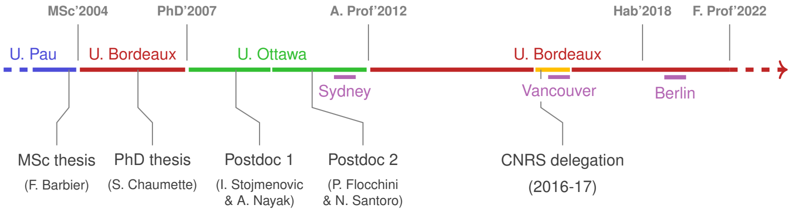
Dynamic networks from inside and outside

Arnaud Casteigts

Public seminar at Unige

January 10, 2023

Overview of academic experience



Current position

- Professor at the university of Bordeaux
- Teaching : *Département informatique de l'IUT* and *UF d'informatique*
- Research : *Laboratoire bordelais de recherche en informatique (LaBRI)*
  > *Algorithms and combinatorics* department
  >> *Distributed algorithms* group (head)
  >> *Graph theory and optimization* group

LaBRI — université de BORDEAUX — cnrs — iut de BORDEAUX — Bordeaux INP AQUITAINE

Main research interests

Main topics :

- Theory of networks
- Distributed algorithms
- Computational complexity
- Dynamic graphs

More recent interests :

- Cryptography & security
- Quantum computing
- Algebraic graph theory

Main teaching activities

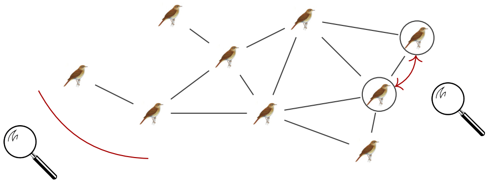
Classical TCS topics :

- Formal languages and automata
- Algorithms and complexity
- Graph theory
- Data structures

Other CS topics :

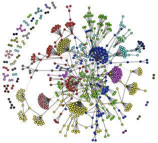
- Android programming
- Operating systems
- Algorithms of mobility
- Low-level programming

Theory of networks



Network as **data**

→ centralized algorithms...

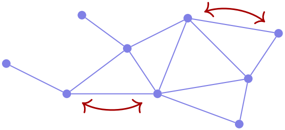


Network as **environment**

→ decentralized algorithms...
(a.k.a. distributed)

Distributed Algorithms

(Think globally, act locally)



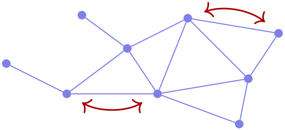
Collaboration of distinct entities to perform a common task.

No centralization available, interactions among neighbors.

Theoretical aspects of collective intelligence.

Distributed Algorithms

(Think globally, act locally)



Collaboration of distinct entities to perform a common task.

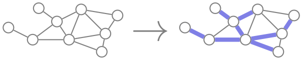
No centralization available, interactions among neighbors.

Theoretical aspects of collective intelligence.

Examples of problems :

Broadcast

Spanning tree



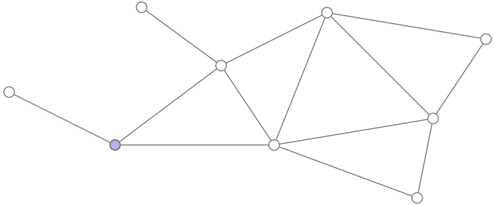
Election

Counting

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

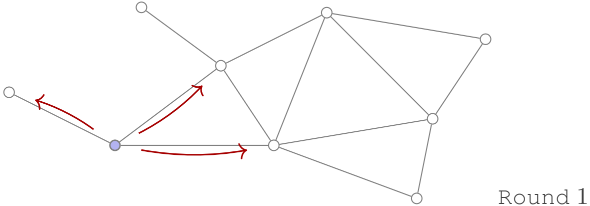
Example : `Broadcasting` → `Spanning tree` → `Counting`

Assumptions : synchronous communication / unique ids / distinguished node



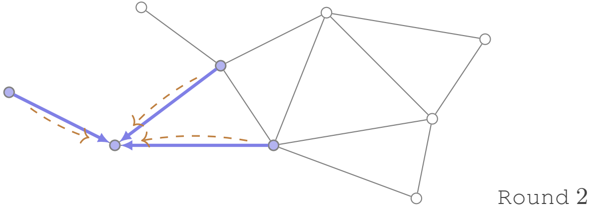
Example : Broadcasting → Spanning tree → Counting

Assumptions : synchronous communication / unique ids / distinguished node



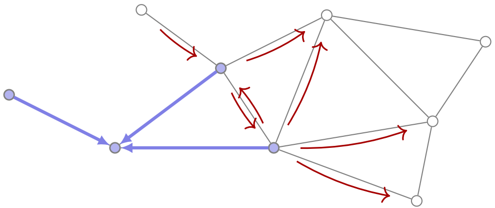
Example : `Broadcasting` → `Spanning tree` → `Counting`

Assumptions : synchronous communication / unique ids / distinguished node



Example : Broadcasting → Spanning tree → Counting

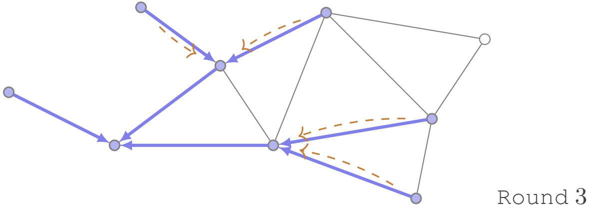
Assumptions : synchronous communication / unique ids / distinguished node



Round 2

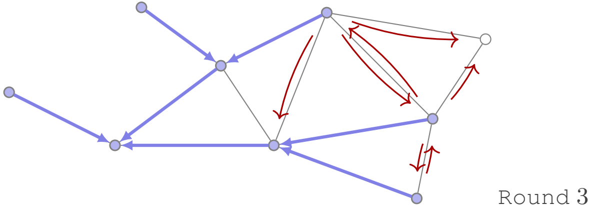
Example : Broadcasting → Spanning tree → Counting

Assumptions : synchronous communication / unique ids / distinguished node



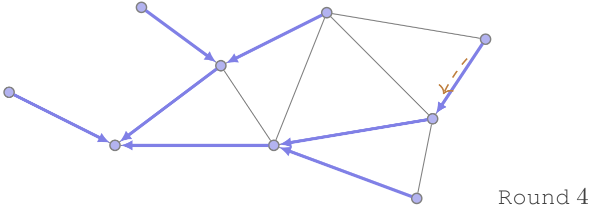
Example : Broadcasting → Spanning tree → Counting

Assumptions : synchronous communication / unique ids / distinguished node



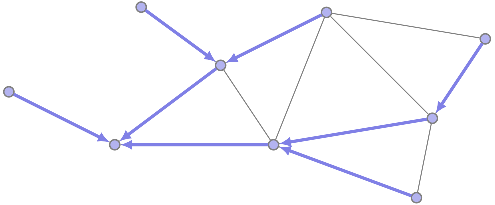
Example : Broadcasting → Spanning tree → Counting

Assumptions : synchronous communication / unique ids / distinguished node



Example : Broadcasting → Spanning tree → Counting

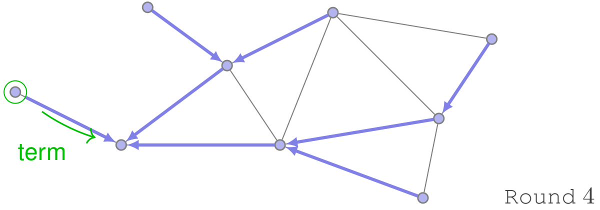
Assumptions : synchronous communication / unique ids / distinguished node



Round 4

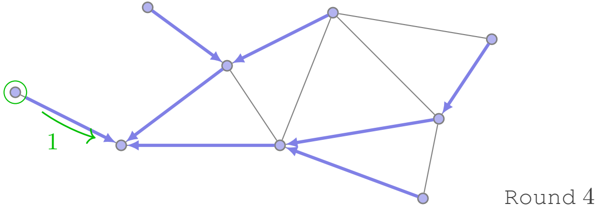
Example : Broadcasting → Spanning tree → Counting

Assumptions : synchronous communication / unique ids / distinguished node



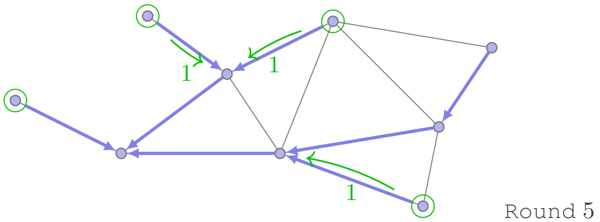
Example : Broadcasting → Spanning tree → Counting

Assumptions : synchronous communication / unique ids / distinguished node



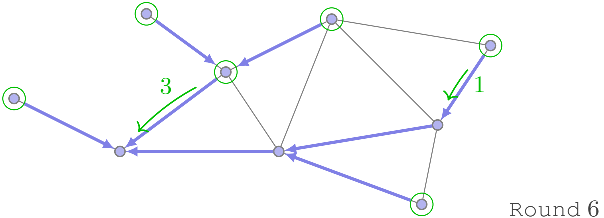
Example : Broadcasting → Spanning tree → Counting

Assumptions : synchronous communication / unique ids / distinguished node



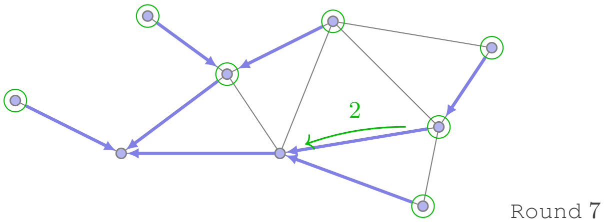
Example : Broadcasting → Spanning tree → Counting

Assumptions : synchronous communication / unique ids / distinguished node



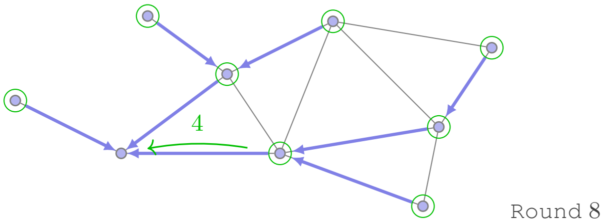
Example : Broadcasting → Spanning tree → Counting

Assumptions : synchronous communication / unique ids / distinguished node



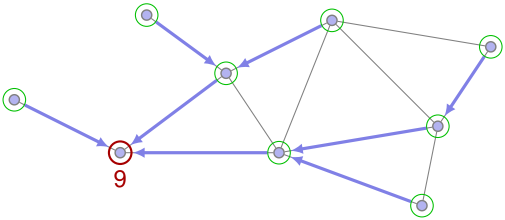
Example : `Broadcasting` → `Spanning tree` → `Counting`

Assumptions : synchronous communication / unique ids / distinguished node



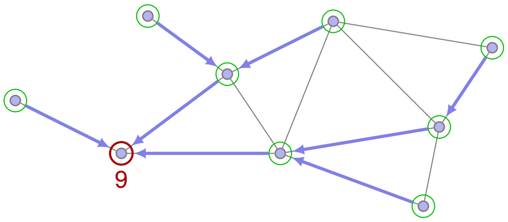
Example : `Broadcasting` → `Spanning tree` → `Counting`

Assumptions : synchronous communication / unique ids / distinguished node



Example : `Broadcasting` → `Spanning tree` → `Counting`

Assumptions : synchronous communication / unique ids / distinguished node

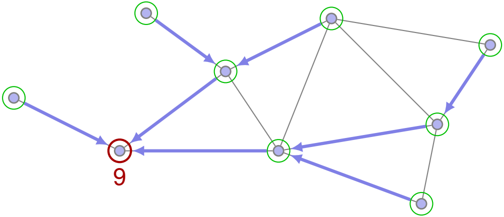


Time complexity : $O(\text{diameter}(G)) = O(n)$ $\qquad n : \#nodes$

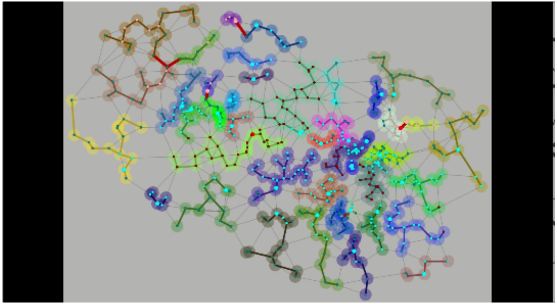
Message complexity : $O(m + n + n) = O(m)$ $\qquad m : \#edges$

Example : `Broadcasting` → `Spanning tree` → `Counting`

Assumptions : synchronous communication / unique ids / distinguished node

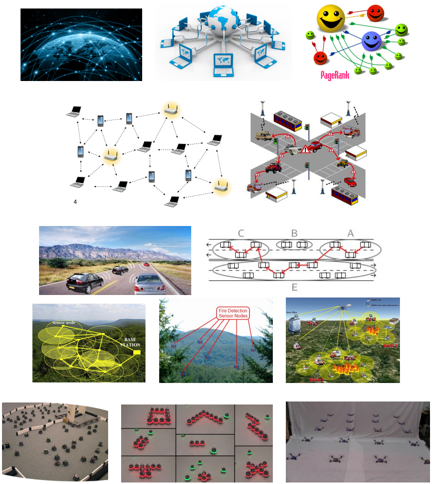


What if there are no distinguished nodes ? (The GHS algorithm, 1983)

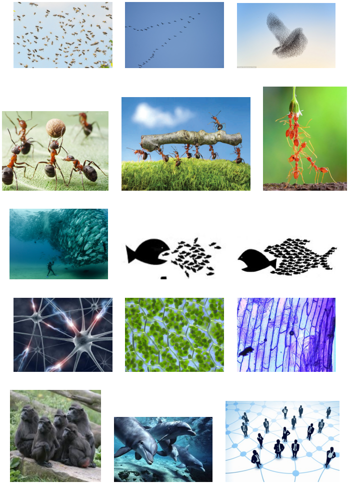


Real-world networks are dynamic

In technologies

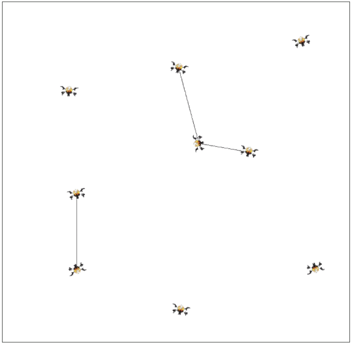


In nature



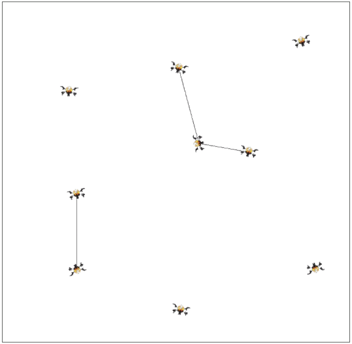
(Highly) dynamic networks ?

Example of scenario

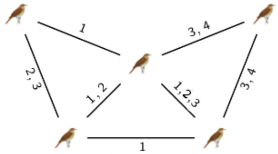


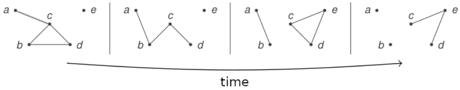
(Highly) dynamic networks ?

Example of scenario



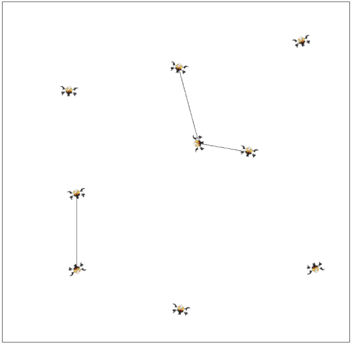
Modeling



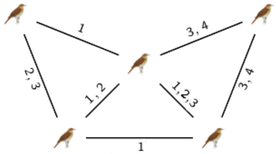


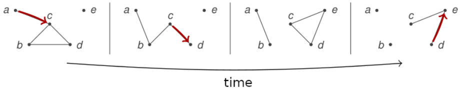
(Highly) dynamic networks ?

Example of scenario



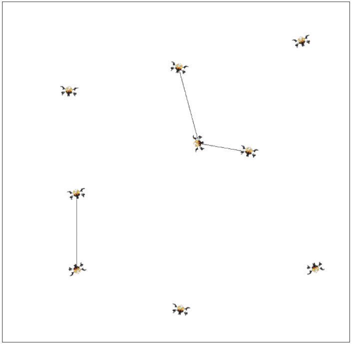
Modeling



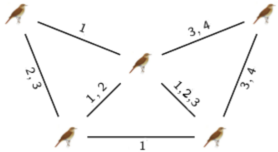


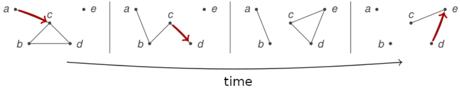
(Highly) dynamic networks ?

Example of scenario



Modeling



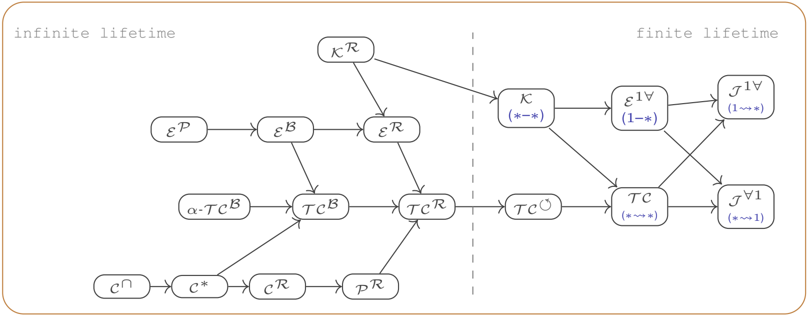


Properties :

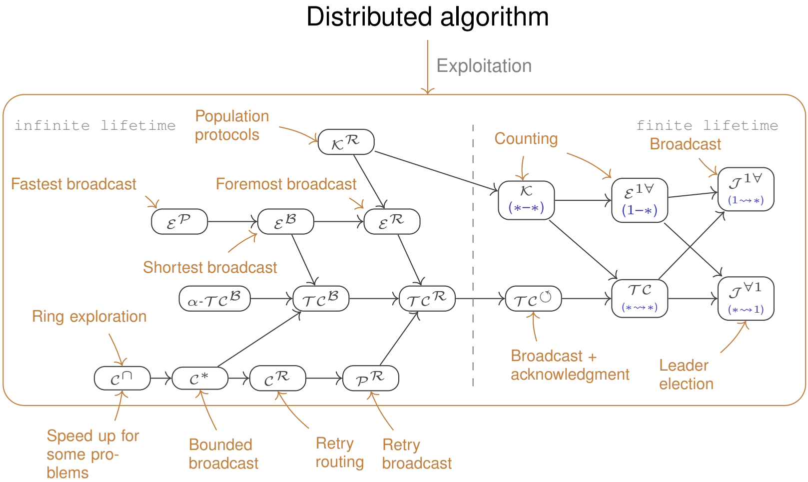
- Temporal connectivity ? $\mathcal{TC}$
- Repeatedly ? $\mathcal{TC}^{\mathcal{R}}$
- Recurrent links ? $\mathcal{E}^{\mathcal{R}}$
- In bounded time ? $\mathcal{E}^{\mathcal{B}}$
- ...

$\rightarrow$ Classes of temporal graphs

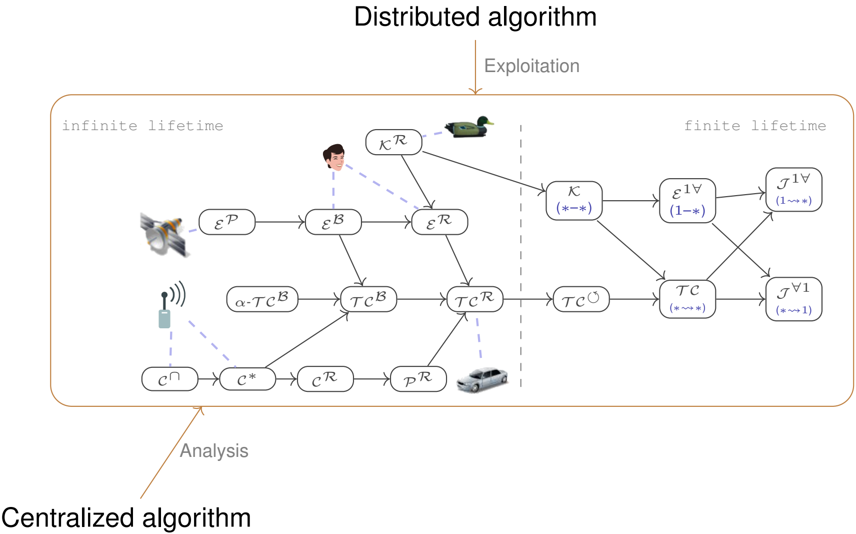
Some classes of temporal graphs



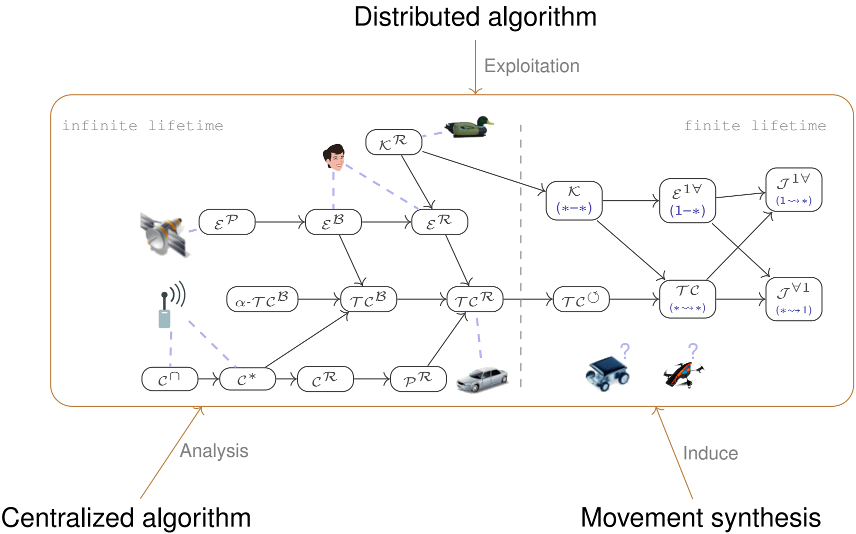
Some classes of temporal graphs



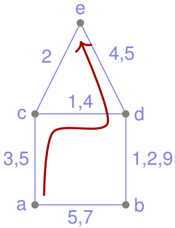
Some classes of temporal graphs



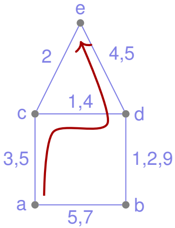
Some classes of temporal graphs



Temporal graphs
for their own sake



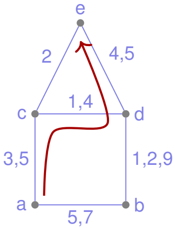
Temporal graphs
for their own sake



Fundamental questions :

- What makes them different ?

Temporal graphs
for their own sake

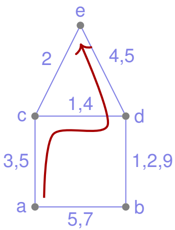


Fundamental questions :

- What makes them different ?

- Why are temporal problems harder ?

Temporal graphs
for their own sake



Fundamental questions :

- What makes them different ?

- Why are temporal problems harder ?

- What techniques work / don't work ?

...

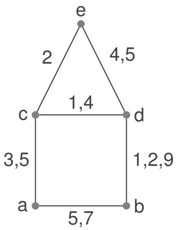
Temporal graphs

(*a.k.a.* time-varying, time-dependent, evolving, dynamic,...)

Basic definition :

$\mathcal{G} = (V, E, \lambda)$, where $\lambda : E \to 2^{\mathbb{N}}$ assigns *presence times* to edges.

Example :



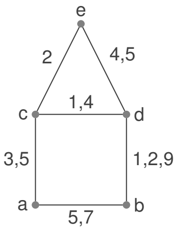
Temporal graphs

(*a.k.a.* time-varying, time-dependent, evolving, dynamic,...)

Basic definition :

$\mathcal{G} = (V, E, \lambda)$, where $\lambda : E \to 2^{\mathbb{N}}$ assigns *presence times* to edges.

Example :



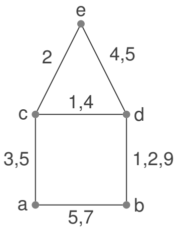
Temporal graphs

(*a.k.a.* time-varying, time-dependent, evolving, dynamic,...)

Basic definition :

$\mathcal{G} = (V, E, \lambda)$, where $\lambda : E \to 2^{\mathbb{N}}$ assigns *presence times* to edges.

Example :



Can also be viewed as a sequence of *snapshots* $\{G_i = \{e \in E : i \in \lambda(e)\}\}$

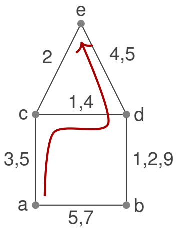
Temporal graphs

(*a.k.a.* time-varying, time-dependent, evolving, dynamic,...)

Basic definition :

$\mathcal{G} = (V, E, \lambda)$, where $\lambda : E \to 2^{\mathbb{N}}$ assigns *presence times* to edges.

Example :



Can also be viewed as a sequence of *snapshots* $\{G_i = \{e \in E : i \in \lambda(e)\}\}$

Temporal paths

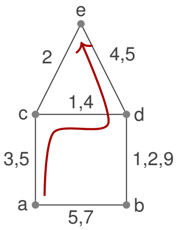
Temporal graphs

(*a.k.a.* time-varying, time-dependent, evolving, dynamic,...)

Basic definition :

$\mathcal{G} = (V, E, \lambda)$, where $\lambda : E \to 2^{\mathbb{N}}$ assigns *presence times* to edges.

Example :



Can also be viewed as a sequence of *snapshots* $\{G_i = \{e \in E : i \in \lambda(e)\}\}$

Temporal paths

- e.g. $\langle (a, c, 3), (c, d, 4), (d, e, 4) \rangle$ (*non-decreasing*)
- e.g. $\langle (a, c, 3), (c, d, 4), (d, e, 5) \rangle$ (*increasing*)

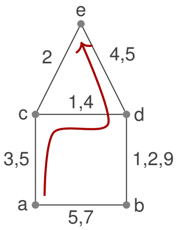
Temporal graphs

(*a.k.a.* time-varying, time-dependent, evolving, dynamic,...)

Basic definition :

$\mathcal{G} = (V, E, \lambda)$, where $\lambda : E \rightarrow 2^{\mathbb{N}}$ assigns *presence times* to edges.

Example :



Can also be viewed as a sequence of *snapshots* $\{G_i = \{e \in E : i \in \lambda(e)\}\}$

Temporal paths

- e.g. $\langle (a, c, 3), (c, d, 4), (d, e, 4) \rangle$ (*non-decreasing*)
- e.g. $\langle (a, c, 3), (c, d, 4), (d, e, 5) \rangle$ (*increasing*)

Temporal connectivity : $\exists$ temporal paths between all vertices.

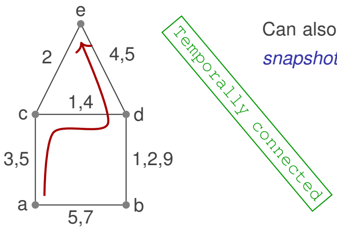
Temporal graphs

(*a.k.a.* time-varying, time-dependent, evolving, dynamic,...)

Basic definition :

$\mathcal{G} = (V, E, \lambda)$, where $\lambda : E \to 2^{\mathbb{N}}$ assigns *presence times* to edges.

Example :



Can also be viewed as a sequence of *snapshots* $\{G_i = \{e \in E : i \in \lambda(e)\}\}$

Temporal paths

- e.g. $\langle (a,c,3),(c,d,4),(d,e,4) \rangle$ (*non-decreasing*)
- e.g. $\langle (a,c,3),(c,d,4),(d,e,5) \rangle$ (*increasing*)

Temporal connectivity : $\exists$ temporal paths between all vertices.

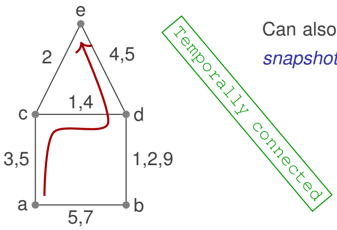
Temporal graphs

(*a.k.a.* time-varying, time-dependent, evolving, dynamic,...)

Basic definition :

$\mathcal{G} = (V, E, \lambda)$, where $\lambda : E \to 2^{\mathbb{N}}$ assigns *presence times* to edges.

Example :



Can also be viewed as a sequence of *snapshots* $\{G_i = \{e \in E : i \in \lambda(e)\}\}$

Temporal paths

- e.g. $\langle (a, c, 3), (c, d, 4), (d, e, 4) \rangle$ (*non-decreasing*)
- e.g. $\langle (a, c, 3), (c, d, 4), (d, e, 5) \rangle$ (*increasing*)

Temporal connectivity : $\exists$ temporal paths between all vertices.

$\rightarrow$ Warning : Reachability is non-symmetrical...

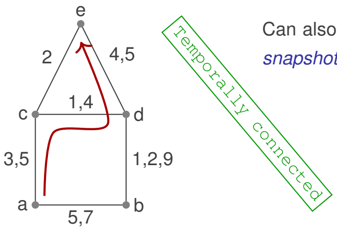
Temporal graphs

(*a.k.a.* time-varying, time-dependent, evolving, dynamic,...)

Basic definition :

$\mathcal{G} = (V, E, \lambda)$, where $\lambda : E \to 2^{\mathbb{N}}$ assigns *presence times* to edges.

Example :



Can also be viewed as a sequence of *snapshots* $\{G_i = \{e \in E : i \in \lambda(e)\}\}$

Temporal paths

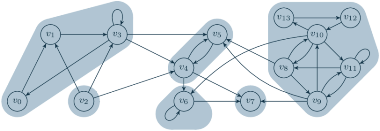
- e.g. $\langle (a, c, 3), (c, d, 4), (d, e, 4) \rangle$ (*non-decreasing*)
- e.g. $\langle (a, c, 3), (c, d, 4), (d, e, 5) \rangle$ (*increasing*)

Temporal connectivity : $\exists$ temporal paths between all vertices.

$\to$ Warning : Reachability is non-symmetrical... and non-transitive !

Impact of non-transitivity (Example : connected components)

In static graphs



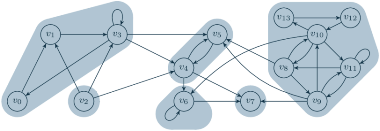
- Components define a partition
- Easy to compute

In temporal graphs

- Maximal components may overlap
- Can be exponentially many

Impact of non-transitivity (Example : connected components)

In static graphs

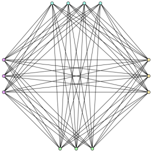


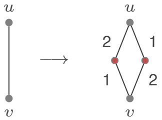
- Components define a partition
- Easy to compute

In temporal graphs

- Maximal components may overlap
- Can be exponentially many

MAX COMPONENT is NP-hard ! (from CLIQUE) Bui-Xuan, Ferreira, Jarry, 2003





- Replace edges with semaphore gadgets
- Cliques become temporal components

Impact of non-transitivity (e.g. spanning structures)

In static graphs

Spanning tree :

- Existence is guaranteed
- Size is always $n - 1$

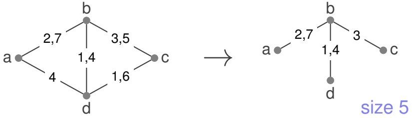
Impact of non-transitivity (e.g. spanning structures)

In static graphs

Spanning tree :

- Existence is guaranteed
- Size is always $n - 1$

In temporal graphs



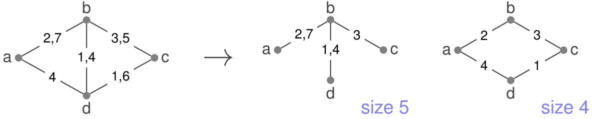
Impact of non-transitivity (e.g. spanning structures)

In static graphs

Spanning tree :

- Existence is guaranteed
- Size is always $n - 1$

In temporal graphs



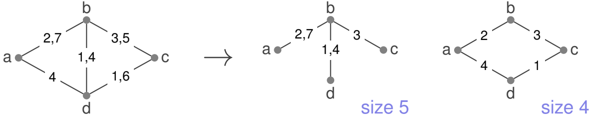
Impact of non-transitivity (e.g. spanning structures)

In static graphs

Spanning tree :

- Existence is guaranteed
- Size is always $n - 1$

In temporal graphs



Minimum size ? (increasingly bad news)

- At least $2n - 4$ Bumby, 1979

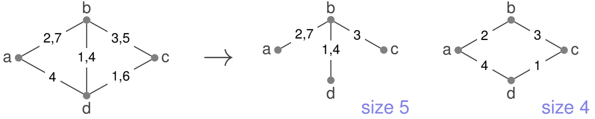
Impact of non-transitivity (e.g. spanning structures)

In static graphs

Spanning tree :

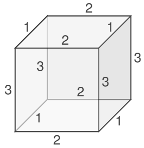
- Existence is guaranteed
- Size is always $n-1$

In temporal graphs



Minimum size ? (increasingly bad news)

- At least $2n-4$ Bumby, 1979
- $\Omega(n \log n)$ Kempe, Kleinberg, Kumar, 2000



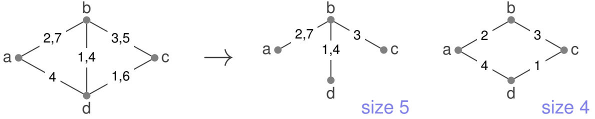
Impact of non-transitivity (e.g. spanning structures)

In static graphs

Spanning tree :

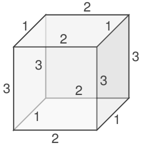
- Existence is guaranteed
- Size is always $n - 1$

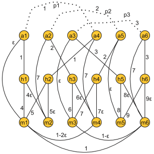
In temporal graphs



Minimum size ? (increasingly bad news)

- At least $2n - 4$ Bumby, 1979
- $\Omega(n \log n)$ Kempe, Kleinberg, Kumar, 2000
- $\Omega(n^2)$ Axiotis, Fotakis, 2016





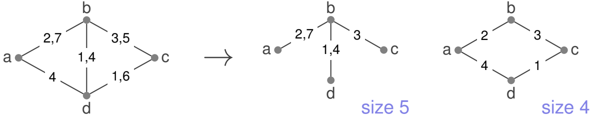
Impact of non-transitivity (e.g. spanning structures)

In static graphs

Spanning tree :

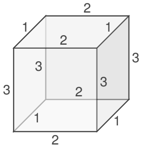
- Existence is guaranteed
- Size is always $n - 1$

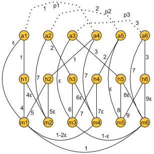
In temporal graphs



Minimum size ? (increasingly bad news)

- At least $2n - 4$ Bumby, 1979
- $\Omega(n \log n)$ Kempe, Kleinberg, Kumar, 2000
- $\Omega(n^2)$ Axiotis, Fotakis, 2016





Computational complexity ?

→ MIN SPANNER is APX-hard ! Akrida, Gąsieniec, Mertzios, Spirakis, 2017

Are there good news for spanning structures ?

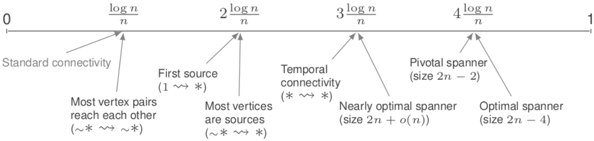
Good news 1 : In **random** temporal graphs, nearly optimal spanners (of size $2n + o(n)$) almost surely exist as soon as the graph is temporally connected !

Are there good news for spanning structures ?

Good news 1 : In **random** temporal graphs, nearly optimal spanners (of size $2n + o(n)$) almost surely exist as soon as the graph is temporally connected !

- ◦ $G \sim G_{n,p}$
- ◦ Permute edges randomly

→ Timeline for p (as $n \to \infty$)



Casteigts, Raskin, Renken, Zamaraev, FOCS 2021

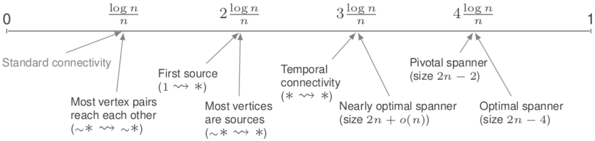
Are there good news for spanning structures ?

Good news 1 : In **random** temporal graphs, nearly optimal spanners (of size $2n + o(n)$) almost surely exist as soon as the graph is temporally connected !

○ $G \sim G_{n,p}$

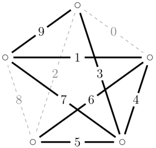
○ Permute edges randomly

→ Timeline for p (as $n \to \infty$)



Casteigts, Raskin, Renken, Zamaraev, FOCS 2021

Good news 2 : Spanners of size $O(n \log n)$ always exist in **complete** temporal graphs



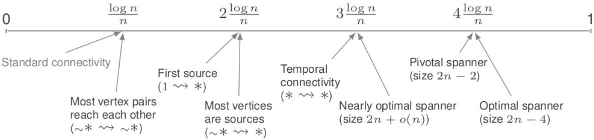
Are there good news for spanning structures ?

Good news 1 : In **random** temporal graphs, nearly optimal spanners (of size $2n + o(n)$) almost surely exist as soon as the graph is temporally connected !

○ $G \sim G_{n,p}$

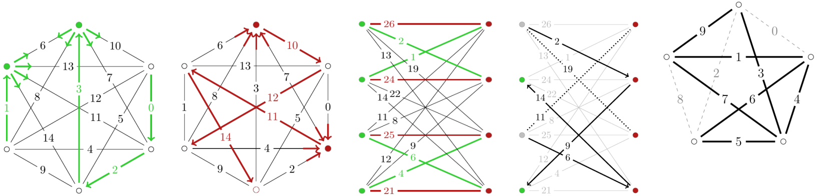
○ Permute edges randomly

→ Timeline for p (as $n \to \infty$)



Casteigts, Raskin, Renken, Zamaraev, FOCS 2021

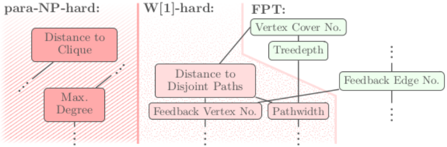
Good news 2 : Spanners of size $O(n \log n)$ always exist in **complete** temporal graphs

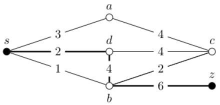


Casteigts, Peters, Schoeters, ICALP 2019

Various impacts of **waiting**

Temporal paths with waiting constraints ISAAC'20 & Algorithmica (2021)



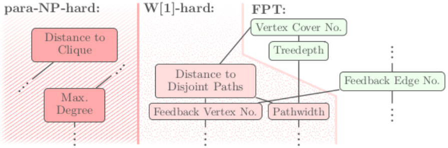


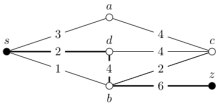
(with A. Himmel, H. Molter, P. Zschoche)

Various impacts of **waiting**

Temporal paths with waiting constraints

ISAAC'20 & Algorithmica (2021)

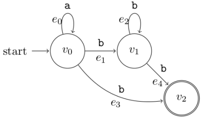




(with A. Himmel, H. Molter, P. Zschoche)

The power of waiting

FCT'13 & Theoretical Computer Science (2015)



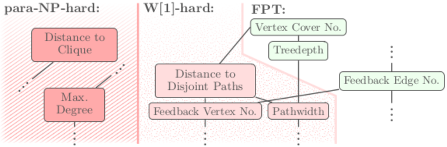
e	Presence $\rho(e,t) = 1$ iff	Latency $\zeta(e,t)$ =
e_0	always true	$(p-1)t$
e_1	$t > p$	$(q-1)t$
e_2	$t \neq p^i q^{i-1}, i > 1$	$(q-1)t$
e_3	$t = p$	any
e_4	$t = p^i q^{i-1},\ i > 1$	any

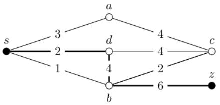
(with P. Flocchini, E. Godard, N. Santoro, M. Yamashita)

Various impacts of **waiting**

Temporal paths with waiting constraints

ISAAC'20 & Algorithmica (2021)

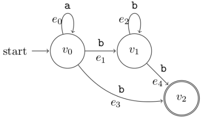




(with A. Himmel, H. Molter, P. Zschoche)

The power of waiting

FCT'13 & Theoretical Computer Science (2015)



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

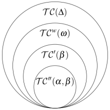
(with P. Flocchini, E. Godard, N. Santoro, M. Yamashita)

Gradual timing assumptions for agreement

EUROPAR'15

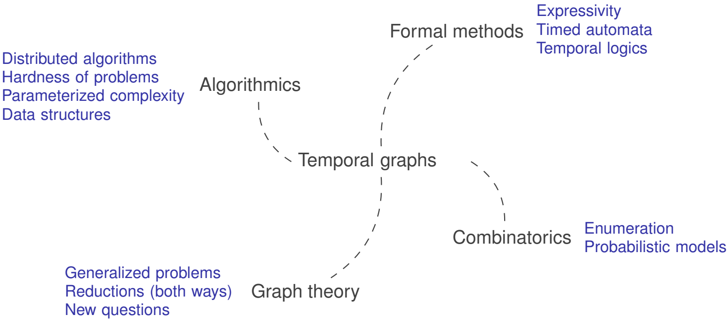
Level of abstraction	Timely Connectivity	Parameters
Level 3 (system)	Δ-components	Δ
Level 2 (multi-hop)	B-journeys, (A, B)-journeys	A and B
Level 1 (link)	(α, β)-journeys	α, β and n

TIMELY CONNECTIVITY DEPENDING ON THE LEVEL OF ABSTRACTION



(with C. Gomez-Calzado, M. Larrea, A. Lafuente)

A new fundamental object



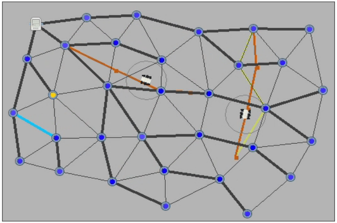
Theoretical boom in the past $\sim$ 5 years.

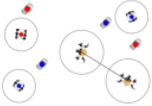
While motivated by real-world applications, it is now studied *per se*.

Theory fed by practical scenarios... and pedagogical practice !

JBotSim : Collective intelligence / Network algorithms / Motion planning / Robotics

(h) Acceleration constraints

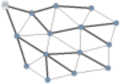




(i) Heterogeneous park cleaning

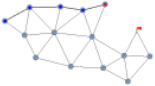
(k) Obstacle management

(o) Navigation (embedding)



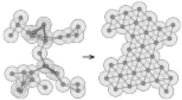
(a) Data aggregation

(b) Vehicular networks



(c) Geograhical routing

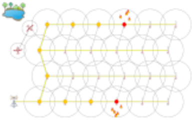
(g) Territory sharing



(d) Deployment by virtual forces

(e) Travelling Salesman Problem

(f) Toroidal space



(j) Fire-fighting aircrafts

Thanks !