

On the Power of Waiting when Exploring Public Transportation Systems

David Ilcinkas Ahmed Wade

LaBRI, CNRS & Université de Bordeaux

MAC 2013

July 4th, 2013

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.

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.

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.

Exploration of dynamic graphs by a mobile agent

Dynamic graphs

- The new generation of network environments are **highly dynamic and evolve in time**.
- Researchers have begun to study these highly dynamic networks in the past few years.

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.

Exploration of dynamic graphs by a mobile agent

Dynamic graphs

- The new generation of network environments are **highly dynamic and evolve in time**.
- Researchers have begun to study these highly dynamic networks in the past few years.

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.

Exploration of dynamic graphs by a mobile agent

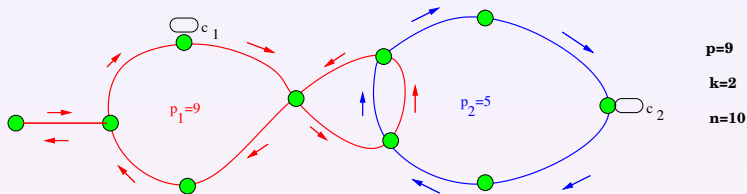
Dynamic graphs

- The new generation of network environments are **highly dynamic and evolve in time**.
- Researchers have begun to study these highly dynamic networks in the past few years.

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.

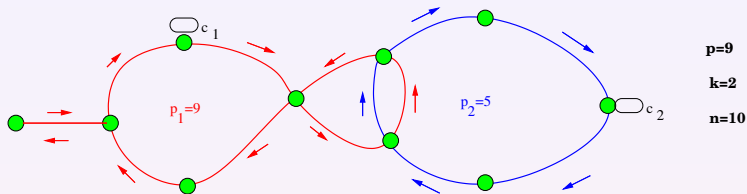
The model of PV-graphs



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.

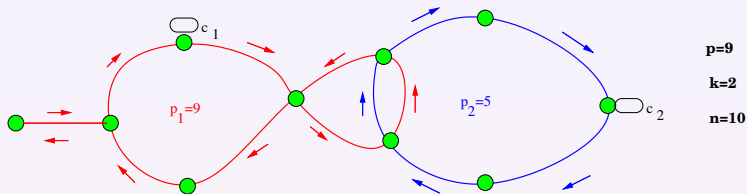
The model of PV-graphs



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.

The model of PV-graphs

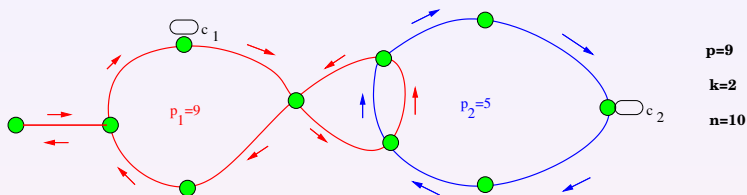
 $p=9$ $k=2$ $n=10$

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.

4/26

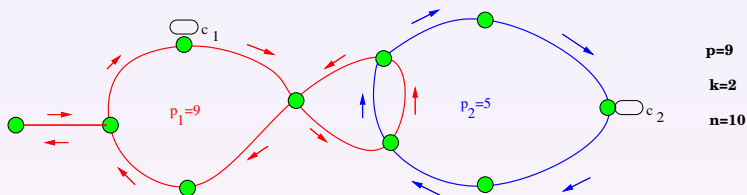
The model of PV-graphs



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 $R(c) = (s_{i_1}, \dots, s_{i_{|R(c)|}})$ of sites, called its route, in a periodic manner.

The model of 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.

Some related work

- [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

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.

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.

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.

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.

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.

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.

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.

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**.

Definitions

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.

Definitions

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.

Definitions

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]

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 (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 (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

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.

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.

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.

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.

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.

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

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.

Our contribution

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.

Our contribution

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.

Our contribution

Our contribution

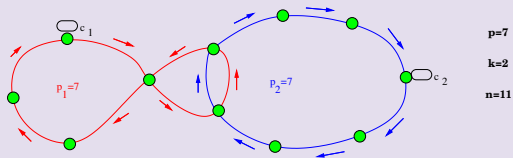
We extend the study of Flocchini, Mans and Santoro to the case when **the agent can leave a carrier** to stay at a site.

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

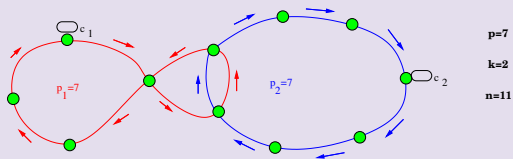
Connected PV-graph



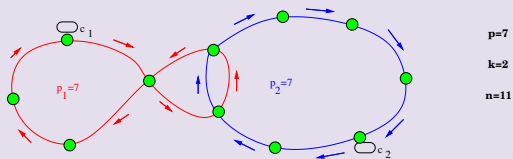
Highly-connected PV-graph

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

Sommaire

- 1 Introduction
- 2 Solvability**
- 3 Lower bounds
- 4 Upper bounds
- 5 Conclusion and perspectives

Solvability

An agent without information on the PV-graphs it has to explore **cannot explore** all PV-graphs.

Theorem

An agent cannot explore all labeled homogeneous highly-connected PV-graphs if it has no information on the PV-graphs it has to explore.

Solvability

An agent without information on the PV-graphs it has to explore **cannot explore** all PV-graphs.

Theorem

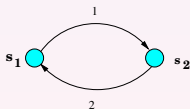
An agent cannot explore all **labeled homogeneous highly-connected** PV-graphs if it has no information on the PV-graphs it has to explore.

Solvability

An agent without information on the PV-graphs it has to explore **cannot explore** all PV-graphs.

Theorem

An agent cannot explore all **labeled homogeneous highly-connected** PV-graphs if it has no information on the PV-graphs it has to explore.



G_0

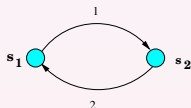
Stopping time in $G_0 = m$

Solvability

An agent without information on the PV-graphs it has to explore **cannot explore** all PV-graphs.

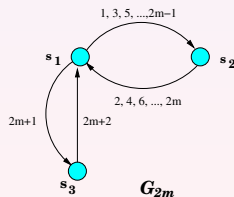
Theorem

An agent cannot explore all **labeled homogeneous highly-connected** PV-graphs if it has no information on the PV-graphs it has to explore.



G_0

Stopping time in $G_0 = m$



G_{2m}

Sommaire

- 1 Introduction
- 2 Solvability
- 3 Lower bounds**
 - On the number of moves
 - On the time
- 4 Upper bounds
- 5 Conclusion and perspectives

Lower bound on the number of moves

Theorem

- For any n , k , and p , there exists a **labeled homogeneous highly-connected** PV-graph such that **any agent** needs at least $\Omega(\min\{kp, np, n^2\})$ moves to explore it.
- This result holds even if the agent knows completely the PV-graph, and has unlimited memory.

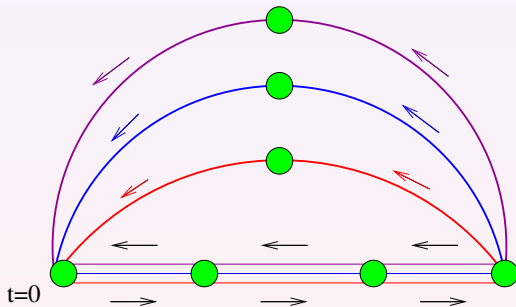
Lower bound on the number of moves

Theorem

- For any n , k , and p , there exists a **labeled homogeneous highly-connected** PV-graph such that **any agent** needs at least $\Omega(\min\{kp, np, n^2\})$ moves to explore it.
- This result holds even if the **agent knows** completely **the PV-graph**, and **has unlimited memory**.

Some part of the proof

There exists a PV-graph requiring $\Omega(\min\{kp, np, n^2\})$ moves.



$$k \geq n/2$$

$$p \geq 5n/2$$

$$n=7$$

Lower bound on time

The exploration time (time steps) \geq Number of moves

Lemma

- For any n, k, p , there exists a family $\mathcal{G}_{n,p,k}$ of labeled homogeneous PV-graphs such that, for any agent, there exists a PV-graph in this family which cannot be explored by the agent using less than $\Omega(np)$ time steps.

Lower bound on time

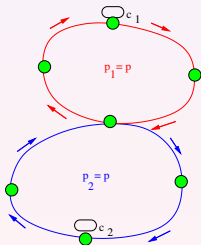
The exploration time (time steps) \geq Number of moves

Lemma

- For any n, k, p , there exists a family $\mathcal{G}_{n,p,k}$ of **labeled homogeneous** PV-graphs such that, for **any agent**, there exists a **PV-graph** in this family which cannot be explored by the agent using less than $\Omega(np)$ time steps.

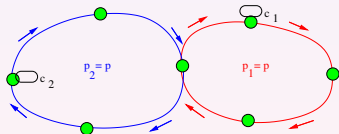
An idea of the proof

For any agent, there exists a PV-graph $\in \mathcal{G}_{n,p,k}$ which cannot be explored by the agent using less than $\Omega(np)$ time steps.



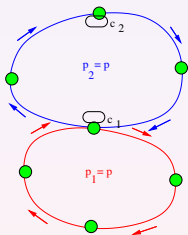
An idea of the proof

For any agent, there exists a PV-graph $\in \mathcal{G}_{n,p,k}$ which cannot be explored by the agent using less than $\Omega(np)$ time steps.



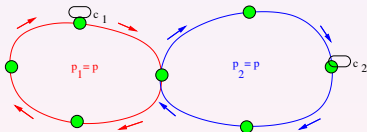
An idea of the proof

For any agent, there exists a PV-graph $\in \mathcal{G}_{n,p,k}$ which cannot be explored by the agent using less than $\Omega(np)$ time steps.



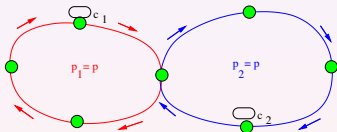
An idea of the proof

For any agent, there exists a PV-graph $\in \mathcal{G}_{n,p,k}$ which cannot be explored by the agent using less than $\Omega(np)$ time steps.



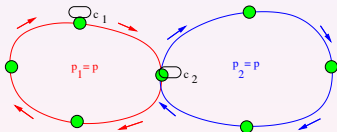
An idea of the proof

For any agent, there exists a PV-graph $\in \mathcal{G}_{n,p,k}$ which cannot be explored by the agent using less than $\Omega(np)$ time steps.



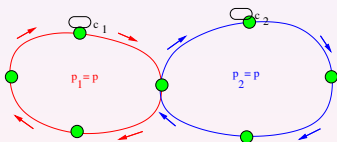
An idea of the proof

For any agent, there exists a PV-graph $\in \mathcal{G}_{n,p,k}$ which cannot be explored by the agent using less than $\Omega(np)$ time steps.



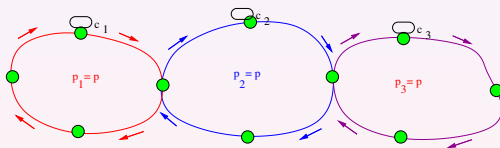
An idea of the proof

For any agent, there exists a PV-graph $\in \mathcal{G}_{n,p,k}$ which cannot be explored by the agent using less than $\Omega(np)$ time steps.



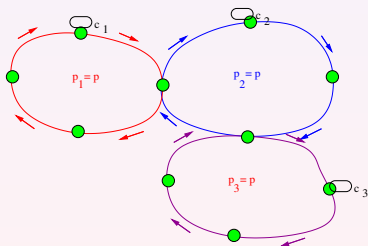
An idea of the proof

For any agent, there exists a PV-graph $\in \mathcal{G}_{n,p,k}$ which cannot be explored by the agent using less than $\Omega(np)$ time steps.



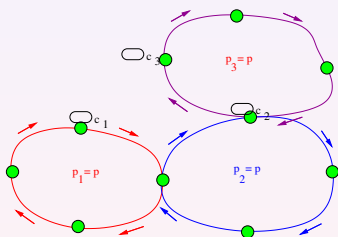
An idea of the proof

For any agent, there exists a PV-graph $\in \mathcal{G}_{n,p,k}$ which cannot be explored by the agent using less than $\Omega(np)$ time steps.



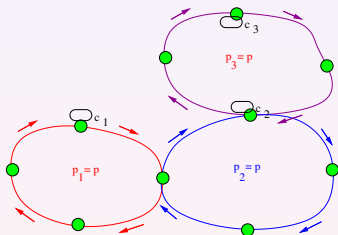
An idea of the proof

For any agent, there exists a PV-graph $\in \mathcal{G}_{n,p,k}$ which cannot be explored by the agent using less than $\Omega(np)$ time steps.



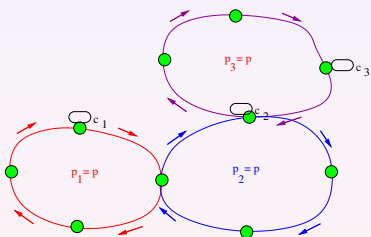
An idea of the proof

For any agent, there exists a PV-graph $\in \mathcal{G}_{n,p,k}$ which cannot be explored by the agent using less than $\Omega(np)$ time steps.



An idea of the proof

For any agent, there exists a PV-graph $\in \mathcal{G}_{n,p,k}$ which cannot be explored by the agent using less than $\Omega(np)$ time steps.



Sommaire

- 1 Introduction
- 2 Solvability
- 3 Lower bounds
- 4 Upper bounds**
 - Our algorithm
 - Complexities
- 5 Conclusion and perspectives

- B is the upper bound given to the agent on the longest period p

Algorithm EXPLORE-WITH-WAIT correctly explores and maps in finite time any PV-graph, even anonymous, provided that an upper bound on the maximum period is known.

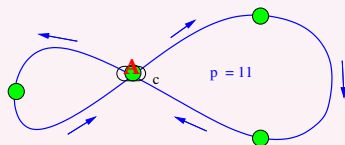


- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

- B is the upper bound given to the agent on the longest period p

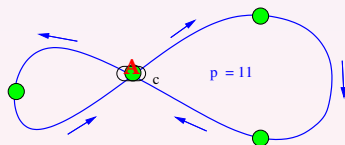
Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

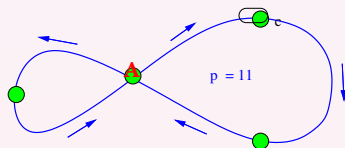
1



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

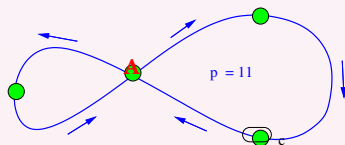
1



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

1

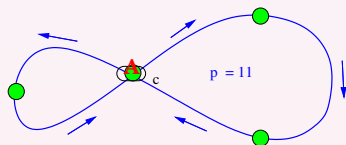


- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

1

1

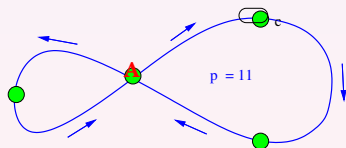


- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

1

1

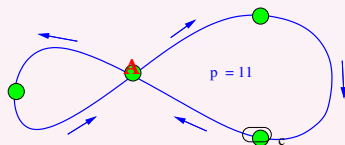


- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

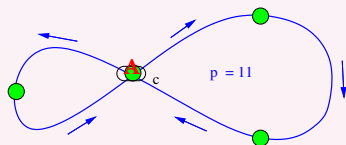
1

1



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.



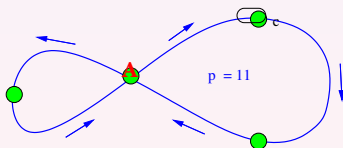
- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

1

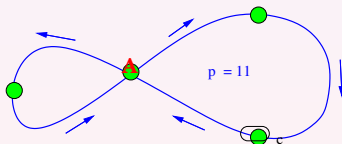
1

1



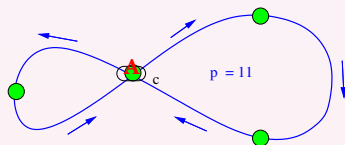
- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.



- B is the upper bound given to the agent on the longest period p

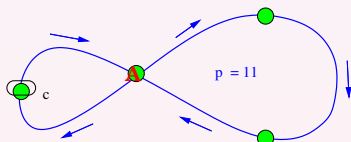
Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

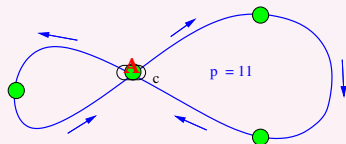
1			1			1													
---	--	--	---	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--



- B is the upper bound given to the agent on the longest period p

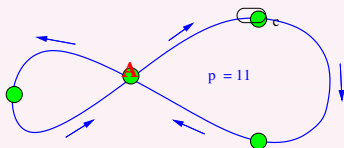
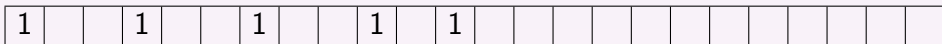
Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

1			1			1			1										
---	--	--	---	--	--	---	--	--	---	--	--	--	--	--	--	--	--	--	--



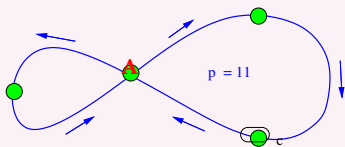
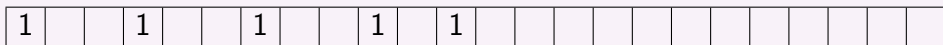
- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.



- B is the upper bound given to the agent on the longest period p

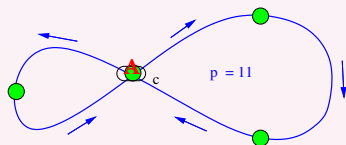
Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.



- B is the upper bound given to the agent on the longest period p

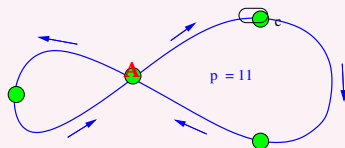
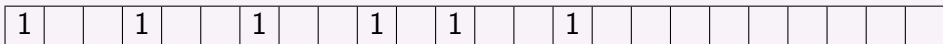
Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

1		1		1		1		1		1									
---	--	---	--	---	--	---	--	---	--	---	--	--	--	--	--	--	--	--	--



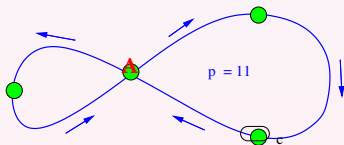
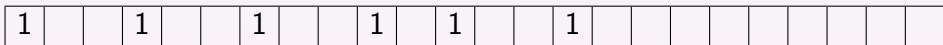
- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.



- B is the upper bound given to the agent on the longest period p

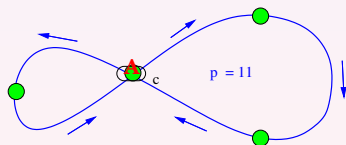
Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.



- B is the upper bound given to the agent on the longest period p

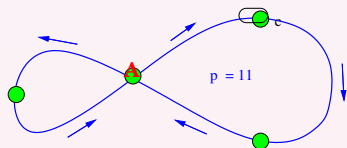
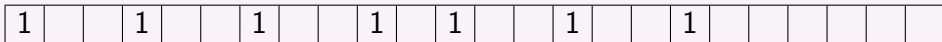
Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

1			1			1			1			1			1							
---	--	--	---	--	--	---	--	--	---	--	--	---	--	--	---	--	--	--	--	--	--	--



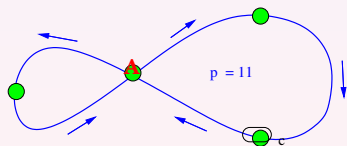
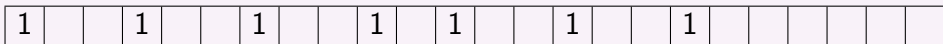
- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.



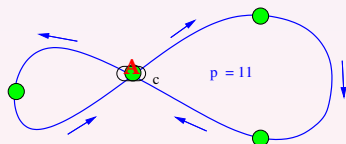
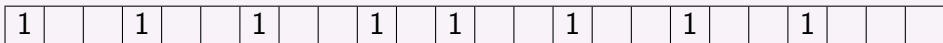
- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.



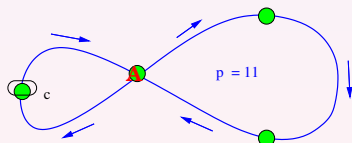
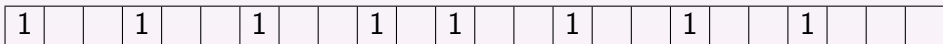
- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and maps in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.



- B is the upper bound given to the agent on the longest period p

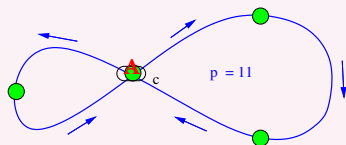
Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

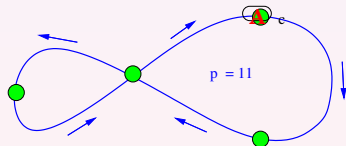
1			1			1			1			1			1			1			1			1
---	--	--	---	--	--	---	--	--	---	--	--	---	--	--	---	--	--	---	--	--	---	--	--	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

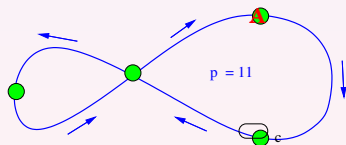
1	2	1		1		1	1		1		1		1	1
---	---	---	--	---	--	---	---	--	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

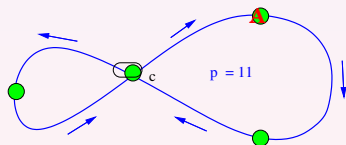
1	2	1		1		1	1		1		1		1	1
---	---	---	--	---	--	---	---	--	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

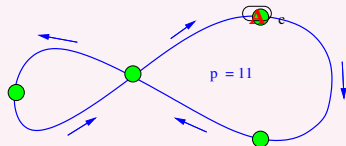
1	2	1		1		1	1		1		1		1	1
---	---	---	--	---	--	---	---	--	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

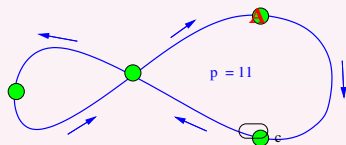
1	2		1	2		1			1	1			1			1		1		1
---	---	--	---	---	--	---	--	--	---	---	--	--	---	--	--	---	--	---	--	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

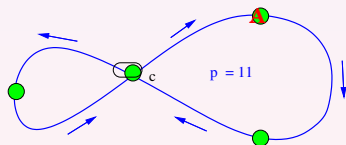
1	2		1	2		1			1		1			1			1		1
---	---	--	---	---	--	---	--	--	---	--	---	--	--	---	--	--	---	--	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

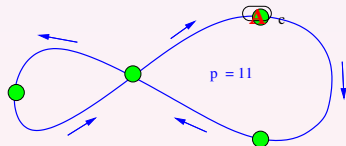
1	2		1	2		1			1		1			1			1		1
---	---	--	---	---	--	---	--	--	---	--	---	--	--	---	--	--	---	--	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

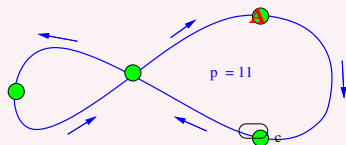
1	2		1	2		1	2		1		1			1			1		1		1		1
---	---	--	---	---	--	---	---	--	---	--	---	--	--	---	--	--	---	--	---	--	---	--	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

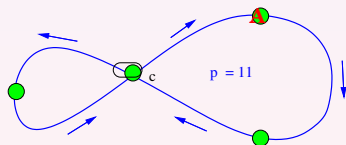
1	2		1	2		1	2		1		1			1			1		1
---	---	--	---	---	--	---	---	--	---	--	---	--	--	---	--	--	---	--	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

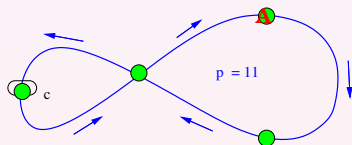
1	2		1	2		1	2		1		1			1			1		1		1
---	---	--	---	---	--	---	---	--	---	--	---	--	--	---	--	--	---	--	---	--	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

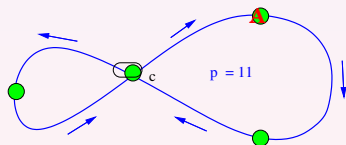
1	2		1	2		1	2		1		1			1			1		1		1			1		1
---	---	--	---	---	--	---	---	--	---	--	---	--	--	---	--	--	---	--	---	--	---	--	--	---	--	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

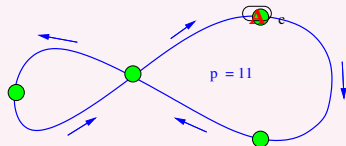
1	2		1	2		1	2		1		1			1			1		1			1		1
---	---	--	---	---	--	---	---	--	---	--	---	--	--	---	--	--	---	--	---	--	--	---	--	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

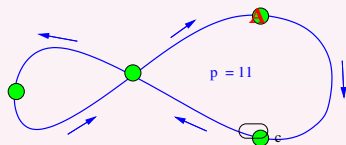
1	2		1	2		1	2		1		1	2		1		1		1		1
---	---	--	---	---	--	---	---	--	---	--	---	---	--	---	--	---	--	---	--	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

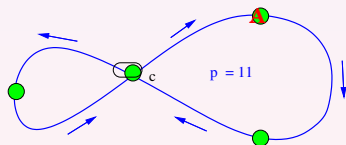
1	2		1	2		1	2		1		1	2		1			1			1		1
---	---	--	---	---	--	---	---	--	---	--	---	---	--	---	--	--	---	--	--	---	--	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

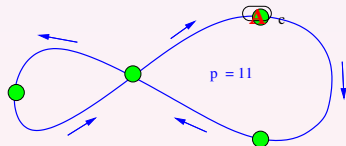
1	2		1	2		1	2		1		1	2		1			1			1		1
---	---	--	---	---	--	---	---	--	---	--	---	---	--	---	--	--	---	--	--	---	--	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

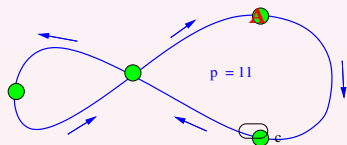
1	2		1	2		1	2		1		1	2		1			1		1
---	---	--	---	---	--	---	---	--	---	--	---	---	--	---	--	--	---	--	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

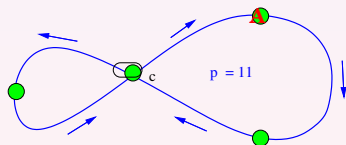
1	2		1	2		1	2		1		1	2		1			1		1	
---	---	--	---	---	--	---	---	--	---	--	---	---	--	---	--	--	---	--	---	--



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

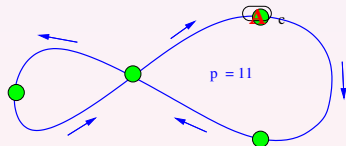
1	2		1	2		1	2		1		1	2		1			1		1
---	---	--	---	---	--	---	---	--	---	--	---	---	--	---	--	--	---	--	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

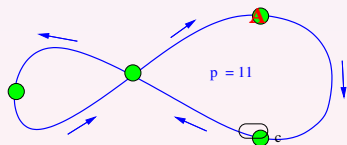
1	2		1	2		1	2		1		1	2		1	2		1		1	
---	---	--	---	---	--	---	---	--	---	--	---	---	--	---	---	--	---	--	---	--



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

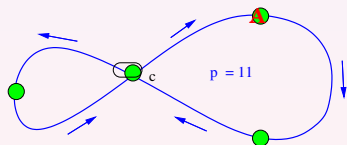
1	2		1	2		1	2		1		1	2		1	2		1		1	
---	---	--	---	---	--	---	---	--	---	--	---	---	--	---	---	--	---	--	---	--



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

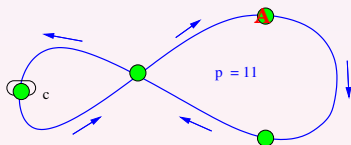
1	2		1	2		1	2		1		1	2		1	2		1		1	
---	---	--	---	---	--	---	---	--	---	--	---	---	--	---	---	--	---	--	---	--



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

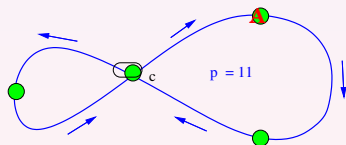
1	2		1	2		1	2		1		1	2		1	2		1		1	
---	---	--	---	---	--	---	---	--	---	--	---	---	--	---	---	--	---	--	---	--



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

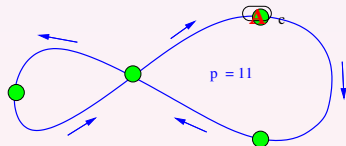
1	2		1	2		1	2		1		1	2		1	2		1		1	
---	---	--	---	---	--	---	---	--	---	--	---	---	--	---	---	--	---	--	---	--



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

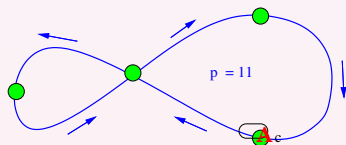
1	2		1	2		1	2		1		1	2		1	2		1		1	2
---	---	--	---	---	--	---	---	--	---	--	---	---	--	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

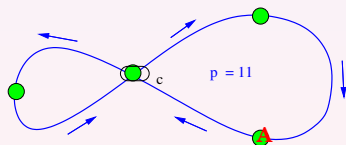
1	2	3	1	2		1	2		1		1	2		1	2		1		1	2
---	---	---	---	---	--	---	---	--	---	--	---	---	--	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

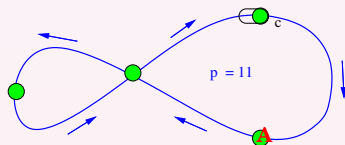
1	2	3	1	2		1	2		1		1	2		1	2		1		1	2
---	---	---	---	---	--	---	---	--	---	--	---	---	--	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

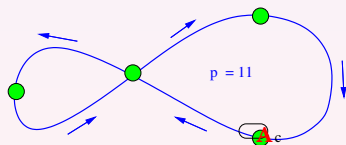
1	2	3	1	2		1	2		1		1	2		1	2		1		1	2
---	---	---	---	---	--	---	---	--	---	--	---	---	--	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

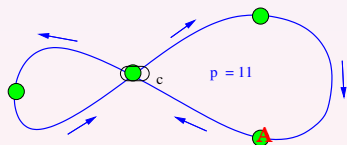
1	2	3	1	2	3	1	2		1		1	2		1	2		1		1	2
---	---	---	---	---	---	---	---	--	---	--	---	---	--	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

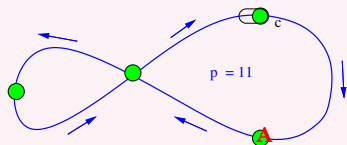
1	2	3	1	2	3	1	2		1		1	2		1	2		1		1	2
---	---	---	---	---	---	---	---	--	---	--	---	---	--	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

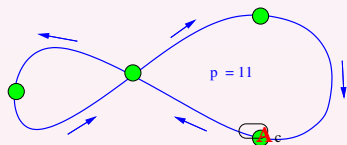
1	2	3	1	2	3	1	2		1		1	2		1	2		1		1	2
---	---	---	---	---	---	---	---	--	---	--	---	---	--	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

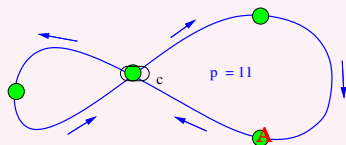
1	2	3	1	2	3	1	2	3	1		1	2		1	2		1	2		1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	--	---	---	--	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

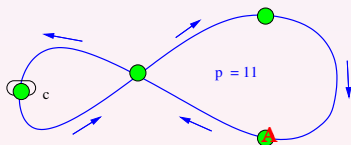
1	2	3	1	2	3	1	2	3	1		1	2		1	2		1	2		1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	--	---	---	--	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

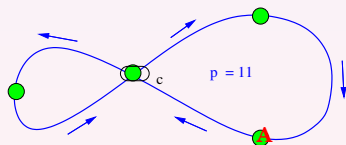
1	2	3	1	2	3	1	2	3	1		1	2		1	2		1	2		1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	--	---	---	--	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

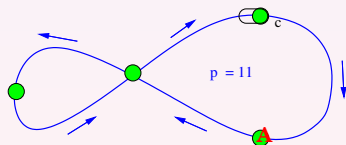
1	2	3	1	2	3	1	2	3	1		1	2		1	2		1	2		1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	--	---	---	--	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

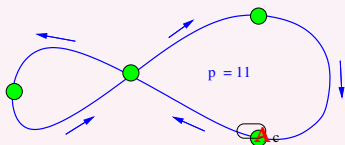
1	2	3	1	2	3	1	2	3	1		1	2		1	2		1	2		1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	--	---	---	--	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

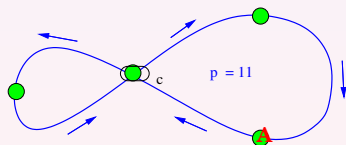
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2		1	2		1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	--	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

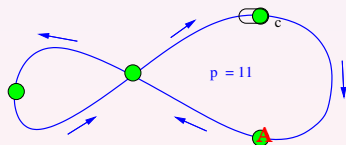
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2		1	2		1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	--	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

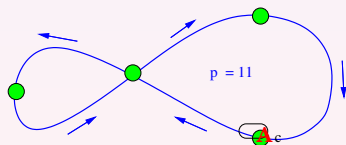
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2		1	2		1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	--	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

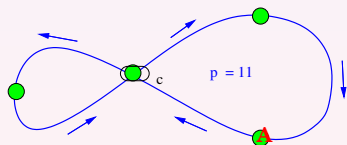
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2		1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

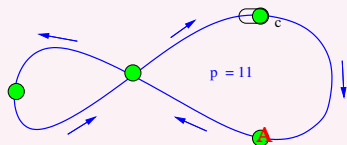
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2		1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

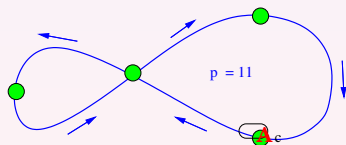
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2		1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	--	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

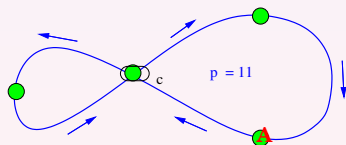
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2	3	1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

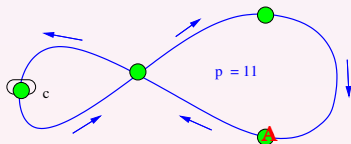
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2	3	1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

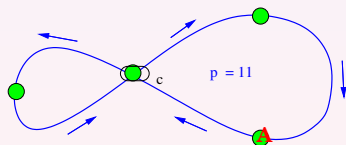
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2	3	1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

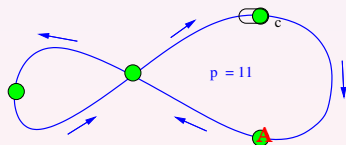
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2	3	1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

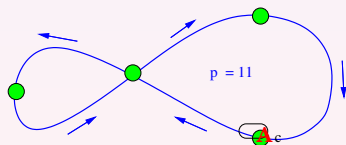
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2	3	1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

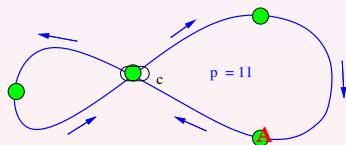
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2	3	1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

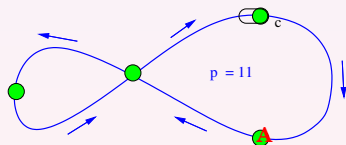
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2	3	1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

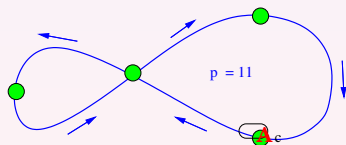
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2	3	1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

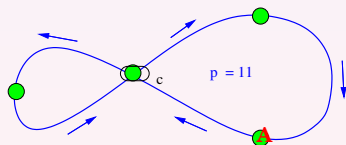
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2	3	1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

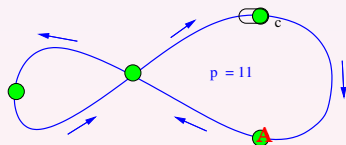
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2	3	1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

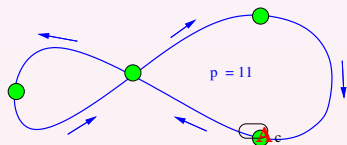
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2	3	1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

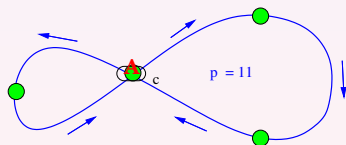
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2	3	1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

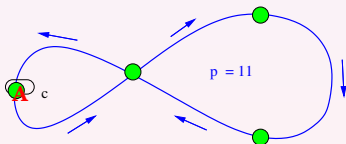
1	2	3	1	2	3	1	2	3	1		1	2	3	1	2	3	1	2	3	1		1	2
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

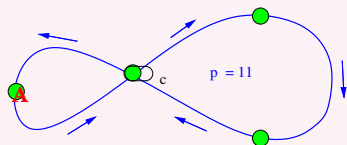
1	2	3	1	2	3	1	2	3	1	4	1	2	3	1	2	3	1	2	3	1			1	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

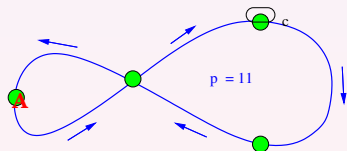
1	2	3	1	2	3	1	2	3	1	4	1	2	3	1	2	3	1	2	3	1			1	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

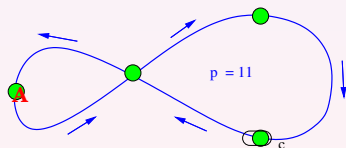
1	2	3	1	2	3	1	2	3	1	4	1	2	3	1	2	3	1	2	3	1			1	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

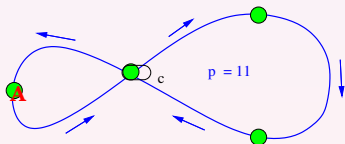
1	2	3	1	2	3	1	2	3	1	4	1	2	3	1	2	3	1	2	3	1			1	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

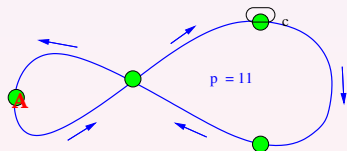
1	2	3	1	2	3	1	2	3	1	4	1	2	3	1	2	3	1	2	3	1			1	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

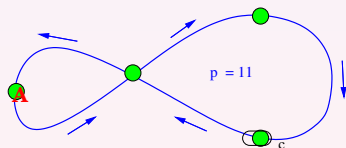
1	2	3	1	2	3	1	2	3	1	4	1	2	3	1	2	3	1	2	3	1			1	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

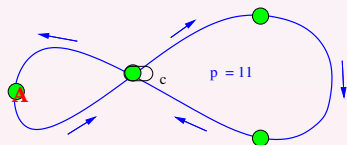
1	2	3	1	2	3	1	2	3	1	4	1	2	3	1	2	3	1	2	3	1			1	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

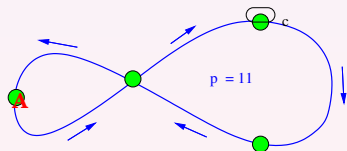
1	2	3	1	2	3	1	2	3	1	4	1	2	3	1	2	3	1	2	3	1			1	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

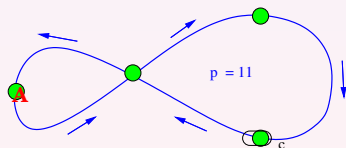
1	2	3	1	2	3	1	2	3	1	4	1	2	3	1	2	3	1	2	3	1			1	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

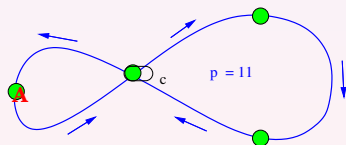
1	2	3	1	2	3	1	2	3	1	4	1	2	3	1	2	3	1	2	3	1			1	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

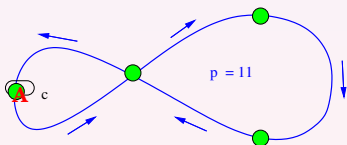
1	2	3	1	2	3	1	2	3	1	4	1	2	3	1	2	3	1	2	3	1			1	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	---	---



- B is the upper bound given to the agent on the longest period p

Algorithm **EXPLORE-WITH-WAIT** correctly explores and **maps** in finite time any PV-graph, **even anonymous**, provided that an upper bound on the maximum period is known.

1	2	3	1	2	3	1	2	3	1	4	1	2	3	1	2	3	1	2	3	1	4	1	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



Complexities

With the algorithm **EXPLORE-WITH-WAIT**

- The agent performs at most $O(\min\{kp, np, n^2\})$ moves to explore any PV-graph.
- The agent explores any in at most $O(nB)$ time steps.

Complexities

With the algorithm **EXPLORE-WITH-WAIT**

- The agent performs at most $O(\min\{kp, np, n^2\})$ moves to explore any PV-graph.
- The agent explores any in at most $O(nB)$ time steps.

Sommaire

- 1 Introduction
- 2 Solvability
- 3 Lower bounds
- 4 Upper bounds
- 5 Conclusion and perspectives**

Conclusion and perspectives

Summary

- Reduction of the optimal number of moves by a multiplicative factor of at least $\Theta(p)$.
- Reduction of the time complexity from $\Theta(kp^2)$ to $\Theta(np)$.
- Optimal algorithm, which also allows to map the PV-graph.

Perspectives

- Bound on the agent's memory
- Use of several agents
- Study of other models of dynamic graphs

Conclusion and perspectives

Summary

- Reduction of the optimal number of moves by a multiplicative factor of at least $\Theta(p)$.
- Reduction of the time complexity from $\Theta(kp^2)$ to $\Theta(np)$.
- Optimal algorithm, which also allows to map the PV-graph.

Perspectives

- Bound on the agent's memory
- Use of several agents
- Study of other models of dynamic graphs

Conclusion and perspectives

Summary

- Reduction of the optimal number of moves by a multiplicative factor of at least $\Theta(p)$.
- Reduction of the time complexity from $\Theta(kp^2)$ to $\Theta(np)$.
- Optimal algorithm, which also allows to map the PV-graph.

Perspectives

- Bound on the agent's memory
- Use of several agents
- Study of other models of dynamic graphs

Conclusion and perspectives

Summary

- Reduction of the optimal number of moves by a multiplicative factor of at least $\Theta(p)$.
- Reduction of the time complexity from $\Theta(kp^2)$ to $\Theta(np)$.
- Optimal algorithm, which also allows to map the PV-graph.

Perspectives

- Bound on the agent's memory
- Use of several agents
- Study of other models of dynamic graphs

Conclusion and perspectives

Summary

- Reduction of the optimal number of moves by a multiplicative factor of at least $\Theta(p)$.
- Reduction of the time complexity from $\Theta(kp^2)$ to $\Theta(np)$.
- Optimal algorithm, which also allows to map the PV-graph.

Perspectives

- Bound on the agent's memory
- Use of several agents
- Study of other models of dynamic graphs

Conclusion and perspectives

Summary

- Reduction of the optimal number of moves by a multiplicative factor of at least $\Theta(p)$.
- Reduction of the time complexity from $\Theta(kp^2)$ to $\Theta(np)$.
- Optimal algorithm, which also allows to map the PV-graph.

Perspectives

- Bound on the agent's memory
- Use of several agents
- Study of other models of dynamic graphs

Thank you for your attention!