



UNIVERSITÉ DE LIÈGE

Introduction aux Processus Stochastiques

Projet Chaîne de Markov en temps discret

Analyse de la propagation d'un virus informatique par chaîne de
Markov

BOUSTANI MEHDI
ALBASHITYALSHAIER ABDELKADER

Table des matières

0	Introduction	2
1	Modèle(s) exact(s) à deux serveurs	3
1.1	Question 1	3
1.2	Question 2	4
1.3	Question 3	4
1.4	Question 4	5
1.5	Question 5	7
1.6	Question 6	9
2	Simulations au niveau des serveurs et seuil d'épidémie	11
2.1	Question 1	11
2.2	Question 2	12
2.3	Question 3	12
2.4	Question 4	14
3	Simulations macroscopiques	16
3.1	Question 1	16
3.2	Question 2	16
3.3	Question 3	17
4	Simulation à plus grande échelle	20
4.1	Question 1	20
4.2	Question 2	21
4.3	Question 3	22
4.4	Question 4	22

0. Introduction

Dans le cadre de ce projet, notre objectif est de répondre méthodiquement aux questions posées dans l'énoncé. Certaines de ces questions nécessitent la création de programmes pour effectuer des calculs ou afficher des graphiques. Pour répondre à ces besoins, nous avons choisi d'utiliser **Python** et **Mathematica**, deux langages de programmation largement utilisés dans le domaine de l'analyse de données et de la modélisation. Notre approche sera systématique, abordant chaque question dans l'ordre spécifié, en utilisant les outils appropriés pour chaque tâche.

1. Modèle(s) exact(s) à deux serveurs

Dans cette section, nous étudions le modèle exact dans le cas où il n'y a que deux serveurs, connectés entre eux. Initialement, on suppose que l'un des deux serveurs est infecté et l'autre est vulnérable, les deux serveurs ayant la même probabilité d'être infectés.

1.1. Question 1

Ce modèle est effectivement un processus de Markov, car pour chaque serveur, la probabilité de transition d'un état à un temps t vers un autre état à un temps $t + 1$ ne dépend que de l'état présent (c'est-à-dire au temps t), et non pas de l'historique des états.

Dans ce modèle ayant $N = 2$ serveurs, nous avons une chaîne de $3^N = 3^2 = 9$ états possibles. Ces états sont : **IV**, **VI**, **IP**, **PI**, **VP**, **PV**, **PP**, **II**, **VV**. Nous représentons ci-dessous le graphe de transition associé à cette chaîne¹.

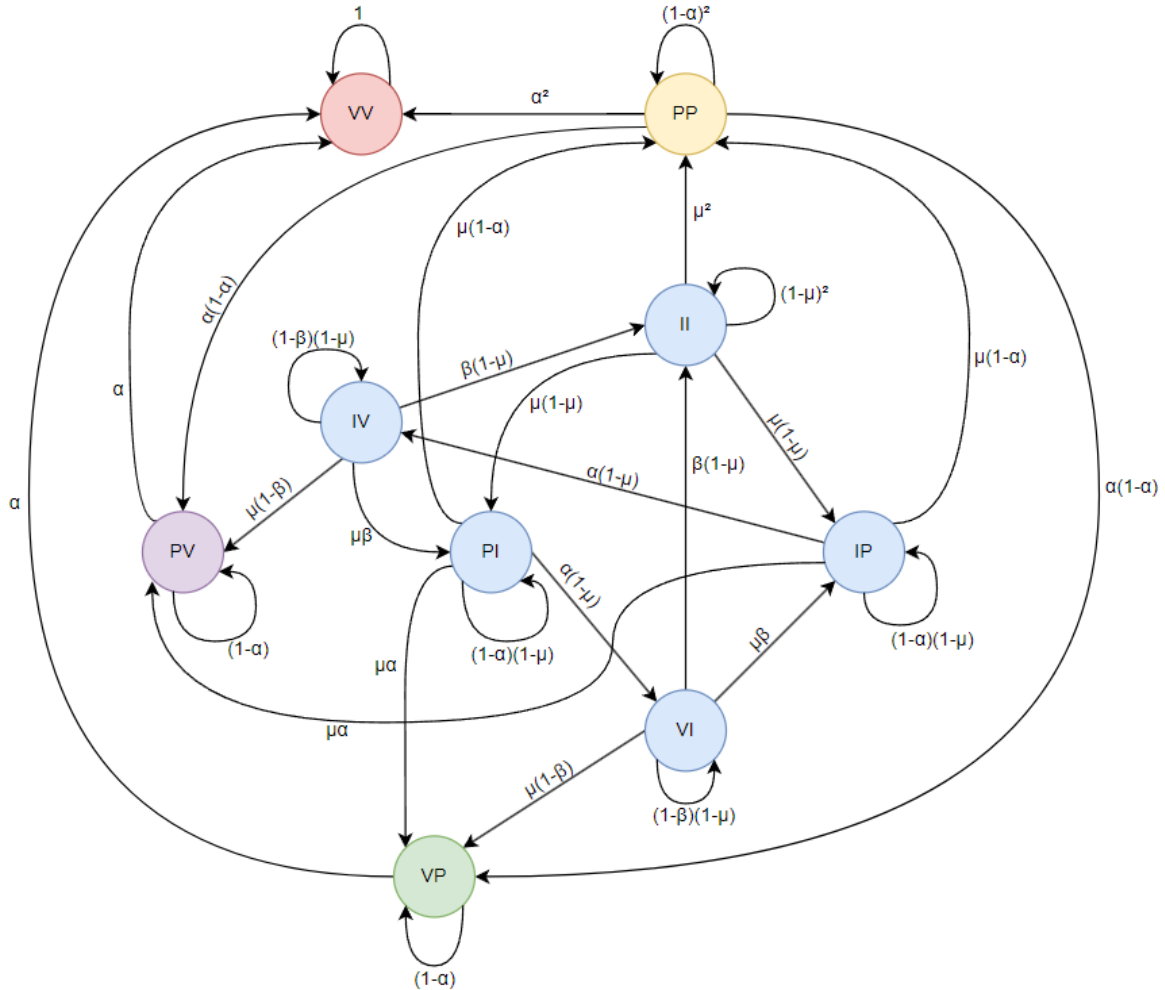


FIGURE 1 – Graphe de transition : Modèle à deux serveurs

1. Dans le cas de VP, PV, PP et VV, β vaut 0 car il n'y a pas de serveur infectieux et donc pas de danger sur le réseau

1.2. Question 2

Commençons par classer les états. Nous avons les classes $\{\text{VI}, \text{IV}, \text{IP}, \text{PI}, \text{II}\}$, $\{\text{VP}\}$, $\{\text{PV}\}$, et $\{\text{PP}\}$ qui sont des classes de passage, et la classe $\{\text{VV}\}$ qui est une classe finale. Par conséquent, les états VI, IV, IP, PI, II, VP, PV, PP sont transitoires, tandis que l'état VV est persistant.

Aperiodicité : Une chaîne de Markov est aperiodique si la période de chaque état vaut 1. La période d'un état est définie comme étant le PGCD de toutes les longueurs des chemins fermés partant de cet état et retournant à lui-même. Dans ce cas, étant donné que les transitions sont possibles entre tous les états de la chaîne et qu'il n'y a pas de cycle avec une longueur commune à tous les états, la chaîne est donc **apériodique**.

Irréductibilité : Une chaîne de Markov est irréductible si tous les états sont accessibles les uns des autres avec une probabilité positive. Cependant, si elle possède plusieurs classes, ce critère n'est pas respecté. Dans notre cas, l'existence d'une classe isolée, comme l'état **VV** d'où aucune transition n'est possible, montre que la chaîne **n'est pas irréductible**.

Régularité : Une chaîne de Markov est régulière si, pour un certain nombre n de pas de temps suffisamment grand, toutes les transitions entre les états deviennent possibles avec une probabilité non nulle. La chaîne que nous examinons **n'est pas régulière**, car à partir de l'état **VV**, on ne peut atteindre aucun autre état.

Absorbance : Une chaîne de Markov est absorbante si une fois qu'elle entre dans un certain état, elle ne peut plus quitter cet état. Cette chaîne est **absorbante**, car elle contient un état absorbant **VV** et il est possible d'atteindre cet état absorbant depuis tout état non-absorbant de la chaîne.

1.3. Question 3

Après avoir substitué les paramètres par leurs valeurs respectives, nous obtenons la matrice de transition suivante, qui décrit les probabilités de transition entre les différentes catégories d'états (Vulnérable, Infecté, Protégé) dans le réseau :

$$\begin{array}{c}
 \begin{array}{c} IV \\ VI \\ IP \\ PI \\ VP \\ PV \\ PP \\ II \\ VV \end{array}
 \begin{pmatrix}
 IV & VI & IP & PI & VP & PV & PP & II & VV \\
 0.45 & 0 & 0 & 0.05 & 0 & 0.05 & 0 & 0.45 & 0 \\
 0 & 0.45 & 0.05 & 0 & 0.05 & 0 & 0 & 0.45 & 0 \\
 0.045 & 0 & 0.855 & 0 & 0 & 0.005 & 0.095 & 0 & 0 \\
 0 & 0.045 & 0 & 0.855 & 0.005 & 0 & 0.095 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0.95 & 0 & 0 & 0 & 0.05 \\
 0 & 0 & 0 & 0 & 0 & 0.95 & 0 & 0 & 0.05 \\
 0 & 0 & 0 & 0 & 0.0475 & 0.0475 & 0.9025 & 0 & 0.0025 \\
 0 & 0 & 0.09 & 0.09 & 0 & 0 & 0.01 & 0.81 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{pmatrix}
 \end{array}$$

En démarrant avec une distribution initiale de $(0.5, 0.5, 0, 0, 0, 0, 0, 0, 0)$ représentant les états VI ou IV, nous avons calculé les probabilités de chaque catégorie d'état à chaque étape de la simulation. Après un nombre suffisant d'itérations, les probabilités associées à chaque catégorie ont convergé vers des valeurs stables : $(0, 0, 0, 0, 0, 0, 0, 0, 1)$. Ce qui est cohérent étant donné que l'état VV est absorbant.

Ces probabilités stables correspondent à un nombre moyen de serveurs dans chaque catégorie de $(2, 0, 0)$, avec arrondis. Ces résultats sont illustrés dans les graphiques suivants (Fichier : question1.3.py) :

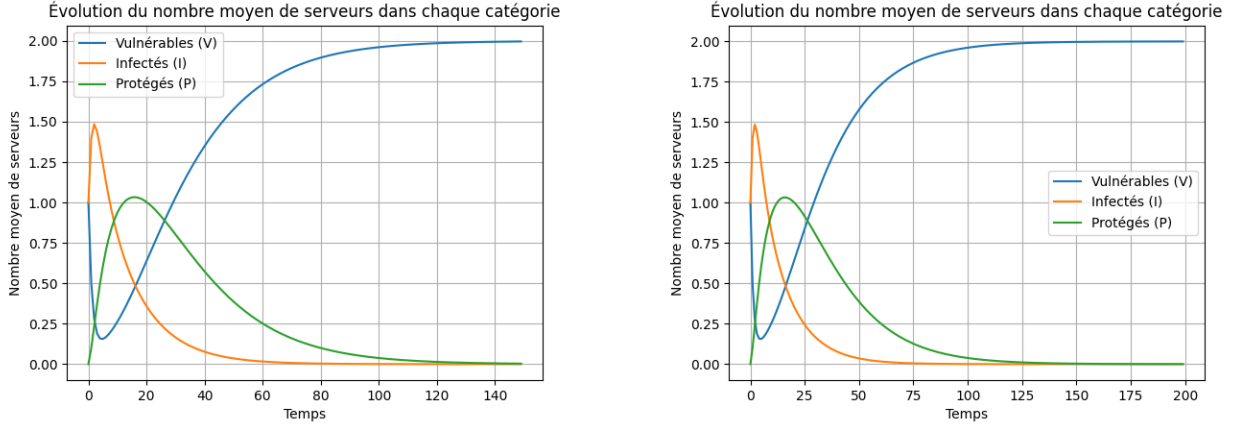


FIGURE 2 – Évolution du nombre moyen de serveurs dans chaque catégorie, avec $\beta = 0.5$, $\mu = 0.1$, $\alpha = 0.05$

1.4. Question 4

Soit A l'ensemble des états dépourvus d'infections.

Nous allons calculer le temps moyen nécessaire à la disparition totale du virus, c'est-à-dire lorsqu'aucun serveur n'est infectieux, correspondant aux états suivants : **VP, PV, PP, VV**. Pour cela, nous cherchons $k_{IV}^A = \mathbb{E}_{IV}[H_A]$ avec $A = \{VP, PV, PP, VV\}$. Cela revient à résoudre le système d'équations suivant :

$$\begin{cases} k_{IV}^A = 1 + \frac{9}{20}k_{IV}^A + \frac{1}{20}k_{PI}^A + \frac{9}{20}k_{II}^A \\ k_{PI}^A = 1 + \frac{171}{200}k_{PI}^A + \frac{9}{200}k_{VI}^A \\ k_{II}^A = 1 + \frac{81}{100}k_{II}^A + \frac{9}{100}k_{IP}^A + \frac{9}{100}k_{PI}^A \\ k_{VI}^A = 1 + \frac{9}{20}k_{VI}^A + \frac{9}{20}k_{II}^A + \frac{1}{20}k_{IP}^A \\ k_{IP}^A = 1 + \frac{171}{200}k_{IP}^A + \frac{9}{200}k_{IV}^A \end{cases}$$

En résolvant ce système, nous aurons la solution : $\{k_{IV}^A = 16.5433, k_{PI}^A = 12.0307, k_{II}^A = 16.6606, k_{VI}^A = 16.5433, k_{IP}^A = 12.0307\}$.

Nous obtenons $k_{IV}^A = k_{VI}^A = 16.5433$. Il faudra donc en moyenne 16.5433 pas de temps afin de se débarrasser complètement du virus dans notre réseau à deux serveurs, que nous soyons partis de VI ou IV.

Lorsque nous réduisons la valeur de β , le temps nécessaire pour éradiquer le virus diminue. Par exemple, avec $\beta = 0.1$ (les autres paramètres étant fixés), le temps moyen s'établit à 13.2805. Cette diminution de la probabilité d'infection facilite ainsi l'élimination du virus dans un délai plus court. En revanche, ce temps s'accroît avec une valeur plus élevée de β : il atteint 17.3563 avec $\beta = 0.9$.

Le paramètre μ exerce un impact significatif sur ce temps. En optant pour une valeur plus faible tout en maintenant les autres paramètres constants ($\mu = 0.05$), le temps nécessaire augmente à 37.8469. Cela s'explique par le prolongement du délai moyen requis pour la détection et l'éradication du virus par l'antivirus d'un serveur ($\mu^2 = 20$), ce qui prolonge la période de

vulnérabilité des serveurs infectés et favorise leur propagation. À l'inverse, une valeur plus élevée de μ réduit le temps (=10.4823 avec $\mu = 0.15$ et 4.95592 avec $\mu = 0.3$), mettant en évidence l'impact majeur de ce paramètre sur la durée nécessaire pour éradiquer complètement le virus.

En augmentant α , le nombre moyen de pas de temps requis pour que la protection d'un serveur soit levée (α^{-1}) diminue, raccourcissant ainsi la durée de protection et favorisant le retour à l'état vulnérable. Ceci est particulièrement pertinent lorsque le serveur est connecté à d'autres serveurs infectieux.

Nous illustrons ces résultats dans les graphes suivants (Fichier : question1.4.nb) :

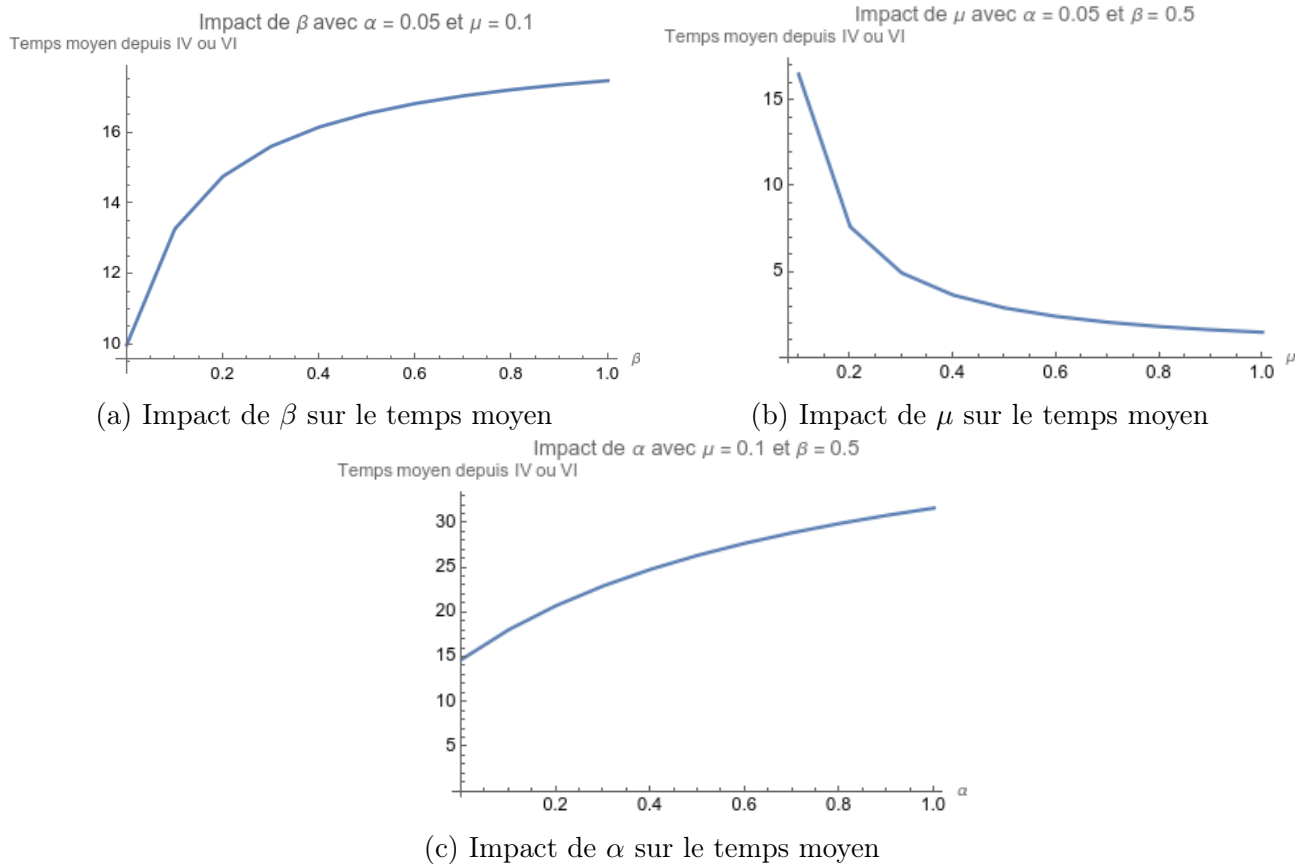


FIGURE 3 – Impact des paramètres sur le temps moyen nécessaire à la disparition du virus

1.5. Question 5

- (a) Dans le cas où le premier serveur est connecté avec l'extérieur, nous avons deux possibilités de transition :
1. Le deuxième serveur est infectieux : Dans ce cas, le premier serveur aura une probabilité $= \beta + \delta$ d'être infecté ou $1 - \beta - \delta$ d'être dans le même état (vulnérable).
 2. Le deuxième serveur est non-infectieux : Dans ce cas, le premier serveur aura une probabilité $= \delta$ d'être infecté ou $1 - \delta$ d'être dans le même état (vulnérable). Car seul l'extérieur pourrait lui rendre infectieux.

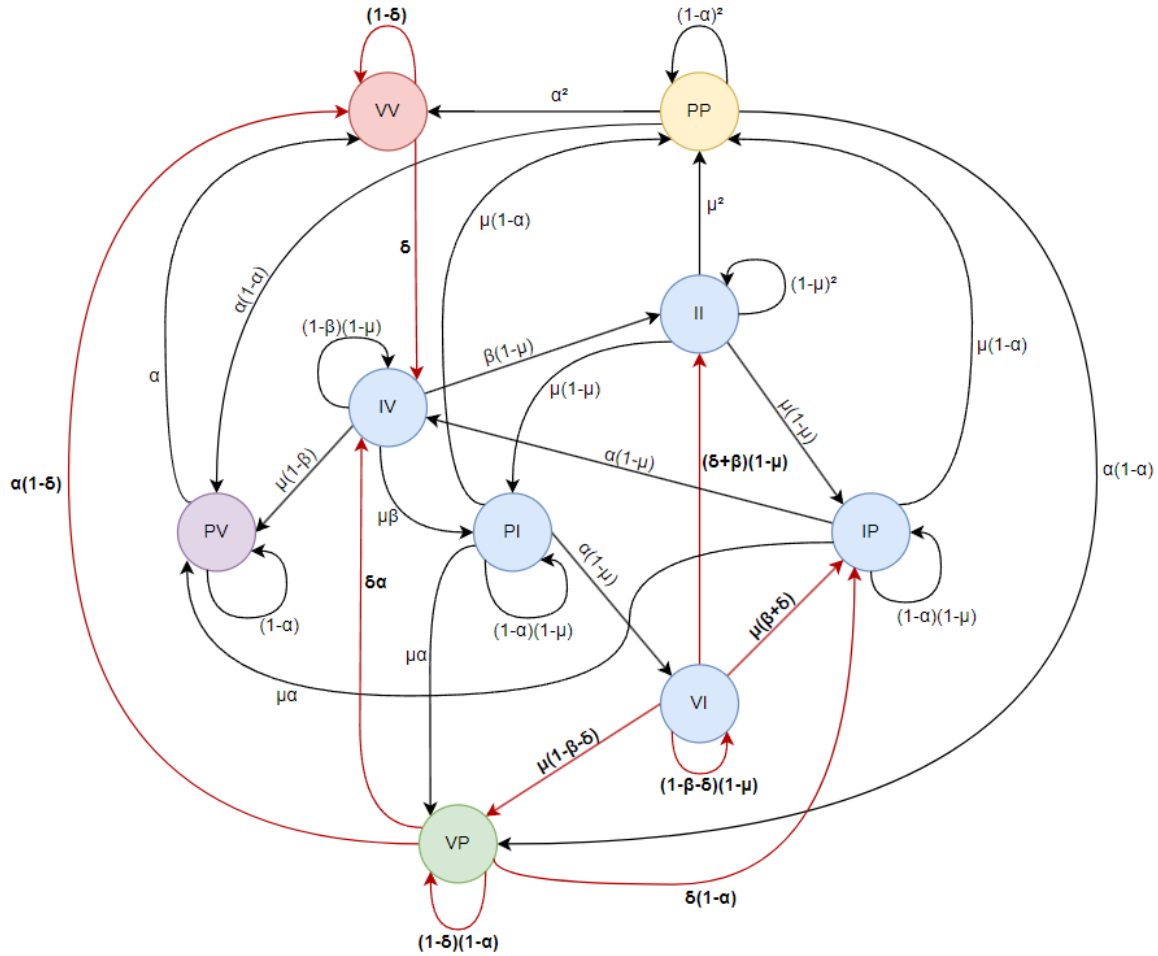


FIGURE 4 – Graphe de transition : Modèle à deux serveurs avec un paramètre δ

Nous étudions une chaîne de Markov comportant une seule classe finale, regroupant les états **IV**, **VI**, **IP**, **PI**, **VP**, **PV**, **PP**, **II**, **VV**. Dans cette configuration, tous les états sont persistants. La chaîne est donc apériodique, car aucun de ses états ne présente de période particulière. Du fait qu'elle ne comporte qu'une seule classe finale, elle est également irréductible, permettant d'atteindre n'importe quel état à partir de n'importe lequel en un nombre fini de transitions. Il convient de noter que l'état **VV** perd son statut d'état absorbant (voir question 1.1). Ainsi, cette chaîne, dépourvue d'états absorbants, ne peut être qualifiée de absorbante.

(b) Ce nouveau paramètre nous permet d'avoir les nouvelles transitions suivantes² :

	<i>IV</i>	<i>VI</i>	<i>IP</i>	<i>PI</i>	<i>VP</i>	<i>PV</i>	<i>PP</i>	<i>II</i>	<i>VV</i>
<i>IV</i>	0.45	0	0	0.05	0	0.05	0	0.45	0
<i>VI</i>	0	0.405	0.055	0	0.045	0	0	0.495	0
<i>IP</i>	0.045	0	0.855	0	0	0.005	0.095	0	0
<i>PI</i>	0	0.045	0	0.855	0.005	0	0.095	0	0
<i>VP</i>	0.0025	0	0.0475	0	0.9025	0	0	0	0.0475
<i>PV</i>	0	0	0	0	0	0.95	0	0	0.05
<i>PP</i>	0	0	0	0	0.0475	0.0475	0.9025	0	0.0025
<i>II</i>	0	0	0.09	0.09	0	0	0.01	0.81	0
<i>VV</i>	0.05	0	0	0	0	0	0	0	0.95

En transition vers le régime stationnaire, avec l'aide de notre programme, nous avons obtenu la distribution stationnaire suivante : $\pi = (0.03290, 0.00514, 0.0860, 0.06804, 0.08357, 0.1930, 0.15950, 0.09134, 0.28039)$. À présent, nous devons calculer le pourcentage de temps passé par chacun des serveurs dans l'état infectieux. Voici les calculs que nous allons effectuer :

- (a) $I_1 = \pi_1 + \pi_3 + \pi_8 \approx 21\%$
- (b) $I_2 = \pi_2 + \pi_4 + \pi_8 \approx 16.4\%$

Ainsi, nous avons tracé le graphe ci-dessous pour étudier le comportement de notre système au cours du temps (Fichier : `question1.5_b.py`).

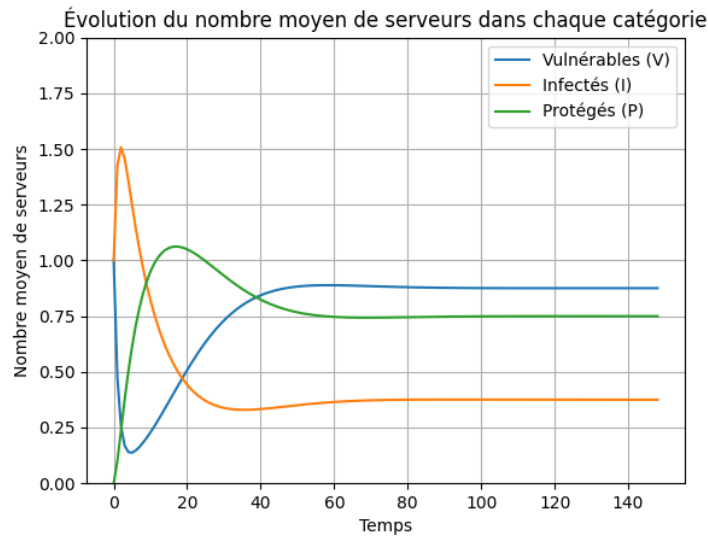


FIGURE 5 – Évolution du temps avec $\beta = 0.05, \delta = 0.05, \mu = 0.1, \alpha = 0.05$

2. Avec $\delta = 0.05, \beta = 0.5, \mu = 0.1, \alpha = 0.05$

1.6. Question 6

(a) Pour la classification des états (V-I-P), nous observons les classes suivantes :

$\{(1-1-0), (0-1-1), (0-2-0)\}$, $\{(1-0-1)\}$, $\{(0-0-2)\}$, qui sont des classes de passage, ainsi que la classe $\{(2-0-0)\}$, qui est une classe finale.

Nous pouvons constater que cette chaîne n'est **pas irréductible**, étant donné qu'elle est composée de plusieurs classes. Cependant, elle demeure **apériodique**. Étant non irréductible et apériodique, elle **ne peut être qualifiée de régulière**. En effet, il n'est pas possible de transiter vers un autre état depuis l'état final (2-0-0). Enfin, cette chaîne est **absorbante** du fait de la présence d'un état absorbant (2-0-0).

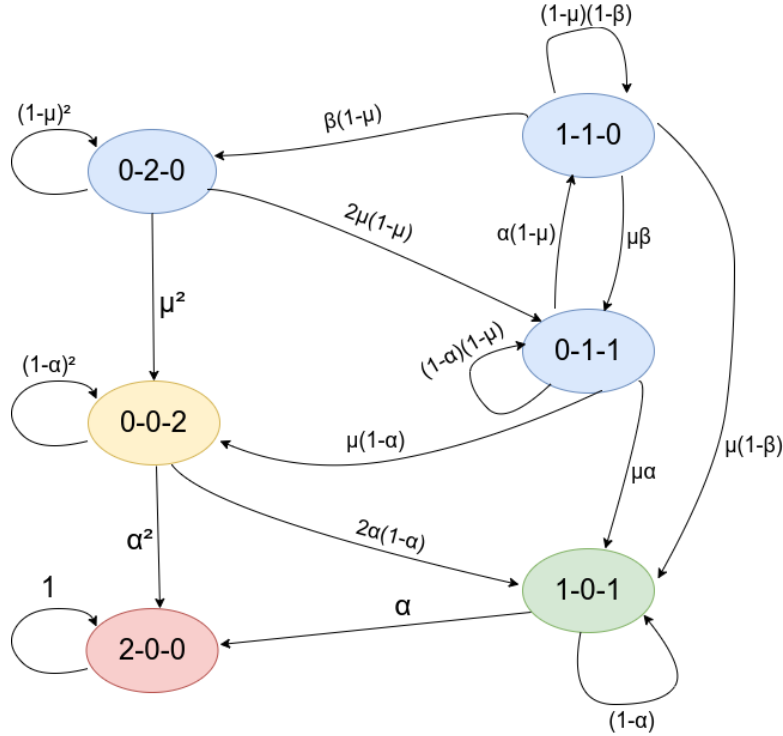


FIGURE 6 – Graphe de transition : Modèle à deux serveurs : V-I-P

(b) 1. **Temps moyen nécessaire avant disparition du virus :**

Les états où aucun serveur n'est infectieux sont : (1-0-1), (0-0-2), (2-0-0). Pour calculer le temps moyen afin d'arriver dans ces états, nous devons calculer $k_{110}^A = \mathbb{E}_{110}[H_A]$ avec $A = \{(1-0-1), (0-0-2), (2-0-0)\}$. Pour ce faire, il suffit de résoudre le système

$$\begin{cases} k_{110}^A = 1 + \frac{9}{20}k_{110}^A + \frac{1}{20}k_{011}^A + \frac{9}{20}k_{020}^A \\ k_{011}^A = 1 + \frac{171}{200}k_{011}^A + \frac{9}{200}k_{110}^A \\ k_{020}^A = 1 + \frac{81}{100}k_{020}^A + \frac{9}{50}k_{011}^A \end{cases}$$

Nous obtenons $\{k_{110}^A = 16.5433, k_{011}^A = 12.0307, k_{020}^A = 16.6606\}$. En partant de notre état initial (1-1-0), nous observons un temps moyen de 16.5433, ce qui correspond au résultat obtenu au point 1.4. Ces résultats confirment ainsi les mêmes impacts des paramètres sur ce temps. Pour obtenir ces valeurs, nous avons résolu le système à l'aide d'un programme développé en *Mathematica*, accessible dans le fichier `question1.6_b.nb`.

2. En reproduisant les expériences des sous-questions 1.3 et 1.4 avec la nouvelle modélisation du système, qui ne prend en compte que le nombre total de serveurs dans chaque catégorie (V_t , I_t , et P_t), et où les serveurs sont indistinguables, nous avons constaté que les résultats obtenus sont identiques. Le graphique caractérisant le nombre moyen de serveurs dans chaque catégorie au fil du temps demeure inchangé, reflétant ainsi la stabilité des résultats malgré le changement de modélisation. (Fichier : `question1.6.py`)

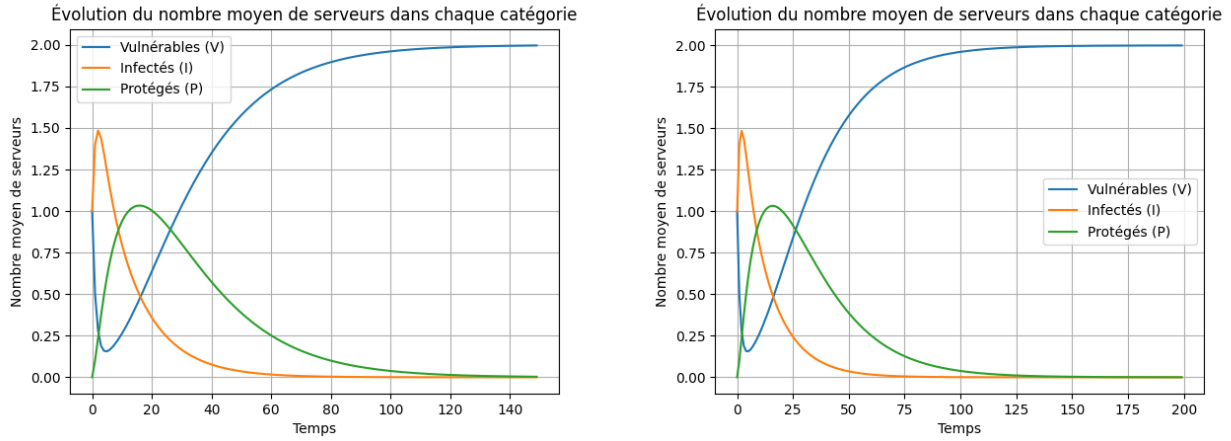


FIGURE 7 – Évolution du nombre moyen de serveurs dans chaque catégorie

2. Simulations au niveau des serveurs et seuil d'épidémie

Dans cette section, nous estimons les mêmes courbes et statistiques que dans la section précédente en se basant sur des simulations de la chaîne de Markov. Pour ce faire, nous allons écrire un programme permettant de générer une réalisation aléatoire de la chaîne, représentée par l'évolution au cours du temps de l'état des N serveurs, étant donné une matrice d'adjacence W modélisant le réseau et des valeurs de β , μ et α fixées à priori.

2.1. Question 1

Nous avons développé un programme en **Python** qui simule l'évolution de notre modèle au fil du temps. Ce programme calcule le nombre de serveurs dans chaque catégorie à chaque pas de temps et produit un graphique illustrant le nombre moyen de serveurs dans chaque catégorie. De plus, il permet de déterminer le temps moyen nécessaire à l'éradication complète du virus, exprimé en nombre de pas de temps.

Nous avons obtenu une moyenne de **16.47258** pour le temps moyen nécessaire à la disparition totale du virus, calculée à partir de plusieurs exécutions du programme. Cette valeur est assez proche du résultat obtenu au point 1.4 de la section précédente.

Grâce à ces simulations de la chaîne de Markov, nous avons pu tracer les graphiques suivants. Il est notable que ces graphiques correspondent bien à ceux présentés au point 1.3 de la section précédente. (Fichier : `question2.1.py`)

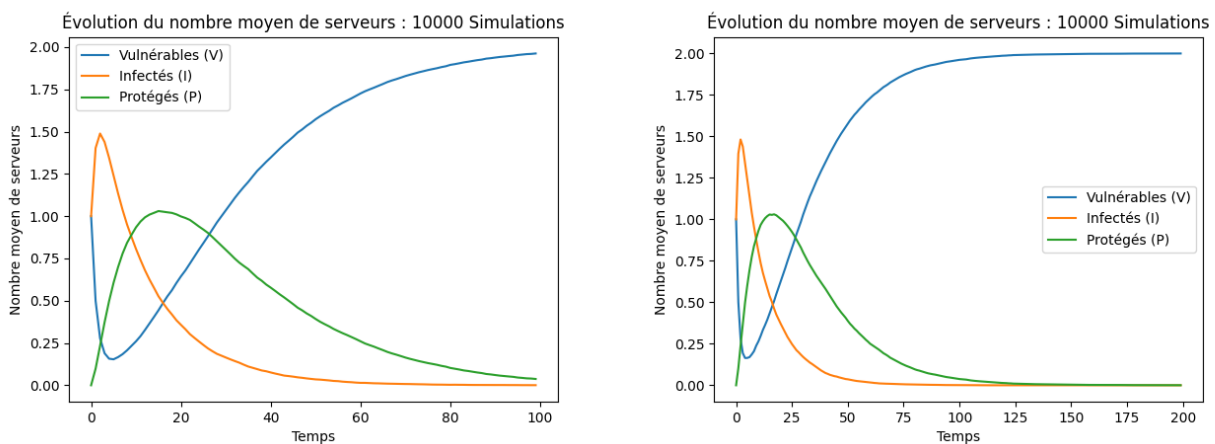


FIGURE 8 – Simulation de l'évolution du nombre moyen de serveurs dans chaque catégorie avec $\beta = 0.5$, $\mu = 0.1$ et $\alpha = 0.05$

2.2. Question 2

Dans le graphique suivant, nous observons une phase de stabilité au niveau des courbes. Pendant cette phase, le nombre moyen de serveurs protégés atteint sa valeur maximale, en raison du paramètre α fixé à 0. Cela signifie que lorsqu'un serveur est protégé, il ne redevient jamais vulnérable. Par conséquent, il ne présente aucun risque d'être infecté et reste protégé tout au long de la simulation.

Nous avons calculé une moyenne de ≈ 24.2033 pour le temps moyen requis pour l'éradication totale du virus, basée sur plusieurs réalisations comprenant 300 simulations chacune. Après ce laps de temps, nous observons la stabilité du nombre de serveurs dans chaque catégorie, avec un certain nombre de serveurs vulnérables maintenant leur statut. (Fichier : `question2.2.py`)

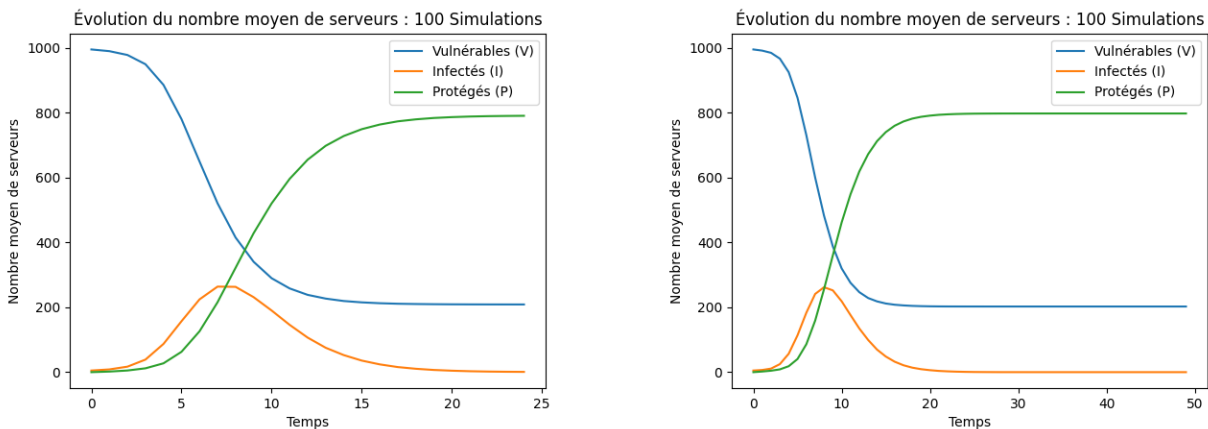
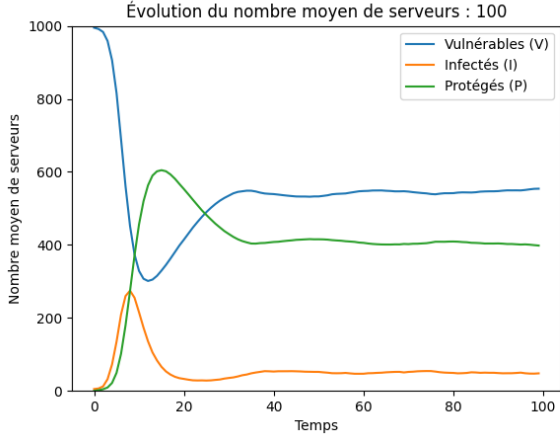


FIGURE 9 – Évolution du nombre moyen de serveurs dans chaque catégorie : $\beta = 0.3$, $\mu = 0.4$ et $\alpha = 0.0$. Réseau : W^{sf} , avec 0.5% des serveurs infectieux initialement

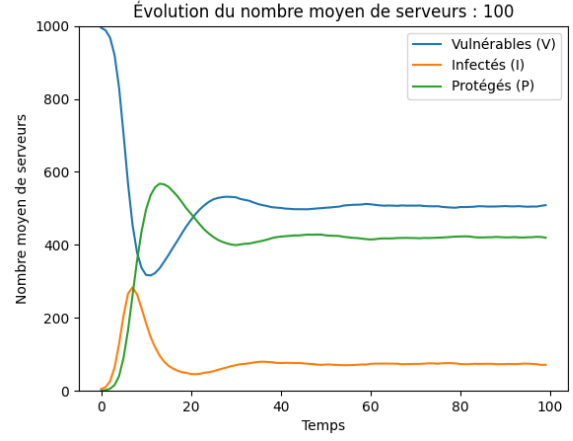
2.3. Question 3

Pour étudier l'impact du paramètre α , nous avons assigné à celui-ci les valeurs croissantes suivantes : 0.05, 0.07, 0.09, 0.1, 0.15, 0.2. En maintenant les autres paramètres inchangés ($\beta = 0.3$ et $\mu = 0.4$), nous avons obtenu les graphiques ci-dessous à partir de 100 simulations de la chaîne de Markov du modèle.

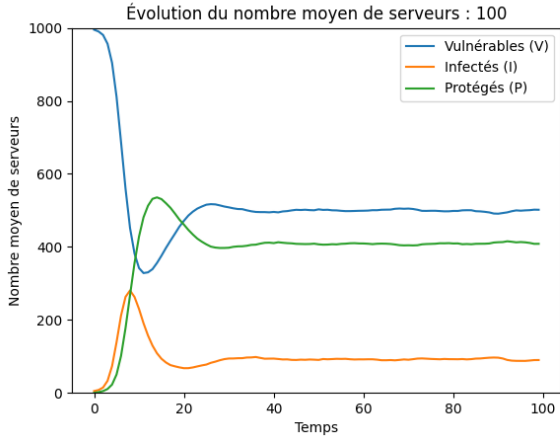
Nous remarquons que le paramètre α exerce une influence significative sur la convergence du nombre de serveurs infectés. En augmentant la valeur de α , le nombre de serveurs infectés tend à s'éloigner de zéro, ce qui se traduit par un écart croissant des courbes vers le haut au cours de la phase de stabilité. Par conséquent, l'éradication totale du virus nécessitera beaucoup plus de temps, voire ne pourra jamais être atteinte. En effet, comme les serveurs protégés ont quand même une certaine probabilité (α) de redevenir vulnérables, il devient de plus en plus compliqué d'éradiquer complètement le virus dans un tel réseau de serveurs, surtout avec une probabilité β qui n'est pas assez faible. Ainsi, nous pouvons conclure que nous avons affaire à une épidémie persistante. (Fichier : `question2.3.py`)



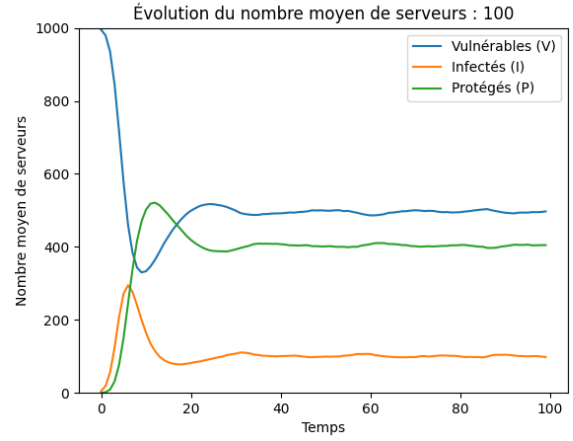
(a) $\beta = 0.3, \mu = 0.4, \alpha = 0.05$



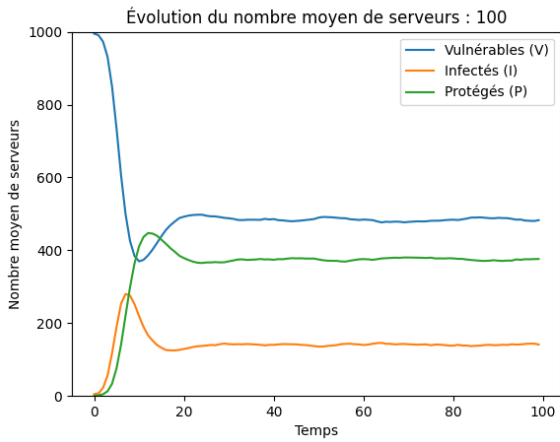
(b) $\beta = 0.3, \mu = 0.4, \alpha = 0.07$



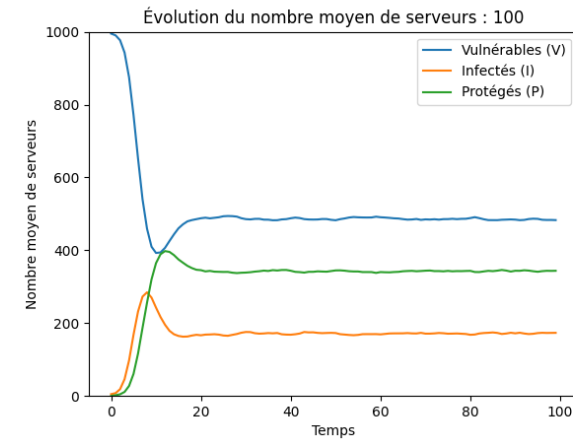
(a) $\beta = 0.3, \mu = 0.4, \alpha = 0.09$



(b) $\beta = 0.3, \mu = 0.4, \alpha = 0.1$



(a) $\beta = 0.3, \mu = 0.4, \alpha = 0.15$



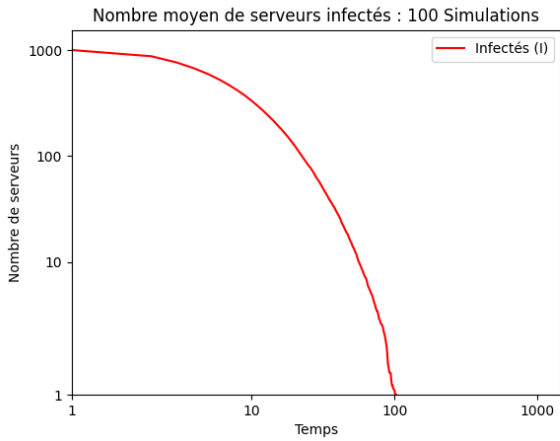
(b) $\beta = 0.3, \mu = 0.4, \alpha = 0.2$

FIGURE 12 – Impact du paramètre α du modèle et la convergence du nombre de serveurs infectés

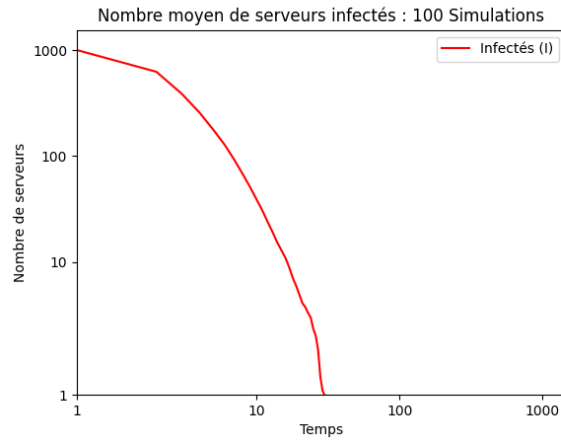
2.4. Question 4

Nous avons adapté notre programme afin d'étudier la convergence des courbes d'infection des serveurs (Fichier : `question2.4.py`).

- ◇ En calculant $\lambda_W^* \approx 10.06$, nous trouvons $\frac{\beta\lambda_W^*}{\mu} \approx 0.8$ (avec $\beta = 0.01, \mu = 0.12575$, et $\beta = 0.03, \mu = 0.37725$). Dans le graphe correspondant ci-dessous, en partant d'un état initial où tous les serveurs sont infectieux, nous constatons une chute rapide du nombre moyen de serveurs infectés dans le réseau. En effet, le rapport $\frac{\beta}{\mu} = 0.08$ est bien inférieur à $\frac{1}{\lambda_W^*} = 0.099$. Selon la théorie, dans ce cas, le nombre de serveurs infectés devrait décroître rapidement vers 0, ce qui peut expliquer le comportement de notre courbe dans le graphique. Finalement, nous pouvons constater qu'il n'y a pas d'épidémie persistante.



(a) $\beta = 0.01, \mu = 0.12575, \alpha = 1$

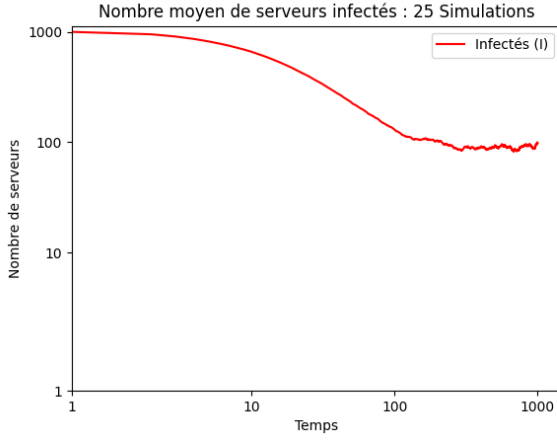


(b) $\beta = 0.03, \mu = 0.37725, \alpha = 1$

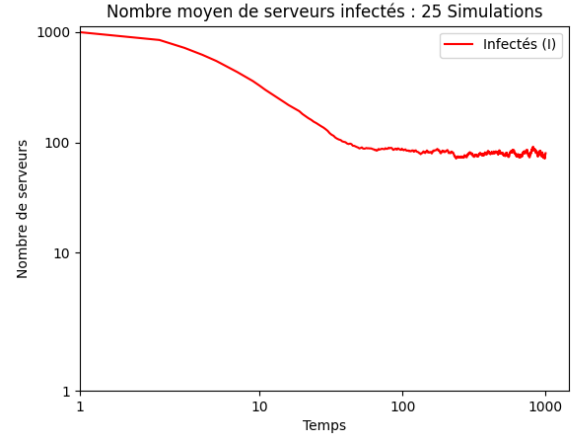
FIGURE 13 – Vérification de l'existence du seuil d'épidémie pour $\frac{\beta\lambda_W^*}{\mu} \approx 0.8$

◇ Contrairement à la situation précédente, après avoir fixé les paramètres $\beta = 0.01$, $\mu = 0.0503$ et $\alpha = 1$ (avec $\lambda_W^* \approx 10.06$), nous obtenons que $\frac{\beta\lambda_W^*}{\mu} \approx 2$. Dans ce scénario, nous constatons que le nombre moyen de serveurs infectés met un temps relativement beaucoup plus long avant de s'annuler. Ce comportement s'explique par le fait que le rapport $\frac{\beta}{\mu} = 0.2$ est supérieur à $\frac{1}{\lambda_W^*} (= 0.099)$. La théorie prédit que lorsqu'un tel cas arrive, le nombre de serveurs infectés mettra un temps très long avant de s'annuler. Ainsi, nous appelons cette situation une **épidémie persistante**.

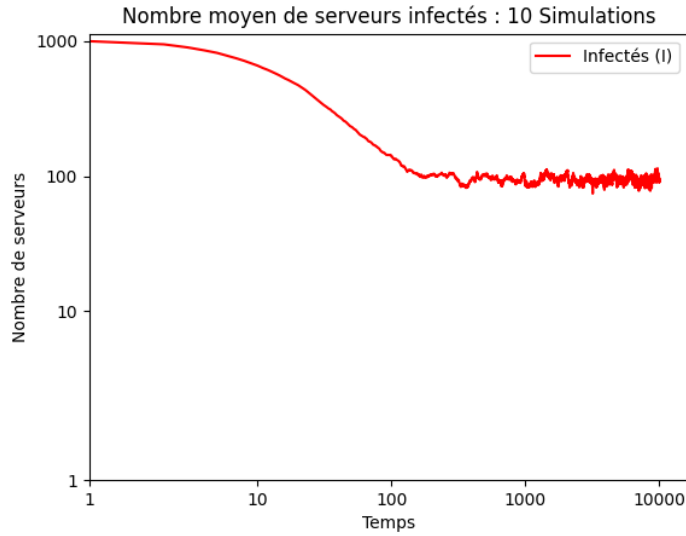
Dès lors, nous constatons que nos résultats sont en accord avec la théorie présentée.



(a) $\beta = 0.01, \mu = 0.0503, \alpha = 1$



(b) $\beta = 0.03, \mu = 0.1509, \alpha = 1$



(c) $\beta = 0.01, \mu = 0.0503, \alpha = 1$

FIGURE 14 – Vérification de l'existence du seuil d'épidémie pour $\frac{\beta\lambda_W^*}{\mu} \approx 2$

3. Simulations macroscopiques

Dans cette section, nous adoptons une approche où nous simulons directement l'évolution du nombre de serveurs dans chaque catégorie, plutôt que de maintenir l'état de tous les serveurs. Cela nous permet de simplifier le traitement des données, notamment dans le cas d'un réseau complètement connecté où les serveurs deviennent indistinguables.

3.1. Question 1

Pour générer de nouvelles valeurs V_{t+1} , I_{t+1} et P_{t+1} , nous mettons à jour le nombre de serveurs dans chaque catégorie en fonction de la transition de chaque serveur vers le prochain état, selon sa probabilité de transition. Étant donné que tous les serveurs sont connectés, nous pouvons facilement déterminer le nombre de serveurs infectés dans le réseau pour calculer la probabilité qu'un serveur passe de l'état vulnérable à l'état infectieux, en utilisant les formules suivantes :

1. La probabilité que le serveur reste vulnérable au temps $t + 1$ alors qu'il est vulnérable au temps t est donnée par $P(S_{t+1} = V | S_t = V) = (1 - \beta)^{I_t}$, où I_t représente le nombre de serveurs infectés au temps t .
2. Par conséquent, la probabilité de passer de l'état vulnérable à l'état infectieux est donnée par $P(S_{t+1} = I | S_t = V) = 1 - (1 - \beta)^{I_t}$.

Ainsi, lorsque le nombre de serveurs infectés au temps t est nul ($I_t = 0$), nous ne devons pas vérifier l'infection des serveurs vulnérables. En effet, étant donné que $P(S_{t+1} = V | S_t = V) = (1 - \beta)^{I_t} = 1$, tous les serveurs vulnérables préservent leur statut. Cette approche pourrait améliorer l'efficacité de la génération des nouvelles valeurs V_{t+1} , I_{t+1} , P_{t+1} .

3.2. Question 2

Après avoir implémenté un programme basé sur la simulation décrite dans la section 3.1, nous avons observé qu'à partir de ce simulateur, nous obtenons une moyenne d'environ 16.6543 pour le temps nécessaire à la disparition du virus. De plus, les graphiques obtenus sont identiques à ceux représentés ci-dessous.

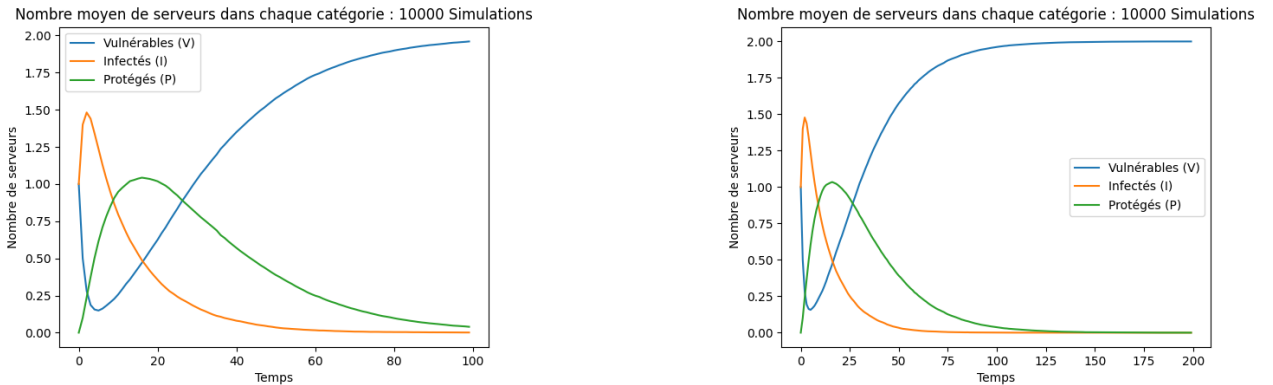


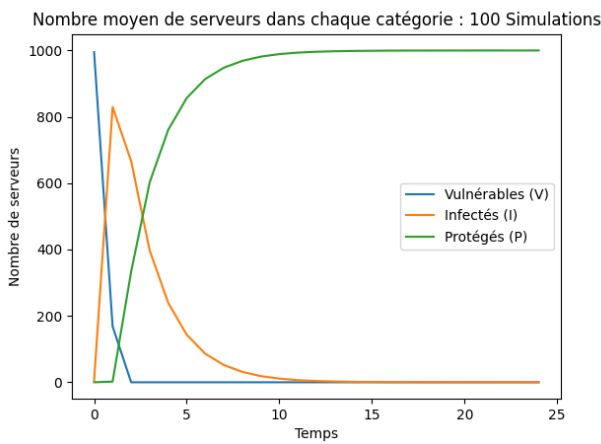
FIGURE 15 – Simulations macroscopiques et l'évolution du nombre moyen de serveurs dans chaque catégorie au cours du temps avec $\beta = 0.5$, $\mu = 0.1$ et $\alpha = 0.05$

En effet, comme il n'y a que deux serveurs connectés, les résultats devraient être identiques par rapport aux simulateurs précédents, car nous sommes dans les mêmes conditions. (Fichier : question3.2.py)

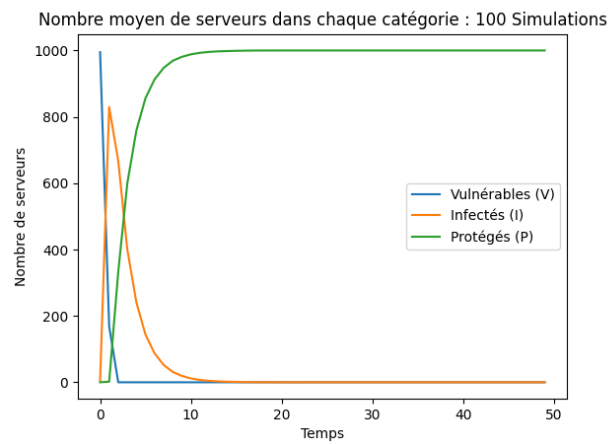
3.3. Question 3

En suivant le même principe (point 3.1), nous avons pu tracer les graphiques suivants dans les mêmes conditions que le simulateur de la section précédente.

1. Initialement, 0.5% des serveurs sont infectés avec $\alpha = 0$. Les graphiques ci-dessous montrent que le nombre de serveurs vulnérables décroît très rapidement pour atteindre 0. Cette observation s'explique par la forte interconnexion entre les serveurs, qui entraîne une augmentation très rapide du nombre de serveurs infectés avant que celui-ci ne commence à diminuer. Ainsi, avec $\alpha = 0$, les serveurs protégés resteront dans cet état indéfiniment (pour toujours), ce qui conduit à une forte diminution des serveurs infectés. En effet, puisque le nombre de serveurs vulnérables ne cesse que de diminuer sans jamais augmenter, la courbe d'infection finit par cesser de décroître. (Fichier : question3.3_a.py)



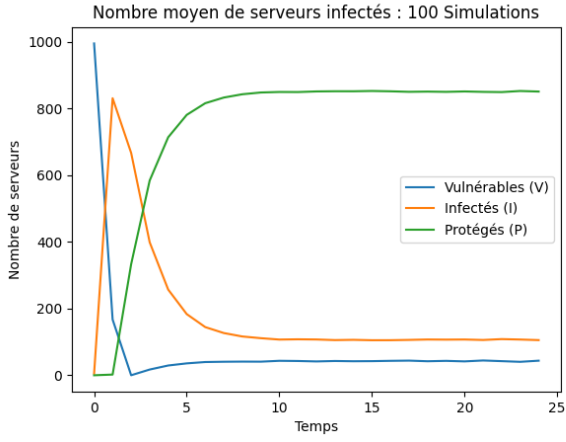
(a) $\beta = 0.3$, $\mu = 0.4$ et $\alpha = 0$



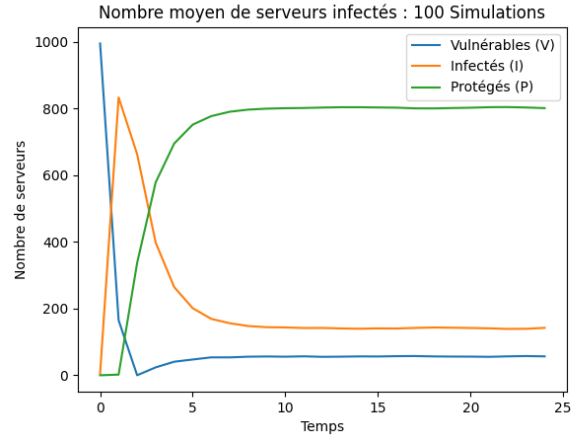
(b) $\beta = 0.3$, $\mu = 0.4$ et $\alpha = 0$

FIGURE 16 – Simulations macroscopiques et l'évolution du nombre moyen de serveurs dans chaque catégorie au cours du temps avec un réseau de 1000 serveurs connectés

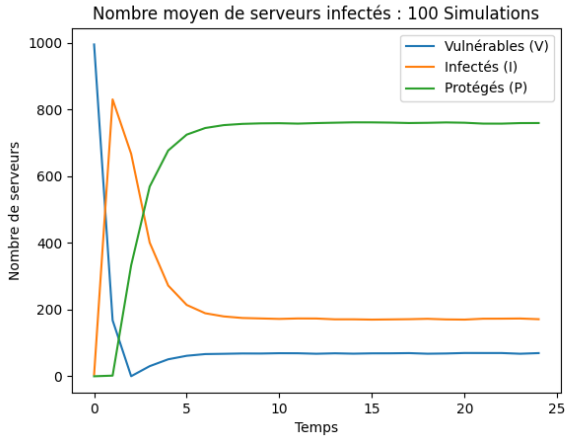
2. Nous allons maintenant tester l'impact du paramètre α sur l'ensemble du réseau en utilisant différentes valeurs faibles et croissantes. En observant les graphiques ci-dessous (Fig 17), nous remarquons que le comportement est similaire à celui observé au point 2.3. Le paramètre α a un impact significatif sur la convergence de la courbe d'infection et sur le nombre d'infections dans le réseau, ainsi que sur le comportement des autres catégories. En effet, nous remarquons également que la courbe d'infection cesse de s'éloigner de sa valeur nulle. (Fichier : question3.3_b.py)



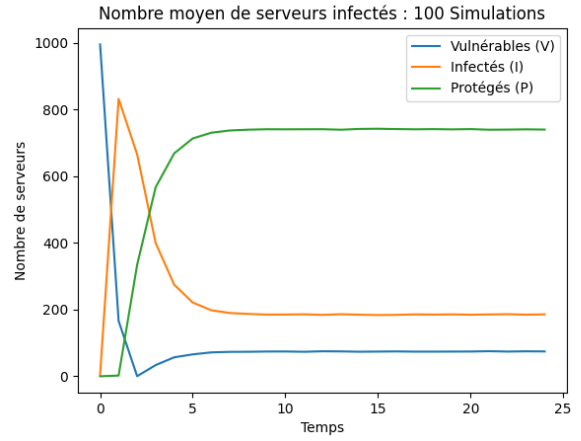
(a) $\beta = 0.3$, $\mu = 0.4$ et $\alpha = 0.05$



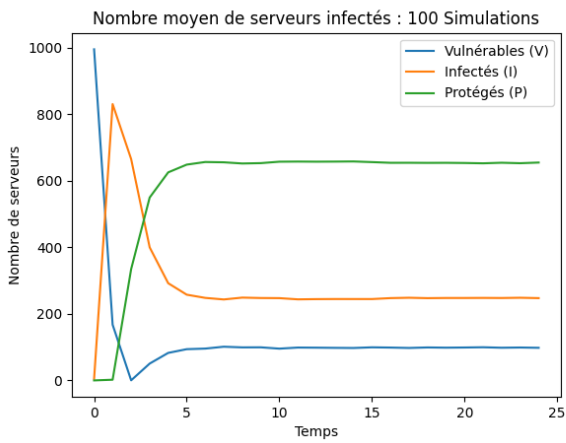
(b) $\beta = 0.3$, $\mu = 0.4$ et $\alpha = 0.07$



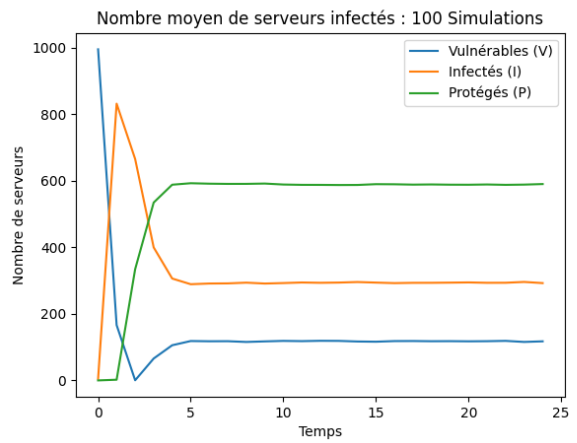
(c) $\beta = 0.3$, $\mu = 0.4$ et $\alpha = 0.09$



(d) $\beta = 0.3$, $\mu = 0.4$ et $\alpha = 0.1$



(e) $\beta = 0.3$, $\mu = 0.4$ et $\alpha = 0.15$



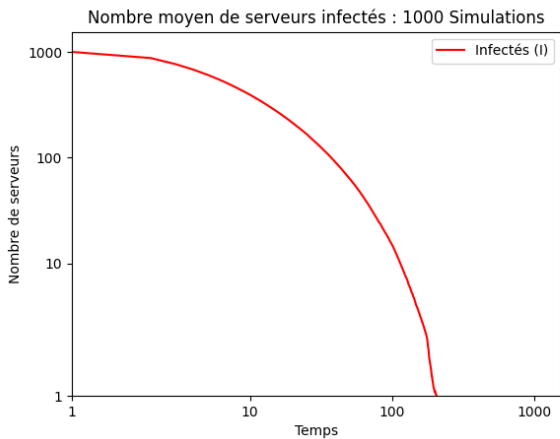
(f) $\beta = 0.3$, $\mu = 0.4$ et $\alpha = 0.2$

FIGURE 17 – Simulations macroscopiques et l'évolution du nombre moyen de serveurs dans chaque catégorie au cours du temps avec un réseau de 1000 serveurs connectés

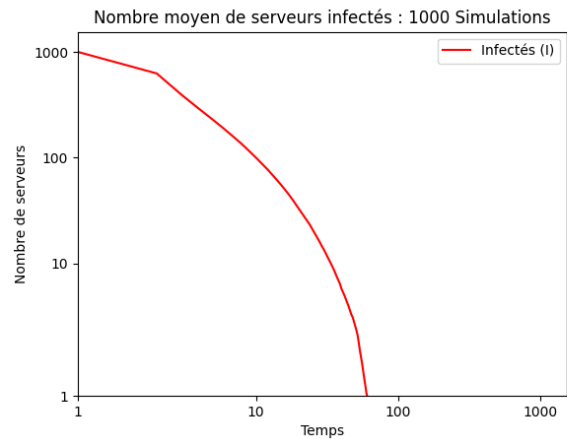
3. Ici, nous allons réaliser la même expérience de la question 2.4, c'est à dire d'étudier le comportement de la courbe d'infections, où initialement, tous les serveurs sont infectés.

Comme la plus grande valeur propre de la matrice de connexion est différente ($\lambda_W^* \approx 999$), nous choisissons différentes valeurs de paramètres. Nous illustrons les résultats obtenus dans les graphiques suivants (Fichier : `question3.3_c.py`) :

- (a) Si le rapport $\frac{\beta\lambda_W^*}{\mu}$ est proche de 0.8, la courbe d'infection présente une décroissance rapide vers 0, comme illustré précédemment. Cette observation concorde avec la théorie exposée dans l'article.

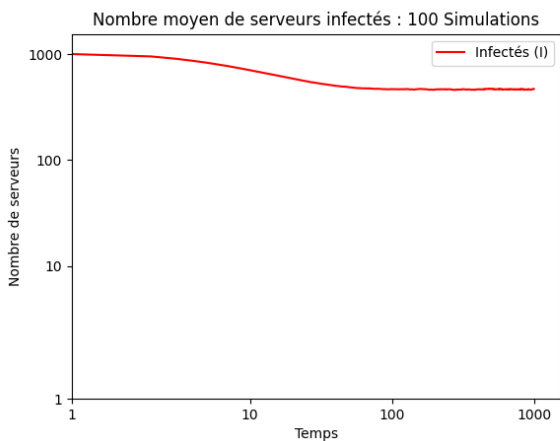


(a) $\beta = 0.0001, \mu = 0.124875, \alpha = 1$

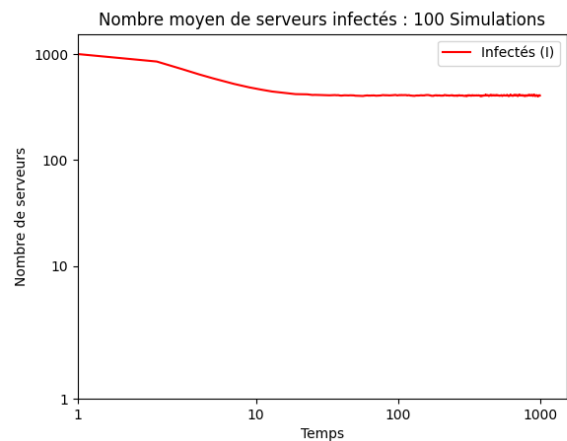


(b) $\beta = 0.0003, \mu = 0.374625, \alpha = 1$

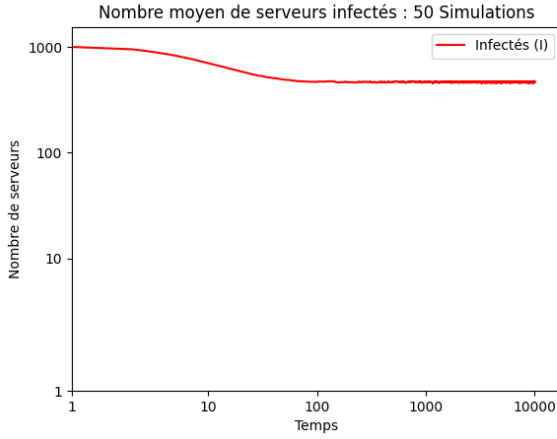
- (b) À l'opposé, lorsque le rapport $\frac{\beta\lambda_W^*}{\mu}$ s'approche de 2, la courbe d'infection présente une décroissance très lente, s'étalant sur une période de temps considérable. Ce qui confirme la prédiction de la théorie.



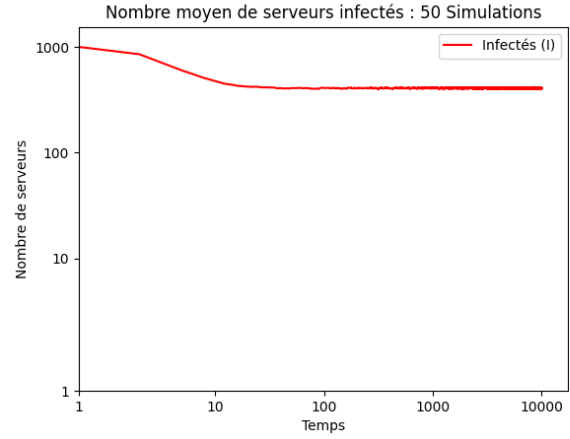
(a) $\beta = 0.0001, \mu = 0.04995, \alpha = 1$



(b) $\beta = 0.0003, \mu = 0.14985, \alpha = 1$



(a) $\beta = 0.0001, \mu = 0.04995, \alpha = 1$



(b) $\beta = 0.0003, \mu = 0.14985, \alpha = 1$

4. Simulation à plus grande échelle

Dans cette dernière section, nous utilisons les outils des sections précédentes pour aborder un scénario un peu plus réaliste : l'introduction d'un nouveau virus au sein d'un réseau connectant différentes entreprises. Notre objectif est d'évaluer la vitesse de propagation ainsi que les dommages potentiels qu'il pourrait causer tant au niveau du réseau dans son ensemble que spécifiquement au sein d'une entreprise.

4.1. Question 1

Afin d'implémenter le modèle de simulation proposé, nous avons combiné les simulations macroscopiques et les simulations dirigées par le graphe. Notre programme consiste à représenter notre réseau à l'aide d'un dictionnaire **Python** qui facilitera l'accès aux différentes données dont on a besoin, chaque élément du dictionnaire représente le numéro de l'entreprise comme clé, et possède comme valeur un dictionnaire contenant deux clés :

1. *connections* : ayant comme valeur une liste des entreprises connectées avec l'entreprise concernée
2. *servers* : ayant comme valeur une liste représentant le nombre de serveurs dans chaque catégorie dans l'entreprise

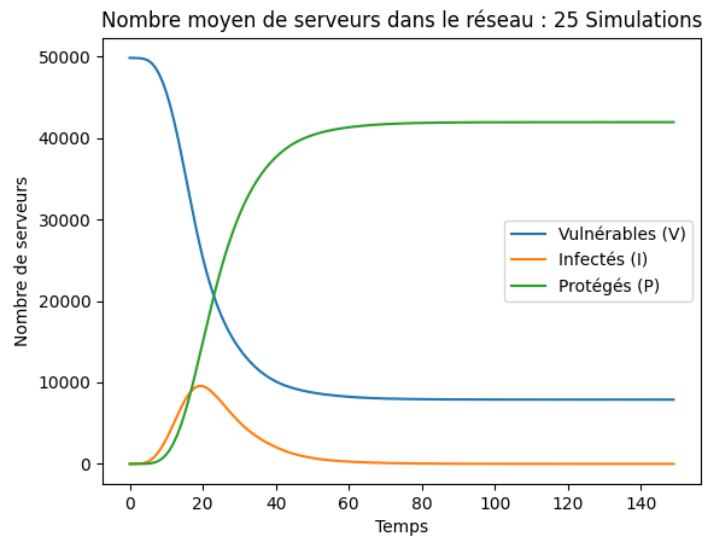
Voici un exemple d'un élément du réseau (entreprise SEGI) : $\{220 : \{"connections" : [92, 296, 752], "servers" : [51, 0, 0]\}\}$

Notre programme consiste à parcourir l'ensemble du réseau à chaque étape, en mettant à jour le nombre de serveurs de chaque catégorie ($V_{t+1}, I_{t+1}, P_{t+1}$) pour chaque entreprise en fonction des différentes valeurs V_t, I_t, P_t de l'entreprise, ainsi que les entreprises en connexion. Pour calculer la probabilité d'infection dans une entreprise, nous cherchons d'abord le nombre d'infections dans la même entreprise (noté $I_{interne}$). Ensuite, nous calculons la somme des nombres d'infections dans les entreprises connectées ($I_{externe}$). (Fichier : `question4.1.py`)

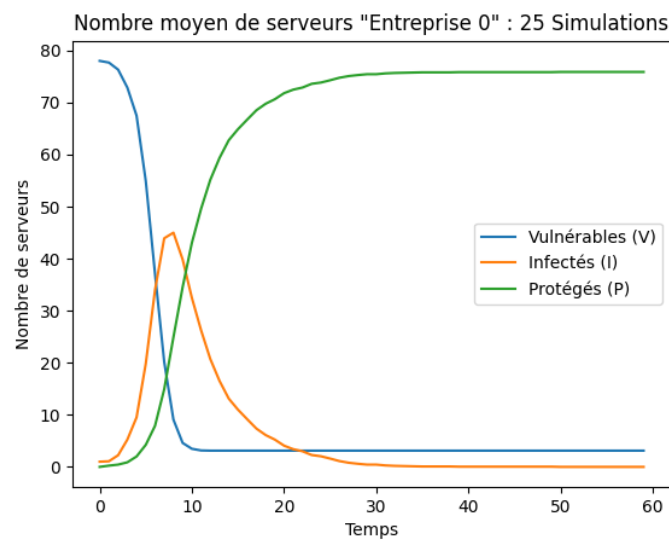
1. $P(S_{t+1} = V | S_t = V) = (1 - \beta_1)^{I_{interne}} * (1 - \beta_2)^{I_{externe}}$.
2. $P(S_{t+1} = I | S_t = V) = 1 - (1 - \beta_1)^{I_{interne}} * (1 - \beta_2)^{I_{externe}}$.

4.2. Question 2

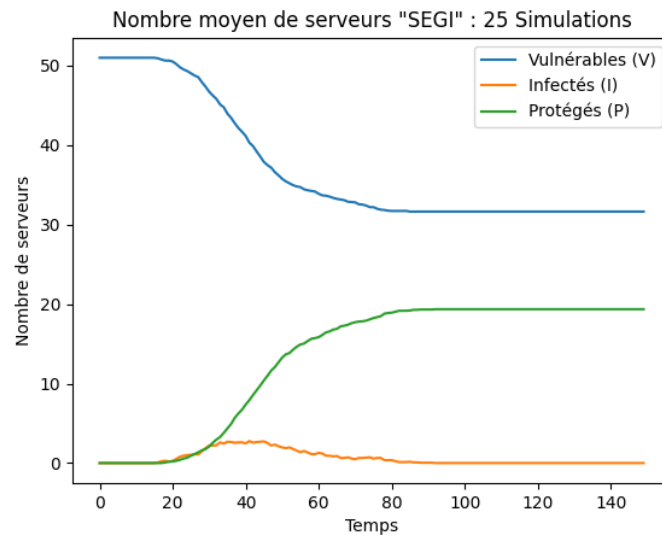
1. **Réseau entier** : En observant l'évolution du temps, nous pouvons constater que le virus terminera par disparaître dans le réseau en l'existence d'une valeur $\mu = 0.2$.



2. **Entreprise 0** : Ici, nous remarquons que le virus touche la moitié des serveurs, et fini par décroître rapidement par la suite et le nombre de serveurs protégés atteint sa valeur maximale.



3. **SEGI** : Nous pouvons constater que SEGI est touché par le virus, où le nombre de serveurs vulnérables diminue à cause du virus. Ainsi, le virus prend un peu plus de temps que l'entreprise 0 avant de disparaître.



4.3. Question 3

Nous trouvons en moyenne une probabilité³ d'avoir au moins 20% de machines infectées simultanément au SEGI de **0.471**. Cette valeur étant trop élevée, elle représente un danger significatif pour le SEGI et le déroulement des délibérations. (Fichier : `question4.3.py`)

4.4. Question 4

Pour calculer la valeur de μ qui permettrait d'améliorer l'antivirus du SEGI, nous avons implémenté un programme permettant d'augmenter⁴ et d'ajuster au fur et à mesure cette valeur jusqu'à atteindre une probabilité de danger⁵ proche de 0, afin d'avoir une valeur optimale de μ . Finalement, nous pouvons choisir une borne inférieure de μ spécifique au SEGI qui pourrait être autour de 0.4. (Fichier : `question4.4.py`)

3. Moyenne sur 10 simulations

4. En augmentant μ SEGI de 0.02 jusqu'à atteindre l'objectif

5. Probabilité d'avoir au moins 20 pourcent de machines infectées simultanément au SEGI