

# Analyse de Réseaux

Matthieu Latapy

Juin 2020

# Plan

## Préliminaires

# Objectifs du cours

**vous donner une *intuition***

et des outils, et des méthodes...

**solide**

théorie et pratique

**de l'approche réseau**

data science / algorithmique



***une histoire***

**un exemple, un cheminement**

en lien avec vos objets

**formalismes, algorithmes, code**

un peu de maths, de stats, de prog

# Fonctionnement du cours

## Plan

Preliminaires

Penser reseau

Faire parler un reseau

Proprietes avancees

Robustesse

Aller plus loin

Encore plus loin

Conclusion

bilan en fin de parties

**questions libres**

# Fonctionnement du cours

## Plan

Préliminaires

Penser réseaux

Faire parler un réseau

Propriétés avancées

Robustesse

Aller plus loin

Encore plus loin

Conclusion

bilan en fin de parties

**questions libres**

**allers-retours cours ↔ manipulations**

transparents + python

(test écran partagé)

<http://latapy.complexnetworks.fr/ESGI/>

# Suivi / Evaluation

**exercices réguliers**

↔ contrôle continu

**examens d'étapes**

↔ note finale

# Suivi / Evaluation

**exercices réguliers**

↔ contrôle continu

**examens d'étapes**

↔ note finale

<http://latapy.complexnetworks.fr/ESGI/VotrePrenom>

réponses courtes, textuelles, *à la main*  
à [matthieu.latapy@lip6.fr](mailto:matthieu.latapy@lip6.fr) avec *subject* clair

**tester**

(appel)

**attention : sujets individuels et originaux**

# Plan

## Penser réseaux

Intuition

Exemples

Cas pratiques

Réseau vs dessins

Définition formelle

Programmation

Exercice

Bilan



# Qu'est-ce qu'un réseau ?

liens / relations  
entre des objets

# Qu'est-ce qu'un réseau ?

## liens / relations entre des objets

"objets" ?

individus, machines, logiciels, lieux,  
entreprises, documents, animaux, ...

"liens, relations" ?

amitié, collaboration, câbles, appels, routes,  
contrats, liens, similarité, interactions, ...

# Qu'est-ce qu'un réseau ?

## liens / relations entre des objets

"objets" ?

individus, machines, logiciels, lieux,  
entreprises, documents, animaux, ...

"liens, relations" ?

amitié, collaboration, câbles, appels, routes,  
contrats, liens, similarité, interactions, ...

ex : relations d'amitié : Alex et Bénédicte, Bénédicte et Claude,  
Claude et Daniel, Daniel et Bénédicte.

# Qu'est-ce qu'un réseau ?

## liens / relations entre des objets

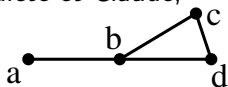
"objets" ?

individus, machines, logiciels, lieux,  
entreprises, documents, animaux, ...

"liens, relations" ?

amitié, collaboration, câbles, appels, routes,  
contrats, liens, similarité, interactions, ...

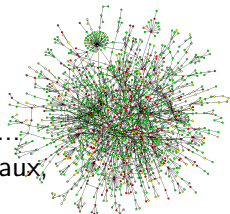
ex : relations d'amitié : Alex et Bénédicte, Bénédicte et Claude,  
Claude et Daniel, Daniel et Bénédicte.



# Quelques exemples

## réseaux d'infrastructures

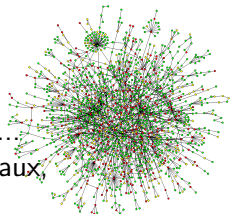
distribution d'énergie, d'eau, de données...  
matériels connectés par des câbles, des tuyaux,  
accessibilité radio, ...



# Quelques exemples

## réseaux d'infrastructures

distribution d'énergie, d'eau, de données...  
matériels connectés par des câbles, des tuyaux,  
accessibilité radio, ...



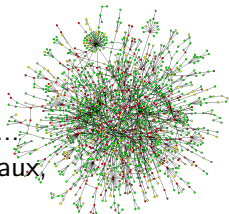
## réseaux sociaux

collaborations professionnelles, amitié, connaissance, contrats, ...  
individus ou entreprises connectés par des relations sociales

# Quelques exemples

## réseaux d'infrastructures

distribution d'énergie, d'eau, de données...  
matériels connectés par des câbles, des tuyaux,  
accessibilité radio, ...



## réseaux sociaux

collaborations professionnelles, amitié, connaissance, contrats, ...  
individus ou entreprises connectés par des relations sociales

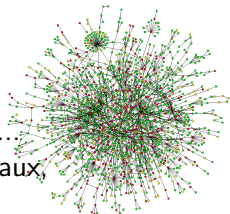
## réseaux de contenus

similarités, liens hypertextes, citations, ...  
textes, articles, pages web, films, ...

# Quelques exemples

## réseaux d'infrastructures

distribution d'énergie, d'eau, de données...  
matériels connectés par des câbles, des tuyaux,  
accessibilité radio, ...



## réseaux sociaux

collaborations professionnelles, amitié, connaissance, contrats, ...  
individus ou entreprises connectés par des relations sociales

## réseaux de contenus

similarités, liens hypertextes, citations, ...  
textes, articles, pages web, films, ...

## réseaux biologiques

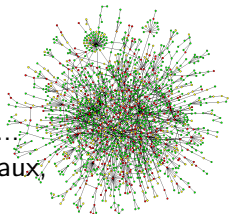
cerveau, interactions protéiques, dépendances entre gènes, ...



# Quelques exemples

## réseaux d'infrastructures

distribution d'énergie, d'eau, de données...  
matériels connectés par des câbles, des tuyaux,  
accessibilité radio, ...



## réseaux sociaux

collaborations professionnelles, amitié, connaissance, contrats, ...  
individus ou entreprises connectés par des relations sociales

## réseaux de contenus

similarités, liens hypertextes, citations, ...  
textes, articles, pages web, films, ...

## réseaux biologiques

cerveau, interactions protéiques, dépendances entre gènes, ...

**etc, etc, etc, etc, etc, etc, etc**

# L'approche réseau

il y a des réseaux  
vs  
on voit des réseaux

# L'approche réseau

**il y a des réseaux**  
**vs**  
**on voit des réseaux**

exemple : élèves de l'ESGI

↔ notes, âges, adresses, études antérieures, régions d'origine, ...

# L'approche réseau

il y a des réseaux  
vs  
on voit des réseaux

exemple : élèves de l'ESGI

↔ notes, âges, adresses, études antérieures, régions d'origine, ...

vs

qui discute avec qui, qui est ami avec qui,  
qui demande conseil à qui, qui travaille avec qui,  
qui appelle qui, qui écrit à qui, ...

# L'approche réseau

**il y a des réseaux**  
**vs**  
**on voit des réseaux**

exemple : élèves de l'ESGI

↔ notes, âges, adresses, études antérieures, régions d'origine, ...

**vs**

qui discute avec qui, qui est ami avec qui,  
qui demande conseil à qui, qui travaille avec qui,  
qui appelle qui, qui écrit à qui, ...

**une façon d'appréhender les données**

# Des réseaux partout

## points communs :

avoir un mot ou un auteur en commun, acheter le même produit, habiter au même endroit, se croiser, faire les mêmes études, ...

## usages et mobilité :

achats, clics, téléchargements, emprunts, spectacles, ... relations entre lieux, déplacements, co-présence, contacts, ...

## échanges :

appels téléphoniques, messages, bétail, transactions financières, ...

# Des réseaux partout

## points communs :

avoir un mot ou un auteur en commun, acheter le même produit, habiter au même endroit, se croiser, faire les mêmes études, ...

## usages et mobilité :

achats, clics, téléchargements, emprunts, spectacles, ... relations entre lieux, déplacements, co-présence, contacts, ...

## échanges :

appels téléphoniques, messages, bétail, transactions financières, ...

**contextes extrêmement divers**

# Des réseaux partout

## points communs :

avoir un mot ou un auteur en commun, acheter le même produit, habiter au même endroit, se croiser, faire les mêmes études, ...

## usages et mobilité :

achats, clics, téléchargements, emprunts, spectacles, ... relations entre lieux, déplacements, co-présence, contacts, ...

## échanges :

appels téléphoniques, messages, bétail, transactions financières, ...

## contextes extrêmement divers

↔ **problématiques** : infrastructures, recommandation, communication, épidémies, robustesse, anomalies, ...



# Cas pratique 1 : la Ğ1

<https://monnaie-libre.fr/>  
approx 3 ans d'existence  
approx 3 000 utilisateurs  
essentiellement francophone

une cryptomonnaie visant (notamment) à

**se réappropriier la création monétaire**

(et dissuader l'accumulation)

# Cas pratique 1 : la Ğ1

<https://monnaie-libre.fr/>  
approx 3 ans d'existence  
approx 3 000 utilisateurs  
essentiellement francophone

une cryptomonnaie visant (notamment) à

**se réappropriier la création monétaire**

(et dissuader l'accumulation)

**chaque *membre* reçoit le *dividende universel* (DU)**

# Cas pratique 1 : la Ğ1

<https://monnaie-libre.fr/>  
approx 3 ans d'existence  
approx 3 000 utilisateurs  
essentiellement francophone

une cryptomonnaie visant (notamment) à

**se réapproprier la création monétaire**

(et dissuader l'accumulation)

**chaque *membre* reçoit le *dividende universel* (DU)**

**problème : comptes multiples  $\Rightarrow$  plusieurs DU**

besoin d'authentifier les membres  
(sans autorité centrale)

**solution : *certifications***

# Cas pratique 1 : la Ğ1 – certifications

les *membres* reçoivent le DU et peuvent certifier  
on devient membre quand on a 5 certifications

# Cas pratique 1 : la Ğ1 – certifications

les *membres* reçoivent le DU et peuvent certifier

on devient membre quand on a 5 certifications

et qu'on est à distance au plus 5 de 80% des *référents*

référent = membre ayant donné 5 certifications

nb de certifications limité à 100

fréquence et durée de validité limitées, etc

**(mécanismes anti attaques)**

# Cas pratique 1 : la Ğ1 – certifications

les *membres* reçoivent le DU et peuvent certifier

on devient membre quand on a 5 certifications

et qu'on est à distance au plus 5 de 80% des *référents*

référent = membre ayant donné 5 certifications

nb de certifications limité à 100

fréquence et durée de validité limitées, etc

(mécanismes anti attaques)

réseau !!

# Cas pratique 1 : la G1 – Toile de Confiance

*Web of Trust (WoT)*

**un réseau de membres  
reliés par des certifications  
orientées**

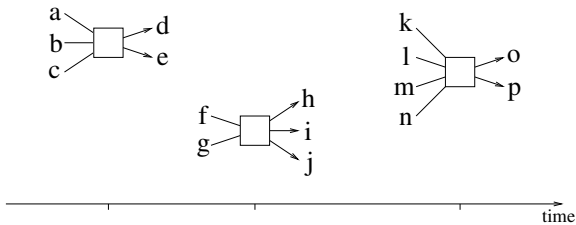
fichier wot.dat  
vidéo wot.avi

plusieurs réseaux :  
le protocole,  
les transactions,  
la toile de confiance

# Cas pratique 2 : Bitcoin

## *transactions bitcoin*

des adresses en entrée et en sortie

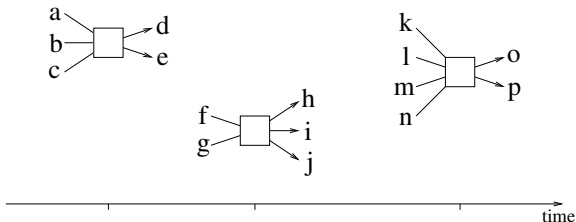




# Cas pratique 2 : Bitcoin

## *transactions bitcoin*

des adresses en entrée et en sortie



↪ **plusieurs réseaux** :  
infrastructure,  
flux monétaires entre adresses,  
qui échange avec qui, ...

# Cas pratique 3 : détection d'anomalies

anomalies ?

fraudes, usurpation d'identité

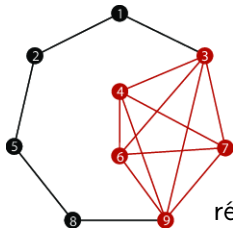
blanchiment, mixing

# Cas pratique 3 : détection d'anomalies

anomalies ?

fraudes, usurpation d'identité

blanchiment, mixing



révélées par des *motifs* (*patterns*)

dans les réseaux d'échanges

↪ **challenge : trouver des motifs anormaux**

# Réseau vs dessins

réseau  $\neq$  dessin :  
un réseau, plein de dessins

# Réseau vs dessins

réseau  $\neq$  dessin :  
un réseau, plein de dessins

La *définition* d'un réseau c'est :

un ensemble de noeuds et un ensemble de liens

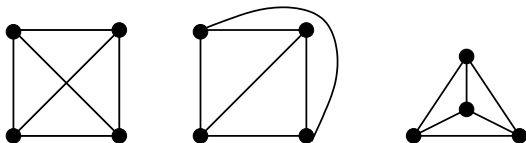
# Réseau vs dessins

réseau  $\neq$  dessin :  
un réseau, plein de dessins

La *définition* d'un réseau c'est :

un ensemble de noeuds et un ensemble de liens

Exemple :  $K_4$  : quatre noeuds tous reliés ; dessins :



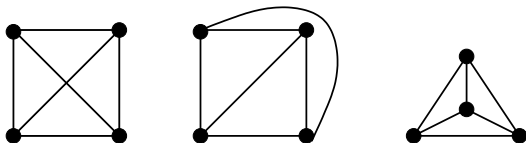
# Réseau vs dessins

réseau  $\neq$  dessin :  
un réseau, plein de dessins

La *définition* d'un réseau c'est :

un ensemble de noeuds et un ensemble de liens

Exemple :  $K_4$  : quatre noeuds tous reliés ; dessins :



pas de *meilleur* dessin

un domaine en soi

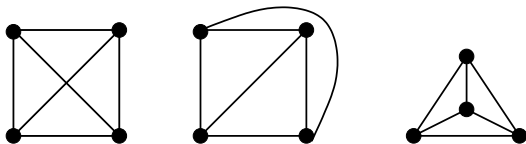
# Réseau vs dessins

réseau  $\neq$  dessin :  
un réseau, plein de dessins

La *définition* d'un réseau c'est :

un ensemble de noeuds et un ensemble de liens

Exemple :  $K_4$  : quatre noeuds tous reliés ; dessins :



pas de *meilleur* dessin

un domaine en soi

!!! attention : trompeur !!!



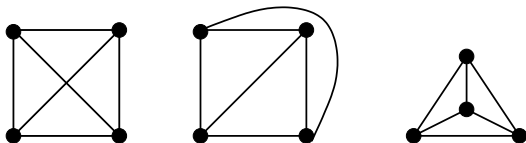
# Réseau vs dessins

réseau  $\neq$  dessin :  
un réseau, plein de dessins

La *définition* d'un réseau c'est :

un ensemble de noeuds et un ensemble de liens

Exemple :  $K_4$  : quatre noeuds tous reliés ; dessins :



pas de *meilleur* dessin

un domaine en soi

!!! attention : trompeur !!!

déf : planaire = dessinable sans croisement

pas toujours planaire

# Définition formelle

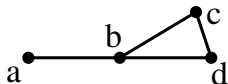
réseaux  $\longrightarrow$  graphes

graphe  $G$  défini par :  
ensemble de noeuds  $V$   
ensemble de liens  $E$

$$\hookrightarrow G = (V, E)$$

Exemple :  $V = \{a, b, c, d\}$ ,  $E = \{ab, bc, dc, db\}$

$\hookrightarrow$  un dessin :



# Définition formelle

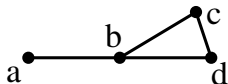
réseaux  $\rightarrow$  graphes

graphe  $G$  défini par :  
ensemble de noeuds  $V$   
ensemble de liens  $E$

$$\hookrightarrow G = (V, E)$$

Exemple :  $V = \{a, b, c, d\}$ ,  $E = \{ab, bc, dc, db\}$

$\hookrightarrow$  un dessin :



nombre de noeuds :  $n = |V|$

nombre de liens :  $m = |E|$

voisins de  $v$  :  $N(v) = \{u \text{ t.q. } uv \in E\}$

dans l'exemple :  $n = 4$ ,  $m = 4$ ,  $N(a) = \{b\}$ ,  $N(b) = \{a, c, d\}$ , ...

# Version orientée

classique (non-orienté) :

chaque lien est une paires de noeuds

paire  $ab$  définie par deux noeuds  $a$  et  $b$

non-ordonnée :  $ab = ba$

voisinage :  $N(v) = \{u \text{ t.q. } uv \in E\}$

# Version orientée

classique (non-orienté) :

chaque lien est une paires de noeuds

paire  $ab$  définie par deux noeuds  $a$  et  $b$

non-ordonnée :  $ab = ba$

voisinage :  $N(v) = \{u \text{ t.q. } uv \in E\}$

**mais ! transactions financières, certifications, ...**

# Version orientée

classique (non-orienté) :

chaque lien est une paires de noeuds

paire  $ab$  définie par deux noeuds  $a$  et  $b$

non-ordonnée :  $ab = ba$

voisinage :  $N(v) = \{u \text{ t.q. } uv \in E\}$

mais ! transactions financières, certifications, ...

↪ orienté :

chaque lien est un couple noeuds

couple  $(a, b)$  défini par deux noeuds  $a$  et  $b$

ordonné :  $(a, b) \neq (b, a)$

voisinages :

$$N^+(v) = \{u \text{ t.q. } (v, u) \in E\}, \quad N^-(v) = \{u \text{ t.q. } (u, v) \in E\}$$

# Programmation

deux fichiers : ex.dat (ci-dessous) et wot.dat

ex.dat :

A

B

C

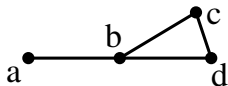
D

A B

B C

D C

D B



Programme : `cat ex.dat | python basic.py`

En live : cas orienté

# Exercice 1



# Graphes planaires

planaire = peut être dessiné sans croisement

comment savoir ?

# Graphes planaires

planaire = peut être dessiné sans croisement

comment savoir ?

trouver un dessin dans le plan

...ou pas ?

# Graphes planaires

planaire = peut être dessiné sans croisement

comment savoir ?

trouver un dessin dans le plan

...ou pas ?

un graphe planaire peut-il  
contenir un sous-graphe non planaire ?

# Graphes planaires

planaire = peut être dessiné sans croisement

comment savoir ?

trouver un dessin dans le plan

...ou pas ?

un graphe planaire peut-il  
contenir un sous-graphe non planaire ?

K5 est-il planaire ?

K33 est-il planaire ?

# Graphes planaires

planaire = peut être dessiné sans croisement

comment savoir ?

trouver un dessin dans le plan

...ou pas ?

un graphe planaire peut-il  
contenir un sous-graphe non planaire ?

K5 est-il planaire ?

K33 est-il planaire ?

conclusion ? rares ?

**réseau = objets et relations**

des réseaux partout ; vision réseaux

**modèle : graphe = noeuds + liens**

$$G = (V, E), E \subseteq V \otimes V$$

$$\text{nb noeuds } n = |V|$$

$$\text{nb liens } m = |E|$$

voisins de  $v$  :

$$N(v) = \{u \text{ t.q. } uv \in E\}$$

**réseau  $\neq$  dessin**

version orientée :

$$E \subseteq V \times V, N^+(v), N^-(v)$$

# Plan

## Faire parler un réseau

Objectifs

Densité

Degrés

Distributions

Exercice

Comparaison à l'aléatoire

Bilan

# Faire parler un réseau

à quoi ressemble un (grand) réseau donné ?  
comment en tirer de l'information pertinente ?

description (ex : clique),  
propriétés du réseau (ex : planaire, robuste),  
recherche d'éléments intéressants,  
assurer des propriétés, ...

## besoin d'un vocabulaire

→ théorie des graphes / science des réseaux  
/ analyse de réseaux : même base

déjà vu :  $n$ ,  $m$ , planarité



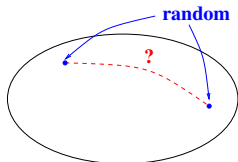
# Propriétés élémentaires : densité

On considère un graphe  $G = (V, E)$ . On connaît :

- ▶  $n = |V|$  le nombre de noeuds
- ▶  $m = |E|$  le nombre de liens

On définit la densité :

probabilité que deux noeuds  
pris au hasard soient reliés



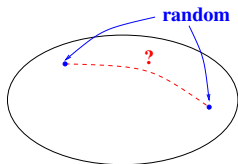
# Propriétés élémentaires : densité

On considère un graphe  $G = (V, E)$ . On connaît :

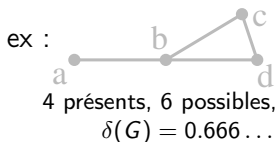
- ▶  $n = |V|$  le nombre de noeuds
- ▶  $m = |E|$  le nombre de liens

On définit la densité :

probabilité que deux noeuds  
pris au hasard soient reliés



$$\delta(G) = \frac{\text{nb liens présents}}{\text{nb liens possibles}} = \frac{2 \cdot m}{n \cdot (n-1)}$$



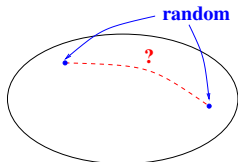
# Propriétés élémentaires : densité

On considère un graphe  $G = (V, E)$ . On connaît :

- ▶  $n = |V|$  le nombre de noeuds
- ▶  $m = |E|$  le nombre de liens

On définit la densité :

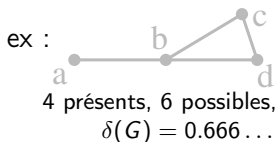
probabilité que deux noeuds pris au hasard soient reliés



$$\delta(G) = \frac{\text{nb liens présents}}{\text{nb liens possibles}} = \frac{2 \cdot m}{n \cdot (n-1)}$$

orienté :

$$\delta(G) = \frac{\text{nb liens présents}}{\text{nb liens possibles}} = \frac{m}{n \cdot (n-1)}$$



# Propriétés élémentaires : degrés

On connaît :  $N(v) = \{u \text{ t.q. } uv \in E\}$  l'ensemble de voisins de  $v$ .

On définit :

- ▶ le **degré** de  $v$ , i.e. son nombre de voisin  $d(v) = |N(v)|$
- ▶ le **degré moyen**  $d(G) = \frac{1}{n} \sum_v d(v) = \frac{2 \cdot m}{n}$
- ▶ le **degré maximum**  $d_{\max}(G) = \max_v (d(v))$



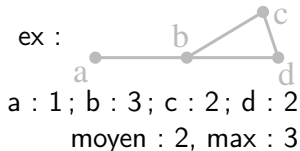
a : 1 ; b : 3 ; c : 2 ; d : 2  
moyen : 2, max : 3

# Propriétés élémentaires : degrés

On connaît :  $N(v) = \{u \text{ t.q. } uv \in E\}$  l'ensemble de voisins de  $v$ .

On définit :

- ▶ le **degré** de  $v$ , i.e. son nombre de voisin  $d(v) = |N(v)|$
- ▶ le **degré moyen**  $d(G) = \frac{1}{n} \sum_v d(v) = \frac{2 \cdot m}{n}$
- ▶ le **degré maximum**  $d_{\max}(G) = \max_v (d(v))$



relation avec la densité :  $d(G) = \delta(G) \cdot (n - 1)$

orienté :

degré entrant  $d^-(v) = |N^-(v)|$ , degré sortant  $d^+(v) = |N^+(v)|$

# Cas pratique : WoT de la Ğ1

Programme : `cat wot.dat | python degree.py`

Observation : degré moyen, min, max, densité, ...

Observation : degré de chaque noeud

`plot "degree.out" u 2 pt 7 notitle`

En live : cas orienté

Observation ?

# Cas pratique : WoT de la Ğ1

Programme : `cat wot.dat | python degree.py`

Observation : degré moyen, min, max, densité, ...

Observation : degré de chaque noeud

`plot "degree.out" u 2 pt 7 notitle`

En live : cas orienté

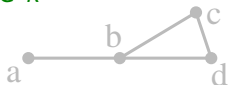
Observation ?

**mais propriétés agrégées...**

# Distributions

distribution des degrés :

**pour tout entier  $k$ ,  
nombre de noeuds ayant degré  $k$**



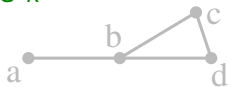
ex : 1 : 1 ; 2 : 2 ; 3 : 1



# Distributions

distribution des degrés :

**pour tout entier  $k$ ,  
nombre de noeuds ayant degré  $k$**



ex : 1 : 1 ; 2 : 2 ; 3 : 1

homogènes (ex : tailles)

vs

hétérogène (ex : revenus)

**des conséquences importantes**

corrélations entre degrés

en orienté : distributions de degrés entrants, sortants, corrélations

# Cas pratique : WoT de la Ğ1

Programme : `cat wot.dat | python degree.py`

Observation : distribution de degrés ; échelles log ; en fraction

```
set logscale xy
```

```
plot "degree.out" u 2 pt 7 notitle
```

# Cas pratique : WoT de la Ğ1

Programme : `cat wot.dat | python degree.py`

Observation : distribution de degrés ; échelles log ; en fraction

```
set logscale xy
```

```
plot "degree.out" u 2 pt 7 notitle
```

En live : cas orienté ; deux distributions ; corrélations entre degrés entrants et sortants

# Cas pratique : WoT de la Ğ1

Programme : `cat wot.dat | python degree.py`

Observation : distribution de degrés ; échelles log ; en fraction

```
set logscale xy
```

```
plot "degree.out" u 2 pt 7 notitle
```

En live : cas orienté ; deux distributions ; corrélations entre degrés entrants et sortants

**surprenant ? information ?**

# Exercice 2

# Graphes aléatoires

principe : on fait 1000 tirages à pile ou face

- ▶ 498 piles et 502 faces ; intéressant ?
- ▶ 980 piles et 20 faces ; intéressant ?

↔ comparaison à l'attendu

# Graphes aléatoires

principe : on fait 1000 tirages à pile ou face

- ▶ 498 piles et 502 faces ; intéressant ?
- ▶ 980 piles et 20 faces ; intéressant ?

↔ comparaison à l'attendu

**et pour un réseau ? quel attendu ?**

↔ **comparaison aux graphes aléatoires**

définition (ER) :

tiré uniformément parmi les graphes de  $n$  noeuds et  $m$  liens

algorithme :

paires de noeuds aléatoires une à une, jusqu'à  $m$  distinctes

# Graphes aléatoires

principe : on fait 1000 tirages à pile ou face

- ▶ 498 piles et 502 faces ; intéressant ?
- ▶ 980 piles et 20 faces ; intéressant ?

↔ comparaison à l'attendu

**et pour un réseau ? quel attendu ?**

↔ **comparaison aux graphes aléatoires**

définition (ER) :

tiré uniformément parmi les graphes de  $n$  noeuds et  $m$  liens

algorithme :

paires de noeuds aléatoires une à une, jusqu'à  $m$  distinctes

même  $n$ , même  $m$

donc même densité, même degré moyen

**même distribution de degrés ?**



# Mise en pratique

Programme : `python make_er.py 3530 23917 > wot.er`

Observation : `plot "wot.deg_d" pt 7 ps 3 notitle,`  
`"wot_er.deg_d" pt 5 ps 3 notitle`

En live : version orientée ?

tirage, distributions de degrés, corrélations, comparaison

# Mise en pratique

Programme : `python make_er.py 3530 23917 > wot.er`

Observation : `plot "wot.deg_d" pt 7 ps 3 notitle,`  
`"wot_er.deg_d" pt 5 ps 3 notitle`

En live : version orientée ?

tirage, distributions de degrés, corrélations, comparaison

**très différent d'un graphe aléatoire comparable**

processus sous-jacent ?

noeuds importants ?

on a tout capturé ?

**conséquences ?**

## un vocabulaire

- $n$ ,  $m$ , orienté, voisins,
- ↪ planarité, clique, densité, degrés, distributions, ...
- ↪ **pouvoir descriptif**

## un vocabulaire

- $n$ ,  $m$ , orienté, voisins,
- ↔ planarité, clique, densité, degrés, distributions, ...
- ↔ **pouvoir descriptif**

tous les noeuds n'ont pas le même rôle

## un vocabulaire

- $n$ ,  $m$ , orienté, voisins,
- ↪ planarité, clique, densité, degrés, distributions, ...
- ↪ **pouvoir descriptif**

tous les noeuds n'ont pas le même rôle

## une méthodologie

comparaison à l'aléatoire

- ↪ **observations significatives**

## un vocabulaire

$n$ ,  $m$ , orienté, voisins,  
↪ planarité, clique, densité,  
degrés, distributions, ...  
↪ **pouvoir descriptif**

tous les noeuds n'ont pas le même rôle

## une méthodologie

comparaison à l'aléatoire

↪ **observations significatives**

– suite –

impact des observations

propriétés plus avancées

application à Bitcoin

# Plan

## Propriétés avancées

Chemins et accessibilité

Exercice

Composantes connexes

Application à Bitcoin

Examen

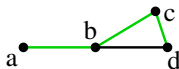
# Chemins

## Définitions :

dans  $G = (V, E)$ , **chemin** de  $u$  à  $v$  :  
suite de liens  $v_0 v_1, v_1 v_2, v_2 v_3, \dots, v_{k-1} v_k$   
avec  $v_0 = u$  et  $v_k = v$

$k$  est la **longueur** du chemin

## Exemple :



un chemin de  $a$  à  $d$  :  $ab, bc, cd$  ; longueur 3



# Chemins

## Définitions :

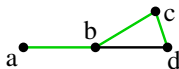
dans  $G = (V, E)$ , **chemin** de  $u$  à  $v$  :  
suite de liens  $v_0 v_1, v_1 v_2, v_2 v_3, \dots, v_{k-1} v_k$   
avec  $v_0 = u$  et  $v_k = v$

$k$  est la **longueur** du chemin

longueur minimale de  $u$  à  $v$  = **distance** de  $u$  à  $v$

distance maximale dans  $G$  = son **diamètre**

## Exemple :

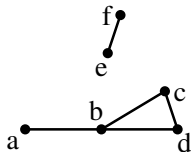


un chemin de  $a$  à  $d$  :  $ab, bc, cd$  ; longueur 3

distance de  $a$  à  $d$  ? 2

diamètre ? 2

# Accessibilité, connexité

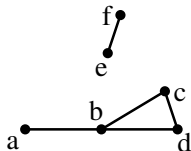


# Accessibilité, connexité

existence d'un chemin de  $u$  à  $v$  ? pas toujours !

↪ **accessibilité**

$$u \longrightarrow v$$



si non-orienté : accessibilité symétrique

$$u \longrightarrow v \Rightarrow v \longrightarrow u$$

↪ **composantes connexes**

*notion de "morceau"*

$X$  maximal tel que  $u \longrightarrow v$  pour tous  $u$  et  $v$  dans  $X$

↪ plus grande composante

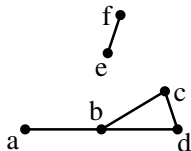
↪ graphe connexe

# Accessibilité, connexité

existence d'un chemin de  $u$  à  $v$  ? pas toujours !

↔ **accessibilité**

$$u \longrightarrow v$$



si non-orienté : accessibilité symétrique

$$u \longrightarrow v \Rightarrow v \longrightarrow u$$

↔ **composantes connexes**

*notion de "morceau"*

$X$  maximal tel que  $u \longrightarrow v$  pour tous  $u$  et  $v$  dans  $X$

↔ plus grande composante

↔ graphe connexe

orienté : *plus compliqué*

↔ plus grand nombre de noeuds accessibles

# Exercice 3

# Composantes connexes : calcul

noeuds accessibles à partir de  $v$  ?

Algorithme :

- ▶  $v$  est accessible
- ▶ les voisins des accessibles sont accessibles

distances ?

plus grand nombre de noeuds accessibles ?

si non-orienté, c'est la taille de la composante maximale

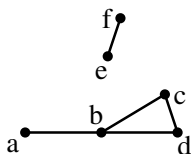
# Composantes connexes : calcul

seen:

$$V = \{a, b, c, d, e, f\}, \quad E = \{ab, bc, bd, cd, ef\}$$

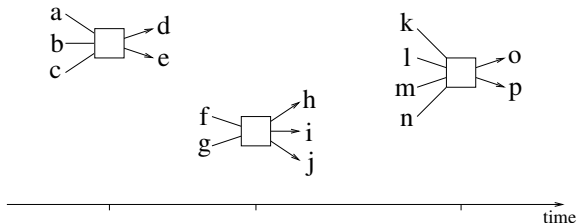
$$N(a) = \{b\}, \quad N(b) = \{a, c, d\}, \quad N(c) = \{b, d\},$$

$$N(d) = \{b, c\}, \quad N(e) = \{f\}, \quad N(f) = \{e\}$$



# Application à Bitcoin

*transactions bitcoin*  
des adresses en entrée et en sortie

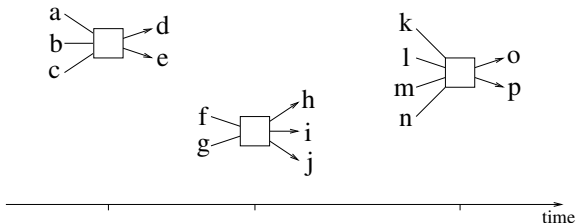




# Application à Bitcoin

## *transactions bitcoin*

des adresses en entrée et en sortie



utilisateurs ? plusieurs adresses ; lesquelles ?

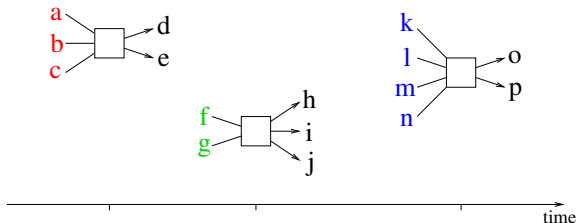
↪ **anti-aliasing d'adresses**

H0 : réutilisation d'adresses (par ex,  $k = a$ )

# Application à Bitcoin

## *transactions bitcoin*

des adresses en entrée et en sortie



utilisateurs ? plusieurs adresses ; lesquelles ?

↪ **anti-aliasing d'adresses**

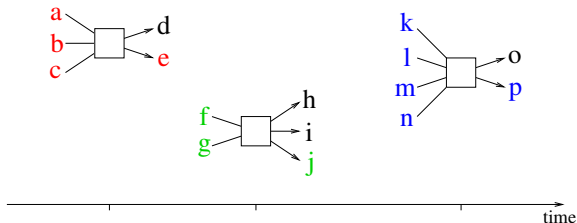
H0 : réutilisation d'adresses (par ex,  $k = a$ )

H1 : adresses en entrée, même utilisateur

# Application à Bitcoin

## *transactions bitcoin*

des adresses en entrée et en sortie



utilisateurs ? plusieurs adresses ; lesquelles ?

↪ **anti-aliasing d'adresses**

H0 : réutilisation d'adresses (par ex,  $k = a$ )

H1 : adresses en entrée, même utilisateur

H2 : une adresse en sortie pour le *change*

# Anti-aliasing Bitcoin

comment trouver toutes les adresses  
associées au même utilisateur ?

# Anti-aliasing Bitcoin

**comment trouver toutes les adresses  
associées au même utilisateur ?**

un graphe entre adresses  
liens H1 : adresses en entrée  
liens H2 : entrées et change

# Anti-aliasing Bitcoin

**comment trouver toutes les adresses  
associées au même utilisateur ?**

un graphe entre adresses  
liens H1 : adresses en entrée  
liens H2 : entrées et change

**un utilisateur  
=  
une composante connexe**

# Examen 1

# Plan

## Robustesse

Modèle

En pratique

Comparaison à l'aléatoire

Exercice

Interprétation

Conclusion



# Rappel

on a vu...

– distribution de degrés –  
pour tout  $k$ , nb de noeuds ayant degré  $k$

– WoT de  $\check{G}_1$  –

distribution hétérogène

– graphe aléatoire comparable –  
distribution homogène

⇒ très différent

conséquences ?

# Robustesse

"robuste" = ?

... "pas solide" ... "facile à casser en morceaux" ?

# Robustesse

"robuste" = ?

... "pas solide" ... "facile à casser en morceaux" ?

réseaux d'infrastructures vs réseaux épidémiologiques

pannes/attaques vs vaccins

# Robustesse

"robuste" = ?

... "pas solide" ... "facile à casser en morceaux" ?

réseaux d'infrastructures vs réseaux épidémiologiques

pannes/attaques vs vaccins

résistance aux suppression de noeuds

taille du plus gros "morceau"

en fonction du nombre de noeuds supprimés

notion de "morceaux" ?

# Modèle de la robustesse

taille de la **plus grande composante connexe**  
(ceux qui peuvent communiquer)

en fonction du **nombre de noeuds supprimés**

# Modèle de la robustesse

taille de la **plus grande composante connexe**  
(ceux qui peuvent communiquer)

en fonction du **nombre de noeuds supprimés**

au **hasard** (pannes) vs avec une **stratégie** (attaque)

stratégie d'attaque : forts degrés d'abord

infrastructures vs épidémiologie

# Robustesse en pratique

Code : `robust.py`

fonction `component`  
puis `get_max_reachable`  
puis version optimisée

# Robustesse en pratique

Code : robust.py

fonction component  
puis get\_max\_reachable  
puis version optimisée

suppression de noeuds aléatoires vs stratégie

```
python robust.py random < wot.dat > wot.random  
plot "wot.random" pt 7 ps 3 title "random", 3530-x lw  
5 title "n"
```

```
python robust.py attack < wot.dat > wot.attack  
plot "wot.attack" pt 5 ps 3 title "attack", 3530-x lw  
5 title "n"
```



# Robustesse en pratique

Code : robust.py

fonction component  
puis get\_max\_reachable  
puis version optimisée

suppression de noeuds aléatoires vs stratégie

```
python robust.py random < wot.dat > wot.random  
plot "wot.random" pt 7 ps 3 title "random", 3530-x lw  
5 title "n"
```

```
python robust.py attack < wot.dat > wot.attack  
plot "wot.attack" pt 5 ps 3 title "attack", 3530-x lw  
5 title "n"
```

conclusion ?

résistant aux pannes, sensible aux attaques

significatif ?

comparaison à l'aléatoire

# Comparaison à l'aléatoire

même expérience sur graphe aléatoire, mêmes  $n$  et  $m$

- ▶ `python make_er.py 3530 23917 > wot.er`
- ▶ `python robust.py random < wot.er > wot.er.random`
- ▶ `python robust.py attack < wot.er > wot.er.attack`
- ▶ `plot "wot.random" pt 7 ps 3 title "wot",  
"wot.er.random" pt 5 ps 2 title "er"`
- ▶ `plot "wot.attack" pt 7 ps 3 title "wot",  
"wot.er.attack" pt 5 ps 2 title "er"`

# Comparaison à l'aléatoire

même expérience sur graphe aléatoire, mêmes  $n$  et  $m$

- ▶ `python make_er.py 3530 23917 > wot.er`
- ▶ `python robust.py random < wot.er > wot.er.random`
- ▶ `python robust.py attack < wot.er > wot.er.attack`
- ▶ `plot "wot.random" pt 7 ps 3 title "wot",  
"wot.er.random" pt 5 ps 2 title "er"`
- ▶ `plot "wot.attack" pt 7 ps 3 title "wot",  
"wot.er.attack" pt 5 ps 2 title "er"`

**robustesse aux pannes : non significatif**  
**sensibilité aux attaques : significatif !**

# Comparaison à l'aléatoire

même expérience sur graphe aléatoire, mêmes  $n$  et  $m$

- ▶ `python make_er.py 3530 23917 > wot.er`
- ▶ `python robust.py random < wot.er > wot.er.random`
- ▶ `python robust.py attack < wot.er > wot.er.attack`
- ▶ `plot "wot.random" pt 7 ps 3 title "wot",  
"wot.er.random" pt 5 ps 2 title "er"`
- ▶ `plot "wot.attack" pt 7 ps 3 title "wot",  
"wot.er.attack" pt 5 ps 2 title "er"`

**robustesse aux pannes : non significatif**  
**sensibilité aux attaques : significatif !**

**dû aux degrés ?**

# Exercice 4

# Un meilleur aléatoire

ER : même  $n$ , même  $m$ , mais c'est tout

↪ change profondément les degrés

**conserver la distribution de degrés ?**

# Un meilleur aléatoire

ER : même  $n$ , même  $m$ , mais c'est tout

↪ change profondément les degrés

**conserver la distribution de degrés ?**

*swap* de liens :

$uv$  et  $xy$  deviennent  $ux$  et  $vy$

effet sur les degrés ?

# Un meilleur aléatoire

ER : même  $n$ , même  $m$ , mais c'est tout

↪ change profondément les degrés

conserver la distribution de degrés ?

*swap* de liens :

$uv$  et  $xy$  deviennent  $ux$  et  $vy$

effet sur les degrés ?

"beaucoup" de *swaps*

↪ **graphe tiré uniformément**  
**même  $n$ , même  $m$ ,**  
**même degrés**

graphe *typique*



# Robustesse vs degrés

robustesse sur les graphes mélangés

```
for i in 3000 30000 300000
do
python shuffle.py "$i" < wot.dat > wot."$i".dat
python robust.py random < wot."$i".dat > wot."$i".random
python robust.py attack < wot."$i".dat > wot."$i".attack
done
```

*temps de calcul*

```
plot "wot.random" pt 5 ps 3 title "original", "wot.3000.random" pt 7 ps 2 title "3000",
"wot.30000.random" pt 5 ps 1 title "30 000", "wot.300000.random" pt 3 ps 2 title "300 000"

plot "wot.attack" pt 7 ps 3 title "original", "wot.3000.attack" pt 7 ps 2 title "3000",
"wot.30000.attack" pt 7 ps 2 title "30 000", "wot.300000.attack" pt 7 ps 2 title "300 000"
```

# Conclusion sur la robustesse

## On modélise la question :

combien de noeuds restent accessibles  
quand on supprime des noeuds  
au hasard ou avec stratégie

## On observe :

tous sont résistants aux suppressions aléatoires

bonne nouvelle

mauvaise nouvelle

réel : très sensible aux attaques

totalément aléatoire (ER) : insensible

aléatoire avec même degrés : sensible

↔ dû aux degrés

vidéo : robustesse

# Plan

## Aller plus loin

Localité des liens

Exercice

Processus sous-jacents

Plus que des graphes

# Triangles

les amis de mes amis sont mes amis  
mes amis sont amis entre eux  
localité géographique des liens

↔ **localité spatiale ou sociale**

↔ **triangles dans le graphe**

**triangle : ensemble de trois noeuds avec trois liens**

# Triangles

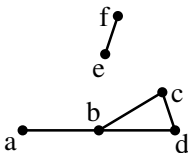
les amis de mes amis sont mes amis  
mes amis sont amis entre eux  
localité géographique des liens

↔ **localité spatiale ou sociale**

↔ **triangles dans le graphe**

**triangle : ensemble de trois noeuds avec trois liens**

Exemple :



un seul triangle :  $\{B, C, D\}$

# Triangles

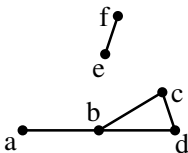
les amis de mes amis sont mes amis  
mes amis sont amis entre eux  
localité géographique des liens

↔ **localité spatiale ou sociale**

↔ **triangles dans le graphe**

**triangle : ensemble de trois noeuds avec trois liens**

Exemple :



un seul triangle :  $\{B, C, D\}$

↔ **mesure de la localité : nombre de triangles**

# Triangles : calcul

plusieurs algorithmes :

- ▶ centré noeud : pour chaque noeud, le nombre de liens entre ses voisins
- ▶ centré lien : pour chaque lien, le nombre de voisins communs à ses extrémités

# Triangles : calcul

plusieurs algorithmes :

- ▶ centré noeud : pour chaque noeud, le nombre de liens entre ses voisins
- ▶ centré lien : pour chaque lien, le nombre de voisins communs à ses extrémités

**implémentation, observation  
comparaison à l'aléatoire**

dans la WoT : approx 43 000 triangles

dans sa version ER : approx 400

dans sa version mélangée : approx 800

↪ **grand nombre de triangles  
100 fois plus que l'aléatoire**



# Triangles : conclusion

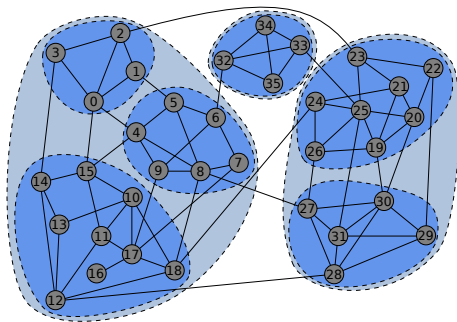
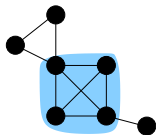
significatif, confirme la localité  
bien distinct de la distribution de degrés

# Triangles : conclusion

significatif, confirme la localité

bien distinct de la distribution de degrés

conséquences : cliques, communautés, sous-communautés, ...



lien avec la planarité ?

# Utilisation pour le mélange

"beaucoup" de swaps

# Utilisation pour le mélange

"beaucoup" de swaps

combien ?

beaucoup !

↔ assez ?

# Utilisation pour le mélange

"beaucoup" de swaps

combien ?

beaucoup !

↔ assez ?

quand les propriétés du graphe ne changent plus

courbe : nombre de triangles en fonction du nombre de swaps

# Utilisation pour le mélange

"beaucoup" de swaps

combien ?

beaucoup !

↔ assez ?

quand les propriétés du graphe ne changent plus

courbe : nombre de triangles en fonction du nombre de swaps

mise en pratique

```
python shuffle_triangles.py 100000 < wot.dat > wot.sh_tr  
plot [:] [0:] "wot.sh_tr" pt 7
```

# Clustering et Jaccard

nombre de triangles : global dans le graphe

pour chaque noeud : nombre de triangles le contenant

= nombre de liens entre ses voisins

# Clustering et Jaccard

nombre de triangles : global dans le graphe

pour chaque noeud : nombre de triangles le contenant

= nombre de liens entre ses voisins

↔ **coefficient de clustering**

$$\leftrightarrow cc(v) = \frac{\text{nb triangles}}{\text{nb possibles}}$$

↔  $cc(v) =$  **densité du voisinage**



# Clustering et Jaccard

nombre de triangles : global dans le graphe

pour chaque noeud : nombre de triangles le contenant

= nombre de liens entre ses voisins

↔ **coefficient de clustering**

$$\leftrightarrow cc(v) = \frac{\text{nb triangles}}{\text{nb possibles}}$$

↔  $cc(v) =$  **densité du voisinage**

pour chaque lien : nombre de triangles le contenant

= nombre de voisins communs des extrémités

↔ **coefficient de Jaccard**

$$\leftrightarrow cc(v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$$

# Clustering et Jaccard

nombre de triangles : global dans le graphe

pour chaque noeud : nombre de triangles le contenant

= nombre de liens entre ses voisins

↔ **coefficient de clustering**

$$\hookrightarrow cc(v) = \frac{\text{nb triangles}}{\text{nb possibles}}$$

↔  $cc(v) =$  **densité du voisinage**

pour chaque lien : nombre de triangles le contenant

= nombre de voisins communs des extrémités

↔ **coefficient de Jaccard**

$$\hookrightarrow cc(v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$$

remarques : entre 0 et 1, pas toujours définis

# Clustering et Jaccard : en pratique

en fonction du degré  
en pratique : clustering.py  
amélioration et ajout de Jaccard

# Comparaison à l'aléatoire

corrélation degrés - nombre de triangles  
**significatif ?**

graphe aléatoire  
à nombre de triangles fixé  
(et distribution de degré)

mêmes observations ?

# Comparaison à l'aléatoire

corrélation degrés - nombre de triangles

**significatif ?**

graphe aléatoire

à nombre de triangles fixé

(et distribution de degré)

mêmes observations ?

**on ne sait pas faire**

# Exercice 5

# Des observations aux causes

On a vu :

- ▶ des observation, des descriptions
- ▶ leur évaluation par la comparaison à l'aléatoire
- ▶ leurs conséquences (robustesse)

quelles causes ?

processus sous-jacents ?

principes ?

(gravité, pomme)

# Des observations aux causes

On a vu :

- ▶ des observation, des descriptions
- ▶ leur évaluation par la comparaison à l'aléatoire
- ▶ leurs conséquences (robustesse)

quelles causes ?

processus sous-jacents ?

principes ?  
(gravité, pomme)

ici : émergence des degrés hétérogènes



# Attachement préférentiel

– principe –

noeuds arrivent un à un  
se relie à un noeud présent  
avec proba proportionnelle à son degré

# Attachement préférentiel

– principe –

noeuds arrivent un à un  
se relie à un noeud présent  
avec proba proportionnelle à son degré

attention : un seul lien  $\Rightarrow$  arbre

$\hookrightarrow k$  liens à chaque étape

$\hookrightarrow$  démarrage ?

$\hookrightarrow$  degré  $\geq k$

# Attachement préférentiel

– principe –

noeuds arrivent un à un  
se relie à un noeud présent  
avec proba proportionnelle à son degré

attention : un seul lien  $\Rightarrow$  arbre

$\hookrightarrow k$  liens à chaque étape

$\hookrightarrow$  démarrage ?

$\hookrightarrow$  degré  $\geq k$

attention : "de l'aléatoire dedans" mais pas *aléatoire*

# Attachement préférentiel

– principe –

noeuds arrivent un à un  
se relie à un noeud présent  
avec proba proportionnelle à son degré

attention : un seul lien  $\Rightarrow$  arbre

$\hookrightarrow k$  liens à chaque étape

$\hookrightarrow$  démarrage ?

$\hookrightarrow$  degré  $\geq k$

attention : "de l'aléatoire dedans" mais pas *aléatoire*

vidéo avec  $k = 2$

# Attachement préférentiel : en pratique

code :

```
python ba.py 10000 2 > ba.test  
cat ba.test | python degree.py  
plot "deg_d" pt 7 + wot + wot.er  
observation, distribution de degrés  
triangles, robustesse
```

# Attachement préférentiel : en pratique

code :

```
python ba.py 10000 2 > ba.test  
cat ba.test | python degree.py  
plot "deg_d" pt 7 + wot + wot.er  
observation, distribution de degrés  
triangles, robustesse
```

principe explicatif  
pour les degrés

# Attachement préférentiel : en pratique

code :

```
python ba.py 10000 2 > ba.test  
cat ba.test | python degree.py  
plot "deg_d" pt 7 + wot + wot.er  
observation, distribution de degrés  
triangles, robustesse
```

principe explicatif  
pour les degrés  
et les triangles ?

# Attachement préférentiel avec triangles

– principe –

noeuds arrivent un à un  
choisissent un lien au hasard  
se relie à ses deux extrémités



# Attachement préférentiel avec triangles

– principe –

noeuds arrivent un à un  
choisissent un lien au hasard  
se relie à ses deux extrémités

attachement préférentiel ?!

# Attachement préférentiel avec triangles

– principe –

noeuds arrivent un à un  
choisissent un lien au hasard  
se relie à ses deux extrémités

attachement préférentiel ?!

en pratique : coder en live

# Attachement préférentiel avec triangles

– principe –

noeuds arrivent un à un  
choisissent un lien au hasard  
se relie à ses deux extrémités

attachement préférentiel ?!

en pratique : coder en live

problèmes :

nb de liens, de triangles

pas aléatoire

# Attachement préférentiel avec triangles

– principe –

noeuds arrivent un à un  
choisissent un lien au hasard  
se relie à ses deux extrémités

attachement préférentiel ? !

en pratique : coder en live

problèmes :

nb de liens, de triangles

pas aléatoire

**planaire !**

**planaire ? !**

# Bilan sur l'attachement préférentiel

**un principe extrêmement simple**

*(rich get richer)*

(effet Matthieu : "Car on donnera à celui qui a, et il sera dans l'abondance, mais à celui qui n'a pas on ôtera même ce qu'il a.")

**suffisant pour expliquer l'observation**

(degrés très hétérogènes)

*interprétation : s'il y a des raisons de penser qu'une forme d'attachement préférentiel est à l'œuvre, alors les degrés hétérogènes ne sont pas surprenants.*

# Bilan sur l'attachement préférentiel

un principe extrêmement simple

(*rich get richer*)

(effet Matthieu : "Car on donnera à celui qui a, et il sera dans l'abondance, mais à celui qui n'a pas on ôtera même ce qu'il a.")

suffisant pour expliquer l'observation

(degrés très hétérogènes)

interprétation : *s'il y a des raisons de penser qu'une forme d'attachement préférentiel est à l'œuvre, alors les degrés hétérogènes ne sont pas surprenants.*

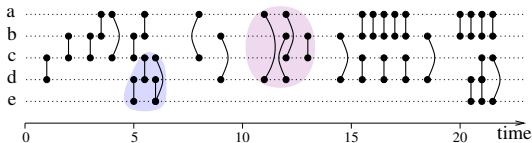
extension aux triangles

d'autres extensions (copie)

**attention aux propriétés cachées**

# Plus qu'orientés vs non-orientés

- ▶ valués → montants, importance, etc
- ▶ étiquetés, multiplexes, ... → types de noeuds/liens
- ▶ bipartis, projetés : adresses bitcoin, achats, etc
- ▶ dynamiques → évolution d'un graphe, séquences de shapshots, etc
- ▶ **flots de liens** : transactions financières ou autres, certifications, appels, etc → interactions, événements



↔ détection d'anomalies,  
de fraudes, de blanchiment

# Plan

## Encore plus loin

Diffusion

Exercice

Centralités

Stream graphs



# Objectifs

On a fait :

- ▶ observations
- ▶ validation
- ▶ conséquences
- ▶ causes

# Objectifs

On a fait :

- ▶ observations
- ▶ validation
- ▶ conséquences
- ▶ causes

**passer à l'utilisation**

# Objectifs

On a fait :

- ▶ observations
- ▶ validation
- ▶ conséquences
- ▶ causes

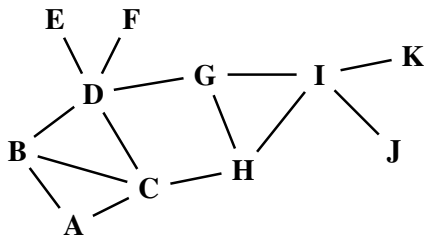
**passer à l'utilisation**

Diffusion

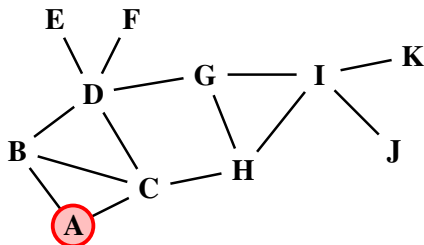
Centralités

Stream graphs

# Diffusion dans un réseau

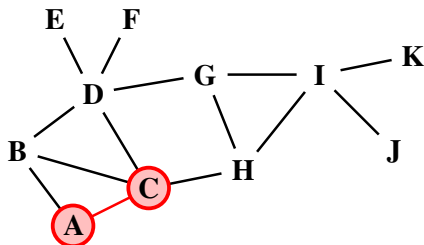


# Diffusion dans un réseau



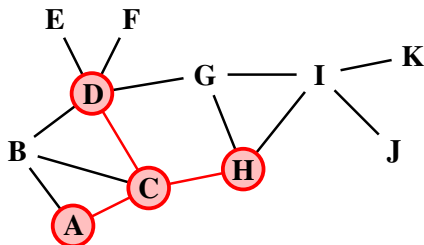
- ▶ time 1 : A introduces the *virus* or *information*

# Diffusion dans un réseau



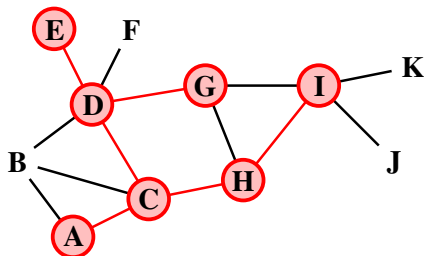
- ▶ time 1 : A introduces the *virus* or *information*
- ▶ time 2 : A spreads to C

# Diffusion dans un réseau



- ▶ time 1 : A introduces the *virus* or *information*
- ▶ time 2 : A spreads to C
- ▶ time 3 : C spreads to D and H

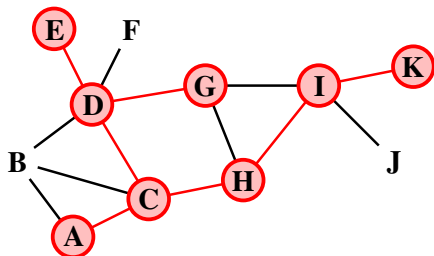
# Diffusion dans un réseau



- ▶ time 1 : A introduces the *virus* or *information*
- ▶ time 2 : A spreads to C
- ▶ time 3 : C spreads to D and H
- ▶ time 4 : D spreads to E and G, and H spreads to I



# Diffusion dans un réseau



- ▶ time 1 : A introduces the *virus* or *information*
- ▶ time 2 : A spreads to C
- ▶ time 3 : C spreads to D and H
- ▶ time 4 : D spreads to E and G, and H spreads to I
- ▶ time 5 : I spreads to K

# Diffusion : problématique

- ▶ en communication, diffusion d'information, de comportement : influence sociale (ex : emploi, innovation)
- ▶ en informatique, diffusion de données : conception de protocoles, optimisation (ex : broadcast, recherche)

# Diffusion : problématique

- ▶ en communication, diffusion d'information, de comportement : influence sociale (ex : emploi, innovation)
- ▶ en informatique, diffusion de données : conception de protocoles, optimisation (ex : broadcast, recherche)

## comment optimiser la diffusion ?

points de départs ?

à qui diffuser ?

critères de succès ?

de coût ?

# Diffusion : problématique

- ▶ en communication, diffusion d'information, de comportement : influence sociale (ex : emploi, innovation)
- ▶ en informatique, diffusion de données : conception de protocoles, optimisation (ex : broadcast, recherche)

## comment optimiser la diffusion ?

points de départs ?

à qui diffuser ?

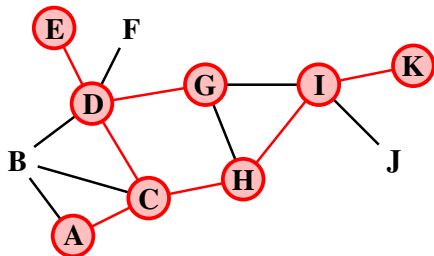
critères de succès ?

de coût ?

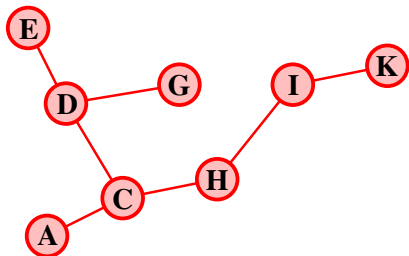
question préliminaire : **diffusions réelles ? ?**

nombre d'atteints au cours du temps  
par où ? réseau sous-jacent ?

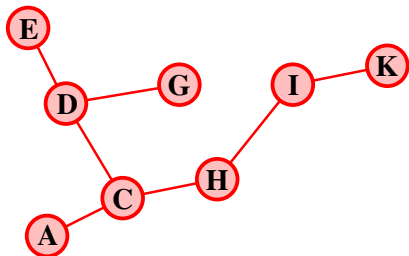
# Arbre de diffusion



# Arbre de diffusion



# Arbre de diffusion



comment en observer un ?

# The *Happy Flu* experiment

*web applet*

participate = copy the applet

target = blogs



# The *Happy Flu* experiment

*web applet*

participate = copy the applet

target = blogs

*incentives :*

fun ego-centered interface

scientific motivation

# The *Happy Flu* experiment

*web applet*

participate = copy the applet  
target = blogs

*incentives :*

fun ego-centered interface  
scientific motivation

*collected data :*

spreading links  
time information  
number of viewers (popularity)

...

# The *Happy Flu* experiment

*web applet*

participate = copy the applet  
target = blogs

*incentives :*

fun ego-centered interface  
scientific motivation

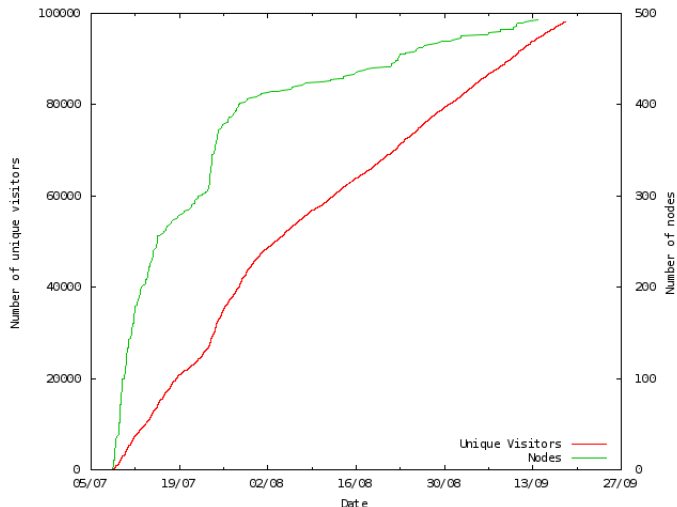
*collected data :*

spreading links  
time information  
number of viewers (popularity)

...

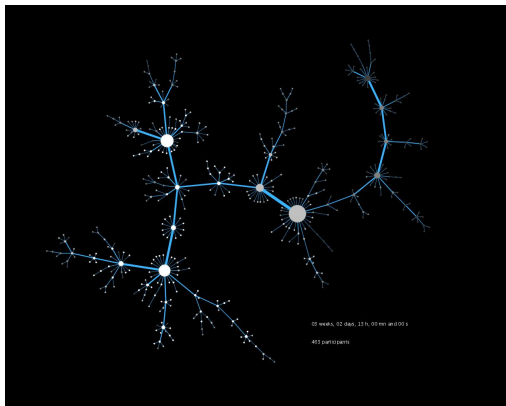
**2 months, 500 participants, 100 000 viewers**  
**spreading tree + *degree* of nodes**

# Happy Flu : spreading



number of **participants** and **viewers** during time

# Happy Flu : visualisation



nodes = web pages with the applet

links = spreading links

node size  $\sim$  its influence (nb of adopting visitors)

node color  $\sim$  arrival time (darker if late)

vidéo

# Happy Flu : observations

**influence is proportional to the presence time**

even for very popular sites

# Happy Flu : observations

influence is proportional to the presence time

even for very popular sites

no *super-spreaders*

many visitors  $\sim$  weakly interested visitors?  
few visitors  $\sim$  high quality visitors?

# Diffusion : modélisation

## le système

noeuds atteints vs non-atteints

temps discret

transmission aux voisins

une seule fois



# Diffusion : modélisation

## le système

noeuds atteints vs non-atteints  
temps discret  
transmission aux voisins  
une seule fois

## stratégie/protocole de diffusion

une ou plusieurs sources ?  
lesquelles ? influenceurs ?  
choix des liens / des voisins ?  
mémoire ?

# Diffusion : modélisation

## le système

noeuds atteints vs non-atteints  
temps discret  
transmission aux voisins  
une seule fois

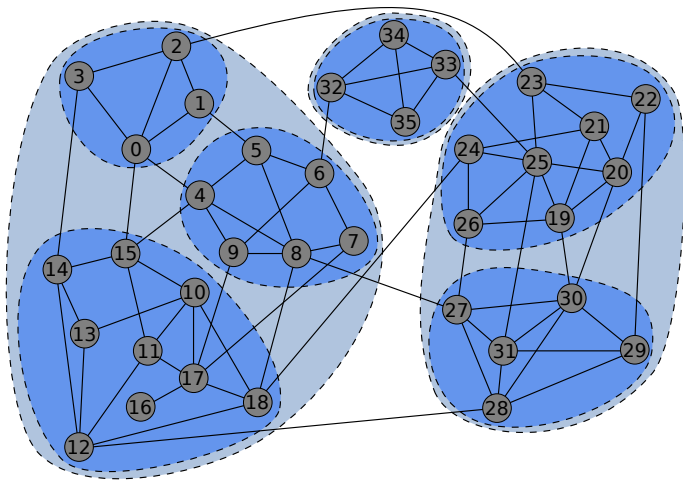
## stratégie/protocole de diffusion

une ou plusieurs sources ?  
lesquelles ? influenceurs ?  
choix des liens / des voisins ?  
mémoire ?

## objectifs / performances

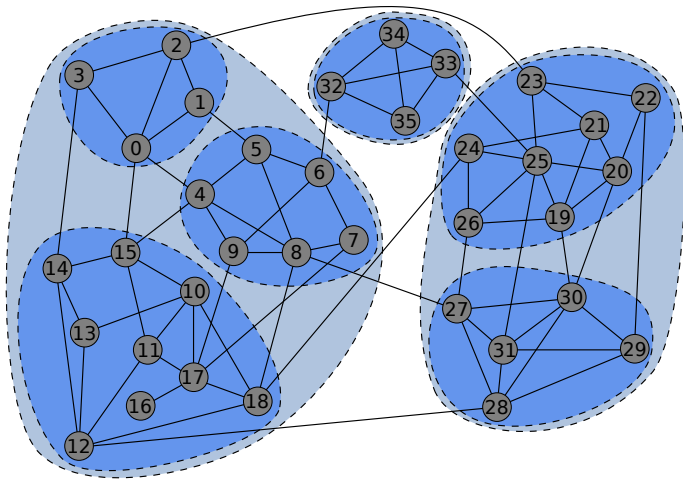
atteindre tout le monde ? 80% du total ? une cible ?  
rapidement ?  
coût en nombre d'échanges  
répartition de la charge

# Stratégies de diffusion : intuition



*l'information reste coincée dans les communautés*

# Stratégies de diffusion : intuition



*l'information reste coincée dans les communautés*

information apportée par un lien d'un triangle ?

impact de sa suppression ?

# Stratégies de diffusion : éléments

**noeuds de départ ?**

un seul vs plusieurs ? lesquels ?

composantes connexes, forts degrés, noeuds centraux (distance)

# Stratégies de diffusion : éléments

## noeuds de départ ?

un seul vs plusieurs ? lesquels ?

composantes connexes, forts degrés, noeuds centraux (distance)

## choix des voisins ?

inondation : tous les non-atteints, dès que possible

  marche aléatoire : un seul, au hasard ou biaisée

  inondation partielles, biaisée

# Stratégies de diffusion : éléments

## noeuds de départ ?

un seul vs plusieurs ? lesquels ?

composantes connexes, forts degrés, noeuds centraux (distance)

## choix des voisins ?

inondation : tous les non-atteints, dès que possible

marche aléatoire : un seul, au hasard ou biaisée

inondation partielles, biaisée

voisin(s) de plus fort degré, mieux ?

liens de faible coefficient de Jaccard

# Stratégies de diffusion : éléments

## noeuds de départ ?

un seul vs plusieurs ? lesquels ?

composantes connexes, forts degrés, noeuds centraux (distance)

## choix des voisins ?

inondation : tous les non-atteints, dès que possible

marche aléatoire : un seul, au hasard ou biaisée

inondation partielles, biaisée

voisin(s) de plus fort degré, mieux ?

liens de faible coefficient de Jaccard

↔ courbes :

nb atteints au fil du temps

en fonction du nb d'échanges

coût de la diffusion vs temps vs portée



# Attention : simulations !

modèles stylisés  
données pauvres (stopcovid)  
beaucoup de facteurs

# Attention : simulations !

modèles stylisés

données pauvres (stopcovid)

beaucoup de facteurs

Happy Flu : qualité des visiteurs

influence vs pression

qualité des liens

force des liens faibles

sociologie

# Charge/importance des noeuds/liens

**tous les noeuds / liens  
ne sont pas égaux**

certains traversés plus souvent  
lors d'une diffusion

↪ charge ; équilibrage ?

↪ importance dans la diffusion

# Exercice 6

# Centralités, importances

- ▶ degré

# Centralités, importances

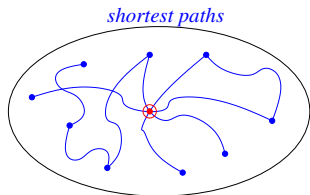
- ▶ degré

- ▶ proximité (*closeness*) :

somme des inverses des distances :  $c(v) = \sum_{u \neq v} \frac{1}{d(v,u)}$

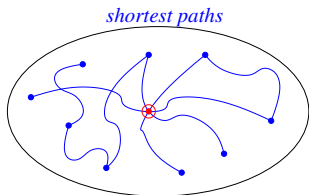
# Centralités, importances

- ▶ degré
- ▶ proximité (*closeness*) :  
somme des inverses des distances :  $c(v) = \sum_{u \neq v} \frac{1}{d(v,u)}$
- ▶ intermédiation (*betweenness*) :  
 $b(v) =$  nombre de plus courts chemins qui passent par  $v$



# Centralités, importances

- ▶ degré
- ▶ proximité (*closeness*) :  
somme des inverses des distances :  $c(v) = \sum_{u \neq v} \frac{1}{d(v,u)}$
- ▶ intermédiarité (*betweenness*) :  
 $b(v) =$  nombre de plus courts chemins qui passent par  $v$

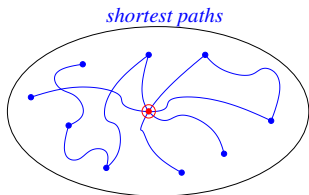


centralité des noeuds, des liens, des groupes



# Centralités, importances

- ▶ degré
- ▶ proximité (*closeness*) :  
somme des inverses des distances :  $c(v) = \sum_{u \neq v} \frac{1}{d(v,u)}$
- ▶ intermédiarité (*betweenness*) :  
 $b(v) =$  nombre de plus courts chemins qui passent par  $v$

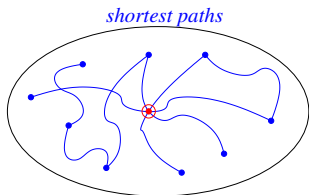


centralité des noeuds, des liens, des groupes

utilisations : robustesse, communautés, ranking, ...

# Centralités, importances

- ▶ degré
- ▶ proximité (*closeness*) :  
somme des inverses des distances :  $c(v) = \sum_{u \neq v} \frac{1}{d(v,u)}$
- ▶ intermédiation (*betweenness*) :  
 $b(v) =$  nombre de plus courts chemins qui passent par  $v$



centralité des noeuds, des liens, des groupes

utilisations : robustesse, communautés, ranking, ...

mais très coûteux  $\rightarrow$  marches aléatoires

$\hookrightarrow$  pagerank

# Pagerank

**classer les noeuds par importance**

google – graphe du web (orienté)

– intuition –

**important si beaucoup de liens entrants (degré)**

**depuis des noeuds importants**

# Pagerank

**classer les noeuds par importance**

google – graphe du web (orienté)

– intuition –

**important si beaucoup de liens entrants (degré)**

**depuis des noeuds importants**

– métaphore –

un surfer cliquant au hasard ; probabilité d'être sur un noeud

↔ marche aléatoire, diffusion

# Pagerank

**classer les noeuds par importance**

google – graphe du web (orienté)

– intuition –

**important si beaucoup de liens entrants (degré)**

**depuis des noeuds importants**

– métaphore –

un surfer cliquant au hasard ; probabilité d'être sur un noeud

↔ marche aléatoire, diffusion

– téléportation –

de temps en temps il saute au hasard

# Pagerank : algo

probabilité que le surfer soit sur  $v$  après  $t$  étapes ?

# Pagerank : algo

probabilité que le surfer soit sur  $v$  après  $t$  étapes ?

$$t = 0 : \text{tous égaux} : \frac{1}{n}$$

# Pagerank : algo

probabilité que le surfer soit sur  $v$  après  $t$  étapes ?

$$t = 0 : \text{tous égaux} : \frac{1}{n}$$

$t + 1$  : proba d'être sur un voisin entrant à  $t$  et de suivre ce lien  
+ proba de se téléporter sur  $v$



# Pagerank : algo

probabilité que le surfer soit sur  $v$  après  $t$  étapes ?

$$t = 0 : \text{tous égaux} : \frac{1}{n}$$

$t + 1$  : proba d'être sur un voisin entrant à  $t$  et de suivre ce lien  
+ proba de se téléporter sur  $v$

**Théorème : ça converge ! et c'est le PageRank.**

# Pagerank : algo

probabilité que le surfer soit sur  $v$  après  $t$  étapes ?

$$t = 0 : \text{tous égaux} : \frac{1}{n}$$

$t + 1$  : proba d'être sur un voisin entrant à  $t$  et de suivre ce lien  
+ proba de se téléporter sur  $v$

**Théorème : ça converge ! et c'est le PageRank.**

remarque : graphe orienté

↔ impasses

# Pagerank : algo

probabilité que le surfer soit sur  $v$  après  $t$  étapes ?

$$t = 0 : \text{ tous égaux : } \frac{1}{n}$$

$t + 1$  : proba d'être sur un voisin entrant à  $t$  et de suivre ce lien  
+ proba de se téléporter sur  $v$

**Théorème : ça converge ! et c'est le PageRank.**

remarque : graphe orienté

↔ impasses

remarque : rapide

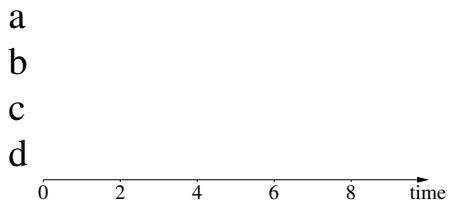
# Pagerank : en pratique

element clé du code :

observation sur le WoT, corrélation avec degrés

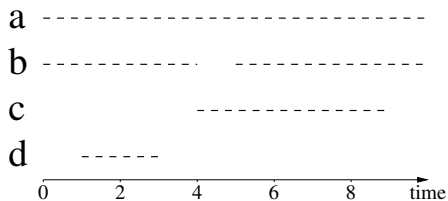
```
cat wot.dat | python pr.py > wot.pr
```

# interactions over time



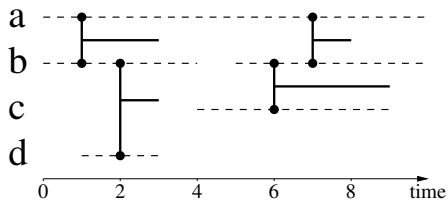
- $a$ ,  $b$ ,  $c$ , and  $d$  for 10 time units

# interactions over time



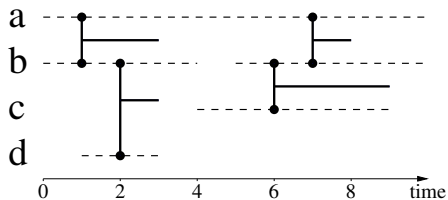
- $a$ ,  $b$ ,  $c$ , and  $d$  for 10 time units
- $a$  always present,  $b$  leaves from 4 to 5,  $c$  present from 4 to 9,  $d$  from 1 to 3

# interactions over time



- *a*, *b*, *c*, and *d* for 10 time units
- *a* always present, *b* leaves from 4 to 5, *c* present from 4 to 9, *d* from 1 to 3
- *a* and *b* interact from 1 to 3 and from 7 to 8; *b* and *c* from 6 to 9; *b* and *d* from 2 to 3.

# interactions over time

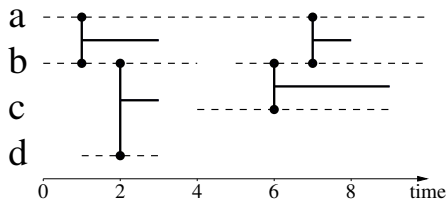


- $a$ ,  $b$ ,  $c$ , and  $d$  for 10 time units
- $a$  always present,  $b$  leaves from 4 to 5,  $c$  present from 4 to 9,  $d$  from 1 to 3
- $a$  and  $b$  interact from 1 to 3 and from 7 to 8;  $b$  and  $c$  from 6 to 9;  $b$  and  $d$  from 2 to 3.

*e.g., social interactions, network traffic,  
money transfers, chemical reactions, etc.*



# interactions over time

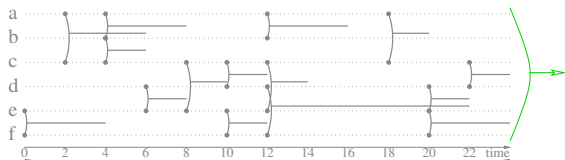


- $a$ ,  $b$ ,  $c$ , and  $d$  for 10 time units
- $a$  always present,  $b$  leaves from 4 to 5,  $c$  present from 4 to 9,  $d$  from 1 to 3
- $a$  and  $b$  interact from 1 to 3 and from 7 to 8;  $b$  and  $c$  from 6 to 9;  $b$  and  $d$  from 2 to 3.

*e.g., social interactions, network traffic,  
money transfers, chemical reactions, etc.*

**how to describe such data ?**

# structure or dynamics

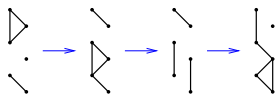
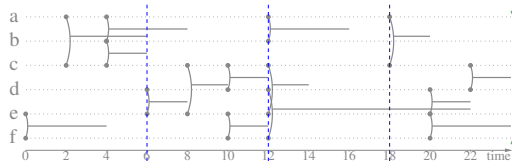


graph theory  
network science  
→ structure

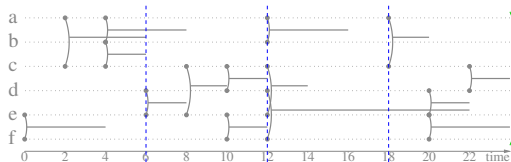


signal analysis, time series → dynamics

# structure and dynamics ?



# structure and dynamics ?



graph theory  
network science  
→ structure



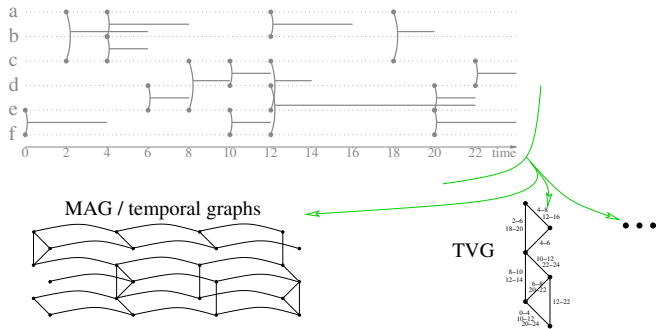
signal analysis, time series → dynamics



time slices  
→ graph sequence

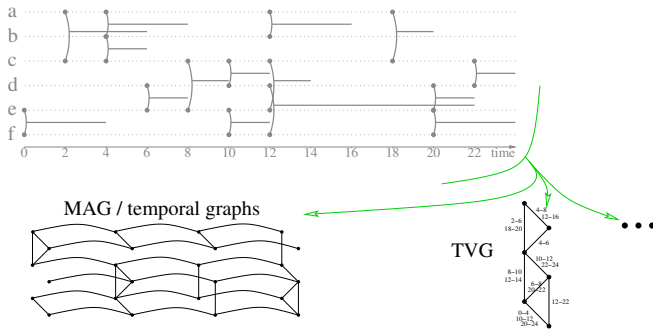
information loss  
what slices ?  
graph sequences ?

# structure and dynamics



lossless but graph-oriented

# structure and dynamics



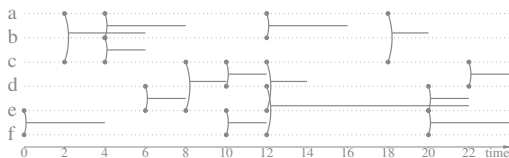
lossless but graph-oriented

+ ad-hoc properties (mostly path-related) + contact sequences + relational event models + ...

# an other approach

deal with the stream directly

stream graphs and link streams



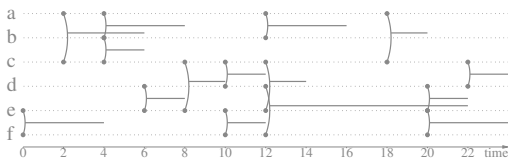
graph theory  
network science

signal analysis, time series

# an other approach

deal with the stream directly

stream graphs and link streams



graph theory  
network science

signal analysis, time series

wanted features : simple and intuitive, comprehensive,  
time-node consistent, generalizes graphs/signal



# definition of stream graphs

Graph  $G = (V, E)$  with  $E \subseteq V \otimes V$   
 $uv \in E \Leftrightarrow u$  and  $v$  are linked

# definition of stream graphs

Graph  $G = (V, E)$  with  $E \subseteq V \otimes V$   
 $uv \in E \Leftrightarrow u$  and  $v$  are linked

**Stream graph**  $S = (T, V, W, E)$

$T$  : time interval,  $V$  : node set

$W \subseteq T \times V$ ,  $E \subseteq T \times V \otimes V$

# definition of stream graphs

Graph  $G = (V, E)$  with  $E \subseteq V \otimes V$   
 $uv \in E \Leftrightarrow u$  and  $v$  are linked

**Stream graph  $S = (T, V, W, E)$**

$T$  : time interval,  $V$  : node set

$W \subseteq T \times V$ ,  $E \subseteq T \times V \otimes V$

$(t, v) \in W \Leftrightarrow v$  is present at time  $t$

$T_v = \{t, (t, v) \in W\}$

# definition of stream graphs

Graph  $G = (V, E)$  with  $E \subseteq V \otimes V$   
 $uv \in E \Leftrightarrow u$  and  $v$  are linked

**Stream graph  $S = (T, V, W, E)$**

$T$  : time interval,  $V$  : node set

$W \subseteq T \times V$ ,  $E \subseteq T \times V \otimes V$

$(t, v) \in W \Leftrightarrow v$  is present at time  $t$

$$T_v = \{t, (t, v) \in W\}$$

$(t, uv) \in E \Leftrightarrow u$  and  $v$  are linked at time  $t$

$$T_{uv} = \{t, (t, uv) \in E\}$$

# definition of stream graphs

Graph  $G = (V, E)$  with  $E \subseteq V \otimes V$   
 $uv \in E \Leftrightarrow u$  and  $v$  are linked

**Stream graph  $S = (T, V, W, E)$**

$T$  : time interval,  $V$  : node set

$W \subseteq T \times V$ ,  $E \subseteq T \times V \otimes V$

$(t, v) \in W \Leftrightarrow v$  is present at time  $t$

$$T_v = \{t, (t, v) \in W\}$$

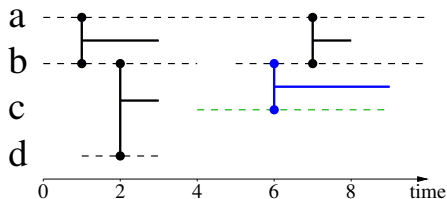
$(t, uv) \in E \Leftrightarrow u$  and  $v$  are linked at time  $t$

$$T_{uv} = \{t, (t, uv) \in E\}$$

$(t, uv) \in E$  requires  $(t, u) \in W$  and  $(t, v) \in W$

*i.e.*  $T_{uv} \subseteq T_u \cap T_v$

## an example



$$T = [0, 10] \quad V = \{a, b, c, d\}$$

$$W = T \times \{a\} \cup ([0, 4] \cup [5, 10]) \times \{b\} \cup [4, 9] \times \{c\} \cup [1, 3] \times \{d\}$$

$$T_a = T \quad T_b = [0, 4] \cup [5, 10] \quad T_c = [4, 9] \quad T_d = [1, 3]$$

$$E = ([1, 3] \cup [7, 8]) \times \{ab\} \cup [6, 9] \times \{bc\} \cup [2, 3] \times \{bd\}$$

$$T_{ab} = [1, 3] \cup [7, 8] \quad T_{bc} = [6, 9] \quad T_{bd} = [2, 3] \quad T_{ad} = \emptyset$$

## a few remarks

works with...

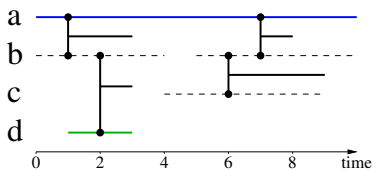
discrete time, continuous time,  
instantaneous interactions or with durations,  
directed, weighted, bipartite...

if  $\forall v, T_v = T$  then  $S \sim L = (T, V, E)$  is a **link stream**

if  $\forall u, v, T_{uv} \in \{T, \emptyset\}$  then  $S \sim G = (V, E)$  is a  
**graph-equivalent stream**

# size of a stream graph

*How many nodes? How many links?*

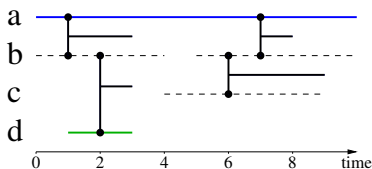


$$|\mathbf{T}_a| = 10 \neq |\mathbf{T}_d| = 2$$



# size of a stream graph

*How many nodes? How many links?*



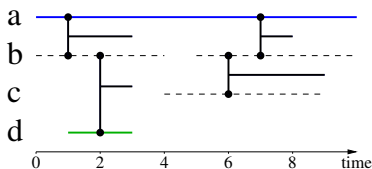
$$|T_a| = 10 \neq |T_d| = 2$$

$$n = \sum_{v \in V} \frac{|T_v|}{|T|}$$

$$n = \frac{|T_a|}{10} + \frac{|T_b|}{10} + \frac{|T_c|}{10} + \frac{|T_d|}{10} = 1 + 0.9 + 0.5 + 0.2 = 2.6 \text{ nodes}$$

# size of a stream graph

How many nodes? How many links?



$$|T_a| = 10 \neq |T_d| = 2$$

$$n = \sum_{v \in V} \frac{|T_v|}{|T|}$$

$$m = \sum_{uv \in V \otimes V} \frac{|T_{uv}|}{|T|}$$

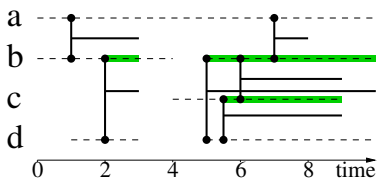
$$n = \frac{|T_a|}{10} + \frac{|T_b|}{10} + \frac{|T_c|}{10} + \frac{|T_d|}{10} = 1 + 0.9 + 0.5 + 0.2 = 2.6 \text{ nodes}$$

$$m = \frac{|T_{ab}|}{10} + \frac{|T_{bc}|}{10} + \frac{|T_{bd}|}{10} = 0.3 + 0.3 + 0.1 = 0.7 \text{ links}$$

# neighborhood

in  $G = (V, E) : N(v) = \{u, uv \in E\}$

in  $S = (T, V, W, E) : N(v) = \{(t, u), (t, uv) \in E\}$

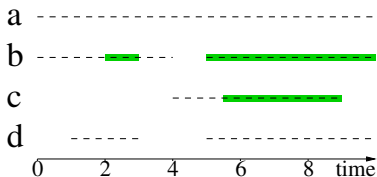


$$N(d) = ([2, 3] \cup [5, 10]) \times \{b\} \cup [5.5, 9] \times \{c\}$$

# degree

in  $G$  and in  $S$  :

$d(v)$  is the size of  $N(v)$



$$N(d) = ([2, 3] \cup [5, 10]) \times \{b\} \cup [5.5, 9] \times \{c\}$$

$$d(d) = \frac{|[2,3] \cup [5,10]|}{10} + \frac{|[5.5,9]|}{10} = 0.6 + 0.35 = 0.95$$

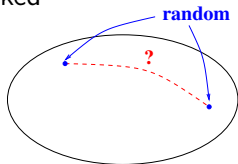
- degree distribution, average degree, etc
- if graph-equivalent stream then graph degree
- relation with  $n$  and  $m$

# density

in  $G$  :

proba two random nodes are linked

$$\begin{aligned}\delta(G) &= \frac{\text{nb links}}{\text{nb possible links}} \\ &= \frac{2 \cdot m}{n \cdot (n-1)}\end{aligned}$$

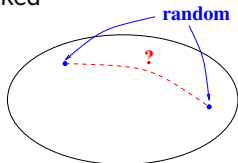


# density

in  $G$  :

proba two random nodes are linked

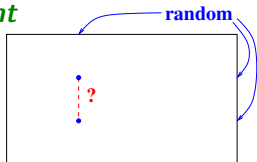
$$\begin{aligned}\delta(G) &= \frac{\text{nb links}}{\text{nb possible links}} \\ &= \frac{2 \cdot m}{n \cdot (n-1)}\end{aligned}$$



in  $S$  :

proba two random nodes are linked  
*at a random time instant*

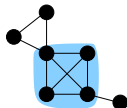
$$\begin{aligned}\delta(S) &= \frac{\text{nb links}}{\text{nb possible links}} \\ &= \frac{\sum_{uv \in V \otimes V} |T_{uv}|}{\sum_{uv \in V \otimes V} |T_u \cap T_v|}\end{aligned}$$



- if graph-equivalent stream then graph density
- relation with  $n$ ,  $m$ , and average degree

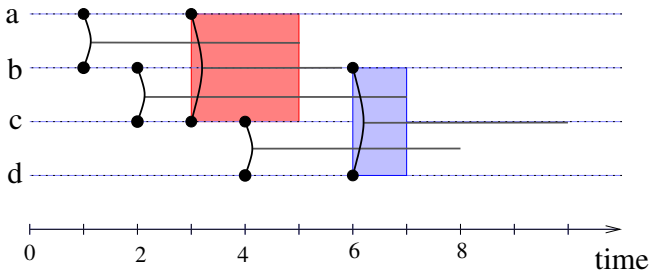
# cliques

in  $G$  : sub-graph of density 1  
*all nodes are linked together*



in  $S$  : **sub-stream of density 1**

*all nodes interact all the time*



# Stream graph dans un fichier



# Exercice 7

# Plan

Conclusion  
Examen

## *analyse de réseaux*

domaine très riche, nombreuses applications

graphes... et beaucoup plus

vaste ensemble de propriétés, vocabulaire

méthodologie : comparaison à l'aléatoire

recherche de principes sous-jacents

manipulations, programmation, statistiques

observations WoT de  $\checkmark$ 1

WoT Ğ1 comme exemple

**mais typique des graphes réels**

- densité très faible – degrés hétérogènes –
- forte localité – communautés – etc –

**graphes réels similaires entre eux**  
**mais très différents d'aléatoires**

**fin des années 90**

# Généralité

WoT Ğ1 comme exemple

**mais typique des graphes réels**

- densité très faible – degrés hétérogènes –
- forte localité – communautés – etc –

**graphes réels similaires entre eux**

**mais très différents d'aléatoires**

**fin des années 90**

également : **problématiques en commun :**

mesure, analyse, modélisation,  
dynamique, algorithmique, ...

# Ce que je n'ai pas dit...

- ▶ **communautés** : leur calcul, leur évaluation ; coeurs de communautés, communautés floues
- ▶ **anomalies** : définitions (scénario, statistiques), détection, validation ; temps réel, filtrage
- ▶ **librairies** de graphes : NetworkX, NetworkKit, graph-tool, etc ; bases de données graphes
- ▶ **biais de mesure**, impact des erreurs (ex : adresses bitcoin ; ajout d'un lien) ; propriétés robustes ?
- ▶ **du constat à l'action** : quels vaccinations/quarantaines ? quels ajouts pour améliorer la robustesse, etc
- ▶ **en vrai** : graphes valués, étiquetés, orientés, multipartis, dynamiques...
- ▶ ...

# Examen 2 – énoncé

# Examen 2 – questions