A Journey through Dynamic Networks (with Excursions)

Arnaud Casteigts

June 4, 2018

Habilitation à diriger des recherches

Rapporteur(e)s	Antonio Fernández Anta	DR. IMDEA-Networks	(Madrid)
	Clémence Magnien	DR. CNRS-LIP6	(Paris)
	Sébastien Tixeuil	PU. Sorbonne Université-LIP6	(Paris)
Examinateurs	Emmanuel Godard	PU. Univ Aix-Marseille-LIS	(Marseille)
	Nicolas Hanusse	DR. CNRS-LaBRI	(Bordeaux)
	Philippe Jacquet	DR. Bell-Labs Nokia	(Paris-Saclay)
Invité	Joseph Peters	PU. Simon Fraser University	(Vancouver)

A Journey through Dynamic Networks (with Excursions)

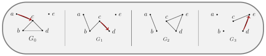
Part I: Finding Structure in Dynamic Networks

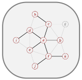
(Chap 1)
Conceptual shift

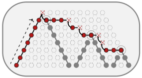
(Chap 2)
Distributed algorithms

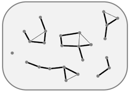
(Chap 3)
Around classes of dynamic networks

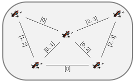
(Chap 4)
Beyond Structure

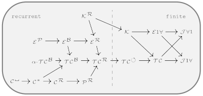


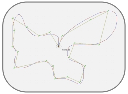


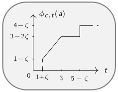












2007 — PhD — 2008 — 1st Postdoc U. Ottawa (Nayak & Stojmenovic) — 2009 — 2nd Postdoc U. Ottawa (Flocchini & Santoro) — 2012 — MCF Bordeaux — 2018 →

A Journey through Dynamic Networks (with Excursions)

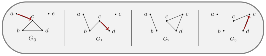
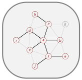
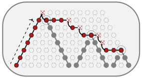
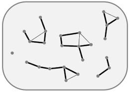
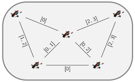
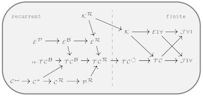
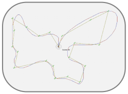
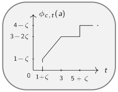
Part I: Finding Structure in Dynamic Networks

(Chap 1)
Conceptual shift

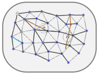
(Chap 2)
Distributed algorithms

(Chap 3)
Around classes of dynamic networks

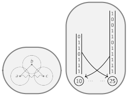
(Chap 4)
Beyond Structure

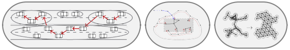
Part II: Excursions



JBotSim
(Chap 5)



AlgoDist (static)
(Chap 6)



Wireless networks
(Chap 7)

SHS & Privacy
(Chap 8)

2007 | 2008 | 1^{st} Postdoc U. Ottawa (Nayak & Stojmenovic) | 2009 | 2^{nd} Postdoc U. Ottawa (Flocchini & Santoro) | 2012 | MCF Bordeaux | 2018 →

PhD

A Journey through Dynamic Networks (with Excursions)

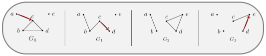
Part I: Finding Structure in Dynamic Networks

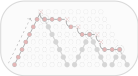
(Chap 1)
Conceptual shift

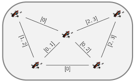
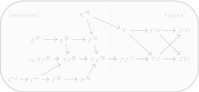
(Chap 2)
Distributed algorithms

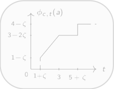
(Chap 3)
Around classes of dynamic networks

(Chap 4)
Beyond Structure

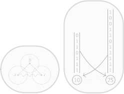
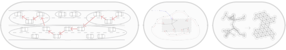






Part II: Excursions


JBotSim
(Chap 5)

AlgoDist (static)
(Chap 6)

Wireless networks
(Chap 7)

SHS & Privacy
(Chap 8)

2007 | PhD — 2008 | — 1^{st} Postdoc U. Ottawa (Nayak & Stojmenovic) — 2009 | — 2^{nd} Postdoc U. Ottawa (Flocchini & Santoro) — 2012 | — MCF Bordeaux — 2018 | →

A Journey through Dynamic Networks (with Excursions)

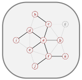
Part I: Finding Structure in Dynamic Networks

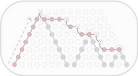
(Chap 1)
Conceptual shift

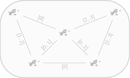
(Chap 2)
Distributed algorithms

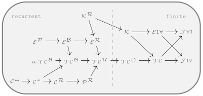
(Chap 3)
Around classes of dynamic networks

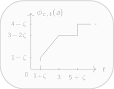
(Chap 4)
Beyond Structure



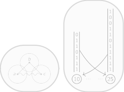


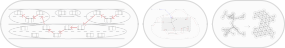






Part II: Excursions





JBotSim
(Chap 5)

AlgoDist (static)
(Chap 6)

Wireless networks
(Chap 7)

SHS & Privacy
(Chap 8)

2007 — PhD — 2008 — 1^{st} Postdoc U. Ottawa (Nayak & Stojmenovic) — 2009 — 2^{nd} Postdoc U. Ottawa (Flocchini & Santoro) — 2012 — MCF Bordeaux — 2018 →

A Journey through Dynamic Networks (with Excursions)

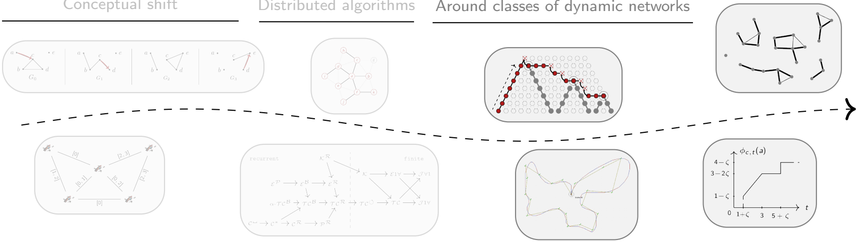
Part I: Finding Structure in Dynamic Networks

(Chap 1) Conceptual shift

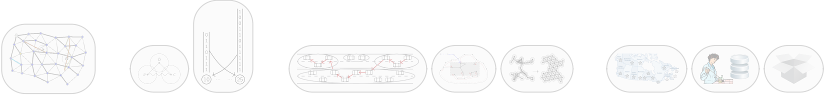
(Chap 2) Distributed algorithms

(Chap 3) Around classes of dynamic networks

(Chap 4) Beyond Structure



Part II: Excursions



JBotSim (Chap 5)

AlgoDist (static) (Chap 6)

Wireless networks (Chap 7)

SHS & Privacy (Chap 8)

2007 | PhD — 2008 | — 1^{st} Postdoc U. Ottawa (Nayak & Stojmenovic) — 2009 | — 2^{nd} Postdoc U. Ottawa (Flocchini & Santoro) — 2012 | — MCF Bordeaux — 2018 | →

A Journey through Dynamic Networks (with Excursions)

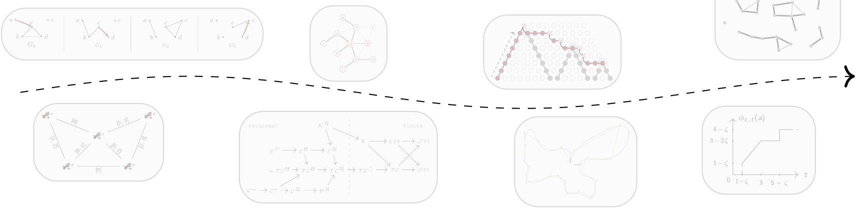
Part I: Finding Structure in Dynamic Networks

(Chap 1) Conceptual shift

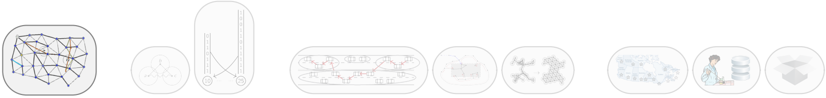
(Chap 2) Distributed algorithms

(Chap 3) Around classes of dynamic networks

(Chap 4) Beyond Structure



Part II: Excursions



JBotSim (Chap 5)

AlgoDist (static) (Chap 6)

Wireless networks (Chap 7)

SHS & Privacy (Chap 8)

2007 — PhD — 2008 — 1^{st} Postdoc U. Ottawa (Nayak & Stojmenovic) — 2009 — 2^{nd} Postdoc U. Ottawa (Flocchini & Santoro) — 2012 — MCF Bordeaux — 2018 →

A Journey through Dynamic Networks (with Excursions)

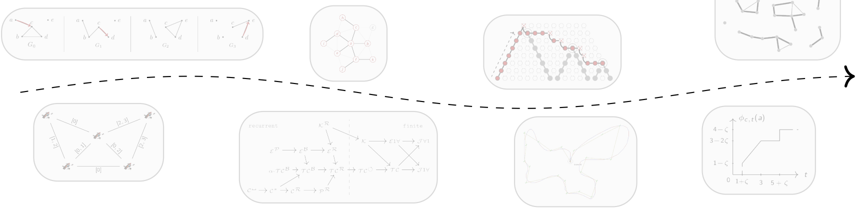
Part I: Finding Structure in Dynamic Networks

(CHAP 1) Conceptual shift

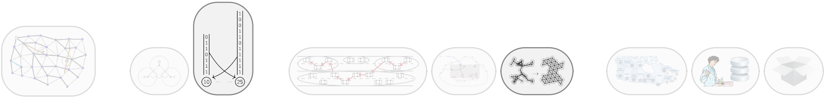
(CHAP 2) Distributed algorithms

(CHAP 3) Around classes of dynamic networks

(CHAP 4) Beyond Structure



Part II: Excursions



JBotSim (CHAP 5)

AlgoDist (static) (CHAP 6)

Wireless networks (CHAP 7)

SHS & Privacy (CHAP 8)

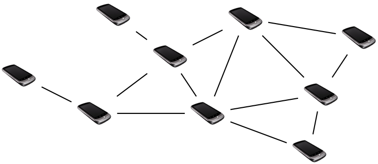
2007 | PhD — 2008 | — 1^{st} Postdoc U. Ottawa (Nayak & Stojmenovic) — 2009 | — 2^{nd} Postdoc U. Ottawa (Flocchini & Santoro) — 2012 | — MCF Bordeaux — 2018 | →

Introduction

Networks?

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

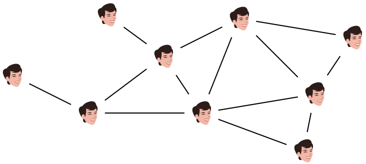
→ A network (or graph) $G = (V, E)$



Networks?

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

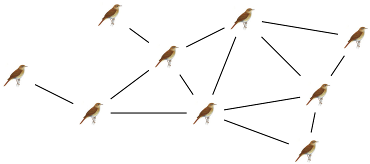
$\rightarrow$ A network (or graph) $G = (V, E)$



Networks?

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

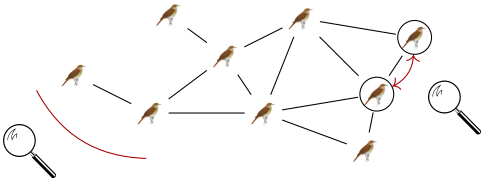
→ A network (or graph) $G = (V, E)$



Networks?

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

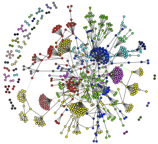
$\rightarrow$ A network (or graph) $G = (V, E)$



Complex networks

$\rightarrow$ compute global metrics

$\rightarrow$ explain and reproduce phenomena



Communication networks

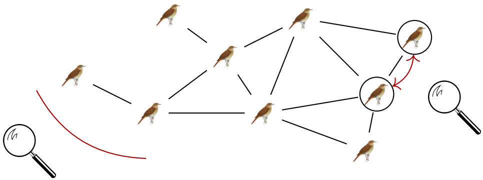
$\rightarrow$ design interactions among entities

$\rightarrow$ 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)

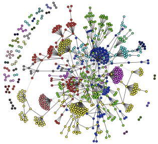
$\rightarrow$ A network (or graph) $G = (V, E)$



Complex networks

$\rightarrow$ compute global metrics

$\rightarrow$ explain and reproduce phenomena



Communication networks

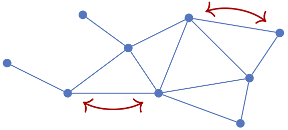
$\rightarrow$ design interactions among entities

$\rightarrow$ study what can be done *from within*

$\rightarrow$ distributed algorithms...

Distributed Algorithms

(Think globally, act locally)

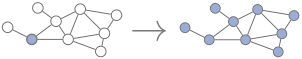


Collaboration of distinct entities to perform a common task.

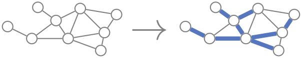
No centralization available.

Examples of problems:

Broadcast



Spanning tree



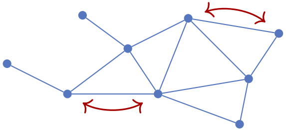
Election

Counting

Consensus, naming, routing, exploration, dominating sets, ...

Distributed Algorithms

(Think globally, act locally)

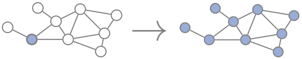


Collaboration of distinct entities to perform a common task.

No centralization available.

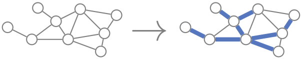
Examples of problems:

Broadcast



Election

Spanning tree



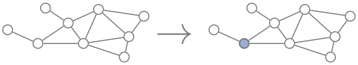
Counting

Consensus, naming, routing, exploration, dominating sets, ...

Leader Election (distributed problem)

→ Distinguishing one node among all

Version with ID → highest ID elected



	Time	# Messages	Message size	
Awerbuch'87	$O(n)$	$\mathbf{\Theta(m + n \log n)}$	$O(\log n)$ bits	
Peleg'90	$\mathbf{\Theta(D)}$	$O(Dm)$	$O(\log n)$ bits	

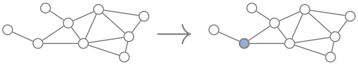
Optimal in time and number of messages?

Excursion: Bit complexity of Leader Election (CHAP 6 SEC 1)

Leader Election (distributed problem)

→ Distinguishing one node among all

Version with ID → highest ID elected

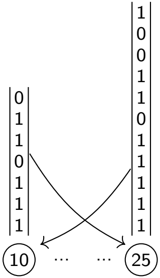


	Time	# Messages	Message size	# Bit rounds
Awerbuch'87	$O(n)$	$\mathbf{\Theta(m + n \log n)}$	$O(\log n)$ bits	$O(n \log n)$
Peleg'90	$\mathbf{\Theta(D)}$	$O(Dm)$	$O(\log n)$ bits	$O(D \log n)$
Our contrib.	$O(D + \log n)$	$O((D + \log n)m)$	$O(1)$ bits	$\mathbf{\Theta(D + \log n)}$

Optimal in time and number of messages?

Yes! With constant size messages → $O(D + \log n)$ bit rounds algorithm

+ matching lower bound: $\Omega(D + \log n)$



The algorithm

Bitwise dissemination of highest ID defining a spanning tree

New encoding technique for IDs

→ ex: $ID = 25 \stackrel{2}{=} 11001$, then $\alpha(Id) = 11111011001$

Casteigts *et al.*, 30th Int. Symposium on Distributed Computing (DISC), 2016

Dynamic networks?

How to approach these contexts?

Applied *versus* theoretical?

Dynamic networks?

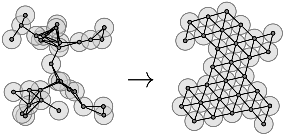
How to approach these contexts?

Applied *versus* theoretical?

Excursion: Biconnecting robots with virtual angular forces (Chap 7 Sec 3)

Problem: Deploying robots from arbitrary connected configuration, with consideration to

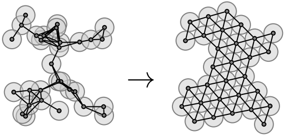
- Coverage (max)
- Movements (min)
- Diameter (min)
- Biconnectivity (fault tolerance)



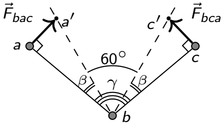
Excursion: Biconnecting robots with virtual angular forces (Chap 7 Sec 3)

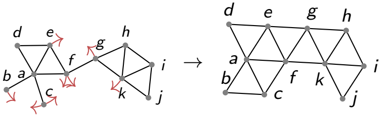
Problem: Deploying robots from arbitrary connected configuration, with consideration to

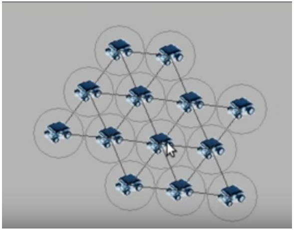
- Coverage (max)
- Movements (min)
- Diameter (min)
- Biconnectivity (fault tolerance)



Approach: spring forces (attraction/repulsion) + angular forces







Casteigts *et al.*, Computer Communication (Elsevier), Vol.35 Issue 9, 2012.

Dynamic networks?

How to approach these contexts?

Applied *versus* theoretical

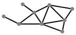
Dynamic networks?

How to approach these contexts?

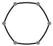
Applied *versus* theoretical

→ Working with structure (mostly theoretical)

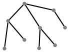
Static networks



Planar

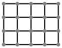


Ring



Tree

Complete graph

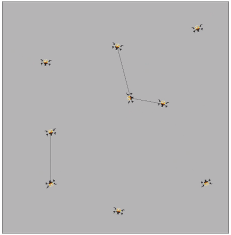


Grid

. . . can be exploited by an algorithm

(Highly) dynamic networks

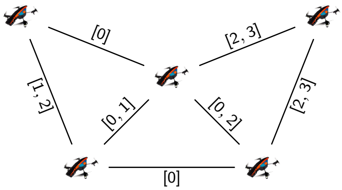
What kind of structure?



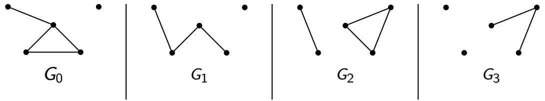
Time-varying graphs (TVG)

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

- $\mathcal{T} \subseteq \mathbb{N}/\mathbb{R}$ (lifetime)
- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ (presence fonction)
- $\zeta : E \times \mathcal{T} \rightarrow \mathbb{N}/\mathbb{R}$ (latency function)



Another classical view $\mathcal{G} = G_0, G_1, \ldots$



Variety of models and terminologies:

Dynamic graphs, evolving graphs, temporal graphs, link streams, etc.

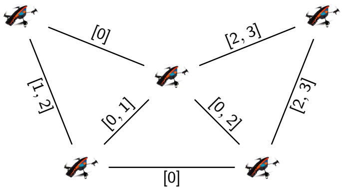
Casteigts *et al.*, Int. J. of Parallel, Emergent and Distributed Systems, Vol. 27, Issue 5, 2012

(among others)

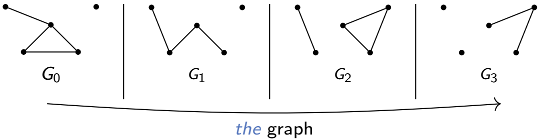
Time-varying graphs (TVG)

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

- $\mathcal{T} \subseteq \mathbb{N}/\mathbb{R}$ (lifetime)
- $\rho : E \times \mathcal{T} \to \{0, 1\}$ (presence fonction)
- $\zeta : E \times \mathcal{T} \to \mathbb{N}/\mathbb{R}$ (latency function)



Another classical view $\mathcal{G} = G_0, G_1, \ldots$



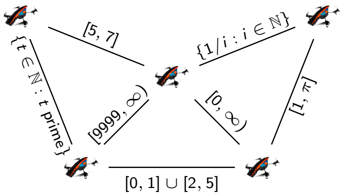
Variety of models and terminologies:

Dynamic graphs, evolving graphs, temporal graphs, link streams, etc.

Casteigts *et al.*, Int. J. of Parallel, Emergent and Distributed Systems, Vol. 27, Issue 5, 2012

(among others)

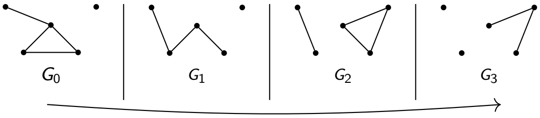
Time-varying graphs (TVG)



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

- $\mathcal{T} \subseteq \mathbb{N}/\mathbb{R}$ (lifetime)
- $\rho : E \times \mathcal{T} \to \{0, 1\}$ (presence fonction)
- $\zeta : E \times \mathcal{T} \to \mathbb{N}/\mathbb{R}$ (latency function)

Another classical view $\mathcal{G} = G_0, G_1, \ldots$



Variety of models and terminologies:

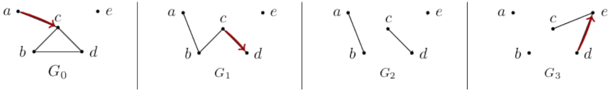
Dynamic graphs, evolving graphs, temporal graphs, link streams, etc.

Casteigts *et al.*, Int. J. of Parallel, Emergent and Distributed Systems, Vol. 27, Issue 5, 2012

(among others)

A conceptual shift (impact of temporal dimension)

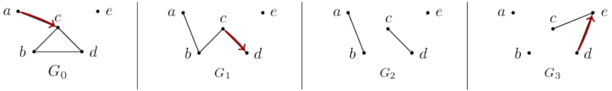
Journeys (temporal paths)



→ Temporal connectivity

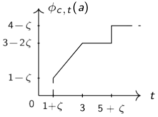
A conceptual shift (impact of temporal dimension)

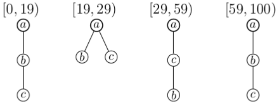
Journeys (temporal paths)

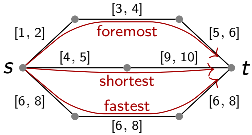


→ Temporal connectivity

Temporal distance & shortest paths

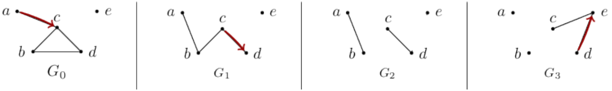






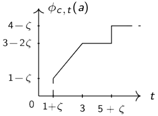
A conceptual shift (impact of temporal dimension)

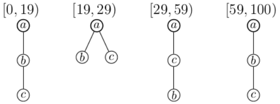
Journeys (temporal paths)

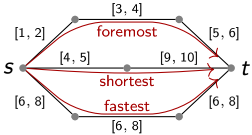


→ Temporal connectivity

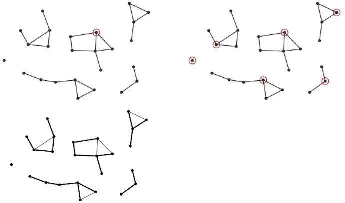
Temporal distance & shortest paths

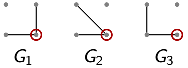






Redefinition of classical problems

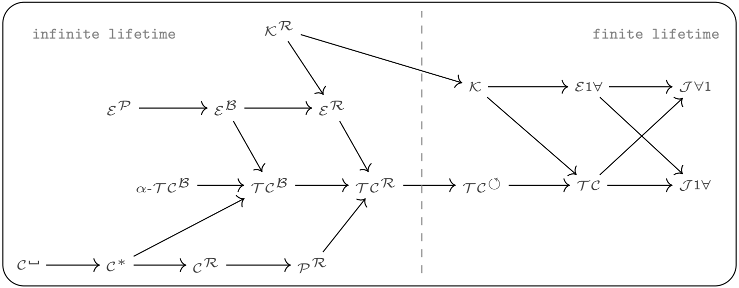




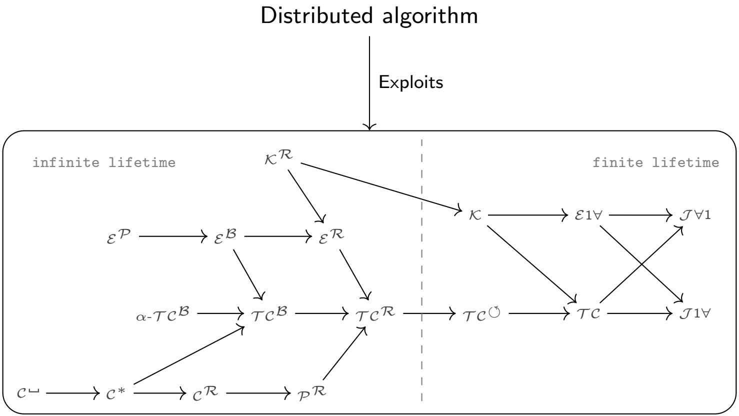
Casteigts *et al.*, Int. J. of Parallel, Emergent and Distributed Systems, Vol. 27, Issue 5, 2012

(among others)

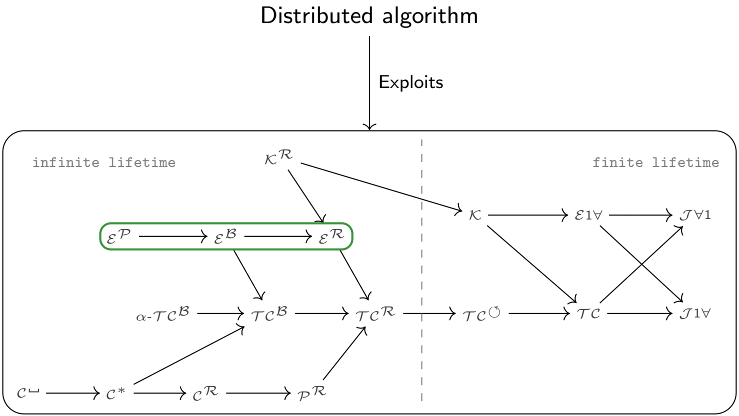
Classes of dynamic networks/graphs



Classes of dynamic networks/graphs



Classes of dynamic networks/graphs



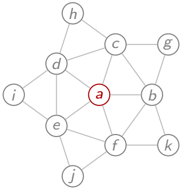
Zoom: Optimal broadcast with termination detection (Chap 2 Sec 2)

Distributed problem:

→ A node must send a piece of information to all other nodes, then detect termination

→ Three criteria: foremost, shortest, fastest

Unsolvable without further assumption.



Structure (+ knowledge)

- Connected footprint (obvious)
  - Recurrent edges $\mathcal{E}^{\mathcal{R}}$ (+ n)
  - Bounded-recurrent edges $\mathcal{E}^{\mathcal{B}}$ (+ Δ)
  - Periodic edges $\mathcal{E}^{\mathcal{P}}$ (+ p)

Note that $\mathcal{E}^{\mathcal{P}} \subset \mathcal{E}^{\mathcal{R}} \subset \mathcal{E}^{\mathcal{B}}$

Casteigts *et al.*, Int. J. of Foundations of Computer Science, Vol. 26, Issue 4, 2015

(optimality metrics defined by Bui-Xuan, Ferreira, Jarry, 2003)

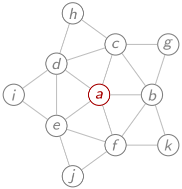
Zoom: Optimal broadcast with termination detection (Chap 2 Sec 2)

Distributed problem:

→ A node must send a piece of information to all other nodes, then detect termination

→ Three criteria: foremost, shortest, fastest

Unsolvable without further assumption.



Structure (+ knowledge)

- Connected footprint (obvious)
  - Recurrent edges $\mathcal{E}^{\mathcal{R}}$ $(+\ n)$
  - Bounded-recurrent edges $\mathcal{E}^{\mathcal{B}}$ $(+\ \Delta)$
  - Periodic edges $\mathcal{E}^{\mathcal{P}}$ $(+\ p)$

Note that $\mathcal{E}^{\mathcal{P}} \subset \mathcal{E}^{\mathcal{R}} \subset \mathcal{E}^{\mathcal{B}}$

Theorems:

- Foremost feasible in $\mathcal{E}^{\mathcal{R}}$
- Shortest feasible in $\mathcal{E}^{\mathcal{B}}$ (but not in $\mathcal{E}^{\mathcal{R}}$)
- Fastest feasible in $\mathcal{E}^{\mathcal{P}}$ (but not in $\mathcal{E}^{\mathcal{B}}$)

Theorem:

$power(\mathcal{E}^{\mathcal{R}} + n) \subsetneq power(\mathcal{E}^{\mathcal{B}} + \Delta) \subsetneq power(\mathcal{E}^{\mathcal{P}} + p)$

Casteigts *et al.*, Int. J. of Foundations of Computer Science, Vol. 26, Issue 4, 2015

(optimality metrics defined by Bui-Xuan, Ferreira, Jarry, 2003)

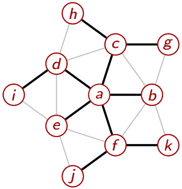
Zoom: Optimal broadcast with termination detection (Chap 2 Sec 2)

Distributed problem:

→ A node must send a piece of information to all other nodes, then detect termination

→ Three criteria: foremost, shortest, fastest

Unsolvable without further assumption.



Structure (+ knowledge)

- Connected footprint (obvious)
  - Recurrent edges $\mathcal{E}^{\mathcal{R}}$ $(+\ n)$
  - Bounded-recurrent edges $\mathcal{E}^{\mathcal{B}}$ $(+\ \Delta)$
  - Periodic edges $\mathcal{E}^{\mathcal{P}}$ $(+\ p)$

Note that $\mathcal{E}^{\mathcal{P}} \subset \mathcal{E}^{\mathcal{R}} \subset \mathcal{E}^{\mathcal{B}}$

Theorems:

- Foremost feasible in $\mathcal{E}^{\mathcal{R}}$
- Shortest feasible in $\mathcal{E}^{\mathcal{B}}$ (but not in $\mathcal{E}^{\mathcal{R}}$)
- Fastest feasible in $\mathcal{E}^{\mathcal{P}}$ (but not in $\mathcal{E}^{\mathcal{B}}$)

Theorem:

$power(\mathcal{E}^{\mathcal{R}} + n) \subsetneq power(\mathcal{E}^{\mathcal{B}} + \Delta) \subsetneq power(\mathcal{E}^{\mathcal{P}} + p)$

Casteigts *et al.*, Int. J. of Foundations of Computer Science, Vol. 26, Issue 4, 2015

(optimality metrics defined by Bui-Xuan, Ferreira, Jarry, 2003)

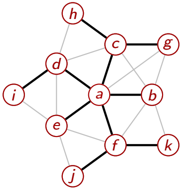
Zoom: Optimal broadcast with termination detection (Chap 2 Sec 2)

Distributed problem:

$\rightarrow$ A node must send a piece of information to all other nodes, then detect termination

$\rightarrow$ Three criteria: foremost, shortest, fastest

Unsolvable without further assumption.



Structure (+ knowledge)

- Connected footprint (obvious)
  - Recurrent edges $\mathcal{E}^{\mathcal{R}}$ $(+\ n)$
  - Bounded-recurrent edges $\mathcal{E}^{\mathcal{B}}$ $(+\ \Delta)$
  - Periodic edges $\mathcal{E}^{\mathcal{P}}$ $(+\ p)$

Note that $\mathcal{E}^{\mathcal{P}} \subset \mathcal{E}^{\mathcal{R}} \subset \mathcal{E}^{\mathcal{B}}$

Theorems:

- Foremost feasible in $\mathcal{E}^{\mathcal{R}}$
- Shortest feasible in $\mathcal{E}^{\mathcal{B}}$ (but not in $\mathcal{E}^{\mathcal{R}}$)
- Fastest feasible in $\mathcal{E}^{\mathcal{P}}$ (but not in $\mathcal{E}^{\mathcal{B}}$)

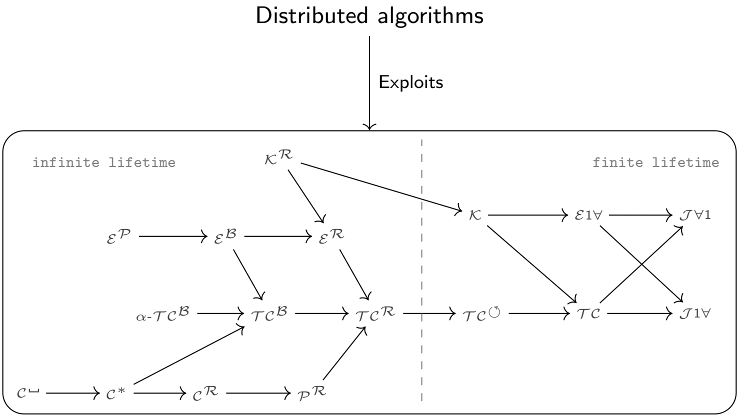
Theorem:

$power(\mathcal{E}^{\mathcal{R}} + n) \subsetneq power(\mathcal{E}^{\mathcal{B}} + \Delta) \subsetneq power(\mathcal{E}^{\mathcal{P}} + p)$

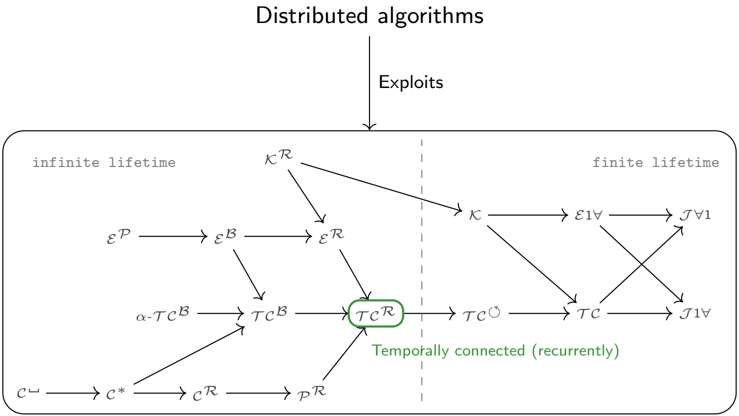
Casteigts *et al.*, Int. J. of Foundations of Computer Science, Vol. 26, Issue 4, 2015

(optimality metrics defined by Bui-Xuan, Ferreira, Jarry, 2003)

Classes of dynamic networks



Classes of dynamic networks



Zoom: Exploiting structure within $\mathcal{TC}^{\mathcal{R}}$ (Chap 2 Sec 4)

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

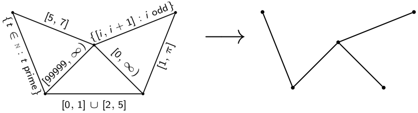
($\mathcal{TC}^{\mathcal{R}} := \forall t, \mathcal{G}_{[t,+\infty)} \in \mathcal{TC}$)

Zoom: Exploiting structure within $\mathcal{TC}^{\mathcal{R}}$ (Chap 2 Sec 4)

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

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

Alternative characterization: Eventual footprint connected — Braud Santoni et al., 2016



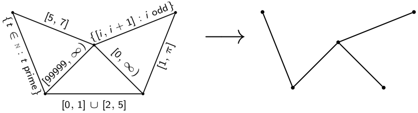
→ Can be exploited in a distributed algorithm — Kaaouachi *et al.*, 2016

Zoom: Exploiting structure within $\mathcal{TC}^{\mathcal{R}}$ (Chap 2 Sec 4)

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

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

Alternative characterization: Eventual footprint connected — Braud Santoni et al., 2016



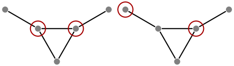
→ Can be exploited in a distributed algorithm — Kaaouachi *et al.*, 2016

Robustness

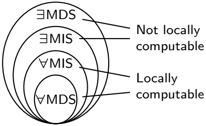
(dealing with uncertainty)

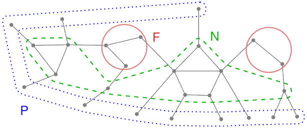
→ New form of heredity in graphs: property/solution holds in all *connected spanning subgraph*

Ex: MinimalDominatingSet (MDS) and MaximalIndependentSet (MIS)



Robust MDS — Non robust MDS

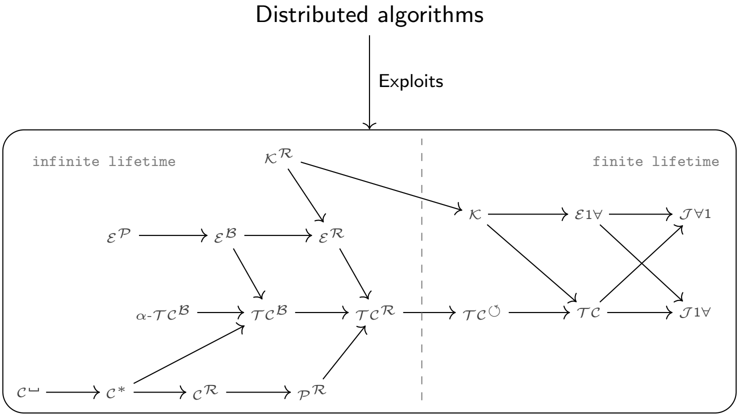




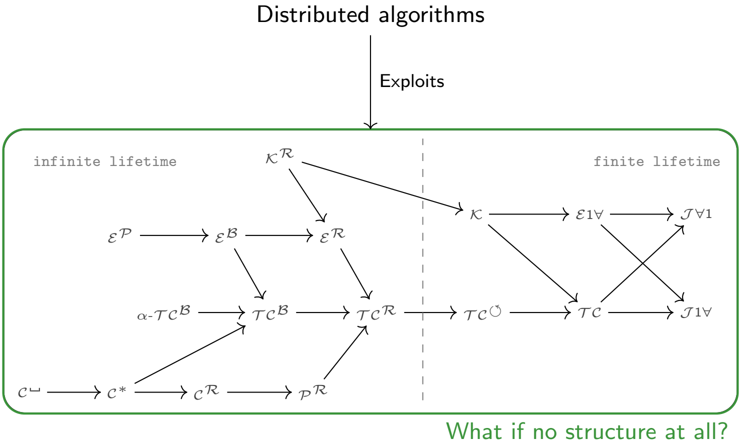
Local algorithm for ∀MIS

Casteigts, Dubois, Petit, Robson, CoRR, abs/1703.03190v2, 2018

Classes of dynamic networks

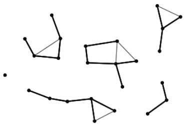


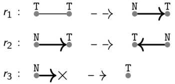
Classes of dynamic networks

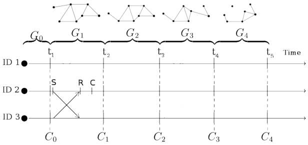


Beyond structure (Chap 4)

Maintaining a Spanning Forest

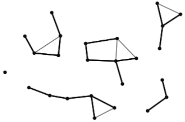


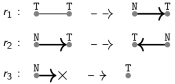


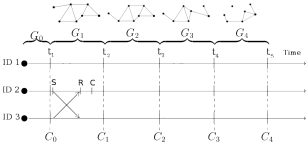


The Computer Journal (Oxford Univ. Press), to appear, 2018

Maintaining a Spanning Forest

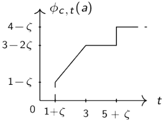


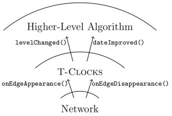


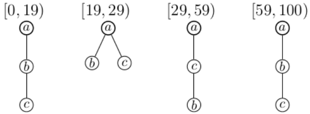


The Computer Journal (Oxford Univ. Press), to appear, 2018

Measuring Temporal Distances



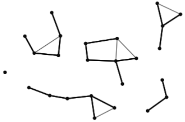


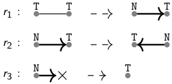


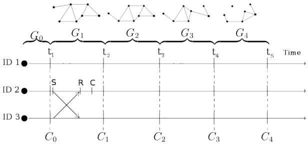
IEEE Transactions on Computers, Vol. 63, Issue 2, 2014

Beyond structure (Chap 4)

Maintaining a Spanning Forest

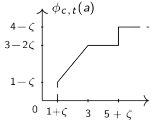


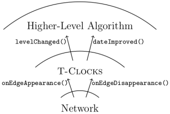


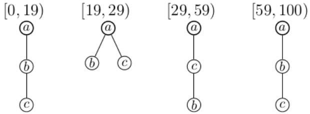


The Computer Journal (Oxford Univ. Press), to appear, 2018

Measuring Temporal Distances

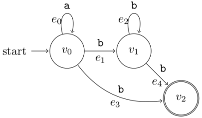






IEEE Transactions on Computers, Vol. 63, Issue 2, 2014

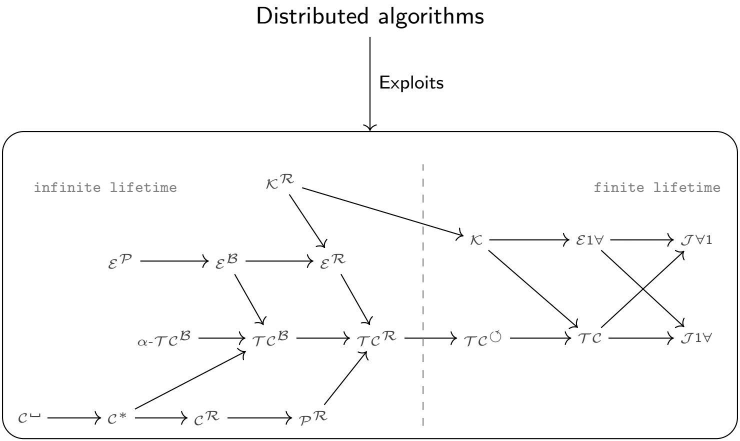
The Power of Waiting



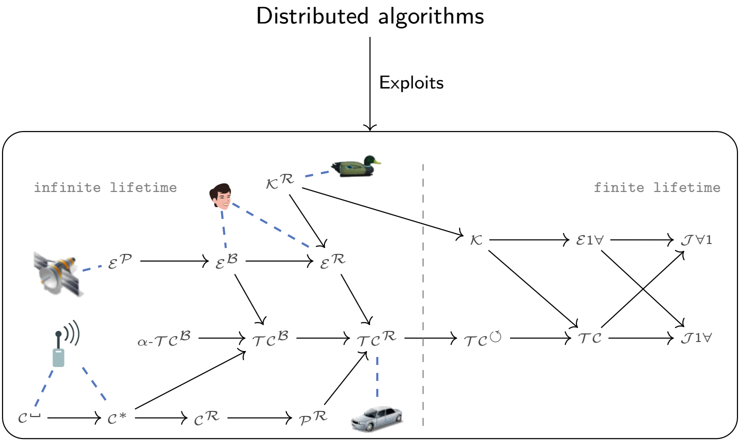
e	Presence $\rho(e,t)=1$ iff	Latency $\zeta(e,t)$ =
e_0	always true	$(p-1)t$
e_1	$t>p$	$(q-1)t$
e_2	$t \neq p^i q^{i-1}, i>1$	$(q-1)t$
e_3	$t=p$	any
e_4	$t=p^i q^{i-1}$, $i>1$	any

Theoretical Computer Science (Elsevier), Vol. 590, 27-37, 2015

Classes of dynamic networks

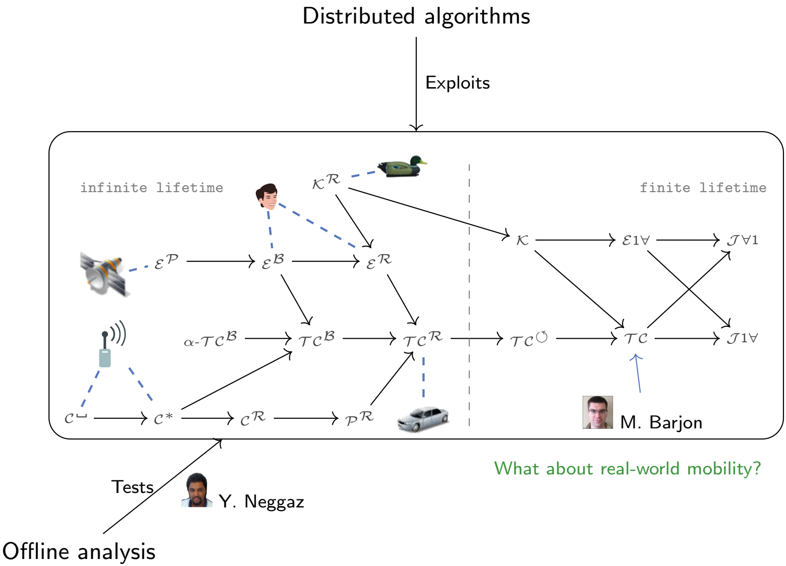


Classes of dynamic networks

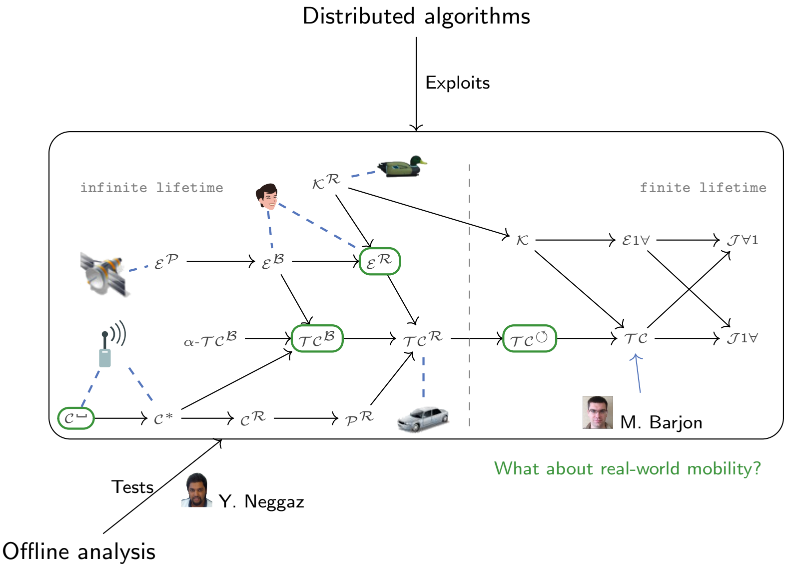


What about real-world mobility?

Classes of dynamic networks



Classes of dynamic networks

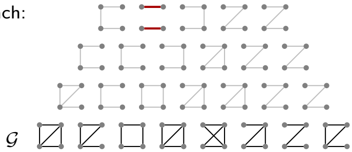


Ex: T-interval connectivity ($\mathcal{C}^{\sqcup}$)

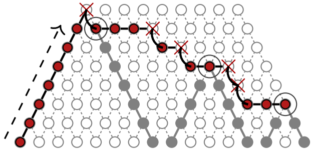
All consecutive T graphs contain a common spanning tree → measures stability

The problem, finding T (given a sequence of δ graphs)

Approach:



Intersections of graphs



→ Walking the hierarchy

→ Theorem: $O(\delta)$ high-level operations (*vs.* $O(\delta^2)$ *naive*)
(intersections and connectivity tests).

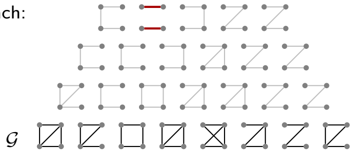
Casteigts *et al.*, 9th Int. Conference on Algorithms and Complexity (CIAC), 2015

Ex: T-interval connectivity ($\mathcal{C}^{\sqcup}$)

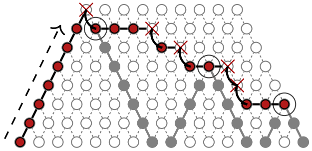
All consecutive T graphs contain a common spanning tree → measures stability

The problem, finding T (given a sequence of δ graphs)

Approach:



Intersections of graphs



→ Walking the hierarchy

→ **Theorem:** $O(\delta)$ high-level operations (*vs.* $O(\delta^2)$ *naive*)
(`intersections` and `connectivity tests`).

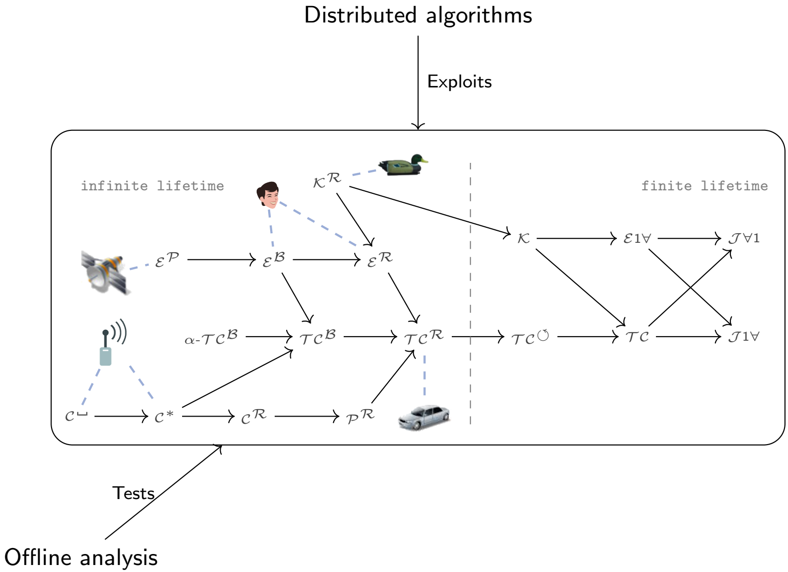
Genericity:

	Property	composition	test	goal
$\mathcal{C}^{\sqcup}$	T-interval connectivity	`intersection`	`connectivity`	max
$\mathcal{E}^{\mathcal{R}}$	Realization of the footprint	`union`	`identity`	min
$\mathcal{TC}^{\mathcal{B}}$	Temporal diameter	`concat TC`	`completeness`	min
$\mathcal{TC}^{\circlearrowleft}$	Round-trip temp. diameter	`concat RTTC`	`completeness`	min

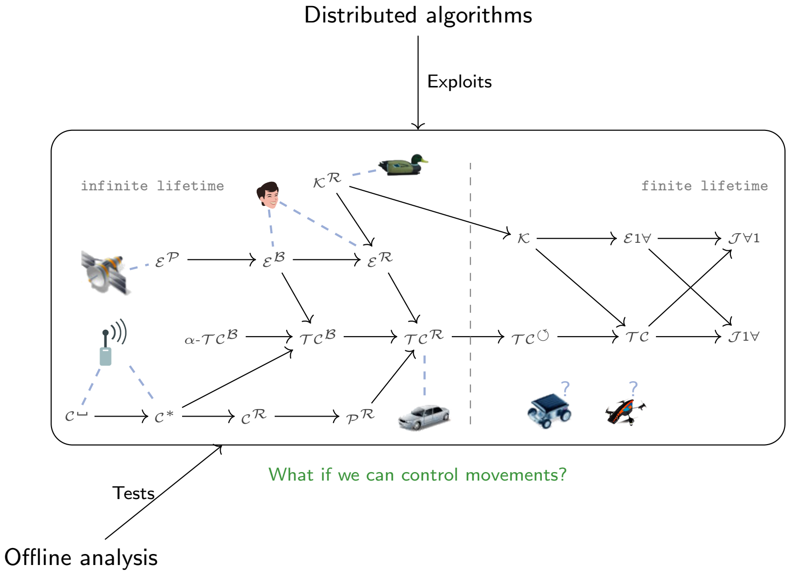
Casteigts *et al.*, 9th Int. Conference on Algorithms and Complexity (CIAC), 2015
Casteigts *et al.*, 19th Int. Conference on Structural Information and Communication Complexity (SIROCCO), 2017

Combined article in minor revision (in ToCS)

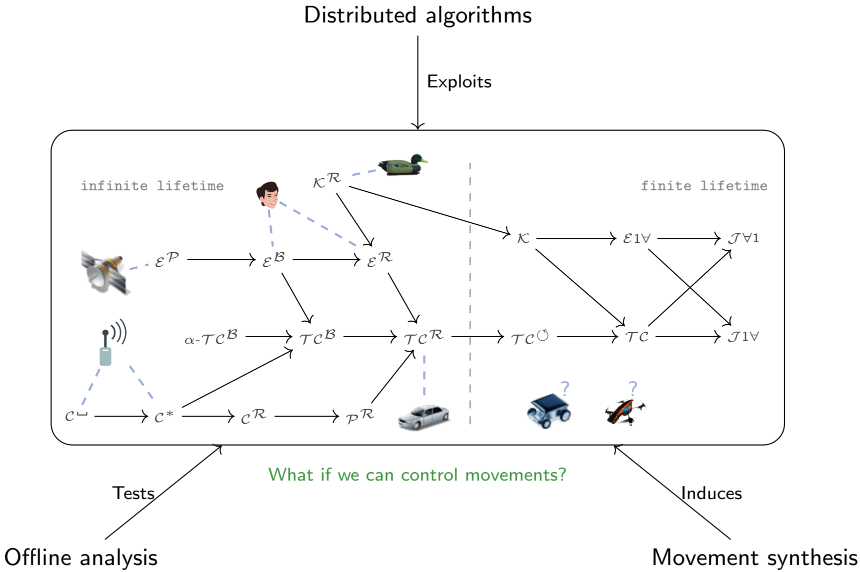
Classes of dynamic networks



Classes of dynamic networks



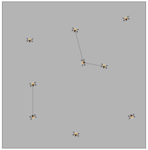
Classes of dynamic networks



Collective movements which induce temporal structure

Jason Schoeters (since Nov. 2017)
PhD funded by ANR ESTATE

Synthesizing collective movements (*a.k.a. mobility models*) so as to satisfies temporal properties on the resulting communication graph.

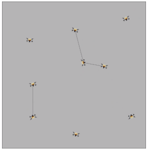


$\leftarrow$ this network $\in \mathcal{E}^{\mathcal{R}}$

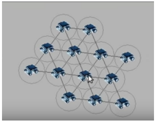
Collective movements which induce temporal structure

Jason Schoeters (since Nov. 2017)
PhD funded by ANR ESTATE

Synthesizing collective movements (*a.k.a.* *mobility models*) so as to satisfies temporal properties on the resulting communication graph.



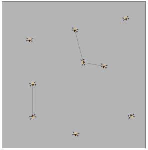
← this network $\in \mathcal{E}^{\mathcal{R}}$ this one $\in \mathcal{C}^*$ →



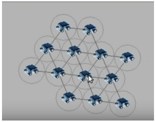
Collective movements which induce temporal structure

Jason Schoeters (since Nov. 2017)
PhD funded by ANR ESTATE

Synthesizing collective movements (*a.k.a.* *mobility models*) so as to satisfies temporal properties on the resulting communication graph.



$\leftarrow$ this network $\in \mathcal{E}^{\mathcal{R}}$ this one $\in \mathcal{C}^*$ $\rightarrow$



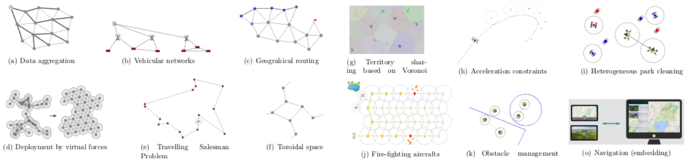
Main objectives:

- Design mobility models independently (warm up)
- Combine them with concrete problems like exploration (more difficult)

Interesting target: $\mathcal{TC}^{\mathcal{B}}$ (bounded temporal diameter)
$\rightarrow$ Weakest setting to detect a crash.

JBotSim

JBotSim Prototyping library for distributed algorithms in dynamic networks



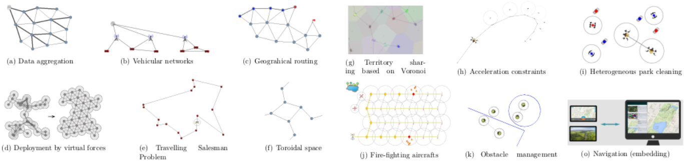
(a) Data aggregation (b) Vehicular networks (c) Geograhical routing (g) Territory sharing based on Voronoi (h) Acceleration constraints (i) Heterogeneous park cleaning (d) Deployment by virtual forces (e) Travelling Salesman Problem (f) Toroidal space (j) Fire-fighting aircrafts (k) Obstacle management (o) Navigation (embedding)

Interactive, simple to use, extensible, event-driven programming (java)

→ Download statistics (SF): 150 (2015), 900 (2016), 1100 (2017), . . .

Growing community, ∼ 10 "universities"

JBotSim Prototyping library for distributed algorithms in dynamic networks



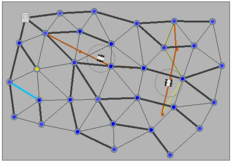
Interactive, simple to use, extensible, event-driven programming (java)

→ Download statistics (SF): 150 (2015), 900 (2016), 1100 (2017), . . .
Growing community, ∼ 10 "universities"

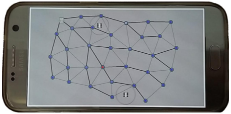
Enables the use of many models of computations (by design), at graph or network level.

Masters course "Algorithmique de la mobilité" (48h, Bordeaux) (others in Ottawa, Marseille, Strasbourg)

Ex: Student project (2017)

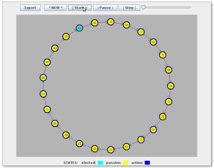


JBotsim on Android!



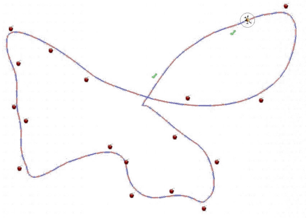
Kinda Al Chahid (M2)

Davis project (E.Godard)



Casteigts, 8^{th} Int. Conf. on Simulation Tools and Techniques (SimuTools), 2015

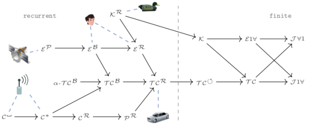
Perspectives



Perspectives

Two natural perspectives:

Structure in dynamic networks (a step further) (Perspective 0)



- → Explore relations among existing and new classes of dynamic networks
- → Focus on $\mathcal{TC}^{\mathcal{R}}$ and $\mathcal{TC}^{\mathcal{B}}$ and robustness
- → Consider studying real-world data sets

Perspectives

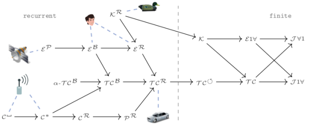
Two natural perspectives:

Structure in dynamic networks (a step further) (Perspective 0)

→ Explore relations among existing and new classes of dynamic networks

→ Focus on $\mathcal{TC}^{\mathcal{R}}$ and $\mathcal{TC}^{\mathcal{B}}$ and robustness

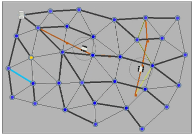
→ Consider studying real-world data sets



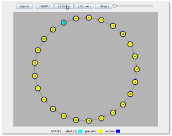
Around JBotSim (Perspective 1)

→ Convergence of tools

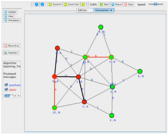
JBotSim



Davis+JBotSim



ViSiDiA

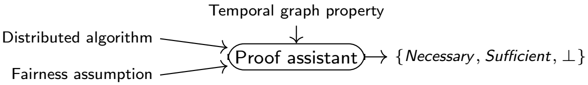


→ Specific extensions, *e.g.* Remote topology viewer through a classroom network

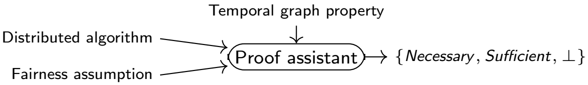
→ Interactive web platform based on Jupyter?

A one-year postdoc is coming (starting Sep. 1, 2018)

Towards formal proofs of temporal requirements (Perspective 2)



Towards formal proofs of temporal requirements (Perspective 2)



Current related efforts:

→ with Event-B (F. Fakhfakh, D. Mery, M. Mosbah, M. Tounsi)
Ex: Proving correctness of our spanning forest algorithm by refinements

→ PADEC: Coq library (K. Altisen, P. Corbineau, S. Devismes) – ANR ESTATE
Ex: Proving correctness of algorithms in the "Locally shared memory" model

Past efforts:

→ LOCO: Coq library (P. Castéran) Ex: Local computation *à la* Métivier

Ambition:

Prove formally that a given property on the dynamics is necessary or sufficient to a given algorithm.

→ Need to pair up with a partner!

Algorithmic movement synthesis (broader view) (Perspective 3)

1) Collective movements which induce temporal structure Already discussed

Synthesize collective movements (*a.k.a.* *mobility models*) such that the resulting graph satisfies temporal properties.

Algorithmic movement synthesis (broader view) (Perspective 3)

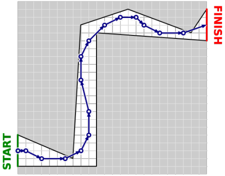
1) Collective movements which induce temporal structure

Already discussed

Synthesize collective movements (*a.k.a.* *mobility models*) such that the resulting graph satisfies temporal properties.

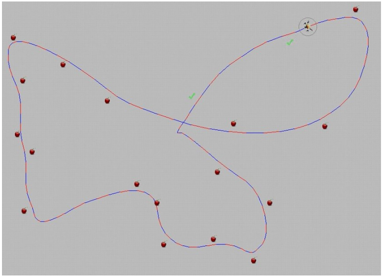
2) Integrating physical constraints in a tractable way

Discrete acceleration models
VECTOR RACER
(paper and pencil game)



→ Impact on problems, *e.g.* TSP

Exploratory work with J. Schoeters and M. Raffinot



Theorem: Acceleration does impact the visit order!

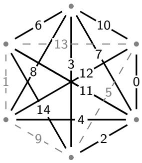
Ambition:

→ Occupy the space between control theory and empirical approaches

Discrete algorithms!

Simplification of temporal cliques

Setting: Complete graph, every edge exists only one instant, remove as many edges as possible while remaining temporally connected (*i.e.*, in $\mathcal{TC}$).



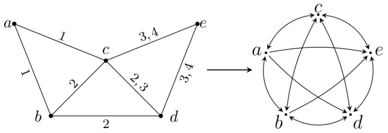
Theorem: $O(n)$ edges can be removed — Akrida *et al.* 2015

Theorem: All but $O(n \log n)$ edges can be removed! — Casteigts *et al.* 2018 (in preparation)

Open question: Is it optimal? Could we remove all but $O(n)$?

Transitive closures of journeys

Def: journey in $\mathcal{G}$ $\iff$ arc in transitive closure.



Question: What is the set of possible transitive closures?

Acknowledgments

PhD students (supervised or co-supervised):

- Jason Schoeters
- Matthieu Barjon and Yessin M. Neggaz (with C. Johnen and S. Chaumette)

PhD students (through collaboration):

- Carlos Gómez Calzado (3 month visit in Bordeaux)
- Jérémie Albert, Ahmed Jedda, Walter Quattrociocchi (joint work in Ottawa)

Masters students:

- Kinda Al Chahid (current)
- Robin Despouys, David Del Campo + 4 of the above

Projects (significant scale):

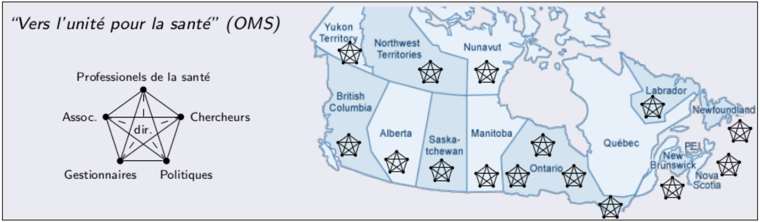
- DRDC W7714-115111/001/SV (Defence Research and Development Canada)
- ANR ESTATE (Enhancing Safety and Self-Stabilization in Time-Varying Distributed Environments)

Other co-authors (alphabetical order):

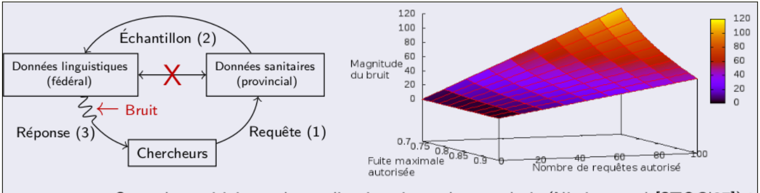
Frédéric Amblard, Lionel Barrère, Jean-Marie Berthelot, Louise Bouchard, Mariette Chartier, Marie-Hélène Chomienne, Swan Dubois, Afonso Ferreira, Paola Flocchini, Colette Johnen, Guy-Vincent Jourdan, Emmanuel Godard, Nishith Goel, Frédéric Guinand, Ralf Klasing, Alberto Lafuente, Mikel Larrea, Bernard Mans, Luke Mathieson, Hussein Mouftah, Yves Métivier, Amiya Nayak, Joseph Peters, Franck Petit, Yoann Pigné, Mike Robson, Nicola Santoro, Ivan Stojmenovic, Alain Trugeon, Jan Warnke, Mark Yamashita, and Akka Zemmari.

Excursions: SHS & Privacy (Chap 8.1 & 8.2)

Health Networks in Canada (with a sociologist L. Bouchard)



Differential Privacy for Linguistic Data (with a physician M.H. Chomienne)



Smooth sensitivity and sampling in private data analysis (Nissim et. al [STOC'07]) ;