

Algorithmique de la mobilité

Parcours RCI (M2)
Université de Bordeaux
(+ version réduite ENSEIRB 2^e année)

Arnaud Casteigts

September 11, 2019

Thèmes du cours



→ Algorithmique dans ce contexte ?

Thèmes du cours



→ Algorithmique dans ce contexte ?

Algorithmes pour :

- ▷ synthèse de mouvements
- ▷ calcul d'itinéraire
- ▷ coordination
- ▷ communications
- ▷ graphes dynamiques

Axes du cours :

- ▷ Scénarios particuliers
(axe "horizontal")
- ▷ Aspects fondamentaux
(axe "vertical")

Thèmes du cours



→ Algorithmique dans ce contexte ?

Algorithmes pour :

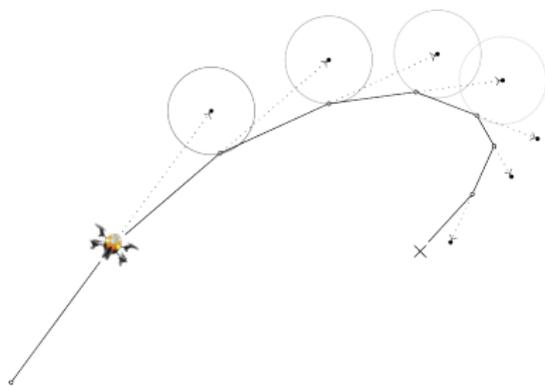
- ▷ synthèse de mouvements
- ▷ calcul d'itinéraire
- ▷ coordination
- ▷ communications
- ▷ graphes dynamiques

Axes du cours :

- ▷ Scénarios particuliers
(axe "horizontal")
- ▷ Aspects fondamentaux
(axe "vertical")

Démarche : Théorie et pratique (implémentation en simulateur)

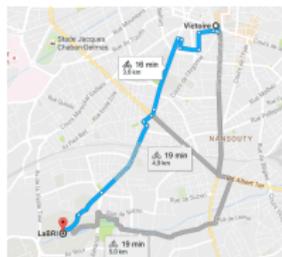
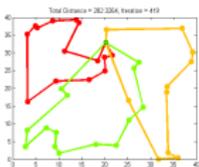
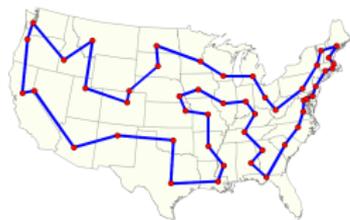
Synthèse de mouvements



Synthèse de mouvements (1)

Macroscopique (\simeq itinéraire)

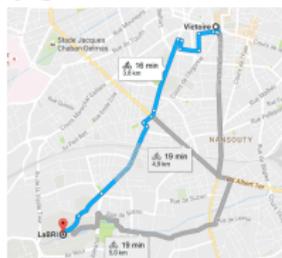
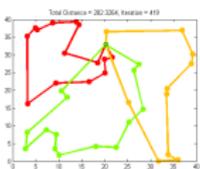
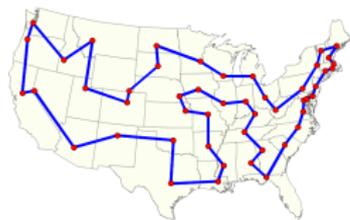
(Ex : Voyageur de commerce, Plus court chemin)



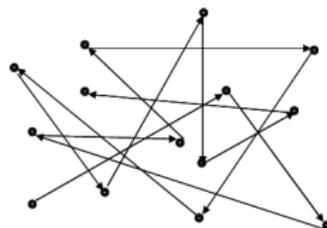
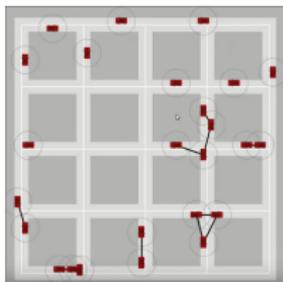
Synthèse de mouvements (1)

Macroscopique (\simeq itinéraire)

(Ex : Voyageur de commerce, Plus court chemin)



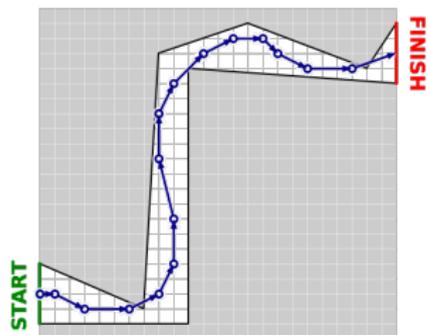
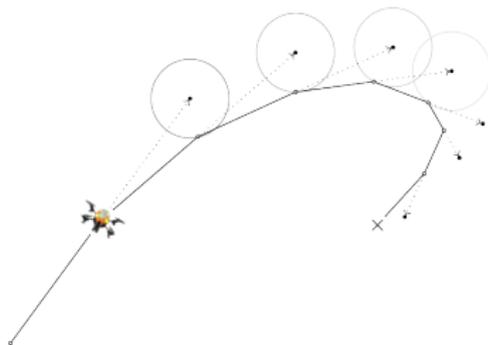
Mésoscopique (modèles de mobilités)



(Ex : Marche aléatoire, Réseaux véhiculaire type Manhattan, Random Waypoint, ...)

Synthèse de mouvements (2)

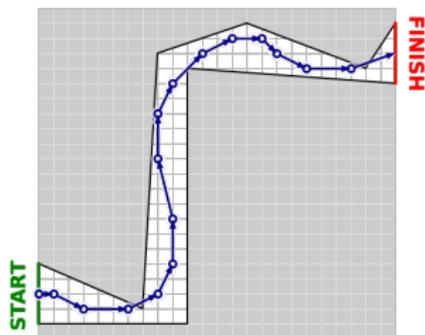
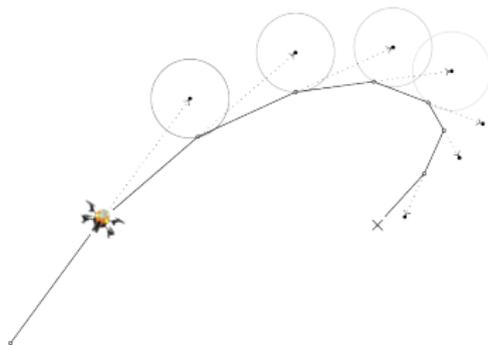
Modèles physiques discrets pour l'accélération



(Recherche de trajectoire optimale)

Synthèse de mouvements (2)

Modèles physiques discrets pour l'accélération



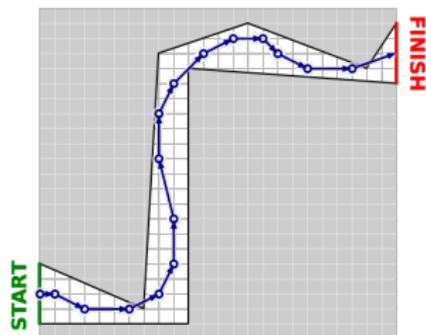
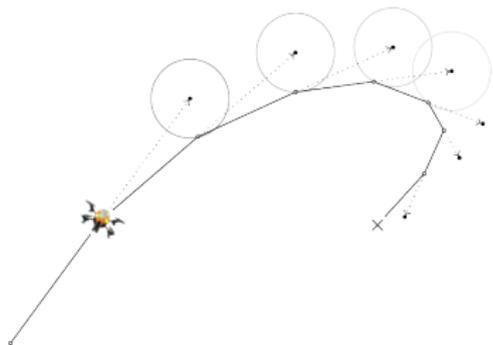
(Recherche de trajectoire optimale)

Exemple d'algorithme: BFS dans l'espace des configurations

("méta-graphe" dont les sommets sont des couples (*position*, *vitesse*))

Synthèse de mouvements (2)

Modèles physiques discrets pour l'accélération

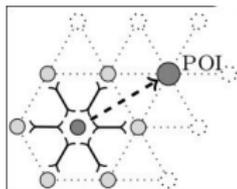
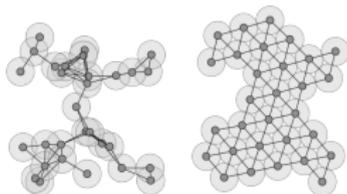


(Recherche de trajectoire optimale)

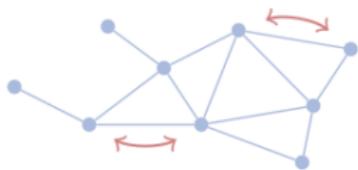
Exemple d'algorithme: BFS dans l'espace des configurations

("méta-graphe" dont les sommets sont des couples (*position*, *vitesse*))

Forces virtuelles artificielles



Coordination et communications

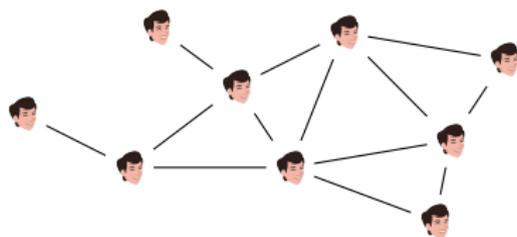


Modélisation des réseaux

On représentera un réseau par un graphe $G = (V, E)$

→ Ensemble de **noeuds** V (*a.k.a.* entités, sommets)

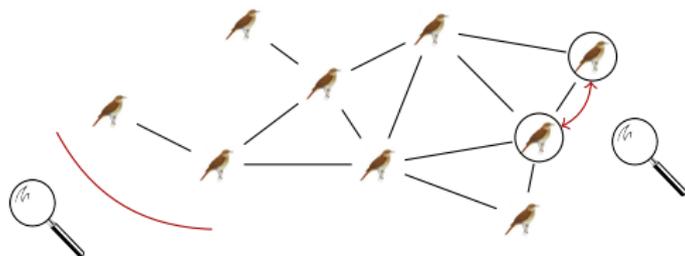
→ Ensemble de **liens** E entre eux (*a.k.a.* relations, arêtes)



Modélisation des réseaux

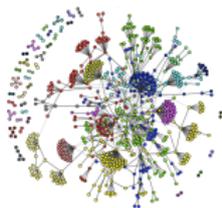
On représentera un réseau par un graphe $G = (V, E)$

- Ensemble de **noeuds** V (a.k.a. entités, sommets)
- Ensemble de **liens** E entre eux (a.k.a. relations, arêtes)



Complex networks

- compute global metrics
- explain and reproduce phenomena

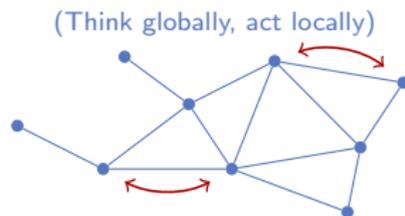


Communication networks

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



Algorithmique distribuée dans les réseaux

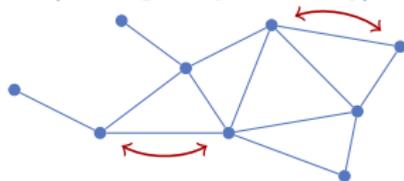


Collaboration d'entités distinctes pour réaliser une tâche commune.

Pas de centralisation.

Algorithmique distribuée dans les réseaux

(Think globally, act locally)



Collaboration d'entités distinctes pour réaliser une tâche commune.

Pas de centralisation.

Exemples de problèmes:

Diffusion



Election



Arbres couvrants

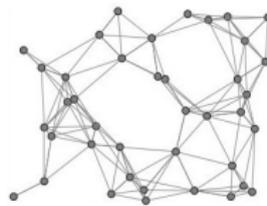


Comptage



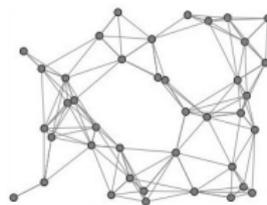
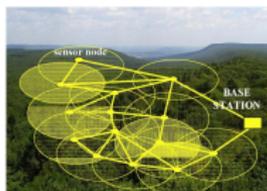
Autres problèmes: consensus, nommage, routage, exploration, ensembles indépendants, ...

Réseaux de capteurs



(agrégation hiérarchique, graphes aléatoires, seuils de connexité)

Réseaux de capteurs

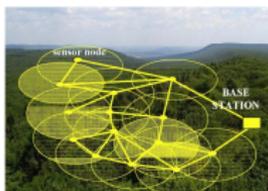


(agrégation hiérarchique, graphes aléatoires, seuils de connectivité)

→ Exemple d'algo récursif dans un arbre d'agrégation:

1) attendre la valeur captée par les enfants, 2) "agréger" les valeurs, 3) transmettre au parent

Réseaux de capteurs

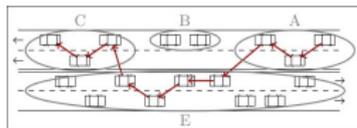
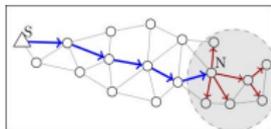


(agrégation hiérarchique, graphes aléatoires, seuils de connexité)

→ Exemple d'algo récursif dans un arbre d'agrégation:

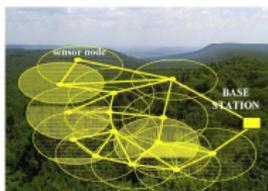
1) attendre la valeur captée par les enfants, 2) "agréger" les valeurs, 3) transmettre au parent

Routage géographique



(information localisée, géorouting, géocasting)

Réseaux de capteurs

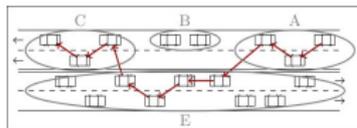
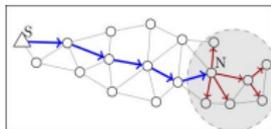


(agrégation hiérarchique, graphes aléatoires, seuils de connexité)

→ Exemple d'algo récursif dans un arbre d'agrégation:

1) attendre la valeur captée par les enfants, 2) "agréger" les valeurs, 3) transmettre au parent

Routage géographique



(information localisée, géorouting, géocasting)

→ Exemple d'algo de géorouting : transmettre au voisin le plus proche de la destination (répéter)

Graphes dynamiques

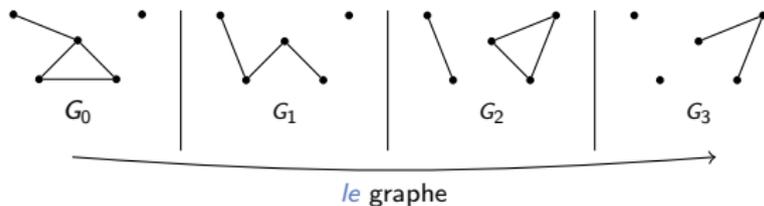
Représentation du réseau



Graphes dynamiques

Représentation du réseau

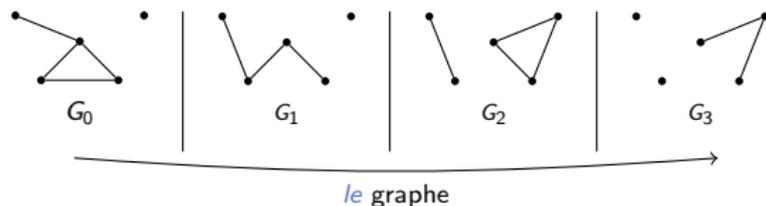
Un modèle simple : une suite de graphe $\mathcal{G} = G_0, G_1, \dots$



Graphes dynamiques

Représentation du réseau

Un modèle simple : une suite de graphes $\mathcal{G} = G_0, G_1, \dots$



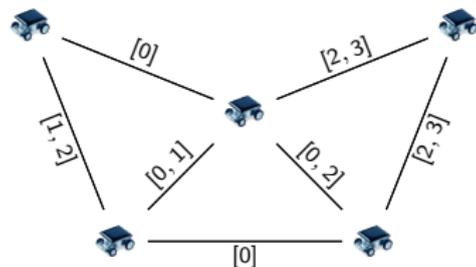
Autre modèle: graphe muni d'une fonction de présence:

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

- $\mathcal{T} \subseteq \mathbb{N}/\mathbb{R}$ (lifetime)

- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ (fonction de présence)

[+ autres fonctions]



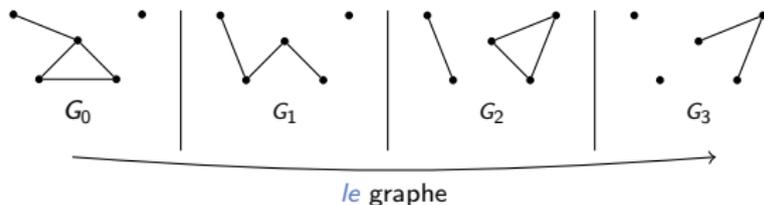
Différentes terminologies :

dynamic graphs, time-varying graphs, evolving graphs, temporal graphs, etc.

Graphes dynamiques

Représentation du réseau

Un modèle simple : une suite de graphes $\mathcal{G} = G_0, G_1, \dots$



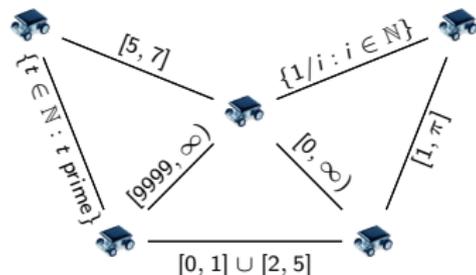
Autre modèle: graphe muni d'une fonction de présence:

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

- $\mathcal{T} \subseteq \mathbb{N}/\mathbb{R}$ (lifetime)

- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ (fonction de présence)

[+ autres fonctions]

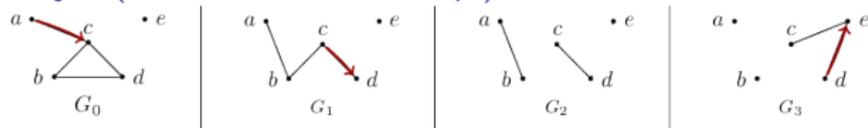


Différentes terminologies :

dynamic graphs, time-varying graphs, evolving graphs, temporal graphs, etc.

Concepts temporels

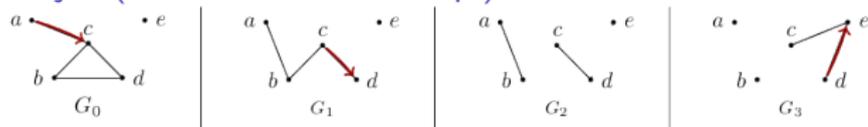
Trajets (chemins à travers le temps)



→ Connexité temporelle

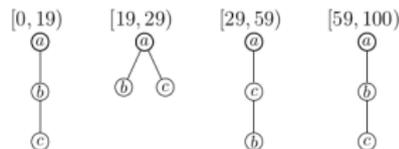
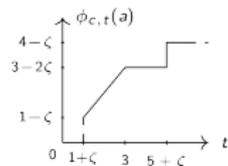
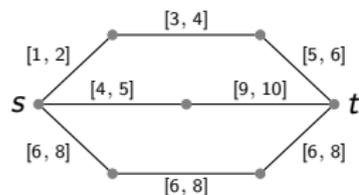
Concepts temporels

Trajets (chemins à travers le temps)



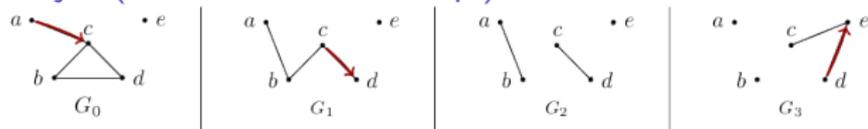
→ Connexité temporelle

Distance temporelle et plus courts chemins



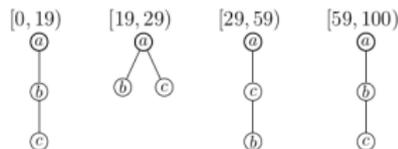
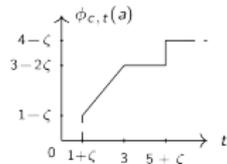
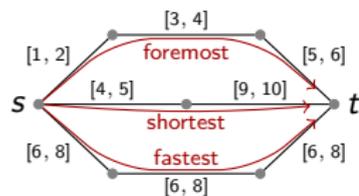
Concepts temporels

Trajets (chemins à travers le temps)



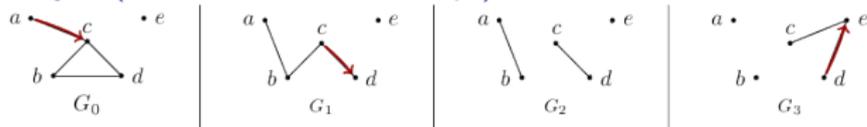
→ Connexité temporelle

Distance temporelle et plus courts chemins



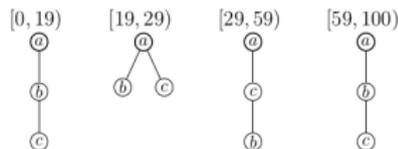
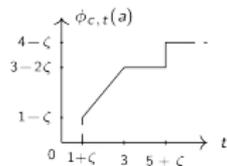
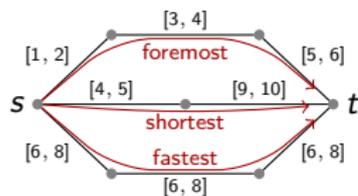
Concepts temporels

Trajets (chemins à travers le temps)

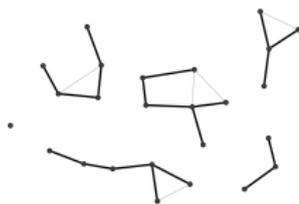


→ Connexité temporelle

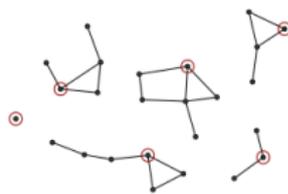
Distance temporelle et plus courts chemins



Re-définition de problèmes classiques (2 exemples)



Arbres couvrants



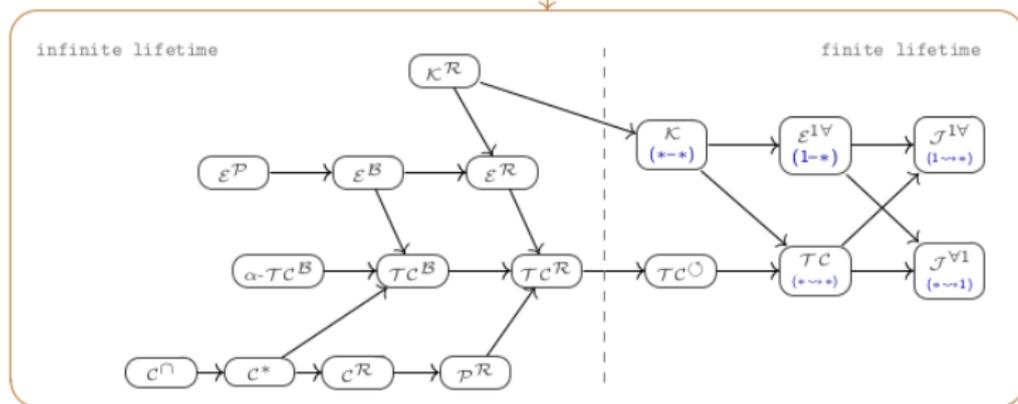
Élection

Ex: Les liens réapparaissent-ils ? Sont-ils périodiques ? Tout le monde peut-il joindre tout le monde indirectement ? Et directement ? Le réseau est-il toujours connexe ?

Ex: Les liens réapparaissent-ils ? Sont-ils périodiques ? Tout le monde peut-il joindre tout le monde indirectement ? Et directement ? Le réseau est-il toujours connexe ?

Network algorithms

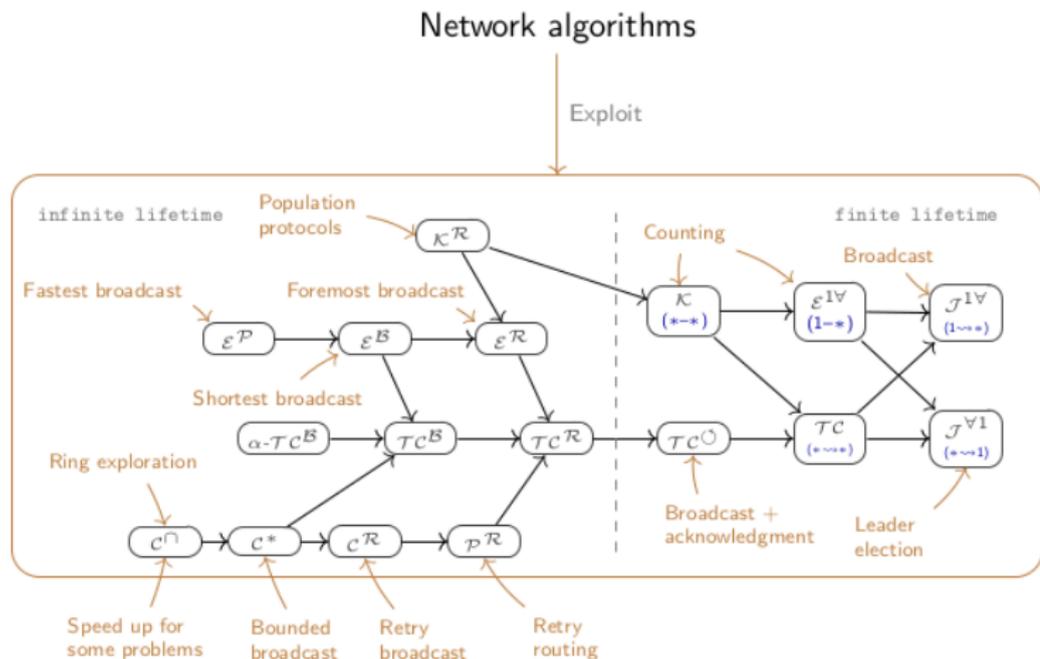
Exploit



Propriétés temporelles

(Classes de graphes dynamiques)

Ex: Les liens réapparaissent-ils ? Sont-ils périodiques ? Tout le monde peut-il joindre tout le monde indirectement ? Et directement ? Le réseau est-il toujours connexe ?



→ Impact sur les problèmes et les algorithmes ?

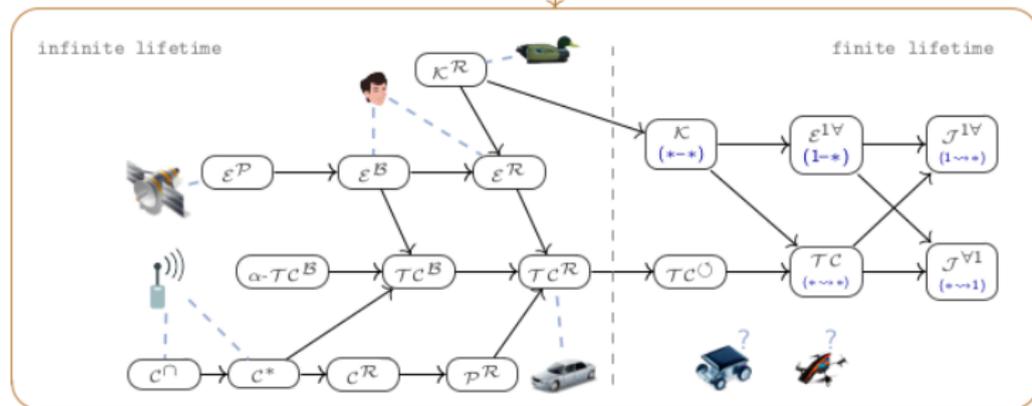
Propriétés temporelles

(Classes de graphes dynamiques)

Ex: Les liens réapparaissent-ils ? Sont-ils périodiques ? Tout le monde peut-il rejoindre tout le monde indirectement ? Et directement ? Le réseau est-il toujours connexe ?

Network algorithms

Exploit



Test

Data analysis

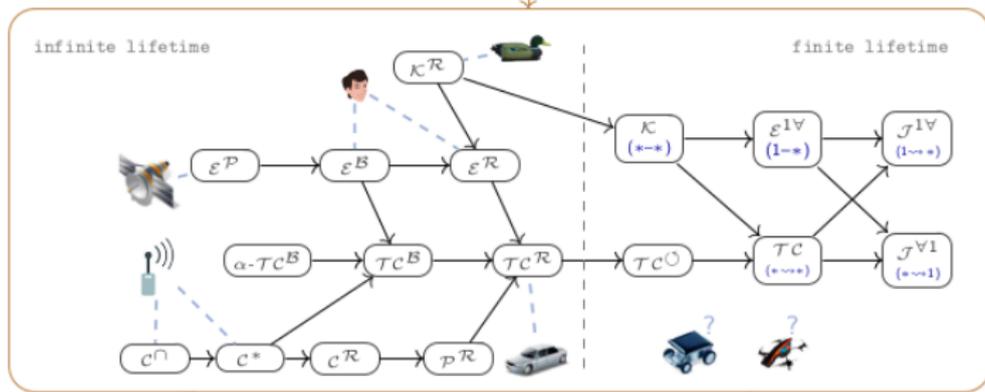
Propriétés temporelles

(Classes de graphes dynamiques)

Ex: Les liens réapparaissent-ils ? Sont-ils périodiques ? Tout le monde peut-il rejoindre tout le monde indirectement ? Et directement ? Le réseau est-il toujours connexe ?

Network algorithms

Exploit



Test

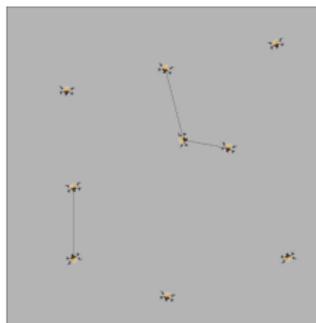
Induce

Centralized algorithms

Movement synthesis

Mouvements collectifs qui induisent des propriétés temporelles

Synthétiser des mouvements collectifs (*a.k.a.* *modèles de mobilités*) qui garantissent des propriétés sur le graphe dynamique induit.



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

Mouvements collectifs qui induisent des propriétés temporelles

Synthétiser des mouvements collectifs (*a.k.a.* *modèles de mobilités*) qui garantissent des propriétés sur le graphe dynamique induit.



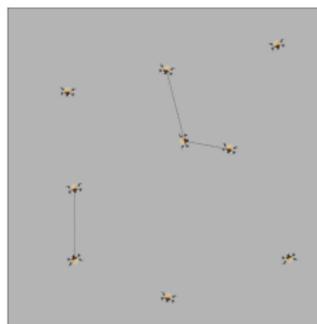
← this network $\in \mathcal{ER}$

this one $\notin \mathcal{AC}$ →



Mouvements collectifs qui induisent des propriétés temporelles

Synthétiser des mouvements collectifs (*a.k.a.* *modèles de mobilités*) qui garantissent des propriétés sur le graphe dynamique induit.



← this network $\in \mathcal{ER}$

this one $\notin \mathcal{AC}$ →



Objectif double:

- ▷ Le modèle de mobilité lui-même
- ▷ Combiné à la réalisation d'une mission collective (p.ex. exploration)

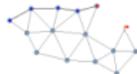
Une cible intéressante: \mathcal{TC}^B (diamètre temporel borné) → détection de crash



(a) Data aggregation



(b) Vehicular networks



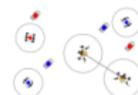
(c) Geographical routing



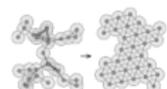
(g) Territory sharing based on Voronoi



(h) Acceleration constraints



(i) Heterogeneous park cleaning



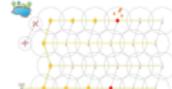
(d) Deployment by virtual forces



(e) Travelling Salesman Problem



(f) Toroidal space



(j) Fire-fighting aircrafts

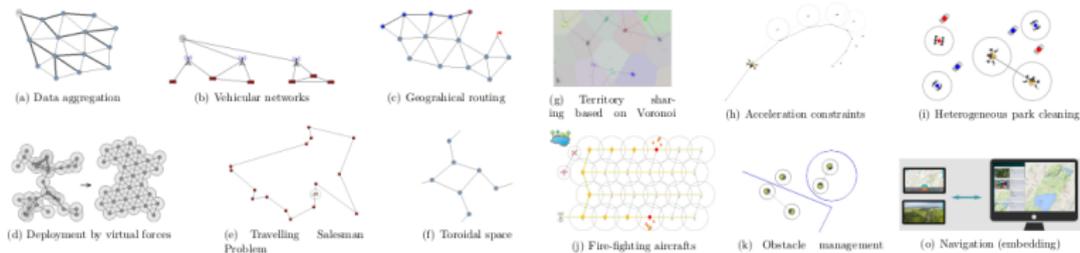


(k) Obstacle management



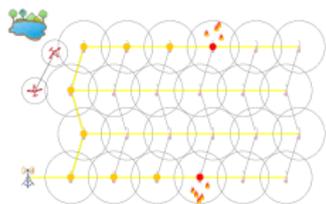
(o) Navigation (embedding)

Interactive, extensible, event-driven programming (java)

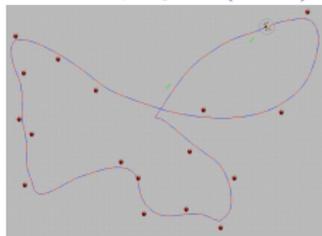


Interactive, extensible, event-driven programming (java)

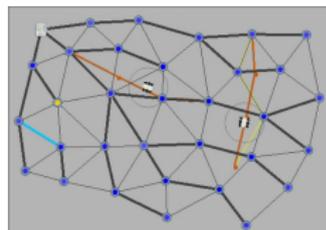
Student project (2014)



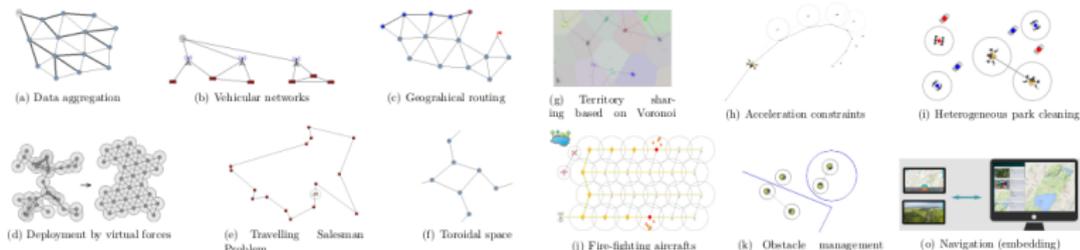
Student project (2016)



Student project (2017)

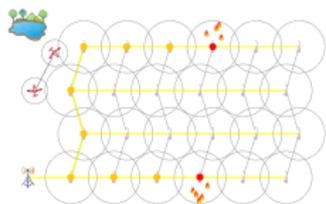


+ Réseaux véhiculaires (2018)

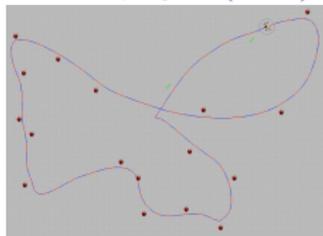


Interactive, extensible, event-driven programming (java)

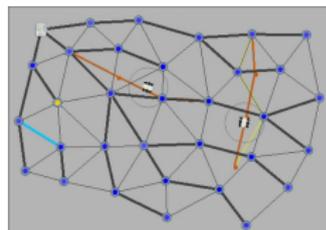
Student project (2014)



Student project (2016)



Student project (2017)



+ Réseaux véhiculaires (2018)

Projet cette année ? Si oui, se substituera au TP rendu.

→ Exploration collective avec obstacles et accélération ?

→ Autres idées ?

Informations pratiques (et récapitulatif)

- ▷ 12 semaines \simeq 40% cours intégré, 60% séances machine
- ▷ Enseignants: Arnaud Casteigts et Jason Schoeters (Cours/TD/TP)
- ▷ URL du cours: (“algomob” sur google)
`http://www.labri.fr/perso/acasteig/teaching/algomob/`
→ Source principale d'information (**à consulter avant chaque cours**)
- ▷ Philosophie du cours: théorie + pratique (algorithmique + simulation)
- ▷ Évaluation : 2 questionnaires + 1 TP rendu (ou projet)
- ▷ Possibilité d'influencer le cours ! Vos idées sont les bienvenues.