# MU4IN014 – Mini-Projet 3 : Rendu

## Sommaire :



## Introduction

Le mini-projet 3 a été réalisé par DIEP Richard (3700573) et comporte trois exercices. Le premier exercice consiste à exécuter le programme Virus.jar dans un code écrit sous Python. Ce dernier doit permettre de faire varier les paramètres de la simulation Virus.jar. Le second exercice consiste à obtenir les informations des différentes simulations et de les récupérer pour l'exercice 3, ainsi que d'observer l'impact des différents paramètres. Le dernier exercice a pour but de récupérer les données de l'exercice précédent, et d'en faire des graphiques pertinents.

## Collection des données expérimentales : exec_prog.py

Un programme appelé exec_prog.py est disponible et permet d'exécuter la simulation et de faire varier les différents paramètres. A la fin du programme, se trouve une boucle dont la variable nb_exec va de n à m, tel que n > 0 et m >= n. nb_exec est la variable qui va faire varier les paramètres.

Dans la figure ci-contre, on peut voir que nb_inf possède la variable nb_exec, et donc lorsqu'on exécute exec_prog, la simulation aura pour nb_inf, les valeurs allant de 1 à 10 (nb_exec). Pour faire varier un paramètre, il suffit de changer la plage de nb_exec, et d'affecter la variable nb_exec au paramètre désiré (ici, nb_inf).

```
for nb_exec in range(1, 10):
        nb_snaps = 2500
        nb_samp = 25
        nb_inf = nb_exec
        travel_dist = 200
        infec_period = 100
        cont_period = 200
        immu_period = 300
        nb_nod = 100
        f name = "exec data/nb infected-"+str(nb exec)+".txt"
```

D'autres paramètres importants sont à notés :

- nb_snaps = 2500, cette valeur a été choisie car la simulation dure environ quinze minutes sur ma machine. Lorsque nb_snaps est atteint, on considère que c'est une exécution infinie.
- nb_samp = 25, la valeur correspond à 25 exécutions de la simulation avec les mêmes paramètres. Cette valeur semble suffisante pour permettre d'obtenir une moyenne assez représentative bien qu'il est possible d'en faire plus. Mais pour la limite de temps accordée pour réaliser tous les tests, 25 semble correct.

- f_name correspond au nom de fichier. Dans ce cas, les fichiers se trouvent dans un dossier exec_data et les fichiers ont pour nom « nb_infected-nb_exec.txt », avec nb_exec la valeur correspondante.
- Le reste des paramètres doivent être modifiés pour faire varier les paramètres de la simulation.

Le programme s'exécute avec des threads pour accélérer le nombre d'exécutions. Il n'est pas nécessaire de protéger les fichiers écrits puisque chaque thread écrit sur un fichier différent.

## Manipuler des données expérimentales : count_samp.py & data_reader.py

Le programme count_samp.py permet de compter le nombre d'exécution réalisé pour un même paramètre et le programme data_reader.py permet de lire les données, et de les traduire. Les fichiers lus proviennent des fichiers qu'a généré exec_prog.

On peut observer plusieurs paramètres modifiables :

```
list_file = glob("nb_nodes/*.txt")

do_output = True #True si on veut un fichier de sortie pour les résultats, False sinon
output_file = "data.txt" #Nom du fichier de sortie contenant les informations récupérées

#open(output_file, 'w').close() #Vide le fichier

cpt_read = 0 #Compte le nombre de fichier lu

option_line = 'nodes.' #Indique quelle ligne l'option a été modifiée
"""
Options list:
infected nodes initially
as travel distance
as infection period
as contagion period
as immune period
nodes.
"""
```

- list_file, c'est le répertoire où on veut lire les fichiers générés.
- output_file est le nom de fichier généré par ce code. Et donc il y a deux types de fichiers générés jusqu'à présent : les informations de la simulation et les données récoltées sous un format spécifiques.
- option_line, cette variable permet de lire les paramètres modifiés. Dans l'exemple précédent, nous avons fait varier nb_inf qui est le nombre de nœuds infectés au départ. Alors, l'option_line va valloir « infected nodes initially ». Cette option est importante pour lire la bonne ligne modifiée.

Le format des données de data_reader.py correspond à ceci :

```
filename=travel_dist/travel_dist-75.txt
nb_tests=25
option=75
list_time=[131, 185, 172, 73, 109, 134, 84, 63, 133, 96, 92, 114, 75, 50, 66, 80, 90, 179, 260, 19
list_max_infec=[2, 2, 2, 2, 2, 2, 2, 3, 3, 4, 6, 10, 10, 11, 12, 15, 15, 14, 15, 15, 15, 17, 18, 2
list_mult_infec=[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 58, 0, 0, 0, 0, 0, 59]
-----
filename=travel_dist/travel_dist-100.txt
nb_tests=25
option=100
list_time=[131, 441, 92, 48, 90, 228, 98, 70, 108, 113, 138, 159, 257, 181, 173, 212, 162, 157, 17
list_max_infec=[2, 2, 2, 2, 2, 2, 2, 3, 4, 5, 6, 7, 8, 8, 8, 10, 10, 11, 12, 13, 13, 12, 12, 12, 1
list_mult_infec=[0, 134, 0, 0, 0, 34, 0, 0, 0, 0, 0, 0, 48, 2, 18, 78, 2, 10, 18, 140, 0, 0, 4, 0,
-----
filename=travel_dist/travel_dist-350.txt
nb_tests=25
option=350
list_time=[2500, 2500, 2500, 2500, 2500, 2500, 2500, 2500, 2500, 58, 105, 2500, 2500, 2500, 2500,
list_max_infec=[2, 2, 2, 2, 2, 2, 2, 3, 4, 4, 6, 9, 9, 10, 11, 12, 12, 14, 15, 15, 18, 21, 24, 26,
list_mult_infec=[3556, 3590, 3445, 3582, 3516, 3426, 3457, 3487, 3717, 0, 4, 3454, 3661, 3556, 356
```

- Le nom de fichier filename
- Le nombre de tests nb_tests effectués pour ce paramètre (nb_samps)
- option correspond à la valeur du paramètre
- list_time est une liste contenant la durée de chaque exécution.
- list_max_infec correspond au nombre d'infecté par itération. Attention, il faut différencier une exécution et une itération. Une exécution contient plusieurs itérations. Cela n'a pas été très clair pour ma part entre choisir la proportion de la population qui est infectée sur une exécution (donc le maximum d'une exécution) ou la proportion de chaque itération (donc le maximum d'une itération). De ce fait, j'ai choisi ce dernier.
- list_mult_infec est le nombre de multi-infections pour chaque exécution. Le calcul se fait pour chaque exécution et compte pour chaque colonne le nombre de fois où le nœud a été infecté. Il faut avoir été infecté au minimum deux fois pour être considéré comme multi-infection.

En ce qui concerne l'impact des paramètres sur la simulation, nous allons expliquer brièvement avec les graphiques générés de l'exercice 3 (le nombre initial d'infectés est expliqué dans la vidéo donc l'explication ne sera pas reprise ici.) : (Les graphiques qui ne sont pas mentionnés ne sont pas forcément pertinents)

- Rayon de mobilité (Toutes les 25 valeurs allant de 25 jusqu'à 1000)
    - Le graphique nous montre que la durée atteint le maximum lorsque les valeurs du rayon de mobilité sont proches du maximum (1000) et de la moyenne (500). La durée atteint le minimum lorsque les valeurs du rayon de mobilité sont proches du minimum (100). Toutes les valeurs dans les intervalles varient du maximum au minimum et la durée moyenne avoisine 1250 itérations. L'histogramme nous montre une certaine concentration vers 100.
    - Le graphique permet de constater une moyenne de 20 infectés au maximum par itération. Cette fois-ci, on remarque que pour les valeurs du rayon se situant vers le minimum (150), le taux d'infecté est toujours inférieur à la moyenne alors que pour les valeurs du rayon se situant vers le maximum (900), il y a deux phases : La première phase jusqu'à 1000 itérations où la courbe passe d'un extrême à un autre et la seconde phase commençant à 1000 itérations où il n'y a aucune donnée. Cela signifie que la simulation se termine de façon précoce. Les autres valeurs avoisinent vers la moyenne.
    - Le graphique pour les multi-infections est similaire au graphique sur la durée d'épidémie. La moyenne avoisine 1750 multi-infections.
- Durée d'infection (Toutes les 5 valeurs allant de 5 à 400)
    - On constate via le graphique deux types de courbes : les valeurs dont la durée d'infection étant proches de 100 sont supérieures à la moyenne qui est d'environ 400 itérations, le reste des valeurs sont inférieures à la moyenne. L'histogramme et la carte de chaleur démontre aussi une abondance de valeur en dessous de la moyenne.
    - On observe sur le graphique deux types de données : de 0 à 500 itérations, on a une moyenne avoisinante 10 infectés au maximum alors qu'à partir de 500 itérations, la moyenne vaut 20 infectés. C'est notamment dû au fait que plus la durée d'infection est élevée, moins la simulation ne dure. L'histogramme et la carte de chaleur montre une fois de plus une concentration proche de 0 à 5.
    - Le graphique montre une fois de plus ce qu'on a constaté pour la durée de l'épidémie mais pour les multi-infections cette fois-ci. Les valeurs sélectionnées allant de 25 à 125 montrent une courbe supérieur ou avoisinant la moyenne de 450 multi-infections par itération alors que les autres valeurs supérieures sont proches de 0. Même remarque que pour la durée de l'épidémie pour l'histogramme et la carte de chaleur.
- Durée de contagiosité (Toutes les 25 valeurs allant de 25 à 1000)
    - La moyenne de la durée est d'environ 2000. Plus les valeurs de la durée de contagiosité sont élevées, plus la durée atteint le maximum. A partir de 300, les durées sont toujours infinies. On peut observer que c'est à partir de 200 de durée qu'on commence à observer des pics atteignant l'infini.
    - Le nombre maximum d'infections avoisine en moyenne 50. Cependant, on peut observer des pics vers 5, qui peut signifier que certaines exécutions se sont terminées à ces moments précis. Plus la durée de contagion est élevée, plus le nombre maximum d'infectés augmente. A partir de 600 de durée, on a un taux maximum d'infectés proche de la moyenne ou supérieur. L'histogramme, ainsi que la carte chaleur, montrent bien la concentration vers 40 à 60 infections.
    - La moyenne de multi-infections est de 2000. On a ici les mêmes remarques que pour les graphiques sur la durée de l'épidémie. Les données les plus pertinents sont sélectionnées et on observe qu'à 150 de durée de contagiosité, le nombre de multi-infections est strictement inférieure à la moyenne de multi-infections alors qu'à partir de 300 de durée, on atteint un

maximum de 3000 multi-infections. Cependant, on peut aussi observer que le nombre de multi-infections diminue à partir de 600 de durée, tout en restant supérieur à la moyenne de 2000. Les valeurs affectant les multi-infections sont plutôt équilibrées d'après ce qu'on observe sur l'histogramme et la carte de chaleur.

- Durée d'immunité (Toutes les 25 valeurs allant de 25 à 1000)
    - Le graphique possède une durée moyenne de 850 itérations et on observe que plus la durée de l'immunité augmente, moins la simulation dure. A partir de la valeur 400, la durée est inférieure à la durée moyenne. L'histogramme nous montre bien qu'il y a deux parties : Ceux qui dure à l'infini, et ceux qui durent vers 250 itérations. La carte de chaleur nous montre également une concentration vers 250.
    - Le nombre moyen d'infectés est de 20. Une fois de plus, plus la durée d'immunité augmente, moins il y a d'infectés. A partir de 400, le nombre d'infectés avoisine la moyenne. Le chronogramme montre que c'est assez équilibré en termes d'infectés.
    - Le nombre de multi-infections avoisine en moyenne 1100. Comme on a vu pour la durée de l'épidémie, à partir de 400, le nombre de multi-infections diminue et est inférieur à la moyenne. Le chronogramme et la carte de chaleur montre aussi ces informations.
- Densité de population (Toutes valeurs de 5 à 200, puis toutes les 5 valeurs entre 200 et 400, les valeurs supérieures ne sont pas utiles)
    - On peut observer une moyenne de 1400 itérations, sachant que pour les valeurs inférieures à 100 nœuds et supérieures à 400 nœuds ont une durée proche de 100. Le chronogramme et la carte de chaleur nous montrent bien cette concentration.
    - Le nombre maximum d'infections est en moyenne de 50. On peut observer qu'à partir de 375 nœuds, le taux d'infectés est quasiment au maximum et la durée se limite à environ 1000 itérations. Pour toutes valeurs se trouvant entre 100 et 325 nœuds, plus la valeur est grande, plus il y a d'infectés, sachant qu'à partir de 200, on se retrouve à la moyenne voire plus. Des pics peuvent être constatés par moment et peuvent signifier la fin de certaines exécutions comme pour la courbe verte à 8000 itérations. La carte de chaleur montre une concentration montre qu'il y a bien une concentration vers 50 infectés. L'histogramme montre la même chose.
    - Le nombre de multi-infections est en moyenne de 4000. Une fois de plus, ce qui a été constaté dans la durée d'épidémie est identique ici mais pour les multi-infections. Les multi-infections sont à 0 pour les valeurs dont la densité est inférieure à 100 nœuds et supérieures à 400 nœuds. Le chronogramme montre deux points importants qui permet de voir qu'on a bien les multi-infections très élevées ou nulles. De même que pour la carte de chaleur, la multi-infection est concentrée à 0.

## Tracer des données expérimentales : graph.py

Vous pouvez constater qu'il n'y a pas de fichier graph.py mais plutôt des fichiers de type « graph_nb_infected.py ». En effet, j'ai décidé de laisser les options choisies pour chaque graph et donc, chaque graph est adapté à un paramètre. Lorsque nous parlons de graph.py, nous parlons de tous les fichiers commençant par « graph_ » et terminant par « .py ».

Nous allons commencer par parler de quelques options présentes dans le code :

```
filename = "nb_infected_data.txt"

out_name = "nb_infected" #Fichier
out_path = f"graphs/{out_name}/"

legend_title = out_name
```

- filename correspond au nom de fichier qu'on veut lire et en créer un graphique. Ce fichier correspond au fichier généré par data_reader.py et non celui généré par exec_prog.py.
- out_name correspond au nom de fichier de sortie, mais aussi au nom de dossier qu'on doit créer en avance.
- out_path correspond au chemin où seront créés les graphiques.
- legend_title correspond au title de la légende si elle est disponible.

Les graphiques générés sont du format : time_graph_« out_name ».pdf dans le cas du graphique sur la durée d'épidémie.

Lors de l'ouverture d'un fichier et de sa lecture, nous utilisons les données présentes provenant de data_reader.py. Comme nous pouvons le constater, le début de chaque ligne est important. Chaque donnée est stockée dans une liste hormis le nom du fichier. La fonction split est utilisée pour récupérer uniquement les valeurs qui nous intéresse.

```
with open(filename, 'r') as f:
        for line in f:
                tmp = []
                line_split = line.split('=')
                if line_split[0] == 'filename':
                        nb_files += 1
                elif line_split[0] == 'option':
                        tab_option.append(int(sub('\n', '', line_split[1])))
                elif line_split[0] == 'list_time':
                        tmp = sub('[][,\n]', '', line_split[1]).split(' ')
                        tab_time.append([int(t) for t in tmp])
                elif line_split[0] == 'list_max_infec':
                        tmp = sub('[][,\n]', '', line_split[1]).split(' ')
                        tab_max_infec.append([int(t) for t in tmp])
                elif line_split[0] == 'list_mult_infec':
                        tmp = sub('[][,\n]', '', line_split[1]).split(' ')
                        tab_mult_infec.append([int(t) for t in tmp])
```

La fonction graph_builder présente dans le code permet de générer le graphique dont les paramètres sont expliqués dans le code.

Ce qui aurait pu être implémenté est le fait de pouvoir choisir les données qu'on veut ajouter dans la moyenne. Dans le cas de toPlot, c'est une liste qui indique uniquement les données qu'on veut afficher. Mais comme je n'ai pas eu besoin de choisir mes données à sélectionner dans la moyenne, je n'ai pas rajouté cette option.

La fonction histogram_builder permet de construire un histogramme dont les paramètres sont spécifiés dans le code. La remarque pour cette fonction est identique à la fonction précédente. La couleur bleue est choisie pour toutes les données puisque la provenance de chaque exécution n'est pas pertinente.

La fonction heatmap_builder permet de construire une carte de chaleur et dont les paramètres sont expliqués dans le code. Une fois de plus, la remarque est la même que pour la fonction graph_builder.

A la fin du code, il y a toutes les exécutions de chaque graphique pour chaque paramètre modifié. Par exemple, pour la durée d'épidémie, il y a donc trois graphiques. Un total de neuf graphiques puisqu'on a la durée d'épidémie, le nombre maximal d'infections et le nombre de multi-infections. Une donnée qui n'est pas intéressante peut être mis en commentaire.

## Conclusion

Nous avons réalisé ici une campagne expérimentale consistant à observer l'impact des différents paramètres pouvant affecter la durée d'un virus, le nombre d'infectés maximal et le nombre de multi-infections. On a réalisé une collecte de données, une lecture des informations présentes et la représentation des informations sous forme de graphique. Les codes ne sont pas expliqués puisqu'ils sont similaires aux codes présentés dans le cours. Lorsqu'il est nécessaire d'expliquer une ligne de code, un commentaire est disponible.