

Grands Réseaux d'Interaction

TP n° 9 : Propagation épidémique

Remarques importantes :

- Les règles générales en TP restent valides, voir feuille du TP 1. De plus il faut rendre un **rapport** au format PDF exclusivement, à joindre à l'archive (qui elle est au format tar non compressé exclusivement), et vous pouvez joindre quelques petits `.dot`, voir ci-dessous.
- À rendre sur Moodle pour le **4 janvier au soir**, date **ferme**, aucun rendu ne sera accepté après.

Dans ce TP on s'intéresse à la propagation d'épidémies, de croyances etc. dans les réseaux. Voir le poly du cours du 24 novembre. L'objectif est d'implémenter un simulateur pas-à-pas avec différentes variantes des règles de contamination, de guérison et de désimmunisation, et le tester sur différents graphes et différents états initiaux. Il vous est également demandé un rapport, au format PDF, précisant la syntaxe de la ligne de commande, ainsi que ce qui a été implémenté parmi toutes les variantes proposées. En effet quand plusieurs variantes (numérotées) sont proposées, au moins une doit être implémentée, mais il n'est pas nécessaire de toutes les implémenter pour avoir une bonne note. Vous pouvez aussi en rajouter (l'écrire dans le rapport). Le rapport précisera aussi quelques expériences que vous aurez faites sur des graphes donnés, par exemple ceux de `/info/master2/Public/GRI/` (à vous de décider ce qui est pertinent comme paramètres et comme graphes). Donnez la syntaxe exacte utilisée à chaque fois et mettez le graphe dans l'archive s'il est petit (moins de 1000 arêtes). Le temps d'exécution d'une simulation sera pris en compte (il sera évalué sur la machine du correcteur).

a) États et transitions

Un sommet, à un moment donné, est soit Sain (S), soit Infecté (I), soit Résistant (R). Les sommets I sont aussi appelés *contaminés*. Les transitions possibles sont :

- Contamination (C) : transition $S \rightarrow I$
- Guérison (G) : transition $I \rightarrow R$
- Désimmunisation (D) : transition $R \rightarrow S$

b) Graphe de départ

Le graphe sur lequel on effectue la simulation est au format `dot` simplifié du TP 1. Il est passé en ligne de commande. L'orientation doit être ignorée : on traitera les arcs comme des arêtes. En effet la propagation est usuellement étudiée sur des graphes non-orientés.

c) État initial

On peut commencer la simulation de plusieurs façons possibles

1. Tous les sommets de départ sont sains, sauf certains dont le numéro est spécifiés en ligne de commande
2. Tous les sommets de départ sont sains, sauf certains dont le numéro est spécifiés dans un fichier donné en ligne de commande
3. Tous les sommets de départ sont sains, sauf un nombre donné en ligne de commande d'infectés, qui sont tirés au sort au début du programme
4. On donne (en ligne de commande) trois probabilités, dont la somme est 1, qui sont les probabilités que chaque sommet soit, initialement, S I ou R (par exemple 50% de S, 10% de I, 40% de R).

d) Transitions C

Un sommet est contaminé par ses voisins. On peut imaginer les règles de contamination suivantes, qui s'appliquent à chaque étape de simulation :

1. Un sommet S devient contaminé s'il a au moins k voisins contaminés. Pour $k = 2$ c'est la Règle 1 du poly : l'épidémie se propage selon la P_3 -convexité du graphe, voir cours.

2. Un sommet S devient contaminé si un certains pourcentage de ses voisins est contaminé. Par exemple avec au moins 50% de voisins contaminés on devient I. Remarquez que si ce seuil est faible les sommets de petit degré ne seront jamais contaminés (s'il est de 10% et qu'on a degré inférieur à 10, on ne risque rien) mais que s'il est élevé alors l'épidémie ne se propagera sans doute pas, sauf cas particuliers.
3. Un sommet S devient contaminé avec probabilité p_C s'il a au moins un voisin I. p_C est passé en ligne de commande.
4. Pour chaque voisin I d'un sommet S donné, ce sommet S est contaminé avec probabilité p_C (paramètre en ligne de commande). Par exemple si $p_C = 50\%$ et le sommet S a trois voisins infectés, il a 87,5% de chances d'être infecté (et non pas 150%! Les chances se multiplient elles ne s'additionnent pas).

On peut imaginer qu'optionnellement (option en ligne de commande) l'orientation soit prise en compte pour les graphes orientés : l'épidémie se propage en suivant les arcs, c'est-à-dire que «voisin» doit être compris comme signifiant «voisin entrant» dans ce qui précède.

e) Transitions G et D

Ces transitions ne dépendent pas du voisinage d'un sommet. On peut imaginer deux type de transitions :

1. Temporelle : le sommet reste un certain nombre de pas de simulation dans l'état avant de faire la transition. Par exemple on peut être infecté pendant 10 tours, avant de guérir (transition G) on fixe donc un temps $t_G = 10$. Pour la règle D si le temps t_D est de 0 il n'y a pas d'immunisation, c'est-à-dire qu'un sommet qui cesse d'être I redevient immédiatement S et n'est jamais R.
2. Probabiliste. À chaque étape de simulation on a une probabilité p_G de faire la transition G, si on est en état I; et une probabilité p_D de faire la transition D, si on est en état R. Remarquez que si $p_D = 0$ alors on reste immunisé à vie.

Les deux règles sont indépendantes, donc il faut prévoir de lire en ligne de commande soit t_G soit p_G ; et soit t_D soit p_G .

f) Condition d'arrêt

On peut imaginer que le programme

1. soit tourne soit jusqu'à stabilisation (plus aucune transition ne peut se déclencher) ce qui n'arrivera peut-être jamais
2. soit tourne un certain nombre d'étapes, donné en paramètre
3. soit les deux : il tourne *au plus* un certain nombre d'étapes et s'arrête si stabilisation

g) Résultat

Le programme ne doit rien afficher quand il tourne, sauf en cas d'erreur. Lorsqu'il s'arrête il doit afficher des informations statistiques concises, telles que le nombre de tours faits, le nombre de sommets en état S I et R, de contaminations effectuées etc. À vous de déterminer les informations les plus pertinentes à afficher. **Si une option est donnée** en ligne de commande et dans ce cas seulement, le programme doit écrire **dans un fichier** texte, à chaque étape, l'état de chaque sommet sous la forme

Tour 1 : 1S 2I 3R 4S 5R

c'est à dire un tour par ligne et sur chaque ligne, le numéro de chaque sommet et son état. Cela permet de vérifier la correction de votre programme et il est donc indispensable de l'implémenter. Mais cela doit être une option de ligne de commande, car sinon l'utilisation sur une longue durée ou sur de gros graphes serait impossible.