

Algorithmique de la mobilité

Parcours RCI (M2)
Université de Bordeaux

Arnaud Casteigts

September 14, 2018

Informations générales sur le cours

- ▷ 12 semaines (x 4 heures).
- ▷ Le jeudi matin
- ▷ Arnaud Casteigts (Cours/TD) + Rémi Laplace (TD/TP)
- ▷ Autres collègues (ponctuellement)
- ▷ Philosophie: d'abord la pratique, puis la théorie
(En théorie, la pratique et la théorie sont identiques; mais en pratique, elles sont différentes...)
- ▷ Évaluation : 1 TP rendu (ou deux) + 2 questionnaires
- ▷ URL <http://www.labri.fr/perso/acasteig/teaching/algomob/>
("algomob" sur google)
→ Source principale d'information (**à consulter avant chaque cours**)

Thèmes du cours



→ Algorithmique dans ce contexte ?

Thèmes du cours



→ Algorithmique dans ce contexte ?

Algorithmes pour :

- ▷ le routage
- ▷ le calcul d'itinéraires
- ▷ les mouvements collectifs
- ▷ l'auto-organisation
- ▷ les interactions
- ▷ ...

Thèmes du cours



→ Algorithmique dans ce contexte ?

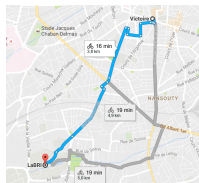
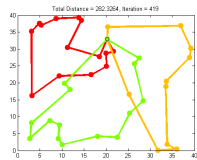
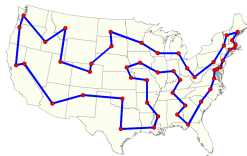
Algorithmes pour :

- ▷ le routage
- ▷ le calcul d'itinéraires
- ▷ les mouvements collectifs
- ▷ l'auto-organisation
- ▷ les interactions
- ▷ ...

Axes du cours :

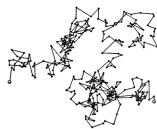
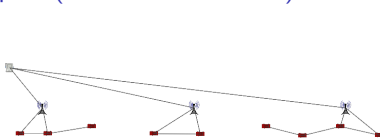
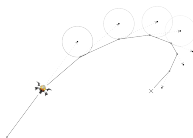
- ▷ Début: Cas applicatifs
("horizontal")
- ▷ Puis: Exploration théorique
("vertical")

Plannification



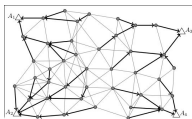
(TSP euclidien, recherche de plus court chemin)

Génération dans le plan (modèles de mobilité)



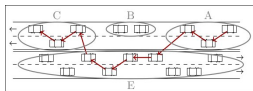
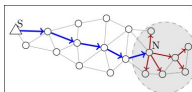
(p.ex.: contraintes d'accélération, véhicules, marches aléatoires, ...)

Ex: Aggrégation de données (réseaux de capteurs)



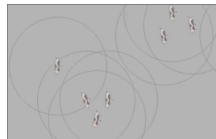
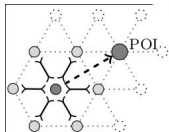
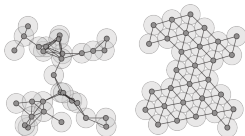
(construction distribuée, agrégation hiérarchique, seuils de connexité)

Ex: Routage géographique

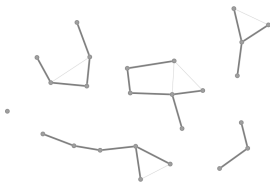


(information localisée, géorouting, géocasting)

Ex: Forces virtuelles (artificielles ou naturelles)



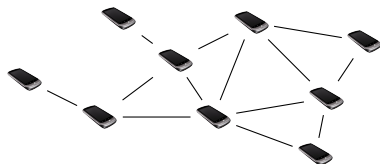
Contenu théorique



Réseau = graphe (dans ce cours)

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

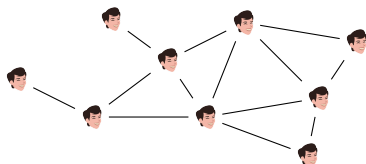
→ A **network** (or graph) $G = (V, E)$



Réseau = graphe (dans ce cours)

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

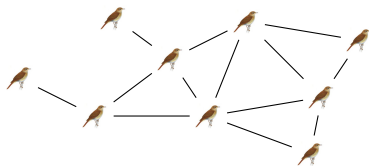
→ A **network** (or graph) $G = (V, E)$



Réseau = graphe (dans ce cours)

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

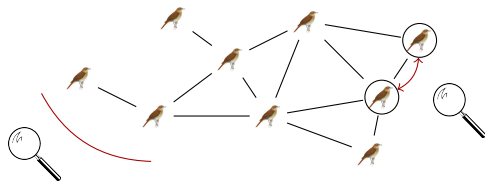
→ A **network** (or graph) $G = (V, E)$



Réseau = graphe (dans ce cours)

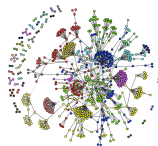
- Set of nodes V (a.k.a. entities, vertices)
- Set of links E among them (a.k.a. relations, edges)

→ A network (or graph) $G = (V, E)$



Complex networks

- compute global metrics
- explain and reproduce phenomena



Communication networks

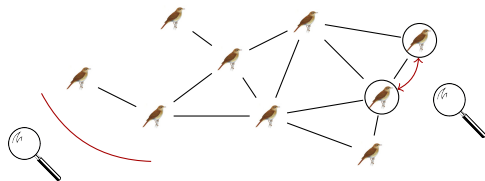
- design interactions among entities
- study what can be done *from within*



Réseau = graphe (dans ce cours)

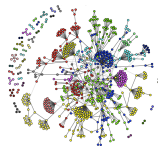
- Set of **nodes** V (a.k.a. entities, vertices)
- Set of **links** E among them (a.k.a. relations, edges)

→ A **network** (or graph) $G = (V, E)$



Complex networks

- compute global metrics
- explain and reproduce phenomena



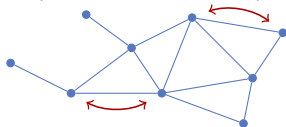
Communication networks

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



Algorithmique distribuée

(Think globally, act locally)

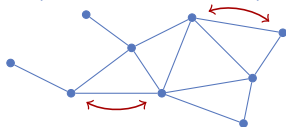


Collaboration of distinct entities to perform a common task.

No centralization available.

Algorithmique distribuée

(Think globally, act locally)

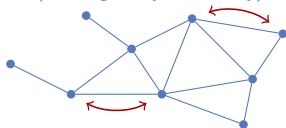


Collaboration of distinct entities to perform a common task.

No centralization available.

Algorithmique distribuée

(Think globally, act locally)



Collaboration of distinct entities to perform a common task.

No centralization available.

Exemples de problèmes:

Diffusion



Election



Arbres couvrants



Comptage

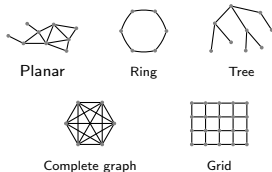


Consensus, nommage, routage, exploration, ensembles indépendants, ...

Se donner de la structure

Propriétés supplémentaires qu'un algorithme peut exploiter.

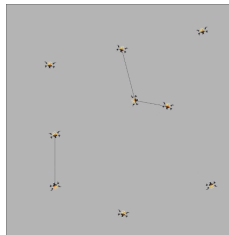
Réseaux statiques



Réseaux très dynamiques



Quel type de structure ?



Graphes dynamiques

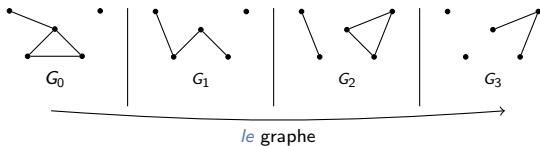
Représentation du réseau



Graphes dynamiques

Représentation du réseau

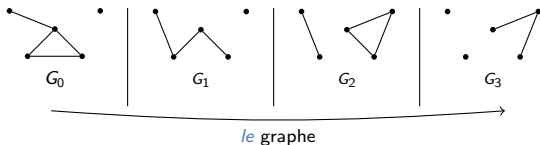
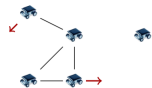
Un modèle simple : une suite de graphe $\mathcal{G} = G_0, G_1, \dots$



Graphes dynamiques

Représentation du réseau

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



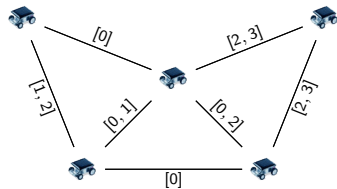
Autre modèle: graphe muni d'une fonction de présence:

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

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

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

[+ autres fonctions]



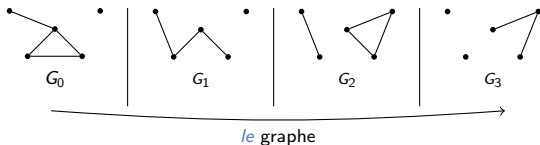
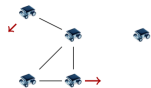
Variété de terminologies :

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

Graphes dynamiques

Représentation du réseau

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



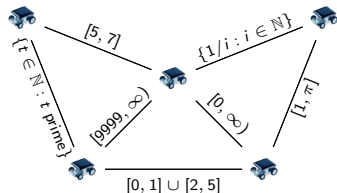
Autre modèle: graphe muni d'une fonction de présence:

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

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

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

[+ autres fonctions]

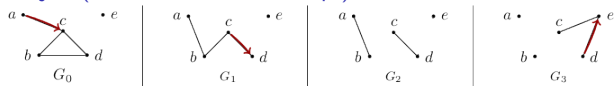


Variété de terminologies :

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

Concepts temporels

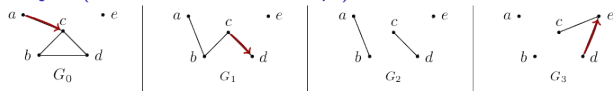
Trajets (chemins à travers le temps)



→ Connexité temporelle

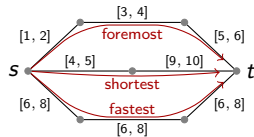
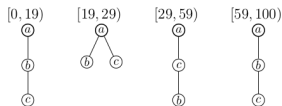
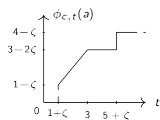
Concepts temporels

Trajets (chemins à travers le temps)



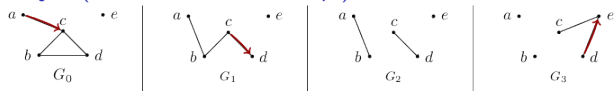
→ Connexité temporelle

Distance temporelle et plus courts chemins



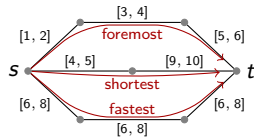
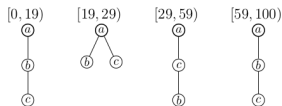
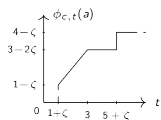
Concepts temporels

Trajets (chemins à travers le temps)

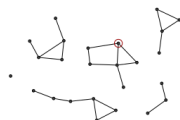


→ Connexité temporelle

Distance temporelle et plus courts chemins



Re-définition de problèmes classiques (2 exemples)



Élection



Arbres couvrants

Répartition des séances

40% de cours intégré et 60% de séances machine

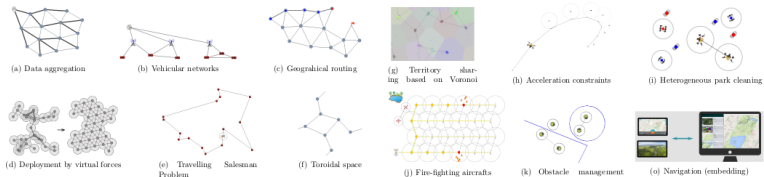
Principe de progression :

- ▷ Début: algorithmes spécialisés (problèmes applicatifs)
- ▷ Puis: problèmes fondamentaux et graphes dynamiques

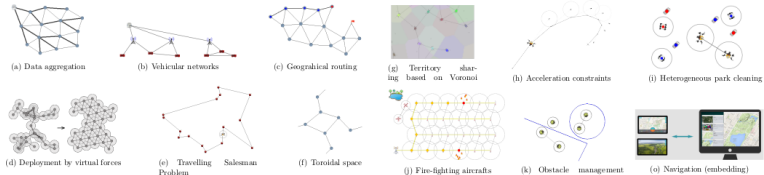
Évaluation : un TP rendu (ou deux) + deux questionnaires

Séances machine:

- bibliothèque JBotSim
- langage: java
- paradigme: surcharge "à la Android"

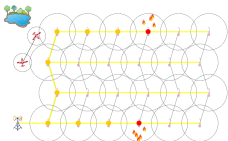


Interactive, extensible, event-driven programming (java)

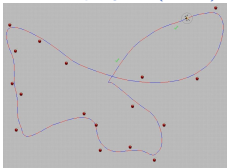


Interactive, extensible, event-driven programming (java)

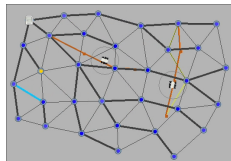
Student project (2014)

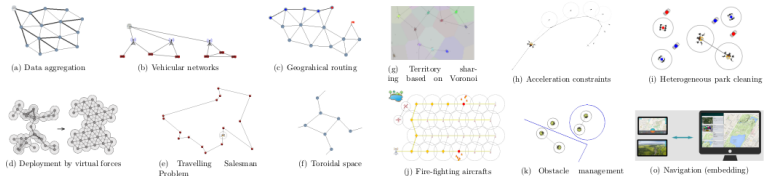


Student project (2016)



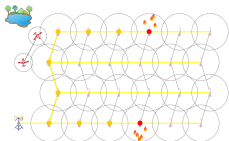
Student project (2017)



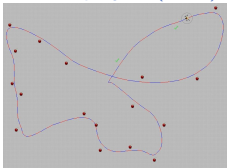


Interactive, extensible, event-driven programming (java)

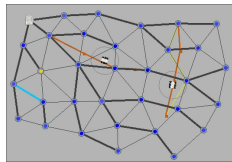
Student project (2014)



Student project (2016)



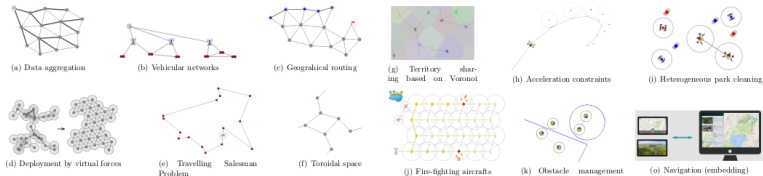
Student project (2017)



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

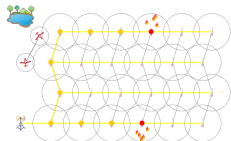
→ Réseaux véhiculaires (maintien de message d'alerte géolocalisé)

→ Algorithme d'exploration collective (p.ex.: robots aspirateurs)

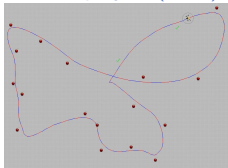


Interactive, extensible, event-driven programming (java)

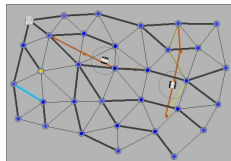
Student project (2014)



Student project (2016)



Student project (2017)



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

→ Réseaux véhiculaires (maintien de message d'alerte géolocalisé)

→ Algorithme d'exploration collective (p.ex.: robots aspirateurs)



questions ?