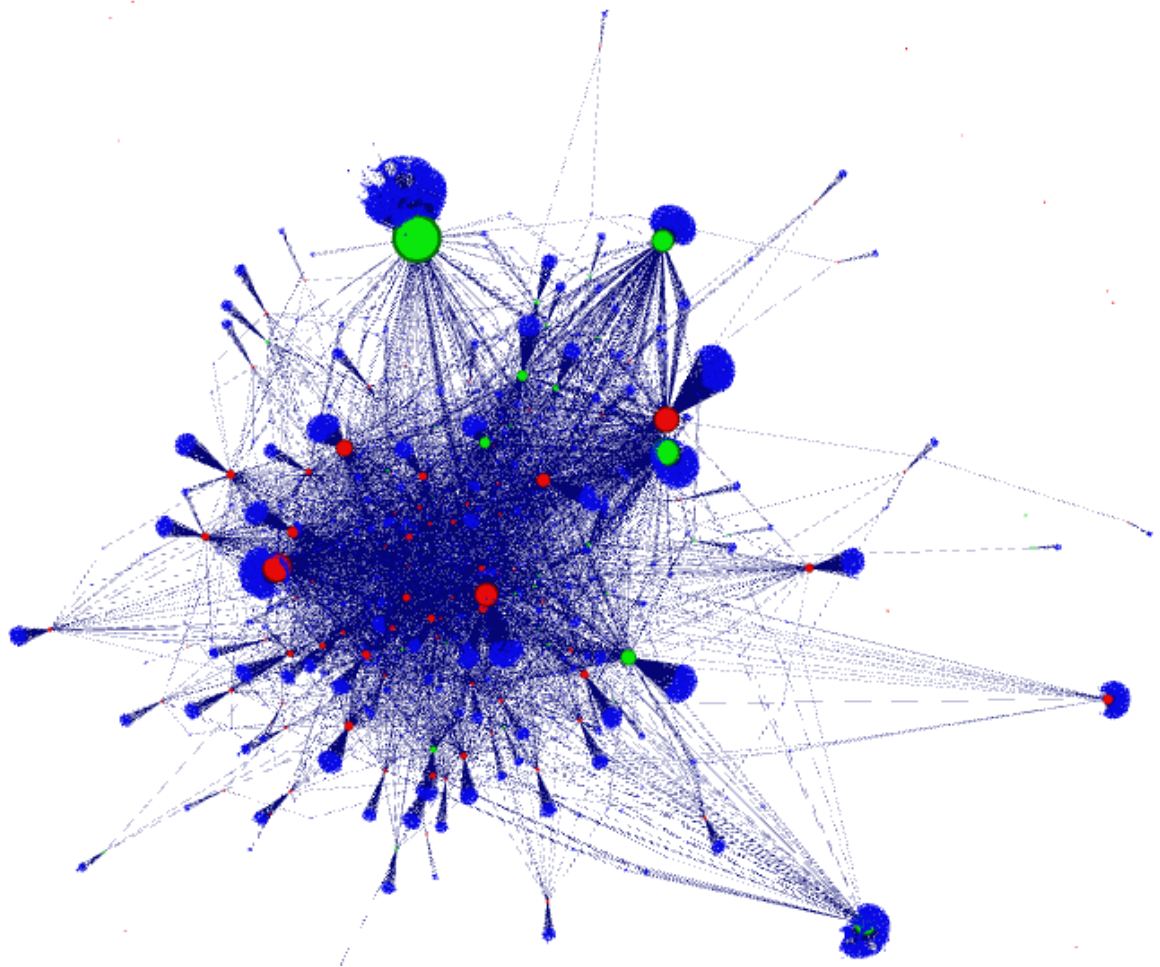

DM - Graphes

Calcul de flot maximal sur réseau « changeant »

Mouad Tahtaoui - Inas Bekkouche - 11 décembre 2020



Introduction

Ce devoir, réalisé en binôme par Mouad Tahtaoui et Inas Bekkouche, a pour but d'implémenter une classe Python FlowNet permettant de :

- > créer un réseau à partir de sa description dans un fichier,
- > créer un réseau à partir d'un graphe orienté NetworkX dans lequel les capacités sont des attributs
- capacity des arcs,
- > obtenir le flot maximal (à l'aide d'un simple accesseur),
- > mettre à jour la capacité d'un arc (1 seul arc à la fois pourra être modifié),
- > afficher le réseau,
- > enregistrer le réseau sous forme d'image,
- > calculer le flot maximal.

La réalisation de ce devoir n'était pas si simple. Pour ce faire, nous avons consacré beaucoup de temps, durant lequel, nous avons pu identifier et comprendre les aspects du sujet. A cause de la situation actuelle difficile nous étions obligés de se réunir en visioconférence, l'organisation et la discussion de nos résultats étaient des problèmes majeurs.

En outre que ce rapport vous trouverez un fichier nommé « fig42.csv » ainsi qu'un programme Python qui prendra en argument :

- > le nom d'un fichier décrivant un réseau, en l'occurrence ici le fichier « fig42.csv »
 - > trois arguments pour effectuer une modification du réseau (point A, point B, capacité),
- et qui produira dans le terminal : le flot maximal avant et après modification et un fichier image avec le réseau après modification de la capacité.

Questions :

1 - Un Flot maximal est un flux saturé (maximum) entre l'entrée et la sortie d'un graphe pondéré. Sa valeur correspond au flux sortant à l'entrée du graphe (source) qui égale aussi au flux entrant à la sortie.

2 - Sans application de l'algorithme on peut calculer le flot maximal , on commence tout d'abord par saturer les chemins entre s et t , on remarque qu'il n'existe plus de

chemin reliant s à t lorsque les arcs $a-c$, $c-d$, et $d-t$ sont saturés . ON peut dire que le flot maximal est de 23.

3 - Les informations à stocker pour modéliser un réseau sont : les capacités de l'arc et les quantités de flot sur l'arc, les sommets et les arcs.

4 - Un chemin augmentant est un chemin simple de s à t (sans répétition de sommets), sa variation de flot est la valeur minimale, le long du chemin des capacités de l'arc moins les quantités de flot sur l'arc pour les arcs augmentant et des quantités de flot sur l'arc pour les arcs diminuant et permet d'améliorer le chemin du départ.

5 - Un réseau résiduel est un chemin d'arc direct et inverse qui indiquent la quantité de flots disponible sur notre graphe, vu que 2 flots de direction opposée s'annulent. Il sert à améliorer le flot de la source vers le puits

6 - Les algorithmes qui calculent le flot maximal sur Networkx sont :

- > Edmonds-Karp ($G, s, t, capacity='capacity', residual=None, value_only=False, cutoff=None$)
- > `maximum_flow(flowG, _s, _t, capacity='capacity', flow_func=None, **kwargs)`
- > `maximum_flow_value(flowG, _s, _t, capacity='capacity', flow_func=None, **kwargs)`
- > `shortest_augmenting_path(G, s, t, capacity='capacity', residual=None, value_only=False, two_phase=False, cutoff=None)`
- > `boykov_kolmogorov(G, s, t, capacity='capacity', residual=None, value_only=False, cutoff=None)`
- > `dinitz(G, s, t, capacity='capacity', residual=None, value_only=False, cutoff=None)`.

7 - Pour recalculer le flot maximal lorsqu'on augmente la capacité d'un arc on a utilisé la fonction déjà définie dans NetworkX (`shortest_augmenting_path`) qui prend en paramètre les deux points et la nouvelle capacité entre ces deux points, et renvoie le réseau résiduel résultant du calcul du flot maximal, et on récupère la nouvelle valeur de notre flot stockée dans `R.graph['flow_value']`

8 - La complexité de notre algorithme est la même capacité que notre fonction définie (`shortest_augmenting_path`) : $O(n^2m)$, n (nbre de sommets) et m (nbre d'arcs).