

# Exploration of the $T$ -Interval-Connected Dynamic Graphs: the Case of the Ring

David Ilcinkas\* and Ahmed Mouhamadou Wade\*

LaBRI, CNRS & Bordeaux University  
{ilcinkas,wade}@labri.fr

**Abstract.** In this paper, we study the  $T$ -interval-connected dynamic graphs from the point of view of the time necessary and sufficient for their exploration by a mobile entity (agent). A dynamic graph (more precisely, an evolving graph) is  $T$ -interval-connected ( $T \geq 1$ ) if, for every window of  $T$  consecutive time steps, there exists a connected spanning subgraph that is stable (always present) during this period. This property of connection stability over time was introduced by Kuhn, Lynch and Oshman [6] (STOC 2010). We focus on the case when the underlying graph is a ring of size  $n$ , and we show that the worst-case time complexity for the exploration problem is  $2n - T - \Theta(1)$  time units if the agent knows the dynamics of the graph, and  $n + \frac{n}{\max\{1, T-1\}}(\delta - 1) \pm \Theta(\delta)$  time units otherwise, where  $\delta$  is the maximum time between two successive appearances of an edge.

**Keywords:** Exploration, Dynamic graphs, Mobile agent, T-interval-connectivity

## 1 Introduction

Partly due to the very important increase of the number of communicating objects that we observe today, the distributed computing systems are becoming more and more dynamic. The computational models for static networks are clearly not sufficient anymore to capture the behavior of these new communication networks. In fact, even the computational models that take into account a certain degree of fault tolerance become insufficient for some very dynamic networks. Indeed, the classical models of fault tolerance either assume that the frequency of fault occurrences is small, which gives enough time to the algorithm to adapt to the changes, or that the system stabilizes after a certain amount of time (as in the self-stabilizing systems for example). Therefore, in the last decade or so, many more or less equivalent models have been developed that take into account the extreme dynamism of some communication networks. An interested reader will find in [1] a very complete overview of the different models and studies of dynamic graphs (see also [7]).

One of the first developed models, and also one of the most standard, is the model of evolving graphs [3]. To simplify, given a static graph  $G$ , called

---

\* Partially supported by the ANR project DISPLEXITY (ANR-11-BS02-014), the INRIA project CEPAGE, and the European project EULER.

the underlying graph, an evolving graph based on  $G$  is a (possibly infinite) sequence of spanning but not necessarily connected subgraphs of  $G$  (see Section 2 for precise definitions). This model is particularly well adapted for modeling dynamic synchronous networks.

In all its generality, the model of evolving graphs allows to consider an extremely varied set of dynamic networks. Therefore, to obtain interesting results, it is often required to make assumptions that reduce the possibilities of dynamic graphs generated by the model. One example is the assumption of connectivity over time, which states that there is a journey (path over time) from any vertex to any other vertex. Another example is the assumption of constant connectivity, for which the graph must be connected at all times. This latter assumption, which is very usual, has been recently generalized in a paper by Kuhn, Lynch and Oshman [6] by the notion of  $T$ -interval-connectivity. Roughly speaking, given an integer  $T \geq 1$ , a dynamic graph is  $T$ -interval-connected if, for any window of  $T$  consecutive time steps, there exists a connected spanning subgraph which is stable throughout the period. (The notion of constant connectivity is thus equivalent to the notion of 1-interval-connectivity). This new notion, which captures the connection stability over time, allows the finding of interesting results: the  $T$ -interval-connectivity allows to reduce by a factor of about  $\Theta(T)$  the number of messages that is necessary and sufficient to perform a complete exchange of information between all the vertices [2, 6] (gossip problem).

In this paper, we carry on the study of these  $T$ -interval-connected dynamic graphs by considering the problem of exploration. A mobile entity (called agent), moving from node to node along the edges of a dynamic graph, must traverse/visit each of its vertices at least once (the traversal of an edge takes one time unit). This fundamental problem in distributed computing by mobile agents has been widely studied in static graphs since the seminal paper by Claude Shannon [8]. As far as highly dynamic graphs are concerned, only the case of periodically-varying graphs has been studied [4, 5]. We focus here on the (worst-case) time complexity of this problem, namely the number of time units used by the agent to solve the problem in the  $T$ -interval-connected dynamic graphs. The problem of exploration, in addition to its theoretical interests, can be applied for instance to the network maintenance, where a mobile agent has to control the proper functioning of each vertex of the graph.

We consider the problem in two scenarios. In the first one, the agent knows entirely and exactly the dynamic graph it has to explore. This situation corresponds to predictable dynamic networks such as transportation networks for example. In the second scenario, the agent does not know the dynamics of the graph, that is the times of appearance and disappearance of the edges. This case typically corresponds to networks whose changes are related to frequent and unpredictable failures. In this second scenario, Kuhn, Lynch and Oshman [6] noted that the exploration problem is impossible to solve under the single assumption of 1-interval-connectivity. In fact, it is quite easy to convince oneself that by adding the assumption that each edge of the underlying graph must appear infinitely often, the exploration problem becomes possible, but the time

complexity remains unbounded. In this article, and only for the second scenario, we therefore add the assumption of  $\delta$ -recurrence, for some integer  $\delta \geq 1$ : each edge of the underlying graph appears at least once every  $\delta$  time units.

It turns out that the problem of exploration is much more complex in dynamic graphs than in static graphs. Indeed, let us consider for example the first scenario (known dynamic graph). The worst-case exploration time of  $n$ -node static graphs is clearly in  $\Theta(n)$  (worst case  $2n - 3$ ). On the other hand, the worst-case exploration time of  $n$ -node (1-interval-connected) dynamic graphs remains largely unknown. No lower bound better than the static bound is known, while the best known upper bound is quadratic, and directly follows from the fact that the temporal diameter of these graphs is bounded by  $n$ . Therefore, we focus here on the study of  $T$ -interval-connected dynamic graphs whose underlying graph is a ring. Note that, in this particular case, the  $T$ -interval-connectivity property, for  $T \geq 1$ , implies that at most one edge can be absent at a given time.

*Our results.* We determine in this paper the exact time complexity of the exploration problem for the  $n$ -node  $T$ -interval-connected dynamic graphs based on the ring, when the agent knows the dynamics of the graph. This is essentially  $2n - T - 1$  time units (see Section 3 for details). When the agent does not know the dynamics of the graph, we add the assumption of  $\delta$ -recurrence, and we show that the complexity increases to  $n + \frac{n}{\max\{1, T-1\}}(\delta - 1) \pm \Theta(\delta)$  time units (see Section 4 for details).

## 2 Model and definitions

This section gives the precise definitions of the concepts and models informally mentioned in the introduction. Some definitions are similar or even identical to the definitions given in [6].

**Definition 1 (Dynamic graph).** A dynamic graph is a pair  $\mathcal{G} = (V, \mathcal{E})$ , where  $V$  is a static set of  $n$  vertices, and  $\mathcal{E}$  is a function which maps to every integer  $i \geq 0$  a set  $\mathcal{E}(i)$  of undirected edges on  $V$ .

**Definition 2 (Underlying graph).** Given a dynamic graph  $\mathcal{G} = (V, \mathcal{E})$ , the static graph  $G = (V, \bigcup_{i=0}^{\infty} \mathcal{E}(i))$  is called the underlying graph of  $\mathcal{G}$ . Conversely, the dynamic graph  $\mathcal{G}$  is said to be based on the static graph  $G$ .

In this article, we consider the dynamic graphs based on the  $n$ -node ring, denoted  $C_n$ .

**Definition 3 ( $T$ -interval-connectivity).** A dynamic graph  $\mathcal{G} = (V, \mathcal{E})$  is  $T$ -interval-connected, for an integer  $T \geq 1$ , if for every integer  $i \geq 0$ , the static graph  $G_{[i, i+T[} = (V, \bigcap_{j=i}^{i+T-1} \mathcal{E}(j))$  is connected.

**Definition 4 ( $\delta$ -recurrence).** A dynamic graph is  $\delta$ -recurrent if every edge of the underlying graph is present at least once every  $\delta$  time steps.

A mobile entity, called *agent*, operates on these dynamic graphs. The agent can traverse at most one edge per time unit. It may also stay at the current node (typically to wait for an incident edge to appear). We say that an agent *explores* the dynamic graph if and only if it visits all the nodes.

### 3 The agent knows the dynamics of the graph

In this section, we assume that the agent perfectly knows the dynamic graph to be explored.

#### 3.1 Upper bound

The following theorem shows that the worst-case exploration time is actually small, bounded by  $2n$ , when the underlying graph is a ring. Furthermore, it shows that the agent can benefit from the  $T$ -interval-connectivity to spare an additive term  $T$ . Note that our upper bound is constructive.

Before proceeding with the formal theorem and its proof, let us informally describe the key ingredients of the proof of the most general case.

We consider two algorithms, being the algorithms always going in the clockwise, resp. counter-clockwise, direction, traversing edges as soon as the dynamic graph allows it. At the beginning of the process, the agents try to traverse distinct edges and thus, at each time step, at least one of them progresses. During this phase, the average speed of the two agents is thus  $1/2$  (edge traversals per time unit). However, when the agents are about to meet each other (thus after time at most  $n$ ), their progression can be stopped by the absence of a unique edge  $e$ .

If this edge  $e$  is absent for at least  $n - 1$  time steps, then any agent has enough time to change its direction and to explore all the nodes of the graph in the other direction, hence completing exploration within  $2n$  steps.

If the edge  $e$  does not stay absent long enough and reappears at time  $t$ , we modify the two algorithms as follows. The agent previously progressing in the clockwise, resp. counter-clockwise, direction, starts now by exploring the ring in the opposite direction, before going back in the usual direction the latest possible so that it reaches the edge  $e$  at time at most  $t$ . At time  $t$ , the two modified algorithms cross each other, and then continue their progression in their usual direction until one of them terminates the exploration. Note that, after time  $t$ , we have again the property that, at each time step, at least one agent progresses.

Globally, except during the period when  $e$  is absent, the average speed of the two agents is  $1/2$ . Besides, the modification of the algorithms allows each of the agent to explore an additional part of the ring. Unfortunately, these parts of the ring are traversed twice instead of once. Nevertheless, intuitively, the speed of both the modified agents is 1 during the period when  $e$  is absent. This compensates the loss induced by traversing twice some parts of the ring. Overall,

the average speed is thus globally of at least  $1/2$ , which implies that at least one of the two modified agents performs exploration within time  $2n$ .

When the dynamic graph is  $T$ -interval connected, all edges must be present during  $T - 1$  steps between the removal of two different edges. This fact is used to gain an additive term of  $T - 1$  on the exploration time, yielding to a time of roughly  $2n - T$ . A much more precise analysis of the modified algorithms allows us to obtain the exact claimed bounds.

**Theorem 1.** *For every integers  $n \geq 3$  and  $T \geq 1$ , and for every  $T$ -interval-connected dynamic graph based on  $C_n$ , there exists an agent (algorithm) exploring this dynamic graph in time at most*

$$\begin{cases} 2n - 3 & \text{if } T = 1 \\ 2n - T - 1 & \text{if } 2 \leq T < (n + 1)/2 \\ \lfloor \frac{3(n-1)}{2} \rfloor & \text{if } T \geq (n + 1)/2 \end{cases}$$

*Proof.* Fix  $n \geq 3$  and an arbitrary dynamic graph based on the ring  $C_n$ . Let  $v_0, v_1, \dots, v_{n-1}$  be the vertices of  $C_n$  in clockwise order. Assume that the agent starts exploration from  $v_0$  at time 0. In order to prove this theorem, we will describe various algorithms, and we will show that at least one of them will allow the agent to perform exploration within the claimed time bound. Let  $\mathcal{T}$  be this bound.

First assume that at most one edge  $e$  is absent during the time interval  $[0, \mathcal{T})$ . Then, an agent going to the closest extremity of  $e$  and then changing direction will explore all nodes of the ring in time at most  $3(n - 1)/2 \leq \mathcal{T}$ . So let us assume from now on that at least two different edges are absent at least once each during the time interval  $[0, \mathcal{T})$ .

Before proceeding with the rest of the proof, we introduce the following notations. Given a time interval  $I$  and two algorithms  $A$  and  $B$ , let  $d_A^I$  be the number of edge traversals performed by agent  $A$  during the time interval  $I$ , let  $\alpha_A^I$ , resp.  $\alpha_{A,B}^I$ , be the number of time steps in  $I$  for which agent  $A$ , resp. both agents  $A$  and  $B$ , do(es) not move. Note that it never helps to wait at a node when all its incident edges are present. Hence, without loss of generality, an agent always stays at a node because of the absence of an incident edge. Finally, let  $\beta^I$  be the number of time steps in  $I$  for which no edges are absent.

Let us now consider two simple algorithms.  $L$ , respectively  $R$ , is the algorithm always going in the clockwise, resp. counter-clockwise, direction, traversing edges as soon as the dynamic graph allows it. Now consider the sum of the number of edges traversed by each of the two algorithms until some time  $t$ . Since only one edge can be absent at a given time, this sum increases by at least one (and obviously by at most two) at each time step, until this sum is larger or equal to  $n - 1$ . So let  $e$  be the unique unexplored edge when this sum reaches  $n - 1$ . If the sum jumps directly from  $n - 2$  to  $n$ , then fix  $e$  to be any of the last two unexplored edges. In both cases, let  $t_1$  be the first time one of the two agents reaches one extremity of  $e$ . We consider two cases.

**Case 1.** The edge  $e$  is absent during the whole interval  $[t_1, t_1 + n - 1)$ .

In this case, the first agent to reach an extremity of  $e$ , at time  $t_1$ , goes back in the opposite direction and explores the ring in  $n - 1$  further steps.

This gives an exploration time of at most  $t_1 + n - 1$ . Let  $I_1 = [0, t_1)$ . We have

$$t_1 = \begin{cases} d_L^{I_1} + \alpha_L^{I_1} & (1) \\ d_R^{I_1} + \alpha_R^{I_1} & (2) \end{cases}$$

and, since  $L$  and  $R$  are always trying to traverse distinct edges during  $I_1$  and at most one edge may be removed at any time, we also have

$$\alpha_L^{I_1} + \alpha_R^{I_1} + \beta^{I_1} \leq t_1 \quad (3)$$

$$\text{Besides, we have } d_L^{I_1} + d_R^{I_1} \leq n - 1 \quad (4)$$

and since there are at least two removed different edges during the whole interval  $[0, t_1 + n - 1)$ , we have

$$\beta^{I_1} \geq T - 1 \quad (5)$$

$$(1)+(2)+(3)+(4)+(5) \rightarrow t_1 + n - 1 \leq 2n - T - 1.$$

For  $T = 1$ , this bound is one unit larger than the claimed bound. If the inequality (4) is in fact strict, then the correct bound is obtained. Otherwise, it means that at time  $t_1 - 1$ , both agents were free to move. This implies that either  $\beta^{I_1} \geq 1$  or that the inequality (3) is strict. In both cases, this also gives the correct bound.

**Case 2.** The edge  $e$  is not absent during the whole interval  $[t_1, t_1 + n - 1)$ .

Then let  $t_2$  be the smallest time  $t \geq t_1$  such that the edge  $e$  is present at time  $t$ . We define two new algorithms, one of which will explore the dynamic graph within  $\mathcal{T}$ .

Let  $L'$  be the algorithm that is equal to  $L$  until some time  $t$ , at which  $L'$  goes back in the other direction forever. More precisely,  $L'$  is the algorithm for which  $t$  is the largest possible value such that  $L'$  arrives at the extremity of  $e$  at time at most  $t_2$ . Similarly, let  $R'$  be the algorithm that is equal to  $R$  until some time  $t$ , at which  $R'$  goes back in the other direction forever. More precisely,  $R'$  is the algorithm for which  $t$  is the largest possible value such that  $R'$  arrives at the extremity of  $e$  at time at most  $t_2$ . Let  $\mathcal{T}_{exp}$  be the exploration time of the first between  $L'$  and  $R'$  exploring the dynamic graph.

In order to analyze the algorithms  $L'$  and  $R'$ , we introduce two other algorithms. Let  $L''$ , respectively  $R''$  be the algorithm defined as  $L'$ , resp.  $R'$ , but turning back exactly one time unit later than  $L'$ , resp.  $R'$ .

Let  $I_1 = [0, t_1)$ ,  $I_2 = [t_1, t_2)$ ,  $I_{1,2} = [0, t_2)$ ,  $I_3 = [t_2, \mathcal{T}_{exp})$ , and  $I = [0, \mathcal{T}_{exp})$ .

On  $I_1$ , we have

$$t_1 \geq \alpha_{L''}^{I_1} + \alpha_{R''}^{I_1} - \alpha_{L'',R''}^{I_1} + \beta^{I_1} \quad (1)$$

$$t_1 \geq \alpha_L^{I_1} + \alpha_R^{I_1} + \alpha_{L'',R''}^{I_1} + \beta^{I_1} \quad (2)$$

As in the first case, we have

$$t_1 = \begin{cases} d_L^{I_1} + \alpha_L^{I_1} & (3) \\ d_R^{I_1} + \alpha_R^{I_1} & (4) \end{cases}$$

$$(1)+(2)+(3)+(4) \rightarrow \alpha_{L''}^{I_1} + \alpha_{R''}^{I_1} + 2\beta^{I_1} \leq d_L^{I_1} + d_R^{I_1} \quad (5)$$

On  $I_{1,2}$ , we have

$$t_1+t_2 = \begin{cases} d_{L''}^{I_{1,2}} + \alpha_{L''}^{I_{1,2}} & (6) \\ d_{R''}^{I_{1,2}} + \alpha_{R''}^{I_{1,2}} & (7) \end{cases}$$

Note that, by definition of  $L''$  and  $R''$

$$d_{L''}^{I_{1,2}} \leq d_{L'}^{I_{1,2}} + 1 \quad (8)$$

$$d_{R''}^{I_{1,2}} \leq d_{R'}^{I_{1,2}} + 1 \quad (9)$$

$$(6)+(7)+(8)+(9) \rightarrow 2(t_1+t_2) \leq d_{L'}^{I_{1,2}} + d_{R'}^{I_{1,2}} + \alpha_{L'}^{I_{1,2}} + \alpha_{R'}^{I_{1,2}} + 2 \quad (10)$$

Note that on  $I_2$   $L''$  and  $R''$  are not blocked because the edge  $e$  is absent during this interval. Hence

$$\alpha_{L''}^{I_{1,2}} = \alpha_{L''}^{I_1} \quad (11)$$

$$\alpha_{R''}^{I_{1,2}} = \alpha_{R''}^{I_1} \quad (12)$$

$$(5)+(10)+(11)+(12) \rightarrow 2(t_1+t_2) + 2\beta^{I_1} \leq d_L^{I_1} + d_R^{I_1} + d_L^{I_{1,2}} + d_R^{I_{1,2}} + 2 \quad (13)$$

On  $I_3$ , we have

$$\mathcal{T}_{exp} - (t_1+t_2) \geq \alpha_{L'}^{I_3} + \alpha_{R'}^{I_3} + \beta^{I_3} \quad (14)$$

and

$$\mathcal{T}_{exp} - (t_1+t_2) = d_{L'}^{I_3} + \alpha_{L'}^{I_3} \quad (15)$$

$$\mathcal{T}_{exp} - (t_1+t_2) = d_{R'}^{I_3} + \alpha_{R'}^{I_3} \quad (16)$$

$$(14)+(15)+(16) \rightarrow \mathcal{T}_{exp} - (t_1+t_2) + \beta^{I_3} \leq d_{L'}^{I_3} + d_{R'}^{I_3} \quad (17)$$

$$(17)+\frac{1}{2}(13) \rightarrow \mathcal{T}_{exp} + \beta^{I_1} + \beta^{I_3} \leq \frac{1}{2}(d_L^{I_1} + d_R^{I_1} + d_{L'}^{I_{1,2}} + d_{R'}^{I_{1,2}}) + d_{L'}^{I_3} + d_{R'}^{I_3} + 1 \quad (18)$$

Note that  $\beta^{I_1} + \beta^{I_3} = \beta^I$

Let  $x$ , resp.  $y$ , be the number of edges traversed by  $L'$ , resp.  $R'$ , before turning back. Then

$$d_{R'}^{I_{1,2}} = 2x + d_L^{I_1} \quad (19)$$

$$d_{L'}^{I_{1,2}} = 2y + d_R^{I_1} \quad (20)$$

$$d_{R'}^{I_3} = d_L^{I_1} - x \quad (21)$$

$$d_{L'}^{I_3} = d_R^{I_1} - x \quad (22)$$

$$d_L^{I_1} + d_R^{I_1} \leq n - 1 \quad (23)$$

and since there are at least two removed different edges during the interval  $\beta^I \geq T - 1$  (24)

Finally, we get the sought result

$$(18)+(19)+(20)+(21)+(22)+(23)+(24) \rightarrow \mathcal{T}_{exp} \leq 2n - T - 1.$$

One can again argue similarly as in the first case to gain one time unit in the case  $T = 1$ , which concludes the proof.  $\square$

### 3.2 Lower bound

We now prove that the precise bound given in Section 3.1 is actually the exact worst-case time complexity of the exploration problem.

**Theorem 2.** *For every integers  $n \geq 3$  and  $T \geq 1$ , there exists a  $T$ -interval-connected dynamic graph based on  $C_n$  such that any agent (algorithm) needs at*

least

$$\begin{cases} 2n - 3 & \text{if } T = 1 \\ 2n - T - 1 & \text{if } 2 \leq T < (n + 1)/2 \\ \lfloor \frac{3(n-1)}{2} \rfloor & \text{if } T \geq (n + 1)/2 \end{cases}$$

time units to explore it.

*Proof.* For any integers  $n \geq 3$ , and  $2 \leq T \leq \lceil (n + 1)/2 \rceil$ , we define a  $T$ -interval-connected dynamic graph  $\mathcal{G}_{n,T}$  based on  $C_n$ . Let  $v_0, v_1, \dots, v_{n-1}$  be the vertices of  $C_n$  in clockwise order. Assume that the exploration starts from  $v_0$  at time 0. In  $\mathcal{G}_{n,T}$ , the edge  $\{v_0, v_1\}$ , respectively  $\{v_{T-1}, v_T\}$ , is absent in the time interval  $[0, n - 2T + 1)$ , respectively  $[n - T, 2n)$ . Note that this dynamic graph is indeed  $T$ -interval-connected.

Consider any agent (algorithm). We will now prove that the time it uses to explore  $\mathcal{G}_{n,T}$  is at least  $2n - T - 1$ . Since the agent must explore all vertices, it must in particular explore both  $v_{T-1}$  and  $v_T$ . We consider two cases.

**Case 1.**  $v_{T-1}$  is explored before  $v_T$ .

To visit  $v_{T-1}$  without going through  $v_T$ , the agent must traverse the edge  $\{v_0, v_1\}$ . By construction, this edge is absent until time  $n - 2T + 1$ . Moreover, the length of the path between  $v_0$  and  $v_{T-1}$  without going through  $v_T$  is  $T - 1$ . Thus the agent needs at least  $n - T$  time units to reach  $v_{T-1}$  for the first time. Since the edge  $\{v_{T-1}, v_T\}$  is absent in the time interval  $[n - T, 2n)$ , the fastest way of reaching  $v_T$  is to traverse the whole ring through  $v_0$ , inducing  $n - 1$  additional time units. So in this first case, the agent needs at least  $2n - T - 1$  time units to explore  $\mathcal{G}_{n,T}$ .

**Case 2.**  $v_T$  is explored before  $v_{T-1}$ .

To visit  $v_T$  without going through  $v_{T-1}$ , the agent must use the path  $v_0, v_{n-1}$ , up to  $v_T$ , which is of length  $n - T$ . When at node  $v_T$ , and since the edge  $\{v_{T-1}, v_T\}$  is absent in the time interval  $[n - T, 2n)$ , the fastest way of reaching  $v_{T-1}$  is to traverse the whole ring through  $v_0$ , inducing  $n - 1$  additional time units. Thus also in the second case, the agent needs at least  $2n - T - 1$  time units to explore  $\mathcal{G}_{n,T}$ .

This proves the theorem for values of  $T$  in  $[2, \lceil (n + 1)/2 \rceil]$ . In fact, this also proves the theorem for  $T = 1$  because  $\mathcal{G}_{n,2}$  is obviously also 1-interval-connected, and the claimed bound is the same for  $T = 1$  and  $T = 2$ . Besides, note that only one edge is ever removed in  $\mathcal{G}_{n, \lceil (n+1)/2 \rceil}$ . This dynamic graph is therefore 1-interval-connected for any  $T$ , and thus the theorem is also proved for values of  $T$  larger than  $(n + 1)/2$ .  $\square$

## 4 The agent does not know the dynamics of the graph

In this section, we assume that the agent does not know the dynamics of the graph, i.e., it does not know the times of appearance and disappearance of the edges. As explained in the introduction, we assume here the  $\delta$ -recurrence property, for a given  $\delta \geq 1$ , in order for the problem to be solvable in bounded time.



## 4.1 Upper bound

We first prove that there exists a very simple algorithm that is able to explore all the  $\delta$ -recurrent  $T$ -interval-connected dynamic graphs based on the ring. This algorithm consists in moving as much and as soon as possible in a fixed arbitrary direction, see Algorithm 1.

---

### Algorithm 1 STUBBORN-TRAVERSAL( $\mathit{dir}$ )

---

**Input:** a direction  $\mathit{dir}$

```

for each time step do
  if the edge in the  $\mathit{dir}$  direction is present then
    traverse it
  else
    wait
  end if
end for

```

---

**Theorem 3.** *For every integers  $n \geq 3$ ,  $T \geq 1$  and  $\delta \geq 1$ , and for any direction  $\mathit{dir}$ , Algorithm STUBBORN-TRAVERSAL( $\mathit{dir}$ ) explores any  $\delta$ -recurrent  $T$ -interval-connected dynamic graph based on  $C_n$  in time at most*

$$n - 1 + \left\lceil \frac{n - 1}{\max\{1, T - 1\}} \right\rceil (\delta - 1).$$

*Proof.* Fix an arbitrary direction  $\mathit{dir}$  and let us analyze the algorithm STUBBORN-TRAVERSAL( $\mathit{dir}$ ). Note first that it will complete exploration after traversing exactly  $n - 1$  edges. To bound its exploration time, it thus remains to bound the number of time steps when the agent cannot move.

Since the dynamic graph is  $\delta$ -recurrent, an edge cannot be absent for more than  $\delta - 1$  consecutive time steps. Furthermore, since the dynamic graph is  $T$ -interval-connected, two time steps in which two different edges are absent must be separated by at least  $T - 1$  time steps in which all edges are present. Therefore, the agent can traverse at least  $\max\{1, T - 1\}$  edges between two consecutive blocks at different nodes. To summarize, the agent can be blocked at most  $\left\lceil \frac{n-1}{\max\{1, T-1\}} \right\rceil$  times during at most  $\delta - 1$  time steps.

Putting everything together, the agent will perform edge traversals for  $n - 1$  time steps and will wait for at most  $\left\lceil \frac{n-1}{\max\{1, T-1\}} \right\rceil (\delta - 1)$  time steps, which gives the claimed bound.  $\square$

## 4.2 Lower bound

It turns out that the simple and natural Algorithm 1, described and analyzed in Section 4.1, is almost optimal, up to an additive term proportional to  $\delta$ .

**Theorem 4.** For every integers  $n \geq 3$ ,  $T \geq 1$ , and  $\delta \geq 1$ , and for every agent (algorithm), there exists a  $\delta$ -recurrent  $T$ -interval-connected dynamic graph based on  $C_n$  such that this agent needs at least

$$n - 1 + \left\lfloor \frac{n - 3}{\max\{1, T - 1\}} \right\rfloor (\delta - 1)$$

time units to explore it.

This result holds even if the agent knows  $n$ ,  $T$  and  $\delta$ .

*Proof.* Let  $n \geq 3$ ,  $T \geq 1$ , and  $\delta \geq 1$ . Fix an arbitrary agent (algorithm)  $A$ . We construct as follows the  $\delta$ -recurrent  $T$ -interval-connected dynamic graph  $\mathcal{G}_{n,T,\delta}(A)$  based on  $C_n$  that this agent will fail to explore in less than the claimed bound.

Let  $v_0, v_1, \dots, v_{n-1}$  be the vertices of  $C_n$  in clockwise order. Assume that the agent starts exploration from  $v_0$  at time 0. For any integer  $1 \leq i \leq n - 1$ , if the node  $v_i$  is explored by going from  $v_0$  in the counter-clockwise direction, then node  $v_i$  is denoted  $v_{i-n}$ . Finally, let  $\tilde{T} = \max\{1, T - 1\}$ .

In the dynamic graph  $\mathcal{G}_{n,T,\delta}(A)$ , only the edges  $\{v_{\tilde{T}+1}, v_{\tilde{T}+2}\}$ ,  $\{v_{2\tilde{T}+1}, v_{2\tilde{T}+2}\}$ , and so on, and  $\{v_0, v_{-1}\}$ ,  $\{v_{-\tilde{T}}, v_{-\tilde{T}-1}\}$ ,  $\{v_{-2\tilde{T}}, v_{-2\tilde{T}-1}\}$ , and so on, may be absent. The actual times of appearance and disappearance of these edges depend on the algorithm  $A$ . For any integer  $i \geq 0$ , each time the agent arrives at node  $v_{-i\tilde{T}}$  in the counter-clockwise direction, the edge  $\{v_{-i\tilde{T}}, v_{-i\tilde{T}-1}\}$  is removed until either the  $\delta$ -recurrence forces the edge to reappear or the agent leaves the node  $v_{-i\tilde{T}}$  to go on  $v_{-i\tilde{T}+1}$ . Similarly, for any integer  $i \geq 1$ , each time the agent arrives at node  $v_{i\tilde{T}+1}$  in the clockwise direction, the edge  $\{v_{i\tilde{T}+1}, v_{i\tilde{T}+2}\}$  is removed until either the  $\delta$ -recurrence forces the edge to reappear or the agent leaves the node  $v_{i\tilde{T}+1}$  to go on  $v_{i\tilde{T}}$ . Note that between two time steps with two different absent edges, there are at least  $T - 1$  time steps for which no edges are absent. The dynamic graph is therefore  $T$ -interval-connected. It is also  $\delta$ -recurrent by construction.

By definition of the dynamics of the graph, the agent needs to wait  $\delta - 1$  time units to go from  $v_{-i\tilde{T}}$  to  $v_{-i\tilde{T}-1}$ , for  $i \geq 0$ , or to go from  $v_{i\tilde{T}+1}$  to  $v_{i\tilde{T}+2}$ , for  $i \geq 1$ . To explore all the vertices, the agent needs to perform at least  $\left\lfloor \frac{n-3}{\max\{1, T-1\}} \right\rfloor$  such traversals. The waiting time of the agent is thus at least  $\left\lfloor \frac{n-3}{\max\{1, T-1\}} \right\rfloor (\delta - 1)$ . Since the agent needs also at least  $n - 1$  time units to traverse enough edges so that all vertices are explored, we obtain the claimed bound.  $\square$

## 5 Conclusion

We studied in this paper the problem of exploration of the  $T$ -interval-connected dynamic graphs based on the ring in two scenarios, when the agent is specific to the dynamic graph, and when the agent does not know the dynamics of the graph. The next objective is obviously to extend these results to larger families of underlying graphs. Unfortunately, this problem is much more difficult than it

seems: proving that any dynamic graph based on a tree of cycles (a cactus) can be explored in time  $O(n)$  is already a challenging open problem.

## References

1. A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, volume 27(5), 2012.
2. C. Dutta, G. Pandurangan, R. Rajaraman, and Z. Sun. Information spreading in dynamic networks. CoRR, abs/1112.0384, 2011.
3. A. Ferreira, Building a reference combinatorial model for MANETs. *Network*, IEEE, volume 18(5), pages 24–29, 2004.
4. P. Flocchini, B. Mans, and N. Santoro. On the exploration of time-varying networks. *Theoretical Computer science*, volume 469, pages 53–68, 2013.
5. D. Ilcinkas and A. M. Wade. On the Power of Waiting when Exploring Public Transportation Systems. In *15th International Conference On Principles Of Distributed Systems (OPODIS)*, LNCS 7109, pages 451–464, 2011.
6. F. Kuhn, N.A. Lynch, and R. Oshman, Distributed computation in dynamic networks. In *42nd ACM symposium on Theory of computing (STOC)*, pages 513–522, 2010.
7. F. Kuhn and R. Oshman, Dynamic networks: models and algorithms. *ACM SIGACT News*, volume 42(1), pages 82–96, 2011.
8. C. E. Shannon, Presentation of a maze-solving machine. 8th Conf. of the Josiah Macy Jr. Found. (Cybernetics), pages 173–180, 1951.