

# TP 12 : Réseaux complexes

## Cours de modélisation numérique

26 mai 2023

### Introduction

Vous avez vu en cours que les graphes sont de puissants outils permettant de construire un modèle simple pour une grande variété de phénomènes différents. On peut citer par exemple les systèmes de distribution électrique, les réseaux de transport routier ou encore différents processus biologiques comme le métabolisme.

Ces modèles, basés sur des graphes, permettent de décrire comment les structures émergent et de quelles manières elles évoluent. Idéalement le générateur de graphes utilisé doit être capable de produire un graphe singeant les propriétés du phénomène étudié. Vous implémenterez et étudierez dans cette série d'exercices un générateur de graphes de type *scale-free*.

### Le modèle de Barabási Albert

Le modèle de Barabási Albert, proposé par Albert-László Barabási, est un modèle générant des graphes de type *scale-free* permettant de reproduire l'évolution de la croissance de nombreux réseaux réels, comme l'internet (Fig. 1) et le WWW ou encore les réseaux sociaux tel le réseau social sous-jacent à Facebook.

La construction peut se décomposer en deux étapes : la croissance et l'attachement préférentiel. La croissance implique que de nouveaux sommets sont ajoutés au graphe. L'attachement préférentiel implique que les nouveaux sommets ont une plus grande probabilité d'établir des liens avec les sommets ayant un degré élevé. On parle de *l'effet Matthieu* en référence à l'évangile selon Matthieu.

« Car on donnera à celui qui a, et il sera dans l'abondance, mais à celui qui n'a pas on ôtera même ce qu'il a. »

– Matthieu 25:29

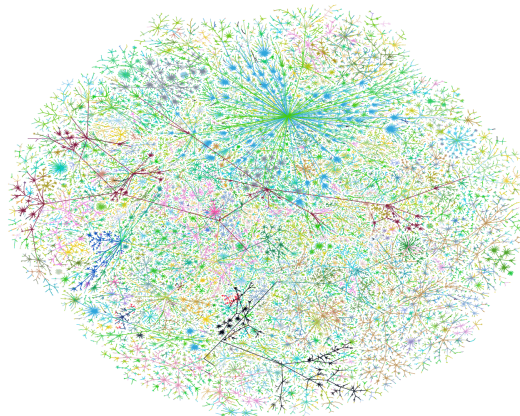


FIGURE 1 – L'internet (en date du 29 juin 1999)

Les étapes de l'algorithme s'enchaînent de la manière suivante avec comme **situation initiale** un graphe  $G = (V, E)$  composé de  $n_0$  sommets ( $n_0 \geq 2$ ) ayant chacun un degré au moins égal à 1. Ensuite les étapes suivantes se répètent pour  $t$  pas de temps :

**Croissance** : à chaque pas de temps, un nouveau sommet est ajouté au graphe avec  $n$  arcs qui le connectent à  $n$  différents sommets déjà présents dans le graphe.

**Attachement préférentiel** : le choix des sommets à connecter au nouveau sommet dépend de leur degré. Le nouveau sommet est connecté à un des sommets existants  $i$  avec la probabilité  $P_i$  suivante :

$$P_i = \frac{k_i}{\sum_j k_j} = \frac{k_i}{2|E|}$$

où  $k_i$  est le degré du sommet  $i$ .

Après avoir répété ces étapes pendant  $t$  itérations, le graphe obtenu est composé de  $t + n_0$  sommets et  $nt$  arcs.

Les graphes générés par ce modèle possèdent, entre autres, les propriétés suivantes :

— la distribution  $P_k$  des degrés  $k$  suit une distribution en loi de puissance :

$$P_k \propto k^{-\gamma} \quad (1)$$

— l'*average path length*  $l_G$  est proportionnel à :

$$l_G \propto \frac{\ln(n)}{\ln(\ln(n))}, \quad (2)$$

où  $n$  est le nombre de sommets du graphe  $G$ . Cette mesure définit le nombre moyen de pas qu'il faut pour relier toutes les paires de sommets du graphe le long des chemins les plus courts.

## Travail à faire

Vous trouverez sur Moodle un début de script **Python** visant à implémenter le modèle de Barabási Albert, et produisant de jolis graphes en sortie. Pour produire les graphes, vous aurez besoin du package **networkx** de **Python**, que vous pouvez installer en tapant la commande `pip install networkx` dans votre environnement **Python**.

Nous vous demandons de compléter le code afin de pouvoir générer les graphes. Pour vous aider dans la compréhension du formalisme utilisé dans le script, la méthode d'Erdős-Rényi a été implémentée.

Une fois que vous pouvez générer des graphes avec la méthode de Barabási Albert, vous pouvez vérifier que les graphes générés possèdent bien la propriété de la distribution des degrés (1). La propriété (2) est laissée comme bonus. Pour cela, vous devrez construire des graphes assez grands, mesurer chaque grandeur *numériquement*, et la comparer avec la valeur calculée *analytiquement* par les formules ci-dessus.