Jeu défense épidémie minimisation des coûts

Règles du jeu :

Jeu basé sur le premier simulateur avec les « honeypots » (intrusion detection systems plutôt), où on étudie comment le défenseur peut minimiser l'attaque en « maximisant » sa défense tout en limitant les coûts.
On rappelle que dans le premier simulateur, plusieurs paramètres sont disponibles pour la défense. L'utilisateur peut :
- modifier la probabilité de changement d'état de i vers s (dans le contexte on peut dire que c'est lorsqu'il y a une intervention d'un technicien pour retirer le virus de l'ordinateur manuellement)
- modifier la probabilité de changement d'état de s vers r (dans le contexte, on peut dire que c'est la probabilité d'installer un antivirus performant sur appareil)
- mettre en place des « honeypots » dits « basiques » <=> surveillance d'une arête tirée aléatoirement, si le virus passe par cette arête quand elle est surveillée alors détection du virus, et le nœud transmetteur et le receveur redeviennent susceptibles et sont libérés du virus. Ils sont dits basiques car ils ne peuvent pas surveiller les arêtes intelligentes, on le détaille au prochain tiret
- mettre en place des « honeypots intelligents » <=> on surveille une arête intelligente, c'est-à-dire une arête qui est liée à un nœud qui a transmis le virus au tour précédent, évidemment, il faut avoir détecté la transmission du virus au tour précédent, autrement c'est comme un honeypot basique.

Le **but du jeu** est alors de savoir **comment le défenseur peut maximiser sa défense tout en minimisant son coût.** Pour évaluer la défense et le coût nous allons utiliser les métriques suivantes :
- évaluation de la défense : on regarde le ***nombre total cumulé d'infectés*** durant l'attaque, c'est-à-dire la somme des infectés à tous les tours. Pour rappel, un tour est un tour de boucle sur les nœuds du graphe durant lequel les nœuds infectés peuvent faire propager le virus et il peut y avoir des changements d'état et des détections par honeypots.
- évaluation du coût : Pour évaluer le coût total de défense, on associe un coût à toutes les opérations de défense listées plus haut, ainsi, il y a un coût pour la ***mise en place de honeypots basiques, la mise en place de honeypots intelligents, pour chaque nœud qui va subir le changement d'état $i \rightarrow s$ et enfin pour chaque nœud qui va subir le changement d'état $s \rightarrow r$.***

On s'intéresse au nombre cumulé d'infectés et pas spécialement au pic épidémique ici, car on considère que minimiser le nombre cumulé d'infectés est plus important si on suppose que l'objectif de l'attaquant est de voler des informations. Ainsi, plus le nombre cumulé d'infectés est élevé, plus l'attaquant dispose de temps où il peut consulter les données des machines. Bien sûr, si le but est d'évaluer une autre perspective, cela peut être changé et on peut s'intéresser à une autre métrique.

Déroulement du jeu :

Au départ, tous les nœuds sont susceptibles sauf un seul nœud qui est infecté, il essaye ensuite de contaminer tous ses voisins, mais il ne les infecte qu'avec une probabilité 1/2. Au départ, il y a quelques tours sans que le changement d'état $s \rightarrow r$ ne soit possible. Car on considère qu'au départ le défenseur ne sait pas qu'il est la cible d'une attaque, il ne met donc pas en place cette solution. Une fois que le nombre d'infectés atteint un certain seuil, ici 5 % du nombre de nœuds du graphe, il y a le début des changements d'état $s \rightarrow r$. Il y a ensuite des tours jusqu'à la fin du jeu, qui advient quand il n'y a plus aucun nœud infecté dans le graphe.

Données :

Pour ce jeu, on génère différentes simulations pour différents paramètres, voici les plages de valeurs des paramètres :

proba i→s : 1/100*[10, 61, 5] (de 10 (inclus) à 61 (exclu) par pas de 5, le tout divisé par 100)
proba s→r : 1/1000*[10, 61, 5]
nombre de hp basiques : [1, 6, 1]
nombre de hp intelligents : [0, 5, 1]

On prend pour cela un graphe de 200 nœuds (« *game_graph.gexf* », présent dans le dossier)

Le fichier *csv* des résultats de la simulation est «*resultats_simu_cout_chgts.csv*» et il contient 3025 données de simulations.
Pour des problèmes de temps de calcul et de temps disponible, on ne fait que trois simulations par uplet de données.

Résultats :

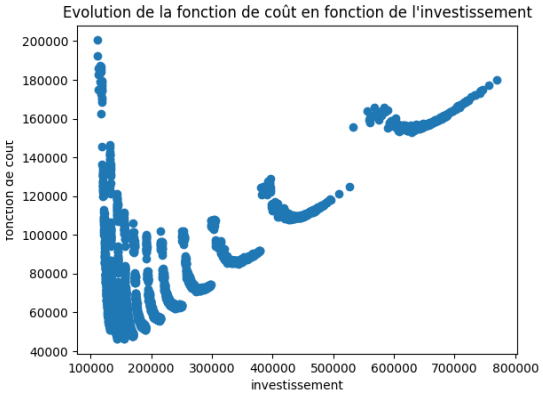
On utilise ces coûts (relativement arbitrairement choisis) pour les différentes méthodes de défense :

Défense	Coût/unité
i→s	47.1
s→r	136.2
Honeypot basique	1002
Honeypot intelligent	2000.5

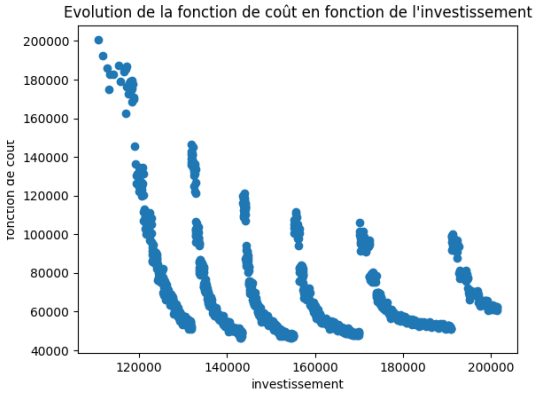
On note *K* le coût total associé à la défense.

Le coût *C* que cherche à minimiser le défenseur est une combinaison du coût de la défense et du nombre cumulé d'infectés :
C = a * K + n_cumul où a est un terme de régularisation, ici *a = max(n_cumul)/max(K)*

Voici les résultats que l'on obtient (La courbe représentée est la courbe C en fonction de K) :



Evolution de la fonction de coût



Zoom

On remarque qu’avec les coûts fixés plus haut, on observe une sorte de courbe convexe, le coût minimal n’est pas forcément obtenu avec les plus gros investissements, mais comme on pouvait s’y attendre il est question de compromis.

Cependant, comme sur la courbe on ne parle que d’investissements en défense, il est difficile de savoir quels sont réellement les dispositifs défensifs qui permettent les meilleurs résultats, on affiche alors les données pour les 15 coûts minimaux, c’est-à-dire les 15 configurations qui permettent d’obtenir les valeurs de C les plus basses.

	proba is	proba sr	nombre hp basiques	nombre hp intelligents	investissement	nombre cumule d'infectés	coût à minimiser
0	0.15	0.060	3	3	117866.4	13702.333333	46263.985325
1	0.20	0.060	3	3	124029.7	11391.000000	46551.034250
2	0.15	0.060	2	4	121525.8	14059.000000	46596.661181
3	0.20	0.060	4	1	122728.8	11477.333333	46676.563195
4	0.20	0.060	1	3	125297.2	11977.000000	46970.327711
5	0.20	0.060	5	3	128465.1	11794.666667	47094.867337
6	0.20	0.060	1	4	131939.4	12117.333333	47111.639796
7	0.20	0.060	3	2	125288.0	11980.000000	47135.937148
8	0.20	0.060	5	2	126172.2	11914.000000	47208.305396
9	0.20	0.055	2	3	131713.2	12478.333333	47216.980615
10	0.20	0.060	4	4	131696.8	12040.666667	47286.193784
11	0.20	0.050	4	4	135616.7	12701.333333	47302.636642
12	0.20	0.060	3	4	131905.4	12139.666667	47305.778285
13	0.20	0.060	5	0	124728.6	12097.666667	47358.694487
14	0.20	0.060	4	3	128847.2	12141.666667	47367.982927

On remarque que les choses les plus importantes à maximiser sont le nombre de honeypots ainsi que la possibilité de faire le changement d’état $s \rightarrow r$.