

Tight bounds on exploration of constantly connected cacti-paths

Ahmed Mouhamadou WADE *

LTISI laboratory GIT, Polytechnic School of Thiès (EPT), Thiès, Sénégal.

World Journal of Advanced Research and Reviews, 2021, 12(01), 355–361

Publication history: Received on 15 September 2021; revised on 18 October 2021; accepted on 20 October 2021

Article DOI: <https://doi.org/10.30574/wjarr.2021.12.1.0534>

Abstract

In this paper, we study the necessary and sufficient time to explore constantly connected dynamics graphs by a mobile entity (agent). A dynamic graph is constantly connected if for each time units, there exists a stable connected spanning tree [10]. We focus on the case where the underlying graph is a cactus-path (graph reduced to a path of k rings in which two neighbor rings have at most one vertex in common) and we assume that the agent knows the dynamics of the graph. We show that $5n - \theta(1)$ time units are necessary and sufficient to explore any constantly connected dynamic graph based on the cactus-path $Ch_{2,n}$ (composed of two same size rings n). The upper bound is generalized on dynamic graphs based on cacti-paths with k rings. We show that for any constantly connected dynamic graph of size N based on a cactus-path, $4N - \max\{n_1, n_k\} - 3k - 3$ time units are sufficient to explore the graph, with k the length of the path, $N = \sum_{i=1}^k n_i - k + 1$ the size of the dynamic graph and n_i the size of the ring which is at position i starting from left to right.

Keywords: Dynamic graph; Exploration; Mobile agent; Cactus-path

1. Introduction

A mobile entity named agent which moves in a dynamic network (modeled by a dynamic graph), of which it does not know the structure, must visit all of its vertices or edges. This classical problem in algorithms by mobile agents named exploration has been much studied in static networks since the seminal paper of Claude Shannon [11]. It is motivated by its applications in logic, complexity, and robotics. Concerning dynamic graphs, only the cases of periodic dynamic graphs [6, 8] and constantly connected dynamic rings [3, 7, 1] have been studied. This is partly due to the fact that communications networks have long been assumed to be static. Since a decade, researchers began to model dynamic networks using dynamic graphs. Many more or less equivalent models which take into account the extreme dynamics of certain communication networks have been developed. One of the first models developed, and also one of the most classic is the evolving graph model [5]. In all its generality, the evolving graphs allow to model a large set of synchronous dynamic networks (see Section 2 for definitions). Consequently, to obtain interesting results, it is almost always necessary to formulate hypotheses allowing to reduce the possibilities of dynamic graphs generated by the model. Let us cite for example the assumption of periodicity of edges (dynamic edge-periodic graphs), where we assume that each edge of the underlying graph appears and disappears periodically, and the hypothesis of constant connectivity, for which the graph must be connected at all times. This last hypothesis, very classic, was generalized by Kuhn, Lynch and Oshman [9] 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 time units, there is a connected spanning subgraph that is stable throughout the period. (The notion of constant connectivity is equivalent to the notion of 1-interval connectivity.) This new notion, which captures connection stability over time, allows to derive interesting results: the T -interval-connectivity allows a saving of a factor of approximately $\theta(T)$ on the number of messages necessary and sufficient to carry out an exchange of information between all the vertices of network [4, 9].

* Corresponding author: Ahmed Mouhamadou WADE
Assistant Professor, LTISI laboratory GIT, Polytechnic School of Thiès (EPT), Thiès, Sénégal.

In this paper, we assume that the agent knows the dynamics of the network and we give an upper bound on the exploration time of constantly connected dynamic graphs based on cacti-paths of size n (cacti reduced to a path of k rings). A cactus is a connected graph in which two cycles have at most one vertex in common (see Section 2).

Our results. We show that to explore the constantly connected dynamic graphs based on the cactus-path $Ch_{2,n}$, $5n \theta(1)$ time units are necessary and sufficient. The upper bound is generalized on all cacti-paths of size k . We show that for any constantly connected dynamic graph of size N based on a cactus-path, $4N - \max\{n_1, n_k\} - 3k - 3$ time units are sufficient to explore the graph, with k the length of the path, $N = \sum_{i=1}^k n_i - k + 1$ the size of the dynamic graph and n_i the size of the ring which is at position i starting from left to right.

2. Preliminaries

This section provides precise definitions of the concepts and models informally mentioned above. We also give some previous results from the literature on the problem studied. The proofs of the theorems mentioned in this section are given in [9].

2.1. Definition 1. (Dynamic graph)

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

2.2. Definition 2. (Underlying graph)

Given a dynamic graph $\mathcal{G} = (V, \mathcal{E})$, the static graph $G = (V, \cup \mathcal{E}(i))$ is called the underlying graph of \mathcal{G} . Conversely, the dynamic graph \mathcal{G} is said to be based on the static graph G .

2.3. Definition 3. (Constant connectivity)

A dynamic graph \mathcal{G} is said to be constantly connected if, for any integer i , the static graph $G_i = (V, \mathcal{E}(i))$ is connected.

2.4. Definition 4. (Cactus)

A cactus is a simple graph $G = (V, E)$ in which two connected cycles have at most one vertex in common [2].

2.5. Definition 5. (Cactus-path)

A cactus-path is a cactus such that, if we represent the cycles by vertices and the connections between cycles by edges, we obtain a path (see **Figure 1**).

The length of the path is the number of cycles of the cactus-path. In the following, we denote it by k . In the **figure 1**, we give an example of cactus-path of length 4

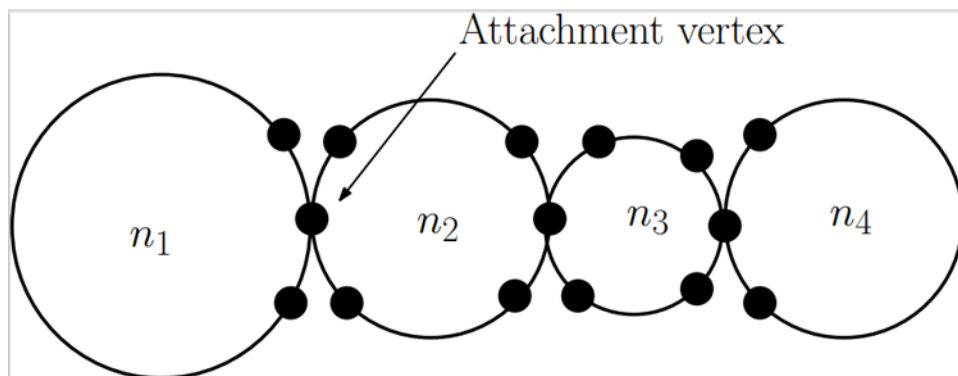


Figure 1 Example of Cactus-path of length 4

In this paper, we consider dynamic graphs based on a cactus-path of size n . A mobile entity, called agent, operates on these dynamic graphs. The agent can traverse at most one edge per time unit. We say that an agent explores a dynamic graph if and only if it visits all its vertices. We also assume that the agent knows the dynamics of the graph, that is to say, the times of appearance and disappearance of the edges of the dynamic graph.

In this article, we will use the following results from the literature.

Theorem 1. [7] For any integer $n \geq 3$ and for any constantly connected dynamic graph based on a ring with n vertices, there exists an agent (algorithm), **Explore-ring**, exploring this dynamic graph in time at most $2n - 3$ (assuming that the agent knows the dynamics of the graph).

Theorem 2. [10] For any constantly connected dynamic graph on n vertices, at most $n - 1$ time units are sufficient for an agent to go from any vertex to any other vertex in the graph, when the agent knows the dynamics of the graph.

3. Particular case

In this section, we consider the constantly connected dynamic graphs based on the cactus-path $Ch_{2,n}$ (see figure 2) composed of two rings of the same size n , and we give the time complexity of its exploration

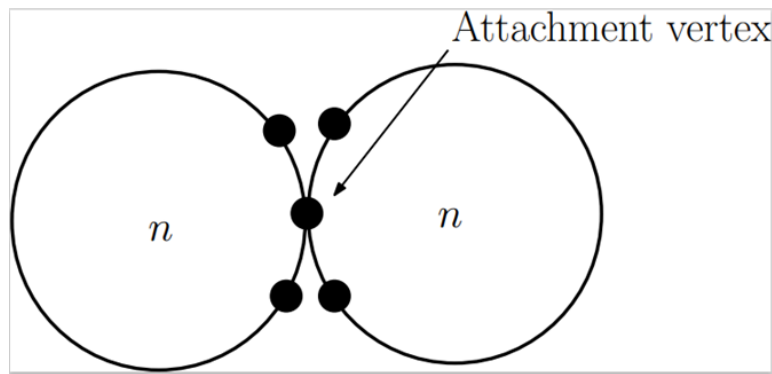


Figure 2 The Cactus-path $Ch_{2,n}$

3.1. Upper bound

We start by giving an upper bound on the exploration time of constantly connected dynamic graphs based on $Ch_{2,n}$. The algorithm we are using here is very simple. And we show in the following that it is optimal. Let's describe it. If the agent begins his exploration on the vertex common to both cycles, then he explores the leftmost ring first and then moves on to the next. Otherwise he explores the starting ring, then he moves on to the next ring and explores it. The ring exploration algorithm we use is the **Explore-ring** algorithm whose complexity is given by **Theorem 1**.

Theorem 3. For any integer $n \geq 3$ and for any constantly connected dynamic graph based on $Ch_{2,n}$, there exists an agent (algorithm), exploring this dynamic graph in time at most $5n - 7$ time units, with n the size of each ring (the agent knows the dynamics of the graph).

Proof. The complexity of the algorithm is $5n - 7$ because, to explore the starting ring, the agent pays at most $2n - 3$ time units (exploration time of a ring with n vertices cf. **Theorem 1**). Going to the vertex common to both cycles cost at most $n - 1$ time units cf. **Theorem 2**. Once on the vertex in common, all that remains is to explore the second ring, which costs at most $2n - 3$ other time units. So in total, the agent will pay at most $5n - 7$ time units to explore any constantly connected dynamic graph based on $Ch_{2,n}$.

3.2. Lower bound

In this section, we show that the simple algorithm described in the **section 3.1** is almost optimal. We have the following theorem which proves it.

Theorem 4. For any integer $n \geq 3$, there exists a constantly connected dynamic graph based on $Ch_{2,n}$, such that any agent must pay at least $5n - 8$ time units to explore it, with n the size of each ring. This bound remains even if the agent knows the dynamics of the graph.

Proof. For any integer $n \geq 3$, we define the constantly connected dynamic graph G based on $Ch_{2,n}$ as follows. Let A_n and A'_n be the two rings which compose $Ch_{2,n}$. Let v_0, v_1, \dots, v_{n-1} , respectively u_0, u_1, \dots, u_{n-1} , be the vertices of A_n , respectively A'_n , in the clockwise direction. Without loss of generality, suppose the exploration starts from the vertex

v_0 at time 0. The two rings are connected as follows to form Ch_{2n} . The vertex u_0 is attached to the vertex v_3 . In the following, we assume that the two vertices form one, and if we are on the ring A_n , it is called v_3 , otherwise its name is u_0 . The dynamic graph \mathcal{G} is built as follows (see Figure 3):

- The edge $\{v_0, v_1\}$ is absent during the time interval $[0, n - 2)$.
- The edge $\{v_1, v_2\}$ is absent during the time interval $[n - 1, 2n - 2)$.
- The edge $\{v_2, v_3\}$ is absent during the time interval $[2n - 2, 5n)$.
- The edge $\{u_0, u_1\}$ is absent during the time interval $[0, 2n - 5) \cup [3n - 6, 4n - 7)$.
- The edge $\{u_1, u_2\}$ is absent during the time interval $[2n - 5, 3n - 6) \cup [4n - 7, 5n)$.

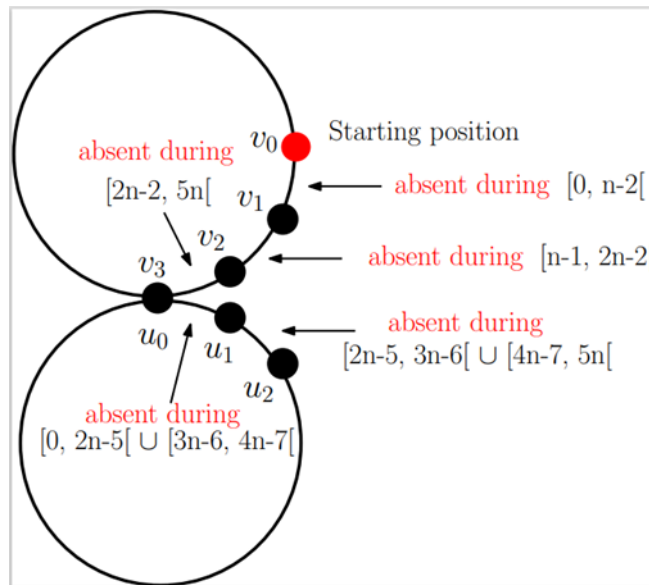


Figure 3 Constantly connected dynamic graph based on Ch_{2n} achieving the worst-case exploration time

Note that \mathcal{G} is connected at each unit of time, so it is constantly connected. Note also that v_3 and u_0 is the same vertex. Consider any agent (algorithm). We now prove that the time the agent uses to explore \mathcal{G} is at least $5n - 8$ time units. To explore the dynamic graph, the agent must visit the set of vertices of \mathcal{G} in particular the vertices v_1, v_2, v_3, u_1 and u_2 . We consider the following cases.

3.3. Case 1 the vertex v_1 is explored before v_2, u_1 and u_2 .

According to the order of visit of the vertices v_2, u_1 and u_2 , we have the following bounds.

- If the agent visits v_1, v_2, u_1 then u_2 in that order
- To visit v_1 without going through the vertex v_2 , the agent must necessarily go through the edge $\{v_0, v_1\}$. By construction, the edge $\{v_0, v_1\}$ is only present after $n - 2$ time units. So the agent will pay at least $n - 1$ time units to visit v_1 for the first time. In addition, the edge $\{v_1, v_2\}$ is absent during the time interval $[n - 1, 2n - 2)$. So to visit v_2 after visiting v_1 , the fastest way is to go around the ring A_n and pay $n - 1$ other time units. So to visit v_2 for the first time, the agent pay at least $2n - 2$ time units. Once the vertices v_1 and v_2 have been visited, the agent will visit the vertices u_1 and u_2 . Since the vertices u_1 and u_2 lie on the ring A'_n , the agent must go to the attachment vertex (v_3) to switch to the other part of the dynamic graph. The edge $\{v_2, v_3\}$ being disconnected during the time interval $[2n - 2, 5n)$, so the agent will pay at least $3n - 3$ time units to arrive on v_3 after visiting v_1 and v_2 . Once on the vertex v_3 (u_0), to visit u_1 without going through u_2 , the agent must go through the edge $\{u_0, u_1\}$, which is absent during the time interval $[3n - 6, 4n - 7)$. So the agent will pay at least $4n - 6$ time units to visit u_1 for the first time. To visit u_2 after visiting u_1 , the best way is to go around the ring A'_n paying $n - 1$ other time units, because the edge $\{u_1, u_2\}$ is absent during the time interval $[4n - 7, 5n)$. So the agent will visit the vertex v_2 for the first time after $5n - 7$ time units.
- If the agent visits v_1, v_2, u_2 then u_1 in that order

- In the previous paragraph, we showed that the agent pays at least $3n - 3$ time units to arrive on v_3 after visiting v_1 and v_2 for the first time. Once on v_3 (u_0), to visit u_2 before visiting u_1 , the agent pays at least $n - 2$ other time units because the path u_0, u_{n-1}, \dots, u_2 is of length $n - 2$. So to visit u_2 for the first time without going through u_1 , the agent pays at least $4n - 5$ time units. To visit u_1 after visiting u_2 , the fastest way is to go around the ring through u_0 and paying $n - 1$ other time units, because the edge $\{u_1, u_2\}$ is absent during the time interval $[4n - 7, 5n)$. So to visit u_1 for the first time the agent pays at least $5n - 6$ time units.
- If the agent visits v_1, u_1, v_2 then u_2 in that order
- In the previous paragraph, we showed that the agent pays at least $n - 1$ time units to visit v_1 for the first time without going through v_2 . After visiting v_1 , to visit u_1 before v_2 , the agent must go to v_3 without going through v_2 which costs him $n - 2$ other time units. So the agent pays at least $2n - 3$ time units to arrive on v_3 after visiting v_1 . Once on u_0 , the edge $\{u_0, u_1\}$ is present so it pays one more time unit to visit u_1 . So the agent pays $2n - 2$ time units to visit u_1 after visiting v_1 . To visit v_2 after visiting v_1 and u_1 and before visiting u_2 , the agent must go to u_0 without going through u_2 this costs him one time unit. So the agent will be on u_0 at sooner after $2n - 1$ time units. Once on u_0 , the fastest way to visit v_2 is to pay $n - 1$ time units through v_0 because the edge $\{v_3, v_2\}$ is absent during the time interval $[2n - 2, 5n)$. So the agent pays at least $3n - 2$ time units to visit v_2 for the first time. Now he only has to visit u_2 to finish his exploration. To visit u_2 , the agent must go to v_3 , as the edge $\{v_3, v_2\}$ is absent during the time interval $[2n - 2, 5n)$, then the faster way is to pay $n - 1$ other time units by going through v_0 . So the agent pays at least $4n - 3$ time units to arrive on v_3 after visiting v_2 for the first time. The edge $\{u_1, u_2\}$ is absent during the time interval $[4n - 7, 5n)$. So once on u_0 , the fastest way to visit u_2 is to override u_1 and pay $n - 2$ other time units. So to visit u_2 for the first time, the agent pays at least $5n - 5$ time units.
- If the agent visits v_1, u_1, u_2 then v_2 in that order
- The previous paragraph shows that to visit u_1 for the first time after visiting v_1 , the agent pays at least $2n - 2$ time units. To visit u_2 for the first time after visiting u_1 , the agent pays at least $3n - 5$ time units. To visit v_2 after visiting u_2 for the first time, the agent must go to the vertex u_0 . As the edge $\{u_0, u_1\}$ is absent during the time interval $[3n - 6, 4n - 7)$. Then the fastest is to go around the ring by paying $n - 2$ other time units. So the agent pays at least $4n - 7$ time units to arrive on u_0 after visiting u_2 for the first time. Once on u_0 , to visit v_2 , the fastest way is to pay $n - 1$ time units crossing the whole ring because the edge $\{v_3, v_2\}$ is absent during the time interval $[2n - 2, 5n)$. So the agent pays at least $5n - 8$ time units to visit v_2 for the first time.
- If the agent visits v_1, u_2, u_1 then v_2 in that order
- We have shown that to visit v_1 and go to the vertex v_3 , the agent pays at least $2n - 3$ time units. Once on v_3 , to visit u_2 without going through u_1 , the agent pays at least $n - 2$ other time units. So to visit u_2 for the first time, the agent pays at least $3n - 5$ time units. To visit u_1 after visiting u_2 , the agent pays one more time units since the edge $\{u_1, u_2\}$ is present. To visit v_2 after visiting u_1 , the agent pays at least $4n - 5$ time units to visit u_0 . Once on u_0 , to visit v_2 , the fastest way is to go around the ring via v_0 because the edge $\{v_3, v_2\}$ is absent during the time interval $[2n - 2, 5n)$. So the agent pays at least $5n - 6$ time units to visit v_2 for the first time.
- If the agent visits v_1, u_2, v_2 then u_1 in that order
- The previous paragraph shows that to visit u_2 after visiting v_1 , the agent pays at least $3n - 5$ time units. To visit v_2 after visiting u_2 and before visiting u_1 , the agent must go to the vertex u_0 without going through u_1 . Which costs at least $n - 2$ time units. So to get to u_0 after visiting u_2 for the first time, the agent pays at least $4n - 7$ time units. Once on u_0 , to visit v_2 for the first time the agent pays at least $5n - 8$ time units because the edge $\{v_2, v_3\}$ is absent during the time interval $[2n - 2, 5n)$. To visit u_1 after visiting v_2 , the agent must go to v_3 . The edge $\{v_2, v_3\}$ is absent during the time interval $[2n - 2, 5n)$, therefore the agent pays at least $5n + 1$ time units to arrive on v_3 after visiting v_2 . Once at v_3 , it takes one more time unit to visit u_1 for the first time. So the agent pays $5n + 2$ to visit u_1 for the first time.
- We do similar analyzes to case 1 to prove the other cases, and we get the lower bound $5n - 8$ claimed in the theorem.

4. General Case

We show here that the algorithm describe in **section 3.1** can be generalized by considering constantly connected dynamic graphs based on cacti-paths of length $k \geq 1$ composed of k rings of same or different sizes. Let's describe the generalized algorithm before giving its complexity. The algorithm we denote **Explore-Cactus-path** is very simple. From the starting vertex, the agent goes to the leftmost ring before starting his exploration. Once on the leftmost cycle, he explores it then he passes over the connecting vertex of the next ring not yet explored, and so on until he explores the

whole graph. The ring exploration algorithm we use is the **Explore-ring** algorithm whose complexity is given by **Theorem 1**.

Theorem 5. For any integer $N \geq 3$, $k \geq 1$ and for any constantly connected dynamic graph based on a cactus-path composed of k rings, the algorithm Explore-Cactus-path allows to explore this dynamic graph in at most $4N - \max\{n_1, n_k\} - 3k - 3$ time units, where k is the length of the path, $N = \sum_{i=1}^k n_i - k + 1$ the size of the dynamic graph and n_i the size of the ring which is at position i from left to right.

Proof

Let $N \geq 3$, $k \geq 1$, G a cactus-path composed of k cycles, and \mathcal{G} a constantly connected dynamic graph based on G . Let n_i be the size of the ring which is at position i starting from left to right and $N = \sum_{i=1}^k n_i - k + 1$ the number of vertices of the dynamic graph. Let A be an agent executing the **Explore-Cacti-path** algorithm. To go to the leftmost cycle, the agent pays at most $N - 1$ time units (cf. Theorem 2). To explore a cycle and move on to the next cycle, the agent executes the algorithm **Explore-ring** and pays at most three times the size of the current ring minus four time units (cf. **Theorem 1** and **Theorem 2**). These two procedures are executed $k - 1$ times. So according to the starting cycle, the agent will pay at most $\sum_{i=2}^k (3n_i - 4)$ or $\sum_{i=1}^{k-1} (3n_i - 4)$ time units to explore the first $k - 1$ cycles and go to the connecting vertex of the last cycle. To explore the last cycle, twice its size minus three time units are sufficient (cf. **Theorem 1**). So to explore the dynamic graph, the agent pays at most $\min(\sum_{i=1}^{k-1} (3n_i - 4) + 2n_k - 3 + N - 1, \sum_{i=2}^k (3n_i - 4) + 2n_1 - 3 + N - 1)$ time units. Which is at most $4N - \max\{n_1, n_k\} - 3k - 3$. This concludes the proof.

5. Conclusion

In this paper, we studied the time complexity for exploring constantly connected dynamic graphs based on cacti-paths, under the assumption that the agent knows the dynamics of the graph. We first considered the particular case of underlying graph $Ch_{2,n}$, and we have shown that $5n - \theta(1)$ time units are necessary and sufficient to explore it. And we then generalized the upper bound on all cacti-paths. This study opens several perspectives. An interesting question to investigate would be if T -interval connectivity (for $T \geq 1$) allows to save a significant factor in the exploration time. A further perspective is to consider the exploration problem of dynamic graphs using more than one agent, assuming standard models of communication between the agents. The objective would be to study whether dynamic graph exploration can be performed more efficiently by using more than one agent.

Compliance with ethical standards

Acknowledgments

The authors would like to thank David Ilcinkas for insightful and valuable discussions regarding the topic of this paper.

References

- [1] Bournat M, Dubois S, Petit F. Computability of Perpetual Exploration in Highly Dynamic Rings. *In 37th IEEE International Conference on Distributed Computing Systems, IEEE Computer Society.* 2017; 794–804.
- [2] Burkard R, Krarup J. A Linear Algorithm for the Pos/Neg-Weighted 1- Median Problem on a Cactus. *Computing.* 1998; 60(3): 193–216.
- [3] Di Luna GA, Dobrev S, Flocchini P, Santoro N. Distributed exploration of dynamic rings. *In Distributed Computing.* 2020; 33(1): 41–67.
- [4] Dutta C, Pandurangan G, Rajaraman R, Sun Z. Information spreading in dynamic networks. *CoRR*, abs/1112.0384, 2011.
- [5] Ferreira A. Building a Reference Combinatorial Model for Dynamic Networks: Initial Results in Evolving Graphs. *INRIA*, RR-5041. 2003.
- [6] Flocchini P, Mans B, Santoro N. On the exploration of time-varying networks. *Theoretical Computer science.* 2013; 469: 53–68,
- [7] Ilcinkas D, Wade AM. Exploration of the T-Interval-Connected Dynamic Graphs: the Case of the Ring. *In Theory Comput. Syst.* 2018; 62: 1144– 1160.

- [8] Ilcinkas D, Wade AM. Exploration of carrier-based time-varying networks: The power of waiting. *In Theoretical Computer Science*. 2020; 841: 50-61,
- [9] Kuhn F, Lynch NA, Oshman R. Distributed computation in dynamic networks. *In 42nd ACM symposium on Theory of computing (STOC)*. 2010; 513–522.
- [10] O’Dell R, Wattenhofer R. Information dissemination in highly dynamic graphs. *In DIALM-POMC*. 2005; 104–110.
- [11] Shannon CE. Presentation of a maze-solving machine. *8th Conf. of the Josiah Macy Jr. Found. (Cybernetics)*. 1951; 173–180.