Modèle d'une propagation d'une épidémie

DRISSI Mohamed Reda
reda-mohamed@isty.uvsq.fr

10 mars 2019

Table des matières

I Introduction

Dans le cadre du module "Calcul Haute Performance et Simulation", il nous a été demandé de modéliser le problème de la propagation d'une maladie dans une population. Le but de ce projet est de faire différentes simulation afin de vérifier l'avantage d'une optimisation pageRank appliquée sur un tel problème.

Nous avons choisi de développer notre projet en python 3.7 pour son abandance de librairies en calcul matriciel et parce que nous le maitrisons bien. Nous utilisons la librairie networkx pour nos graphes. Nous utilisons ensuite des matrices numpy pour représenter nos matrices de transitions. Pour avoir les représentation graphique de nos simulation, nous utilisons la bibliothèque matplot.

Ce rapport est découpé en deux parties. Dans un premier temps nous aborderons la modélisation du problème. Puis nous développerons les résultats que nous avons obtenus après simulation.

II Modélisation

Le problème de la propagation d'un virus dans une population peut être assimilé au problème de la promenade aléatoire d'un individus sur internet (d'où le choix de l'optimisation pageRank).

- Internet → Une population
- Une page sur internet → Un individu dans la population
- Un promeneur → Un virus
- La promenade du marcheur → La propagation du virus
- Le rang d'une page est la probabilité de la présence du promeneur sur cette page → Le rand est la probabilité d'être infecté par le virus durant une épidémie

II.1 Graphes

Nous avons receuilli plusieurs graphes de ce lien, les fichiers contiennent des listes d'arrêtes, le test par défaut exécutera le graphe <> de taille <> et de densité <>.

II.2 Spécifications de la machine utilisée

- CPU : Intel Core i7
  - Modèle : 6700K
  - Fréquence : 4.0GHZ
  - nombre de coeurs/nombre de threads : 4 cores/8 logical threads(HyperThreading©)
  - Turbo boost : off
- RAM : Corsair CMK16GX4M2B3000C15 Vengeance LPX 16GB DDR4 3000MHz C15 XMP 2.0
- Stockage : Samsung 850 PRO SSD 512GB

II.3 Système

- OS : Debian 9.8 Stretch (stable) x86_64
- Kernel : 4.9.0-8-amd64
- Python 3.7.0

II.4 Topologie du système

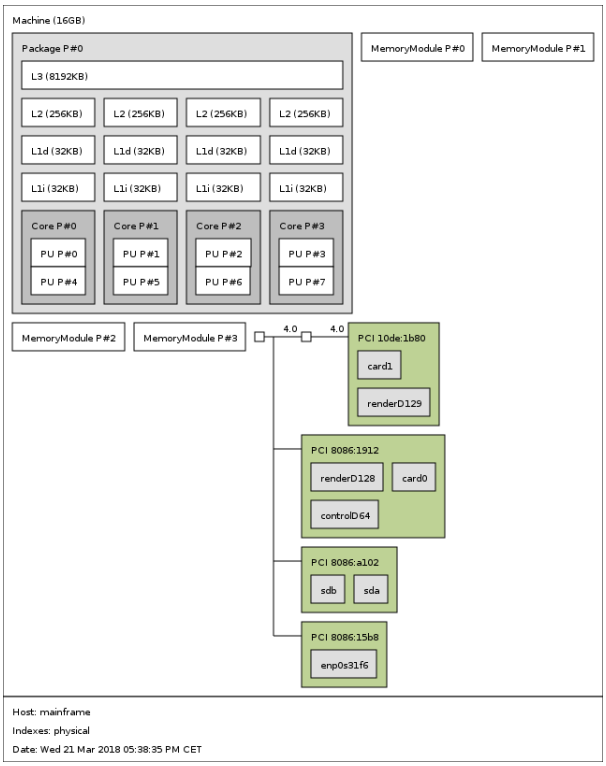

FIGURE 1 – Topologie générée par lstopo

II.5 Paramètres de simulation

Voici quelques paramètres que nous avons utilisé pour les différentes simulations.

- Nombre d'itérations = 150
- Facteur de damping $\alpha = 0.8$
- Pourcentage de personnes infecté initialement : $x = 5\%$
- Probabilité d'infecter un voisin : $\nu = 1 - \alpha = 0.2$
- Probabilité de guérir par hasard :$\gamma = 0.02$

III Simulations

III.1 Résultats

Nous allons utiliser le script `full.sh` qui nous calculera tous les graphes fournis, nous avons trouvé ces graphes sur le site de snaps de Stanford [1].
Pour un usage granulaire, une aide est fournie (seul le nom de fichier est obligatoire) :

```
bash~$  cd src
bash~$  python -m app.main -h
usage: main.py [-h] [-g FILENAME] [-i INFECTED] [-v RANDOM_VACCINATION]
               [-H HEAL] [-c CONTAMINATION] [-t ITER]

Simple epidemic modeling example using PageRank

optional arguments:
  -h, --help            show this help message and exit
  -g FILENAME, --input-graph FILENAME
                        File containing the edge list of the matrix
                        to process
  -i INFECTED, --infected INFECTED
                        Ratio of initially randomly individuals
  -v RANDOM_VACCINATION, --vaccinate RANDOM_VACCINATION
                        Ratio of initially randomly vaccinated individuals
  -H HEAL, --heal HEAL  Probability of randomly healing when infected
  -c CONTAMINATION, --contamination CONTAMINATION
                        Probability of contaminating each neighbor when
                        infected
  -t ITER, --iteration ITER
                        Number of iterations of the simulation
```

III.1.1 email-Eu-core.txt

The network was generated using email data from a large European research institution Réseau généré avec les données mail d'une institution de recherche européenne.

```
Name:
Type: DiGraph
Number of nodes: 1005
Number of edges: 25571
Average in degree:  25.4438
Average out degree:  25.4438
Density:0.050684822897464864
Ratio of initially infected individuals:0.05
Ratio of initially randomly vaccinated individuals:0.22
Probability of contaminating each neighbor when infected:0.12
Probability of randomly healing when infected:0.26
Number of iterations:150
Time spent to run each simulation:
  No Vaccination: 641 ms
  Random Vaccination: 374 ms
  PageRank Vaccination: 391 ms
```

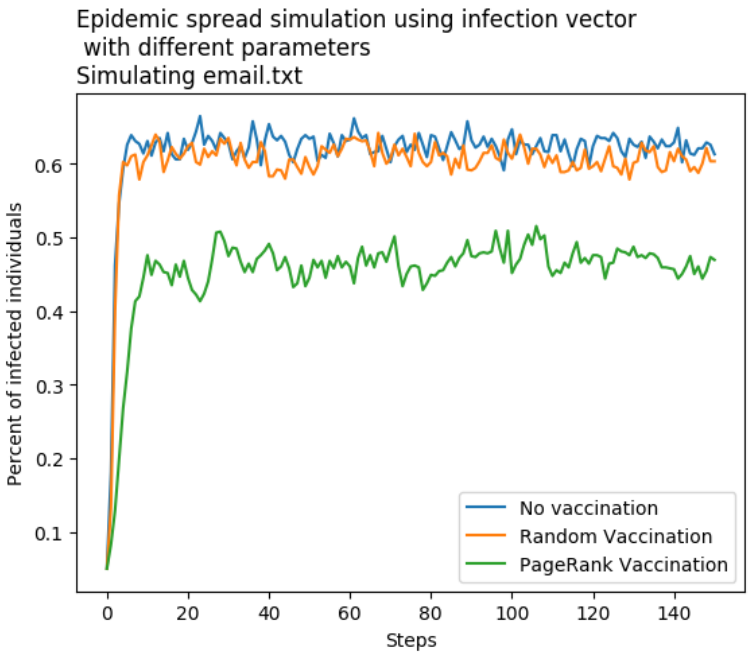

FIGURE 2 – Graphe de simulation email-Eu-core

III.1.2 p2p-Gnutella08.txt

Réseau de partage de fichiers Gnutella en P2P d'août 2002

```
Name:
Type: DiGraph
Number of nodes: 6301
Number of edges: 20777
Average in degree:   3.2974
Average out degree:   3.2974
Density:0.0010467978123905755
Ratio of initially infected individuals:0.05
Ratio of initially randomly vaccinated individuals:0.22
Probability of contaminating each neighbor when infected:0.2
Probability of randomly healing when infected:0.26
Number of iterations:150
Time spent to run each simulation:
No Vaccination: 14 seconds 744 miliseconds
Random Vaccination: 3 seconds 276 miliseconds
PageRank Vaccination: 974 miliseconds
```

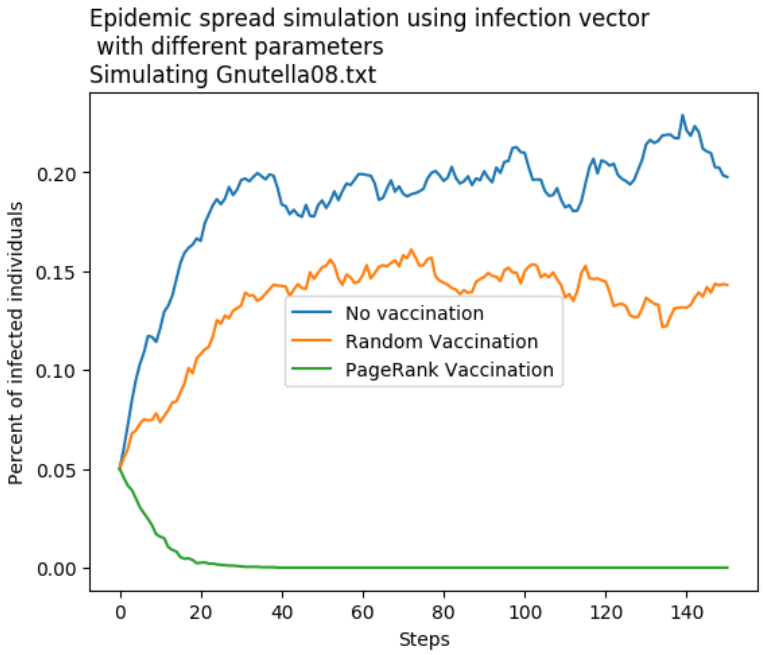

FIGURE 3 – Graphe de simulation p2p-Gnutella08

III.1.3 p2p-Gnutella09.txt

Réseau de partage de fichiers Gnutella en P2P d'août 2002

```
Name:
Type: DiGraph
Number of nodes: 8114
Number of edges: 26013
Average in degree:   3.2059
Average out degree:   3.2059
Density:0.0007903217921884197
Ratio of initially infected individuals:0.05
Ratio of initially randomly vaccinated individuals:0.22
Probability of contaminating each neighbor when infected:0.2
Probability of randomly healing when infected:0.26
Number of iterations:150
Time spent to run each simulation:
No Vaccination: 25 seconds 538 miliseconds
Random Vaccination: 7 seconds 89 miliseconds
PageRank Vaccination: 1 seconds 549 miliseconds
```

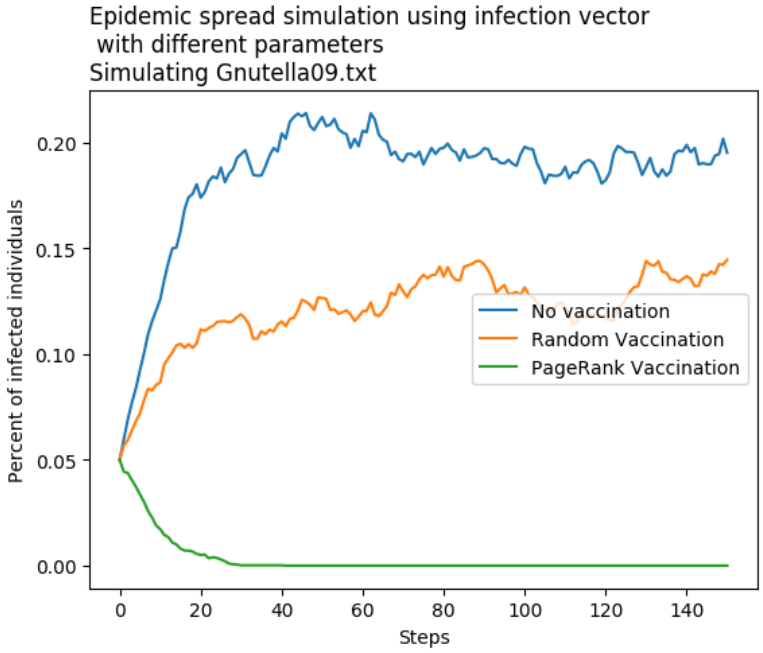

FIGURE 4 – Graphe de simulation p2p-Gnutella09

III.1.4 soc-sign-bitcoinalpha.csv

Réseau représentant les personnes se faisant confiance dans un échange de bitcoins sur la plateforme bitcoin-alpha.

```
Name:
Type: DiGraph
Number of nodes: 3783
Number of edges: 24186
Average in degree:   6.3933
Average out degree:   6.3933
Density:0.0033809299947872786
Ratio of initially infected individuals:0.05
Ratio of initially randomly vaccinated individuals:0.22
Probability of contaminating each neighbor when infected:0.2
Probability of randomly healing when infected:0.26
Number of iterations:150
Time spent to run each simulation:
No Vaccination: 11 seconds 582 miliseconds
Random Vaccination: 4 seconds 650 miliseconds
PageRank Vaccination: 1 seconds 354 miliseconds
```

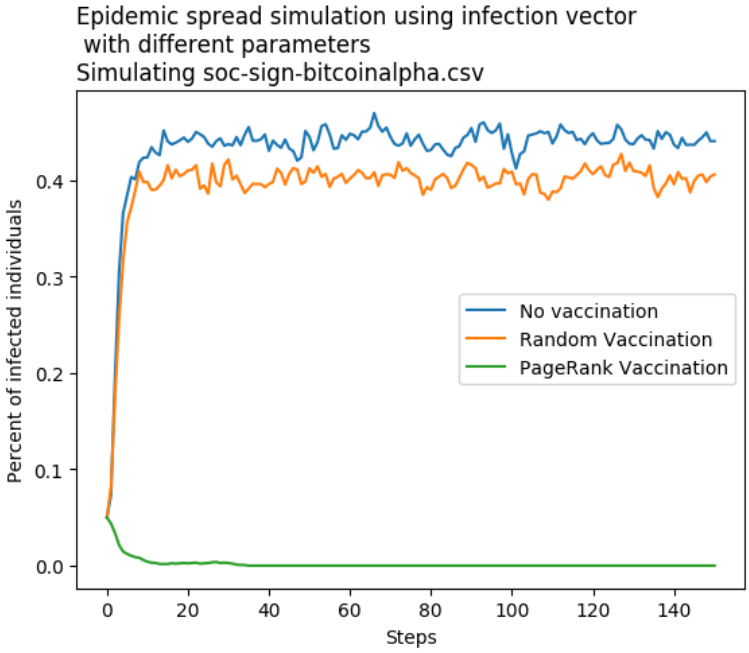

FIGURE 5 – Graphe de simulation soc-sign-bitcoinalpha

III.1.5 wiki-Vote.txt

Réseau représentant les votes entre les différents utilisateurs de wikipedia.

```
Name:
Type: DiGraph
Number of nodes: 7115
Number of edges: 103689
Average in degree:  14.5733
Average out degree:  14.5733
Density:0.004097075022161917
Ratio of initially infected individuals:0.05
Ratio of initially randomly vaccinated individuals:0.22
Probability of contaminating each neighbor when infected:0.2
Probability of randomly healing when infected:0.26
Number of iterations:150
Time spent to run each simulation:
No Vaccination: 22 seconds 609 miliseconds
Random Vaccination: 12 seconds 966 miliseconds
PageRank Vaccination: 1 seconds 993 miliseconds
```

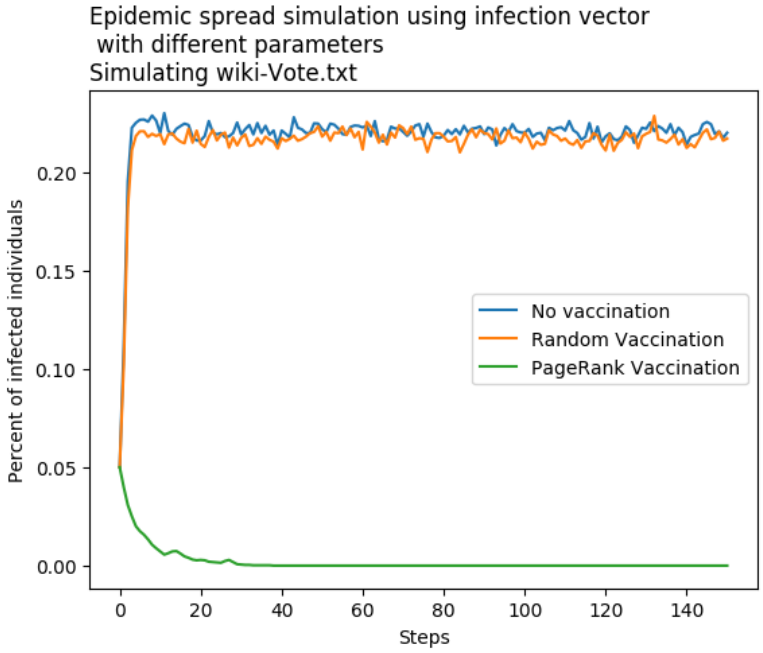

FIGURE 6 – Graphe de simulation wiki-Vote

IV Analyses

Nous remarquons l'importance de l'optimisation pageRank dans presque tous les exemples, les graphes ont la même allure, et change drastiquement le résultat, nous remarquons aussi que les vaccinations aléatoires génère un graphe à allure similaire à celui de la non vaccination. Donc sur un échantillon assez important nous ne remarquerons donc pas de différence.
Le changement du pourcentage de vaccination vers 0.12 et 0.32 ne change pas l'allure du résultat.
À chaque fois, l'algorithme pageRank commence à se stabiliser entre les itérations 20 et 40, si le pourcentage de vaccination est de 22%. Quand nous augmentons ce pourcentage, cette stabilisation peut changer mais que légèrement.

Bibliographie

[1] Jure Leskovec and Andrej Krevl. SNAP Datasets : Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.