

# Campagne de Simulation

**Sorbonne Université  
Sciences Sorbonne Université  
Master informatique: RES  
PROGRES**

Adrien Koumgang Tegantchouang 21109170

31 décembre 2021

## Table des matières

<b>1</b>	<b>Présentation du projet</b>	<b>3</b>
1.1	Introduction . . . . .	3
1.2	La collecte des données expérimentales. . . . .	3
1.3	2. Manipulation des données expérimentales . . . . .	3
1.4	Tracer des données expérimentales . . . . .	4
<b>2</b>	<b>Présentation des parties principales de 'Simulation Campaign'</b>	<b>4</b>
2.1	La simulation . . . . .	4
2.2	L'analyse des simulations . . . . .	7
2.3	Data graph . . . . .	7
2.4	Enregistrement des résultats . . . . .	15
<b>3</b>	<b>Liens utiles</b>	<b>16</b>
3.1	Lien vers la repository Github du projet . . . . .	16
3.2	Lien vers la vidéo de présentation du projet . . . . .	17

# 1 Présentation du projet

## 1.1 Introduction

Le but de ce projet est d'utiliser un simulateur fourni pour mener une campagne de simulation : exécuter le simulateur avec de nombreux paramètres différents, collecter et manipuler les données produites par le simulateur, obtenir des graphiques résumant l'impact des paramètres suivis sur les indicateurs choisis.

Ce simulateur est écrit en Java et modélise la propagation d'un virus dans une population. Il est fourni sous la forme d'un fichier Virus.jar qui intègre toutes les bibliothèques nécessaires. On peut lancer le simulateur avec les paramètres par défaut avec la commande (Java doit être installé sur la machine qui exécute) :

```
“ java -jar Virus.jar “
```

Le code source du simulateur est fourni en annexe. Ce code source est fourni à titre indicatif pour comprendre la signification de chaque paramètre passé au simulateur.

## 1.2 La collecte des données expérimentales.

Réalisation en Python et en utilisant la bibliothèque 'subprocess' un programme qui exécute plusieurs fois le simulateur avec les paramètres de simulation. Ces exécutions multiples sont utilisées pour obtenir des intervalles de confiance suffisants lors de la phase de visualisation des données expérimentales. Les résultats intermédiaires sont stockés dans un ensemble de fichiers pour l'ensemble des simulations menées (pour un ensemble de paramètres donné).

Dans un deuxième temps, on réalise en Python un programme qui exécute le simulateur en faisant varier un paramètre de la simulation, en réalisant plusieurs simulations pour chaque ensemble de paramètres. Les résultats intermédiaires sont stockés dans un ensemble de fichiers pour l'ensemble des simulations menées.

## 1.3 2. Manipulation des données expérimentales

A l'aide des fichiers produits précédemment, on réalise en Python et en utilisant les bibliothèques 'numpy' et 'pandas' un programme qui permette d'obtenir les informations suivantes :

- Quel est l'impact du nombre initial d'infectés sur :
  - la durée de l'épidémie
  - la proportion maximale de la population qui est infectée
  - la distribution des multi-infections
- Quel est l'impact du rayon de mobilité sur :
  - la durée de l'épidémie
  - la proportion de la population qui est infectée
  - la distribution des multi-infections
- Quel est l'impact de la proportion de vaccinés sur :
  - la durée de l'épidémie
  - la proportion de la population qui est infectée
  - distribution des multi-infections
- Quel est l'impact de l'efficacité vaccinale sur :

- la durée de l'épidémie
- la proportion de la population qui est infectée
- la distribution des multi-infections
- Quel est l'impact de la durée d'infection sur :
  - la durée de l'épidémie
  - la proportion de la population qui est infectée
  - la distribution des multi-infections
- Quel est l'impact de la durée de contagiosité sur :
  - la durée de l'épidémie
  - la proportion de la population qui est infectée
  - la distribution des multi-infections
- Quel est l'impact de la durée d'immunité sur :
  - la durée de l'épidémie
  - la proportion de la population qui est infectée
  - la distribution des multi-infections
- Quel est l'impact de la densité de population sur :
  - la durée de l'épidémie
  - la proportion de la population qui est infectée
  - la distribution des multi-infections

Les informations ainsi produites sont stockés dans plusieurs fichiers.

## 1.4 Tracer des données expérimentales

A l'aide des fichiers produits précédemment, on réalise en Python et en utilisant la bibliothèque 'matplotlib' un programme qui produit des graphiques pertinents pour évaluer l'impact des paramètres suivis (nombre initial d'infectés, rayon de mobilité, proportion de vaccinés, efficacité vaccinale, durée de contagiosité, durée d'immunité, densité de population) sur les indicateurs sélectionnés (la durée de l'épidémie, la proportion de la population qui est infectée, la distribution des multi-infections). Les graphiques ainsi produits figures dans le rapport.

## 2 Présentation des parties principales de 'Simulation Campaign'

### 2.1 La simulation

Elle se fait grace au code source simulator.py. Ce fichier python permet l'exécution d'un ensemble de simulation en fonction d'une variable passé en ligne de commande au programme et aussi d'un ensemble de paramètres, tout aussi passé en ligne de commande lors du lancement de l'exécution du programme. La syntax générale d'exécution du programme de simulation est la suivante :

- `python3 -name_var=name [-number_of_simulation=x]`
- `[-variable_value_begin=x]`
- `[-variable_value_end=x]`
- `[-var_value_incremented=x]`
- `[-other_interest_var=list_var]`
- `[-path_save_result=pathname]`

- name\_var : [name] name of the variable of interest
- =0 for nb nodes
- =1 for nb infected
- =2 for travel distance
- =3 for nb vaccinated
- =4 for vaccine efficiency
- =5 for infection period
- =6 for contagion period
- =7 for immune period
- =x number of simulations to do for each value of the variable of interest
- variable\_value\_begin : [=x] initial value for variable simulation for the simulations
- variable\_value\_end : [=x] final value for variable simulation for the simulations
- var\_value\_incremented : [=x] increment value for variable simulation
- other\_interest\_var : [=list\_var] list of other parameters to switch to the simulator. ex : -stop\_all\_sane=1,-gui=0,-printout=2
- path\_save\_result : [=pathname] path to directory where save file to result of the simulator

Ces informations peuvent être obtenu depuis le code source en l'exécutant sans arguments.

Pour ce projet, nous nous sommes intéressés à 8 variables et créer des fichiers 'bash' pour chacune d'elle enfin d'y passer des paramètres exact pour leur simulation :

- contagion\_period (var\_contagion\_period\_simulation.sh) :
  - name\_var = 6
  - nombre de simulations = 50
  - valeur de début = 150
  - valeur de fin = 300
  - shift = 10
  - paramètres à passer au simulateur : '-stop\_all\_sane=1, -gui=0, -printout=2'
  - dossiers de sauvegardes des fichiers résultats des simulations : 'result\_simulations/contagion\_period'
- immune\_period (var\_immune\_period\_simulation.sh) :
  - name\_var = 7
  - nombre de simulations = 100
  - valeur de début = 200
  - valeur de fin = 400
  - shift = 20
  - paramètres à passer au simulateur : '-stop\_all\_sane=1, -gui=0, -printout=2'
  - dossiers de sauvegardes des fichiers résultats des simulations : 'result\_simulations/immune\_period'
- nb\_infected (var\_nb\_infected\_simulation.sh) :
  - name\_var = 1
  - nombre de simulations = 100
  - valeur de début = 1
  - valeur de fin = 10
  - shift = 1
  - paramètres à passer au simulateur : '-stop\_all\_sane=1, -gui=0, -printout=2'
  - dossiers de sauvegardes des fichiers résultats des simulations : 'result\_simulations/nb\_infected'

- `infection_period` (`var_infection_period_simulation.sh`) :
    - `name_var` = 5
    - nombre de simulations = 100
    - valeur de début = 50
    - valeur de fin = 150
    - `shift` = 10
    - paramètres à passer au simulateur : `'-stop_all_sane=1, -gui=0, -printout=2'`
    - dossiers de sauvegardes des fichiers résultats des simulations : `'result_simulations/infection_period'`
  - `nb_nodes` (`var_nb_nodes_simulation.sh`) :
    - `name_var` = 0
    - nombre de simulations = 100
    - valeur de début = 100
    - valeur de fin = 150
    - `shift` = 10
    - paramètres à passer au simulateur : `'-stop_all_sane=1, -gui=0, -printout=2'`
    - dossiers de sauvegardes des fichiers résultats des simulations : `'result_simulations/nb_nodes'`
  - `travel_distance` (`var_travel_distance_simulation.sh`) :
    - `name_var` = 2
    - nombre de simulations = 100
    - valeur de début = 150
    - valeur de fin = 250
    - `shift` = 10
    - paramètres à passer au simulateur : `'-stop_all_sane=1, -gui=0, -printout=2'`
    - dossiers de sauvegardes des fichiers résultats des simulations : `'result_simulations/travel_distance'`
  - `nb_vaccinated` (`var_nb_vaccinated_simulation.sh`) :
    - `name_var` = 3
    - nombre de simulations = 100
    - valeur de début = 0
    - valeur de fin = 10
    - `shift` = 1
    - paramètres à passer au simulateur : `'-stop_all_sane=1, -gui=0, -printout=2'`
    - dossiers de sauvegardes des fichiers résultats des simulations : `'result_simulations/nb_vaccinated'`
  - `vaccine_efficiency` (`var_vaccine_efficiency_simulation.sh`) :
    - `name_var` = 4
    - nombre de simulations = 100
    - valeur de début = 1
    - valeur de fin = 10
    - `shift` = 1
    - paramètres à passer au simulateur : `'-stop_all_sane=1, -gui=0, -printout=2'`
    - dossiers de sauvegardes des fichiers résultats des simulations : `'result_simulations/vaccine_efficiency'`
- Le programme de simulation offre un feedback de 2 niveaux :
- le premier : il affiche les paramètres passés au simulateur
  - le deuxième : pour chaque valeur de la variable de simulation, il indique à quelle simulation il

se situe

Nous avons choisit d'implémenter le simulateur de cette manière enfin de permettre la simulation précise d'une variable ainsi que de ses valeurs aux choix. (Ceci dû au tant que prend les simulations, cela nous a permis de lancer une simulation puis de l'arreter pour la relancer plus tard, donnant ainsi une grande flexibilité dans l'exécution des simulations)

## 2.2 L'analyse des simulations

Elle se fait grâce au code source analyse.py. Ici, l'analyse des fichiers résultats de la simulation se fait en parallèle pour chaque variable. Ainsi on a un thread qui s'occupe de l'analyse de chaque variable.

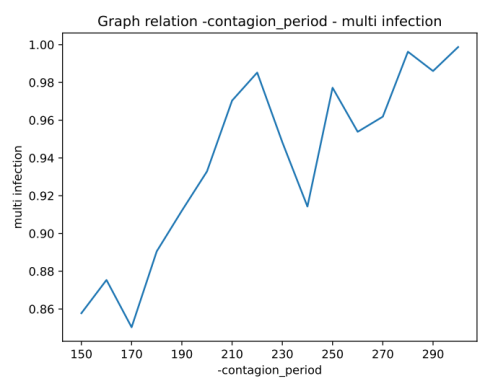
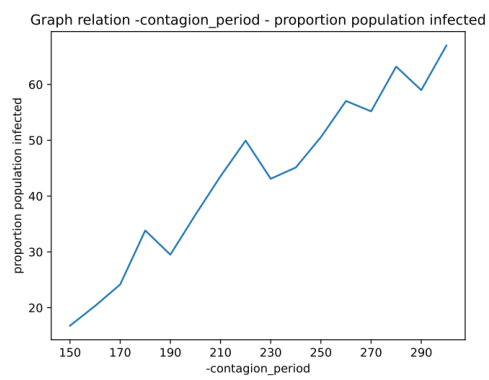
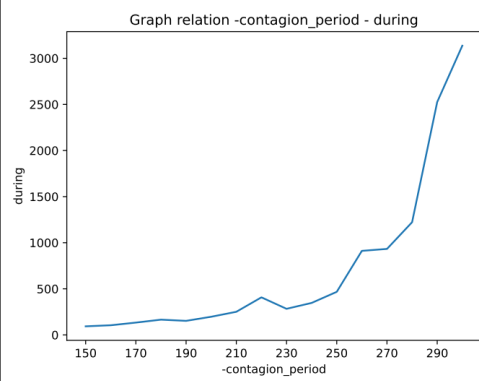
L'analyse des fichiers est orienté principalement sur 3 points :

- Calcul de la durée de chaque simulation. Et elle se fait grâce à la fonction '`calcul_during_epidemic`'. Ici nous avons choisi de calculer la durée en nombre de snapshot, ainsi la durée n'est point exprimé en temps mais en nb shapshot.
- La proportion (max ou pas) de la population infectée. Et elle se fait grâce à la fonction '`calcul_proportion_population_infected`'. Ici nous faisons le calcul en déterminant le nombre d'infecté que nous avons à un moment t (ici fesant référence à un snapshot) et pour la proportion simple, nous prenons la valeur moyenne et pour celle max, nous prenons le maximum obtenu.
- Le cas de calcul de la proportion max fait référence à la simulation du nombre initial d'infecté
- La proportion des multi-infections. Et elle se fait grâce à la fonction '`calcul_proportion_multi_infection`'. Ici nous faisons le calcul en déterminant le nombre de fois qu'un node est infecté durant toute la durée de la simulation.

## 2.3 Data graph

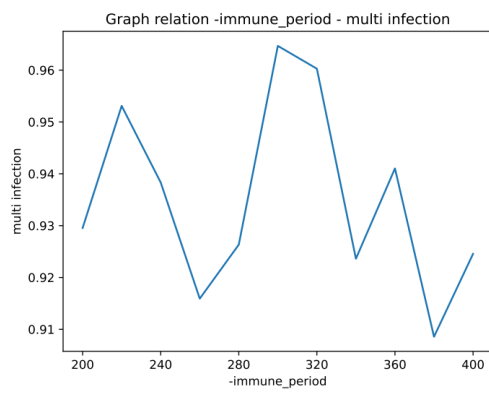
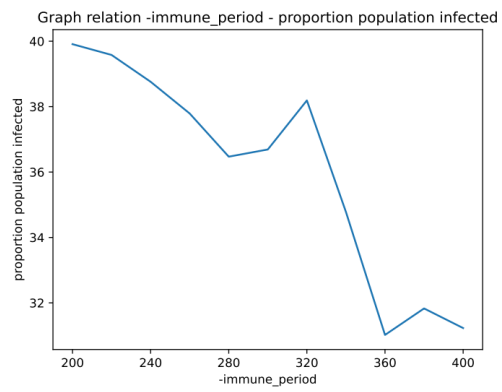
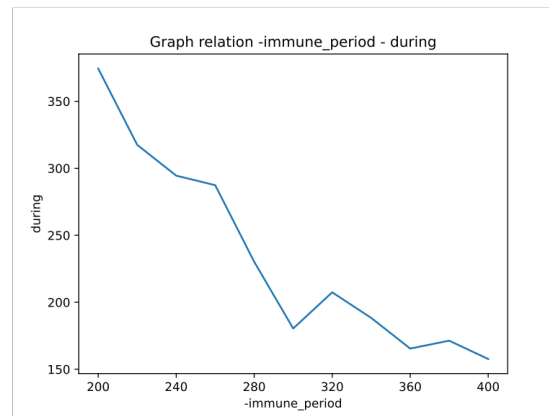
Elle se fait grâce au code source analyse.py et se fait directement après l'analyse des simulations. Il fait appel à la library '`pyplot`' et trace un graph pour chaque variable et centre d'interet.

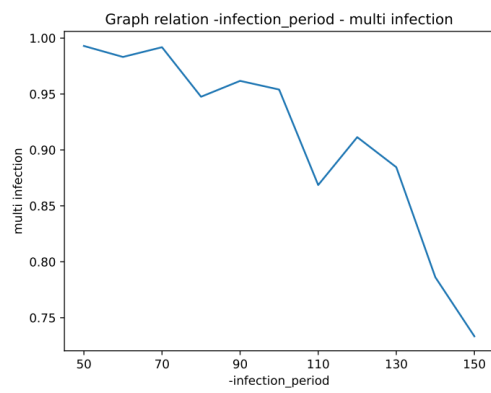
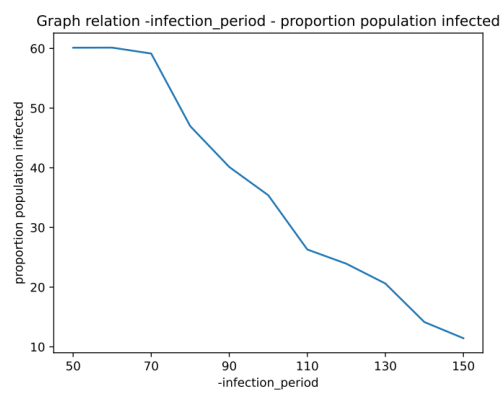
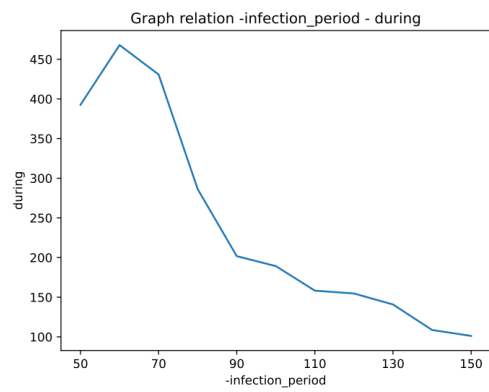
Ainsi on obtient :

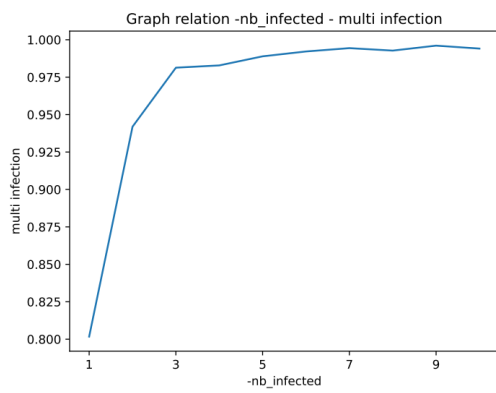
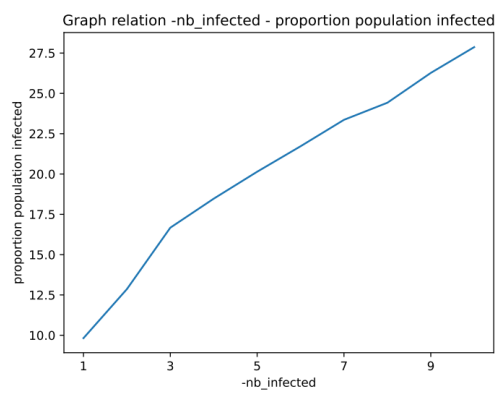
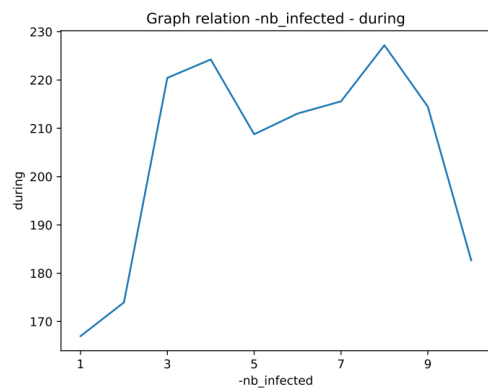


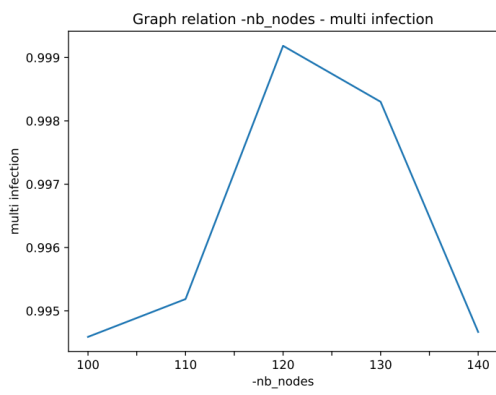
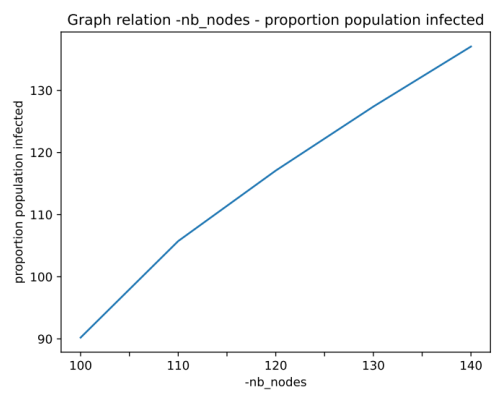
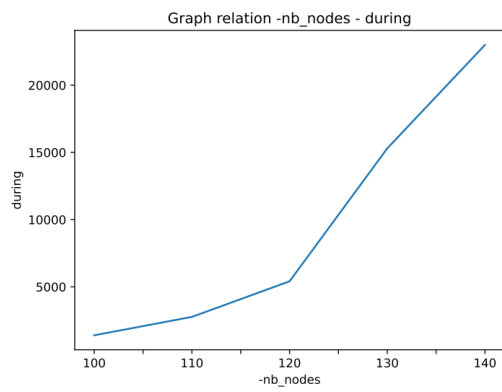


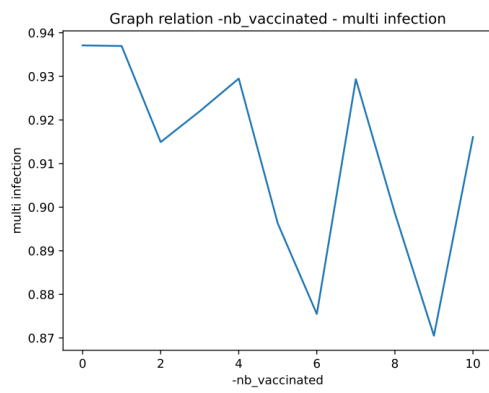
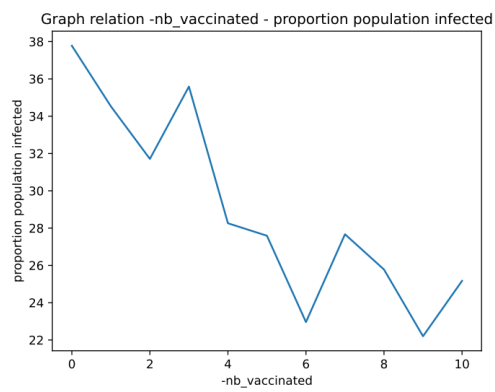
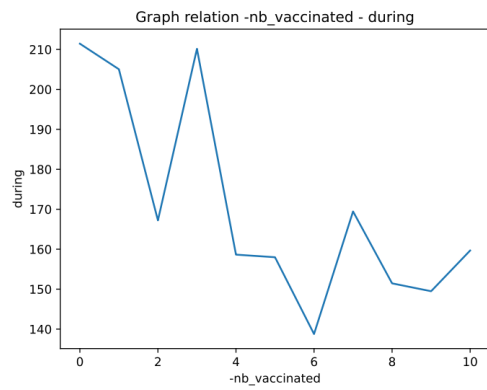
.

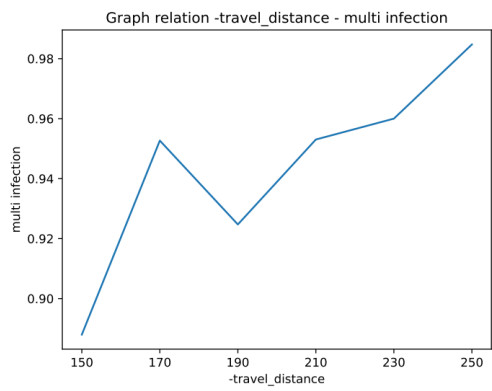
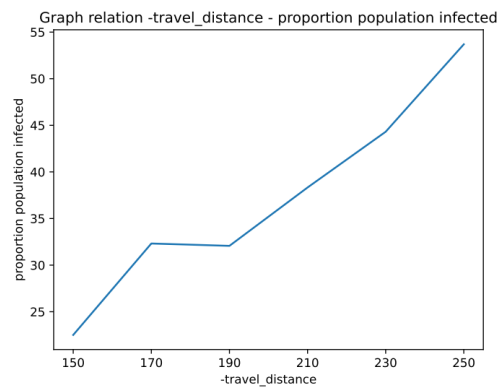
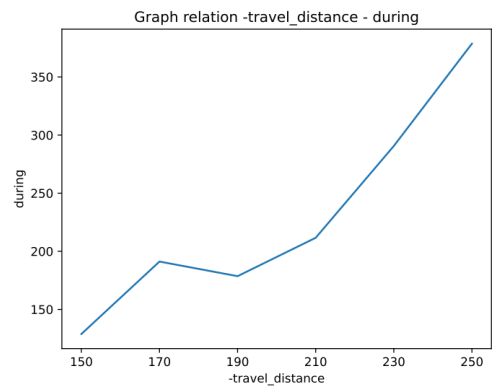


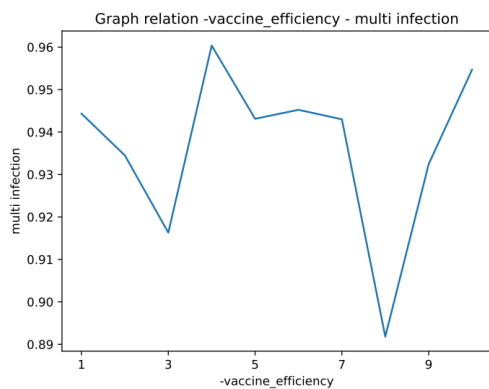
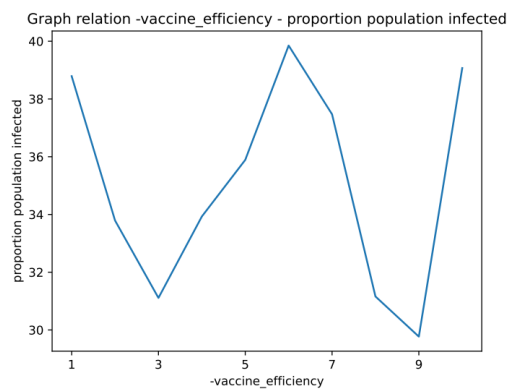
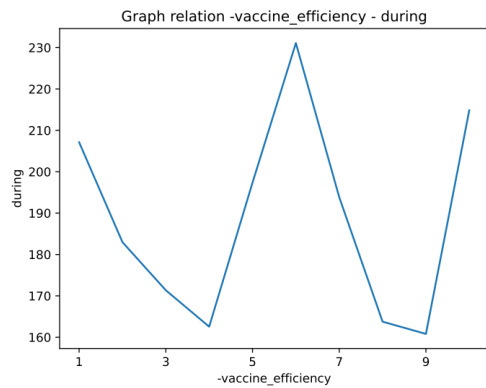












## 2.4 Enregistrement des résultats

Les résultats ainsi obtenus (tableaux et graph) sont sauvegardés dans des fichiers :

- nb\_infected :
  - dossier : result\_analyse/nb\_infected
- durée : infected\_during.csv et infected\_during.pdf
- proportion population infected : infected\_proportion.csv et infected\_proportion.pdf
- multi infection : infected\_multi\_infection.csv et infected\_multi\_infection.pdf

- travel\_distance :
  - dossier : result\_analyse/travel\_distance
  - durée : travel\_distance\_during.csv et travel\_distance\_during.pdf
  - proportion population infected : travel\_distance\_proportion.csv et travel\_distance\_proportion.pdf
  - multi infection : travel\_distance\_multi\_infection.csv et travel\_distance\_multi\_infection.pdf
- nb\_vaccinated :
  - dossier : result\_analyse/nb\_vaccinated
  - durée : vaccinated\_during.csv et vaccinated\_during.pdf
  - proportion population infected : vaccinated\_proportion.csv et vaccinated\_proportion.pdf
  - multi infection : vaccinated\_multi\_infection.csv et vaccinated\_multi\_infection.pdf
- vaccine\_efficiency :
  - dossier : result\_analyse/vaccine\_efficiency
  - durée : vaccine\_efficiency\_during.csv et vaccine\_efficiency\_during.pdf
  - proportion population infected : vaccine\_efficiency\_proportion.csv et vaccine\_efficiency\_proportion.pdf
  - multi infection : vaccine\_efficiency\_multi\_infection.csv et vaccine\_efficiency\_multi\_infection.pdf
- infection\_period :
  - dossier : result\_analyse/infection\_period
  - durée : infection\_period\_during.csv et infection\_period\_during.pdf
  - proportion population infected : infection\_period\_proportion.csv et infection\_period\_proportion.pdf
  - multi infection : infection\_period\_multi\_infection.csv et infection\_period\_multi\_infection.pdf
- contagion\_period :
  - dossier : result\_analyse/contagion\_period
  - durée : contagion\_period\_during.csv et contagion\_period\_during.pdf
  - proportion population infected : contagion\_period\_proportion.csv et contagion\_period\_proportion.pdf
  - multi infection : contagion\_period\_multi\_infection.csv et contagion\_period\_multi\_infection.pdf
- immune\_period :
  - dossier : result\_analyse/immune\_period
  - durée : immune\_period\_during.csv et immune\_period\_during.pdf
  - proportion population infected : immune\_period\_proportion.csv et immune\_period\_proportion.pdf
  - multi infection : immune\_period\_multi\_infection.csv et immune\_period\_multi\_infection.pdf
- nb\_nodes :
  - dossier : result\_analyse/nb\_nodes
  - durée : nodes\_during.csv et nodes\_during.pdf
  - proportion population infected : nodes\_proportion.csv et nodes\_proportion.pdf
  - multi infection : nodes\_multi\_infection.csv et nodes\_multi\_infection.pdf

## 3 Liens utiles

### 3.1 Lien vers la repository Github du projet

[projet gihhub](https://github.com/adrienKoumgangT/SimulationCampaignPython).



### **3.2 Lien vers la vidéo de présentation du projet**

[video presentation mp3](<https://drive.google.com/drive/folders/1dzkosGB8VdcnGMDaBYeceW4Zbv5ZyTo?usp=sharing>).