Optimal Bit Complexity Randomised Distributed MIS and Maximal Matching Algorithms for Anonymous Rings

A. Fontaine, Y. Métivier, J.M. Robson and A. Zemmari

Université de Bordeaux, LaBRI, CNRS UMR 5800
351 cours de la Libération, 33405 Talence, France
{fontaine, metivier, robson, zemmari}@labri.fr

Abstract

We present and analyse Las Vegas distributed algorithms which compute a MIS or a maximal matching for anonymous rings. Their bit complexity and time complexity are $O(\sqrt{\log n})$ with high probability. These algorithms are optimal modulo a multiplicative constant. Beyond the complexity results, the interest of this work stands in the description and the analysis of these algorithms which may be easily generalised. Furthermore, these results show a separation between the complexity of the MIS problem (and of the maximal matching problem) on the one hand and the colouring problem on the other. Colouring can be computed only in $\Omega(\log n)$ rounds on rings with high probability, while MIS is shown to have a faster algorithm. This is in contrast to other models, in which MIS is at least as hard as colouring.

1. Introduction

1.1. The Problem

Let $G = (V, E)$ be a simple connected undirected graph. An independent set is a subset I of V such that no two members of I are adjacent. An independent set I is said to be maximal (MIS for short) if any vertex of G is in I or adjacent to a vertex of I. A matching is a subset M of E such that no two edges of M have a common vertex. A matching M is said to be maximal if any edge of G is in M or has an extremity linked to an edge in M.

In this paper we discuss how greedy selection for the computation of a maximal independent set or of a maximal matching in a ring of processors can be accomplished by exchange of messages between adjacent processors.

Usually, the topology of a distributed system is modelled by a graph and paradigms of distributed systems are encoded by classical problems in graph theory; among these classical problems one may cite the problems of vertex colouring, computing a maximal independent set, finding a vertex cover, finding a maximal matching or finding a graph decomposition. Each solution to one of these problems is a building block for many distributed algorithms: symmetry breaking, topology control, routing, resource allocation, network synchronisation.

Even if ring graphs are simple, they are used as a case study in many problems, as explained by Attiya and Welch in [AW04], page 31: *"rings are a convenient structure for message-passing systems and correspond to physical communication systems, for example, token rings."*

1.2. The Model

A general presentation may be found in [Tel00] (Chapter 9) or in [Lyn96] (Chapter 8).

The Network. We consider the standard message passing model for distributed computing. The communication model consists of a point-to-point communication network described by a simple unoriented ring graph $G = (V, E)$ where the vertices V represent network processors and the edges represent bidirectional communication channels. Processors communicate by message passing: a processor sends a message to another by depositing the message in the corresponding channel. The channels are FIFO, i.e., for each channel, the messages are delivered in the order they have been sent. Note that we consider only reliable systems: no fault can occur on processors or communication links.

We assume that the system is synchronous with simultaneous wakeup of processors: processors have access to a global clock and all processors start the algorithm at the same time.

Time Complexity. A round (cycle) of each processor is composed of the following three steps: 1. Send messages to (some of) the neighbours, 2. Receive messages from (some of) the neighbours, 3. Perform some local computation. As usual (see for example Peleg [Pel00]) the time complexity is the maximum possible number of rounds needed until every node has completed its computation.

Bit Complexity. A bit round is a round such that each processor can send/receive at most 1 bit to/from each neighbour. As in [KOSS06], the bit complexity of an algorithm $\mathcal{A}$ is the number of bit rounds to complete algorithm $\mathcal{A}$. One round of the algorithm contains 1 or more bit rounds.

Network and Processor Knowledge. The network is anonymous: unique identities are not available to distinguish the processors. We do not assume any knowledge on the size of the ring or on an upper bound on the size of the ring, any position or distance information. Each processor knows from which channel it receives a message. An important fact due to the initial symmetry is: *there is no deterministic distributed algorithm for anonymous ring graphs for solving the MIS problem or the maximal matching problem assuming all vertices wake up simultaneously,* see [Pel00].

Las Vegas Distributed Algorithms. A probabilistic algorithm is an algorithm which makes some random choices based on some given probability distributions.

A distributed probabilistic algorithm is a collection of local probabilistic algorithms. Since our networks is an anonymous ring, two processes have the same degree thus their local probabilistic algorithms are identical and have the same probability distribution.

A Las Vegas algorithm is a probabilistic algorithm which terminates with a positive probability (in general 1) and always produces a correct result.

1.3. Our Contribution

We present and analyse Las Vegas distributed algorithms which compute a MIS or a maximal matching for anonymous rings. Their bit complexity and time complexity are $O(\sqrt{\log n})$ with high probability[1] (w.h.p. for short). From [FMRZ], we deduce that these algorithms are optimal (modulo a multiplicative constant).

Beyond the complexity results, the interest of this work stands in the descriptions and the analyses of these algorithms which follow the same scheme and thus which may be easily generalised: this is illustrated in Conclusion with the 2-MIS problem.

Furthermore, these results show a separation between the complexities of the MIS problem and of the maximal matching problem on the one hand and the colouring problem on the other. Colouring can be computed only in $\Omega(\log n)$ rounds on rings with high probability (see [KOSS06]), while MIS is shown to have a faster algorithm. This is in contrast to other models, in which MIS is at least as hard as colouring.

Khothapalli et al. [KOSS06] prove that a Las Vegas algorithm can achieve a colouring of a cycle (in which all edges have the same orientation) with a $O(\sqrt{\log n})$ bit complexity with high probability and they prove that this result is tight by showing that the bit complexity of colouring an oriented cycle is $\Omega(\sqrt{\log n})$ with high probability. This leads to the question whether the bit complexity of the MIS or of the maximal matching is lower in an oriented ring than the bit complexity of the MIS or of the maximal matching in a non oriented ring. As it is easy to deduce from a MIS (or from a maximal matching) a 3-colouring in an oriented ring, we deduce from the result of [KOSS06] that the answer is no.

1.4. Related Works: Comparison and Comments

1.4.1. Bit Complexity and Single Bit Messages.

Classically, there are two models for the size of the messages: the LOCAL model and the CONGEST model (see [Pel00] p. 27). The first allows message of unlimited size while the second allows messages with a size bounded by $O(\log n)$ (n is the size of the network). In both models vertices have unique identifiers.

[1]With high probability means with probability $1 - o(n^{-1})$.

The model of this paper is anonymous (no uniqueness of identifiers for the nodes) and nodes have no global knowledge on the network such as its size. Thus, in this context, when a processor builds a message its size cannot depend on the size of the network and it is natural to consider single bit messages or more generally messages with bounded sizes.

In addition, bit complexity is considered as a finer measure of communication complexity and it has been studied for breaking symmetry or for colouring in [BMW94, BNNN90] or in [KOSS06, DMR08]. Dinitz et al. explain in [DMR08] that it may be viewed as a natural extension of communication complexity (introduced by Yao [Yao79]) to the analysis of tasks in a distributed setting. An introduction to this area can be found in Kushilevitz and Nisan [KN99].

1.4.2. MIS, Maximal Matching, Colouring for General Graphs

General considerations and many examples of Las Vegas distributed algorithms related to MIS, maximal matching or colouring can be found in [Pel00].

The computation of a MIS has been the object of extensive research on parallel and distributed complexity [ABI86, Lub86, AGLP89, Lin92]; Karp and Widgerson [KW84] have proved that the MIS problem is in NC. Some links with distributed graph colouring and some recent results on this problem can be found in [KW06]. The complexity of some special classes of graphs such as growth-bounded graphs is studied in [KMNW05]. Results have been obtained also for radio networks [MW05]. A major contribution is due to Luby [Lub86]. He gives a Las Vegas distributed algorithm. The main idea is to obtain for each vertex a *local total order* or a local election which breaks the local symmetry and then each vertex can decide locally whether it joins the MIS or not. Its time complexity is $O(\log n)$ and its bit complexity is $O(\log^2 n)$. Recently, a Las Vegas distributed algorithm has been presented in [MRSDZ11] which improved the bit complexity: its bit complexity is optimal and equal to $O(\log n)$ with high probability.

General considerations on the complexity of the maximal matching problem and a $O(\log^4 n)$ deterministic algorithm are presented in [HKP01] (in this paper the model allows an orientation of edges). Concerning the maximal matching problem in anonymous non oriented graphs, Israeli and Itai presented in [II86] a Las Vegas distributed algorithm for general graphs whose bit and time complexity is $O(\log n)$ with high probability, from [FMRZ] we deduce that it is optimal.

Vertex colouring is mainly studied under two assumptions: - vertices have unique identifiers, and more generally, they have an initial colouring, - every vertex has the same initial state and initially only knows its own edges. If vertices have an initial colour, Kuhn and Wattenhofer [KW06] have obtained efficient time complexity algorithms to obtain $O(\Delta)$ colours in the case where every vertex can only send its own current colour to all its neighbours. In [Joh99], Johansson analyses a simple randomised distributed vertex colouring algorithm for anonymous graphs. He proves that this algorithm runs in $O(\log n)$ rounds with high probability on graphs of size n. The size of each message is $\log n$, thus the bit complexity per channel of this algorithm is $O(\log^2 n)$.

[MRSDZ10] presents an optimal bit and time complexity Las Vegas distributed algorithm for colouring any anonymous graph in $O(\log n)$ bit rounds with high probability.

1.4.3. MIS, Maximal Matching, Colouring for Rings

Kothapalli et al. consider the family of anonymous rings and show in [KOSS06] that if only one bit can be sent along each edge in a round, then every Las Vegas distributed vertex colouring algorithm (in which every node has the same initial state and initially only knows its own edges) needs $\Omega(\log n)$ rounds with high probability to colour the ring of size n with any finite number of colours. Kothapalli et al. consider also the family of oriented rings and they prove that the bit complexity in this family is $\Omega(\sqrt{\log n})$ with high probability.

The table of Figure 1 summarises known results for the bit complexity and time complexity of the maximal independent set, the maximal matching and the colouring. We can note that corresponding algorithms need no initial knowledge on the graph such as the size, position or identities of vertices. Furthermore these bounds are tight.

	MIS	Maximal Matching	Colouring
General graphs	$\Theta(\log n)$	$\Theta(\log n)$	$\Theta(\log n)$
Rings	$\Theta(\sqrt{\log n})$	$\Theta(\sqrt{\log n})$	$\Theta(\log n)$
Cycles	$\Theta(\sqrt{\log n})$	$\Theta(\sqrt{\log n})$	$\Theta(\sqrt{\log n})$

Figure 1: Bit complexity and time complexity for general graphs, for rings and for cycles in which all edges have the same orientation.

2. An Algorithm for the MIS Problem in Rings

2.1. Overview of the Algorithm

The outcome of the algorithm is defined on each vertex v by a special variable $\eta(v)$ which represents the state of v. A vertex v joining the MIS sets $\eta(v)$ to 1 and a vertex v not joining the MIS sets $\eta(v)$ to 0. Once it takes the state 1 or the state 0, a vertex is said to be in a final state and becomes inactive. Initially, v is in an undetermined state: $\eta(v) =?$. A vertex can take an intermediate state setting $\eta(v)$ to X_0 or X_1, thus $\eta(v) \in \{?, X_0, X_1, 0, 1\}$.

The algorithm consists of phases, each phase composed of 3 stages: Drawing (D), Expansion (E) and Filling (F). A phase is defined as the sequential execution of D, E and F, denoted DEF. The algorithm is therefore a sequence of phases DEF denoted $(DEF)^*$.

The algorithm is presented as a timed rewriting system through rewriting rules. For a phase DEF, rewriting rules which correspond to D, E and F are authorised if the time is correct modulo 3. This system can easily be converted into a message passing algorithm in which each processor records its own state and a counter modulo 3. Furthermore after each stage D, E, and F each active processor needs 6 rounds of collecting information so that it knows the random choices or the new states in its neighborhood to distance 6. Below is the overview of each stage:

- Drawing: Every vertex with state ? and belonging to a connected subgraph of the ring of size 7 whose vertices have the state ? chooses uniformly at random one vertex among: its two neighbours and itself. We represent the fact "vertex v chooses vertex w" by an arrow from v to w. Then the state of each vertex may change according to the rule in Figure 2. This rule describes what happens if the pattern described in the top line in Figure 3 occurs. If this is not the case, vertices remain in the same state and are ready to take part in the next round. It may be noted that when a vertex needs to decide if it is in the configuration of Figure 2, it needs to examine 7 potential 7-tuples, at most one of which could be in this pattern.

$$\begin{tikzcd}
?_{v_0} \arrow[r] & ?_{v_1} \arrow[loop above] \arrow[r, dash] & ?_{v_2} \arrow[r] & ?_{v_3} \arrow[loop above] \arrow[d] & ?_{v_4} \arrow[l] \arrow[r, dash] & ?_{v_5} \arrow[loop above] & ?_{v_6} \arrow[l] \\
X_0 \arrow[r, dash] & X_1 \arrow[r, dash] & 0 \arrow[r, dash] & 1 \arrow[r, dash] & 0 \arrow[r, dash] & X_1 \arrow[r, dash] & X_0
\end{tikzcd}$$

Figure 2: The Drawing rule: this rule indicates that if a vertex v_0, in the state ?, chooses a neighbour, denoted v_1, which has chosen itself and which has a neighbour v_2 etc... then the state of v_0 becomes X_0, the state of v_1 becomes X_1, the state of v_2 becomes 0 etc...

Remark 1. Vertices with state 0 or 1 in Figure 2 are in a final state. At the end of this stage, vertices v_2, v_3 and v_4 in Figure 2 are in a final state with probability at least $(1/3)^7$.

Remark 2. If we consider the drawing rule described in Figure 2, it is clear that a vertex cannot be in "two different rules" on the same round (there is no possible ambiguity for modification of states).

A *final area* is a set of contiguous final-state vertices with the property that the subset formed from its vertices having state 1 is a maximal independent set, more precisely, the area is described by the following rational expression: $01((0+00)1)^*0$. An important invariant, denoted I, of the algorithm is the following: as soon as the drawing phase D leads to a final area, at the start of each stage, the ring is divided into final areas bounded on both sides by vertices in intermediate states separated by "?" as shown in Figure 3. A set of contiguous vertices in final states with vertices labelled X_1 and X_0 on both sides is called an extended final area. The invariant I holds until every vertex has a state in $\{0, 1\}$, and in this case, the algorithm is ended.

$$? - ? - X_0 - X_1 - 0 - 1 - 0 - X_1 - X_0 - ? - ?$$

$$? - ? - X_0 - X_1 - 0 - 1 - \cdots - 1 - 0 - X_1 - X_0 - ? - ?$$

Figure 3: Forms of the configurations occurring in the ring.

- Expansion: Every final area has two sides and attempts to expand by two vertices on each side. The expansion occurs on a side if there are at least four undetermined vertices adjacent to this side according to the rule in Figure 4.

  The expansion in a direction affects four vertices: two vertices obtain a final state (changing X_i into i), the states of two other vertices become intermediate. The two last vertices with ? state may not change and stay ? or change into X_i because of an expansion occurring on the other side.

  If there are 5 or 6 undetermined vertices between two extremes each of the extremes will try to apply expansion.

$$\begin{array}{c} 0 - 1 - 0 - X_1 - X_0 - ? - ? - ? - ? \\ \downarrow \\ 0 - 1 - 0 - 1 - 0 - X_1 - X_0 - - - - \end{array}$$

Figure 4: The Expansion rule.

- Filling: When extended final areas are separated by at most six vertices, the gap between them is filled as in Figure 5. Thus the filling stage gives a final state to all vertices in such gaps.

Remark 3. Rules $F4$, $F5$ and $F6$ are not necessary. If they are omitted the corresponding areas will be filled at the next phase.

Remark 4. The invariant I concerning the configuration of the ring described in Figure 3, is preserved by each phase.

Remark 5. Once a final area is formed, the number of vertices in final state is increased, at each phase, by at least 4. If the expansion stage occurs then this fact is trivial. Otherwise, that means that the filling stage occurs and in this case also the fact is trivial.

Remark 6. The algorithm is available only for graphs with at least 7 vertices. For fewer than 7 vertices one can apply an alternative $O(1)$ algorithm.

$F0$	$\cdots \cdot 0-1-0 \cdot X_1\text{-}X_0\text{-}X_0\text{-}X_1 \cdot 0-1-0 \cdot \cdots$ $\downarrow$ $\cdots \cdot 0-1-0-1-0-0-1-0-1-0 \cdot \cdots$
$F1$	$\cdots \cdot 0-1-0 \cdot X_1\text{-}X_0 \cdot ? \cdot X_0\text{-}X_1 \cdot 0-1-0 \cdot \cdots$ $\downarrow$ $\cdots \cdot 0-1-0-1-0-1-0-1-0-1-0 \cdot \cdots$
$F2$	$\cdots \cdot 0-1-0 \cdot X_1\text{-}X_0 \cdot ?-? \cdot X_0\text{-}X_1 \cdot 0-1-0 \cdot \cdots$ $\downarrow$ $\cdots \cdot 0-1-0-0-1-0-0-1-0-0-1-0 \cdot \cdots$
$F3$	$\cdots \cdot 0-1-0 \cdot X_1\text{-}X_0 \cdot ?-?-? \cdot X_0\text{-}X_1 \cdot 0-1-0 \cdot \cdots$ $\downarrow$ $\cdots \cdot 0-1-0-0-1-0-1-0-1-0-0-1-0 \cdot \cdots$
$F4$	$\cdots \cdot 0-1-0 \cdot X_1\text{-}X_0 \cdot ?-?-?-? \cdot X_0\text{-}X_1 \cdot 0-1-0 \cdot \cdots$ $\downarrow$ $\cdots \cdot 0-1-0-1-0-1-0-0-1-0-1-0-1-0 \cdot \cdots$
$F5$	$\cdots \cdot 0-1-0 \cdot X_1\text{-}X_0 \cdot ?-?-?-?-? \cdot X_0\text{-}X_1 \cdot 0-1-0 \cdot \cdots$ $\downarrow$ $\cdots \cdot 0-1-0-1-0-1-0-1-0-1-0-1-0-1-0 \cdot \cdots$
$F6$	$\cdots \cdot 0-1-0 \cdot X_1\text{-}X_0 \cdot ?-?-?-?-?-? \cdot X_0\text{-}X_1 \cdot 0-1-0 \cdot \cdots$ $\downarrow$ $\cdots \cdot 0-1-0-1-0-0-1-0-0-1-0-0-1-0-1-0 \cdot \cdots$

Figure 5: Filling rules. The rule $F0$ indicates that in contiguous extended final areas the states of vertices labelled X_0 (resp. X_1) become 0 (resp. 1). The rule $F1$ indicates that if there is exactly one vertex between two extended final areas then the state of this vertex becomes 1 and states of vertices labelled X_0 (resp. X_1) become 0 (resp. 1). The actions of the other rules are of the same kind.

2.2. Analysis of the Algorithm

We have the following lemma :

Lemma 1. *Let v be a vertex that enters the MIS or its complement at time $t_0 > 0$. Then, for any $t \geq 0$, after t phases, all the vertices at distance at most $2t \geq 0$ from v are in the MIS or in its complement.*

PROOF. The proof is direct from Remark 5, since any determined area is incremented by at least four vertices at the end of each phase whenever enough vertices are free. □

Theorem 2. *Algorithm* RingMIS *computes a MIS in a ring of size n in $O\left(\sqrt{\log n}\right)$ rounds w.h.p.*

PROOF. For any $u \in V$, and any $t > 1$, let $P(u,t)$ denote the path of length $2t+1$ centered on u and for any $i \in \{1, \cdots, t\}$, let $E(u,i)$ (resp. $\overline{E(u,i)}$) denote the event *"a vertex in the path $P(u, 2(t-i+1))$ enters the set in phase i"* (resp. *"no vertex in the path $P(u, 2(t-i+1))$ enters the set in phase i"*).

Let v be a vertex which is not yet in the MIS nor in its complement. By Lemma 1, v will enter the set or its complement within t phases unless none of the events $E(v,i)$, for any $i \in \{1, \cdots, t\}$ happen.

On the other hand, any vertex u in $P(v, 2t)$ distant 6 or more from all determined areas will enter the set in a phase if it is at the center of the pattern in Figure 2, and so with probability at least $\rho = \left(\frac{1}{3}\right)^7$ independently of what happens outside these seven vertices.

Thus:

$$\mathbb{P}r\left(\overline{E(v,i)}\right) \le (1-\rho)^{\frac{4(t-i+1)}{7}}.$$

Note that this is an upper-bound, and any more accurate (and tedious) computation will just affect the constants in the following formulas.

Now, if we denote $p_{>t}(v)$ the probability that v does not enter the set, or its complement, within t phases, then we have :

$$\begin{aligned}
p_{>t}(v) &\le \prod_{i=1}^{t} \mathbb{P}r\left(\overline{E(v,i)}\right) \qquad (1)\\
&\le \prod_{i=1}^{t} (1-\rho)^{\frac{4(t-i+1)}{7}}\\
&= (1-\rho)^{\frac{2}{7}(t^2+t)} < (1-\rho)^{\frac{2}{7}t^2}.
\end{aligned}$$

Let $t = 124\sqrt{\log n}$, then :

$$\begin{aligned}
p_{>124\sqrt{\log n}}(v) &\le (1-\rho)^{\frac{2}{7}124^2 \log n}\\
&\sim e^{-\frac{2}{7}124^2 \rho \log n} \text{ as } n \to \infty\\
&= o\left(\frac{1}{n^2}\right) \text{ as } n \to \infty (\text{ since } \frac{2}{3^7 7}124^2 > 2). \qquad (2)
\end{aligned}$$

Adding over all vertices we see that the probability that some vertex remains undetermined after $124\sqrt{\log n}$ phases is $o\left(n^{-1}\right)$. □

It follows that:

Corollary 1. *Algorithm* RingMIS *computes a MIS in a ring of size* n *in* $O\left(\sqrt{\log n}\right)$ *rounds on average.*

Proof. If some vertex remains undetermined after $124\sqrt{\log n}$ phases, either all vertices are undetermined and the time to end the algorithm is exactly the initial time (on average) or some vertices are determined and all vertices will be determined in, at most, $n/4$ more phases. However, the last case happens with probability $o(\frac{1}{n})$. It follows that the average time is also $O\left(\sqrt{\log n}\right)$. □

3. An Algorithm for the Maximal Matching Problem

3.1. Overview of the Algorithm

As for the MIS, the algorithm is a sequence of the form $(DEF)^*$: Drawing, Expansion, Filling.

The outcome is still defined on each vertex v by the variable $\eta(v)$. A vertex v joining the maximal matching sets $\eta(v)$ to 1. A vertex v not joining the maximal matching sets $\eta(v)$ to 0. Once it takes the state 1 or the state 0, a vertex is in a final state and becomes inactive. Initially, v is in an undetermined state: $\eta(v) = ?$. A vertex can take an intermediate state setting $\eta(v)$ to X, thus $\eta(v) \in \{?, X, 0, 1\}$.

Each extremity of an edge entering in the maximal matching knows the one it is linked with thanks to a port number. The representation of an edge entering in the maximal matching is as follows: $1 \overset{*}{-} 1$.

In the case of the maximal matching problem, a *final area* is defined as a set of contiguous final-state vertices with the property that the subset formed from its edges labelled with $*$ is a maximal matching, more precisely, the area is described by the following expression: $(1 \overset{*}{-} 1) + ((1 \overset{*}{-} 1) + (0 - 1 \overset{*}{-} 1))^*$. Here the invariant I of the algorithm is based on the configuration as given in Figure 6.

As for the MIS, the maximal matching algorithm is presented as a timed rewriting system through rewriting rules. This system can be converted into a message passing algorithm in which each processor records its own state and a counter modulo 3. Furthermore at each stage D, E, and F each active processor exchanges bit messages to know the results of the random choices and the states of vertices at distance 1, at distance 2, until 5, in order to reconstitute the corresponding subgraph.

Remark 7. An edge is labelled with * to represent the edges which may enter in the maximal matching. This information is known by each vertex thanks to port numering.

$$X \overset{*}{\longrightarrow} X \longrightarrow 1 \overset{*}{\longrightarrow} 1 \longrightarrow X \overset{*}{\longrightarrow} X$$
$$X \overset{*}{\longrightarrow} X \longrightarrow 1 \overset{*}{\longrightarrow} 1 \cdot [(1 \overset{*}{-} 1) + ((1 \overset{*}{-} 1) + (0 - 1 \overset{*}{-} 1))^*] \cdot 1 \overset{*}{\longrightarrow} 1 \longrightarrow X \overset{*}{\longrightarrow} X$$

Figure 6: Configurations occurring in the ring.

As previously we describe the various rules of rewriting for each stage:

- Drawing: Every vertex with state ? chooses uniformly at random one vertex among its two neighbours. We represent the fact "vertex v chooses vertex w" by an arrow from v to w. Two vertices which have chosen each other may change their state ? into 1 or X according to these two rules:
  - the state ? is changed into 1 if the edge is surrounded on each side by an edge whose extremities have chosen each other,
  - the state ? is changed into X if the edge is surrounded on one side by an edge whose extremities have chosen each other and on the other side by an edge whose extremities have not chosen each other,
  - no change.

$$\begin{tikzcd}
Y \arrow[r, dotted, dash] & Y \arrow[r, dotted, dash] & ? \arrow[r, leftrightarrow] & ? \arrow[r, dash] & (? \arrow[rr, leftrightarrow] & {} \arrow[d] & ?)^+ \arrow[r, dash] & ? \arrow[r, leftrightarrow] & ? \arrow[r, dotted, dash] & Y \arrow[r, dotted, dash] & Y \\
Y \arrow[r, dotted, dash] & Y \arrow[r, dotted, dash] & X \arrow[r, dash, "*"] & X \arrow[r, dash] & (1 \arrow[rr, dash, "*"] & {} & 1)^+ \arrow[r, dash] & X \arrow[r, dash, "*"] & X \arrow[r, dotted, dash] & Y \arrow[r, dotted, dash] & Y
\end{tikzcd}$$

Figure 7: The Drawing rule representing following cases. The pair $(Y \cdots Y)$ can be one of these patterns: $(-? \to ? \to)$, $(\leftarrow ? \leftarrow ? -)$, $(\leftarrow ? - ? \to)$, $(X \overset{*}{-} X)$, or $(? \to X)$, or $(\leftarrow ? - X)$ etc.

The rule describing the drawing is given as Figure 7.

Remark 8. An edge $1 \overset{*}{-} 1$ is in the maximal matching and remains in it.

- Expansion: The expansion occurs in a direction if there are at least four undetermined vertices adjacent to the extremity according to the rule in Figure 8.

$$\begin{tikzcd}
1 \arrow[r, leftrightarrow] & 1 \arrow[r, dash] & X \arrow[r, leftrightarrow] & X \arrow[rr, dash] & {} \arrow[d] & ? \arrow[r, dash] & ? \arrow[r, dash] & ? \arrow[r, dash] & ? \\
1 \arrow[r, dash, "*"] & 1 \arrow[r, dash] & 1 \arrow[r, dash, "*"] & 1 \arrow[rr, dash] & {} & X \arrow[r, dash, "*"] & X \arrow[r, dash] & - \arrow[r, dash] & -
\end{tikzcd}$$

Figure 8: The Expansion rule.

Remark 9. The state of the two last vertices with state ? may remain the same or may change into X. This is represented in the figure with $-$. Indeed, the rule can occur on the other side of the ring, implying a change of state.

- Filling: When final areas are separated by at most five vertices, the gap between them is filled as in Figure 9.

Remark 10. The invariant I concerning the configuration of the ring described in Figure 6, is preserved by each phase. At the end of the algorithm, only vertices in final state remain.

Remark 11. As in MIS, at each phase, the number of vertices of a final area is increased by at least 4.

Remark 12. The algorithm is available only for graphs with at least 6 vertices.

$F0$	$\begin{array}{c} \cdots - 1 \overset{*}{-} 1 - X \overset{*}{-} X - X \overset{*}{-} X - 1 \overset{*}{-} 1 - \cdots \\ \downarrow \\ \cdots - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - \cdots \end{array}$
$F1$	$\begin{array}{c} \cdots - 1 \overset{*}{-} 1 - X \overset{*}{-} X - ? - X \overset{*}{-} X - 1 \overset{*}{-} 1 - \cdots \\ \downarrow \\ \cdots - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - 0 - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - \cdots \end{array}$
$F2$	$\begin{array}{c} \cdots - 1 \overset{*}{-} 1 - X \overset{*}{-} X - ? - ? - X \overset{*}{-} X - 1 \overset{*}{-} 1 - \cdots \\ \downarrow \\ \cdots - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - \cdots \end{array}$
$F3$	$\begin{array}{c} \cdots - 1 \overset{*}{-} 1 - X \overset{*}{-} X - ? - ? - ? - X \overset{*}{-} X - 1 \overset{*}{-} 1 - \cdots \\ \downarrow \\ \cdots - 1 \overset{*}{-} 1 - 0 - 1 \overset{*}{-} 1 - 0 - 1 \overset{*}{-} 1 - 0 - 1 \overset{*}{-} 1 - \cdots \end{array}$
$F4$	$\begin{array}{c} \cdots - 1 \overset{*}{-} 1 - X \overset{*}{-} X - ? - ? - ? - ? - X \overset{*}{-} X - 1 \overset{*}{-} 1 - \cdots \\ \downarrow \\ \cdots - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - \cdots \end{array}$
$F5$	$\begin{array}{c} \cdots - 1 \overset{*}{-} 1 - X \overset{*}{-} X - ? - ? - ? - ? - ? - X \overset{*}{-} X - 1 \overset{*}{-} 1 - \cdots \\ \downarrow \\ \cdots - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - 0 - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - 1 \overset{*}{-} 1 - \cdots \end{array}$

Figure 9: Filling rules. $F0$ describes the behaviour when two final areas are adjacent. $F1$ describes the behaviour when two final areas are separated by one vertex with state ?. More generally Fi ($i \leq 5$) describes the behaviour when two final areas are separated by i vertices. The undetermined state X and the undetermined state ? are changed into final one (1 or 0) ensuring a maximal matching.

3.2. Analysis of the Algorithm

We have the following theorem:

Theorem 3. *Algorithm* RingMM *computes a maximal matching in a ring of size n in $O\left(\sqrt{\log n}\right)$ rounds w.h.p. and on average.*

Proof. As for the proof of Theorem 2, the main idea is that if a vertex v enters its final state at time t_0, i.e., $\eta(v)$ belongs to the set $\{0, 1\}$, all the vertices at distance less than $2t$ from v enter their final state at time at most $t_0 + t$. On the other hand, any vertex distant 4 or more from all determined areas will enter the maximal matching in a phase if its state is 0 or 1 in one of the configurations described in Figure 7, and so with probability at least $\rho = \frac{1}{2^6}$ independently of what happens outside this configuration. If we denote by $p_{>t}(v)$ the probability for a vertex v to obtain its final state, an adaptation of the proof of Theorem 2 yields :

$$p_{>24\sqrt{\log n}}(v) \quad = \quad o\left(\frac{1}{n^2}\right) \text{ as } n \to \infty. \qquad (3)$$

Hence, the probability that some vertex remains undetermined after $24\sqrt{\log n}$ phases is $o\left(n^{-1}\right)$.

The second claim is also proved using the same argument as for Corollary 1. □

4. Conclusion

We present and analyse Las Vegas distributed algorithms which compute a MIS or a maximal matching for anonymous rings. Their bit complexity and time complexity are $O(\sqrt{\log n})$. These algorithms are optimal modulo a multiplicative constant.

The description and the analysis of these algorithms can be generalised. We illustrate this idea through the computation of a 2-MIS.

A 2-independent set is a subset K of V such that the distance between any two vertices of K is at least 3. A 2-independent set K is said to be maximal (2-MIS for short) if any vertex of G is in K or at distance 1 or 2 of an element of K.

As for the RingMIS, the outcome of the algorithm is defined on each vertex v by a special variable $\eta(v)$ which represents the state of v. A vertex v joining the MIS sets $\eta(v)$ to 1 and a vertex v not joining the MIS sets $\eta(v)$ to 0. Once it takes the state 1 or the state 0, a vertex is said to be in a final state and becomes inactive. Initially, v is in an undetermined state: $\eta(v) =?$. A vertex can take an intermediate state setting $\eta(v)$ to X_0 or X_1, thus $\eta(v) \in \{?, X_0, X_1, 0, 1\}$.

The algorithm Ring2MIS consists of phases, each phase is composed of 3 stages: Drawing (D), Expansion (E) and Filling (F). A phase is defined as a the sequential execution of D, E and F, denoted DEF. The algorithm is therefore a sequence of phases DEF denoted $(DEF)^*$.

The algorithm is presented as a timed rewriting system through rewriting rules. For a phase DEF, rewriting rules which correspond to D, E and F are authorised if the time is correct modulo 3. This system can easily be converted into a message passing algorithm in which each processor records its own state and a counter modulo 3. Furthermore at each stage D, E, and F each active processor exchanges bit messages to know the results of the random choices and the states of vertices at distance 1, at distance 2, until 10, in order to reconstitute the corresponding subgraph.

Below is the overview of each stage:

- Drawing: Every vertex with state ? and belonging to a connected subgraph of the ring of size 11 whose vertices have the state ? chooses uniformly at random one vertex among: its two neighbours and itself. We represent the fact "vertex v chooses vertex w" by an arrow from v to w. Then the state of each vertex changes according to the rule in Figure 10.

$$\begin{tikzcd}
? \arrow[r] & ? \arrow[r] & ? \arrow[loop above] \arrow[r, dash] & ? \arrow[r] & ? \arrow[r] & ? \arrow[loop above] \arrow[d] & ? \arrow[l] & ? \arrow[l] \arrow[r, dash] & ? \arrow[loop above] & ? \arrow[l] & ? \arrow[l] \\
X_0 \arrow[r, dash] & X_0 \arrow[r, dash] & X_1 \arrow[r, dash] & 0 \arrow[r, dash] & 0 \arrow[r, dash] & 1 \arrow[r, dash] & 0 \arrow[r, dash] & 0 \arrow[r, dash] & X_1 \arrow[r, dash] & X_0 \arrow[r, dash] & X_0
\end{tikzcd}$$

Figure 10: The Drawing rule.

- Expansion: Every final area attempts to expand by 3 vertices in each direction. The expansion occurs in a direction if there are at least 6 undetermined vertices adjacent to the extremity according to the following rule (we only indicate the states of vertices for short):

  ??????$X_0X_0X_1$00100$X_1X_0X_0$??????
  becomes ???$X_0X_0X_1$00100100100$X_1X_0X_0$???.

The expansion in a direction affects 6 vertices: 3 vertices obtained a final state (changing X_i into i), the states of 3 other vertices become intermediate. The 3 last vertices with ? state may not change and stay ? or change into X_i because of an expansion occurring on the other side.

- Filling: When extended final areas are separated by at most 10 vertices, the gap between them is filled as (we only give the states of the end and of the beginning of final areas):

  –$F0$: $100X_1X_0X_0X_0X_0X_1001$ becomes 100100001001

  –$F1$: $100X_1X_0X_0?X_0X_0X_1001$ becomes 1001001001001

  –$F2$: $100X_1X_0X_0??X_0X_0X_1001$ becomes 01001000010010

  –$F3$: $100X_1X_0X_0???X_0X_0X_1001$ becomes 100100010001001

  –$F4$: $100X_1X_0X_0????X_0X_0X_1001$ becomes 1001001001001001

  –$F5$: $100X_1X_0X_0?????X_0X_0X_1001$ becomes 10010000100001001

  –$F6$: $100X_1X_0X_0??????X_0X_0X_1001$ becomes 100100100001001001

  –$F7$: $100X_1X_0X_0???????X_0X_0X_1001$ becomes 1001001001001001001

  –$F8$: $100X_1X_0X_0????????X_0X_0X_1001$ becomes 10010000100100001001

  –$F9$: $100X_1X_0X_0?????????X_0X_0X_1001$ becomes 100100100010001001001

  –$F10$: $100X_1X_0X_0??????????X_0X_0X_1001$ becomes 1001000010000100001001

  Remark 13. As for the MIS, Rules $F6$, $F7$, $F8$, $F9$ and $F10$ are not necessary. If they are omitted the corresponding areas will be filled at the next phase.

Finally, we can prove:

Theorem 4. *Algorithm* Ring2MIS *computes a* 2*-MIS in a ring of size* n *in* $O\left(\sqrt{\log n}\right)$ *rounds w.h.p. and on average.*

Our results raise open questions, here are some (suggested by a referee).

1. Under which conditions is the MIS problem easier than the the coloring problem?
2. An oriented ring seems to be more constrained than a non-oriented ring but our results suggest that the orientation of the ring helps the design of distributed algorithms such as MIS or maximal matching. Is this due to the fact that the orientation already breaks the symmetry of the ring?
3. If IDs are allowed to the nodes (or knowledge of an upper-bound of the size of the ring is given) how significantly will such hypotheses help the design of MIS/maximal matching algorithms?

References

[ABI86] N. Alon, L. Babai, and A. Itai. A fast and simple randomized parallel algorithm for the maximal independent set. *Journal of Algorithms*, 7(4):567–583, 1986.

[AGLP89] B. Awerbuch, A. V. Goldberg, M. Luby, and S. A. Plotkin. Network decomposition and locality in distributed computation. In *Proceedings of the 30th ACM Symposium on FOCS*, pages 364–369. ACM Press, 1989.

[AW04] H. Attiya and J. Welch. *Distributed Computing.* Wiley, 2004.

[BMW94] H. L. Bodlaender, S. Moran, and M. K. Warmuth. The distributed bit complexity of the ring: from the anonymous case to the non-anonymous case. *Inf. and comput.*, 114(2):34–50, 1994.

[BNNN90] A. Bar-Noy, J. Naor, and M. Naor. One-bit algorithms. *Distributed Computing*, 4:3–8, 1990.

[DMR08] Y. Dinitz, S. Moran, and S. Rajsbaum. Bit complexity of breaking and achieving symmetry in chains and rings. *Journal of the ACM*, 55(1), 2008.

[FMRZ] A. Fontaine, Y. Métivier, J.-M. Robson, and A. Zemmari. On lower bounds for the time and the bit complexity of some probabilistic distributed graph algorithms. *Submitted.*

[HKP01] M. Hanckowiak, M. Karonski, and A. Panconesi. On the distributed complexity of computing maximal matchings. *SIAM J. Discrete Math.*, 15(1):41–57, 2001.

[II86] A. Israeli and A. Itai. A fast and simple randomized parallel algorithm for maximal matching. *Inf. Process. Lett.*, 22(2):77–80, 1986.

[Joh99] Ö. Johansson. Simple distributed $(\Delta + 1)$-coloring of graphs. *Information Processing Letters*, 70(5):229–232, 1999.

[KMNW05] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer. Fast deterministic distributed maximal independent set computation on growth-bounded graphs. In *DISC*, pages 273–287, 2005.

[KN99] E. Kushilevitz and N. Nisan. *Communication complexity.* Cambridge University Press, 1999.

[KOSS06] K. Kothapalli, M. Onus, C. Scheideler, and C. Schindelhauer. Distributed coloring in $O(\sqrt{\log n})$ bit rounds. In *20th International Parallel and Distributed Processing Symposium (IPDPS 2006), Proceedings, 25-29 April 2006, Rhodes Island, Greece.* IEEE, 2006.

[KW84] R. M. Karp and A. Widgerson. A fast parallel algorithm for the maximal independent set problem. In *Proceedings of the 16th ACM Symposium on Theory of computing (STOC)*, pages 266–272. ACM Press, 1984.

[KW06] F. Kuhn and R. Wattenhofer. On the complexity of distributed graph coloring. In *Proceedings of the 25 Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 7–15. ACM Press, 2006.

[Lin92] N. Linial. Locality in distributed graph algorithms. *SIAM J. Comput.*, 21:193–201, 1992.

[Lub86] M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM J. Comput.*, 15:1036–1053, 1986.

[Lyn96] N. A. Lynch. *Distributed algorithms.* Morgan Kaufman, 1996.

[MRSDZ10] Y. Métivier, J. M. Robson, N. Saheb-Djahromi, and A. Zemmari. About randomised distributed graph colouring and graph partition algorithms. *Inf. Comput.*, 208(11):1296–1304, 2010.

[MRSDZ11] Y. Métivier, J.-M. Robson, N. Saheb-Djahromi, and A. Zemmari. An optimal bit complexity randomized distributed mis algorithm. *Distributed Computing*, 23(5-6):331–340, 2011.

[MW05] T. Moscibroda and R. Wattenhofer. Maximal independent set in radio networks. In *Proceedings of the 25 Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 148–157. ACM Press, 2005.

[Pel00] D. Peleg. *Distributed computing - A Locality-sensitive approach.* SIAM Monographs on discrete mathematics and applications, 2000.

[Tel00] G. Tel. *Introduction to distributed algorithms.* Cambridge University Press, 2000.

[Yao79] A. C. Yao. Some complexity questions related to distributed computing. In *Proceedings of the 11th ACM Symposium on Theory of computing (STOC)*, pages 209–213. ACM Press, 1979.