

TD1 : Python et analyse de réseaux dans le domaine du risque
Exploration de la « vulnérabilité combinatoire » au sein d'un graphe

Ce TD s'appuie sur les données « data.py » : <http://sergelhomme.fr/data/data.py>

Partie 1 : Quantification du nombre de relations rendues impossibles par la perturbation d'un nœud.

- 1) Créez un graphe G contenant les arcs de la liste arcs_w.
- 2) Récupérez les arcs adjacents au nœud 15 à l'aide d'une fonction du type : list(G.edges(x)).
- 3) Attribuez un poids arbitraire de 1 000 000 à ces arcs adjacents qui seront ainsi considérés impraticables. Pour modifier la pondération d'un arc, il est possible d'utiliser la commande suivante : G[x][y]['weight'].
- 4) Calculez les distances des plus courts chemins entre tous les nœuds du graphe G.
- 5) En parcourant toutes les distances obtenues, comptez le nombre de plus courts chemins ayant une distance supérieure ou égale à 1 000 000 (c'est-à-dire le nombre de « chemins impossibles »). Affichez le résultat final.
- 6) Généralisez la séquence précédente afin de quantifier pour chaque nœud (et pas seulement pour le nœud 15) le nombre de relations rendues impossibles par leur perturbation. Stockez les résultats dans une liste.

Partie 2 : Visualisation de l'impact produit par les perturbations successives de plusieurs nœuds.

- 1) Reprenez la séquence précédente afin de perturber uniquement le nœud 0, puis « ajoutez » à cette première perturbation, la perturbation du nœud 1. A chaque itération, affichez le résultat.
- 2) Faites de même pour les 10 premiers nœuds, c'est-à-dire du nœud 0 au nœud 9. Stockez les résultats de chaque itération dans une liste sous la forme de taux (relations impossibles / relations possibles avant perturbation).
- 3) Représentez cette liste dans un graphique comportant en abscisse le nombre de nœuds perturbés (ou les taux correspondants) et ordonnée les taux de relations impossibles. Ce graphique sera composé de points reliés entre eux par des segments de droite.

Partie 3 : Comparaison de stratégies visant à créer les plus forts impacts produits par les perturbations successives de plusieurs nœuds.

- 1) Calculez la centralité intermédiaire (betweenness) de chaque nœud (sans tenir compte des distances), puis triez les nœuds par ordre décroissant de leur centralité. Utilisez ce classement pour « perturber » successivement les 10 nœuds qui présentent les plus fortes centralités. Comparez cette courbe d'impact à la courbe d'impact de la stratégie précédente.
- 2) Faites de même avec l'indicateur de proximité (closeness).
- 3) Il est aussi possible d'étudier comment a réagi le réseau aux perturbations successives subies et de recalculer à chaque itération l'indicateur utilisé afin de prendre en considération les reconfigurations induites. Par exemple, après chaque perturbation, recalculez les centralités (en utilisant les poids modifiés) et perturbez ainsi à chaque itération le nœud le plus central. Comparez cette stratégie aux stratégies précédentes.
- 4) Enfin, il est possible de remplacer la stratégie de la partie 2 par des tirages aléatoires qui détermineront différentes listes de dix nœuds à perturber. La moyenne des impacts obtenus pour chaque itération tendra vers le résultat attendu pour un tirage aléatoire. La bibliothèque « random » propose pour cela une fonction « shuffle ».