

Distributed Algorithms in Dynamic Networks

(a short introduction)

Arnaud Casteigts

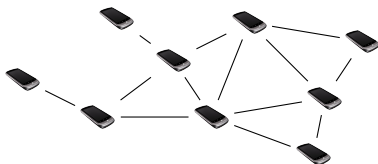
The content of this talk can be found in

<https://arxiv.org/abs/1807.07801>

February 8, 2019

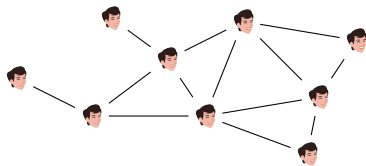
Networks?

- Set of **nodes** V (a.k.a. entities, vertices)
 - Set of **links** E among them (a.k.a. relations, edges)
- A **network** (or graph) $G = (V, E)$



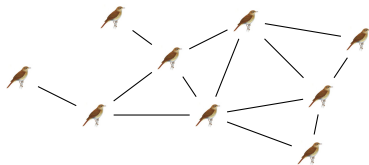
Networks?

- Set of nodes V (a.k.a. entities, vertices)
 - Set of links E among them (a.k.a. relations, edges)
- A network (or graph) $G = (V, E)$



Networks?

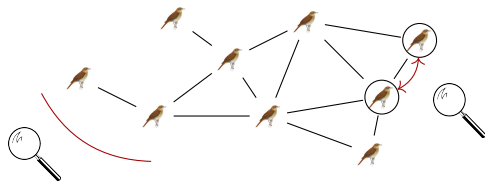
- Set of **nodes** V (a.k.a. entities, vertices)
 - Set of **links** E among them (a.k.a. relations, edges)
- A **network** (or graph) $G = (V, E)$



Networks?

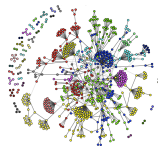
- Set of nodes V (a.k.a. entities, vertices)
- Set of links E among them (a.k.a. relations, edges)

→ A network (or graph) $G = (V, E)$



Complex networks

- compute global metrics
- explain and reproduce phenomena



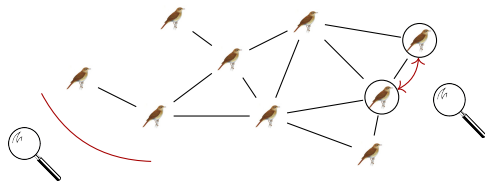
Communication networks

- design interactions among entities
- study what can be done *from within*



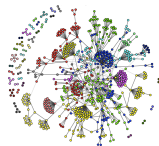
Networks?

- Set of nodes V (a.k.a. entities, vertices)
 - Set of links E among them (a.k.a. relations, edges)
- A network (or graph) $G = (V, E)$



Complex networks

- compute global metrics
- explain and reproduce phenomena

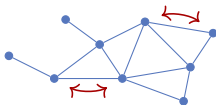


Communication networks

- design interactions among entities
- study what can be done *from within*
- distributed algorithms...



Distributed Algorithms



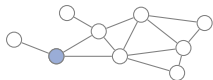
Collaboration of distinct entities to perform a common task.

No centralization available. No global knowledge.

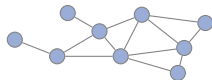
(Think globally, act locally)

Examples of problems

Broadcast

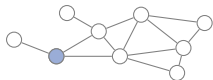


Propagating a piece of information from one node to all others.

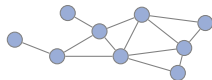


Examples of problems

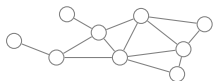
Broadcast



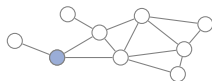
Propagating a piece of information from one node to all others.



Election

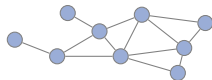
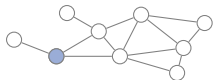


Distinguishing exactly one node among all.



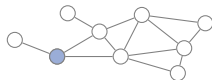
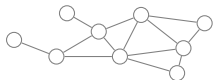
Examples of problems

Broadcast



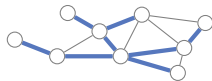
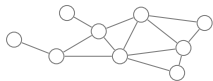
Propagating a piece of information from one node to all others.

Election



Distinguishing exactly one node among all.

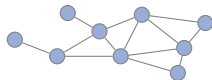
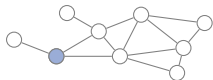
Spanning tree



Selecting a cycle-free set of edges that interconnects all nodes.

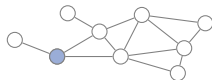
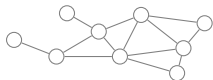
Examples of problems

Broadcast



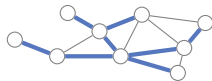
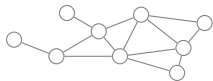
Propagating a piece of information from one node to all others.

Election



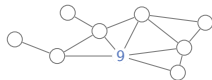
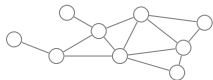
Distinguishing exactly one node among all.

Spanning tree



Selecting a cycle-free set of edges that interconnects all nodes.

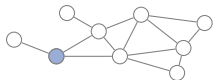
Counting



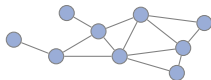
Determining how many participants there are.

Examples of problems

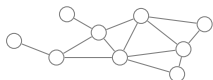
Broadcast



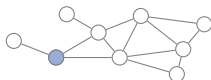
Propagating a piece of information from one node to all others.



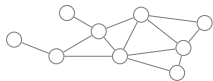
Election



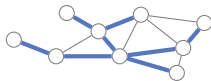
Distinguishing exactly one node among all.



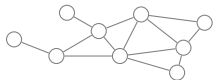
Spanning tree



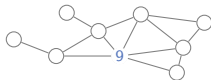
Selecting a cycle-free set of edges that interconnects all nodes.



Counting

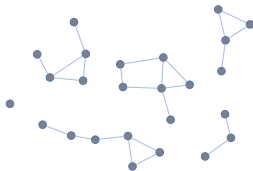


Determining how many participants there are.



Consensus, naming, routing, exploration, ...

Dynamic Networks



Dynamic networks?

In fact, *highly* dynamic networks.

Ex:



Dynamic networks?

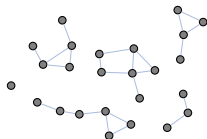
In fact, *highly* dynamic networks.

Ex:



How changes are perceived?

- Faults and Failures? —
- Nature of the system. Change is normal.
- Partitioned network



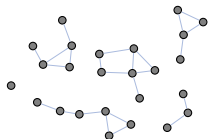
Dynamic networks?

In fact, *highly* dynamic networks.



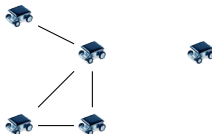
How changes are perceived?

- Faults and Failures? —
- Nature of the system. Change is normal.
- Partitioned network



Example of scenario

(say, exploration by mobile robots)



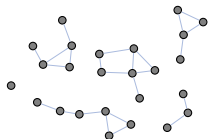
Dynamic networks?

In fact, *highly* dynamic networks.



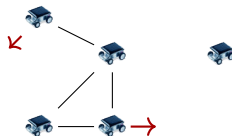
How changes are perceived?

- Faults and Failures? —
- Nature of the system. Change is normal.
- Partitioned network



Example of scenario

(say, exploration by mobile robots)



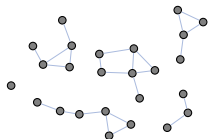
Dynamic networks?

In fact, *highly* dynamic networks.



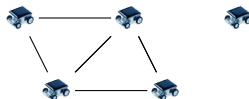
How changes are perceived?

- Faults and Failures? —
- Nature of the system. Change is normal.
- Partitioned network



Example of scenario

(say, exploration by mobile robots)



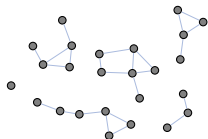
Dynamic networks?

In fact, *highly* dynamic networks.



How changes are perceived?

- Faults and Failures? —
- Nature of the system. Change is normal.
- Partitioned network



Example of scenario

(say, exploration by mobile robots)



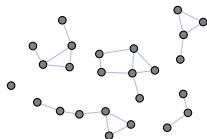
Dynamic networks?

In fact, *highly* dynamic networks.



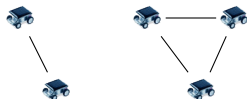
How changes are perceived?

- Faults and Failures? —
- Nature of the system. Change is normal.
- Partitioned network



Example of scenario

(say, exploration by mobile robots)



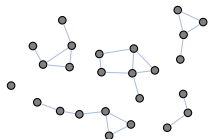
Dynamic networks?

In fact, *highly* dynamic networks.



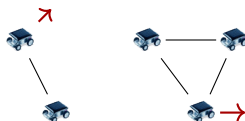
How changes are perceived?

- Faults and Failures? —
- Nature of the system. Change is normal.
- Partitioned network



Example of scenario

(say, exploration by mobile robots)



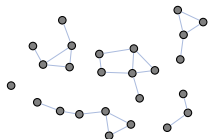
Dynamic networks?

In fact, *highly* dynamic networks.



How changes are perceived?

- Faults and Failures? —
- Nature of the system. Change is normal.
- Partitioned network



Example of scenario

(say, exploration by mobile robots)



Dynamic Graphs

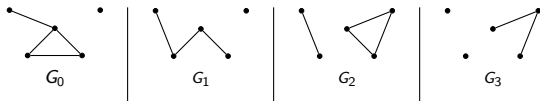
Also called *time-varying graphs*, *evolving graphs*, *temporal graphs*, etc.

Dynamic Graphs

Also called *time-varying graphs*, *evolving graphs*, *temporal graphs*, etc.

As a sequence

Sequence of static graphs $\mathcal{G} = G_0, G_1, \dots$ [+table of dates]

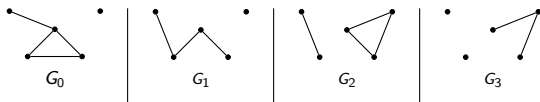


Dynamic Graphs

Also called *time-varying graphs*, *evolving graphs*, *temporal graphs*, etc.

As a sequence

Sequence of static graphs $\mathcal{G} = G_0, G_1, \dots$ [+table of dates]

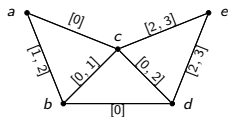


With a presence function

$$\mathcal{G} = (V, E, \mathcal{T}, \rho),$$

with ρ being a *presence function*

$$\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$$

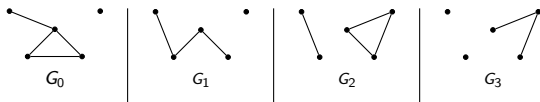


Dynamic Graphs

Also called *time-varying graphs*, *evolving graphs*, *temporal graphs*, etc.

As a sequence

Sequence of static graphs $\mathcal{G} = G_0, G_1, \dots$ [+table of dates]

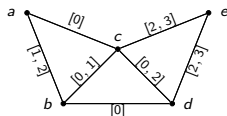



With a presence function

$\mathcal{G} = (V, E, \mathcal{T}, \rho)$,

with ρ being a *presence function*

$$\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$$



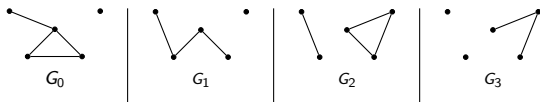
→ Both are theoretically equivalent if $\mathcal{T} \subseteq \mathbb{N}$ (e.g. not like this )

Dynamic Graphs

Also called *time-varying graphs*, *evolving graphs*, *temporal graphs*, etc.

As a sequence

Sequence of static graphs $\mathcal{G} = G_0, G_1, \dots$ [+table of dates]

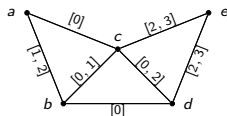



With a presence function

$\mathcal{G} = (V, E, \mathcal{T}, \rho)$,

with ρ being a *presence function*

$$\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$$



→ Both are theoretically equivalent if $\mathcal{T} \subseteq \mathbb{N}$ (e.g. not like this )

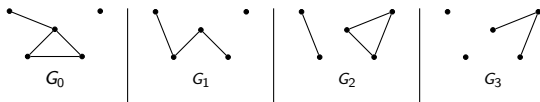
→ Further extensions possible (latency function, node-presence function, ...)

Dynamic Graphs

Also called *time-varying graphs*, *evolving graphs*, *temporal graphs*, etc.

As a sequence

Sequence of static graphs $\mathcal{G} = G_0, G_1, \dots$ [+table of dates]

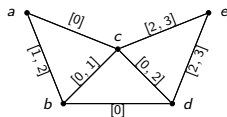



With a presence function

$\mathcal{G} = (V, E, \mathcal{T}, \rho)$,

with ρ being a *presence function*

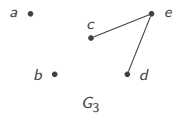
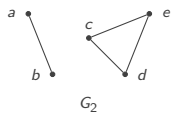
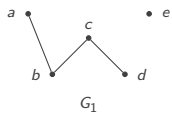
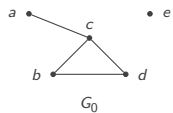
$$\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$$



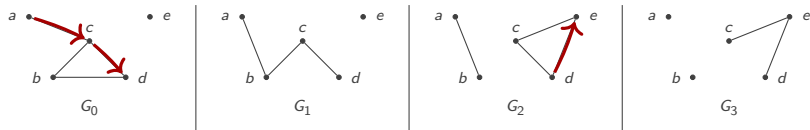
→ Both are theoretically equivalent if $\mathcal{T} \subseteq \mathbb{N}$ (e.g. not like this )

→ Further extensions possible (latency function, node-presence function, ...)

Basic graph concepts



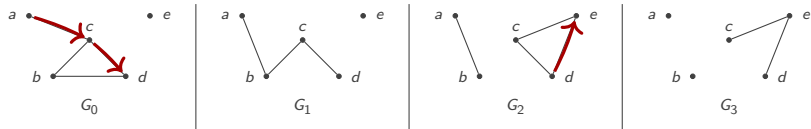
Basic graph concepts



→ Paths become temporal (*journey*)

Ex: $((ac, t_1), (cd, t_2), (de, t_3))$ with $t_{i+1} \geq t_i$ and $\rho(e_i, t_i) = 1$

Basic graph concepts

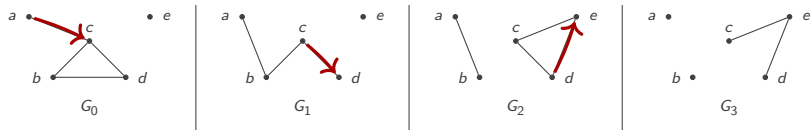


→ Paths become temporal (*journey*)

Ex: $((ac, t_1), (cd, t_2), (de, t_3))$ with $t_{i+1} \geq t_i$ and $\rho(e_i, t_i) = 1$

→ *Strict* journeys vs. *non-strict* journeys. (Relevant in discrete time only.)

Basic graph concepts

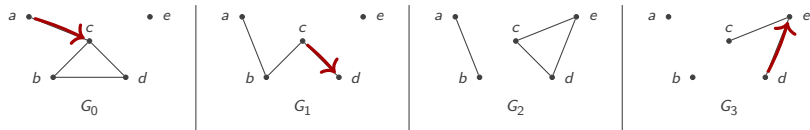


→ Paths become temporal (*journey*)

Ex: $((ac, t_1), (cd, t_2), (de, t_3))$ with $t_{i+1} \geq t_i$ and $\rho(e_i, t_i) = 1$

→ *Strict* journeys vs. *non-strict* journeys. (Relevant in discrete time only.)

Basic graph concepts

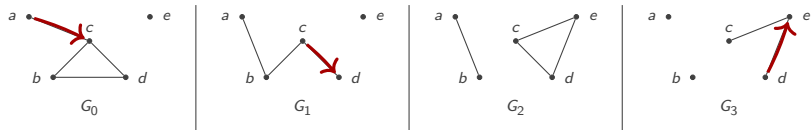


→ Paths become temporal (*journey*)

Ex: $((ac, t_1), (cd, t_2), (de, t_3))$ with $t_{i+1} \geq t_i$ and $\rho(e_i, t_i) = 1$

→ *Strict* journeys vs. *non-strict* journeys. (Relevant in discrete time only.)

Basic graph concepts



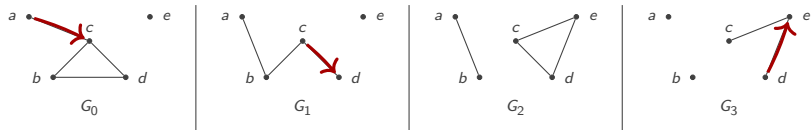
→ Paths become temporal (*journey*)

Ex: $((ac, t_1), (cd, t_2), (de, t_3))$ with $t_{i+1} \geq t_i$ and $\rho(e_i, t_i) = 1$

→ *Strict* journeys vs. *non-strict* journeys. (Relevant in discrete time only.)

→ *Temporal connectivity*. Not symmetrical! (e.g. $a \rightsquigarrow e$, but $e \not\rightsquigarrow a$)

Basic graph concepts



→ Paths become temporal (*journey*)

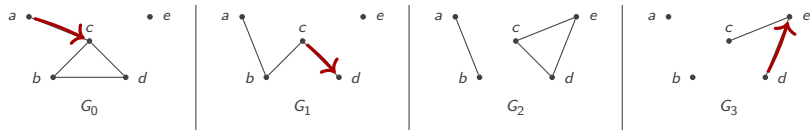
Ex: $((ac, t_1), (cd, t_2), (de, t_3))$ with $t_{i+1} \geq t_i$ and $\rho(e_i, t_i) = 1$

→ *Strict* journeys vs. *non-strict* journeys. (Relevant in discrete time only.)

→ *Temporal connectivity*. Not symmetrical! (e.g. $a \rightsquigarrow e$, but $e \not\rightsquigarrow a$)

→ Footprint, snapshots, etc.

Basic graph concepts



→ Paths become temporal (*journey*)

Ex: $((ac, t_1), (cd, t_2), (de, t_3))$ with $t_{i+1} \geq t_i$ and $\rho(e_i, t_i) = 1$

→ *Strict* journeys vs. *non-strict* journeys. (Relevant in discrete time only.)

→ *Temporal connectivity*. Not symmetrical! (e.g. $a \rightsquigarrow e$, but $e \not\rightsquigarrow a$)

→ Footprint, snapshots, etc.

→ Hierarchy of *classes of dynamic networks* (basic examples)

$$1 - * \\ \mathcal{E}^{1\forall}$$

$$1 \overset{st}{\rightsquigarrow} * \\ \mathcal{J}^{1\forall >}$$

$$1 \rightsquigarrow * \\ \mathcal{J}^{1\forall}$$

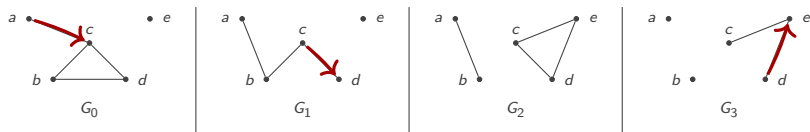
$$\mathcal{K} \\ * - *$$

$$\mathcal{TC} > \\ * \overset{st}{\rightsquigarrow} *$$

$$\mathcal{TC} \\ * \rightsquigarrow *$$

$$\mathcal{J}^{\forall 1} \\ * \rightsquigarrow 1$$

Basic graph concepts



→ Paths become temporal (*journey*)

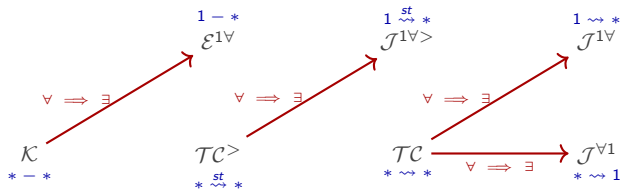
Ex: $((ac, t_1), (cd, t_2), (de, t_3))$ with $t_{i+1} \geq t_i$ and $\rho(e_i, t_i) = 1$

→ *Strict* journeys vs. *non-strict* journeys. (Relevant in discrete time only.)

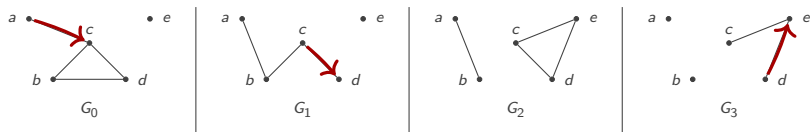
→ *Temporal connectivity*. Not symmetrical! (e.g. $a \rightsquigarrow e$, but $e \not\rightsquigarrow a$)

→ Footprint, snapshots, etc.

→ Hierarchy of *classes of dynamic networks* (basic examples)



Basic graph concepts



→ Paths become temporal (*journey*)

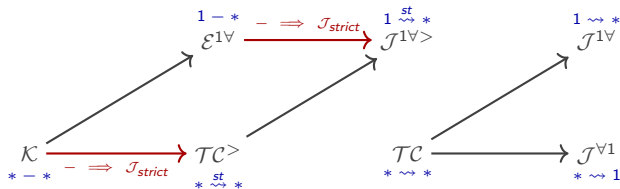
Ex: $((ac, t_1), (cd, t_2), (de, t_3))$ with $t_{i+1} \geq t_i$ and $\rho(e_i, t_i) = 1$

→ *Strict* journeys vs. *non-strict* journeys. (Relevant in discrete time only.)

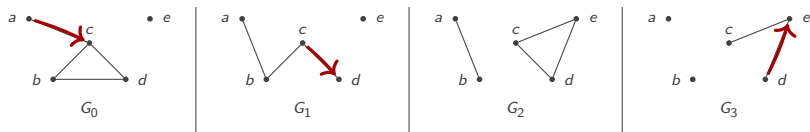
→ *Temporal connectivity*. Not symmetrical! (e.g. $a \rightsquigarrow e$, but $e \not\rightsquigarrow a$)

→ Footprint, snapshots, etc.

→ Hierarchy of *classes of dynamic networks* (basic examples)



Basic graph concepts



→ Paths become temporal (*journey*)

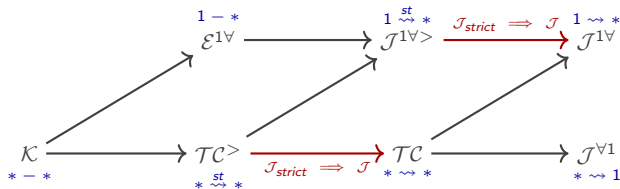
Ex: $((ac, t_1), (cd, t_2), (de, t_3))$ with $t_{i+1} \geq t_i$ and $\rho(e_i, t_i) = 1$

→ *Strict* journeys vs. *non-strict* journeys. (Relevant in discrete time only.)

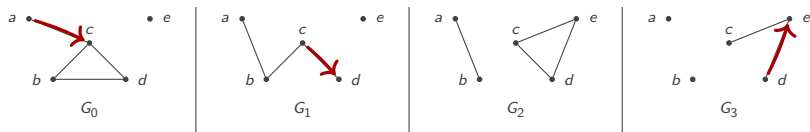
→ *Temporal connectivity*. Not symmetrical! (e.g. $a \rightsquigarrow e$, but $e \not\rightsquigarrow a$)

→ Footprint, snapshots, etc.

→ Hierarchy of *classes of dynamic networks* (basic examples)



Basic graph concepts



→ Paths become temporal (*journey*)

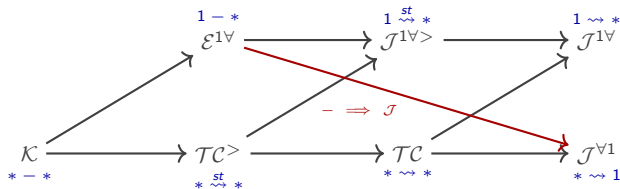
Ex: $((ac, t_1), (cd, t_2), (de, t_3))$ with $t_{i+1} \geq t_i$ and $\rho(e_i, t_i) = 1$

→ *Strict* journeys vs. *non-strict* journeys. (Relevant in discrete time only.)

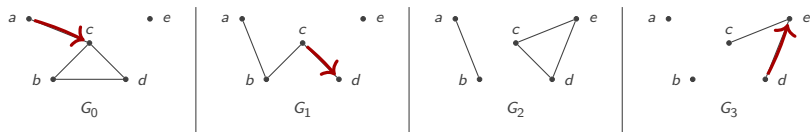
→ *Temporal connectivity*. Not symmetrical! (e.g. $a \rightsquigarrow e$, but $e \not\rightsquigarrow a$)

→ Footprint, snapshots, etc.

→ Hierarchy of *classes of dynamic networks* (basic examples)



Basic graph concepts



→ Paths become temporal (*journey*)

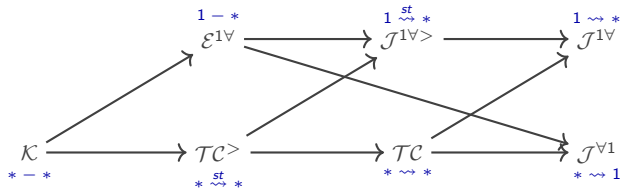
Ex: $((ac, t_1), (cd, t_2), (de, t_3))$ with $t_{i+1} \geq t_i$ and $\rho(e_i, t_i) = 1$

→ *Strict* journeys vs. *non-strict* journeys. (Relevant in discrete time only.)

→ *Temporal connectivity*. Not symmetrical! (e.g. $a \rightsquigarrow e$, but $e \not\rightsquigarrow a$)

→ Footprint, snapshots, etc.

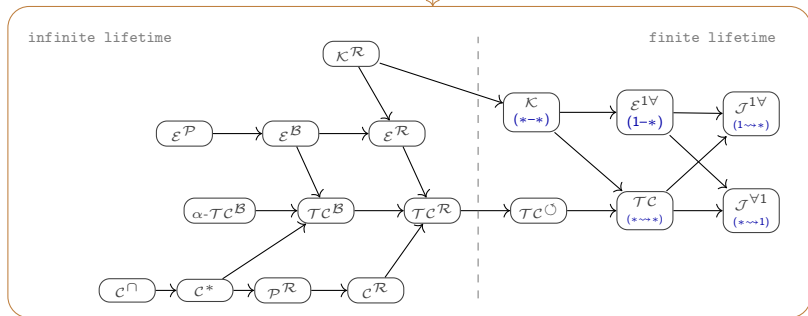
→ Hierarchy of *classes of dynamic networks* (basic examples)



Classes of dynamic networks

Network algorithms

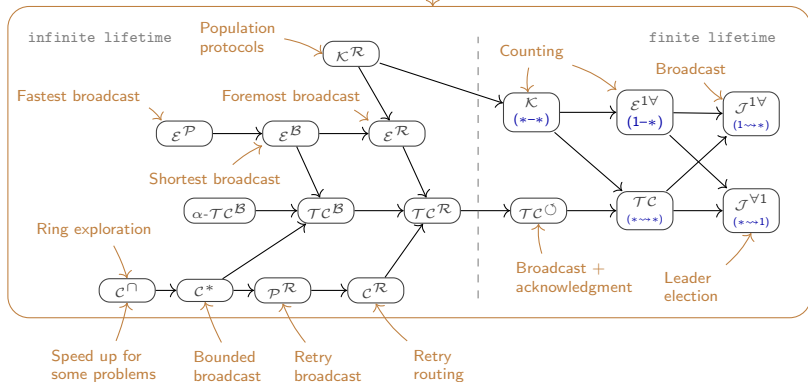
Exploit



Classes of dynamic networks

Network algorithms

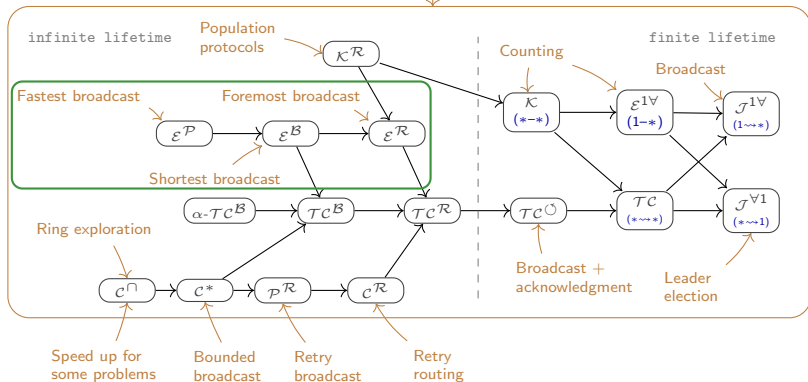
Exploit



Classes of dynamic networks

Network algorithms

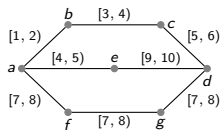
Exploit



Zoom: Optimal broadcast in DTNs?

What optimality ?

(Bui-Xuan, Jarry, Ferreira, 2003)

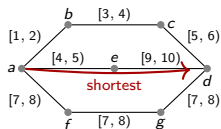


Which way is optimal from a to d ?

Zoom: Optimal broadcast in DTNs?

What optimality ?

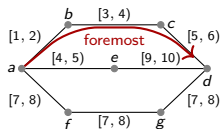
(Bui-Xuan, Jarry, Ferreira, 2003)



Which way is optimal from a to d ?
-min hop? (shortest)

Zoom: Optimal broadcast in DTNs?

What optimality ?



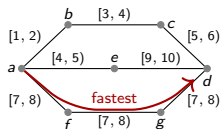
(Bui-Xuan, Jarry, Ferreira, 2003)

Which way is optimal from a to d ?

- min hop? (shortest)
- earliest arrival? (foremost)

Zoom: Optimal broadcast in DTNs?

What optimality ?



(Bui-Xuan, Jarry, Ferreira, 2003)

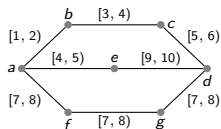
Which way is optimal from a to d ?

- min hop? (shortest)
- earliest arrival? (foremost)
- fastest traversal? (fastest)

Zoom: Optimal broadcast in DTNs?

What optimality ?

(Bui-Xuan, Jarry, Ferreira, 2003)



Which way is optimal from a to d ?

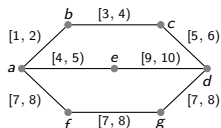
- min hop? (shortest)
- earliest arrival? (foremost)
- fastest traversal? (fastest)

→ Computing shortest, foremost and fastest journeys in dynamic networks

Zoom: Optimal broadcast in DTNs?

What optimality ?

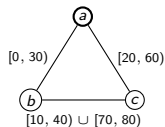
(Bui-Xuan, Jarry, Ferreira, 2003)



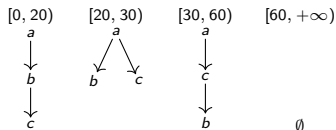
Which way is optimal from a to d ?

- min hop? (shortest)
- earliest arrival? (foremost)
- fastest traversal? (fastest)

Difficulty: Depends on the starting date! (E.g. for in the *foremost* case)



(a) The network.

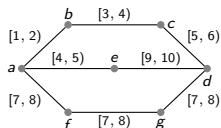


(b) Foremost broadcast trees from a depending on the starting date.

Zoom: Optimal broadcast in DTNs?

What optimality ?

(Bui-Xuan, Jarry, Ferreira, 2003)



Which way is optimal from a to d ?

- min hop? (shortest)
- earliest arrival? (foremost)
- fastest traversal? (fastest)

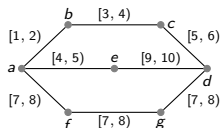
What about the distributed version?

→ Can we broadcast a message to all the nodes in a foremost, shortest, or fastest way? (with termination detection at the emitter and without knowing the schedule)

Zoom: Optimal broadcast in DTNs?

What optimality ?

(Bui-Xuan, Jarry, Ferreira, 2003)

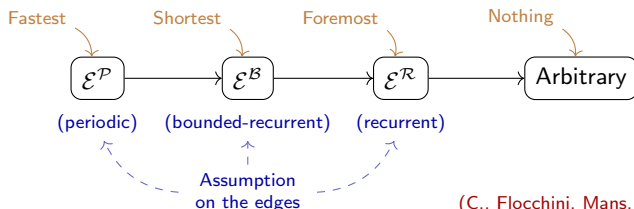


Which way is optimal from a to d ?

- min hop? (shortest)
- earliest arrival? (foremost)
- fastest traversal? (fastest)

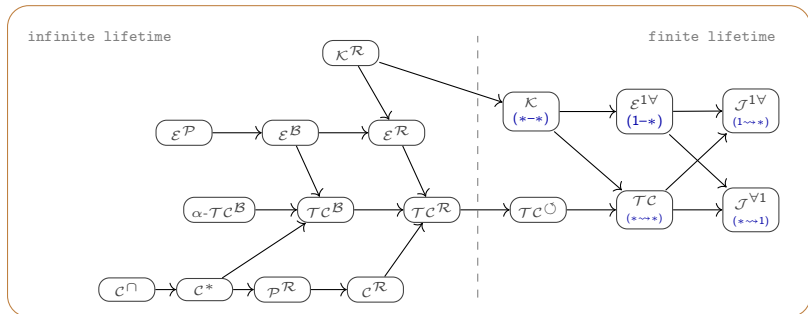
What about the distributed version?

→ Can we broadcast a message to all the nodes in a foremost, shortest, or fastest way? (with termination detection at the emitter and without knowing the schedule)

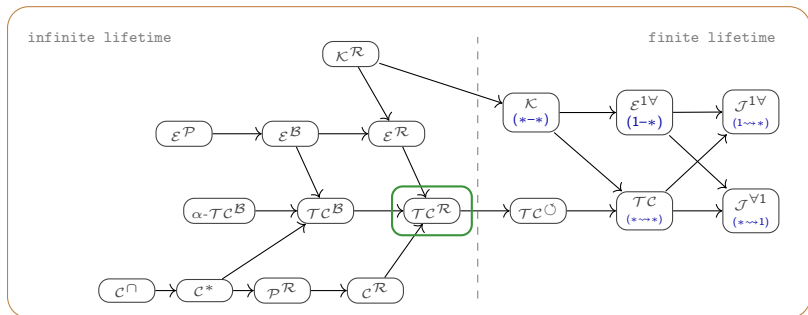


(C., Flocchini, Mans, Santoro, 2015)

Classes of dynamic networks



Classes of dynamic networks



Zoom: Exploiting structure within $\mathcal{TC}^{\mathcal{R}}$

$\mathcal{TC}^{\mathcal{R}} :=$ All nodes can reach each other through journeys infinitely often
(Formally, $\mathcal{TC}^{\mathcal{R}} := \forall t, \mathcal{G}_{[t, +\infty)} \in \mathcal{TC}$)

Zoom: Exploiting structure within $\mathcal{TC}^{\mathcal{R}}$

$\mathcal{TC}^{\mathcal{R}}$:= All nodes can reach each other through journeys infinitely often

(Formally, $\mathcal{TC}^{\mathcal{R}} := \forall t, \mathcal{G}_{[t, +\infty)} \in \mathcal{TC}$)

\equiv A connected spanning subset of the edges must be recurrent

(Braud Santoni et al., 2016)

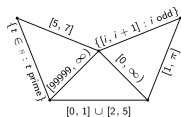
Zoom: Exploiting structure within $\mathcal{TC}^{\mathcal{R}}$

$\mathcal{TC}^{\mathcal{R}}$:= All nodes can reach each other through journeys infinitely often
(Formally, $\mathcal{TC}^{\mathcal{R}} := \forall t, \mathcal{G}_{[t, +\infty)} \in \mathcal{TC}$)

≡ A connected spanning subset of the edges must be recurrent

(Braud Santoni et al., 2016)

For example,



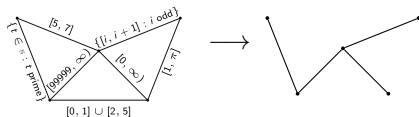
Zoom: Exploiting structure within $\mathcal{TC}^{\mathcal{R}}$

$\mathcal{TC}^{\mathcal{R}} :=$ All nodes can reach each other through journeys infinitely often
(Formally, $\mathcal{TC}^{\mathcal{R}} := \forall t, \mathcal{G}_{[t, +\infty)} \in \mathcal{TC}$)

\equiv A connected spanning subset of the edges must be recurrent

(Braud Santoni et al., 2016)

For example,



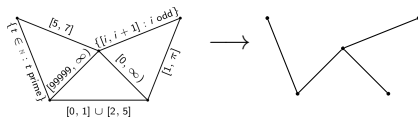
Zoom: Exploiting structure within $\mathcal{TC}^{\mathcal{R}}$

$\mathcal{TC}^{\mathcal{R}} :=$ All nodes can reach each other through journeys infinitely often
(Formally, $\mathcal{TC}^{\mathcal{R}} := \forall t, \mathcal{G}_{[t, +\infty)} \in \mathcal{TC}$)

\equiv A connected spanning subset of the edges must be recurrent

(Braud Santoni et al., 2016)

For example,



Can we exploit this property even if we don't know which subset of edges is recurrent?

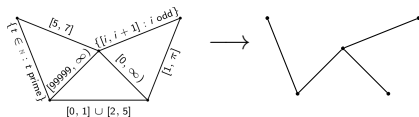
Zoom: Exploiting structure within $\mathcal{TC}^{\mathcal{R}}$

$\mathcal{TC}^{\mathcal{R}} :=$ All nodes can reach each other through journeys infinitely often
(Formally, $\mathcal{TC}^{\mathcal{R}} := \forall t, \mathcal{G}_{[t, +\infty)} \in \mathcal{TC}$)

\equiv A connected spanning subset of the edges must be recurrent

(Braud Santoni et al., 2016)

For example,



Can we exploit this property even if we don't know which subset of edges is recurrent?

→ New form of heredity in graphs: a solution must hold in *all connected spanning subgraph*

(C., Dubois, Petit, Robson, 2018)

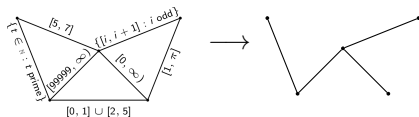
Zoom: Exploiting structure within $\mathcal{TC}^{\mathcal{R}}$

$\mathcal{TC}^{\mathcal{R}} :=$ All nodes can reach each other through journeys infinitely often
 (Formally, $\mathcal{TC}^{\mathcal{R}} := \forall t, \mathcal{G}_{[t, +\infty)} \in \mathcal{TC}$)

\equiv A connected spanning subset of the edges must be recurrent

(Braud Santoni et al., 2016)

For example,

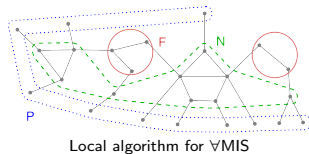
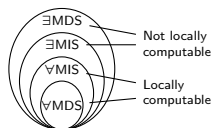


Can we exploit this property even if we don't know which subset of edges is recurrent?

\rightarrow New form of heredity in graphs: a solution must hold in *all connected spanning subgraph*

(C., Dubois, Petit, Robson, 2018)

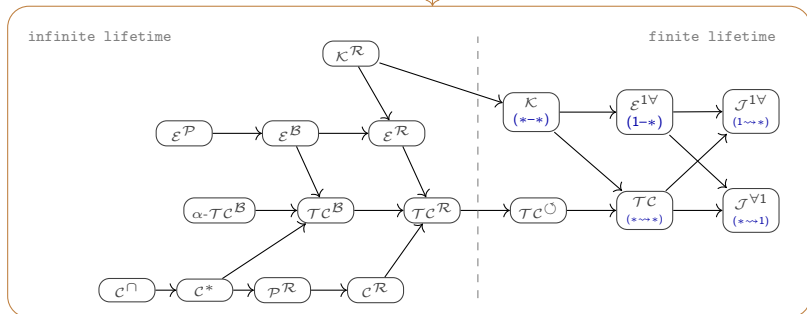
Ex: MINIMALDOMINATINGSET and MAXIMALINDEPENDENTSET



Classes of dynamic networks

Network algorithms

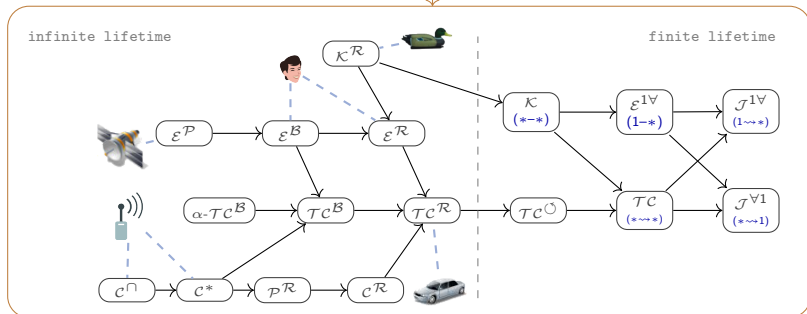
Exploit



Classes of dynamic networks

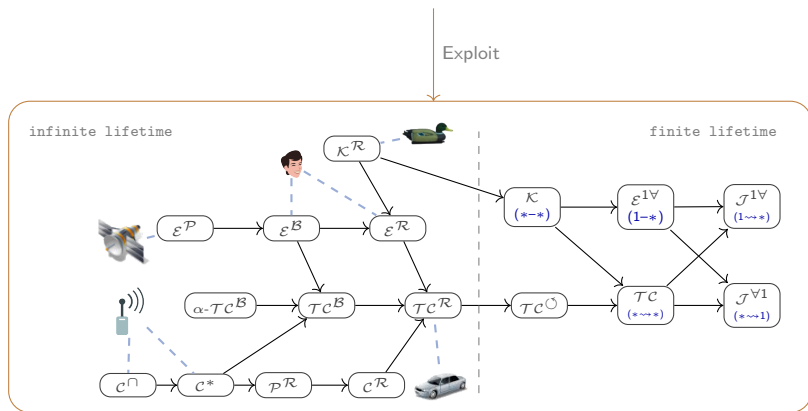
Network algorithms

Exploit



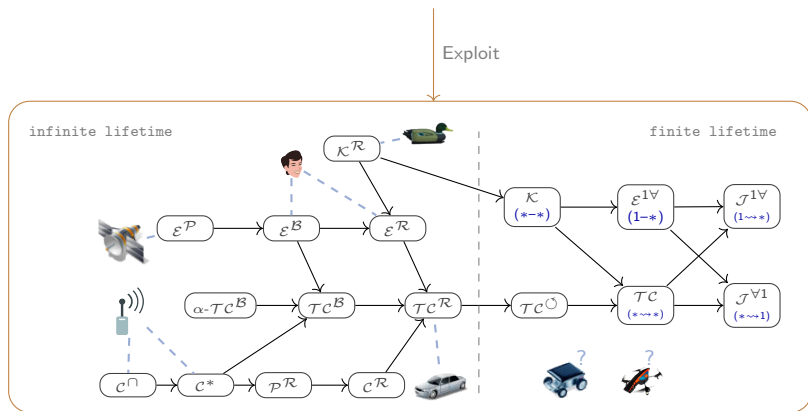
Classes of dynamic networks

Network algorithms



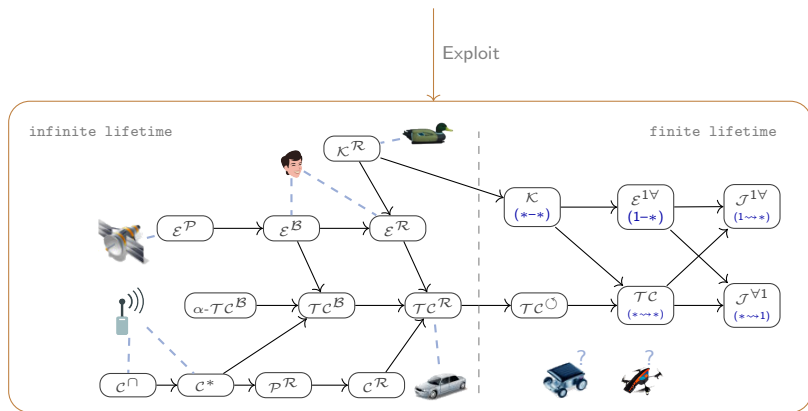
Classes of dynamic networks

Network algorithms



Classes of dynamic networks

Network algorithms

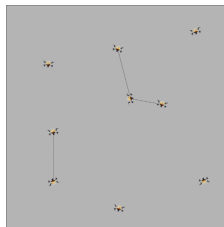


Centralized algorithms

Movement synthesis

Algorithmic movement synthesis

1) Collective movements which induce temporal structure



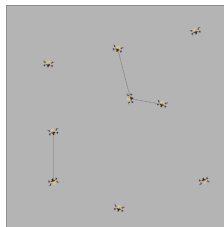
→ Synthesizing collective movements (*a.k.a. mobility models*) that satisfy temporal properties on the resulting communication graph (combined with a target mission).

← Ex: this network $\in \mathcal{ER}$

(Credit video: Jason Schoeters / JBotSim)

Algorithmic movement synthesis

1) Collective movements which induce temporal structure



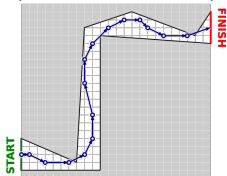
→ Synthesizing collective movements (a.k.a. *mobility models*) that satisfy temporal properties on the resulting communication graph (combined with a target mission).

← Ex: this network $\in \mathcal{ER}$

(Credit video: Jason Schoeters / JBotSim)

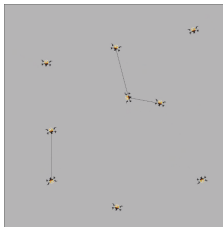
2) Integrating physical constraints in a tractable way

Discrete acceleration models
VECTOR RACER
(paper and pencil game)



Algorithmic movement synthesis

1) Collective movements which induce temporal structure



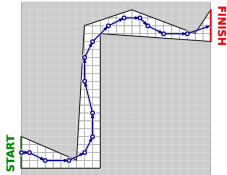
→ Synthesizing collective movements (*a.k.a. mobility models*) that satisfy temporal properties on the resulting communication graph (combined with a target mission).

← Ex: this network $\in \mathcal{E}^{\mathcal{R}}$

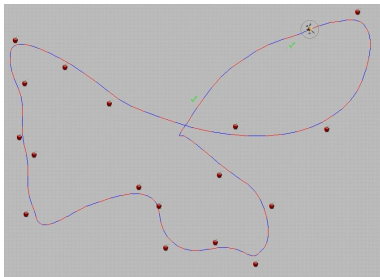
(Credit video: Jason Schoeters / JBotSim)

2) Integrating physical constraints in a tractable way

Discrete acceleration models
VECTOR RACER
(paper and pencil game)



→ Impact on problems, e.g. TSP ↗



Acceleration does impact the visit order!

The content of this talk can be found in

<https://arxiv.org/abs/1807.07801>