

Exploration of Dynamic Graphs

David Ilcinkas Ahmed Wade

LaBRI, CNRS & Université de Bordeaux, France

ANR MACARON, January 10, 2014

Graph exploration by a mobile agent

Graph exploration

- The mobile entity, called **agent**, has to **visit every node** at least once and then terminate.

Graph exploration by a mobile agent

Graph exploration

- The mobile entity, called **agent**, has to **visit every node** at least once and then terminate.
- Well studied classical problem.

Graph exploration by a mobile agent

Graph exploration

- The mobile entity, called **agent**, has to **visit every node** at least once and then terminate.
- Well studied classical problem.
- Focus on the time and space complexity, on the impact of a priori knowledge, on fault-tolerance, etc.

Context

Distributed computing systems are becoming more and more dynamic.

Context

Distributed computing systems are becoming more and more dynamic.

Causes

- Frequent connections / disconnections to/in the network

Distributed computing systems are becoming more and more dynamic.

Causes

- Frequent **connections** / **disconnections** to/in the network
- **Fault or crash** of communicating objects and links

Distributed computing systems are becoming more and more dynamic.

Causes

- Frequent **connections** / **disconnections** to/in the network
- **Fault or crash** of communicating objects and links
- **Sleep** modes to save energy

Context

Distributed computing systems are becoming more and more **dynamic**.

Causes

- Frequent **connections** / **disconnections** to/in the network
- **Fault or crash** of communicating objects and links
- **Sleep** modes to save energy

Classical models: Static fault-tolerant networks

- Assume that the **frequency** of fault occurrences is **small**

Context

Distributed computing systems are becoming more and more **dynamic**.

Causes

- Frequent **connections** / **disconnections** to/in the network
- **Fault or crash** of communicating objects and links
- **Sleep** modes to save energy

Classical models: Static fault-tolerant networks

- Assume that the **frequency** of fault occurrences is **small**
- Assume that the **number** of fault occurrences is **small**

Distributed computing systems are becoming more and more **dynamic**.

Causes

- Frequent **connections** / **disconnections** to/in the network
- **Fault or crash** of communicating objects and links
- **Sleep** modes to save energy

Classical models: Static fault-tolerant networks

- Assume that the **frequency** of fault occurrences is **small**
- Assume that the **number** of fault occurrences is **small**

In fact, these models become **insufficient** for **very dynamic networks**.

The models we consider

Periodically-varying graphs (PV-graphs)

- PV-graphs model in particular various types of **public transportation systems** (bus, subway systems, ...).

The models we consider

Periodically-varying graphs (PV-graphs)

- PV-graphs model in particular various types of **public transportation systems** (bus, subway systems, ...).
- A **PV-graph** consists of a set of routes and a set of **carriers**, each following **periodically** its respective **route** among the sites of the system.

The models we consider

Periodically-varying graphs (PV-graphs)

- PV-graphs model in particular various types of **public transportation systems** (bus, subway systems, ...).
- A **PV-graph** consists of a set of routes and a set of **carriers**, each following **periodically** its respective **route** among the sites of the system.

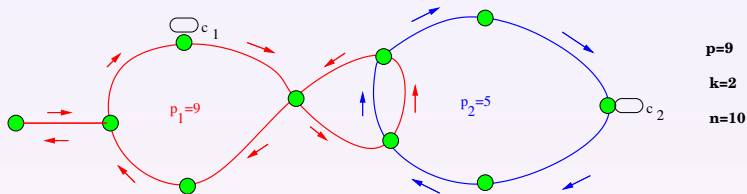
Evolving graphs

- One of the **first developed** models, and also one of the **most standard**.
- An **evolving graph** is a (possibly infinite) sequence $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \dots$ of static graphs based on the same set of nodes.

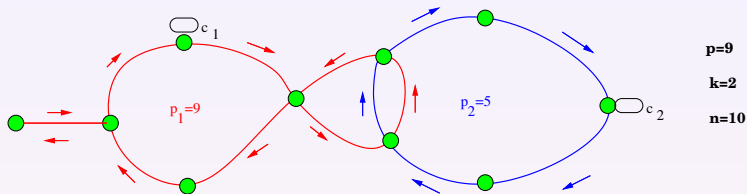
Some related work (on mobile agents)

- [FKMS10] P. Flocchini, M. Kellett, P. C. Mason, and N. Santoro.
Mapping an unfriendly subway system.
FUN 2010
- [FKMS12a] P. Flocchini, M. Kellett, P. C. Mason, and N. Santoro.
Searching for black holes in subways.
Theory of Computing Systems 2012
- [FKMS12b] P. Flocchini, M. Kellett, P. C. Mason, and N. Santoro.
Finding Good Coffee in Paris.
FUN 2012
- [FMS13] P. Flocchini, B. Mans, and N. Santoro.
On the exploration of time-varying networks.
Theoretical Computer science 2013 (ISAAC 2009)
- [IYK12] T. Izumi, Y. Yamauchi, and S. Kamei.
BA: Mobile Agent Rendezvous on Edge Evolving Rings.
SSS 2012

First part: the PV-graphs



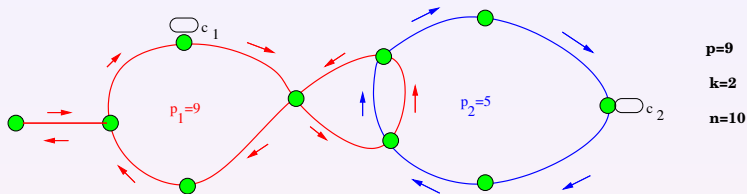
First part: the PV-graphs



The model of PV-graphs

- A **PV-graph** is defined by the tuple (S, C, Id, R) .

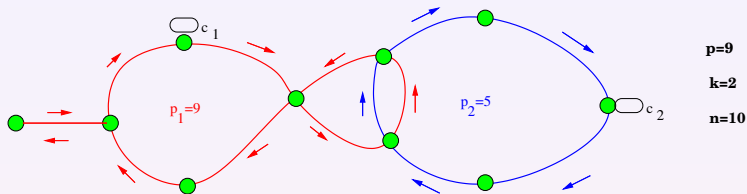
First part: the PV-graphs



The model of PV-graphs

- A **PV-graph** is defined by the tuple (S, C, Id, R) .
- $S = \{s_1, \dots, s_n\}$ is the set of **sites** (or nodes).

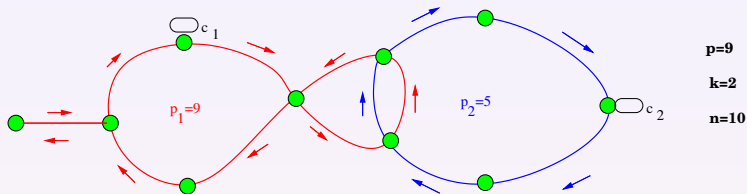
First part: the PV-graphs



The model of PV-graphs

- A **PV-graph** is defined by the tuple (S, C, Id, R) .
- $S = \{s_1, \dots, s_n\}$ is the set of **sites** (or nodes).
- $C = \{c_1, c_2, \dots, c_k\}$ is the set of **carriers**.

First part: the PV-graphs



The model of PV-graphs

- A **PV-graph** is defined by the tuple (S, C, Id, R) .
- $S = \{s_1, \dots, s_n\}$ is the set of **sites** (or nodes).
- $C = \{c_1, c_2, \dots, c_k\}$ is the set of **carriers**.
- Each carrier c has an identifier $Id(c)$ and follows a finite sequence $R(c) = (s_{i_1}, \dots, s_{i_{p(c)}})$ of sites, called its **route**, in a **periodic** manner.

PV-graph exploration

- An entity, called agent, is operating on these PV-graphs.

PV-graph exploration

- An entity, called agent, is operating on these PV-graphs.
 - It can see the carriers and their identities

PV-graph exploration

- An entity, called agent, is operating on these PV-graphs.
 - It can see the carriers and their identities
 - It can ride on a carrier to go from a site to another

PV-graph exploration

- An entity, called agent, is operating on these PV-graphs.
 - It can see the carriers and their identities
 - It can ride on a carrier to go from a site to another
 - It can also leave a carrier to ride with another carrier

PV-graph exploration

- An entity, called agent, is operating on these PV-graphs.
 - It can see the carriers and their identities
 - It can ride on a carrier to go from a site to another
 - It can also leave a carrier to ride with another carrier
- We do not assume any restriction on:

PV-graph exploration

- An entity, called agent, is operating on these PV-graphs.
 - It can see the carriers and their identities
 - It can ride on a carrier to go from a site to another
 - It can also leave a carrier to ride with another carrier
- We do not assume any restriction on:
 - the **memory size** of the agent

PV-graph exploration

- An entity, called agent, is operating on these PV-graphs.
 - It can see the carriers and their identities
 - It can ride on a carrier to go from a site to another
 - It can also leave a carrier to ride with another carrier
- We do not assume any restriction on:
 - the **memory size** of the agent
 - its **computational capabilities**

PV-graph exploration

- An entity, called agent, is operating on these PV-graphs.
 - It can see the carriers and their identities
 - It can ride on a carrier to go from a site to another
 - It can also leave a carrier to ride with another carrier
- We do not assume any restriction on:
 - the **memory size** of the agent
 - its **computational capabilities**
- We say that an agent **explores** a PV-graph if and only if, starting at time 0 on the starting site, the agent eventually **visits all sites** of the PV-graph and **switches** afterwards to **a terminal state**.

A PV-graph is said to be

- **anonymous** if the nodes don't have any identities, or the agent is not able to see them.

A PV-graph is said to be

- **anonymous** if the nodes don't have any identities, or the agent is not able to see them.
- **labeled** if the nodes have distinct identities and the agent can see and memorize them.

A PV-graph is said to be

- **anonymous** if the nodes don't have any identities, or the agent is not able to see them.
- **labeled** if the nodes have distinct identities and the agent can see and memorize them.
- **homogeneous** if all its carriers have the same period.

Previous result (1/2)

[Paola Flocchini, Bernard Mans and Nicola Santoro, TCS 2013]

Previous result (1/2)

[Paola Flocchini, Bernard Mans and Nicola Santoro, TCS 2013]

Anonymous PV-Graphs

The knowledge of an **upper bound** on the longest **period** is **necessary and sufficient**.

Previous result (1/2)

[Paola Flocchini, Bernard Mans and Nicola Santoro, TCS 2013]

Anonymous PV-Graphs

The knowledge of an **upper bound** on the longest **period** is **necessary and sufficient**.

Labeled PV-Graphs

The knowledge of the **number of nodes** or an **upper bound** on the longest **period** is **necessary and sufficient**.

Previous result (2/2)

[Paola Flocchini, Bernard Mans and Nicola Santoro, TCS 2013]

- k : number of carriers
- p : longest period
- B : upper bound on p given to the agent

Previous result (2/2)

[Paola Flocchini, Bernard Mans and Nicola Santoro, TCS 2013]

- k : number of carriers
- p : longest period
- B : upper bound on p given to the agent

General case

- $\Omega(kp^2)$ moves and time steps.
- $O(kB^2)$ moves and time steps.

Previous result (2/2)

[Paola Flocchini, Bernard Mans and Nicola Santoro, TCS 2013]

- k : number of carriers
- p : longest period
- B : upper bound on p given to the agent

General case

- $\Omega(kp^2)$ moves and time steps.
- $O(kB^2)$ moves and time steps.

Homogeneous PV-graphs

- $\Omega(kp)$ moves and time steps.
- $O(kB)$ moves and time steps.

Model of PV-graphs

[FMS13] considered that the agent **cannot leave the carrier** to stay on a site.

Model of PV-graphs

[FMS13] considered that the agent **cannot leave the carrier** to stay on a site.

Motivation

Not being able to stay on a site is particularly legitimate in low earth orbiting satellite systems for example, where the **sites do not correspond to any physical station**.

Model of PV-graphs

[FMS13] considered that the agent **cannot leave the carrier** to stay on a site.

Motivation

Not being able to stay on a site is particularly legitimate in low earth orbiting satellite systems for example, where the **sites do not correspond to any physical station**.

Remark

In most public transportation systems, it is possible for the agent (human or not) to **stay on a site** in order to wait for a (possibly different) carrier.

Our contribution on PV-graphs

Our contribution

We extend the study of Flocchini, Mans and Santoro to the case when **the agent can leave a carrier** to stay at a site.

Our contribution on PV-graphs

Our contribution

We extend the study of Flocchini, Mans and Santoro to the case when **the agent can leave a carrier** to stay at a site.

We prove that this new ability allows

- to explore PV-graphs that are less connected over time.

Our contribution on PV-graphs

Our contribution

We extend the study of Flocchini, Mans and Santoro to the case when **the agent can leave a carrier** to stay at a site.

We prove that this new ability allows

- to explore PV-graphs that are less connected over time.
- to reduce the **move** and **time complexities** in the **general case**.

Our contribution on PV-graphs

Our contribution

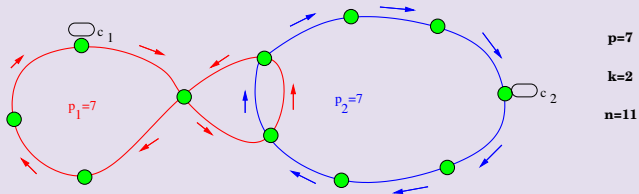
We extend the study of Flocchini, Mans and Santoro to the case when **the agent can leave a carrier** to stay at a site.

We prove that this new ability allows

- to explore PV-graphs that are less connected over time.
- to reduce the **move** and **time complexities** in the **general case**.
- the agent to completely **map the PV-graph**, in addition to just **explore** it.

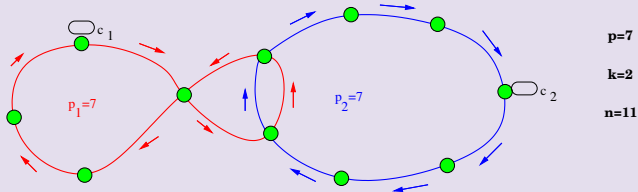
Definition: notions of connectivity

Connected PV-graph

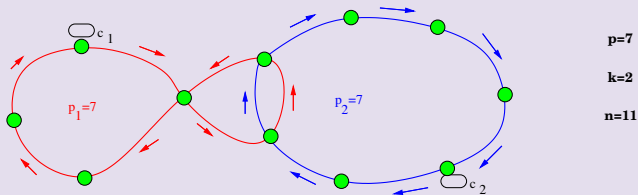


Definition: notions of connectivity

Connected PV-graph



Highly-connected PV-graph



Results comparison

- In this table we assume that $B = O(p)$
- [FMS13] (the agent cannot leave the carrier)
- Our results (the agent can leave the carrier)

	Connected	Highly-connected
General case	Impossible	$\Theta(kp^2)$ moves and time steps
	$\Theta(\min\{kp, np, n^2\})$ moves $\Theta(np)$ time steps	$\Theta(\min\{kp, np, n^2\})$ moves $\Theta(np)$ time steps
Homogeneous	Impossible	$\Theta(kp)$ moves and time steps
	$\Theta(\min\{kp, np, n^2\})$ moves $\Theta(np)$ time steps	$\Theta(\min\{kp, np, n^2\})$ moves $O(np)$ time steps

Results comparison

- In this table we assume that $B = O(p)$
- [FMS13] (the agent cannot leave the carrier)
- Our results (the agent can leave the carrier)

	Connected	Highly-connected
General case	Impossible	$\Theta(kp^2)$ moves and time steps
	$\Theta(\min\{kp, np, n^2\})$ moves $\Theta(np)$ time steps	$\Theta(\min\{kp, np, n^2\})$ moves $\Theta(np)$ time steps
Homogeneous	Impossible	$\Theta(kp)$ moves and time steps
	$\Theta(\min\{kp, np, n^2\})$ moves $\Theta(np)$ time steps	$\Theta(\min\{kp, np, n^2\})$ moves $O(np)$ time steps

Results comparison

- In this table we assume that $B = O(p)$
- [FMS13] (the agent cannot leave the carrier)
- Our results (the agent can leave the carrier)

	Connected	Highly-connected
General case	Impossible	$\Theta(kp^2)$ moves and time steps
	$\Theta(\min\{kp, np, n^2\})$ moves $\Theta(np)$ time steps	$\Theta(\min\{kp, np, n^2\})$ moves $\Theta(np)$ time steps
Homogeneous	Impossible	$\Theta(kp)$ moves and time steps
	$\Theta(\min\{kp, np, n^2\})$ moves $\Theta(np)$ time steps	$\Theta(\min\{kp, np, n^2\})$ moves $O(np)$ time steps

Results comparison

- In this table we assume that $B = O(p)$
- [FMS13] (the agent cannot leave the carrier)
- Our results (the agent can leave the carrier)

	Connected	Highly-connected
General case	Impossible	$\Theta(kp^2)$ moves and time steps
	$\Theta(\min\{kp, np, n^2\})$ moves $\Theta(np)$ time steps	$\Theta(\min\{kp, np, n^2\})$ moves $\Theta(np)$ time steps
Homogeneous	Impossible	$\Theta(kp)$ moves and time steps
	$\Theta(\min\{kp, np, n^2\})$ moves $\Theta(np)$ time steps	$\Theta(\min\{kp, np, n^2\})$ moves $O(np)$ time steps

Second part: the evolving graphs

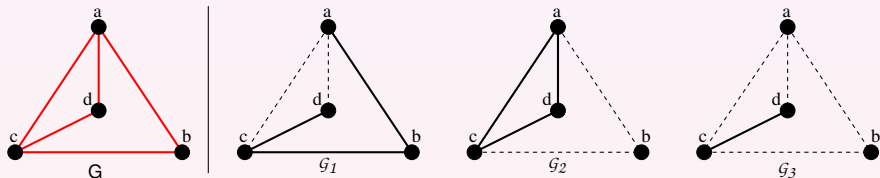
The model of evolving graphs (definition)

Given a static graph G , called the **underlying graph**, an **evolving graph** based on G is a (possibly infinite) sequence $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \dots$ of spanning but not necessarily connected subgraphs of G .

Second part: the evolving graphs

The model of evolving graphs (definition)

Given a static graph G , called the **underlying graph**, an **evolving graph** based on G is a (possibly infinite) sequence $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \dots$ of spanning but not necessarily connected subgraphs of G .



The model of evolving graphs

Advantages

- This model is particularly well **adapted** for modeling **synchronous dynamic networks**.
- It allows to **consider** an extremely **varied** set of **dynamic networks**.

The model of evolving graphs

Advantages

- This model is particularly well **adapted** for modeling **synchronous dynamic networks**.
- It allows to **consider** an extremely **varied** set of **dynamic networks**.

Remark

Note that, to obtain interesting results, it is often required to make **assumptions** that reduce the possibilities of dynamic graphs generated by the model.

Examples of assumptions

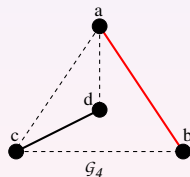
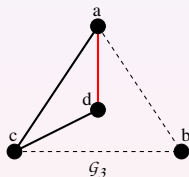
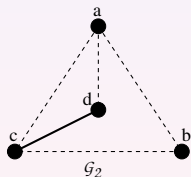
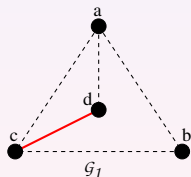
The assumption of connectivity over time

There is a **journey** (path over time) from **any vertex** to **any other vertex**.

Examples of assumptions

The assumption of connectivity over time

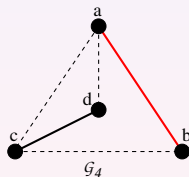
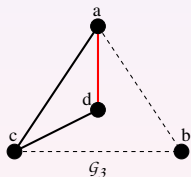
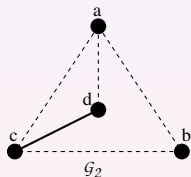
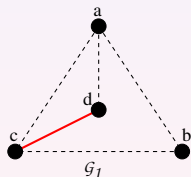
There is a **journey** (path over time) from **any vertex** to **any other vertex**.



Examples of assumptions

The assumption of connectivity over time

There is a **journey** (path over time) from **any vertex** to **any other vertex**.



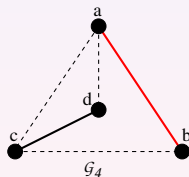
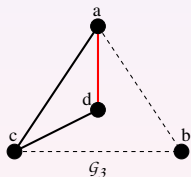
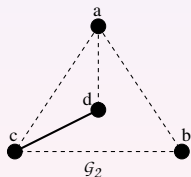
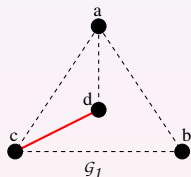
The assumption of constant connectivity

The graph must be connected at all times.

Examples of assumptions

The assumption of connectivity over time

There is a **journey** (path over time) from **any vertex** to **any other vertex**.



The assumption of constant connectivity

The graph must be connected at all times.

T -interval-connected graphs

[Kuhn, Lynch and Oshman (STOC 2010)]

T -interval-connected graphs

[Kuhn, Lynch and Oshman (STOC 2010)]

Definition

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.

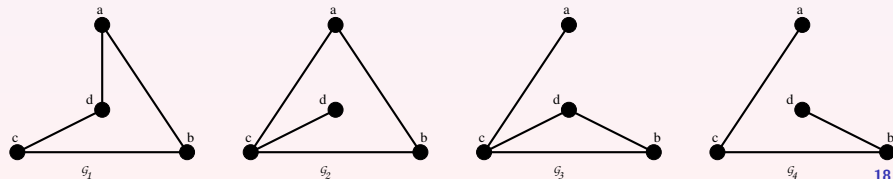
T -interval-connected graphs

[Kuhn, Lynch and Oshman (STOC 2010)]

Definition

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.

Example of **2-interval-connected** dynamic graph.



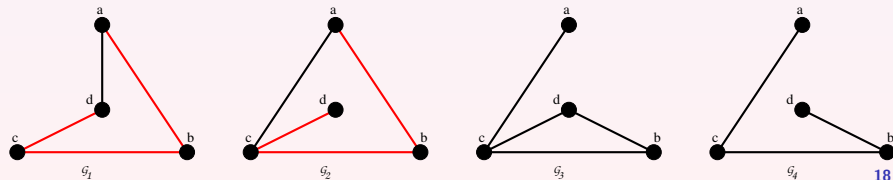
T -interval-connected graphs

[Kuhn, Lynch and Oshman (STOC 2010)]

Definition

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.

Example of **2-interval-connected** dynamic graph.



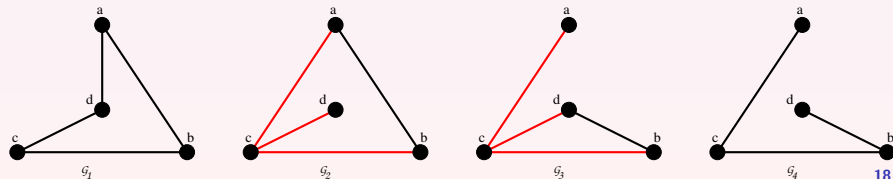
T -interval-connected graphs

[Kuhn, Lynch and Oshman (STOC 2010)]

Definition

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.

Example of **2-interval-connected** dynamic graph.



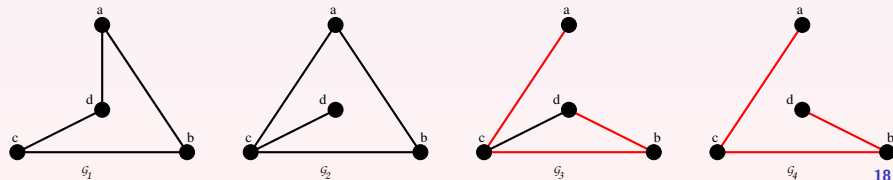
T -interval-connected graphs

[Kuhn, Lynch and Oshman (STOC 2010)]

Definition

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.

Example of **2-interval-connected** dynamic graph.



T -interval-connected graphs

Some interesting results [KLO10]

- In **1-interval-connected graphs** it is possible for nodes to determine the size of the network and compute any computable function of their initial inputs in $O(n^2)$ rounds.

T -interval-connected graphs

Some interesting results [KLO10]

- In **1-interval-connected graphs** it is possible for nodes to determine the size of the network and compute any computable function of their initial inputs in $O(n^2)$ rounds.
- If the **graph is T -interval-connected**, for $T > 1$, the computation can be **sped up** by a factor of T .

Model (1)

- We carry on **the study** of these T -interval-connected dynamic graphs by considering the **problem of exploration**.

Model (1)

- We carry on **the study** of these T -interval-connected dynamic graphs by considering the **problem of exploration**.
- The **traversal** of an edge takes one time unit.

Model (1)

- We carry on **the study** of these T -interval-connected dynamic graphs by considering the **problem of exploration**.
- The **traversal** of an edge takes one time unit.
- **A mobile agent explores the dynamic graph** if and only if it **visits all its vertices**.

Model (1)

- We carry on **the study** of these T -interval-connected dynamic graphs by considering the **problem of exploration**.
- The **traversal** of an edge takes one time unit.
- **A mobile agent explores the dynamic graph** if and only if it **visits all its vertices**.

Goal

Calculate the (worst-case) **time complexity** of this problem.

Scenarios

We consider the problem in **two scenarios**.

Scenarios

We consider the problem in **two scenarios**.

Scenario 1

The agent **does not know** the **dynamics of the graph**, that is the times of appearance and disappearance of the edges.

Scenarios

We consider the problem in **two scenarios**.

Scenario 1

The agent **does not know** the **dynamics of the graph**, that is the times of appearance and disappearance of the edges.

Scenario 2

The agent **knows** entirely and exactly the **dynamic graph** it has to explore.

Scenarios

We consider the problem in **two scenarios**.

Scenario 1

The agent **does not know** the **dynamics of the graph**, that is the times of appearance and disappearance of the edges.

Networks whose changes are related to **frequent and unpredictable** failures.

Scenario 2

The agent **knows** entirely and exactly the **dynamic graph** it has to explore.

Scenarios

We consider the problem in **two scenarios**.

Scenario 1

The agent **does not know** the **dynamics of the graph**, that is the times of appearance and disappearance of the edges.

Networks whose changes are related to **frequent and unpredictable** failures.

Scenario 2

The agent **knows** entirely and exactly the **dynamic graph** it has to explore.

Predictable dynamic networks, such as transportation networks for example.

Model (2)

In the first scenario (the agent **does not know** the **dynamics of the graph**)

Model (2)

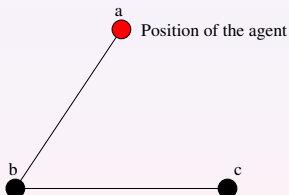
In the first scenario (the agent **does not know** the **dynamics of the graph**)

- The exploration problem is **impossible** to solve [KLO10]

Model (2)

In the first scenario (the agent **does not know** the **dynamics of the graph**)

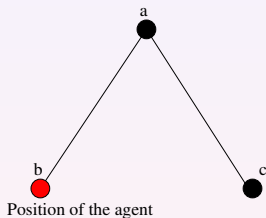
- The exploration problem is **impossible** to solve [KLO10]



Model (2)

In the first scenario (the agent **does not know** the **dynamics of the graph**)

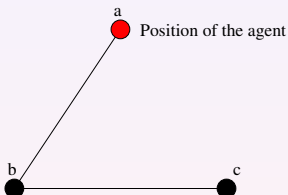
- The exploration problem is **impossible** to solve [KLO10]



Model (2)

In the first scenario (the agent **does not know** the **dynamics of the graph**)

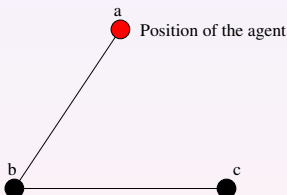
- The exploration problem is **impossible** to solve [KLO10]



Model (2)

In the first scenario (the agent **does not know** the **dynamics of the graph**)

- The exploration problem is **impossible** to solve [KLO10]

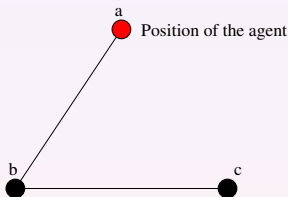


- We assume that the dynamic graph is **δ -recurrent**

Model (2)

In the first scenario (the agent **does not know** the **dynamics of the graph**)

- The exploration problem is **impossible** to solve [KLO10]



- We assume that the dynamic graph is **δ -recurrent**

δ -recurrence (definition)

A dynamic graph is **δ -recurrent** if every edge of the underlying graph is present at least once every δ time steps.

Some observations

In the second scenario (the agent **knows** the **dynamics**)

Some observations

In the second scenario (the agent **knows** the **dynamics**)

- The temporal diameter is bounded by $n - 1$ [KLO10]

Some observations

In the second scenario (the agent **knows** the **dynamics**)

- The temporal diameter is bounded by $n - 1$ [KLO10]
- The exploration time is thus at most $(n - 1)^2$

Some observations

In the second scenario (the agent **knows** the **dynamics**)

- The temporal diameter is bounded by $n - 1$ [KLO10]
- The exploration time is thus at most $(n - 1)^2$
- Best known lower bound is $2n - 3$

Some observations

In the second scenario (the agent **knows** the **dynamics**)

- The temporal diameter is bounded by $n - 1$ [KLO10]
- The exploration time is thus at most $(n - 1)^2$
- Best known lower bound is $2n - 3$
- Can we improve these bounds?

Some observations

In the second scenario (the agent **knows** the **dynamics**)

- The temporal diameter is bounded by $n - 1$ [KLO10]
- The exploration time is thus at most $(n - 1)^2$
- Best known lower bound is $2n - 3$
- Can we improve these bounds?
- We focus here on the study of n -node **T -interval-connected dynamic graphs** whose underlying graph is a **ring** A_n .

Our results

Result (1)

When the agent **knows the dynamics of the graph**,
 $2n - T - \Theta(1)$ time units are **necessary** and **sufficient**.

Our results

Result (1)

When the agent **knows the dynamics of the graph**, $2n - T - \Theta(1)$ time units are **necessary** and **sufficient**.

Result (2)

When the agent **does not know the dynamics of the graph**, $n + \frac{n}{\max\{1, T-1\}}(\delta - 1) \pm \Theta(\delta)$ time units are **necessary** and **sufficient**.

Theorem (Upper bound)

For every integer $n \geq 3$, $T \geq 1$ and $\delta \geq 1$, and for any direction dir , **Algorithm STUBBORN-TRAVERSAL**(dir) explores any δ -recurrent T -interval-connected dynamic graph based on A_n in time at most

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

Theorem (Upper bound)

For every integer $n \geq 3$, $T \geq 1$ and $\delta \geq 1$, and for any direction dir , **Algorithm STUBBORN-TRAVERSAL**(dir) explores any δ -recurrent T -interval-connected dynamic graph based on A_n in time at most

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

Theorem (Lower bound)

For every integer $n \geq 3$, $T \geq 1$, and $\delta \geq 1$, and for every agent (algorithm), there exists a δ -recurrent T -interval-connected dynamic graph based on A_n such that this agent needs at least

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

An idea of the proof

We obtain this complexity $n + \frac{n}{\max\{1, T-1\}}(\delta - 1) \pm \Theta(\delta)$ because

An idea of the proof

We obtain this complexity $n + \frac{n}{\max\{1, T-1\}}(\delta - 1) \pm \Theta(\delta)$ because

- An **edge cannot be absent** for more than $\delta - 1$ **consecutive time steps** (δ -recurrent)

An idea of the proof

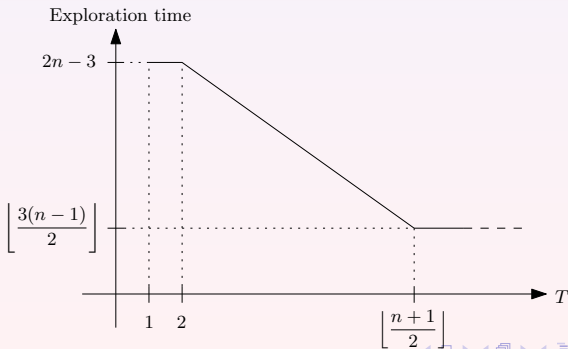
We obtain this complexity $n + \frac{n}{\max\{1, T-1\}}(\delta - 1) \pm \Theta(\delta)$ because

- An **edge cannot be absent** for more than $\delta - 1$ **consecutive time steps** (δ -recurrent)
- The **agent can traverse** at least $\max\{1, T - 1\}$ **edges** between two consecutive blocks at different nodes (T -interval-connectivity)

Theorem

For every integer $n \geq 3$ and $T \geq 1$, the **worst-case exploration time** of a **T -interval-connected** dynamic graph based on A_n is

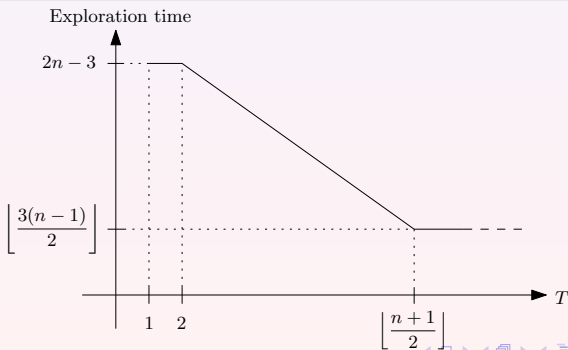
$$\begin{cases} 2n - 3 & \text{if } T = 1 \\ 2n - T - 1 & \text{if } 2 \leq T < (n + 1)/2 \\ \lfloor \frac{3(n-1)}{2} \rfloor & \text{if } T \geq (n + 1)/2 \end{cases}$$



Theorem

For every integer $n \geq 3$ and $T \geq 1$, the **worst-case exploration time** of a **T -interval-connected** dynamic graph based on A_n is

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

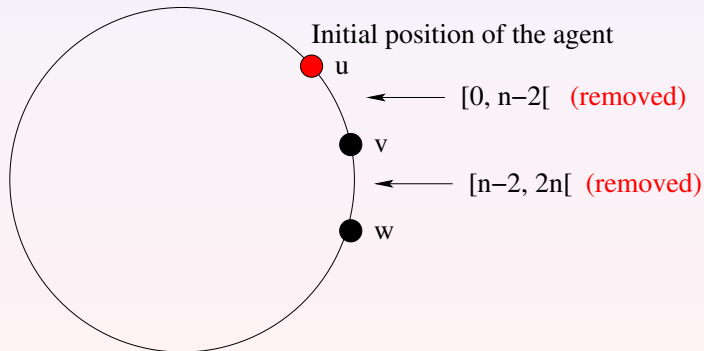


An idea of the proof

For every integer $n \geq 3$ and $T = 1$, there exists a 1-interval-connected dynamic graph based on A_n such that any agent (algorithm) needs at least $2n - 3$ time units to explore it.

An idea of the proof

For every integer $n \geq 3$ and $T = 1$, there exists a 1-interval-connected dynamic graph based on A_n such that any agent (algorithm) needs at least $2n - 3$ time units to explore it.



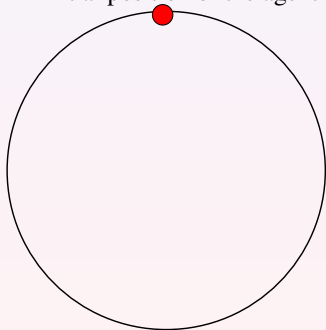
An idea of the proof

For every integer $n \geq 3$ and $T = 1$, and for every 1-interval-connected dynamic graph based on A_n , there exists an agent (algorithm) exploring this dynamic graph in time at most $2n - 3$ time units.

An idea of the proof

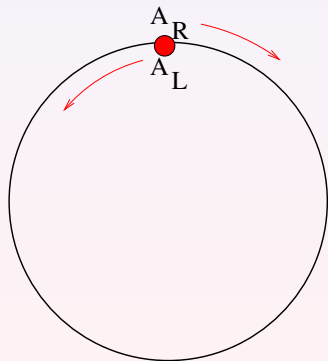
For every integer $n \geq 3$ and $T = 1$, and for every 1-interval-connected dynamic graph based on A_n , there exists an agent (algorithm) exploring this dynamic graph in time at most $2n - 3$ time units.

Initial position of the agent



An idea of the proof

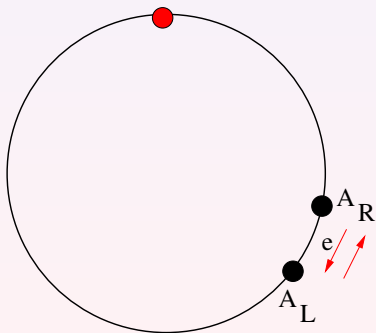
For every integer $n \geq 3$ and $T = 1$, and for every 1-interval-connected dynamic graph based on A_n , there exists an agent (algorithm) exploring this dynamic graph in time at most $2n - 3$ time units.



- We consider two algorithms (A_L , A_R).

An idea of the proof

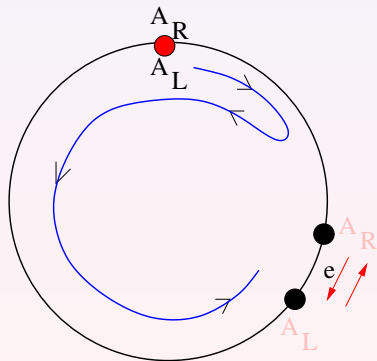
For every integer $n \geq 3$ and $T = 1$, and for every 1-interval-connected dynamic graph based on A_n , there exists an agent (algorithm) exploring this dynamic graph in time at most $2n - 3$ time units.



- We consider two algorithms (A_L , A_R).
- At each time step, at least one of them progresses.

An idea of the proof

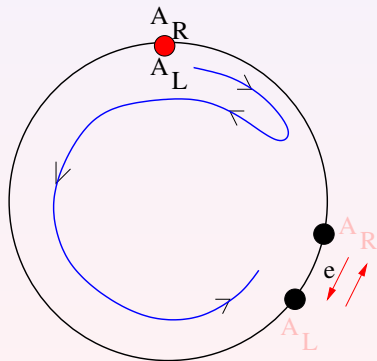
For every integer $n \geq 3$ and $T = 1$, and for every 1-interval-connected dynamic graph based on A_n , there exists an agent (algorithm) exploring this dynamic graph in time at most $2n - 3$ time units.



- We consider two algorithms (A_L , A_R).
- At each time step, at least one of them progresses.

An idea of the proof

For every integer $n \geq 3$ and $T = 1$, and for every 1-interval-connected dynamic graph based on A_n , there exists an agent (algorithm) exploring this dynamic graph in time at most $2n - 3$ time units.



- We consider two algorithms (A_L , A_R).
- At each time step, at least one of them progresses.
- At least one of them performs exploration within $2n$ time units.

Summary

- The time complexity of exploration of T -interval-connected dynamic graphs whose underlying graph is a ring

Summary

- The time complexity of exploration of T -interval-connected dynamic graphs whose underlying graph is a ring
 - The agent **knows** the dynamics of the graph
 - The agent **does not know** the dynamics of the graph

Summary

- The time complexity of exploration of T -interval-connected dynamic graphs whose underlying graph is a ring
 - The agent **knows** the dynamics of the graph
 - The agent **does not know** the dynamics of the graph
- Optimal bounds

Summary

- The time complexity of exploration of T -interval-connected dynamic graphs whose underlying graph is a ring
 - The agent knows the dynamics of the graph
 - The agent does not know the dynamics of the graph
- Optimal bounds

Further work

To study T -interval-connected dynamic graphs whose underlying graph belong to a family of graphs larger than rings.

Thank you for your attention

Thank you for your attention

