACM 22 (1979) 281-283.

[11] Dolev, D., M. Klawe, and N. Rodeh, *An $\mathcal{O}(n \log N)$ unidirectional distributed algorithm for extrema finding in a circle*, J. Algorithms 3 (1982) 245-260.

[12] Duris, P., and Z. Galil, *Two lowerbounds in Asynchronous Distributed Computation*, Proc. 28th Ann. IEEE Symp. on Foundations of Computer Science, 1987, pp. 326-330.

[13] Flayolet, P., *Personal communication*, 1986.

[14] Frederickson, G.N., and N.A. Lynch, *Electing a leader in a synchronous ring*, J.ACM 34 (1987) 95-115.

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

[16] van Leeuwen, J., and R.B. Tan, *An improved upperbound for decentralized extrema-finding in bidirectional rings of processors*, Techn. Rep. RUU-CS-85-23, Dept. of Computer Science, Univ. of Utrecht, Utrecht, 1985. To appear in Distributed Computing.

[17] Moran, S., M. Shalom, and S. Zaks, *An algorithm for distributed leader finding in bidirectional rings without common sense of direction*, Techn. Rep. Technion, Haifa, 1985.

[18] Moran, S., and M. Warmuth, *Groep theorems for distributed computation*, Proc. 5th Ann. ACM Symp. on Principles of Distributed Computing, 1986, pp. 131-140.

[19] Pachl, J., *A lowr bound for probabilistic distributed algorithms*, J. Alg. 8 (1987) 53-65.

[20] Pachl, J., E. Korach, and D. Rotem, *Lowerbounds for distributed maximum-finding algorithms*, J. ACM 31 (1984) 905-918.

[21] Peterson, G.L., it An $\mathcal{O}(n \log n)$ unidirectional algorithm for the circular extrema problem, ACM Trans. Prog. Lang. & Syst. 4(1982) 758-762.