On Lower Bounds for the Time and the Bit Complexity of Some Probabilistic Distributed Graph Algorithms
(Extended Abstract)

Allyx Fontaine, Yves Métivier, John Michael Robson, and Akka Zemmari

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

Abstract. This paper concerns probabilistic distributed graph algorithms to solve classical graph problems such as colouring, maximal matching or maximal independent set. We consider anonymous networks (no unique identifiers are available) where vertices communicate by single bit messages. We present a general framework, based on coverings, for proving lower bounds for the bit complexity and thus the execution time to solve these problems. In this way we obtain new proofs of some well known results and some new ones.

1 Introduction

The Problems. For many problems on graphs, lower bounds on the bit complexity and on the execution time of a probabilistic distributed algorithm can be obtained in a simple way by considering disconnected graphs. These results may be considered unsatisfactory since we are normally interested in connected graphs and networks. In this paper we present a general framework, based on coverings, for proving such results for (dis)connected graphs and apply it to problems such as colouring, maximal matching, maximal independent set (MIS for short), or some generalisations such as the following. The MIS problem or the colouring problem may be generalised to a distance k for any positive integer k. More precisely, let $G = (V, E)$ be a graph and let k be a positive integer; a k-independent set is a subset M of V such that the distance between any two vertices of M is at least $k + 1$. If M is maximal for this property M is said to be a maximal k-independent set (k-MIS for short). A distance-k colouring of G is a colouring of vertices of G such that any two different vertices connected by a path of at most k edges have different colours. The distributed complexity of problems given above is of fundamental interest for the study and analysis of distributed algorithms. Usually, the topology of a distributed system is modelled by a graph and paradigms of distributed systems are represented by classical problems in graph theory cited above. Each solution to one of these problems is a building block for many distributed algorithms: symmetry breaking, topology control, routing, resource allocation or network synchronisation.

V. Geffert et al. (Eds.): SOFSEM 2014, LNCS 8327, pp. 235–245, 2014.

Network, Time Complexity, Bit Complexity, Network Knowledge. (A general presentation may be found in [Tel00]). We consider the standard message passing model for distributed computing. The communication model consists of a point-to-point communication network described by a connected graph $G = (V, E)$, where the vertices V represent network processes and the edges E represent bidirectional communication channels. Processes communicate by message passing: a process sends a message to another by depositing the message in the corresponding channel. The state of each process, with respect to a distributed algorithm, is represented by a label $\lambda(v)$ associated to the corresponding vertex $v \in V$. We denote by $\mathbf{G} = (G, \lambda)$ such a labelled graph. We assume the system is fully synchronous, namely, all processes start at the same time and time proceeds in synchronised rounds.

A round (cycle) of each process is composed of the following three steps. Firstly, it sends messages to (some) neighbours ; secondly, it receives messages from (some) neighbours ; thirdly, it performs some local computation. As usual the time complexity is the number of rounds needed until every node has completed its computation. By definition, in a bit round each vertex can send/receive at most 1 bit from each of its neighbours. The bit complexity of algorithm $\mathcal{A}$ is the number of bit rounds to complete algorithm $\mathcal{A}$ (see [KOSS06]).

The network $G = (V, E)$ is anonymous: unique identities are not available to distinguish the processes. We do not assume any global knowledge of the network, not even its size or an upper bound on its size. The processes do not require any position or distance information. Each process knows from which channel it receives or to which it sends a message, thus one supposes that the network is represented by a connected graph with a port numbering function defined as follows (where $I_G(u)$ denotes the set of vertices of G adjacent to u): given a graph $G = (V, E)$, a ***port numbering*** function δ is a set of local functions $\{\delta_u \mid u \in V\}$ such that for each vertex $u \in V$, δ_u is a bijection between $I_G(u)$ and the set of natural numbers between 1 and $\deg_G(u)$.

The network is anonymous; thus two processes with the same degree are identical. Note that we consider only reliable systems: no fault can occur on processes or communication links.

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 the network is anonymous, if two processes have the same degree then their local probabilistic algorithms are identical and have the same probability distribution. We assume that choices of vertices are independent. A Las Vegas algorithm is a probabilistic algorithm which terminates with a positive probability (in general 1) and always produces a correct result. In this paper, results on graphs having n vertices are expressed with high probability (w.h.p. for short), meaning with probability $1 - o(n^{-1})$.

Our Contribution. The main contributions of this work are general constructions, based on coverings, which present in a unified way proofs of lower bounds for the time complexity and the bit complexity of some graph problems. Some

of these lower bounds are already known, others are new. More precisely, thanks to coverings we build infinite families of disconnected or connected graphs with some port numberings, such that for each graph K having n vertices in those families, any Las Vegas distributed algorithm which uses messages of one bit cannot break some symmetries inside K after $c\sqrt{\log n}$ or after $c \log n$ rounds (for a certain constant c) with high probability. (i.e., some vertices remain in the same state for at least $c \log n$ rounds w.h.p.). From these constructions and results we deduce that:

- solving problems such as MIS[1], colouring[1], maximal matching[1], 2-MIS or distance-2 colouring takes $\Omega(\log n)$ rounds w.h.p. for an infinite family of disconnected graphs;
- solving the MIS problem or the maximal matching problem takes $\Omega(\sqrt{\log n})$ rounds w.h.p. for an infinite family of rings;
- solving problems like MIS[1], colouring[1], maximal matching, 2-MIS or distance-2 colouring takes $\Omega(\log n)$ rounds w.h.p. for an infinite family of connected graphs.

We deduce also that the maximal matching (resp. colouring, resp. MIS) Las Vegas distributed algorithm presented in [II86] (resp. [MRSDZ10], resp. [MRSDZ11]) is optimal (time and bit) modulo multiplicative constants. More precisely, the bit complexity of solutions presented in these papers is $O(\log n)$ w.h.p. for anonymous graphs having n vertices. These results can be summarised by:

Theorem 1.1. *The bit complexity of the MIS problem, the colouring problem, the maximal matching problem and the distance-2 colouring problem is $\Theta(\log n)$ w.h.p. for anonymous graphs with n vertices.*

If we consider the particular case of rings, we prove [FMRZar] that: $O(\sqrt{\log n})$ rounds are sufficient w.h.p. to compute a MIS or a maximal matching in a ring with n vertices. Thus, the bit complexity of the MIS problem or the maximal matching problem is $\Theta(\sqrt{\log n})$ w.h.p. for anonymous rings with n vertices.

Related Work. Bit complexity is considered as a finer measure of communication complexity and it has been studied for breaking and achieving symmetry or for colouring in [BMW94, 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].

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. With the same assumptions, as is explained in [MRSDZ10], from this result we also deduce that every distributed algorithm that computes a MIS needs, in general, $\Omega(\log n)$ rounds with high probability.

[1] Already known result ([KOSS06],[MRSDZ10]).

2 Preliminaries

We will consider digraphs with multiple arcs and self-loops. A ***digraph*** $D = (V(D), A(D), s_D, t_D)$ is defined by a set $V(D)$ of vertices, a set $A(D)$ of arcs and by two maps s_D and t_D that assign to each arc two elements of $V(D)$: a source and a target.

A ***symmetric*** digraph D is a digraph endowed with a symmetry, that is, an involution $Sym : A(D) \to A(D)$ such that for every $a \in A(D), s(a) = t(Sym(a))$. In a symmetric digraph D, the degree of a vertex v is $\deg_D(v) = |\{a \mid s(a) = v\}| = |\{a \mid t(a) = v\}|$.

A ***homomorphism*** between two digraphs maps vertices to vertices, arcs to arcs while preserving the incidence relation. More precisely, a homomorphism γ between the digraph D and the digraph D' is a mapping $\gamma\colon V(D) \cup A(D) \to V(D') \cup A(D')$ such that for each arc $a \in A(D)$, $\gamma(s(a)) = s(\gamma(a))$ and $\gamma(t(a)) = t(\gamma(a))$. A homomorphism $\gamma : D \to D'$ is an ***isomorphism*** if γ is bijective.

Throughout the paper, we will consider digraphs where the vertices and the arcs are labelled with labels from a recursive label set L. A digraph D labelled over L will be denoted by (D, λ), where $\lambda\colon V(D) \cup A(D) \to L$ is the ***labelling function***. A mapping $\gamma\colon V(D) \cup A(D) \to V(D') \cup A(D')$ is a homomorphism from (D, λ) to (D', λ') if γ is a digraph homomorphism from D to D' which preserves the labelling, i.e., such that $\lambda'(\gamma(x)) = \lambda(x)$ for every $x \in V(D) \cup A(D)$. Labelled graphs or digraphs will be designated by bold letters such as $\mathbf{D}, \mathbf{D}', \ldots$

Let (G, λ) be a labelled graph with the port numbering δ. We will denote by $(\mathrm{Dir}(\mathbf{G}), \delta')$ the symmetric labelled digraph $(\mathrm{Dir}(G), (\lambda, \delta'))$ constructed in the following way. The vertices of $\mathrm{Dir}(G)$ are the vertices of G and they have the same labels in $\mathbf{G}$ and in $\mathrm{Dir}(\mathbf{G})$. Each edge $\{u, v\}$ of G is replaced in $(\mathrm{Dir}(\mathbf{G}), \delta')$ by two arcs $a_{(u,v)}, a_{(v,u)} \in A(\mathrm{Dir}(G))$ such that $s(a_{(u,v)}) = t(a_{(v,u)}) = u$, $t(a_{(u,v)}) = s(a_{(v,u)}) = v$, $\delta'(a_{(u,v)}) = (\delta_u(v), \delta_v(u))$ and $\delta'(a_{(v,u)}) = (\delta_v(u), \delta_u(v))$. These arcs correspond for each vertex to input ports and output ports. By extension, the labelling δ' of arcs is called a port numbering. (see Figure 1). Note that this digraph does not contain loops or multiple arcs. The object we use for our study is $(\mathrm{Dir}(G), (\lambda, \delta'))$ and some results are stated with symmetric labelled digraphs.

Given a labelled graph $\mathbf{G} = (G, \lambda)$ with a port numbering δ, let $\mathbf{D} = (\mathrm{Dir}(\mathbf{G}), \delta')$ be the corresponding labelled digraph $(\mathrm{Dir}(G), (\lambda, \delta'))$. Let $\mathcal{A}$ be a synchronous distributed algorithm. We speak indifferently of an execution of $\mathcal{A}$ on $(\mathbf{G}, \delta)$ or on $\mathbf{D}$. The state of each process is represented by the label $\lambda(v)$ of the corresponding vertex v. Let $\mathbf{D}' = (Dir(\mathbf{G}'), \delta')$ be the labelled digraph obtained by the application of a step of $\mathcal{A}$ to $\mathbf{D}$. We recall that the system is synchronous, thus each vertex of D has a new state computed by a transition function which depends on the state of the vertex and the messages it has received. This transition is denoted by: $(Dir(\mathbf{G}), \delta') \underset{\mathcal{A}}{\Longrightarrow} (Dir(\mathbf{G}'), \delta')$ or $(\mathbf{G}, \delta) \underset{\mathcal{A}}{\Longrightarrow} (\mathbf{G}', \delta)$.

Let v be a vertex of $\mathbf{G}$. We denote by $(v, \lambda(v)) \underset{\mathcal{A}}{\Longrightarrow} (v, \lambda'(v))$ the transition associated to the vertex v of G.

Let r be a non-negative integer. A sequence $(\mathbf{D}_i)_{0\leq i\leq r}$ of labelled digraphs is called an $\mathcal{A}$-***execution*** (or an ***execution*** when $\mathcal{A}$ is clear from the context) of length r if $\mathbf{D}_{i+1}$ is obtained from $\mathbf{D}_i$ in one step of a run of $\mathcal{A}$; this is denoted by $\mathbf{D}_i\mathcal{A}\mathbf{D}_{i+1}$ for every $0 \leq i < r$. An execution of length 1 is a ***step***. Furthermore if $\mathcal{A}$ is a probabilistic synchronous distributed algorithm and if the execution of this step has probability p, then it will be denoted by: $\mathbf{G} \underset{p}{\Longrightarrow} \mathbf{G}'$.

Let $\mathcal{A}$ be a probabilistic synchronous distributed algorithm. Let G be a graph. Let δ be a port numbering of G. Let λ be a labelling of G. We have: $(\mathbf{G}, \delta) \underset{p}{\Longrightarrow} (\mathbf{G}', \delta)$ (with $\mathbf{G}' = (G, \lambda')$) if and only if p equals the product over $V(G)$ of the probabilities of the transitions $(v, \lambda(v)) \underset{\mathcal{A}}{\Longrightarrow} (v, \lambda'(v))$.

2.1 Coverings and Synchronous Distributed Algorithms

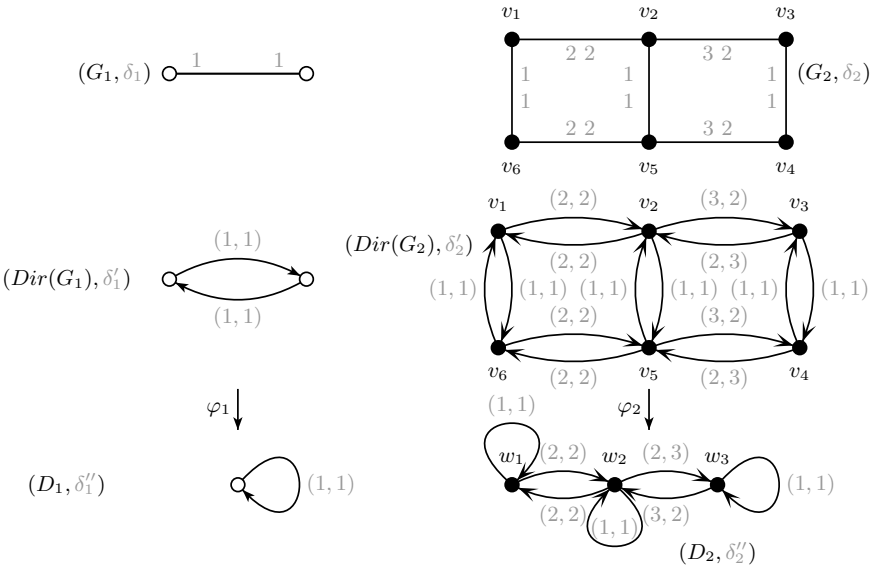

Fig. 1. The digraph $Dir(G_1)$ (resp. $Dir(G_2)$) is a symmetric covering via φ_1 (resp. φ_2) of D_1 (resp. D_2) where: φ_1 maps the vertices of $Dir(G_1)$ on the unique vertex of D_1 and $\varphi_2(v_1) = \varphi_2(v_6) = w_1$, $\varphi_2(v_2) = \varphi_2(v_5) = w_2$, and $\varphi_2(v_3) = \varphi_2(v_4) = w_3$. The number of sheets of these two coverings is 2. In grey, the port numbering δ_1'' (resp. δ_2'') of D_1 (resp. D_2) induces via φ_1^{-1} (resp. φ_2^{-2}) the port numbering δ_1' (resp. δ_2') of $Dir(G_1)$ (resp. $Dir(G_2)$) and thus the port numbering δ_1 (resp. δ_2) of G_1 (resp. G_2).

Definitions and principal properties of coverings are presented in [BV02]. A labelled digraph $\mathbf{D}$ is a ***covering*** of a labelled digraph $\mathbf{D}'$ via φ if φ is a homomorphism from $\mathbf{D}$ to $\mathbf{D}'$ such that for each arc $a' \in A(D')$ and for each vertex $v \in \varphi^{-1}(t(a'))$ (resp. $v \in \varphi^{-1}(s(a'))$), there exists a unique arc $a \in A(D)$ such that $t(a) = v$ (resp. $s(a) = v$) and $\varphi(a) = a'$.

The ***fibre*** over a vertex v' (resp. an arc a') of D' is defined as the set $\varphi^{-1}(v')$ of vertices of D (resp. the set $\varphi^{-1}(a')$ of arcs of D). An interesting property satisfied by coverings is that all the fibres of a given digraph have the same cardinality, that is called the ***number of sheets*** of the covering.

A symmetric labelled digraph $\mathbf{D}$ is a ***symmetric covering*** of a symmetric labelled digraph $\mathbf{D}'$ via φ if $\mathbf{D}$ is a covering of $\mathbf{D}'$ via φ and if for each arc $a \in A(D)$, $\varphi(Sym(a)) = Sym(\varphi(a))$. The homomorphism φ is a ***symmetric covering projection*** from $\mathbf{D}$ to $\mathbf{D}'$. Two examples are given in the parts in black of Figure 1.

Let D be a symmetric covering of D' via the homomorphism φ. Any port numbering on D' induces naturally via φ^{-1} a port numbering on D. Conversely, a port numbering on D induces a port numbering on D' via φ. (It is illustrated in grey in Figure 1).

Remark 2.1. Let $\mathbf{D}$ be a symmetric labelled covering of $\mathbf{D}'$ via φ. Let δ' be a port numbering of $\mathbf{D}'$ and let δ be the port numbering of $\mathbf{D}$ induced by φ^{-1}. Let $\mathcal{A}$ be a synchronous distributed algorithm. We consider the execution of one round of $\mathcal{A}$ on $(\mathbf{D}', \delta')$. This round can be lifted to $(\mathbf{D}, \delta)$ via φ^{-1} in the following way:

1. "send messages to (some) neighbours of $v' \in V(D')$" becomes "send messages to (some) neighbours of each vertex of $\varphi^{-1}(v') \subseteq V(D)$" (the same messages are sent from v' and from each vertex w of $\varphi^{-1}(v')$ through the same port numbers to (some) neighbours of v' and w);
2. "receive messages from (some) neighbours of $v' \in V(D')$" becomes "each vertex of $\varphi^{-1}(v') \subseteq V(D)$ receives messages from (some) its neighbours" (for each vertex w of $\varphi^{-1}(v')$ the same messages are received from (some) neighbours of v' and of w through the same port numbers);
3. "perform some local computation on $v' \in V(D')$" becomes "perform some local computation on $\varphi^{-1}(v') \subseteq V(D)$" (the same local computations are perform on v' and on each vertex of $\varphi^{-1}(v')$, as a consequence the vertex v' and the vertices of $\varphi^{-1}(v')$ are in same state).

Finally, we obtain the following classical lemma (see [Ang80, CM07]):

Lemma 2.2. *Let $\mathbf{D}$ be a symmetric labelled covering of $\mathbf{D}'$ via φ. Let δ' be a port numbering of $\mathbf{D}'$ and let δ be the port numbering of $\mathbf{D}$ induced by φ^{-1}. Then any execution of a distributed algorithm $\mathcal{A}$ on $(\mathbf{D}', \delta')$ can be lifted to an execution on $(\mathbf{D}, \delta)$, such that at the end of the execution, for any $v \in V(D)$, v is in the same state as $\varphi(v)$.*

As a direct consequence we have. An execution of a Las Vegas distributed algorithm $\mathcal{A}$ on a triangle induces, via the natural morphism, an execution on an hexagon. Hence, the following impossibility result holds:

Proposition 2.3. *Let k be a positive integer such that $k \geq 3$. There is no Las Vegas distributed algorithm for solving the distance-k colouring problem or the k-MIS problem.*

We are interested, in particular, in some problems like the computation of an MIS, colouring vertices or the computation of a maximal matching. In each case we have to break some symmetries inside a labelled graph. To break symmetries inside a labelled graph it suffices to distinguish a vertex. This is precisely the aim of an election algorithm. A distributed algorithm solves the election problem if it always terminates and in the final configuration, exactly one process is marked as *elected* and all the other processes are *non-elected*. Moreover, it is required that once a process becomes *elected* or *non-elected* then it remains in such a state until the end of the execution of the algorithm. The election problem is closely related to coverings. Indeed, a symmetric labelled digraph $\mathbf{D}$ is ***symmetric covering prime*** if there does not exist any symmetric labelled digraph $\mathbf{D}'$ not isomorphic to $\mathbf{D}$ such that $\mathbf{D}$ is a symmetric covering of $\mathbf{D}'$. We have [CM07]:

Theorem 2.4. *Given a connected graph G, there exists an election algorithm for G if and only if $Dir(G)$ is symmetric covering prime.*

We study some graph problems which need to break some initial symmetries; these symmetries are precisely encoded by some non-covering-prime digraphs and correspond to vertices inside a fibre. Thus in the sequel we consider non-covering-prime digraphs.

3 Obtaining Lower Bounds by Considering Disconnected Graphs

Proposition 3.1. *Let G be a graph having n_G vertices. Assume $Dir(G)$ is not symmetric covering prime. Let δ_1 be a port numbering of G. Let (K, δ) be the graph K with the port numbering δ formed by α copies of (G, δ_1) and let $n = \alpha n_G$. Then, there exists a constant $c > 0$ such that, for any Las Vegas distributed algorithm $\mathcal{A}$ that uses messages of 1 bit, there is at least one copy of (G, δ_1) such that for each fibre all its vertices are in the same state for at least $c \log n$ rounds w.h.p.*

The idea of the proof of Proposition 3.1 is based on the assumption that $Dir(G)$ is not covering prime. Then it is a covering of a non-isomorphic graph, say D'. Any execution of the Las Vegas algorithm $\mathcal{A}$ on D' induces an execution of $\mathcal{A}$ on $Dir(G)$. We consider the execution having the highest probability. We show that, *w.h.p.*, there exists a constant c such that there is a fibre in one of the copies of $Dir(G)$ in which all vertices are in the same state for at least $c\ log\ n$ rounds.

Corollary 3.2. *For every Las Vegas distributed algorithm $\mathcal{A}$ there is an infinite family $\mathcal{F}$ of disconnected graphs such that $\mathcal{A}$ has a bit complexity $\Omega(\log n)$ w.h.p. on $\mathcal{F}$ to solve either: the colouring problem, the MIS problem, the maximal matching problem, the 2-MIS problem, the distance-2 colouring problem.*

The idea of the proof is based on the choice of G. Take G, for each problem respectively as a: 1. single edge, 2. single edge, 3. triangle, 4. single edge, 5. single edge.

4 Obtaining Lower Bounds of the Form $\Omega(\sqrt{\log n})$ for Connected Graphs

The previous definition of coverings for digraphs becomes in the case of graphs: a graph G is a covering of a graph H via the homomorphism φ if φ is a homomorphism from G onto H such that for every vertex v of V the restriction of φ to neighbours of v is a bijection between neighbours of v and neighbours of $\varphi(v)$. The construction we give now has been presented by Reidemeister [Rei32] to describe all coverings of a given graph. We present briefly a precise description given by Bodlaender in [Bod89]. Let G be a graph. Consider a spanning tree (and more generally a spanning graph) S_T of G. Make α copies of S_T. For every edge in G that is not an edge of S_T, there are α copies of both its endpoints. We connect on a one-to-one basis every copy of the first endpoint to a unique copy of the second endpoint (see Figures 2 and 3). This construction builds a covering of G. Furthermore, given a spanning tree S_T of G, each covering of G can be obtained in this way from S_T. The number of copies is the number of sheets of the covering.

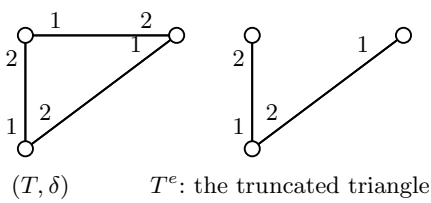

Fig. 2. A triangle T with a port numbering δ and the associated truncated triangle

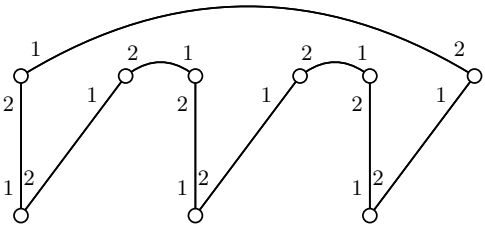

Fig. 3. Reidemeister's construction with 3 copies of the truncated triangle T^e. We obtain $R(T, e, 3)$ and the port numbering induced by δ: a covering of the triangle (T, δ) with 3 sheets.

In the sequel, we use a particular case of this construction. Let G be a connected graph. Let e be an edge of G. Let G^e be the subgraph of G obtained by deleting the edge e. We denote by $R(G, e, \alpha)$ the graph constructed as: make α copies of G^e denoted G^e_i, $1 \leq i \leq \alpha$; for each i such that $1 \leq i \leq \alpha$ and $i + 1$ computed modulo α, connect the first endpoint of e in G^e_i to the second endpoint of e in G^e_{i+1} (see Figures 2 and 3). This construction is extended in a natural way to graphs with a port numbering. From this construction, by considering a sequence of consecutive copies and analysing what is produced in the middle in such a sequence, we can state the following lower bound:

Proposition 4.1. *Let G be a non-covering-prime graph having n_G vertices and at least one cycle. Let δ be a port numbering on G. Let e be an edge of G such that G^e, the graph obtained by deleting the edge e, is connected. Let α be a non-negative integer. Let $n = \alpha n_G$. Let δ' be the port numbering induced by δ on $R(G, e, \alpha)$. Then there exists a constant $c > 0$ such that, for any Las Vegas distributed algorithm $\mathcal{A}$ that uses messages of 1 bit, there is at least one copy of*

G^e of $R(G, e, \alpha)$ such that for each fibre all its vertices are in the same state for at least $c\sqrt{\log n}$ rounds w.h.p.

By applying Proposition 4.1 to the triangle ($G = T$) as is explained by Figures 2 and 3, we obtain:

Corollary 4.2. *For every Las Vegas distributed algorithm $\mathcal{A}$ there is an infinite family of rings $\mathcal{R}$ such that $\mathcal{A}$ has a bit complexity $\Omega(\sqrt{\log n})$ w.h.p. on $\mathcal{R}$ to compute a MIS or a maximal matching.*

5 Obtaining Lower Bounds of the Form $\Omega(\log n)$ for Connected Graphs

Let $G = (V, E)$ be a graph such that $D = Dir(G)$ is not covering prime. Let D' be a symmetric digraph such that D is a covering of D' via the homomorphism φ and D is not isomorphic to D'. Let F be a fibre of D, and let w be the corresponding vertex of D', i.e., $F = \varphi^{-1}(w)$. Let b be the size of the fibre $F = \{f_1, \ldots, f_b\}$. As D is not isomorphic to D' the size b of F is greater than or equal to 2. We can note also that F is a subset of the set of vertices of G. Let α be a positive integer. We denote by H the graph formed by α copies of G. Let u be a vertex which does not belong to H. Now we define the graph K obtained from H by adding a new vertex, the vertex u, and by adding an edge between u and all vertices of all copies of the fibre F. Let n be the number of vertices of K, we have: $n = \alpha n_G + 1$. We consider a port numbering of D'; it induces via φ^{-1} a port numbering on D thus on G and on H. Let d be the degree of any vertex of the fibre F. We assign to any port corresponding to u incident to any vertex of any copy of the fibre F the number $d + 1$. The numbers of the ports incident to u are chosen randomly, uniformly among the permutations of $[1, \alpha \cdot b]$. Let $\mathcal{A}$ be a Las Vegas distributed algorithm. The aim of this section is to prove that if $\mathcal{A}$ halts on K w.h.p. in a time less than $c \log n$, it has a high probability of giving incorrect output. It follows from this that there exists some port numbering for which the algorithm does not compute a correct result in time less than or equal to $c \log n$ w.h.p. Let $D'_{w'}$ be the graph obtained by adding a new vertex, denoted w', to the graph D' and by adding an edge between w' and w (the image of the fibre F by φ). Let D_F be the graph we obtain by adding b new vertices f'_i, $1 \leq i \leq b$, to D and by adding an edge between f_i and f'_i ($1 \leq i \leq b$). In the same manner we define G_F. It is easy to verify that D_F is a covering of $D'_{w'}$ via the morphism γ defined by: the restriction of γ to D is equal to φ and $\gamma(f'_i) = w'$ (for $1 \leq i \leq b$). An execution $(\mathbf{D}'_{w'i})_{0 \leq i \leq \ell}$ of $\mathcal{A}$ on $\mathbf{D}'_{w'}$ (with $\mathbf{D}'_{w'0} = \mathbf{D}'_{w'}$) induces an execution $(\mathbf{D}_{Fi})_{0 \leq i \leq \ell}$ of $\mathcal{A}$ on $\mathbf{D}_F$ (with $\mathbf{D}_{F0} = \mathbf{D}_F$) via γ. It induces also an execution on G_F. For this kind of execution, vertices of D_F, thus of G_F, which belong to the same fibre are in the same state at each step. Let D_f and G_f be the graphs obtained from D_F and G_F by fusing vertices f'_1,...f'_b into the same vertex, denoted f. Thus in D_f and G_f there is an edge between vertices f_1,...,f_b and f. Now we consider an execution of $\mathcal{A}$ on D_f (thus on G_f); if we assume that this execution is induced by an execution of $\mathcal{A}$ on $D'_{w'}$

for vertices different from f and the vertex f sends the same bit to every vertex of F then vertices of a given fibre of D_f (thus of G_f) are in the same state after each step of the execution. Finally, steps occur as if the execution is induced by a covering relation. By definition, an execution on D_f thus on G_f which satisfies this property is called uniform and, by extension, D_f is uniform. By choosing a constant $c_1 > 0$ depending on G, we show that after r rounds of $\mathcal{A}$, there is a uniform set of copies of G in K, denoted U_r of size nc_1^r. Choosing a suitable c such that the random variable $|U_{c \log n}| = \Omega(n^{1/2})$ gives the following result:

Proposition 5.1. *Let $\mathcal{A}$ be a Las Vegas distributed algorithm that produces at each round on each port the bit* 1 *or the bit* 0*. Let G be a connected graph having n_G vertices. Assume $Dir(G)$ is not symmetric covering prime. Let α be a non-negative integer. Let K be the connected graph defined above (a vertex added to α copies of G) and with the associated port numbering. Let $n = \alpha n_G$. There exists a constant $c > 0$ such that for at least $c \log n$ rounds of execution of Algorithm $\mathcal{A}$ on K, with high probability, there is a copy of G in K for which the execution is uniform.*

Corollary 5.2. *For every Las Vegas distributed algorithm $\mathcal{A}$ there is an infinite family $\mathcal{C}$ of connected graphs such that $\mathcal{A}$ has a bit complexity $\Omega(\log n)$ w.h.p. on $\mathcal{C}$ to solve either: the colouring problem, the MIS problem, the maximal matching problem, the* 2*-MIS problem or the distance-*2 *colouring problem.*

References

[Ang80] Angluin, D.: Local and global properties in networks of processors. In: Proceedings of the 12th Symposium on Theory of Computing, pp. 82–93 (1980)

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

[Bod89] Bodlaender, H.-L.: The classification of coverings of processor networks. J. Parallel Distrib. Comput. 6, 166–182 (1989)

[BV02] Boldi, P., Vigna, S.: Fibrations of graphs. Discrete Math. 243, 21–66 (2002)

[CM07] Chalopin, J., Métivier, Y.: An efficient message passing election algorithm based on Mazurkiewicz's algorithm. Fundam. Inform. 80(1-3), 221–246 (2007)

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

[FMRZar] Fontaine, A., Métivier, Y., Robson, J.M., Zemmari, A.: The bit complexity of the MIS problem and of the maximal matching problem in anonymous rings. Information and Computation (to appear)

[II86] Israeli, A., Itai, A.: A fast and simple randomized parallel algorithm for maximal matching. Information Processing Letters 22, 77–80 (1986)

[KN99] Kushilevitz, E., Nisan, N.: Communication complexity. Cambridge University Press (1999)

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

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

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

[Rei32] Reidemeister, K.: Einführung in die Kombinatorische Topologie. Vieweg, Brunswick (1932)

[Tel00] Tel, G.: Introduction to distributed algorithms. Cambridge University Press (2000)

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