Modélisation de parcours du Web et calcul de communautés par émergence

Bennouas Toufik

ACADÉMIE DE MONTPELLIER

UNIVERSITÉ MONTPELLIER II

— **S**CIENCES ET **T**ECHNIQUE DU **L**ANGUEDOC —

THÈSE

présentée à l'Université des Sciences et Techniques du Languedoc
pour obtenir le diplôme de doctorat

SPÉCIALITÉ	: **INFORMATIQUE**
Formation Doctorale	: **Informatique**
École Doctorale	: **Information, Structures, Systèmes**

MODÉLISATION DE PARCOURS DU WEB

ET

CALCUL DE COMMUNAUTÉS PAR ÉMERGENCE

par

Toufik BENNOUAS

Soutenue le 16 décembre 2005 devant le jury composé de :

M. Olivier COGIS, professeur, LIRMM, Montpellier . . . . . president
M. Fabien DE MONTGOLFIER, maître de conférence, LIAFA, Paris . . . . . examinateur
M. Michel HABIB, professeur, LIAFA, Paris . . . . . directeur de thèse
M. Jean-Claude KÖNIG, professeur, LIRMM, Montpellier . . . . . examinateur
M. Michel MORVAN, professeur, LIP-ENS, Lyon . . . . . rapporteur
M. Laurent VIENNOT, Chargé de recherche, Gyroweb, INRIA Rocquencourt . . . . . rapporteur

Table des matières

Introduction

Depuis l'année 1989, date à laquelle le **Web** fut créé, ce dernier n'a cessé de croître au point d'être aujourd'hui estimé à plusieurs dizaines de milliard de pages Web [1]. Le réseau du Web n'est pas le seul réseau de cette importance, on rencontre dans différents domaines des réseaux d'interactions de très grande taille.

Des récentes études ont montré qu'en modélisant ces réseaux par des graphes, on observait des propriétés communes malgré leurs origines diverses (Web, Internet, Sociologie, Chimie, etc.). La connaissance des propriétés des **réseaux d'interactions** est nécessaire afin de prévoir leurs évolutions et de déterminer leurs capacités à résister à différents phénomènes ou tout simplement de comprendre leur nature. En effet, il est important de savoir si le réseau Internet est robuste aux attaques et aux pannes ou encore comment une maladie peut se propager dans un réseau social, etc.

La modélisation d'un crawl du Web

Pour mener ces études, il est nécessaire de se procurer le réseau à étudier. Malheureusement, cela est impossible pour la majorité des réseaux d'interactions. Généralement, on dispose d'un échantillon obtenu en utilisant différents outils. Le **crawler** est le programme utilisé pour construire un sous-graphe du Web. Lors de l'utilisation de tels outils deux problèmes apparaissent : le premier est de savoir si l'échantillon obtenu est assez grand pour regrouper toutes les propriétés du réseau étudié et le second est de s'assurer que les outils utilisés lors de la construction de l'échantillon ne provoque pas l'ajout d'autres propriétés non existantes dans le grand réseau (un artefact).

Devant la difficulté d'obtenir des réseaux réels, une solution consiste à générer sur machine des graphes avec les mêmes propriétés que les échantillons obtenus : la **modélisation**. Nous nous sommes intéressés aux problèmes de la modélisation de crawls du Web et nous proposons dans ce mémoire un nouveau modèle de **crawls aléatoires** basé sur des **simulations** de parcours de graphes. Le but était de regarder quelles propriétés connues du graphe du Web pouvaient être imputées au seul processus de crawl, au lieu d'être intrinsèques au Web lui-même.

Calcul des communautés par émergence

L'une des propriétés communes des réseaux d'interaction est la présence de zones très denses en liens, généralement appelées **communautés**. En particulier, le Web contient un grand nombre de communautés, c'est-à-dire qu'il existe dans le Web un ensemble de pages très liées les unes aux autres que l'on peut interpréter comme étant des pages qui traitent d'un thème bien précis. Donc, il est possible de voir le Web comme un ensemble de communautés où chaque communauté traite un sujet particulier. Cette vue est très intéressante pour les concepteurs de moteurs

1. http://www.oclc.org/research/projects/archive/wcp/default.htm

de recherche et pour la recherche d'information en général.

Nous nous sommes intéressés au problème d'extraction de communautés dans le graphe du Web. Dans ce but, nous proposons deux algorithmes de calcul de communautés par émergence ainsi qu'une métrique pour mesurer la pertinence de nos communautés.

Organisation de la thèse

Les travaux présentés dans cette thèse sont divisés en deux parties : la modélisation des réseaux d'interactions et la modélisation et l'extraction des communautés dans les réseaux d'interactions.

—

La première partie présente un nouveau modèle de crawls aléatoire du Web. Elle est composée de trois chapitres. Les deux premiers sont principalement des chapitres introductifs décrivant, après une bref présentation des différents réseaux d'interactions, les propriétés communes de ces derniers, ensuite une présentation détaillée de plusieurs modèles aléatoires de réseaux d'interactions est faite. Le dernier chapitre présente notre modèle de crawls aléatoires.

—

Le premier chapitre présente les principales propriétés communes des réseaux d'interactions telles que la distribution des degrés en loi de puissance, une distance moyen faible, un fort coefficient de regroupement (clustering), l'abondance de bicliques (noyaux ou cores), la faiblesse de la densité et l'existence d'une composante connexe géante. Ces propriétés sont communes à des réseaux issus de différents domaines (Sociologie, Web, Internet, Biologie,etc), elles permettent de mieux comprendre la structure du réseau étudié et le lien de parenté entre eux, qui n'est pas intuitif.

—

Le deuxième chapitre est consacré aux modèles de graphes aléatoires. Beaucoup de réseaux d'interactions sont difficiles à obtenir. Il est donc intéressant de simuler des graphes avec les mêmes propriétés que les réseaux étudiés. Cela permet de tester des algorithmes, mais aussi de vérifier si l'on comprend le processus de génération du réseau réel : mieux il est maîtrisé, plus les graphes générés auront des propriétés des graphes réels. Nous allons notamment parler du modèle d'Erdös et Rényi [42, 43, 44] qui génère des graphes avec une distribution des degrés en loi de Poisson, du modèle de Barabasi [16] qui génère des graphes avec une distribution des degrés en loi de puissance mais avec un coefficient de regroupement très faible et nous décrirons le modèle petits mondes de Watts et Strogatz [104] qui génère des graphes avec un fort coefficient de regroupement et une distance moyenne faible. En réalité, il est difficile de trouver un modèle regroupant toutes les propriétés communes des réseaux d'interactions.

—

Le troisième chapitre décrit notre modèle de crawls aléatoires du Web. Il est motivé par l'intuition que certaines des propriétés attribuées au graphe du Web sont en réalité des artefacts

dus au parcours du Web. En effet, les «crawls» étudiés ont été obtenus en parcourant le Web. Nous pensons que si nous parcourons le Web de différentes manières alors nous obtenons différents résultats. Nous avons proposé un modèle de génération de crawls aléatoires basé sur la simulation d'un parcours. Nous montrons empiriquement que les crawls générés en utilisant un parcours en largeur ont beaucoup des propriétés du graphe du Web. Le résultat de ce chapitre n'a pas encore fait l'objet d'une publication.

—

La seconde partie décrit deux modèles d'extraction de communautés sur le Web. Elle est composée de deux chapitres, le premier est un chapitre introductif sur les algorithmes de tri des pages Web et le second présente nos deux modèles : gravitationnel et intentionnel.

—

Le quatrième chapitre présente l'algorithme PageRank pour la classification de pages Web. Cet algorithme est utilisé par le moteur de recherche *Google*. Il permet de donner un poids à toutes les pages Web. Ce poids dépend seulement de la structure hypertexte du Web et non de la requête utilisateur et est une évaluation de l'accessibilité d'une page, donc de sa popularité. Ce chapitre présente également l'algorithme HITS (Hypertext Induced Topic Search) qui permet de donner deux poids a une page Web donnée, un poids d'*autorité* et un poids d'*annuaire* (*hub*). De manière récursive, une bonne autorité est une page Web pointée par des bons annuaires et un bon annuaire est une page Web qui pointe vers des bonnes autorités. Cette relation est appelée renforcement mutuel. Concrètement, dans un graphe, cela arrive dans des sous-graphes très denses en liens. Les autorités et les annuaires représentent généralement des communautés dans le Web. Les autorités sont des pages Web pertinentes pour un thème donné et les annuaires sont des bons pointeurs vers le thème en question.

—

Le cinquième et dernier chapitre présente deux approches pour extraire des communautés dans un réseaux d'interactions. Nous présentons deux modèles de calcul par *émergence* des communautés. Nous présentons d'abord le modèle gravitationnel où un noeuds d'un réseau d'interaction est doté d'une masse, d'une force d'attraction, d'une vitesse et d'une accélération permettant à ce dernier de se déplacer dans l'espace. Les forces entre les noeuds liés vont agir dans le but de faire émerger les communautés. Le deuxième modèle est une amélioration du modèle gravitationnel où les noeuds du réseau sont dotés d'un objectif : rejoindre sa communauté qui est quelque part dans l'espace. En réalité, les communautés émergent et sont seulement «cueillies». Dans le premier modèle : le modèle gravitationnel, les noeuds du réseau sont dotes de masses (poids) et exercent sur les noeuds une force d'attraction. Cette force permet aux noeuds de se mouvoir dans l'espace. L'idée est de faire interagir les noeuds liés entre eux afin de former des points où les noeuds vont se concentrer pour former des communautés. Il suffit a la fin, de détecter les noeuds accumulés au même endroit. Le second modèle est une amélioration du premier, dotant cette fois-ci les noeuds d'intentions ou d'objectif. Dans ce modèle, un noeud a un objectif qui consiste a rejoindre sa communauté. Le premier modèle a fait l'objet d'un publication

[18, 19] et le deuxième n’a pas encore fait l’objet d’une publication.

—

Enfin des **annexes** présentent tout d’abord une historique du Web et un rappel des notions algébriques utilisées par HITS. La troisième annexe présente un outil de crawl que nous avons développé.

Première partie

Analyse et modélisation des réseaux d’interactions : application aux crawls du Web

Chapitre 1

Propriétés des réseaux d'interactions

UN RÉSEAU d'interactions est caractérisé par un grand ensemble d'objets qui, comme son nom l'indique, interagissent entre eux. Ils sont issus de différents domaines, par exemple : Internet, qui est constitué d'un grand nombre de machines liées par des supports de communication (câble, satellites, etc.), le Web qui est constitué d'un grand nombre de pages Web liées par des liens hypertextes, ou encore, les interactions entre protéines qui est constitué d'un grand nombre de protéines qui interagissent entre elles. Il existe également des grands réseaux en sociologie, économie, linguistique, etc.

L'hétérogénéité de ces réseaux laisse à penser que à part leur taille gigantesque, ils n'ont *a priori* rien de commun. Des études menées sur plusieurs réseaux de différents domaines ont révélé qu'il n'en était rien. Bien que issus de domaines différents, ces réseaux ont en commun des propriétés très intéressantes.

Dans ce chapitre, nous allons présenter les propriétés partagées par les réseaux d'interactions. Elles seront utilisées dans la génération de réseaux d'interactions où nous présentons un éventail de modèles aléatoires.

Ce chapitre est organisé de la manière suivante : la section 1.1 présente quelques définitions de base sur les graphes. Dans la section 1.2, nous présentons quelques réseaux d'interactions rencontrés dans la pratique. Enfin, dans la section 1.3, nous présentons les propriétés partagées par les réseaux d'interactions.

1.1 Définitions et notations

Nous présentons tout d'abord quelques définitions générales sur les graphes que l'on utilisera dans tout le mémoire.

Soit $G(V,E)$ un graphe orienté, où V est l'ensemble des sommets et $E \subseteq V \times V$ l'ensemble des arcs.

Le **degré sortant** d'un sommet, noté $d^+(u)$, est le nombre d'arcs qui en partent, et le **degré entrant** d'un sommet, noté $d^-(u)$, est le nombre d'arcs qui y arrivent. Le **degré** d'un sommet,

noté $d(u)$, est égal à la somme de son degré entrant et de son degré sortant.

Le **degré sortant moyen** (respectivement entrant moyen) d'un graphe orienté est la moyenne des degrés sortant (respectivement entrant) de tous les sommets de ce graphe, il est généralement noté par $\overline{d}^+$ (respectivement $\overline{d}^-$). Dans le cas où le graphe n'est pas orienté, on parle de **degré moyen** et est noté $\overline{d}$.

Si (u,v) est un arc d'un graphe $G(V,E)$, on dit que le sommet v est **adjacent** au sommet u. Une **boucle** est un arc dont l'origine et l'extrémité sont les mêmes.

Le **voisinage** d'un sommet u, noté $N(u)$, est **l'ensemble** des sommets entrants vers u et sortants de u.

On dit qu'un graphe $G'(V',E')$ est un **sous-graphe** de $G(V,E)$ si $V' \subseteq V$ et $E' \subseteq E$.

Un **chemin** de longueur k d'un sommet u vers un sommet v dans $G(V,E)$ est une suite $(v_0,v_1,v_2,...,v_k)$ de sommets tels que $u = v_0$, $v = v_k$, et $(v_{i-1},v_i) \in E$ pour $i = 1,2,...,k$.

La **longueur** du chemin est le nombre k d'arcs dans le chemin. Le chemin contient les sommets $v_0,v_1,...,v_k$ et les arcs $(v_0,v_1),(v_1,v_2),...,(v_{k-1},v_k)$. S'il existe un chemin c de u à v, on dit que v est accessible à partir de u via c.

Un **cycle** dans un graphe non orienté de longueur k est une suite $(v_0,v_1,v_2,...,v_k)$ de sommets tels que $v_0 = v_k$ et $v_1,v_2,\ldots,v_{k-1}$ sont distincts. Dans le cas orienté, on parle de **circuit**.

Si dans $G(V,E)$ chaque sommet a un arc vers tous les autres sommets du graphe alors $G(V,E)$ est une **clique**.

Le **plus court chemin** de u a v est chemin quelconque reliant le sommet u au sommet v avec la longueur la plus courte possible.

La **distance** entre u et v notée $\delta(u,v)$ est la longueur d'un plus court chemin. Si v n'est pas accessible à partir de u, la distance est infinie. La **distance moyenne** d'un graphe notée $\overline{\delta}$ est la moyenne des distances dans le graphe.

Le diamètre d'un graphe est la plus grande de toutes les distances dans le graphe.

Un graphe est **fortement connexe** (respectivement *connexe* dans un *graphe non orienté*) si chaque sommet est accessible à partir de n'importe quel autre. **Les composantes fortement connexes** d'un graphe forment les classes d'équivalence de la relation entre sommets «sont accessibles mutuellement». Un graphe est fortement connexe s'il n'est composé que d'une seule composante fortement connexe.

Défini par Watts et Strogatz [104], le **coefficient de regroupement** (*Clustering coefficient*) est une mesure de la connectivité d'un graphe **non orienté**. Soit $d(u)$ le degré du sommet u, et soit $N(u)$ l'ensemble des sommets voisins de u et soit $G(N(u))$ le sous graphe contenant que les arêtes entre les sommets de $N(u)$. Le coefficient de regroupement d'un sommet u noté $C(u)$ est définie comme suit :

$$C(u) = \frac{k}{\frac{d(u)\times(d(u)-1)}{2}}$$

où k est le nombre d'arêtes dans $G(N(u))$. $C(u)$ n'est rien d'autre que la probabilité de l'existence d'une arête dans le sous graphe $G(N(u))$. Si $G(N(u))$ est une clique alors $C(u) = 1$

et si $G(N(u))$ ne contient aucune arêtes alors $C(u) = 0$. Dans l'exemple de la figure 1.1, le coefficient de regroupement du sommet u est $C(u) = \frac{5}{\frac{4\times3}{2}} \simeq 0.83$.

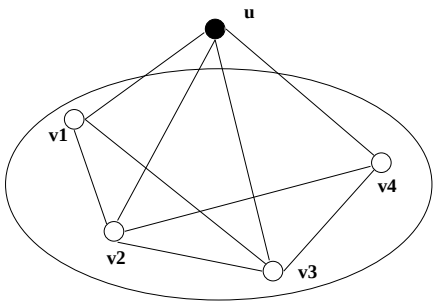


FIG. 1.1 – *Coefficient de regroupement d'un sommet.*

Le coefficient de regroupement du graphe $G(V,E)$ noté $C(G)$ est la moyenne des coefficients de regroupement de chaque sommet de G.

La **densité** est définie comme le rapport du nombre d'arêtes dans un graphe $G(V,E)$ sur le nombre d'arêtes maximal qu'il pourrait contenir :

$$\text{densité}(G) = \frac{m}{\frac{n(n-1)}{2}} = \frac{2m}{n(n-1)}$$

Où $|E| = m$ et $|V| = n$ sont respectivement le nombre d'arêtes et le nombre de sommets dans le graphe.

Un graphe admet une **racine** r si pour tout $v \in V$ il existe un chemin de r à v.

Remarque 1
Un graphe sans circuit qui admet une racine, en admet une seule.

Une **arborescence** est un graphe *connexe* sans *cycle* et muni d'une *racine*.

L'**ordre** d'un graphe (respectivement d'un arbre) correspond au nombre de sommets du graphe (respectivement d'un arbre). Il est noté $n(G)$ ou n.

Une **forêt** est un graphe non orienté sans cycle.

Le complément d'un graphe $G(V,E)$ est un graphe $\overline{G}(V,\overline{E})$ où $\overline{E} = (u,v)/(u,v) \notin E$. C'est à dire, $\overline{G}$ contient exactement les arêtes qui ne sont pas dans G (voir figure 1.2).

Un réseau d'interactions peut être modélisé par un graphe $G(V,E)$. La plupart sont des graphes non orientés mais certains peuvent être orientés, par exemple, le *Web*.

1.2 Réseaux d'interactions

Il existe dans de nombreux domaines des réseaux de très grande taille. Nous étudions ici trois grandes familles (mais il en existe d'autres) : les réseaux liés à Internet, les réseaux sociaux et les

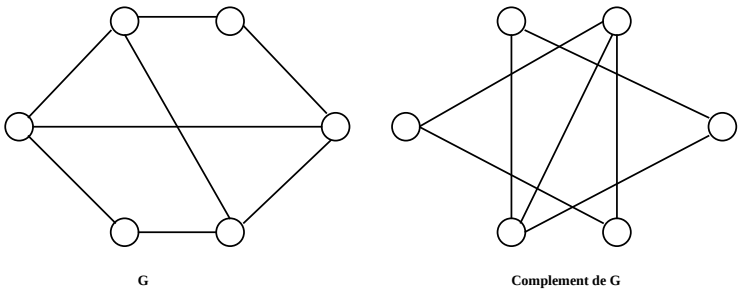


FIG. 1.2 – *Graphe et son complément.*

réseaux utilisés en biologie. Présentons plus en détails ces différents objets.

1.2.1 Modèles de réseaux issus de l'Internet

Un des réseaux les plus connu est le réseau Internet (ou encore les réseaux d'Internet). Plusieurs applications utilisent ce réseau pour transmettre de l'information. Les relations qu'entretiennent les instances de ces applications entre elle peuvent chacune être également vues comme des réseaux d'interactions.

Internet

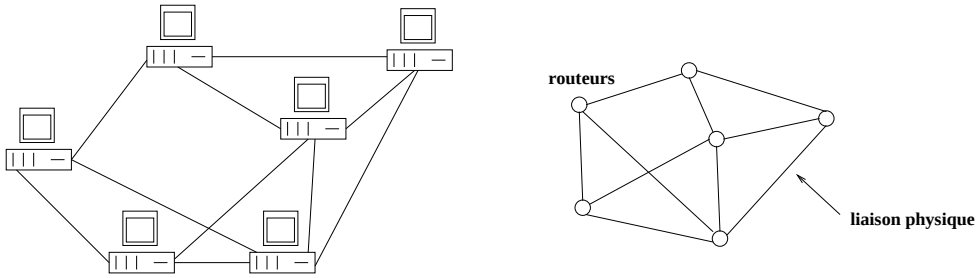


FIG. 1.3 – *Modélisation du réseau Internet par un graphe.*

Le graphe d'Internet (voir figure 1.3) est un graphe dont les sommets sont des ordinateurs ou des routeurs et les arêtes représentent des liaisons physiques entre les ordinateurs [45, 50].

Le graphe des domaines Internet est un graphe dont les sommets représentent des domaines ou systèmes autonomes, c'est-à-dire un ensemble d'ordinateurs gérés par le même administrateur (au sens large) et deux domaines sont liés par une arête s'il existe au moins une connexion entre les deux domaines [105, 50].

World Wide Web

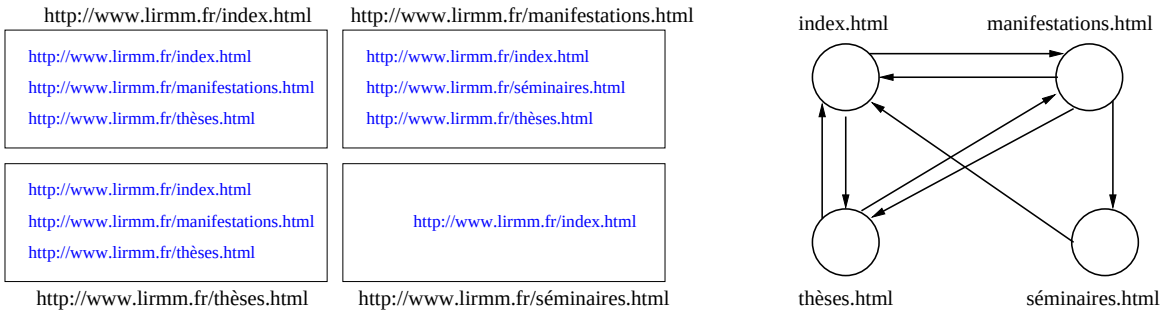


FIG. 1.4 – *Modélisation du Web par un graphe.*

Le Web est une application qui utilise le réseau Internet. Le graphe du Web est un graphe dont les sommets sont des pages HTML statiques et les arcs des liens hypertextes (voir figure 1.4). Notons bien qu'il s'agit d'un réseau logique et non physique, contrairement à Internet.

Actuellement, le moteur de recherche Google indexe plus de 8 milliards de pages Web (voir annexe C page 119). Ce chiffre représenterait un peu moins de 30% de la partie visible du Web. En effet, il existe une partie cachée où l'accès à l'information nécessite l'utilisation de formulaire et donc la saisie de données [9, 66, 62, 28, 23].

Sites Web

Le graphe des sites Web est un graphe dont les sommets représentent des serveurs Web (exemple : *www.lirmm.fr*) et deux sites Web sont liés si au moins une pages Web du premier site Web pointe vers une page Web du deuxième site Web [56, 5].

1.2.2 Les réseaux sociaux

Les graphes sont utilisés couramment en sciences sociales pour modéliser des interactions entre individus. Les sommets représentent alors les entités ou individus, et les arêtes ou arcs une relation ou interaction entre eux.

Les réseaux acteurs

Le graphe des acteurs est un graphe dont les sommets sont des acteurs et deux acteurs sont reliés s'ils ont joué ensemble dans un film [16, 12]. En 2000, la taille du graphe d'acteurs étudié était de 449 913 sommets [88]. Les données sont disponibles grâce à Internet Movie DataBase [1].

1. `http://www.imdb.com/`

Les réseaux de citations

Le graphe des citations est un graphe dont les sommets sont des publications scientifiques et deux publications u et v sont liées par l'arc (u,v) si la publication u a cité en références la publication v. Dans [95], des études ont été réalisées sur un graphe de 783 339 sommets issu du catalogue de l'*Institute for Scientific Information* et un graphe de 24 296 sommets issu des publications de la revue *Physical Review D*.

Les réseaux de co-auteurs

Le graphe de co-auteurs est un graphe dont les sommets sont des auteurs scientifiques et deux auteurs sont reliés s'ils ont une publication commune [88, 84, 85, 15]. Les données sont issus de la recherche en physique, en informatique et en biomédicale entre 1995 et 1999 [88, 84, 85]. Barabasi *et al.* [15] ont utilisé des données issues des mathématiques et des neurosciences entre 1991 et 1998.

Les réseaux de connaissance

Le graphe de connaissance est un graphe dont les sommets sont des personnes et deux personnes sont liés si elles se connaissent [77]. Bien entendu, une telle relation ne peut être définie de façon très formelle et le graphe ne peut être totalement construit.

Les réseaux d'appels téléphoniques

Le graphes des appels téléphoniques est un graphe orienté dont les sommets représentent des numéros de téléphones et un arc signale qu'un numéro a au moins une fois appelé un autre [8].

Les réseaux de contacts sexuels

Le graphe des contacts sexuels est un graphe dont les sommets sont des personnes et deux personnes sont liées si elles ont eu un rapport sexuel [71]. Notons que les données sont relativement difficiles à obtenir.

1.2.3 Les réseaux utilisés en biologie

Réseau d'interactions entre protéines

Le réseau d'interactions entre protéines est modélisé par un graphe dans lequel les sommets sont des protéines et deux protéines sont reliées si elles réagissent l'une avec l'autre [57].

Les réseaux trophiques

Les réseaux trophiques sont représentés par un graphe dans lequel un sommet est une espèce animale ou végétale et deux espèces sont reliées si une des deux espèces est un prédateur de l'autre.

1.2.4 D'autres réseaux d'interactions

Les réseaux de co-occurrence lexicale

Le graphe de co-occurrence lexicale d'un document donné est un graphe dont les sommets sont des mots d'un document et deux mots sont reliés s'ils apparaissent proche dans le document [46].

Les réseaux de distribution électrique

Le graphe de réseaux de distribution électrique est un graphe dont les sommets sont des stations (générateurs, transformateurs) électriques et deux stations sont liées si une ligne de haute tension existe entre les deux stations [104, 12].

Plusieurs applications sont utilisées dans les réseaux d'interactions : routage efficace sur Internet, algorithmes de recherche d'information efficaces sur le Web,etc. Une connaissance approfondie et exacte des réseaux d'interactions permet de concevoir des applications mieux adaptées et plus performantes.

En effet, l'étude des réseaux d'interactions en tant que graphe a permis, dans le Web, de développer des algorithmes très efficaces tel que l'algorithme PageRank utilisé par le moteur de recherche Google. Cette thèse est largement consacrée à de telles problématiques.

Dans la prochaine section, nous allons décrire six propriétés communes aux réseaux d'interactions, elles ont été introduites afin de mieux comprendre la structure de ces réseaux d'interactions en tant que graphe.

1.3 Propriétés communes aux réseaux d'interactions

L'étude des réseaux d'interactions en tant que graphe a révélée que ces derniers, bien que issus de différents domaines, partagent quelques propriétés intéressantes.

Dans cette section, nous présentons six propriétés communes aux réseaux d'interactions : distribution des degrés en loi de puissance, distance moyenne faible, diamètre petit, fort coefficient de regroupement (clustering coefficient), densité faible, composante connexe géante et nombre élevé de bicliques.

1.3.1 Distribution des degrés en loi de puissance

La distribution des degrés d'un réseau d'interactions suit une loi de puissance de type :

$$P(d) = C.d^{-\lambda},$$

où d est le degré, C un paramètre qui dépend de la taille du réseau et λ est l'exposant de la loi de puissance. Dans la pratique la valeur de λ est compris entre 2 et 3. Le tableau 1.3.1 donne l'exposant de la loi de puissance de quelques réseaux d'interactions. On observe que $WWW_{1,2,3}$ (trois sous graphes du Web, issues de trois *crawls*[2] différents) sont de taille variable et que les exposants varient très peu. Ce paramètre semble donc intrinsèque à la nature du graphe, et non relié à la taille de l'échantillon.

Tableau 1.3.1 : Distribution des degrés de différents réseaux d'interactions.

Graphe	Taille	$\overline{d}$	λ_+	λ_-	Références
WWW_1	325729	4.51	2.45	2.1	[9]
WWW_2	4×10^7	7	2.38	2.1	[62]
WWW_3	2×10^8	7.5	2.72	2.1	[28]
sites Web	260000	?	1.94	?	[56]
Domaine, internet	3015-4389	3.52-4.11	2.1-2.2	2.1-2.2	[45]
Routeur, internet	3888	2.57	2.48	2.48	[45]
Routeur, internet	150000	2.66	2.4	2.4	[50]
Appels	53×10^6	3.16	2.1	2.1	[8]
Citations	783339	8.57	?	3	[95]
Acteurs	212250	28.78	2.3	2.3	[16]
co-auteurs, SPIRES	56627	173	1.2	1.2	[88]
co-auteurs, neuro.	209293	11.54	2.1	2.1	[15]
co-auteurs, math.	70975	3.9	2.5	2.5	[15]
Contacts sexuels	2810	?	3.4	3.4	[71]
co-occurrence des mots	22311	70.13	2.8	2.8	[46]

Une distribution des degrés en loi de puissance signifie qu'il existe beaucoup de sommets de faible degré et très peu de sommets de fort degré. L'exposant représente la vitesse de décroissance de la courbe des degrés. Plus λ est grand et plus la probabilité d'obtenir des sommets de fort degré est petite. Le degré moyen dans un graphe en loi de puissance n'est pas significatif car l'écart-type est très important. Les graphes en loi de puissances sont appelés graphes à invariance d'échelle (scale-free graphs).

1.3.2 Diamètre petit et distance moyenne faible

Beaucoup ont pensé que le diamètre des réseaux d'interactions pourrait être grand et que leur distance moyenne pourrait être forte. Mais des études (voir le tableau 1.3.2) sur différents

2. voir annexe C page 119.

réseaux d'interactions ont montré qu'il n'en est rien. Les réseaux d'interactions ont un diamètre petit et une distance moyenne faible de l'ordre de $(log(n))$ ou n est la taille du graphe.

Le calcul de la distance moyenne et du diamètre étant coûteux en temps ((nm)), une estimation de la distance moyenne est faite en l'évaluant seulement pour un certain nombre de paires de sommets.

Tableau 1.3.2 : Le degré moyen $\overline{d}$ et la distance moyenne $\overline{\delta}$ de différents réseaux d'interactions.

Graphe	Taille	$\overline{d}$	$\overline{\delta}$	Références
sites Web	153127	35.21	3.1	[5]
Domaine, internet	3015-6209	3.52-4.11	3.7-3.76	[105]
Routeur, internet	3888	2.57	12.15	[45]
Routeur, internet	150000	2.66	11	[50]
Appels téléphoniques	53×10^6	3.16	?	[8]
Citations	783339	8.57	?	[95]
Acteurs	225226	61	3.65	[104]
Co-auteurs, SPIRES	56627	173	4	[88]
Co-auteurs, neuro.	209293	11.5	6	[15]
Co-auteurs, math.	70975	3.9	9.5	[15]
Contacts sexuels	2810	?	?	[71]
Co-occurrence des mots	22311	13.48	4.5	[46]

Le tableau 1.3.2 montre que la plupart des réseaux d'interactions ont une distance moyenne très faible. Dans la littérature, le diamètre des réseaux d'interactions n'est souvent pas donné mais est souvent d'un facteur de 2 à 5 de la distance moyenne (pour les graphes non orientés).

Broder *et al.* [28] ont montré empiriquement que le graphe du Web a un diamètre très petit par rapport à sa taille. Sur un graphe de 200 millions de pages Web, le diamètre est égal à 500 alors que la distance moyenne est seulement de 16. C'est-à-dire, si l'on choisit aléatoirement deux pages Web, *si un chemin existe entre elles*, alors la distance moyenne est de 16 dans le cas ou le graphe est orienté et 6 dans le cas non orienté [3].

En 1967, le sociologue *Stanley Milgram* a réalisé une séries d'expériences dont le résultat est connu sous le nom de «six degrés de séparation» [77]. La conclusion de ses études est en effet que deux personnes quelconques dans le monde ont une chaîne de connaissance de longueur six en moyenne. En d'autres termes, il y a cinq personnes intermédiaires qui séparent les deux personnes étrangère l'une de l'autre.

1.3.3 Densité faible

Lorsque n est assez grand, la densité du graphe peut alors être réécrite en fonction du degré moyen du graphe :

3. Ce résultat est sans doute dû au fait qu'un crawler réalise en général un parcours en largeur du Web, ce qui biaise la mesure (voir chapitre 3).

$$densité(G) = \frac{2m}{n}\frac{1}{(n-1)} = \frac{\overline{d}(G)}{n-1} \simeq \frac{\overline{d}(G)}{n}$$

La densité des réseaux d'interactions est très peu élevée (de l'ordre de $1/n$). en effet, le nombre d'arêtes dans un graphe d'interactions est du même ordre que son nombre de sommets. Le tableau 1.3.3 donne la densité de quelques graphes d'interactions[Gui04].

Tableau 1.3.3 : Nombre de sommets, de liens, degré moyen et densité de quelques réseaux d'interactions [Gui04].

	Internet	Web	Acteurs	co-auteurs	co-occurrence	protéines
$\|V\|$	75885	325729	392340	16401	9297	2113
$\|E\|$	357317	1090108	15038083	29552	392066	2203
$\overline{d}(G)$	9.42	6.69	76.66	3.6	84.34	2.09
$densité(G)$	1.2e-4	2.1e-5	1.9e-4	2.2e-4	9.1e-3	9.9e-4

1.3.4 Coefficient de regroupement fort

Introduit par Watts et Strogatz [104], le coefficient de regroupement (*Clustering*) mesure la probabilité que le voisinage d'un sommet soit dense en liens (voir § 1.1). Le tableau 1.3.4, donne le coefficient de regroupement de certains réseaux d'interactions.

Tableau 1.3.4 : Le coefficient de regroupement C de quelques réseaux d'interactions.

Graphe	Taille	$\overline{d}$	C	Références
sites Web	153127	35.21	0.1078	[5]
Domaine, internet	3015-6209	3.52-4.11	0.18-0.3	[105]
Acteurs	225226	61	0.79	[104]
Co-auteurs, SPIRES	56627	173	0.726	[88]
Co-auteurs, neuro.	209293	11.5	0.76	[15]
Co-auteurs, math.	70975	3.9	0.59	[15]
Co-occurrence des mots	460902	70.13	0.437	[46]

Le tableau 1.3.4, montre que les réseaux d'interactions ont un fort coefficient de regroupement. Ce résultat est assez surprenant du fait que la densité de ces graphes est assez faible. En résumé, on peut dire que localement les réseaux d'interactions sont dense en liens et globalement leur densité est faible.

1.3.5 Composante connexe géante

Broder *et al.* [28] ont étudié un sous-graphe du Web de 200 millions de pages Web et de 1,5 milliard de liens. Cette étude a affirmé que le Web peut être partitionné en quatre ensembles

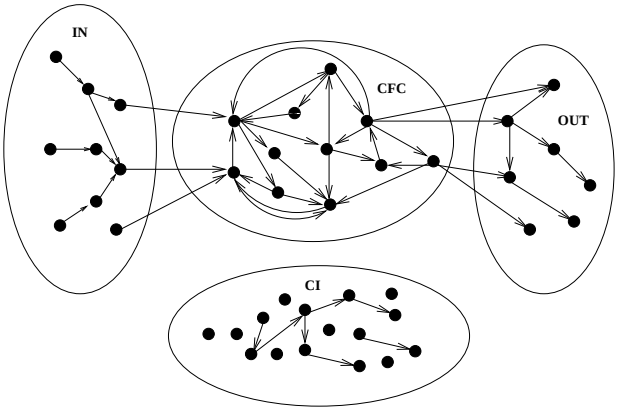


FIG. 1.5 – *Structure en noeud Papillon.*

approximativement de même taille (voir la figure 1.5) :

- Une composante fortement connexe notée **CFC** où toute paire de page Web est reliée par un chemin. Sa taille est de 56 millions de pages Web. Le diamètre de cette composante est de 28 ;
- Une composante notée **OUT** où les pages Web sont accessibles par les pages Web de la composante *CFC* mais pas l'inverse. La taille de cette composante est de 44 millions de pages Web ;
- Une composante notée **IN** où les pages Web de cette composante accèdent aux pages Web de la composante *CFC* mais pas l'inverse. La taille de la composante *IN* est de 44 millions de pages Web. On remarque que les composantes *IN* et *OUT* sont de même taille ;
- Une composante isolée notée **CI** où les pages Web de cette composante n'ont pas accès aux pages Web des composantes *IN*, *OUT* et *CFC* et réciproquement. Sa taille est proche de celle de la composante *IN* et *OUT*.

Il existe également des **tubes** qui relient des pages Web de la composantes *IN* aux pages de la composante *OUT*. Enfin les **Vrilles** (*Tendrils*) sont des longs chemins connectés aux composantes *IN* et *OUT*.

La figure 1.6 montre comment un site Web peut aussi être structuré en forme de *noeud Papillon*. La propriété de noeud papillon peut donc s'appliquer au niveau local. Néanmoins, la composante *IN* et notamment sa taille nécessite quelques explications.

On peut supposer que la composante IN contiendrait surtout des pages Web personnelles. En effet sur des pages personnelles ont trouve beaucoup de liens vers d'autres sites Web. En revanche, très peu de sites Web commerciaux ou gouvernementaux pointent vers leurs pages Web.

De même, on peut supposer que la composante OUT contiendrait des sites Web à caractère commercial, ces sites sont souvent fermés sur eux même et ne pointent pas vers l'extérieur (les

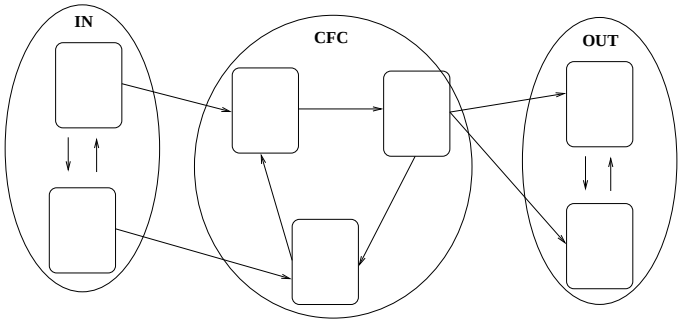


FIG. 1.6 – *Site Web en noeud Papillon.*

moteurs de recherche les plus populaire font partie de cette composante).

Pour la composante CFC, on peut supposer qu'elle contiendrait des pages à caractères social : journaux, ambassades, portails des villes, pays, encyclopédies, dictionnaires, etc. Ces sites Web donnent des informations aux utilisateurs d'Internet et ne sont pas à caractère commercial.

Enfin, on peut supposer que la composante CI contient des sites Web pas très populaires et fermés sur eux mêmes ou encore des pages Web qui ne sont plus d'actualité.

Un site Web ou une page Web peut évoluer d'une composante vers l'autre. Ce changement intervient généralement après une mise à jour de la page Web considérée ou des pages Web faisant référence.

Néanmoins, le modèle de *noeud papillon* ne peut être une représentation réelle du Web. En effet, Mathieu [75] a construit une page Web dynamique à partir de laquelle toutes les pages Web sont accessibles[4]. Il suffit de construire une URL et si cette dernière est valide, alors il est possible d'y accéder. Il s'agit là bien sûr d'un contre-argument théorique. Mais on peut se demander si cette soi-disant forme du Web ne serait pas un artefact issu du *crawling*, la façon dont la donnée est obtenue. Nous y reviendrons au chapitre 3.

A notre connaissance, aucune validation plus récente de ce modèle n'a été réalisée au moment de rédiger ce mémoire.

Le Web ne contient pas qu'une seule composante connexe géante mais également plusieurs petites composantes connexes de différentes tailles. La distribution de la taille des composantes fortement connexes du Web (CFC) suit une loi de puissance. Dans le cas non orienté (CC), la taille de la composante connexe géante est la somme des 3 parties OUT, CFC et IN [28].

La plupart des graphes d'interactions contiennent une composante connexe géante. Dans le Web, si l'orientation des arcs est supprimée, le graphe du Web contient une composante connexe comprenant les $\frac{3}{4}$ des pages Web selon le modèle de [28].

4. http://www.liafa.jussieu.fr/ fmathieu/arbre.php

1.3.6 Abondance de bicliques

Un graphe biparti est un graphe orienté $G(V,E)$ dans lequel V peut être partitionné en deux ensembles V_1 et V_2 tel que $(u,v) \in E$ implique que, soit $u \in V_1$ et $v \in V_2$, soit $u \in V_2$ et $v \in V_1$. En d'autres termes, tous les arcs passent entre les deux ensembles V_1 et V_2.

Une *biclique* est un graphe biparti *complet* orienté $K_{|V_1|,|V_2|}$ où chaque sommet de V_1 pointe (a un arc sortant) vers chaque sommet de V_2.

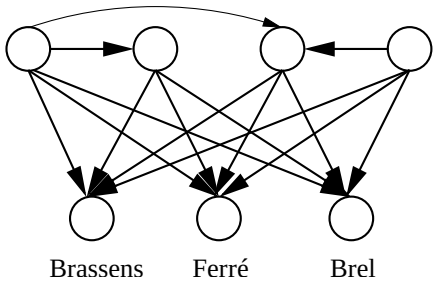


FIG. 1.7 – *Une biclique sur le Web.*

On peut tenter d'analyser la signification d'une biclique dans le Web. Des pages Web d'un ensemble l pointent toutes vers des pages Web d'un ensemble w. Cela signifierait que les pages Web de l et w **partagent des informations sur un sujet particulier**. Dans ce cas, on dit que les pages Web ont un **intérêt** commun. Les deux ensembles l et w forment ce que [60] définissent comme étant une **communauté** (voir figure 1.7).

Kleinberg [60] considère deux qualités pour une page. Un annuaire (hub) pointe des pages intéressantes, tandis qu'une autorité (authority) est une page Web de référence pour un sujet donné. De manière récursive, un *bon* annuaire est une page Web qui pointe vers des page Web considérées comme des *bons* autorité et une page Web est considérée comme un *bonne* autorité si elle est pointée par des pages Web considérées comme des *bons* annuaires. Cette relation entre annuaire et autorité est appelée renforcement mutuel. l'ensemble des hubs et des fans peut être utilisée comme une définition d'une communauté sur le Web. Nous verrons (voir chapitre 5 page 79) qu'il existe d'autres définitions des communautés.

Kumar *et al.* [64] ont montré empiriquement que le Web contient un grand nombre de bicliques de différentes tailles.

Tableau 1.3.6 : Énumération de bicliques dans un crawl du Web [64]

$\|V_1\|$	$\|V_2\|$	bicliques
3	3	89565
3	5	70168
3	7	60614
3	9	53567
4	3	29769
4	5	21598
4	7	17754
4	9	1258

$\|V_1\|$	$\|V_2\|$	bicliques
5	3	11438
5	5	8062
5	7	6626
5	9	4684
6	3	4854
6	5	3196
6	7	2549
6	9	2141

Le tableau 1.3.6 énumère les bicliques dans un crawl du Web [64]. On observe que le graphe du Web contient un grand nombre de bicliques et de différentes tailles.

En résumé, les réseaux d'interactions issus de différents domaines partagent des propriétés intéressantes : une distribution des degrés en loi de puissance d'exposant compris entre 2 et 3, une distance moyenne faible, un diamètre petit, un fort coefficient de regroupement, une densité faibles et un nombre de biclique élevé.

Ces propriétés permettent d'avoir une description approfondie de ces réseaux et ainsi avoir la possibilité de faire face à différentes situations tel que les attaques et la résistance aux pannes (Internet), la lutte efficace contre la propagation des épidémies (sociaux), la protection d'un écosystème (réseaux trophiques),etc.

Deux problèmes majeurs interviennent lorsqu'on désire se procurer des données de certains réseaux d'interactions : La taille du réseaux (quantité de données) et la dynamique (modification des données). On peut citer comme exemple le Web, le nombre de pages Web est assez grand, il est n'est pas possible avec la technologie actuelle de se procurer tout le Web en temps raisonnable. De plus, le Web est toujours en activité, c'est-à-dire, des pages Web sont ajoutées, mises à jours et supprimées quotidiennement.

Pour pouvoir mieux étudier les graphes d'interactions, il est nécessaire de construire des modèles qui génèrent des graphes avec les mêmes caractéristiques que les réseaux d'interactions. C'est l'objet du prochain chapitre.

Chapitre 2

Modélisation des réseaux d'interactions

LA MODÉLISATION de réseaux d'interactions est le fait de construire des graphes aléatoires avec les mêmes propriétés que les réseaux réels. L'intérêt est de pouvoir effectuer sur machines des simulations de pannes, d'attaques, de propagation et d'autres événements qui peuvent survenir sur les réseaux réels. L'avantage de la modélisation est de pouvoir obtenir des graphes de grande taille en temps raisonnable.

Dans ce chapitre nous allons faire une présentation de différents modèles de réseaux d'interactions en nous focalisant sur ceux modélisant le Web. Au cours de cette présentation, nous allons montrer qu'il n'est pas simple de trouver un modèle regroupant toutes les propriétés des réseaux d'interactions décrites dans la section 1.3.

Deux techniques sont principalement utilisées pour la génération de réseaux d'interactions. La première consiste à expliquer comment les propriétés communes aux réseaux d'interactions émergent. La seconde technique consiste à utiliser une propriété commune aux réseaux d'interactions et de faire émerger les autres.

Soit $G_t(V,E)$, le graphe généré à l'étape t. L'algorithme de génération peut être écrit comme suit :

1. G_0, à l'état initial (généralement vide) ;
2. G_t est un sous graphe de G_{t+1} ;
3. $|V(G_{t+1})| = |V(G_t)| + nbsommets$, $nbsommets > 0$, le nombre de nouveaux sommets ajoutés à l'étape t ;
4. $|E(G_{t+1})| = |E(G_t)| + nbarcs$, $nbarcs \geqslant 0$, le nombre d'arcs ou d'arêtes ajouté à l'étape t ;
5. Les arcs (ou arêtes) sont ajoutés de différentes manières :
   - entre les nouveaux sommets ajoutés (si plusieurs),
   - entre un nouveau sommet et un sommet déjà existant, ce dernier étant choisi équiprobablement ou avec une probabilité qui dépend de son degré entrant,
   - entre deux sommets déjà existants, ces derniers étant choisis équiprobablement ou avec une probabilité qui dépend du degré entrant,

– Enfin, entre un sommet existant et le nouveau sommet. Le sommet existant est choisi équiprobablement ou avec une probabilité qui dépend du degré entrant.

Ou encore, l'algorithme de génération se limite à ré-organiser les arêtes d'un graphe possédant déjà quelques propriétés communes des réseaux d'interactions [104].

Dans la section 1, nous allons décrire le modèle aléatoire d'Erdös et Rényi [42, 43, 44] longtemps utilisé pour modéliser les réseaux d'interactions. Ensuite, nous allons présenter plusieurs modèle de graphe avec une distribution des degrés en loi de puissance. Enfin, dans la section 3, une étude comparative est faite entre les diffèrents modèles.

2.1 Modèle aléatoire d'Erdös et Rényi

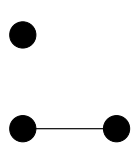
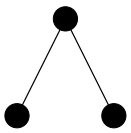
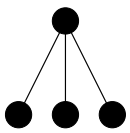
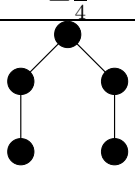
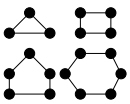
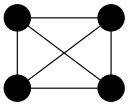
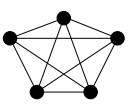
Lors d'une série de séminaires entre 1950 et 1960, Paul Erdös et Alfréd Rényi ont proposé et étudié les premiers modèles de réseaux d'interactions appelés les graphes aléatoires [42, 43, 44]. Le modèle minimal est un graphe à n sommets liés par des arcs choisis aléatoirement et uniformément. Erdös et Rényi ont apporté plusieurs versions de ce modèle, le plus étudié est noté $G_{n,p}$.

Définition 1
Un $G_{n,p}$ est un graphe à n sommets où, pour un couple de sommets donné, une arête les relie avec une probabilité p.

Erdös et Rényi [42, 43, 44] ont démontré que la connexité des $G_{n,p}$ présente un phénomène de transition de phase quant on fait croître la valeur de p. En effet, le graphe passe d'un état composé de plusieurs composantes connexes de petites tailles à un état contenant une composante connexe géante. Cette phase de transition se produit lorsque $\overline{d} = 1$ ($(p = \frac{1}{n})$).

Le tableau 2.1 montre une analyse plus fine des différents seuils dans un $G_{n,p}$ à partir duquel apparaissent différents sous-graphes.

Tableau 2.1 : Caractéristiques d'un $G_{n,p}$.

$p \sim n^z$							
z	$-\infty$	$-\frac{3}{2}$	$-\frac{4}{3}$	$-\frac{5}{4}$	-1	$-\frac{2}{3}$	$-\frac{1}{2}$
							
	sommets isolés	arbres d'ordre 3	arbres d'ordre 4	arbres d'ordre 5	arbre et cycles de tout ordre	cliques d'ordre $\leqslant 5$	clique géante

Lorsque la taille du graphe est grande, la distribution des degrés d'un $G_{n,p}$ suit une loi de Poisson [42, 43, 44] tel que :

$$p(d) \sim e^{-\lambda}\frac{\lambda^d}{d!}$$

Une telle distribution signifie que les degrés des sommets sont proches du degré moyen $\overline{d} = pn$ où n est la taille du graphe.

Fan et Lu [36] ont démontré que, pour $np \geqslant c > 1$ ($\overline{d} > 1$), le diamètre Δ est égal à :

$$\Delta(G_{n,p}) \leqslant \frac{\ln n}{\ln np} + 2\frac{(10c/(\sqrt{c}-1)^2+1)}{c-\ln(2c)}\frac{\ln n}{np} + 1$$

Le coefficient de regroupement d'un $G_{n,p}$ est de l'ordre de :

$$C(G_{n,p}) = \frac{\bar{d}}{n} \simeq \frac{np}{n} \simeq p.$$

Enfin, un $G_{n,p}$ avec toujours $p > \frac{1}{n}$ ($\overline{d} > 1$) contient presque sûrement une composante connexe géante [42, 43, 44].

C'est résultats montrent que la distribution des degrés des $G_{n,p}$ est différente d'une loi de puissance et que le coefficient de regroupement est plus fort dans les réseaux d'interactions que dans les $G_{n,p}$ (voir tableau 1.3.4). Par contre, on note deux points communs entre les $G_{n,p}$ et les réseaux d'interactions : une distance moyenne faible et un diamètre petit.

En résumé, les études menés sur les $G_{n,p}$ et sur les réseaux d'interactions permettent de dire que les $G_{n,p}$ ne peuvent être utilisés pour modéliser les réseaux d'interactions.

De plus les $G_{n,p}$ sont des graphes statiques, c'est-à-dire, le nombre de sommets est fixe et ne change pas au cours du temps. Or, il existe des réseaux d'interactions tel que le Web, où le nombre de pages Web et le nombre de liens augmentent chaque jour [70]. En d'autres termes, la dynamique des réseaux d'interactions n'est pas prise en considération dans les $G_{n,p}$. Actuellement, la dynamique des graphes est un sujet de recherche en pleine expansion.

2.2 Modélisation des graphes à invariance d'échelle

Les $G_{n,p}$ ont longtemps été utilisés pour modéliser les réseaux d'interactions. Des récentes études menés sur les réseaux d'interactions ont révélé que la distribution des degrés de ces derniers suivent une loi de puissance or que les $G_{n,p}$ suivent une loi de Poisson.

Depuis, plusieurs modèles aléatoires ont été proposé pour générer des graphes aléatoires avec une distribution des degrés en loi de puissance. Dans cette section, nous allons présenter quelques modèles de graphes sans-échelle.

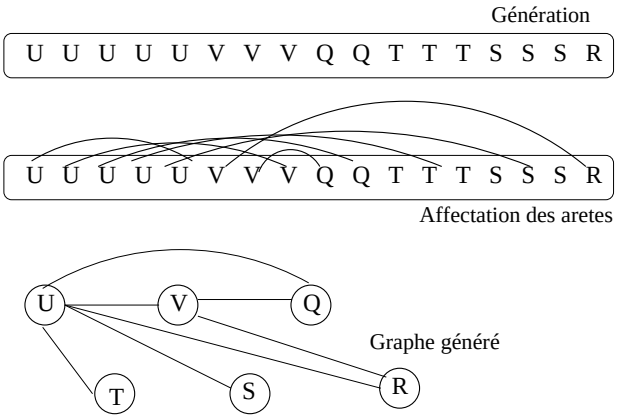


FIG. 2.1 – *Modèle* ACL.

2.2.1 Distribution des degrés fixés

Aiello *et al.* [8] proposent une méthode basée sur l'appariement de demi-arêtes où pour chaque sommet du graphe, son degré est fixé suivant une distribution des degrés en loi de puissance d'exposant donné et chaque sommet reçoit autant de demi-arêtes que son degré. Par la suite, les demi-arêtes sont reliées entre elles de manière aléatoire et indépendante (voir figure 2.1).

La distribution des degrés dépend de deux paramètre α et β tel que :

$$|\{v \backslash d(v) = x\}| = y = \frac{e^{\alpha}}{x^{\beta}}$$

Le paramètre α représente le logarithme du nombre de sommets de degré 1 et β l'exposant de la loi de puissance qui représente la vitesse de décroissance des degrés. Le nombre de sommets et d'arêtes ne sont pas des paramètres du modèle. Le degré maximum du graphe est $e^{\frac{\alpha}{\beta}}$.

Par définition, la distribution des degrés suit une loi de puissance de paramètre β. Le graphe généré par cette méthode peut contenir des boucles et des arêtes multiples mais pour des graphes de grande taille, le nombre de boucles et d'arêtes multiples est négligeable. Si $\beta < 1$ alors le graphe est sûrement connexe (c'est-à-dire, le graphe est connexe avec une probabilité proche de 1).

Si $1 < \beta < 2$ alors le graphe contient une composante connexe géante de taille $O(n)$ et plusieurs petites composantes connexe de taille $O(1)$. Si $2 < \beta < \beta_0 \simeq 3{,}4785$ alors le graphe contient une composante connexe géante de taille $O(n)$ et plusieurs petites composantes connexes de taille $O(\ln(n))$. Si $\beta > 4$ alors le nombre de composantes connexe de taille donnée suit une loi de puissance. Enfin, si $\beta > 8$ alors avec une probabilité proche de 1, le graphe ne contient aucune composante connexe [78, 79].

Si $\beta < \beta_0 \simeq 3{,}4785$ alors le graphe obtenu a une distance moyenne logarithmique, de même pour le diamètre. Néanmoins, le coefficient de regroupement tend vers 0 quand la taille du graphe

augmente [83].

2.2.2 Modèle à base d'attachement préférentiel

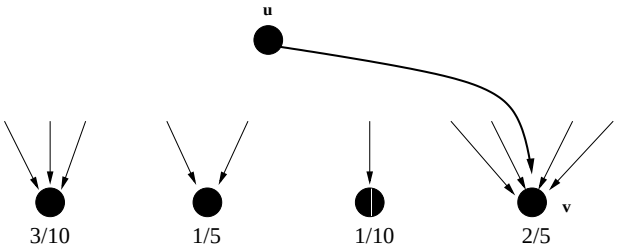


FIG. 2.2 – *Modèle d'attachement préférentiel (en dessous de chaque sommet est écrit son importance en termes de degré entrant).*

Le modèle de Albert et Barabasi [16] est basé sur le concept **d'attachement préférentiel** où un sommet de fort degré a une grande probabilité d'être choisi pour relier le nouveau sommet au reste du graphe (voir figure 2.2). L'attachement prérérentiel cherche à modéliser la façon dont un acteur de page Web lie sa page aux pages existantes en fonction de sa connaissance du Web. Or il connait plus probablement les pages plus populaires !

Un graphe $G(V,E)$ du modèle de Albert et Barabasi est construit de la manière suivante : initialement le graphe contient n_0 sommets, aucune information n'est donnée sur le degré de ces sommets. À l'étape t, un sommet est ajouté ainsi que m ($0 \leqslant m < n_0$) arêtes pour le relier au reste du graphe. La probabilité qu'un sommet v soit choisi comme destination de l'arête est $p(d(v)) = \frac{d(v)}{\sum_j d(j)}$ où $d(v)$ est le degré de v et $\sum_j d(j)$ est le nombre d'arêtes à l'étape $t-1$.

À l'étape t, le modèle converge vers un graphe avec $t + n_0$ sommets et $m \times t$ arêtes. La distribution des degrés de ce graphe suit une loi de puissance d'exposant $\lambda = 3$ [16].

Albert et Barabasi ont montré que l'attachement préférentiel était nécessaire dans leur modèle afin de générer des graphes avec une distribution des degrés en loi de puissance. En effet, lorsque l'attachement préférentiel est remplacée par une probabilité uniforme $p(d(v)) = \frac{1}{(n_0+t-1)} = cste$, la distribution des degrés $p(d) \sim \exp(-\beta d)$, est différente d'une loi de puissance.

De même, si l'on supprime l'évolution du graphe (le nombre de sommets reste fixe), après $t \simeq n^2$ étapes où n est le nombre de sommets du graphe, tous les sommets du graphe seront liés les uns aux autres (une clique).

Avec l'attachement préférentiel, un sommet choisi à l'étape t a une grande probabilité d'être encore choisi à l'étape $t+1$. Avec une forte probabilité, le graphe généré se décompose en n_0 forêts disjointes [22], sa distance moyenne est logarithmique, son taux de regroupement tend vers 0 quand sa taille augmente et le nombre de bicliques est négligeable.

Le modèle de Albert et Barabasi ne regroupe pas toutes les propriétés communes des réseaux d'interactions. En effet, les graphes générés ont un taux de regroupement quasi nul et très peu de

bicliques.

2.2.3 Modèle de copie

Ce modèle est basé sur le comportement de l'utilisateur sur le Web. En effet, lorsqu'un utilisateur crée une page Web, il va relier cette dernière au reste du Web par des liens hypertextes. Pour cela, il va choisir des pages sur le Web (des prototypes). Un prototype est une page Web populaire et donc possédant de nombreux liens entrants. Le choix des prototypes n'est pas aléatoire mais plutôt en relation avec le contenu de la nouvelle page Web. Par la suite, l'utilisateur copie des liens hypertextes des prototypes dans la nouvelle page Web. Un modèle à base de copie permet de construire des graphes riches en bicliques.

Kumar *et al.* [61] proposent deux modèles de copie : le modèle linéaire et le modèle exponentiel. Ces deux modèles génèrent des graphes aléatoires riches en bicliques et avec une distribution des degrés en loi de puissance.

Modèle linéaire

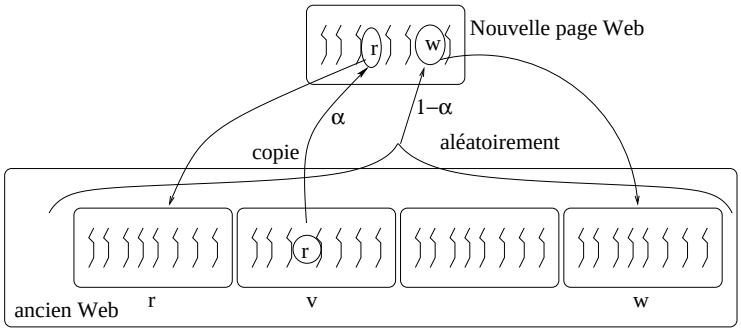


FIG. 2.3 – *Modèle linéaire.*

Un graphe $G(V,E)$ du modèle linéaire [61] est construit de la manière suivante (voir figure 2.3) :

1. G_0, aucune information n'est donnée du graphe à l'état initial ;
2. $|V(G_{t+1})| = |V(G_t)| + 1$, un seul sommet u est ajouté à chaque étape ;
3. $|E(G_{t+1})| = |E(G_t)| + d$ ($d \geqslant 1$), d arcs sont utilisés pour relier le nouveau sommet u au reste du graphe ;
4. deux sommets v et w sont choisit aléatoirement tel que :
   - avec une probabilité $\alpha \in [0,1]$, le $i^{\text{ème}}$ lien du sommet v est copié dans la page u,
   - avec une probabilité $1 - \alpha$, un lien est ajouté entre u et w.

α est un paramètre du modèle. Un graphe du modèle linéaire est caractérisé par une distribution des degrés entrants en loi de puissance d'exposant $\lambda_{in} = \frac{2-\alpha}{1-\alpha}$, un degré sortant constant égal à d et pour $i \leqslant \ln t$, un nombre de bicliques $K_{t,i,d} = t \times \exp(-i)$ [61].

Modèle exponentiel

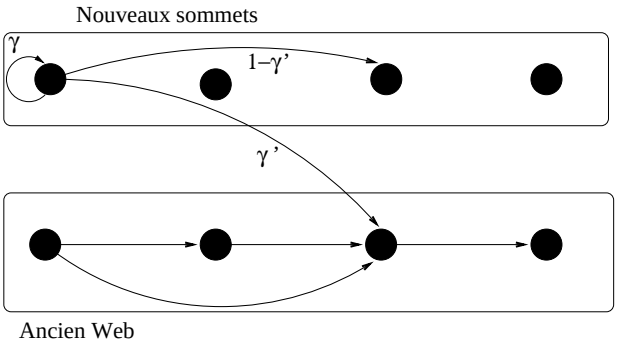


FIG. 2.4 – *Modèle exponentiel.*

Un graphe $G(V,E)$ du modèle exponentiel [61] est construit de la manière suivante (voir figure 2.4) :

1. à l'état initial aucune information n'est donnée ;
2. $|V(G_{t+1})| = |V(G_t)| + Binomial(V(G_t),p)$, en moyenne $(1+p)^t$ sommets sont ajoutés à chaque étape, $p > 0$;
3. $|E(G_{t+1})| = |E(G_t)| + (d+\gamma) * p * |E(G_t)|$ ($\gamma > 1$), $(d+\gamma) * p * |E(G_t)|$ arcs sont utilisés pour relier les nouveaux sommets au reste du graphe ;
4. chaque sommet a $\gamma > 1$ boucles ;
5. les arcs sont ajoutés de la manière suivante :
   - avec une probabilité $1-\gamma'$, la destination est choisie aléatoirement parmi les $(1+p)^t$ nouveaux sommets,
   - avec une probabilité γ', un sommet est choisi parmi les sommets déjà existant. la probabilité de choisir le sommet v dépend du degré sortant d_v^{out} tel que $p(d_v^{out}) = \frac{d_v^{out}}{((d+\gamma)|E(G_t)|)}$.

$p > 0$, $\gamma > 1$, $\gamma' \in [0,1]$ et $d > 0$ sont des paramètres du modèle. Les graphes générés ont une distribution des degrés entrants en loi de puissance d'exposant $\lambda_{in} = \ln_\mu(1+p)$ où $\mu = 1 + \frac{(pd)}{(d+\gamma)}$. Pour $i \leqslant \ln t$, le nombre de bicliques $K_{t,i,d} = \Omega(t \times \exp(-i))$ où (t, i) est la taille de la biclique.

Le modèle de copie permet de générer des graphes proche des crawls du Web. Dans le modèle linéaire, à chaque étape un seul sommet est ajouté or dans le Web plusieurs pages Web sont

ajoutées en même temps (le graphe est très dynamique). Le modèle exponentiel permet donc d'ajouter à chaque étape un nombre de sommets qui dépend de la taille du graphe à un instant t et aussi d'autres paramètres (voir au dessus). L'inconvénient majeur de ce modèle est l'absence de la loi de puissance sur les degrés sortants des graphes générés.

2.2.4 Modèles petits mondes

Un graphe **petit monde** (*Small World*) est caractérisé par une distance moyenne faible et un fort coefficient de regroupement. Il existe plusieurs modèles générant des graphes avec une distance moyenne faible et d'autres générant des graphes avec un fort coefficient de regroupement mais il n'existe pas de modèles regroupant les deux propriétés. Plus précisément, on ne sait pas générer de façon aléatoire des graphes petits mondes ayant un coefficient de regroupement donné.

L'anneau de Watts et Strogatz

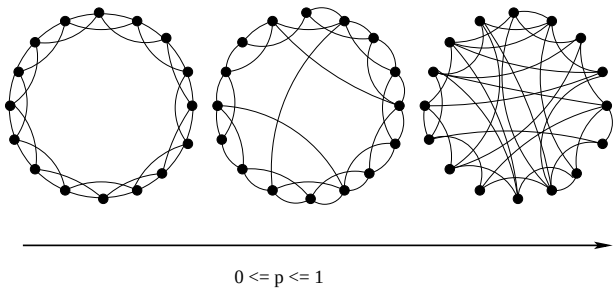


FIG. 2.5 – *Anneau de Watts et Strogatz.*

Watts et Strogatz [104] proposent une méthode pour générer des graphes petits mondes, partant d'un anneau régulier à n sommets où chaque sommet est relié à ses $2k$ plus proches voisins (k voisins de chaque coté). Le coefficient de regroupement d'un sommet u de l'anneau régulier est déja assez important :

$$C(u) = \frac{3(k-2)}{4(k-1)}$$

Quant à la distance moyenne, dans un anneau régulier, elle est très élevée (de l'ordre de n).

L'idée de Watts et Strogatz est de modifier suffisamment l'anneau régulier en déplaçant les arêtes afin de faire baisser la distance entre les sommets (créer des raccourcis). Concrètement, pour chaque sommet et pour chaque arête, avec une probabilité p, l'arête est re-dirigée vers un autre sommet choisi aléatoirement et uniformement et avec une probabilité $1-p$ l'arête est gardée (voir figure 2.5).

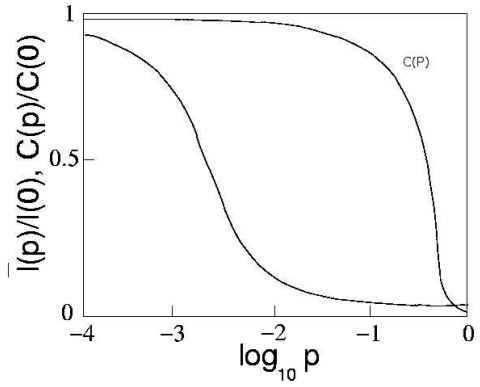


FIG. 2.6 – *Évolution de la distance moyenne et du coefficient de regroupement en fonction de la probabilité de re-direction des arêtes [104].*

La figure 2.6 montre l'évolution de la distance moyenne et du coefficient de regroupement en fonction de la probabilité p. On observe que le coefficient de regroupement est constant dans l'intervalle où la distance moyenne décroît. Cette dernière remarque permet de dire que pour une certain valeur de p, il est possible d'avoir un anneau avec une distance moyenne faible et un fort coefficient de regroupement. Enfin, la distribution des degrés dans l'anneau de Watts et Strogatz suit une loi de Poisson.

La grille de Kleinberg

Contrairement à Watts et Strogatz, Kleinberg [59] utilise une grille pour reproduire le phénomène *petit monde*. La méthode de Kleinberg consiste à poser les sommets sur une grille à deux dimensions (voir figure 2.7). Chaque sommet est identifié par sa position (i,j) dans la grille tel que $i,j \in 1,2,\ldots,n$. La distance entre deux sommets est définie par : $d((i,j),(k,l)) = |k-i| + |l-j|$ (distance de Manhattan).

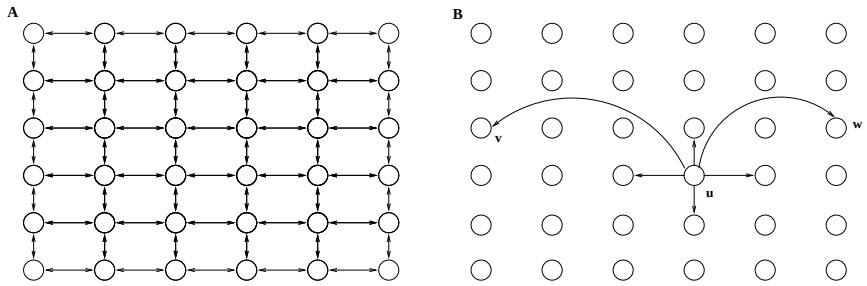


FIG. 2.7 – *(A) Grille à deux dimensions avec $n = 6$, $p = 1$ et $q = 0$. (B) Voisinage de u avec $p = 1$ et $q = 2$. v et w sont des voisins éloignés.*

L'idée de Kleinberg est de relier un sommet à tous ses proche voisins et à quelques sommets lointains. La première démarche permet d'augmenter la valeur du coefficient de regroupement (la densité du 2-voisinage est plus importante que celle du voisinage directe) et la deuxième de diminuer la distance moyenne. Pour cela, un sommet u est relié à tous les sommets à distance p ($p \geqslant 1$). C'est derniers sont appelés les voisins proches. Le sommet u est également relié à q ($q \geqslant 0$) sommets éloignés. La probabilité qu'un sommet v soit choisi comme sommet éloigné est $P(v) = \frac{[d(u,v)]^{-r}}{\sum_j [d(u,j)]^{-r}}$.

Kleinberg montre que lorsque $r = 2$, un algorithme de routage glouton, trés simple et décentralisé, calcule un chemin de longueur polylogarithmique en la taille du graphe ($O(ln^2 n)$) pour n noeuds) entre toute paire de sommets. Schabanel et Lebhar [68] propose un algorithme en $O(\ln n \ \ln\ln n)$.

Modèle biparti de Guillaume et Latapy.

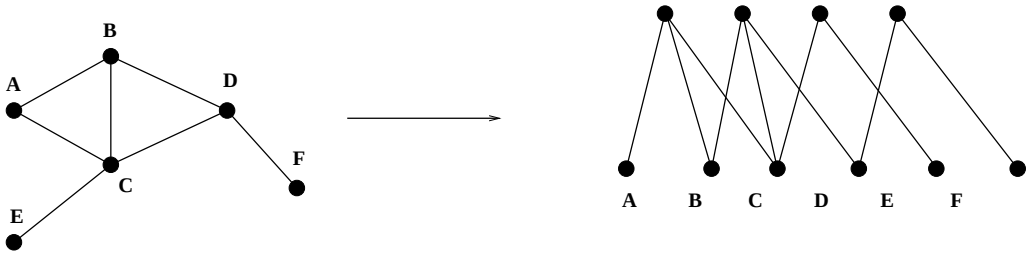


FIG. 2.8 – *Graphe classique et la vision biparti du même graphe (en haut les cliques maximales, en bas les sommets.*

Guillaume et Latapy [52] proposent deux modèles aléatoires pour les réseaux d'interactions qui peuvent être représentés naturellement par des graphes bipartis. Par exemple, le graphe des Acteurs est un graphe dont les sommets sont les acteurs de cinéma qui sont reliés s'ils ont joué dans un même film, ce graphe peut être vu comme un graphe biparti $G(V_1,V_2,E)$ dans lequel V_1 est l'ensemble des films, V_2 est l'ensemble des acteurs et où on trouve un lien entre un film et un acteur si cet acteur a joué dans le film. Les réseaux de co-auteurs et de co-occurrence sont également des réseaux décomposable en plusieurs bipartis.

Ces réseaux sont appelés *réseaux d'affiliation* par les sociologues car une arête (x,y) exprime l'affiliation d'un individu $x \in V_1$ à une *entité* (organisation,etc.) $y \in V_2$.

Les auteurs observent que la distribution des degrés des trois réseaux réels (Acteurs, co-auteurs et co-occurrence) ont une propriétés commune : la présence de lois de puissance dans les distributions des degrés de V_2. Les distribution des degrés de V_1 sont de deux types : en loi de Poisson pour Co-occurrence et Co-auteurs, en loi de puissance pour Acteurs.

Le premier modèle proposé est le modèle uniforme, il consiste à générer des graphes bipartis aléatoires ayant des distributions de degrés données. La génération peut être faite de la manière

suivante :

1. créer les sommets de V_1 et V_2 et assigner à chacun un degré choisi suivant la distribution ;
2. Dupliquer chaque sommet de V_1 et V_2 autant de fois que son degré ;
3. relier de manière aléatoire les sommets de V_1 aux sommets de V_2.

La figure 2.8 montre comment passer du graphe biparti à la vision classique du même graphe. Effectuer le cheminement inverse, qui consiste à retrouver le graphe biparti est généralement impossible.

Le deuxième modèle est le modèle à attachement préférentiel qui utilise donc l'attachement préférentiel pour relier les sommets de V_1 aux sommet de V_2. Ce modèle a un paramètre λ appelé *taux de recouvrement*, il varie d'un réseau à un autre et représente la proportion de sommets de V_2 préexistants à laquelle un nouveau sommet de V_1 est relié. Le processus de génération du graphe biparti peut être écrit comme suit :

1. ajouter un sommet u dans V_1 et choisir son degré d en accord avec une distribution donnée ;
2. pour chacun des d liens du sommet u, ajouter un lien vers un sommet de V_2 selon l'attachement préférentiel avec une probabilité λ ou, avec une probabilité $1-\lambda$, créer un nouveau sommet dans V_2 et le relier à u.

La distance moyenne d'un graphe $\delta(G(V_1,V_2,E)) = O(log(|V_2|))$ et le coefficient de regroupement d'un sommet u est supérieur à $\frac{1}{2d(u)-1}$. Cette borne est indépendante de la taille du graphe. Donc, le modèle biparti de Guillaume et Latapy génère des graphes avec un fort coefficient de regroupement, une distance moyenne faible et une distribution des degrés en loi de puissance.

Ce modèle est très intéressant dans la mesure où le processus de génération prend en considération la nature du réseaux en question et la manière dont il est construit. Nous y reviendrons dans le chapitre 3.

2.2.5 Autres modèles

Modèle *LCD* de Bollobás *et al.*

Bollobás *et al.* [21] proposent le modèle orienté LCD (Linearized Chord Diagram) très proche du modèle de Barabasi et al. [16]. Un graphe $G(V,E)$ de ce modèle est construit de la manière suivante : à l'étape $t+1$, un sommet u et un arc sont ajoutés au graphe G_t. La probabilité p que u soit connecté au sommet v_i ($1 \leqslant i \leqslant t$) dépend du degré $d(v_i) = d^+(v_i) + d^-(v_i)$ tel que : si $1 \leqslant i \leqslant t-1$ alors $p(d(v_i)) = \frac{d_{G_{t-1}}(v_i)}{2t-1}$ et si $i = t$ alors $p(d(v_i)) = \frac{1}{2t-1}$.

Il est possible d'ajouter $m > 1$ arcs à chaque étape en exécutant l'algorithme précédent m fois. Dans ce cas, les u_1,u_2,u_m sommets ajoutés sont regroupés pour former un unique sommet u.

Le graphe G_t est une arborescence avec des boucles et des arcs multiples. Si $d = d^+ + d^- \leqslant t^{1/15}$ alors la distribution des degrés entrants (respectivement sortants) suit une loi de puissance

d'exposant λ^- (respectivement λ^+) tel que $\lambda = \lambda^+ + \lambda^- = 3$. Il est possible d'étendre le résultat pour des degrés $d > t^{1/15}$. La valeur du diamètre est en $O(\frac{\ln t}{\ln \ln t})$ [21].

Le modèle de Ballobás est intéressant car il est orienté, basé sur l'attachement préférentiel et que les caractéristiques des graphes générés sont connues et prouvables en fonction des paramètres du modèle.

Modèle *CL-del* de Chung *et al.*

Dans le processus de génération du modèle *CL-del* de Chung et al. [35], il est possible d'ajouter et de supprimer les sommets et les arêtes. Un graphe du modèle *CL-del* est construit de la manière suivante : à l'étape $t+1$, avec une probabilité p_1 un nouveau sommet u est ajouté au graphe G_t. m arêtes sont ajoutées pour relier le sommet u au reste du graphe. Un sommet v de G_t est choisi proportionnellement à son degré $d(v)$. Avec une probabilité p_2, une arête seulement est ajoutée dans le graphe G_t. La source et la destination de la nouvelle arête sont choisies avec une probabilité proportionnelle à leurs degrés. Avec une probabilité p_3 ($p_1 > p_3$), un sommet choisi aléatoirement est supprimé du graphe G_t. Enfin, avec une probabilité p_4 ($p_2 > p_4$ et $p_1 + p_2 + p_3 + p_4 = 1$), une arête choisi aléatoirement est supprimée.

Lorsque $t \to \infty$, la distribution des degrés du graphe G_t suit une loi de puissance avec un exposant $\lambda = 2 + \frac{(p_1+p_2)}{(p_1+2p_2-p_3-2p_4)}$. Si $m > \ln^{2+\epsilon}(n)$, et si $p_2 < p_3 + 2p_4$, $2 < \lambda < 3$ alors le diamètre du graphe G_t est $\Delta(G_t) = \ln t$ et la distance moyenne du graphe $L(G_t) = O(\frac{\ln \ln t}{-\ln(\lambda-2)})$. Si $m > \ln^{2+\epsilon} n$, et si $p_2 \geqslant p_3 + 2p_4$, et si $\lambda > 3$ alors le diamètre du graphe G_t est $\Delta(G_t) = \Theta(\ln t)$ et la distance moyenne $L(G_t) = O(\frac{\ln t}{\ln \overline{d}})$ où $\overline{d}$ est le degré moyenne du graphe G_t.

Modèle *CFV* de Cooper *et al.*

Ce modèle est également basé sur l'attachement préférentiel et dans le processus de génération, il est possible d'ajouter et de supprimer des sommets.

Un graphe $G(V,E)$ de ce modèle [37] est construit de la manière suivante :

1. à l'état initial, le graphe contient un sommet ;
2. $|V(G_{t+1})| = |V(G_t)| \pm 1$, tel que :
   - avec une probabilité α_1, un sommet u est ajouté,
   - avec une probabilité $1 - \alpha - \alpha_0$, un sommet choisi aléatoirement est supprimé ;
3. $|E(G_{t+1})| = |E(G_t)| \pm m$, tel que :
   - avec une probabilité α_1, $m > 0$ arêtes sont ajoutées,
   - avec une probabilité α_0, m arêtes sont choisis aléatoirement et supprimées ;
4. une arête est ajoutée de la manière suivante :
   - la probabilité p de choisir le sommet v est : $p(v) = \frac{d(v,t)}{2 \times |E(G_t)|}$ où $d(v,t)$ est le degré de v à l'étape t et $|E(G_t)|$ est le nombre d'arêtes à l'étape t.

La distribution des degrés d'un graphe généré par le modèle *CFV* de Cooper *et al.* suit une loi de puissance de paramètre $\beta = 1 + \lambda$ avec $\lambda = \frac{2(\alpha-\alpha_0)}{(3\alpha-1-\alpha_1-\alpha_0)}$.

Modèle α de Kumar *et al.*

Un graphe $G(V,E)$ de ce modèle [63] est construit de la manière suivante : à l'étape $t+1$, un sommet et un arcs sont ajoutés dans le graphe G_t. Avec une probabilité $1-\alpha$, un arc est ajouté vers le sommet u (une boucle) et avec une probabilité α, un sommet v est choisi aléatoirement. La probabilité p que u soit connecté au sommet v dépend du degré entrant $d^-_{v,t}$ de v tel que :

$$p(v) = \frac{d^-_{v,t}}{t},$$

où $d^-_{v,t}$ est la valeur du degré entrant de v à l'étape t.

Le graphe G_t est une arborescence avec des boucles et des arcs multiples. La distribution des degrés entrants est en loi de puissance avec $\beta_{in} = \frac{1}{\alpha}$.

2.3 Comparaison des différents modèles

Tableau 2.3 : Propriétés des modèles aléatoires pour les réseaux d'interactions.

Modèle	Orienté ?	Dynamique	Loi de puissance	Clustering	Bicliques	β
Barabasi	Oui	Oui	Oui	Non	Non	3
LCD	Oui	Oui	Oui	Oui	Non	3
ACL	Oui	Oui	Oui	Non	Non	$(2,\infty)$
Copie	Oui	Oui	Oui	Non	Oui	$(2,\infty)$
CL-del	Non	Oui	Oui	Oui	Non	$(2,\infty)$
CFV	Non	Oui	Oui	Non	Non	$(2,\infty)$
Biparti	Non	Non	Oui	Oui	Oui	$(2,\infty)$

Le tableau montre qu'il n'existe pas beaucoup de modèles avec toutes les propriétés communes aux réseaux d'interactions. En effet, il est difficile de trouver un modèle qui à la fois génère un graphe avec une distribution des degrés en loi de puissance (scale-free) et un coefficient de regroupement élevé (petit monde).

En effet, le modèle de Watts et Strogatz génère des graphes petits mondes mais avec une distribution des degrés différente d'une loi de puissance. A l'inverse, le modèle de Barabasi, génère un graphe avec une distribution des degrés en loi de puissance mais pas avec un fort coefficient de regroupement.

Seul le modèle Biparti de Guillaume et Latapy semble regrouper les trois propriétés : loi de puissance, diamètre petit et fort coefficient de regroupement. En effet, Guillaume et Latapy ont montré que tous les graphes d'affiliation pouvaient être naturellement transformés en graphes bipartis.

Nous avons présenté un éventail de modèles de graphes aléatoires avec une distribution des degrés en loi de puissance. L'objectif étant de générer des graphes proches des réseaux d'interactions en reproduisant les propriétés communs à ces derniers.

Cette objectif est difficile à atteindre dans la mesure où tous les modèles exposés dans ce chapitre n'arrivent pas à regrouper toutes les propriétés communes des réseaux d'interactions à l'exception du modèle de Guillaume et Latapy.

Il est difficile de prouver qu'un modèle «compliqué» (tel que le modèle biparti de Guillaume et Latapy ou celui qui sera proposé au chapitre 3) possède des caractéristiques donnée (c'est-à-dire qu'un graphe généré par ce modèle aura presque sûrement telle propriété). C'est à cause de la **dépendance** entre les arêtes, qui rend les preuves difficiles, voire *impossibles* [83, 86, 87].

Nous nous sommes particulièrement intéressé au problème de la modélisation des réseaux d'interactions. Dans le chapitre qui suit, nous avons proposé un modèle qui permet de regrouper toutes les propriétés d'un crawl du Web.

Chapitre 3

Un modèle de crawls aléatoires du Web

Il est généralement difficile d'obtenir un réseau d'interaction en entier. On doit souvent se contenter d'un sous-réseau de taille assez grande (la plus grande possible) afin de mener différentes études. Certains de ces sous-réseaux sont obtenus en utilisant des outils, par exemple pour le Web, on utilise un crawler pour obtenir un sous-graphe du Web (un crawl) [75].

Nous essayons ici de prouver que certaines propriétés communes des **sous-réseaux** d'interactions sont en réalité des artefacts dus aux outils utilisés lors de la capture des sous-réseaux. Dans notre démarche, nous faisons une distinction entre le crawl du Web obtenu par un *parcours du Web* et le Web tout entier qui est impossible à définir (à cause de la dynamique entre autre) et donc impossible à obtenir.

Suite à l'analyse de différents *crawls* du Web, plusieurs modèles ont été proposés (voir chapitre 2 page 27) . On peut reprocher à ces derniers d'être des modèles de graphe du Web alors que l'objet modélisé n'est pas accessible (voir chapitre 1 page 13). Il y a donc lieu d'essayer de modéliser ce que nous en connaissons : *les crawls*.

Dans ce chapitre, nous proposons un modèle de *crawls* aléatoires du Web. Le but est que les graphes générés par ce modèle possèdent les mêmes propriétés observées sur les crawls du Web. Le modèle de crawls aléatoires est basé sur la simulation de parcours du Web, cette méthode de construction est inspirée de la démarche réelle utilisée pour construire un crawler, c'est-à-dire, parcourir le Web (ou plutôt une partie du Web).

Bien que la méthode semble assez simple, l'analyse du modèle, quant à elle, s'avère très difficile. En effet, il n'est pas possible d'utiliser les résultats obtenus sur les graphes aléatoires puisque dans notre modèle, les arêtes ne sont pas indépendantes les unes des autres. Néanmoins, en utilisant des outils de visualisation de graphes, nous pouvons faire une comparaison visuelle entre les différents réseaux réels ou simulés.

Dans la première section, nous donnons une simple définition d'un *crawler* (le programme permettant de parcourir le Web à la recherche de nouvelles pages Web). Nous décrivons ensuite les caractéristiques de notre modèle permettant de générer des crawls aléatoires du Web. Nous utilisons la décomposition en k-core pour la construction d'une image de la structure du graphe par le biais d'un outil de visualisation LaNetVi [11]. À partir de cette image, il est possible

de faire une comparaison avec des images d'autres réseaux d'interactions. Enfin, nous vérifions expérimentalement que les graphes obtenus ont les mêmes propriétés que les crawls réels du Web.

Ce travail a été réalisé avec Fabien de Montgolfier (maître de conférence à l'université Paris 7) et n'a pas encore fait l'objet d'une publication.

3.1 Le processus de crawl

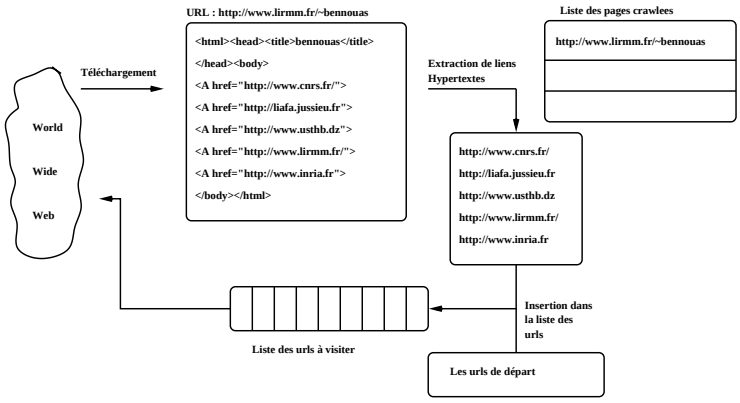


FIG. 3.1 – *Schéma simplifié d'un crawler*

Tous les moteurs de recherche disposent d'un programme qui parcourt le Web à la recherche de nouvelles pages Web, il est appelé *crawler*.

Le processus de crawl (*crawling*) consiste à choisir une URL [1] parmi l'ensemble des URLs à visiter. Ce choix dépend de la stratégie de parcours adoptée (voir figure 3.1).

Il existe plusieurs stratégies de crawl, nous citons les plus classiques : le parcours en largeur (*BFS*) et le parcours en profondeur (*DFS*). Il existe aussi d'autres types de parcours, tels que le parcours suivant le degré entrant maximum (*DEG*), où à chaque étape le sommet le plus cité est choisi en premier, ou encore le parcours aléatoire (*RAND*) dans lequel, un sommet est choisi au hasard parmi tous les sommets déjà cités (un mélange entre le parcours en largeur et le parcours en profondeur).

Après avoir choisi une URL parmi l'ensemble des URLs à visiter (suivant la stratégie adoptée), le *crawler* télécharge la ressource (page Web), extrait les liens hypertextes (sous forme d'URLs) de la page. Les URLs non encore visitées sont insérées dans la liste des pages Web à visiter. Ce processus est réitéré jusqu'à ce que la liste des URLs à visiter soit vide ou que le processus soit arrêté volontairement. Comme le Web est gigantesque, la liste des URLs à visiter

1. Uniform Ressources Locator, c'est une adresse unique d'une ressource sur Internet [**?**].

n'est jamais vide. Par conséquent, c'est généralement la deuxième solution ; à savoir l'arrêt du crawler ; qui est adoptée.

Le *crawling* permet donc de construire un graphe du Web généralement appelé *crawl* du Web. Ce dernier peut être modélisé par un graphe $G(V,E)$.

Dans le chapitre 1, nous avons étudié les propriétés de ce crawl. Notre but est de générer des graphes aléatoires avec les mêmes propriétés que les crawls du Web.

3.2 Objectifs

Un *bon* modèle de crawl du Web doit selon nous reproduire les lois de distributions observées dans les crawls réels, et si possible émuler les modes de génération de pages Web et de crawl du Web (êtres aussi conforme à la réalité que possible). Ainsi le modèle doit produire des graphes :

1. ayant un fort coefficient de regroupement ;
2. dont les degrés suivent une loi de puissance ;
3. ayant une faible distance moyenne ;
4. connexe a source unique (un sommet de départ) ou multiples (plusieurs sommets de départ) ;
5. avec beaucoup de sommets sans successeurs ;
6. tels que les sommets de forte connectivité (de fort PageRank dans le crawl final) aient été découvert tôt dans le processus de crawl.

Les trois premiers points sont des paradigmes classiques des réseaux d'interactions; les trois derniers sont typiques des crawls. Dans le but de satisfaire ses six paradigmes, nous proposons un modèle de génération de graphes aléatoires dont les caractéristiques sont décrites dans la prochaine section.

3.3 Description du modèle

Notre modèle peut être décomposé en deux parties. La partie *pré-calcul* où *primo*, on attribue aléatoirement à chaque sommet un degré entrant et sortant et *secondo*, on construit une liste de *voisinage* donnant l'ordre de découverte du crawl. Et la partie *crawl* où une stratégie de parcours est adoptée et une simulation de parcours est réalisée.

3.3.1 Les stratégies de crawls

Le crawl débute à partir d'un sommet choisi aléatoirement. Chaque sommet est visité une seule fois. Si les successeurs d'un sommet n'ont pas encore été visités alors ils sont marqués.

Contrairement à un crawl réel, tous les sommets sont accessibles. A chaque étape du parcours, les sommets sont divisés en trois ensembles :

1. *visités* : contient tous les sommets visités (crawlées) et leurs successeurs sont connus ;
2. *marqués* : contient les sommets connus mais pas encore visités ;
3. *non visités* : contient les sommets qui n'ont pas encore été marqués ou visités.

L'algorithme de crawl choisi un sommet u de la liste des sommets *marqués*, les successeurs de u non visités sont insérés dans la liste des sommets marqués. Les sommets crawlés sont ordonnés par date de découverte. Le choix du sommet à visiter dépend de la stratégie de parcours adoptée. La stratégie dépend de la façon dont la liste des sommets marquées est gérée :

– DFS (depth-first search) La stratégie est LIFO et la structure de données est une pile ;
– BFS (breadth-first search) la stratégie est FIFO et la structure de données est une file ;
– DEG (higher degree) le sommet le plus pointé est choisi. La structure de données est un tas ;
– RAND (random) le sommet est choisi de façon aléatoire et uniforme.

3.3.2 Génération de crawl

En phase de pré-calcul, on associe aléatoirement un degré entrant et sortant à chaque sommet. On fait en sorte que la distribution des degrés suit une loi de puissance de paramètres données (voir figure 3.2).

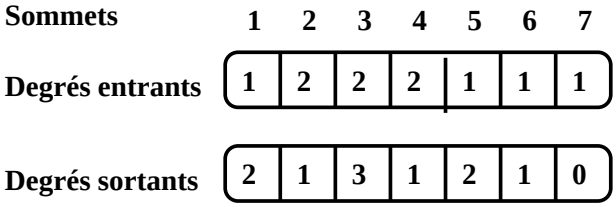


FIG. 3.2 – *Distribution en lois de puissance pour les degrés entrants et sortants et affectation aléatoire des degrés.*

Pour assurer que la distribution des degrés de tout sous crawl suit également une loi de puissance avec les mêmes paramètres, nous dupliquons chaque sommet un nombre de fois égal à son degré entrant et nous insérons les sommets dans une liste de *voisinage* que nous mélangeons de manière aléatoire (voir figure 3.3).

L'algorithme de génération de crawl est tout simplement :

1. Supprimer un sommet p de la liste des sommet *marqués* selon la stratégie adoptée (DFS, BFS, DEG ou RAND) ;

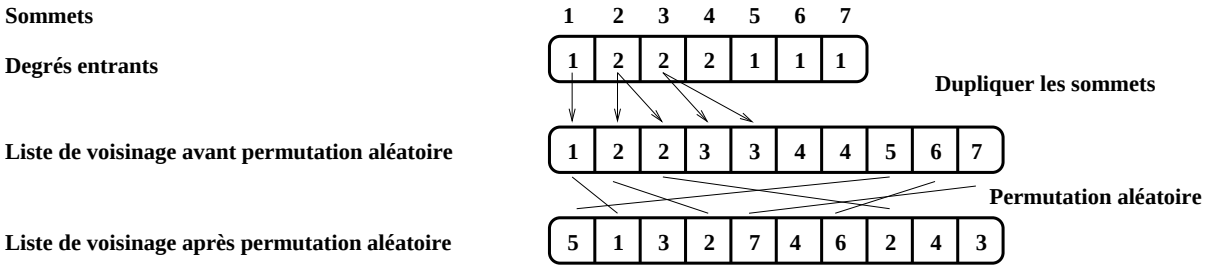


FIG. 3.3 – *Permutation aléatoire de la liste de voisinage.*

2. Les $d_{out}(p)$ sommets en tête de la liste de voisinage sont les successeurs de p ;
3. Ajouter les successeurs de p non-marqués à la liste des sommets marqués ;
4. aller à 1 ;

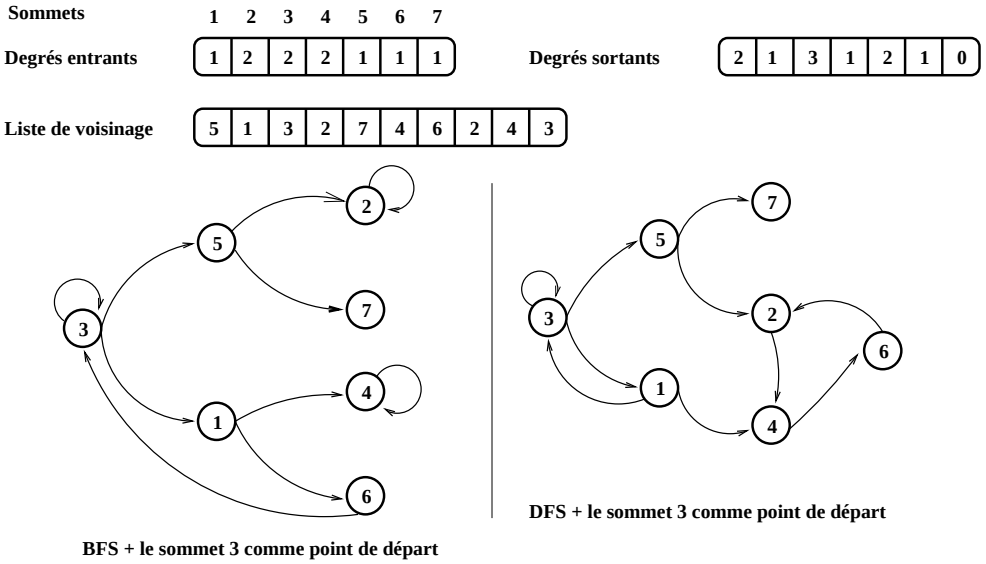


FIG. 3.4 – *Exemple de crawl avec BFS et DFS.*

Le processus de génération ne s'arrête que lorsque la liste des sommets marqués est vide ou encore que tous les sommets sont visités et donc la liste de voisinage est vide. La figure 3.4 donne un exemple à 7 sommets d'un crawl avec BFS et DFS. Les graphes générés peuvent avoir des boucles et des multi-arcs.

Notre modèle est différent du modèle d'attachement préférentiel (voir § 2.2.2 page 31) ou du modèle de copie (voir § 2.2.3 page 32) du fait de la liste de voisinage est pré-calculée, donc les extrémités finales des arcs sont connues avant le parcours. Le processus de crawl affecte tous simplement à chaque arc l'extrémité de départ.

Bien que simple à implémenter, notre modèle est difficile à analyser. En effet, comme les arêtes du graphe sont dépendantes les unes des autres, les résultats obtenus sur les graphes aléatoires en loi de puissance (voir § 2.2.1 page 30) ne peuvent être utilisés dans notre cas.

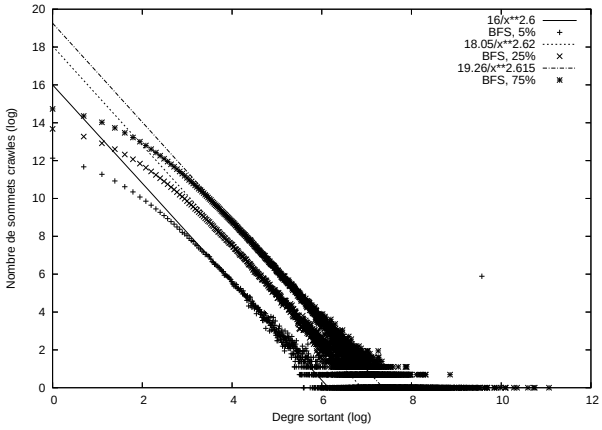
Dans la prochaine section, nous allons donner les propriétés de nos crawls observées lors des différentes expérimentations.

3.4 Résultats

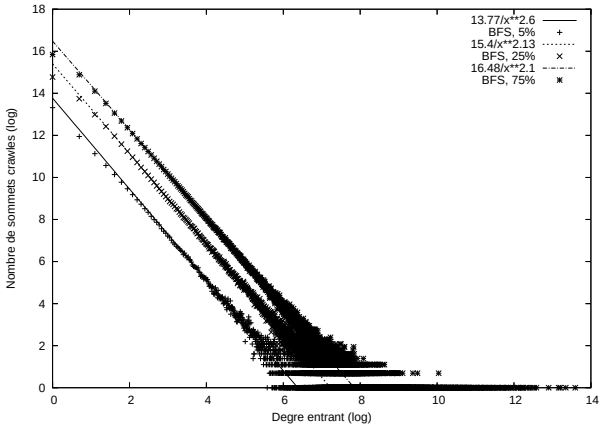
Nous présentons dans cette section quelques caractéristiques des crawls générés par notre modèle. Ces résultats sont issus de simulations sur ordinateur. Nous insistons sur l'évolution de certaines mesures au cours de la progression du crawl. Nous avons généré des crawls de différentes tailles mais avec une distribution des degrés qui suit une loi de puissance de paramètres $\lambda_{in} = 2.1$ et $\lambda_{out} = 2.72$. Ces derniers ont été observés expérimentalement dans [28]. Tous les graphes générés possèdent un nombre négligeable de boucles et de multi-arcs. Un sous-crawl de taille t est constitué des t premiers sommets visités lors de la simulation de crawl. Les sommets marqués sont supprimés.

3.4.1 Crawls à caractère sans-échelle (Scale-Free)

Nous avons généré un crawl de 20 millions de sommets et de 1,4 milliard d'arcs. La figure 3.5, montre les distributions des degrés sortants et entrants de sous-crawls de 1,000,000 (5%), 5,000,000 (25%) et 15,000,000 (75%) de sommets.



Distribution des degrés sortants



Distribution des degrés entrants

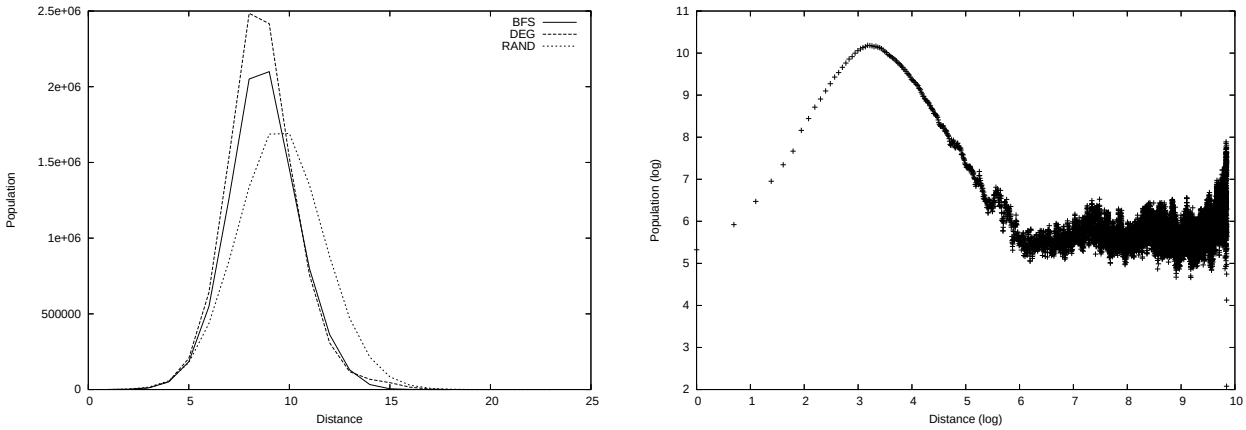
FIG. 3.5 – *Distribution des degrés.*

Indépendamment de la stratégie utilisée, la distribution des degrés suit une loi de puissance. Les sous-crawls respectent une propriété importante observée sur le Web : la propriété *scale-*

free, c'est-à-dire, chaque sous-crawl garde la loi de puissance sur les degrés avec les mêmes paramètres.

3.4.2 Évolution de la distance moyenne et diamètre

Afin d'analyser l'évolution de la distance entre tous couples de sommets, nous avons généré un crawl de 500,000 sommets et 3,500,000 arcs.



BFS, DEG et RAND DFS

FIG. 3.6 – *Distribution des distances entre couple de sommets*

Comme le montre la figure 3.6, la distribution des distances suit une loi gaussienne pour les trois stratégies BFS, DEG et RAND. Cette distribution est calculée après un crawl de 150,000 sommets mais cette dernière persiste jusqu'à la fin du crawl (voir figure 3.7). Pour le DFS, le crawl obtenu contient de grandes distances entre sommets et la distribution ne ressemble pas à une loi connue. Ceci s'explique de fait que le DFS va chercher en premier les sommets puits et donc construit des branches longues.

La figure 3.7 montre l'évolution de la distance moyenne et du diamètre au cours du crawl. Pour le BFS, DEG, RAND, ces deux paramètres évoluent très lentement. Pour la distance moyenne sa valeur est autour du degré moyen, qui est de 7. Quant au diamètre, sa valeur est supérieure au degré moyen. Pour le DFS, le diamètre est grand. Ceci s'explique du fait que le DFS construit un arbre profond.

3.4.3 Évolution du coefficient de regroupement

Pour analyser l'évolution du coefficient de regroupement [2] (clustering), nous avons utilisé le crawl de 500,000 sommets et 3,500,000 arcs de la section 3.4.2. Le coefficient de regroupement

2. voir § 1.1 page 13

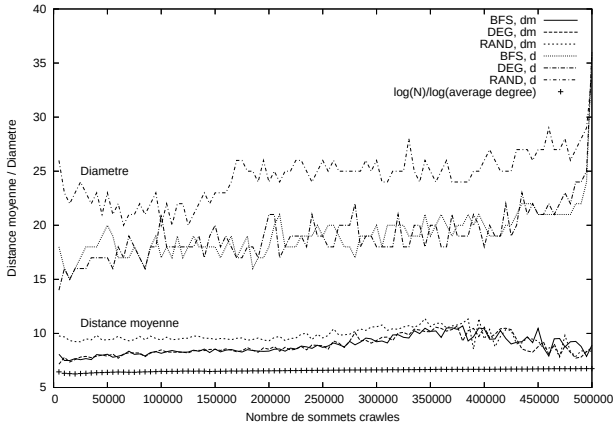


FIG. 3.7 – *Évolution du diamètre et de la distance moyenne au cours du crawl.*

se calcule sur la symétrisation du crawl (il n'est défini que pour les graphes non orientés).

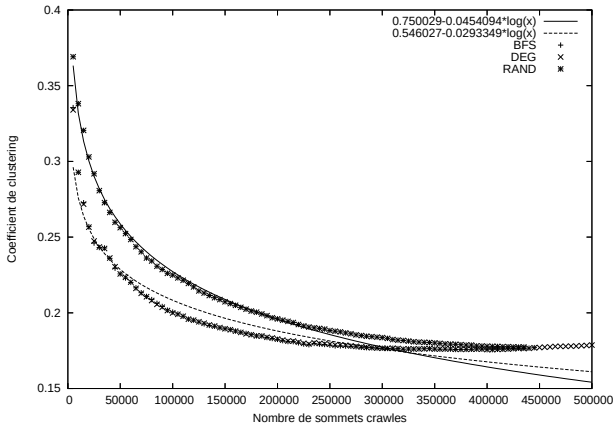
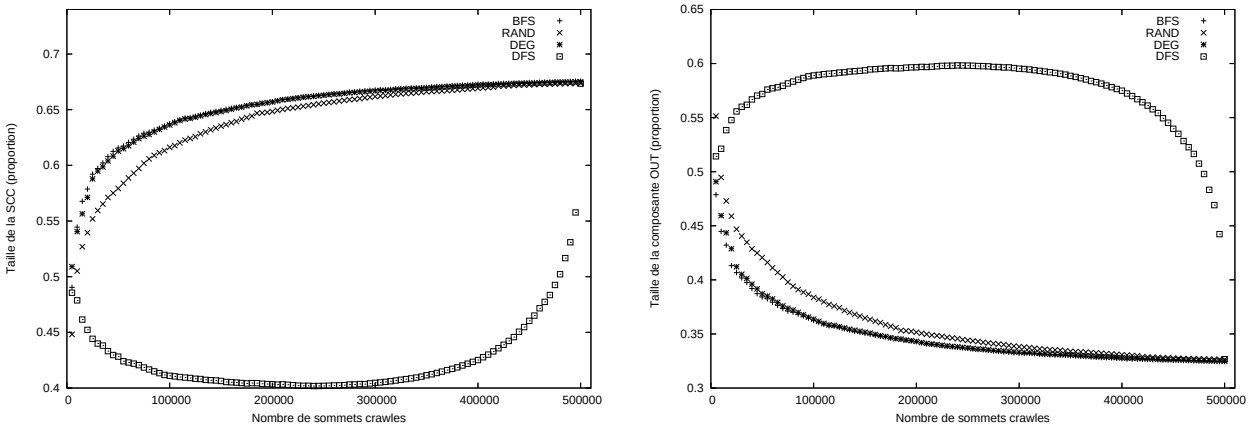


FIG. 3.8 – *Évolution du coefficient de regroupement.*

Le coefficient de regroupement décroît au cours du crawl (voir figure 3.8) mais semble converger vers une constante. Mais on observe qu'il est nettement supérieur au coefficient de regroupement d'un graphe aléatoire d'Erdös et Rényi. Cette observation est très importante, nous pouvons dire que la stratégie de parcours joue un rôle important pour la connectivité du crawl et donc pour la valeur du coefficient de regroupement des crawls. Le DFS construit un graphe très arborescent, par conséquent, le coefficient de regroupement est très faible.

3.4.4 Taille des composantes SCC et OUT

Nous nous sommes intéressé également aux différentes composantes du crawl au sens du modèle en noeud papillon [28]. Notamment à la composante fortement connexe géante (*SCC*) et à la composante *OUT* du crawl. Cette dernière contient des sommets accessibles à partir de la composante SCC et pas l'inverse (voir § 1.3.5 page 22).



Composante fortement connexe composante OUT

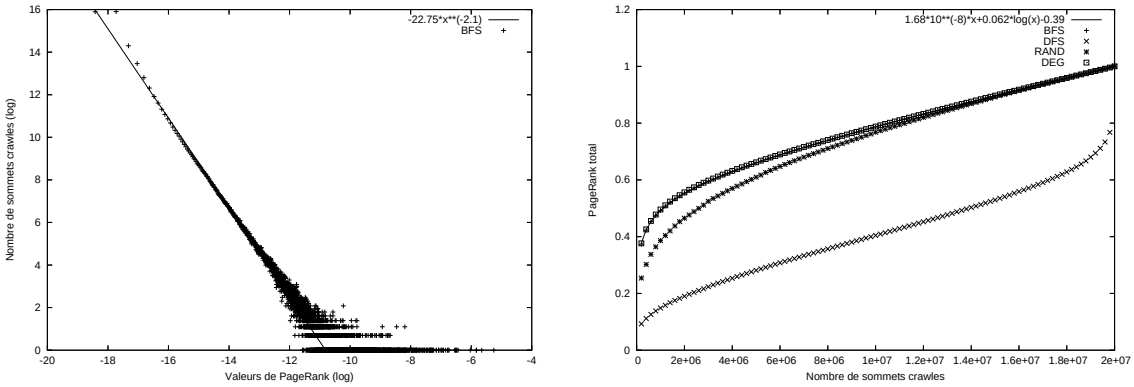
FIG. 3.9 – *Évolution des différentes composantes d'un crawl de 500,000 sommets.*

La figure 3.9 montre l'évolution de la SCC et de la composante OUT au cours d'un processus de crawl. On observe que le taille de la SCC augmente jusqu'à atteindre plus de 65% du graphe. Inversement, la taille de la composante OUT diminue mais n'est pas négligeable.

On observe que nos crawls ne contiennent pas ou très peu de composante *IN*. Un sommet de la composante IN a accès à tous les sommets des composantes SCC et OUT et pas l'inverse. La composante IN est plutôt réduite à un sommet, voire à un ensemble vide. Cela est dû au fait que les crawls ne contiennent pas beaucoup de sources (points de départ du crawl).

3.4.5 Distribution du PageRank

Le PageRank est un algorithme qui permet de donner un rang (une valeur réelle) à chaque sommet d'un graphe. Ce rang représente la probabilité qu'un surfeur aléatoire accède à ce sommet. Pour plus de détail sur le PageRank (voir § 4.2 page 68). Nous avons donc calculé le PageRank sur un crawl de 20,000,000 sommets. Et nous nous sommes intéressé à l'évolution du PageRank au cours du crawl. Plus précisément, nous nous sommes intéressés à la valeur totale du PageRank accumulé au cours du crawl dans les différentes stratégies de parcours. Ceci afin de tester l'affirmation couramment admise que les pages de fort PageRank sont découvertes rapidement lors d'un crawl [4].



Distribution du PageRank Le PageRank total accumulé

FIG. 3.10 – *Évolution du PageRank*

La figure 3.10 montre que la distribution des valeurs de PageRank suit une loi de puissance de même paramètre que celle des degrés entrants du graphe, c'est-à-dire 2.1. Cette caractéristique a été observée dans [91]. On observe également que BFS et DEG accumulent dès 10% du crawl plus de 60% du PageRank total. Si le PageRank est utilisé pour mesurer la pertinence d'un sommet alors on peut dire qu'un BFS permet d'obtenir des sommets *pertinents* rapidement. Cette caractéristique du BFS a été observée par [82]. [75] note qu'il existe une assez forte corrélation entre le PageRank et les degrés entrants.

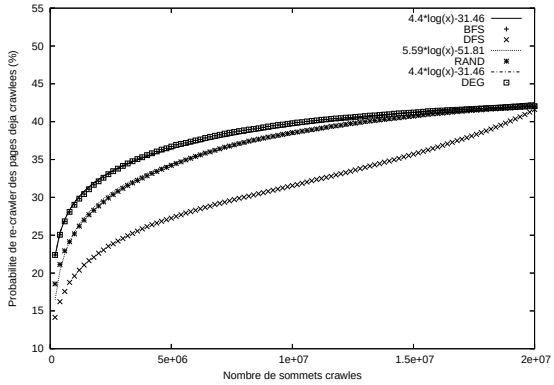
3.4.6 Proportion de sommets revisités et puits

En générant un crawl de 20,000,000 sommets, nous avons calculé la proportion de sommets revisités (c'est-à-dire, des sommets de la liste de voisinage déjà marqués ou visités au moment de la visite du sommet p) et la proportion de puits (au moment de la visite du sommet p).

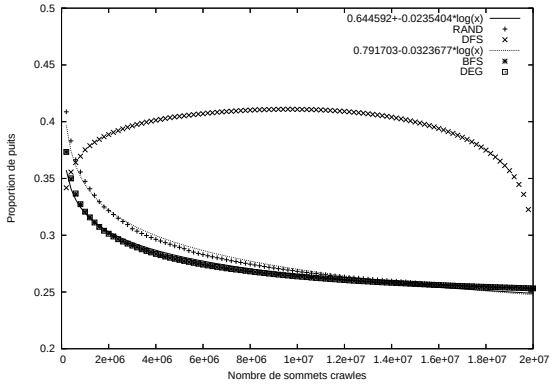
La figure 3.11 montre que 35% des sommets sont revisités après un crawl de 3,500,000 sommets. Ce taux augmente lentement jusqu'à atteindre un peu plus de 40% à la fin du crawl. À *contrario*, cela veut dire que 60% des sommets sont nouveaux, donc que le crawl s'étend vers l'inconnu [89]. Quant à la quantité de puits, cette dernière décroît progressivement pour atteindre un taux de 25%. Donc, un quart des sommets du crawl sont des puits. Cette caractéristique a été observée par [28]. On remarque que le DFS trouve les puits rapidement (courbe de droit) et donc par conséquent revisite peu (courbe de gauche).

3.4.7 Énumération des bicliques

Enfin, nous nous sommes intéressés à l'énumération des bicliques (voir § 1.3.6 page 25). Ce problème étant NP-complet, nous nous sommes intéressés particulièrement aux bicliques de taille (4,4) et cela sur un crawl de 10000 sommets.



Évolution de la proportion de pages Web revisitées



Évolution de la proportion de puits

FIG. 3.11 – *Évolution de la proportion de pages Web revisitées et de la proportion de puits.*

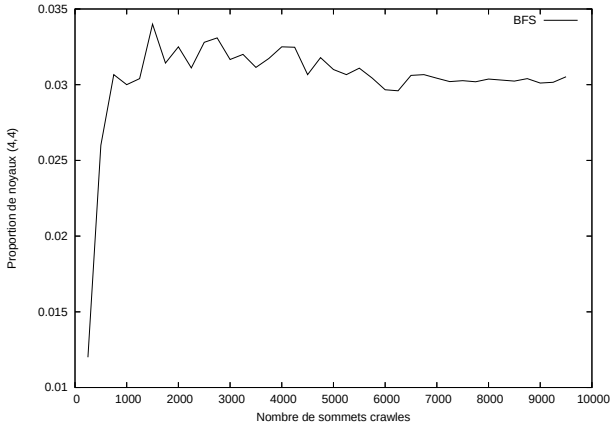


FIG. 3.12 – *Évolution des noyaux de taille 4,4*

La figure 3.12 montre que la proportion de bicliques de taille (4,4) augmente rapidement et est de l'ordre de 3% (c'est-à-dire, le nombre de bicliques disjointes par rapport au nombre de sommets du crawl) et cela après avoir crawlé 1000 sommets. Cette proportion reste constante jusqu'à la fin du crawl. Nous avons observé que nos crawls contiennent également d'autres bicliques de différentes tailles (voir le tableau 3.4.7).

Tableau 3.4.7 : Énumération des bicliques dans un crawl de 10000 sommets.

Hubs	Auth.	# bicliques
2	2	220
2	3	83
2	4	37
2	5	14
2	6	14
2	7	14
3	3	84
3	4	37
3	5	14

Hubs	Auth.	# bicliques
3	6	5
3	7	2
4	4	40
4	5	14
4	6	5
4	7	2
5	5	12
5	6	3
6	6	3

Les récents travaux dans le domaine de la visualisation des réseaux d'interactions permettent de faire une comparaison (visuellement) entre les différentes propriétés des réseaux réels et simulés. C'est l'objet de la prochaine section.

3.5 La visualisation de graphes

Depuis quelques années, plusieurs travaux ont été réalisés sur le thème de la visualisation des grands réseaux d'interactions. Cette dernière permet, d'une part, de détecter différentes structures dans les graphes telles que la structure de communauté (cluster) et d'autres part, elle permet de faire une comparaison (visuelle) entre les différents réseaux réels ou simulés. C'est pour cette deuxième raison que nous nous sommes intéressés à l'outil de visualisation développé par Alvarez-Hamelin *et al.* [11] appelé LaNetVi [3] (Large Network Visualization). Il permet d'obtenir une image en deux dimensions de la *décomposition en k-core* d'un graphe. Nous l'avons utilisé pour comparer des crawls réels avec les crawls aléatoires de notre modèle ou d'autres modèles.

3.5.1 La décomposition en k-core

La décomposition en k-core [17] consiste à identifier des sous ensembles particuliers du graphe appelés *k-core*.

Soit un graphe $G(V,E)$ à $|V| = n$ sommets et $|E| = e$ arêtes. Un k-core est défini comme suit :

Définition 2
Un sous-graphe $H = G[C]$, est un k-core ou un core d'ordre k si et seulement si $\forall v \in C$: $degre_H(v) \geqslant k$, et H est un sous ensemble maximal avec cette propriété.

Définition 3
Notons que le k-core est unique [11].

3. http://xavier.informatics.indiana.edu/lanet-vi/

Définition 4
Un sommet i a un «coreness» de c s'il appartient à l'ensemble c-core mais pas à l'ensemble $(c+1)$-core. Le «coreness» du sommet i est noté par c_i.

Définition 5
L'ensemble de tous les sommets de coreness c est noté C_c. La valeur maximale de c telle que C_c est non vide est notée c_{max}.

Définition 6
L'ensemble connexe de coreness c forme un cluster (une communauté) au sens de [11].

Le k-core de G peut être obtenu par une suppression récursive de tout les sommets de degré inférieur à k. Le graphe restant ne contient que des sommets de degré supérieur ou égal à k (voir figure 3.13).

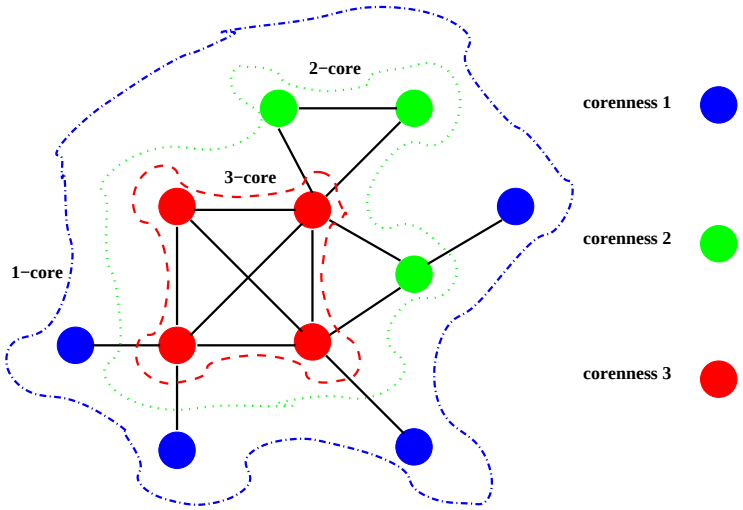


FIG. 3.13 – *Décomposition en k-core d'un petit graphe. Chaque cercle contient un ensemble de sommets appartenant à un k-core. Chaque sommet du graphe connexe appartient à l'ensemble 1-core. Les différents cores sont entourés par des lignes de différentes couleurs. La ligne bleue englobe tous les sommets appartenant à l'ensemble 1-core (tous les sommets du graphe). Pour calculer les sommets de l'ensemble 2-core, tous les sommets de degré $d < 2$ sont supprimés de manière récursive. Tous ces sommets sont colorés en bleu. Les autres sommets restent avec des degrés $d \geqslant 2$ même après suppression des sommets bleus et donc ne sont pas éliminés. Les sommets restants (verts et rouges) entourés par un trait vert forment l'ensemble 2-core. Une étape de suppression supplémentaire permet d'identifier l'ensemble le plus profond, le 3-core. Il est facile de vérifier que tous les sommets rouges ont un degré au moins de 3. Ce core est entouré par une ligne rouge.*

La décomposition en k-core permet d'obtenir un partitionnement hiérarchique des sommets tel que l'ensemble de coreness 1 se trouve en haut de la hiérarchie et l'ensemble de coreness c_{max} est en bas de la hiérarchie. Cette partition dépend du degré de chaque sommet et des degrés dans le voisinage. La complexité en temps de l'algorithme de décomposition en k-core de Alvarez-Hamelin *et al.* est $O(n+m)$ où n et m sont respectivement la le nombre de noeuds et d'arêtes dans le réseau [11].

Alvarez-Hamelin *et al.* [11] ont développé l'outil de visualisation LaNetVi permettant d'avoir une image de la décomposition en k-core.

3.5.2 Description de l'outil de visualisation

L'algorithme de visualisation décrit dans Alvarez-Hamelin *et al.* [11] place les sommets dans un espace à 2 dimensions. La position de chaque sommet dépend de son coreness et de celles de ses voisins. Chaque sommet a une couleur permettant d'identifier sa valeur de coreness. Pour chaque valeur de coreness c, un cercle est dessiné de rayon proportionnel à $(c_{max} - c)$. Tous les cercles ont le même centre. Ainsi on obtient un ensemble de cercles imbriqués les uns aux autres. Le cercle de plus petit rayon correspond à la plus grande valeur de coreness (c_{max}). Chaque sommet a une taille proportionnelle à son degré.

L'image obtenue permet d'avoir une structure hiérarchique du réseau, elle permet également de retrouver les propriétés de connexité et de clustering de chaque core. Il est possible visuellement de trouver les relations entre les différents core, de rechercher l'inter-connexité de chaque core et enfin de connaître l'existence d'une corrélation entre les degrés et les coreness (voir figure 3.14).

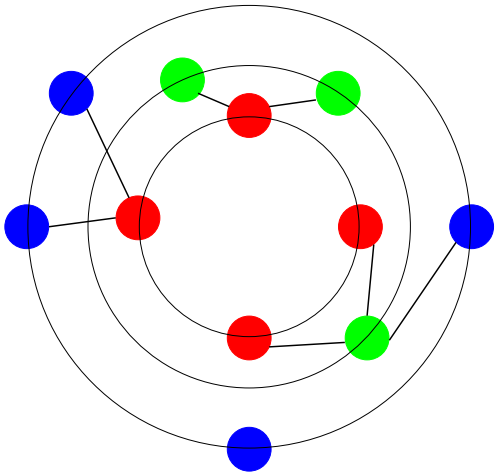


FIG. 3.14 – *Image du graphe de la figure (3.13) après l'utilisation de LaNetVi. Les droites entre les sommets ne représentent rien d'autre que les arêtes entre sommets dans le graphe d'origine. L'ensemble 3-core forme une communauté (cluster).*

3.5.3 Comparaison entre les réseaux

Nous avons utilisé l'outil de visualisation développé par Alvarez-Hamelin *et al.* [11] afin de mener une etude comparative entre l'empreinte (image) de nos crawls aléatoires avec celles de plusieurs réseaux réels et simulés. Il est clair que deux images très différentes signifient que les deux réseaux ont des caractéristiques différentes. Dans le cas contraire, on ne peut rien dire.

Nous avons généré un crawl aléatoire en utilisant notre modèle décrit plus haut dans ce chapitre. Par la suite, nous avons utilisé l'outil de visualisation LaNetVi pour construire une image de notre graphe. Nous allons faire une comparaison entre l'image de notre crawl avec celles de différents réseaux réels ou simulés.

Réseaux d'Internet

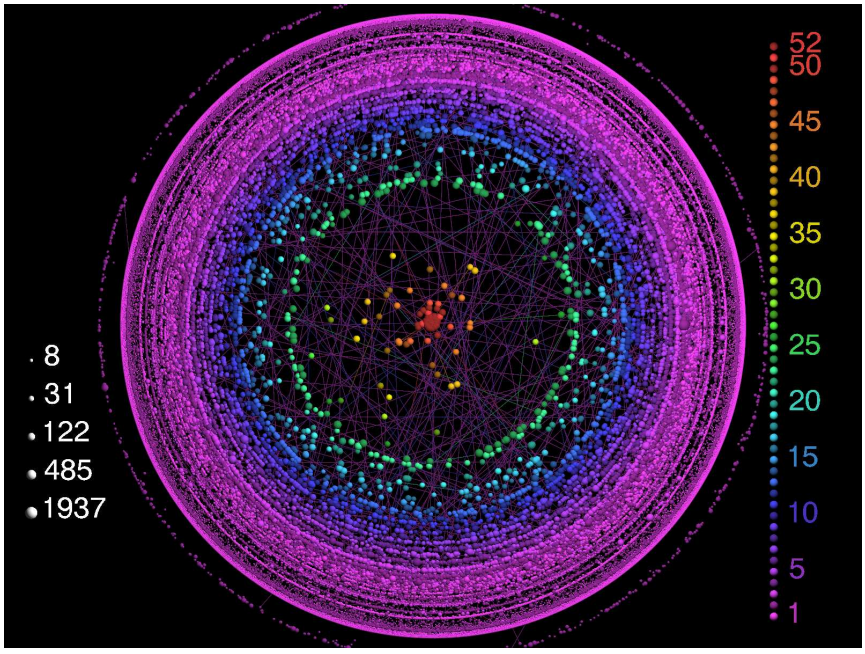


FIG. 3.15 – *Réseaux de routeurs.*

La figure 3.15 montre une image de la décomposition en k-core du réseau de routeurs IR fournit par Govindan et Tangmunarunkit [50]. C'est un graphe de 150000 sommets (routeurs) et 200000 liens (connections). La coreness de graphe est de 52 (voir en haut à droite). La valeur maximale du degré est de 1937 (voir en bas à gauche).

On observe des sommets de fort degré dans chaque cercle. Cela signifie qu'il n'existe pas une forte corrélation entre les degrés et les valeurs de coreness. Cette caractéristique est propre au réseaux des routeurs et au crawls du Web. Dans ces derniers, il n'existe pas de structure hiérarchique globale.

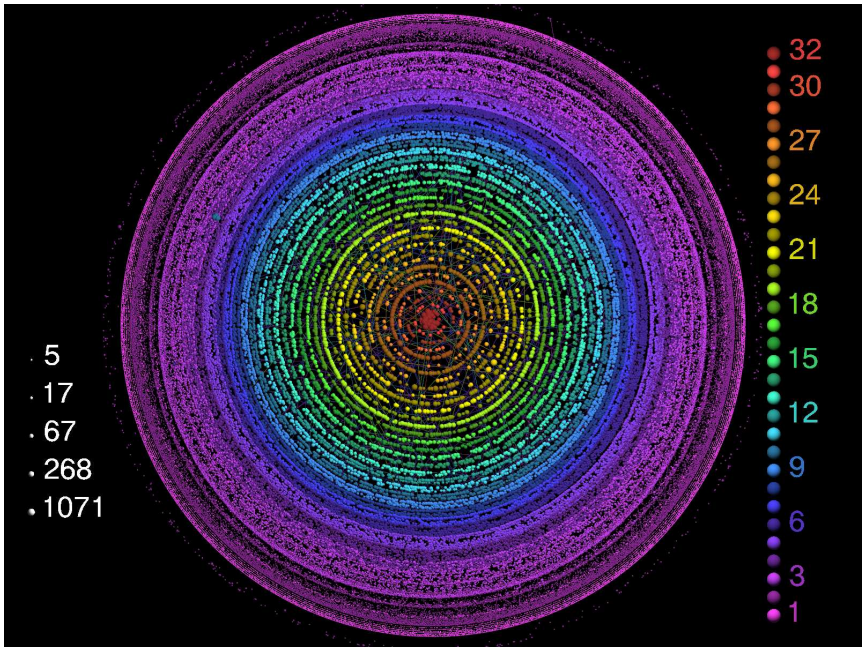


FIG. 3.16 – *Réseau CAIDA.*

La figure 3.16 est un autre réseau de routeurs fourni par CAIDA [30]. C'est un graphe de 200000 sommets (routeurs) et 610000 liens (connections). Il contient plus d'arêtes que celui de la figure 3.15

Contrairement au réseau de routeurs de la figure 3.15, le réseau CAIDA est caractérisé par une forte corrélation entre le degré et le coreness. Par conséquent, les sommets de fort degré sont localisés dans le cercle central (voir le petit cercle au milieu de l'image) et absente des cercles en bordure (voir figure 3.16).

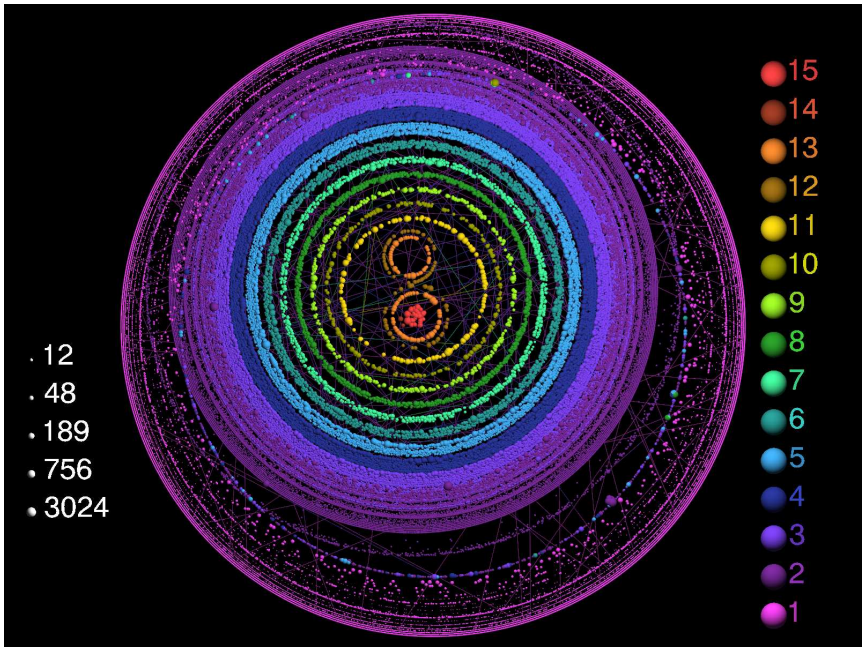
Crawl du Web et crawl aléatoire

Le crawl francophone est un crawl du Web restreint aux pages «.fr» (Web francophone) de 8 millions d'URLs triées dans l'*ordre lexicographique* de «.fr» fait en juin 2001 dans le cadre de l'action de recherche coopérative Soleil Levant [69].

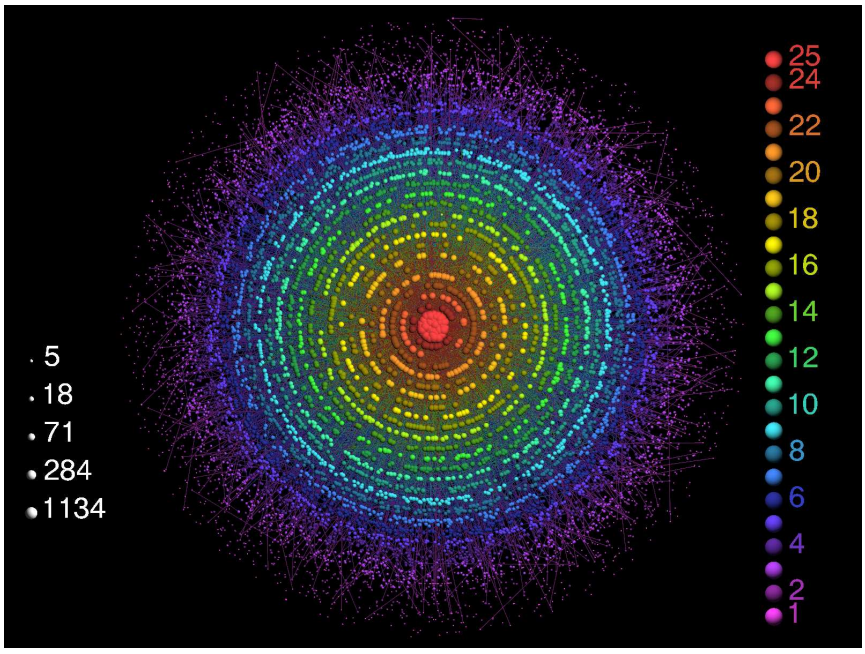
Nous lui avons opposé un crawl aléatoire de 10000 sommets [4] avec un degré moyen de 7. La distribution des degrés suit une loi de puissance de paramètre $\lambda_{in} = 2.1$ et $\lambda_{out} = 2.7$. Lors de la génération, nous avons choisi le parcours en largeur comme stratégie de crawl.

La figure 3.17 montre que les deux crawls diffèrent. Dans le crawl du Web francophone, il existe peu de corrélation entre le degré et le coreness. En effet, le Web n'a pas une structure

4. La version disponnible sur http://xavier.informatics.indiana.edu/lanet-vi/ ne permet pas d'utiliser des graphes plus grands.



Crawl francophone



Crawl aléatoire (notre modèle)

FIG. 3.17 – *Visualisation d'un crawl francophone [69] et d'un crawl aléatoire généré.*

hiérarchique globale. Si on observe bien le crawl aléatoire, on remarque qu'il n'existe pas de sommet de fort degré dans les cercles supérieurs. Ils sont tous concentrés au centre (le plus petit cercle). Donc, les sommets de forts degrés sont liés entre eux avec une plus forte probabilité que dans le crawl francophone où la probabilité qu'un sommet de fort degré soit lié a un sommet de faible degré est petite mais non négligeable.

On en déduit, que notre crawl aléatoire diffère des crawls des pages Web ou du graphe des routeurs IR. Cela s'explique du fait que les sites Web ont une structure arborescente [74] et lors de la recherche de k-core, l'algorithme élimine cette arborescence de manière récursive et entraîne parfois des sommets de fort degré appartenant à cette arborescence [74].

Graphe d'Erdös et Rényi

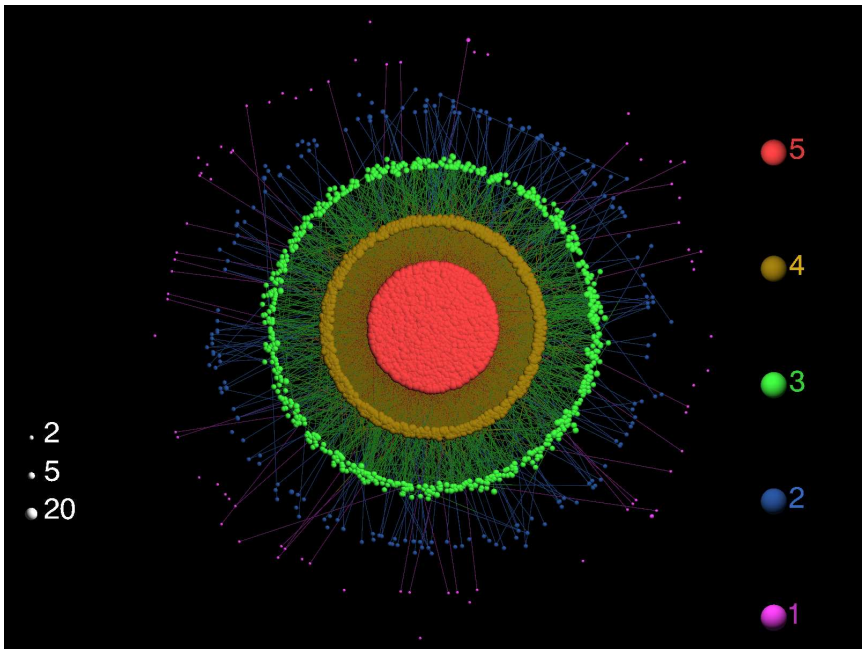


FIG. 3.18 – *Modèle d'Erdös et Rényi $G_{n,p}$ avec $n = 10000$ et $p = 0.0007$.*

La figure 3.18 montre que le graphe $G_{n,p}$ généré par le modèle d'Erdös et Rényi (voir § 2.1 page 28) a une valeur maximale de coreness proche du degré moyen. En effet, la distribution des degrés suit une loi de poisson de paramètre $n * p = \overline{d}$ et très peu de sommets ont un degré éloigné du degré moyen. Cette caractéristique des graphes aléatoires d'Erdös et Rényi est visible sur la figure 3.18.

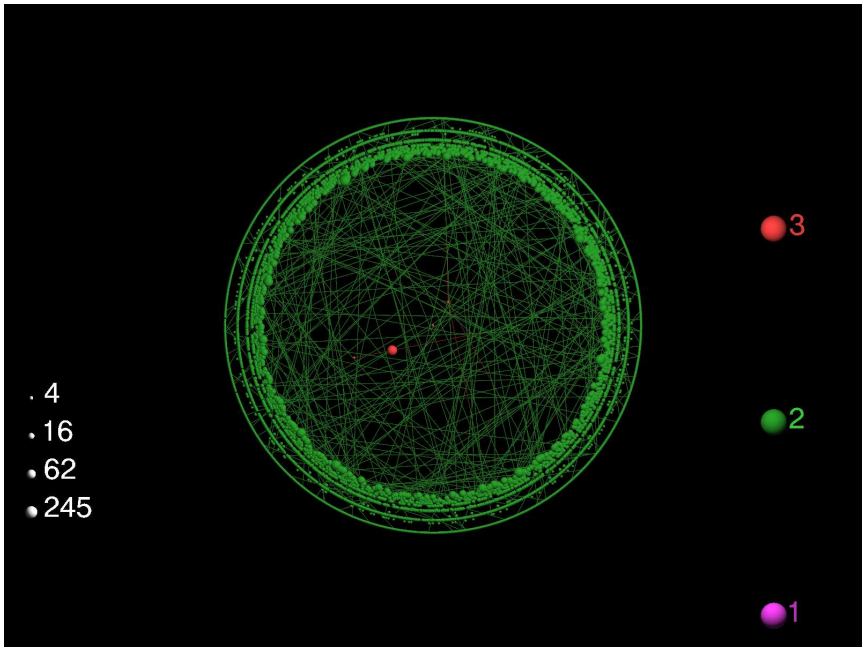


FIG. 3.19 – *Modèle de Barabasi avec* $m = 2$.

Graphe avec attachement préférentiel

Les graphes générés par le modèle de Barabasi sont décomposables en m forêts (voir § 2.2.2 page 31). On remarque sur la figure 3.19 qu'à partir des deux premiers coreness, la majorité des sommets sont éliminés. Comme les graphes de Barabasi ont une structure arborescente, on retrouve des sommets de fort degré sur les deux premiers cercles (ceux qui sont reliés à des sommets de faible degré).

Graphe petits mondes

On termine notre étude comparative avec l'anneau de Watts et Strogatz. Nous avons pris un anneau régulier de Watts et Strogatz (voir § 2.2.4 page 34) tel que tous les sommets ont le même degré qui est de 7. Ensuite, nous avons modifié les arêtes de manière aléatoire et uniforme. Dans la figure 3.20, 25% des arêtes ont changé d'extrémité finale. On observe que la majorité des sommets sont dans l'ensemble de coreness 6. Le sommet de coreness 5 est apparu suite à la modification du graphe. L'image de l'anneau de Watts et Strogatz est différente de celle du crawl du Web francophone ou du crawl aléatoire.

Nous pouvons dire que la structure de nos crawls est proche de celle du crawl du Web francophone, néanmoins, nos crawls ont la particularité d'avoir une forte corrélation entre les degrés

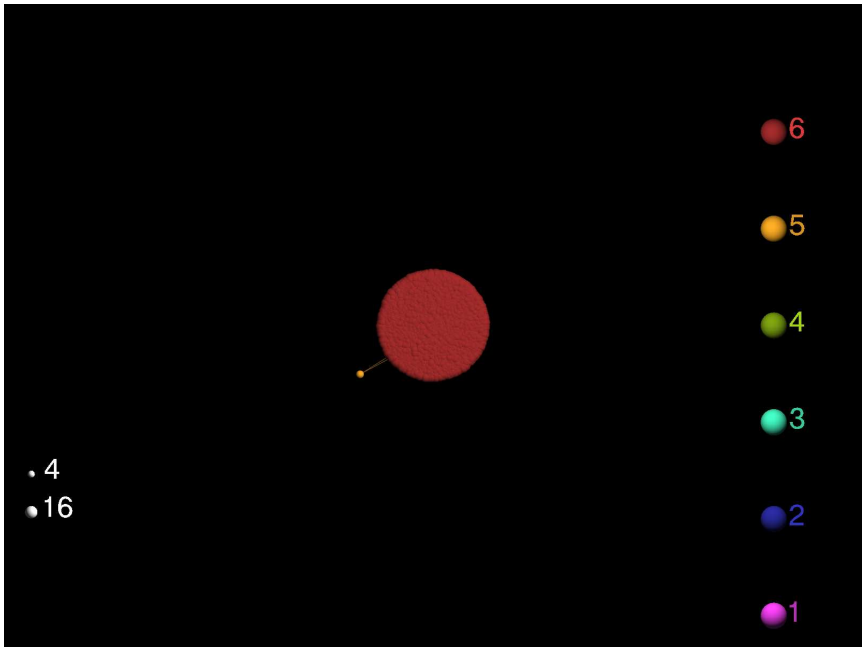


FIG. 3.20 – *Modèle de Watts et Strogatz* $G(10000,7,0.75)$.

et les coreness or que le Web francophone contient quelques sommets de forts degrés reliés à des sommets de faibles degrés. On peut dire que l'image de la décomposition en k-core du crawl aléatoire est plus proche des images des réseaux d'interactions que celles des modèles existants (Barabasi, $G_{(n,p)}$, petits mondes).

3.6 Conclusion

Dans la première partie de ce mémoire, nous avons vu que les réseaux d'interactions sont différents des $G_{n,p}$ d'Erdös et Rényi essentiellement pour deux raisons. La première est que la distribution des degrés d'un réseaux d'interaction suit une loi de puissance alors que celle d'un $G_{n,p}$ suit une loi de Poisson. La deuxième est que le coefficient de regroupement d'un réseau d'interactions est beaucoup plus fort que celui d'un graphe $G_{n,p}$.

Plusieurs modèles aléatoires ont été proposés afin de modéliser au mieux les réseaux d'interactions. Les plus connus sont le modèle d'attachement préférentiel, le modèle de copie et le modèle petits mondes. Tous ces modèles n'arrivent pas à générer des graphes aléatoires avec toutes les propriétés communes des réseaux d'interactions à l'exception des travaux récents de Guillaume et Latapy.

Nous avons proposé un modèle de génération de crawl du Web simple, les seuls paramètres nécessaires sont ceux des lois de puissances pour la génération des degrés sortants et entrants. Le

processus de génération quant à lui n'est rien d'autre qu'une simulation de parcours de graphe par une stratégie donnée. Ce processus est identique au processus de crawl sur le Web. Les résultats montrent que les crawls obtenus ont plusieurs propriétés intéressantes et semblables aux propriétés d'un graphe du Web. L'inconvénient de notre modèle est que les sommets de fort degrés sont très liés entre eux. En utilisant une stratégie de parcours qui alterne parcours en largeur et parcours en profondeur permettra de réduire la corrélation entre le degré et le coreness.

Nous affirmons donc que ce modèle capture suffisamment bien la structure dite «du graphe du Web», sans qu'il soit nécessaire de faire appel à des présupposés sociologiques sur la façon dont les auteurs lient leurs pages Web, tels que l'attachement préférentiel ou la copie.

Nous pouvons dire qu'un crawler fournit différentes vues du Web selon la stratégie de parcours adoptée. Peut-on dire que les crawls étudiés jusqu'à présent sont assez larges pour capturer les «vraies» propriétés du Web ou bien que les propriétés observées ne sont rien d'autres qu'un artefact dû au parcours du Web ?

Une étude formelle de notre modèle permettrait de répondre à la question précédente. Étant donné que dans notre modèle, les arêtes ne sont pas indépendantes les unes des autres, il n'est donc pas possible d'utiliser les résultats sur les graphes aléatoires pour prouver directement les résultats observés empiriquement [83, 86, 87].

Deuxième partie

Calcul et visualisation de communautés

Chapitre 4

Structure de liens hypertextes

LA RECHERCHE DOCUMENTAIRE est un domaine de l'informatique dont le but est de trouver des documents pertinents pour une requête donnée dans une collection de documents. Avant la naissance du Web, les algorithmes de recherche d'information étaient basés exclusivement sur l'analyse texte du contenu des documents ou sur des méta-données (mots clefs, etc.). Aujourd'hui, les utilisateurs du Web ont accès à différents moteurs de recherche dont les algorithmes de recherche utilisent non seulement le contenu texte des pages Web mais également la structure des liens hypertextes du Web.

Comment l'analyse des liens hypertextes peut-elle améliorer la recherche d'information sur le Web ? Savoir qu'une page Web u contient un lien vers une page Web v n'est pas en soi une information utile pour la recherche d'information. Cependant, la manière d'utiliser les liens hypertextes par un concepteur de pages Web peut donner une information sur le contenu des pages Web liées. En effet, un concepteur de sites Web crée des liens qu'il pense utiles à l'utilisateur. Parmi ces liens, certains ne sont utilisés que pour la navigation (par exemple, un lien dans une page Web qui pointe vers la page d'accueil du site Web). Et d'autres liens permettent d'accéder à des pages Web contenant de l'information supplémentaire sur les sujets traités par les pages pointantes. Ces derniers liens tendent à pointer vers des pages Web pertinentes pour un sujet donnée.

La combinaison du contenu texte des pages Web et de l'analyse des liens hypertextes a donné naissance à une nouvelle génération de moteurs de recherche, améliorant de manière significative la pertinence des résultats obtenus lors d'une recherche d'information. Actuellement, la plupart des moteurs de recherche d'information sur le Web prétendent utiliser une analyse de liens hypertexte. Cependant, pour raison de confidentialité, ils ne donnent que peu d'information sur le type d'analyse utilisé.

Généralement un moteur de recherche tri les pages Web présentées à l'utilisateurs dans l'ordre décroissant de leurs pertinences. Le processus effectuant cette tâche est appelé processus de tri. L'algorithme permettant de trier les pages Web est appelé algorithme de classification ou de tri.

Les algorithmes de tri dans les systèmes de recherche d'information classiques utilisent les

mots dans les pages Web. On peut citer comme exemple le *modèle de l'espace vectoriel* [101]. Dans ce modèle un mot est défini comme une dimension de l'espace de modélisation. Une page Web est alors représentée par un vecteur, en fonction des mots qu'elle contient. Ainsi, une page Web contenant les mots *informatique* et *mathématique* sera un vecteur intermédiaire entre les directions des axes informatique et mathématique. Chaque mot est doté d'un poids qui reflète son importance dans la page Web. La fonction qui calcule la pondération est une fonction croissante de la fréquence du mot dans la page, et de la rareté du mot dans la collection. Plus un mot d'une page Web a une forte pondération, plus le vecteur de la page se rapproche de l'axe de l'espace correspondant à ce mot. La similarité de deux pages Web se mesure alors à l'aide d'une métrique sur l'espace vectoriel défini par les mots.

Les techniques classiques de recherche d'information sont efficaces dans différents domaines mais ne fonctionnent pas bien sur le Web. En effet, pour des raisons diverses, des personnes sont prêtes à tout pour voir leurs sites Web très bien classé par les moteurs de recherches. Avec les techniques classiques de recherche d'information, il existe plusieurs façon pour améliorer le classement d'une page Web. Par exemple, ajouter plusieurs fois le même texte dans la page et avec une police invisible. Ce procédé permet de fausser le résultat de l'algorithme de tri. Si le modèle de l'espace vectoriel étaient utilisé pour le tri et si on ajoute 1,000 fois le mot *papillon* dans une page Web, cette dernier sera bien classée pour la requête *papillon*. N'importe quel algorithme basé que sur le contenu des pages Web est susceptible de ce genre de manipulation.

La puissance de l'analyse des liens hypertexte vient du fait que pour calculer la pertinence d'une page Web, elle va utiliser la pertinence des pages Web liées. Ces dernières ayant été créées plus ou moins de manière indépendante de la page à classer.

Ce chapitre présente deux algorithmes de tri basés sur l'analyse des liens hypertextes : L'algorithme PageRank de Brin et Page [26] et l'algorithme HITS (Hypertext Induced Topic Search) de Kleinberg [60]. Ils sont utilisés par la plupart des nouveaux moteurs de recherche (moteurs de recherche de deuxième génération).

Dans la section 1, nous présentons les différents domaines d'applications où l'analyse des liens hypertextes peut intervenir. L'algorithme de tri PageRank de Page et Brin [89] est décrit dans la section 2, il permet assigner un rang à chaque page Web indépendamment de la requête utilisateur. Nous présentons d'abord la forme simplifiée du PageRank et ensuite sa forme généralisée et pratique. Enfin, nous présentons l'algorithme HITS de Kleinberg [60], il affecte pour chaque requête utilisateur, un rang aux pages Web liées à la requêtes. Les deux algorithmes s'appuient sur une analyse spectrale du graphe du Web.

4.1 Analyse de liens hypertexte

Pour un algorithme d'analyse de liens hypertextes, l'existence d'un lien entre la page u et v peut être interprétée de deux manières :

– *Supposition 1* : si un lien hypertexte existe entre u et v, alors l'auteur de la page u recommande la page v.

– *Supposition 2* : si un lien hypertexte existe entre u et v, alors les deux pages traitent du même sujet.

Ces deux suppositions sont utilisées dans différents domaines notamment pour la collecte de pages Web (*crawling*), pour mesurer la qualité d'une collection de pages Web et enfin pour classer les pages Web par pertinence.

4.1.1 Collecte de pages Web

Un bon *crawler* (voir annexe C page 119) devrait être capable de «détecter» les pages Web de bonnes qualités et de les visiter en premier. L'analyse des liens hypertextes joue un rôle important dans l'évaluation de la pertinence d'une page Web visitée ou non. La supposition 1, implique qu'une page Web pointée par beaucoup d'autres pages Web possède une *qualité* plus importante qu'une page Web très peu pointée. Par conséquent, l'estimation [1] du degré entrant d'une page Web peut être utilisé pour classer les pages Web à visiter. Ainsi, les pages Web de fort degrés entrants sont visitées en premier. On obtient ainsi une collection de pages Web de bonnes «qualités». Le PageRank (voir § 4.2 page 68) peut également être utilisé pour mesurer la qualité d'une page Web [26, 89]. Abiteboul *et al.* [4] proposent un algorithme incrémental pour calculer le PageRank estimé d'une page Web.

4.1.2 Tri des pages Web

Lorsqu'un utilisateur soumet une requête [2] à un moteur de recherche, ce dernier renvoie une liste d'URLs. Les pages Web associées aux URLs pouvant être en relation avec la requête et souvent contiennent les mots clés de la requête utilisateur (mais pas forcément). Cette liste URLs dépend de la requête et de l'algorithme de tri utilisé par le moteur de recherche.

Il existe deux classes d'algorithmes de tri basé sur la connectivité du graphe du Web. La classe des algorithmes globaux donnant à chaque page Web un rang indépendamment de la requête utilisateur (ce rang est utilisé pour toutes les requêtes utilisateurs). Et la classe des algorithmes locaux qui dépendent de la requête utilisateur et où à chaque page Web est assigné un rang n'ayant de sens que pour une requête donnée (ce rang varie d'une requête à une autre).

Afin de décrire les différents algorithmes, on modélise le Web par un graphe $G(V,E)$ où chaque sommet $p \in V$ représente une page Web et un arc $h \in E$ représente un lien hypertexte entre deux pages Web.

1. Il n'est pas possible de connaître exactement le degré rentrant d'une page Web.
2. Une requête contient un ou plusieurs mots clés.

4.2 Tri global : algorithme PageRank

Le principe de l'algorithme de tri global est de donner un rang à une page Web p sans tenir compte des requêtes utilisateurs. En d'autres termes, le rang est indépendant de la requête et est le même pour toutes les requêtes. Ce rang est généralement combiné avec d'autre rangs, ces derniers pouvant dépendre ou pas de la requête utilisateur ou du contenu des pages Web.

Intuitivement, on peut dire qu'actuellement la page Web de Google est plus importante que la page Web de notre méta moteur[3]. Cette importance peut être exprimée en nombre de pages Web qui pointent vers les deux sites Web. En effet, beaucoup de pages Web pointent vers le site Web de Google, par contre très peu de pages Web pointent vers notre méta moteur. Donc le rang d'une page Web p peut être défini comme étant le nombre de pages Web pointant sur elle. Ce dernier est totalement indépendant de la requête utilisateur.

Le problème avec cette technique, est qu'on ne fait aucune distinction entre une page Web u pointée par un nombre t de pages Web de mauvaise qualité et une page Web v pointée par le même nombre t de pages Web mais de bonne qualité. Les pages u et v auront une mesure de qualité identique. De plus, il est facile d'augmenter la qualité d'une page Web u, pour cela, il suffit de créer beaucoup de pages Web qui toutes pointent vers la page Web u.

L'algorithme *PageRank* créé par Page et Brin [26, 89] tient compte de ce problème. Les auteurs calculent le PageRank de chaque page Web en donnant un poids à chaque lien d'une page Web. Ce dernier est proportionnel à la qualité de la page contenant ce lien. Pour déterminer la qualité d'une page donnée, le PageRank est calculé de manière récursive avec un valeur initiale quelconque. L'algorithme PageRank est utilisé par le moteur de recherche Google[4]. Il permet de distinguer entre les pages Web de bonnes qualités des autres pages Web.

4.2.1 PageRank simplifié

Nous présentons d'abord une définition simplifiée du PageRank. Soit N le nombre de pages Web numérotées de 1 à N, $d^+(p)$ le nombre de liens hypertextes contenus dans la page p et $N^-(p)$ l'ensemble des pages Web qui pointent vers la page p. Le PageRank de la page p noté $r(p)$ est le point fixe du système d'équations suivantes :

$$r(p) = \sum_{q \in N^-(p)} \frac{r(q)}{d^+(q)} \tag{4.1}$$

Et :

$$\sum_{p \in V} r(p) = 1$$

Le processus itératif peut être vu comme un calcul de flot qui converge vers l'équilibre (loi des noeuds) : chaque page redistribue le flot qui la traverse.

3. `http://www.lirmm.fr/~bennouas/moteur/index.php`
4. `http://www.google.com`

Notons que grâce au théorème de Perron-Frobenius, ce système admet une solution unique si le graphe est fortement connexe et apériodique [25, 24, 75].

Avec le PageRank simplifié, une page Web q distribue la même portion de son PageRank à toutes les pages Web qu'elle pointe. C'est pour cela que le PageRank de q est divisé par $d^+(q)$ (voir figure 4.1).

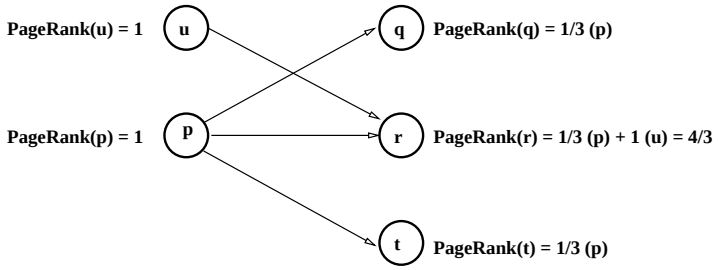


FIG. 4.1 – *Propagation du PageRank*

L'équation 4.1 peut être écrite en termes de produit de matrice comme ceci : soit $r = M^T r$ où r est un vecteur $N \times 1$ $[r(1),r(2),...,r(N)]$ et M est la matrice d'adjacence modifiée du graphe $G(V,E)$ tel que :

$$M(p,q) = \begin{cases} \frac{1}{d^+(p)} & \text{si la page } p \text{ pointe vers la page q} \\ 0 & sinon \end{cases}$$

Le vecteur PageRank r n'est rien d'autre que le vecteur propre de la matrice M^T associé à la valeur propre 1. Comme nous supposons que le graphe $G(V,E)$ est fortement connexe, il est facile de montrer que la valeur 1 est une valeur propre de M^T et que le vecteur propre r associé à cette valeur est unique sous certaines conditions d'après le théorème de Perron Frobenius [25, 24, 75].

Modèle du surfeur aléatoire

La forme simplifiée du PageRank modélise le comportement d'un surfeur aléatoire sur le Web. En effet, il est possible d'imaginer un utilisateur en train de surfer sur le Web en cliquant sur des liens de manière totalement aléatoire. Le parcours aléatoire du Web est équivalent à une marche aléatoire du graphe du Web. Le problème de marche aléatoire d'un graphe a été longuement étudié (processus markoviens). Le vecteur r n'est rien d'autre que la distribution stationnaire d'une marche aléatoire sur le graphe du Web [80, 24, 25].

Calcul du PageRank simplifié

Pour calculer le PageRank, il est nécessaire de calculer le vecteur propre de la matrice M^T associé à la valeur propre 1. La méthode la plus simple est d'utiliser la méthode de la puissance itérée. C'est une méthode itérative de calcul de la valeur propre dominante d'une matrice (celle de plus grand module) et du vecteur propre correspondant (voir l'algorithme 1).

Algorithme 1: Calcul du PageRank simplifié

Données : M^T transposée de la matrice d'adjacence modifiée de $G(V,E)$
Résultat : r vecteur PageRank
début
  $r \leftarrow (\frac{1}{N}, \ldots, \frac{1}{N})$
  $s \leftarrow (0, \ldots, 0)$
  tant que $\|r - s\| > \epsilon$ **faire**
    $s \leftarrow r$
    $r \leftarrow M^T \times s$
  fin
  retourner $r \quad telque \quad \|r\|_1 = 1$
fin

La figure 4.2 montre un exemple de calcul du PageRank sur un graphe simple de 5 sommets. On remarque que le vecteur est normalisé à un, c'est-à-dire que la somme des rangs est égal à 1. On remarque que le retrait de l'arc en pointillés ferait perdre au graphe la propriété de forte connexité.

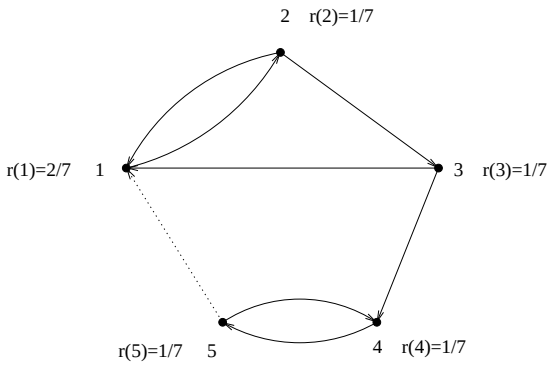


FIG. 4.2 – *PageRank simplifié*

4.2.2 PageRank pratique

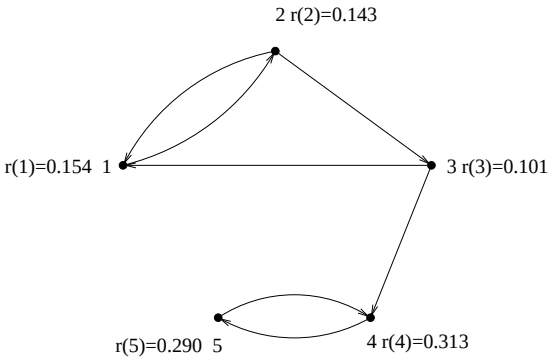


FIG. 4.3 – *PageRank pratique*

La première version du PageRank n'est valide que si le graphe du Web est fortement connexe. Mais le Web contient des sommets sans successeurs (des puits) et des composantes puits. Ce qui fait que le Web n'est pas un graphe fortement connexe. L'existence des sommets puits et les composantes puits ne permet pas l'utilisation de la forme simplifiée du PageRank.

Une page puits est pointée par une ou plusieurs pages Web mais ne pointe sur aucun page Web. par conséquent, elle ne fait qu'accumuler du PageRank mais elle ne redistribue pas sa valeur de PageRank. Dans l'exemple 4.2, si on supprime le sommet 5 et l'ensemble des liens entrant et sortant de ce dernier, alors le sommet 4 est puits.

De même, une composante puits est un ensemble P de sommets liés par des arcs. P est pointé par des sommets n'appartenant pas à P (sommets extérieur). Mais aucun sommet de P ne point vers un ou plusieurs sommet de l'extérieurs (n'appartenant pas à P). Toujours avec l'exemple 4.2, si l'arc $5 \rightarrow 1$ est supprimé, alors l'ensemble composé des deux sommets 4 et 5 est une composante puits. Les sommets 4 et 5 ne pointent vers aucun autre sommet (voir figure 4.3).

Pour éliminer ce problème, Page et Brin [89] proposent de supprimer tous les puits et introduisent un facteur $d \in [0,1]$ dans la définition du PageRank. Formellement, le PageRank $r(p)$ d'une page Web p est défini par le point fixe du système d'équations suivant :

$$r(p) = \frac{d}{n} + (1-d) * \sum_{(q,p)\in V} \frac{r(q)}{d^+(q)} \tag{4.2}$$

où

- d est une constante réelle, dans la pratique comprise entre 0.1 et 0.2;
- n est le nombre de sommets dans le graphe (le nombre de pages Web);
- $d^+(q)$ est le nombre d'arcs sortant du sommet q, c'est-à-dire, le nombre de liens hypertextes dans la page Web associée au sommet q;

L'équation 4.2 peut également être vue comme une modélisation du comportement d'un surfeur aléatoire. Ce dernier visite des pages Web en cliquant sur les liens se trouvant sur ces dernières. Mais à tout moment le surfeur peut interrompre sa progression et reprendre le processus à partir d'une autre page Web choisie de manière arbitraire. Le premier terme de l'équation 4.2, représente la probabilité que la page Web soit sélectionnée de manière aléatoire. Le deuxième terme, est la probabilité qu'une page Web u soit sélectionnée (ou visitée) à partir de son voisinage. Pour $d = 0$, on retrouve la formule simplifiée du PageRank (voir L'algorithme 2).

Algorithme 2: Calcul du PageRank pratique

Données : A^T transposée de la matrice d'adjacence modifiée de $G(V,E)$
Résultat : r vecteur PageRank
début
  $s \leftarrow (0,\ldots,0)$
  $E \leftarrow (1/N)_{N\times N}$
  $r \leftarrow (1/N,\ldots,1/N)$
  tant que $\|r - s\| > \epsilon$ **faire**
    $s \leftarrow r$
    $r \leftarrow (1-d) * A^T \times s + d * E$
  fin
  retourner $r \quad telque \quad \|r\|_1 = 1$
fin

La figure 4.3 montre un exemple de calcul de PageRank avec $d = 0.2$. Les sommets 4 et 5 ont un fort PageRank indiquant que le surfeur aléatoire a une forte probabilité de se trouver sur un des deux sommets.

Comme expliqué par [75], le facteur d a un rôle important dans la vitesse de convergence du calcul et dans la pertinence du classement obtenu.

Il existe dans la littérature une variété d'algorithmes basés sur le principe du PageRank. Par exemple, un surfeur souvent utilise la touche back du navigateur pour revenir sur la page précédente, Mathieu et Bouklit [73] ont introduit ce comportement dans le calcul de l'importance d'une page Web et l'ont nommé le *BackRank*. La convergence de l'algorithme BackRank est plus rapide que l'algorithme PageRank de Page et Brin [89]. Cette performance est due à l'utilisation de la méthode de Gauss-Seidel qui améliore considérablement la convergence du calcul du PageRank [73, 14].

Algorithme 3: Algorithme de construction du graphe de voisinage

Données : σ : la requête utilisateur
ζ : Un moteur de recherche basé sur l'analyse texte
t,d : des constantes
Résultat : S_σ : graphe de voisinage
Soit R_σ : les t meilleurs pages Web obtenues par ζ pour la requête σ
début
  pour chaque *page* $p \in R_\sigma$ **faire**
    Soit $\Gamma^+(p)$ l'ensemble de toutes les pages Web pointées par p
    Soit $\Gamma^-(p)$ l'ensemble de toutes les pages Web qui pointent vers p
    Ajouter toutes les pages de $\Gamma^+(p)$ dans S_σ
    si $\mid \Gamma^-(p) \mid \leq d$ **alors**
      Ajouter toutes les pages de $\Gamma^-(p)$ dans S_σ
    sinon
      Ajouter d pages de $\Gamma^-(p)$ prises au hasard dans S_σ
    fin
  fin
  retourner S_σ
fin

4.3 Tri local : algorithme HITS

Le principe du tri local est de construire un sous-graphe du Web appelé graphe de *voisinage* et d'ordonner les pages Web de ce sous-graphe en affectant un rang à chacune d'elle. Ce sous-graphe doit contenir des pages Web en relation avec la requête utilisateur. Donc, on distingue deux phase dans l'algorithme de tri, la première consiste à construire le sous-graphe du Web et la deuxième consiste à ordonner les pages Web de ce sous-graphe par pertinence.

Construction du graphe de voisinage

Carriere et Kazman [31] proposent une méthode pour construire le graphe de voisinage. Elle consiste à utiliser un moteur de recherche basé sur l'analyse texte pour récupérer un ensemble de pages Web en relation avec la requête utilisateur, ensuite étendre cet ensemble en ajoutant le voisinage direct des pages Web de cet ensemble (voir figure 3). L'algorithme 3 permet de construire le graphe de voisinage. En pratique, seulement les $t = 200$ premières pages Web obtenues par le moteur de recherche (basé sur le texte) sont prises et si lors de l'extension du voisinage, une page Web a beaucoup de liens entrants alors seulement $d = 50$ de ses voisins choisi aléatoirement sont ajoutés au graphe. La taille de l'ensemble S_σ varie entre 1000 et 5000

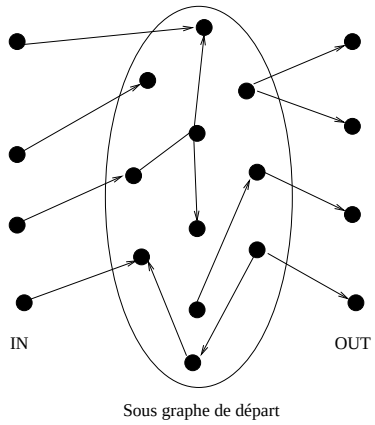


FIG. 4.4 – *Construction du graphe de voisinage S_σ.*

sommets (voir l'algorithme 3).

Suppression des liens internes

Dans le graphe de voisinage, on distingue deux types de liens : internes et externes. Les liens internes sont des liens entre pages Web de même sites Web, les liens externes sont des liens entre pages Web de sites différents. Un site est identifié par la première partie d'une URL. Par exemple, `http://www.lirmm.fr/xml/fr/lirmm.html` est une page Web du site `www.lirmm.fr`. Noter que cette définition de site comme étant les pages d'un même serveur est assez arbitraire. Donc si deux pages Web ont le même préfixe alors elle appartiennent au même site Web et si un lien existe entre deux pages Web de même site, on dit que c'est un lien interne.

Tous les liens internes sont supprimés du graphe du voisinage. La suppression de leurs liens permet d'éviter que certaines pages Web populaires soient avantagées par rapport à d'autres. Par exemple, beaucoup de pages Web de même site pointent toutes vers la page d'entrée du site (page home). Ces liens de navigation n'apportent aucune information et risque même de provoquer des erreurs lors de l'ordonnancement des pages Web.

Après la construction du graphe de voisinage (voir figure 4.4), la deuxième étape consiste à trier les pages Web par pertinence. Pour cela, plusieurs algorithmes de tri peuvent être utilisés : le degré entrant, le PageRank [5], etc.

Calcule des autorités et des annuaires

Kleinberg [60] s'intéresse à des pages Web appelées les *autorités* et les *annuaires*. Il existe sur le Web des pages très pertinentes mais ne contenant que quelques mots clés (voir aucun mot clé) de la requête donnée par l'utilisateur. Prenons l'exemple de la requête «moteur de recherche», la

5. [13] a montré que le PageRank produit le même résultat que l'utilisation des degrés entrants.

page Web «`www.google.fr`» (page home du moteur de recherche Google) et «`fr.altavista.com`» (page home du moteur de recherche Altavista) sont des pages Web pertinentes pour la requête «moteur de recherche». Il est facile de vérifier que ces pages Web ne contiennent pas les mots clés de la requête. Les pages Web «`www.google.fr`» et «`fr.altavista.com`» sont des *autorités* pour la requête «moteur de recherche». De plus, il est facile d'admettre que les deux pages Web sont parmi les résultats les plus pertinents.

Il existe également sur le Web des pages Web contenant un ou plusieurs mots clés de la requête avec beaucoup de liens sortants vers des pages home de moteurs de recherche et notamment vers les pages Web «`www.google.fr`» et «`fr.altavista.com`». Ces pages Web pointent vers des pages Web pertinentes. Ces pages Web sont appelées *annuaires*.

Formellement, Kleinberg [60] considère une page Web comme une autorité si elle est pointée par beaucoup de pages Web et comme un annuaire si elle pointe vers beaucoup de pages Web. De manière récursive, il considère une page Web comme une *bonne* autorité si elle est pointée par de *bons* annuaires et comme un *bon* annuaire si elle pointe vers des *bonnes* autorités. Cette relation entre autorités et annuaires est appelée *renforcement mutuel*.

Vers un algorithme itératif

Algorithme 4: Algorithme HITS

Données : $G(S_\sigma)$: Le graphe de voisinage
k : une constante
Résultat : (A_k,H_k) : vecteur d'autorités et annuaires
Soit z : un vecteur $(1,1,...,1) \in \mathbb{R}^n$
$A_0 \leftarrow z$
$H_0 \leftarrow z$
début
  pour *i = 1 à k* **faire**
    pour chaque *sommet* p *de* $G(S_\sigma)$ **faire**
      $A_i^{(p)} = \sum_{q:(q,p)\in G(S_\sigma)} H_{i-1}^{(q)}$
      $H_i^{(p)} \longleftarrow \sum_{q:(p,q)\in G(S_\sigma)} A_i^{(q)}$
    fin
    $A_i = \frac{A_i}{\|A_i\|}$
    $H_i = \frac{H_i}{\|H_i\|}$
  fin
  retourner (A_k,H_k)
fin

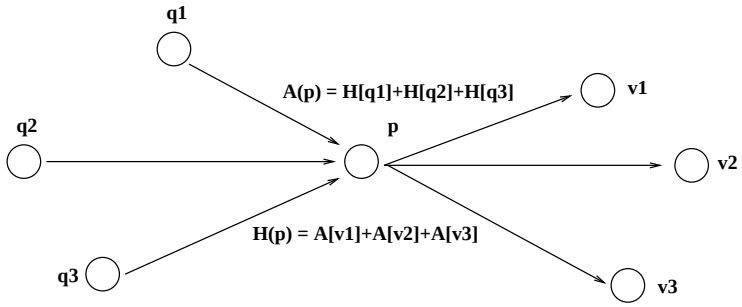


FIG. 4.5 – *Calcul des autorités et des annuaires*

À partir d'un graphe de voisinage, l'algorithme *HITS* de Kleinberg [60] permet de déterminer les autorités et les annuaires en associant à chaque pages Web deux valeurs : une valeur d'autorité notée *A* et une valeur de annuaire notée *H* (voir algorithme 4).

Pour une page Web p (voir figure 4.5), la valeur de $A(p)$ et $H(p)$ sont le point fixe du système d'équations suivant :

$$A(p) \longleftarrow \sum_{q:(q,p)\in G(S_\sigma)} H(q) \tag{4.3}$$

$$H(p) \longleftarrow \sum_{q:(p,q)\in G(S_\sigma)} A(q) \tag{4.4}$$

Ces deux équations expriment la notion de renforcement mutuel.

Convergence de l'algorithme HITS

Les équations 4.3 et 4.4 peuvent être exprimées en termes de produit de matrices. Soit M la matrice d'adjacence du graphe S_σ. Alors les vecteur A_k et H_k peuvent s'écrire de la manière suivante :

$$A_k = M^T \times H_{k-1} \tag{4.5}$$

$$H_k = M \times A_k \tag{4.6}$$

Si les vecteur A_0 et H_0 sont des vecteurs quelconque, alors les équation 4.5 et 4.6 peuvent être écrites comme suit :

$$A_k = (M^T M)^k \times A_0 \tag{4.7}$$

$$H_k = (M M^T)^k \times H_0 \tag{4.8}$$

Les matrices $M^T M$ et $M M^T$ sont des matrices symétriques carrées. Donc, le produit $(M^T M)^k \times A_0$ (respectivement $(M M^T)^k \times H_0$) converge vers le vecteur propre associé à la plus grande

(en module) valeur propre de la matrice $M^T M$ (respectivement MM^T). L'annexe B page 115 décrit les notion algébriques utilisées dans l'algorithme HITS.

L'algorithme HITS est beaucoup utilisé pour détecter certaines types de pages Web appelées *fan* et *stars* [60]. Il est également utilisé pour trouver des pages Web liées à une page Web donnée [38].

En résumé, il existe plusieurs techniques basées sur l'analyse des liens hypertexte permettant essentiellement d'orienter un crawler lors du parcours du Web et de mesurer la pertinence des pages Web. Ces techniques peuvent être divisées en deux classes. La classe des algorithmes généraux ou globaux où la qualité d'une page Web ne dépend pas de la requête utilisateur et la classe des algorithmes spécifiques ou locaux où la qualité d'une page Web est liée à la requête utilisateur.

Ces techniques basées sur l'analyse des liens hypertexte sont plus fiables que celles basées sur l'analyse texte, en effet, il n'est pas facile de fausser le résultat de ces algorithmes, de fait que la qualité d'une page Web dépend de la qualité des pages Web liées à cette dernière et est hors de contrôle du concepteur de la page Web.

Les deux algorithmes s'appuient sur une analyse spectrale du graphe du Web et sur un processus itératif pour le calcul des rangs. Les deux algorithmes convergent rapidement (au bout de quelques itérations). La majorité des moteurs de recherches d'aujourd'hui utilisent un algorithme basé sur l'analyse des liens hypertextes.

Chapitre 5

Calcul de cyber-communautés par émergence

DANS la littérature, il n'existe pas de définition formelle consensuelle d'une structure de communautés dans un réseau d'interaction, mais on pense généralement à l'existence d'un ensemble de noeuds densément liés. Pour avoir une bonne connaissance des réseaux d'interactions, il est nécessaire de bien connaître leurs structures et de savoir calculer les communautés existantes.

Lorsqu'on s'intéresse au problème d'extraction de la structure de communautés, il est nécessaire d'avoir une définition formelle de cette dernière ; or il en existe plusieurs. Les algorithmes d'extraction de communautés doivent être capable de manipuler des réseaux de très grande taille et de trouver rapidement les communautés.

Pour répondre à ces questions, nous avons conçu un outil contribuant à la solution de trois problèmes : identifier les communautés dans le Web, fournir un outil de visualisation de la structure du Web et enfin offrir une mesure de qualité des pages Web. Contrairement aux méthodes d'extractions de communautés déjà existantes, notre algorithme ne forme pas les communautés mais les extrait seulement. En effet, nos communautés émergent par elles-mêmes et aucune supposition n'est faite *a priori*.

Dans ce chapitre, nous présentons d'abord les différentes définitions de la notion de communautés dans le graphe du Web. La section 2 décrit très brièvement les différentes méthodes d'extraction de communautés dans les réseaux d'interactions ; les algorithmes utilisés partitionnent le graphe en communautés (ils sont généralement appelés algorithmes de partitionnement ou de clustering). Dans la section 3, nous présentons notre mesure de qualité d'une partition ainsi que la mesure de Newman [87]. Ces mesures permettent d'évaluer la qualité du partitionnement obtenu par les méthodes de clustering. La section 4 présente le modèle gravitationnel où les pages Web deviennent des particules qui se déplacent dans un espace tri-dimensionnel. Une amélioration de ce modèle est présentée dans la section 5 : le modèle intentionnel où chaque particule est dotée d'un objectif qui consiste à rejoindre sa communauté. Chaque modèle est décrit en détail. Nous

commentons les différentes expérimentations réalisées dans les deux modèles. Nos algorithmes permettent de faire émerger les communautés par un processus qui consiste a faire rapprocher une particule de sa communauté sans aucune définition préalable de la notion de communauté.

Ce travail a été réalisé avec Mohamed Bouklit (doctorant à université de Montpellier II), Fabien de Montgolfier (maître de conférence à l'université de Paris VII) et Jean Privat (Doctorant à l'université de Montpellier II). Le modèle gravitationnel a fait l'objet de deux publications [18, 19].

5.1 Quelques définitions des communautés

Une communauté du Web est généralement définie comme étant un ensemble de pages Web partageant le même thème (sujet). Cette définition est utilisée par les concepteurs de moteurs de recherche pour la classification des pages Web. Plusieurs auteurs ont proposé des définitions formelles basées sur l'analyse sémantique ou sur l'analyse de la topologie. Notre travail est basée sur une analyse topologique du Web. Dans cette section, nous présentons quelques définitions de la notion de communauté (cluster), principalement celles liées au Web.

Les fans et les stars

Gibson *et al.* [49] utilisent l'algorithme HITS de Kleinberg [60] (voir § 4.3 page 73) pour extraire les communautés du Web. L'ensemble de pages Web trouvé a une structure de bicliques : d'un coté, il y a l'ensemble des pages appelées fan, ces pages contiennent beaucoup de liens vers des pages pertinentes, par exemple, un marque-pages (bookmarks). De l'autre, des pages Web très pointées (référencées) appelée stars. Un exemple d'autorité est un site officiel (voir figure 1.7 page 25).

Les noyaux

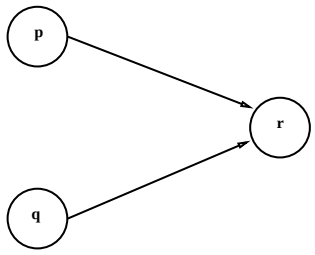


FIG. 5.1 – *La co-citation. La page p et q pointent sur la page r. D'après Kumar* et al. *[64], la co-citation est à la base de la formation des communautés.*

Kumar *et al.* [64] proposent un algorithme permettant de détecter des petites communautés. L'algorithme extrait des graphes bipartis denses d'une taille donnée appelés noyaux. Selon les auteurs, la co-citation (deux fans pointent sur la même autorité) est à la base de la structure de communauté.

Pages liées

Dean et Henzinger [38] ont utilisé également l'algorithme HITS dans un graphe de voisinage d'une page donnée. L'algorithme retrouve des pages liées à une page p, qui peuvent être vues comme un ensemble de pages Web membres de la communauté de p.

Flot maximal

Flake *et al.* [47] définissent une communauté comme un ensemble de pages Web, tel que chaque page de la communauté contient plus de liens vers des pages de la communauté que vers des pages n'appartenant pas à la communauté. Ils proposent une méthode de résolution du problème de flot maximal pour la détection de telles communautés.

Petits mondes

Watts et Strogatz [104] définissent les réseaux petits mondes comme des graphes avec une faible distance moyenne et un fort coefficient de regroupement (voir § 1.1 page 13). Ils considèrent qu'un graphe petit monde modélise bien la structure de communauté.

Adamic [5] utilise la structure de la composante fortement connexe d'un crawl du Web dans le but de rechercher des petits mondes. Efe *et al.* [40] présentent un état de l'art des méthodes d'extraction de communautés du Web en utilisant différent sous-graphes comme motifs pour la recherche.

On remarque qu'il existe plusieurs définitions de communautés centrées autour du clustering (une communauté est un ensemble de pages Web avec un grand nombre de liens internes et très peu de liens externes) ou autour de la co-citation (une communauté a une structure d'un graphe biparti).

Toutes ces approches définissent *a priori* un motif de structure de communauté, l'algorithme consiste ensuite à parcourir le graphe à la recherche de ces motifs pour exhiber les communautés préalablement définies. Dans la prochaine section, nous allons voir quelques algorithmes permettant d'extraire les communautés d'un graphe.

5.2 Les méthodes d'extraction de communautés

L'une des propriétés communes aux réseaux d'interactions est la présence d'une structure de communautés : les noeuds du réseau peuvent être regroupés pour former des parties telles

que chaque partie est caractérisée par une forte concentration de liens (dense en liens) et une faible concentration de liens entre chaque partie (voir figure 5.2). Par exemple dans les réseaux relationnels, on observe des «communautés» au sens habituel du mot.

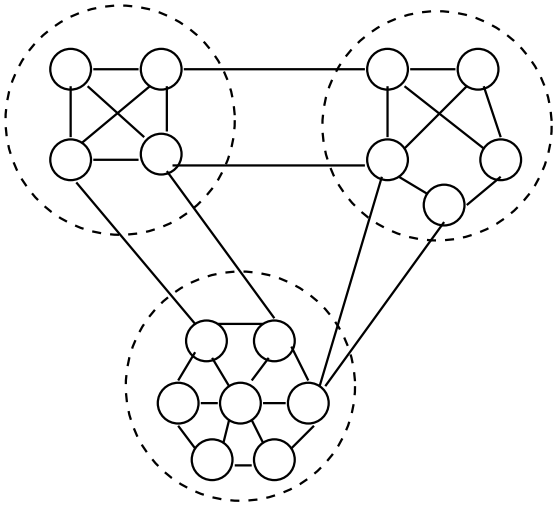


FIG. 5.2 – *Un réseau avec trois communautés. Chaque communauté est dense en liens. Très peu de liens reliant les communautés.*

Le problème de recherche de structures de communautés dans les réseaux d'interactions est très proche du problème de partitionnement de graphe (en théorie des graphes) et du clustering hiérarchique en sociologie [3, 2].

5.2.1 Le p-partitionnement

Le problème du p-partitionnement de graphe est de grouper les sommets d'un graphe en p parties (de taille q) tout en minimisant le nombre d'arêtes entre les différentes parties. Les paramètres p et q sont fixés au préalable. Ce problème est rencontré dans différents domaines notamment dans le parallélisme. Un problème de parallélisme peut être représenté par un graphe où les sommets représentent des tâches (processus) et un lien représente une communication entre deux processus. Le problème consiste à partitionner le graphe en plusieurs parties de telle sorte que le nombre de liens entre les parties soit minimisé (réduire la communication entre processus). Trouver une solution exacte au partitionnement de graphe est NP-Complet mais il existe des heuristiques qui permettent d'obtenir un résultat raisonnable. Le meilleur algorithme connu est celui de Kernighan-Lin [58] qui s'exécute en $O(n^3)$. Une variante est de fixer le nombre de parties mais laisser la taille libre.

Le problème de p-partitionnement de graphe n'est pas adapté pour la recherche de communautés dans les réseaux d'interactions. En effet, on ne connaît pas à l'avance le nombre de

communautés existant dans un réseau d'interaction. De plus, les communautés ne sont pas nécessairement de même taille. Enfin, le nombre de liens entre communautés ne doit pas être nécessairement minimisé.

Nous allons présenter des algorithmes beaucoup plus récents et mieux adaptés aux problèmes de détection et d'extraction de communautés dans les réseaux d'interactions. Ils peuvent être regroupés en deux classes : les méthodes agglomératives et les méthodes séparatives.

5.2.2 Les méthodes agglomératives

Elles consistent de manière itérative à grouper les sommets présentant de *fortes similarités* pour former les communautés. Un algorithme basé sur la méthode agglomérative (voir algorithme 5) considère qu'à l'état initial le graphe contient n communautés de taille 1. La première étape de l'algorithme consiste à calculer les distances (similarité) entre chaque communauté. La deuxième étape est consacrée à la fusion des deux communautés les plus proches, formant ainsi une nouvelle communauté. Les deux étapes précédentes sont itérées, de telle sorte, qu'à chaque itération, le nombre de communautés diminue de un. Le processus s'arrête lorsqu'il n'y a plus qu'une seule communauté. On obtient ainsi une structure hiérarchique de communautés qui peut être représentée sous forme arborescente appelée dendrogramme (voir figure 5.3).

Algorithme 5: Algorithme de la méthode agglomérative

Données : Un graphe $G(V,E)$
Une métrique d
Résultat : Arborescence de dendrogramme
début
  $C = \{C_1, \ldots, C_n\}$ tel que $|C_i| = v_i$
  tant que $|C| \neq 1$ **faire**
    pour chaque *communauté* $C_i \subset C$ **faire**
      pour chaque *communauté* $C_j \subset C$ **faire**
        Calculer la distance $d(C_i,C_j)$
      fin
    fin
    Fusionner les deux communautés les plus proches : $C_k = C_i \cup C_j$
    $C = C \backslash \{C_i,C_j\} \cup C_k$, les deux communautés ne font qu'une
  fin
  retourner *Arborescence de dendrogramme*
fin

Les méthodes agglomératives nécessitent le calcul des valeurs de similarités entre paires de sommets ou communautés. Comme les réseaux d'interactions sont généralement modélisés par

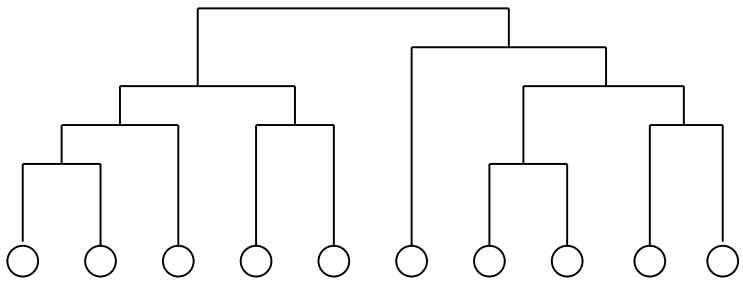


FIG. 5.3 – *Structure de dendrogramme. Les feuilles du dendrogramme sont les sommets du graphe et les noeuds représentent les communautés créées. Ces dernières sont reliées en fonction des fusions de communautés. La racine de la structure correspond au graphe entier.*

des graphes, il existe plusieurs méthodes pour calculer la similarité entre paires de sommets basées sur la structure de graphe.

Une mesure de similarité simple est de considérer la distance minimale (dans le graphe) entre deux sommets, c'est-à-dire, le plus court chemin entre les deux sommets. Ainsi, les sommets les plus proches dans le graphe sont regroupés dans une communauté. De même pour les communautés, les deux communautés les plus proches sont regroupées en une seule. Comme une communauté peut contenir plusieurs sommets, il est possible de prendre la distance minimale entre toutes les paires de sommets (la paire est constituée d'un sommet de chaque communauté), ou encore la distance maximale entre sommets ou enfin, la distance intermédiaire [58, 10].

Latapy et Pons [65] ont proposé une nouvelle mesure de similarité entre sommets ou communautés basée sur une marche aléatoire. Elle s'appuie sur la constatation intuitive ; proposé par Bruno Gaume [48] ; qu'une marche aléatoire de *courte longueur* du graphe a tendance à rester piégée dans les communautés. En utilisant cette métrique, Latapy et Pons ont également proposé un algorithme agglomératif pour la détection de communautés, cet algorithme a une complexité en temps de $O(mn^2)$ au pire des cas et une complexité de $O(n^2 \log n)$ avec des réseaux réels.

Il est possible de définir des mesures de similarités adaptées à la nature du réseau étudié. Par exemple, si le réseau étudié est un réseau d'acteurs [104, 12] (voir § 1.2.2 page 17), alors une mesure de similarité appropriée est de prendre le nombre de films dans lesquels deux acteurs ont joué ensemble [72]. Les acteurs qui ont le plus joué ensemble sont regroupés pour former une communauté.

L'inconvénient des méthodes agglomératives est qu'elles ont tendance à découvrir essentiellement le noyau d'une communauté et d'ignorer la périphérie. En effet, le noyau d'une communauté est caractérisé par des sommets possédant une forte valeur de similarité contrairement aux sommets de la périphérie. Les méthodes agglomératives généralement échouent à placer les sommets avec de faibles valeurs de similarité.

5.2.3 Les méthodes séparatives

Elles consistent à trouver dans le réseau le lien le moins «*significatif*» afin de le supprimer. Cette opération est réitérée divisant ainsi le réseau en plusieurs composantes connexes (communautés). Le processus pouvant être arrêté à n'importe quelle itération.

Newman et Givran [87], ont proposé un algorithme qui consiste, d'abord, à calculer le plus court chemin entre chaque paire de sommets ou encore de faire une marche aléatoire dans le graphe, ensuite, de supprimer le lien le moins emprunté lors des parcours. Le processus est réitéré plusieurs fois. Au final, on obtient un réseau décomposé en plusieurs communautés.

L'inconvénient majeur des méthodes séparatives est qu'à chaque étape, il est nécessaire de recalculer la valeur de similarité des liens restants dans le réseau nécessitant un temps de calcul considérable. Pour plus de détails sur les méthodes de clustering, on pourra par exemple consulter le mémoire de Pascal Pons [93].

5.3 Qualité d'un partitionnement

C'est la quantité qu'on cherche à maximiser. Elle mesure si le graphe est partitionné en communautés denses en liens, avec très peu de liens entre communautés. Dans certains algorithmes de détection et d'extraction de communautés (algorithmes de clustering), la qualité du partitionnement est utilisée pour arrêter le processus d'itération. En effet, à chaque étape, la qualité est calculée et augmente au cours du processus [86, 65] et dès que la qualité commence à se détériorer, le processus itératif est stoppé.

5.3.1 Modularité de Newman

Newman [86] a introduit la notion de modularité permettant de mesurer la qualité d'un partitionnement d'un réseau en communautés. Soit e_{ij} la fraction d'arêtes dans le réseau qui relient les sommets de la communauté i à ceux de la communauté j, et soit $a_i = \sum_j e_{ij}$. La modularité Q' est définie par :

$$Q' = \sum_i (e_{ii} - a_i^2)$$

Elle représente la fraction d'arêtes à l'intérieur d'une communauté moins la proportion d'arêtes entre communautés. La complexité de l'algorithme calculant la modularité d'une partition est en $O(m)$.

5.3.2 Une nouvelle mesure de qualité

Étant donnée une partition C du graphe $G(V,E)$ en communautés, on définit la fraction e_{ij} d'arêtes du graphe G joignant la communauté i à la communauté j. Soit $\overline{e}_{ij}$ la fraction d'arêtes

du graphe $\overline{G}$ joignant la communauté i à la communauté j. On rappelle que $\overline{G}$ est le graphe complémentaire de G (voir § 1.1 page 13). La modularité Q est définit par :

$$Q = \sum_i (e_{ii} - \overline{e}_{ii})$$

Cette mesure représente la proportion d'arêtes qui sont à l'intérieur d'une communauté moins la proportion d'arêtes dans le graphe complémentaire. Cette valeur varie entre -1 et 1 (voir l'exemple de la figure 5.4). Elle est basée sur l'idée intuitive qu'une partition en communautés est de bonne qualité si la partition dans le graphe complémentaire est de mauvaise qualité.

Cette métrique a plusieurs avantages. On constate que les deux partitions, $C = G$ (une partition contenant tous les sommets du graphe) et $C = \{C_1, \ldots, C_n\}$ (n communautés de taille 1), ont une qualité nulle ($Q = 0$). De plus, la qualité d'une partition aléatoire est également nulle avec une très forte probabilité. Enfin, le nombre de communautés n'intervient pas dans le calcul de la qualité.

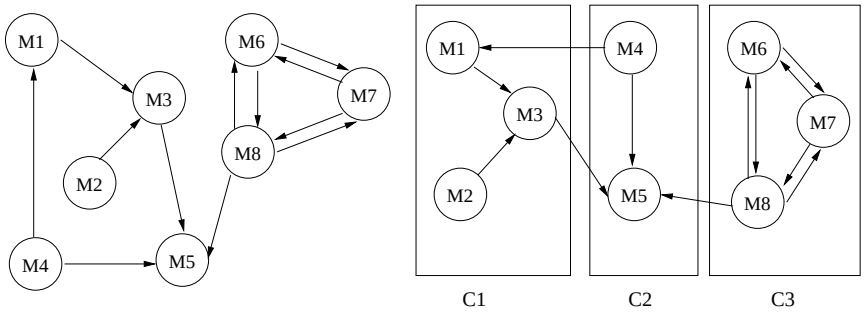


FIG. 5.4 – *Un exemple de calcul de notre qualité.* $Q(\mathcal{C}) = \frac{9}{12} - \frac{5}{44} \simeq 0.11$

Dans la prochaine section, nous allons décrire nos deux modèles pour l'émergence des communautés dans le Web : Le modèle gravitationnel et le modèle intentionnel. Ces deux modèles peuvent être utilisés pour tous les réseaux d'interactions et pas seulement pour le Web.

5.4 Le modèle gravitationnel

Nous proposons un modèle **particulaire** : les pages Web deviennent des *particules* évoluant dans un espace tridimensionnel. Les liens se traduisent en **forces** gravitationnelles s'exerçant sur les pages ; ainsi le mouvement de l'ensemble est induit par sa structure densément liée. Enfin, l'audience d'un page donne un **poids** à la particule.

Nous nous sommes inspirés du modèle cosmologique du Big Bang [53], qui décrit comment la matière, uniformément répartie dans l'univers à son commencement, a été façonnée en galaxies par deux actions : la **gravitation** et l'**expansion**. La première tend à **regrouper** les particules

qu'elle lie, tandis que la seconde, dilatation de l'espace qui *diminue* à mesure que l'univers vieillit, tend à **écarter** les particules sans relation. Notre modèle adapte ces deux phénomènes au Web. Ils agissent au cours du temps et isolent les ensembles densément liés de pages, qui conservent la somme de leurs masses et se regroupent en «globules». Ils nous permettent de proposer une nouvelle définition *par émergence* des communautés.

Nous nous sommes inspirés des lois physiques, et en particulier cosmologiques, qui produisent de bonnes métaphores pour décrire le monde d'un réseau d'interaction. Ainsi, notre modèle fait apparaître une tendance des page à se regrouper dans l'espace, au gré des forces gravitationnelles subies, en *galaxies*. Cette **floculation** permet d'inférer visuellement des communautés c'est à dire des pages densément connectées entre elles, que leur mutuelle attraction regroupe.

L'autorité des particules fournit une autre analogie avec la masse : une page de référence se comportera comme un soleil, immobile autour d'un nuage de planètes décrivant certaines caractéristiques structurelles des crawls du Web.

5.4.1 Modélisation de l'Univers

Notre modélisation distingue deux entités. La première est le graphe du Web $G = (V,E)$ où V désigne l'ensemble des pages. $(p,p') \in E$ si et seulement si il existe un lien de $p \in V$ vers $p' \in V$. Ce graphe est la donnée du problème, on considère ici un réseau orienté mais notre modèle s'adapte facilement aux réseaux non orientés. La deuxième entité est l'**espace** et le **temps** au sein desquels évolue le système. Pour obtenir un modèle se rapprochant le plus possible de la physique, nous avons pris l'espace euclidien $E = \mathbb{R}^3$, mais les nécessités de l'algorithmique nous ont fait choisir un temps discret $T = \mathbb{N}$. Les pages y sont présentes en tant que particules massives.

5.4.2 Modélisation des actions

Les forces mettent en mouvement les pages/particules. Une interaction gravitationnelle n'a lieu qu'entre pages unies par un lien (voir figure 5.5). Cette interaction respecte le principe galiléen *d'action et de réaction* : la force subie par la page pointée est la même que celle subie par la page qui pointe. Le sens du lien ne compte que pour le transfert de masse (voir § 5.4.3).

Nous avons utilisé simplement la force de gravitation newtonienne :

$$F_{pq} = G\frac{m(p).m(q)}{dist(p,q)^2}$$

Si une page possède plusieurs successeurs alors elle sera attirée lentement vers le successeur de plus forte masse. Sa trajectoire oblique (voir figure 5.6) est due au fait qu'à chaque étape, elle est attirée d'avantage par le successeur de forte masse.

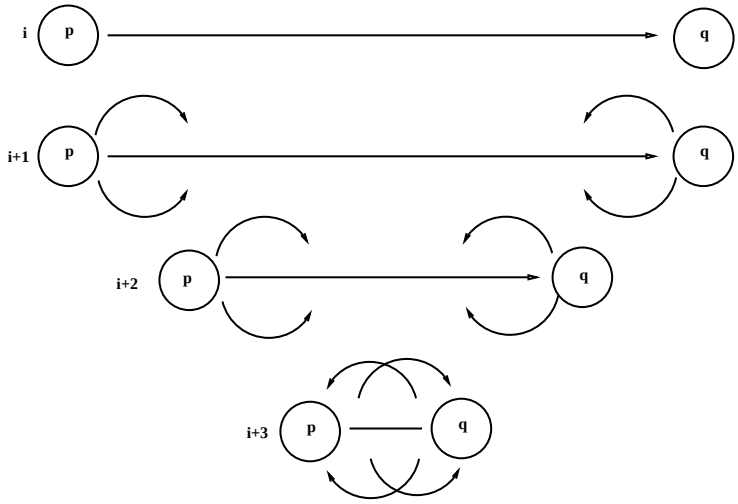


FIG. 5.5 – *Les particules se rapprochent sous l'effet de la force gravitationnelle.*

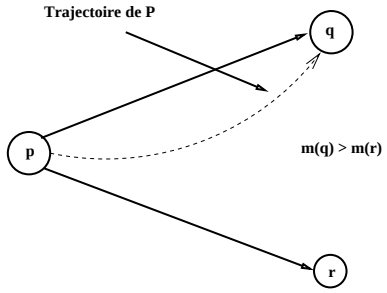


FIG. 5.6 – *Exemple de trajectoire de la particule p vers la particule q de forte masse. On remarque que la trajectoire est oblique.*

L'autre action subie par les particules est l'**expansion** de l'univers, qui les sépare au commencement. Nous avons pris une définition où l'univers a un centre O. Un point P de l'espace est translaté en un instant t en suivant :

$$FOP_{t+1} = (1 + \lambda e^{-\alpha t}).FOP_t$$

L'expansion s'arête asymptotiquement (assez vite, car $\sum e^{-\alpha t}$ converge).

Notre algorithme ne converge pas vers un état fixe, l'expansion permet de garder les communautés bien séparées les unes des autres. Le processus est arrêté au bout d'un certain nombre d'itérations.

En se basant sur les travaux de Newman (voir § 5.3 page 85) concernant la mesure de qualité d'une partition, nous avons supprimé la force d'expansion. Par conséquent, les communautés ont tendance à se regrouper sous l'effet de la force gravitationnelle mais le processus est arrêté

lorsque la valeur de la qualité est au maximum (on n'a pas besoin de prédéfinir un nombre d'itérations).

5.4.3 Modélisation des transferts de masse

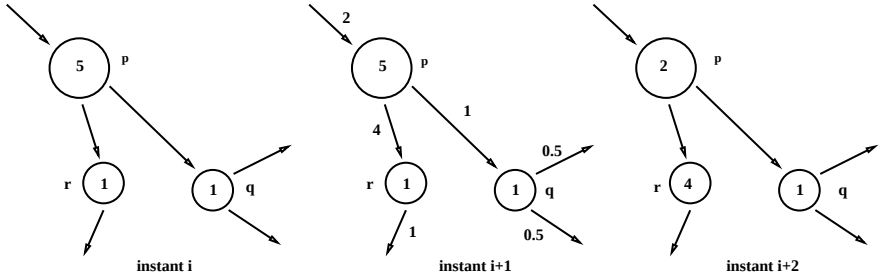


FIG. 5.7 – *Exemple de transfert de masse. p est deux fois plus proche de r que de q. Par conséquent, q reçoit quatre fois moins de masse.*

La masse représente l'*autorité* d'une page. Elle varie au cours du temps, ce qui viole le bon sens physique[1].

Les transferts de masse s'inspirent du modèle PageRank (voir § 4.2.2 page 71), en y ajoutant la notion de distance. À chaque étape, une page **répartit** la totalité de sa masse entre les pages qu'elle pointe. La masse circule donc le long des liens. Cette circulation respecte la loi des noeuds pour chaque sommet du graphe ; cela pose un problème pour les pages sans successeur. Une page transfère préférentiellement sa masse à ses proches voisines (renforcement local) ; la proportion de masse transférée est asymptotiquement nulle avec la distance. Enfin, la masse totale du système se conserve. Le transfert de masse blesse l'intuition du physicien dans la mesure où l'énergie ne se conserve pas. Mais il contribue au renforcement mutuel des pages spatialement proches en communautés (voir figure 5.7).

La masse d'une page p à l'étape t est définie comme suit :

$$m_t(p) = \sum_{q \to p} \frac{1}{dist_t(p,q)^\delta} \frac{m_{t-1}(q)}{S_t(q)} \quad \text{avec} \quad S_t(q) = \sum_{q \to r} \frac{1}{dist_t(q,r)^\delta}$$

Cette loi se ramène à celle de PageRank pour $\delta = 0$, la masse étant alors équitablement répartie entre les successeurs de la page, dont $S(q)$ est alors le degré sortant. Nous utilisons $\delta = 2$ par cohérence avec la force gravitationnelle.

Mais les pages sans successeur ne redistribuent pas leur masse (voir § 4.2 page 68). Pour résoudre ce problème, il est nécessaire d'introduire le facteur *facteur zap* (également appelé facteur d'amortissement) d brassant la masse totale du système, accélérant ainsi sa convergence.

1. Mais d'après Fabien Mathieu, c'est pour la bonne cause.

Le transfert de masse devient :

$$m_t(p) = d\Big(\sum_{q \to p} \frac{1}{dist_t(p,q)^\delta} \frac{m_{t-1}(q)}{S(q)}\Big) + (1-d) \sum_{r \in P} m_{t-1}(r)$$

Nous prenons $d = 0.85$ (comme dans Page et Brin [89]). Signalons enfin que les pages sans successeur et les erreurs d'arrondi font perdre de la masse au système. La masse est donc renormalisée après chaque itération pour que la masse totale se conserve.

5.4.4 Implémentation

Le calcul d'une itération (passage de l'instant t à $t+1$) se fait en temps linéaire par rapport au nombre de sommets et d'arcs du graphe. Les liens n'ont pas besoin d'être en mémoire : une seule passe le long du fichier des listes d'adjacence suffit à faire les calculs. Le facteur limitant est la *mémoire vive* plus que le temps, car chaque sommet occupe 32 octets (position, vitesse et masse), limitant à quelques dizaines de millions de sommets les expérimentations. Le programme pourrait facilement être parallélisé pour vaincre cette barrière. Le choix des constantes G, λ,α et d se fait empiriquement.

Graphes utilisés

Nous avons utilisé deux sortes de jeux de données : tout d'abord des graphes artificiels. Nous avons en particulier testé des **graphes petits mondes** (voir § 2.2.4 page 34) qui nous ont permis de vérifier que ces derniers se regroupaient bien en galaxies. Pour ce faire, nous avons utilisé un algorithme de réorientation aléatoire des arêtes proposé par Watts et Strogatz permettant de générer un graphe intermédiaire entre un graphe régulier et un graphe aléatoire sans altérer le nombre de sommets dans le graphe. Partant d'un graphe k-régulier à n sommets disposé en anneau, l'algorithme réoriente chaque arête avec une probabilité p. Leur construction leur permet de générer un graphe *petit monde* intermédiaire entre régularité ($p = 0$) et désordre ($p = 1$).

Nous avons également utilisé des *crawls*, parcours réels d'une partie du Web par des robots [90, 69], images forcément incomplètes du Web mais qui en donnent une bonne idée. Le graphe «théorique» et instantané diffère nécessairement des différents avatars que peut en fournir un crawler ; l'existence des pages dynamiques le rend potentiellement infini.

Conditions initiales

Notre modèle renforce la proximité des pages proches, il est donc très sensible aux conditions initiales. La méthode la plus simple est de répartir les pages dans l'espace de manière aléatoire. La deuxième consiste à prendre en considération la nature du réseau. Par exemple pour le Web, il est possible de placer les pages Web dans un cube.

En effet, nous avons pris le parti de faire la répartition initiale selon les sites. Les pages Web du crawl sont d'abord regroupées en un arbre : domaines/serveurs/répertoires (voir figure

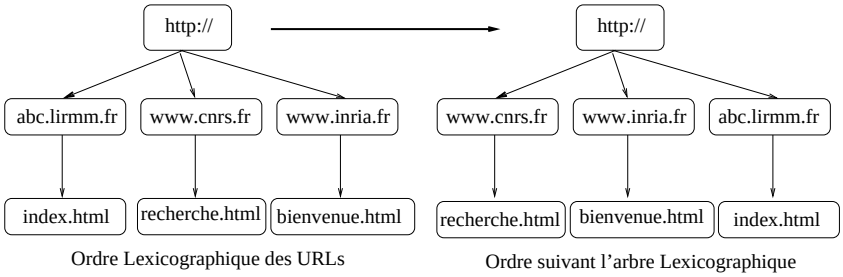


FIG. 5.8 – *Ordres des URLs.*

5.8). Puis cet arbre est parcouru en largeur. Une page à f fils donne naissance à un *cube* dans l'espace, dans lequel chacun de ses fils prend place comme cube de côté $\sqrt[3]{f}$, jusqu'aux feuilles qui donnent les points (voir figure 5.9).

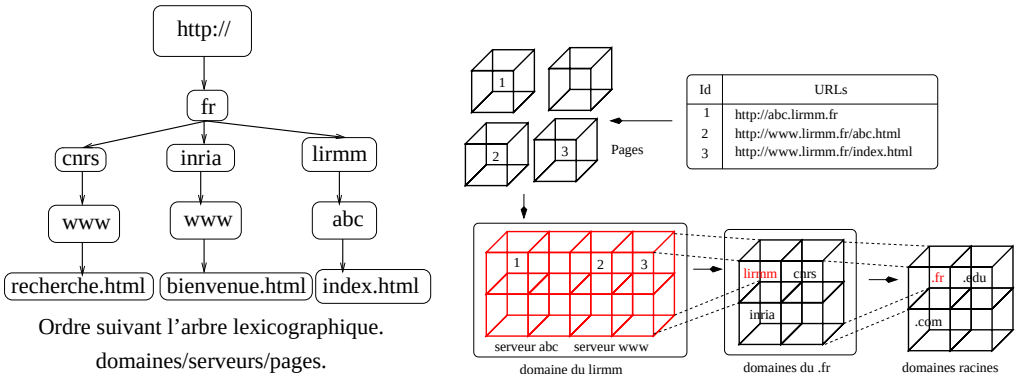
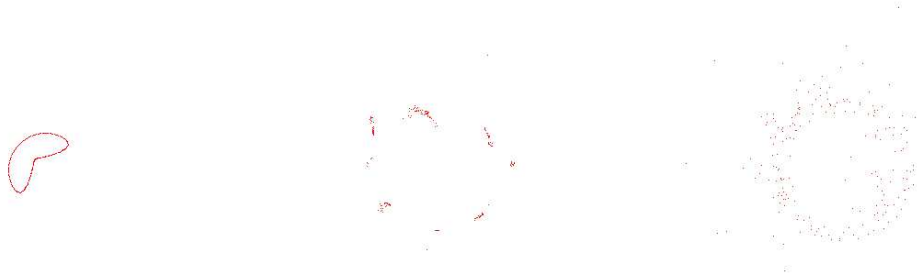


FIG. 5.9 – *Conditions initiales.*

Remarquons que les pages d'un même site sont très denses en **liens navigationnels** qui les lient les unes aux autres. Nous avons estimé leur densité à 95 % lors de nos expérimentations ! Ces liens doivent être préalablement enlevés, sous peine de ne détecter que les sites et non les communautés. Paradoxalement, rien ne lie donc les sommets proches initialement : chacun est libre de migrer vers sa communauté.

5.4.5 Résultats

Comme on peut le voir sur l'image B de la figure 5.10, avec notre modèle, les graphes *petits mondes* se regroupent bien en communautés. En revanche, les graphes totalement aléatoires se dissolvent rapidement dans l'espace si on laisse l'expansion agir (voir image C de la figure 5.10).



A. Un graphe 4-régulier de 200 sommets ($p = 0$)

B. Un graphe *petits mondes* ($p = 0.25$)

C. Un graphe totalement aléatoire ($p = 1$)

FIG. 5.10 – *L'anneau de Watts et Strogatz.*

Au contraire, les graphes réguliers se regroupe en une seule partie (voir image A de la figure 5.10).

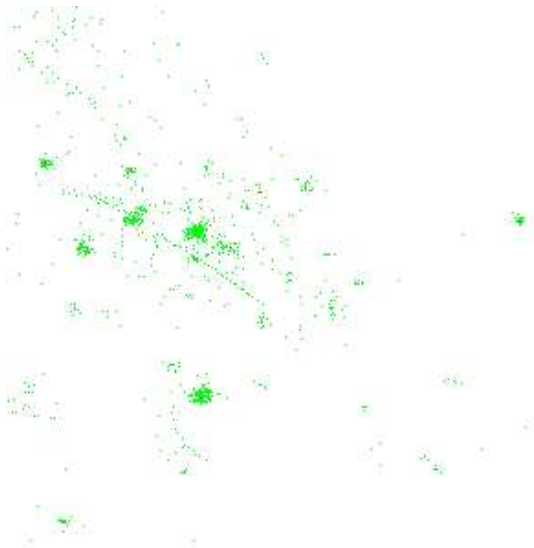
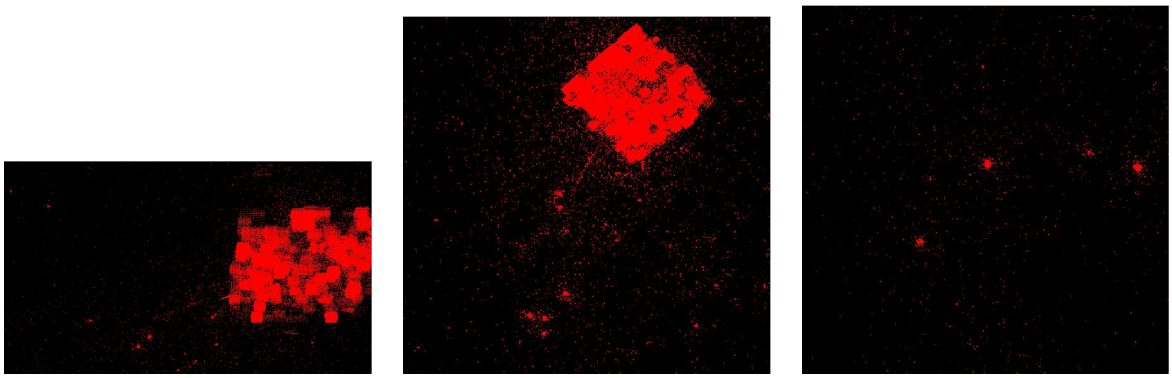


FIG. 5.11 – *Crawl de 8,000,000 pages (sans les liens de navigation).*

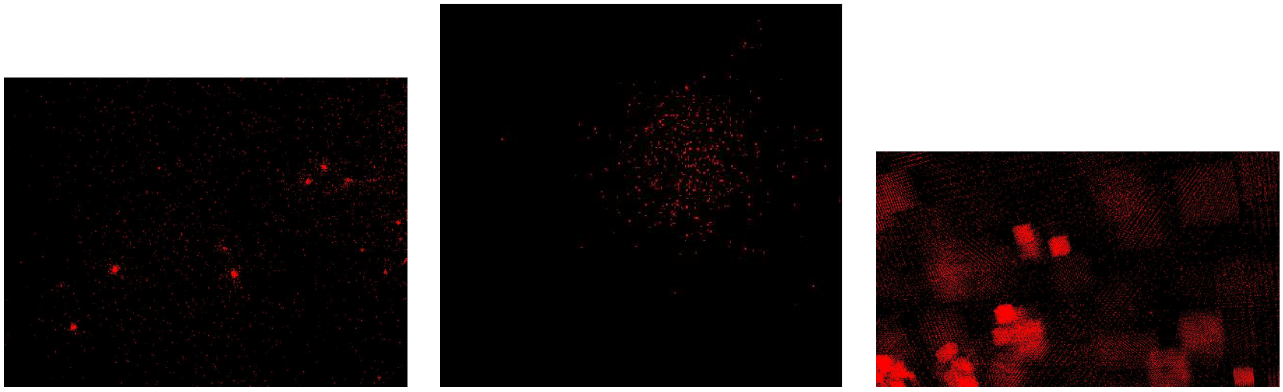
Enfin, la figure 5.11 représente un crawl de 8 millions de pages. En quelques itérations, nous voyons se former à l'écran des communautés. Par ailleurs, nous avons constaté que 80 % des pages ont tendance à quitter leur emplacement d'origine (site) pour migrer vers une communauté.



Position initiale en cubes | Début de formation des communautés | Quelques communautés formées

FIG. 5.12 – *Les communautés du graphe du Web.*

Les figures 5.12 et 5.13 montrent les différentes étapes de notre algorithme. Initialement les pages Web sont positionnées dans des cubes. Dès les premières itérations les communautés se forment. Les pages Web sans lien restent elles immobiles. On remarque que les communautés sont éloignées les unes des autres. Si l'on se rapproche d'une communauté, on remarque qu'une communauté n'a pas une structure bien précise.



Communautés éloignées les unes des autres | Communauté sans structure précise | Les pages Web immobiles

FIG. 5.13 – *Les communautés du graphe du Web.*

Partant d'une idée originale de représentation des réseaux d'interactions, nous avons aussi

trouvé une définition alternative de la notion de communauté, qui donne des bons résultats. On a beaucoup dit que le graphe du Web possède une structure de *petit monde* : notre modèle en fournit une preuve *visuelle*, les mondes en question se formant et tournant effectivement à l'écran !

Certains réseaux d'interactions sont très dynamiques, et notre modèle se prête particulièrement bien à l'évolution des pages et des liens. Sur le plan algorithmique, ce n'est en revanche pas aussi simple ; il serait intéressant de travailler dans ce sens. De plus, il est souvent dit qu'un bon modèle est un modèle possédant très peu de paramètres (contrairement au notre).

Pour réponde a ces questions, nous avons modifié le modèle gravitationnel et ajouté une nouvelle métrique mesurant la qualité de nos partitions. Dans la prochaine section, nous présentons le modèle intentionnel qui est une amélioration du modèle gravitationnel.

5.5 Le modèle intentionnel

Dans ce modèle les pages d'un réseau sont comme dans le modèle précédent, c'est-à-dire des particules qui se déplacent dans un espace tri-dimensionnel. Les particules sont positionnées sur la surface d'une sphère. Elles sont dotées d'un but qui consiste à atteindre sa communauté.

5.5.1 Description du modèle

On utilise un espace sphérique. Chaque particule est positionnée sur la surface de la sphère. Cette position est définie en utilisant les deux coordonnées sphériques ρ et ϕ ou les trois coordonnées cartésiennes x, y et z. À chaque page u du réseau correspond un point U sur la surface de la sphère.

La distance $d(u,v)$ entre les pages u et v est définie par la distance euclidienne $\|\overrightarrow{UV}\|$ (voir annexe A, page 118). Le diamètre de la sphère est sans perte de généralité fixée à 1. Les particules évoluent dans le temps, nous avons choisi un temps discret $T = \mathbb{N}$.

Chaque page est dotée d'un poids. Ce dernier peut être choisi selon plusieurs métriques : degré (entrant ou sortant) de la page, PageRank de la page, la valeur d'autorité et d'annuaire (HITS) de la page, etc. Dans notre cas, nous avons utilisé le PageRank. Le poids d'une particule u est noté w_u.

Chaque particule a un objectif bien précis. À chaque étape, chaque particule calcule son objectif et effectue un déplacement vers cet objectif.

5.5.2 Identification de l'objectif

Nous voulons que les communautés soient des pages présentes au même point de l'espace. Nous supposons que les communautés sont tissées par les liens. Ainsi, si les pages de la communauté à laquelle la page p appartient se trouvent dans un endroit de l'espace alors la page p doit se déplacer vers cette endroit.

Une page est «*attirée*» par les pages voisines. L'attraction est proportionnelle aux poids des pages voisines : plus une page a un fort poids, plus elle peut attirer des pages vers elle. Il faut donc une attraction croissante en w et décroissante en d. Nous posons $\frac{w}{d}$.

Dans le cas d'un réseau orienté, nous supposons que le sens n'a pas d'importance, une page est attirée par ses successeurs de la même façon que par ses prédécesseurs. Ainsi, les pages sans successeur seront également attirées vers leurs communautés.

L'objectif d'une page p est le barycentre du voisinage de p, chaque voisin q de p étant affecté d'un coefficient $\frac{w_q}{d(p,q)}$ (voir figure 5.14). L'objectif de p est donc le seul point G solution de l'équation :

$$\sum_{q \leftrightarrow p} \frac{w_q}{d(p,q)} \overrightarrow{GQ} = \overrightarrow{0}$$

5.5.3 Déplacement vers l'objectif

Une page se déplace donc en direction de son objectif. Nous avons limité à s la distance que peut parcourir une page à chaque étape. Si la page p est à une distance d de son objectif, il lui faut $\lceil d/s \rceil$ étapes (itérations) pour atteindre son objectif. On prend $s = 0.05$, il faut donc une vingtaine de pas pour traverser la sphère.

En cas de préférences symétrique, il est fort possible d'avoir une instabilité du système. Prenons un exemple simple où une page p a un objectif qui consiste à atteindre la position de la page q et inversement. Les deux pages vont continuellement échanger leurs positions. Pour résoudre ce problème, si la distance entre les deux est inférieure à s, la page ne peut parcourir que 90% de la distance. Cela assure une convergence (rapide) sur la même position.

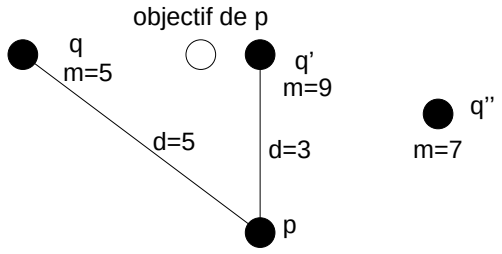


FIG. 5.14 – *Objectif de la particule p, qui est le barycentre de ses voisins, ici q et q'.*

Les différentes expérimentations ont montré que l'utilisation de la distance euclidienne

$$d_e(p,q) = \|\overrightarrow{PQ}\|$$

ou de la distance sphérique

$$d_s(p,q) = 2 \arcsin(\frac{1}{2} d_e(p,q))$$

donnent les même résultats. Elles sont en effet presque identiques pour de petite valeurs.

Néanmoins, le barycentre n'est pas défini dans la sphère. En fait, le barycentre est calculé dans $\mathbb{R}^3$ et projeté sur la surface de la sphère. Initialement le système n'est pas très stable, en effet, comme les sommets sont positionnés aléatoirement, les barycentres sont généralement proches du centre de la sphère mais au bout de quelques itérations le système se stabilise.

5.5.4 Forme de l'univers

À l'état initial, les sommets du graphe sont placés aléatoirement sur la surface de la sphère. Les résultats obtenus montrent que les communautés retrouvées dépendent de l'état initial. En d'autres termes, pour deux états initiaux, on obtient deux partitions différentes mais de qualité très proche.

5.5.5 Extraction de communautés

Nous utilisons une heuristique simple de calcul de communauté ; chaque page a un *chef de communauté* ; les communautés sont les pages de même chef. À chaque étape le chef de communauté d'une page devient le chef de communauté de la plus lourde page située à distance moins que R_1 d'elle. Si une page s'éloigne à distance plus que $R_2 = 2R_1$, de son chef de communauté, alors elle reprend son indépendance (devient son propre chef). Le chef de communauté est ainsi (à peu prés) la plus lourde page au centre d'une sphère de rayon R_1 (voir figure 5.15).

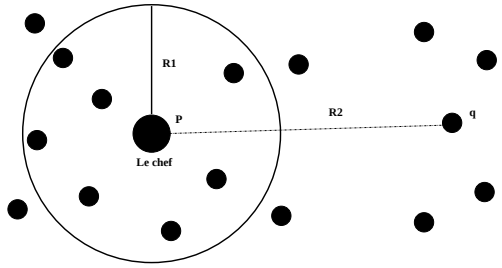


FIG. 5.15 – *Émergence d'une communauté. Les sommets à une distance $R1$ du représentant sont regroupés pour former une communauté. Si un sommet est à une distance $R2 = 2R1$ du représentant alors il peut décider de quitter la communauté.*

5.5.6 Condition d'arrêt

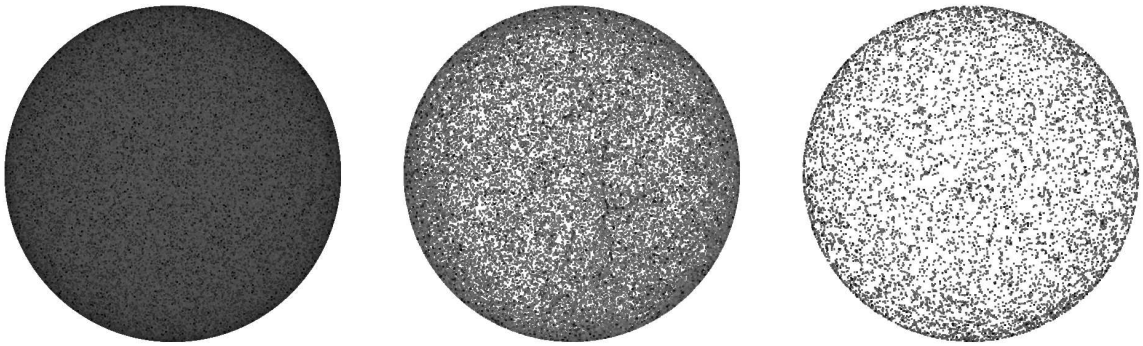
À chaque étape, la mesure de qualité du partitionnement peut être calculée en utilisant notre définition (voir § 5.3.2 page 85). Au bout d'un temps infiniment long, tous les sommets d'un

graphe connexe s'effondrent en un seul point. Mais cela correspond à une qualité nulle. Entretemps la qualité aura atteind son maximum. La qualité augmente à chaque étape jusqu'à atteindre sa limite. Lorsque cette limite est atteinte, le processus peut être arrêté.

5.5.7 Résultats

Les graphes utilisés

Graphe du Web : Nous avons utilisé un graphe du Web de $800{,}000$ pages Web et de 4.5 millions d'hyperliens. Ce crawl a été obtenu à partir de différentes expérimentations de crawl de Web de janvier 2004 et de mai 2004.



$1^{\text{ère}}$ itération $100^{\text{ème}}$ itération $1000^{\text{ème}}$ itération

FIG. 5.16 – *Communautés d'un graphe du Web.*

La première sphère (voir figure 5.16) montre les positions initiales des particules sur la sphère. À chaque itération, toutes les particules se déplacent. Les deux dernières sphères (voir figure 5.16) montre l'évolution du système après 100 et 1000 itérations. Des zones blanches apparaissent indiquant la formation de communautés.

Graphe aléatoire : Nous avons généré un graphe aléatoire d'Erdös et Rényï (voir chapitre 2.1) de 1 million de sommets et approximativement 7 millions d'arcs. Le graphe aléatoire contient une composante fortement connexe regroupant 99,815% des sommets.

Avec ce graphe aléatoire après quelques itérations (approximativement 150), les particules fusionnent pour former quelques communautés (approximativement une centaine).

Graphe aléatoire clusterisé : C'est un graphe construit à partir de différents graphe aléatoire de type Edrös et Rényi. On part d'un graphe aléatoire de 10 sommets et 140 arcs, ensuite on génère un deuxième graphe avec 20 sommets et 340 arcs, ce dernier est relié au premier graphe avec une probabilité de p : c'est-à-dire, pour deux graphes aléatoires de taille respective n et m,

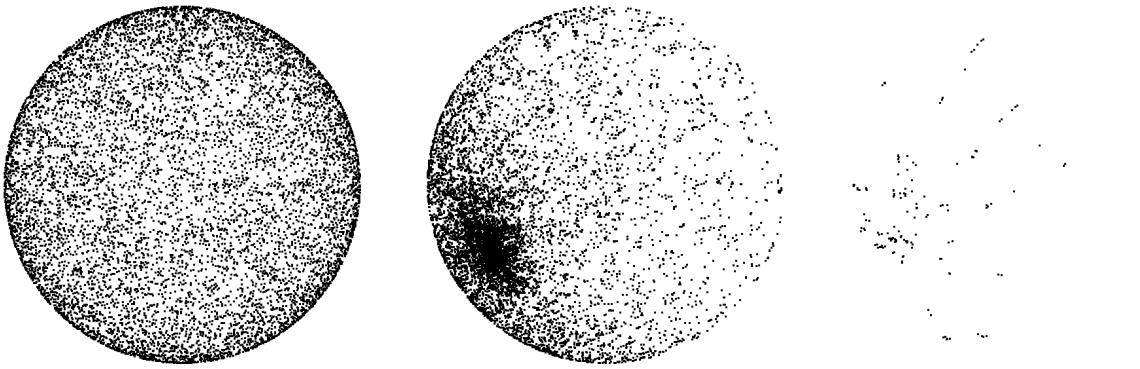


FIG. 5.17 – *Les communautés dans un graphe aléatoire.*

on aura, $n * m$ arêtes possibles, on tire avec une probabilité p, l'existance d'une arête. Donc, à la fin, on a $n * m * p$ arêtes entre les deux graphes. Ainsi de suite, des graphes aléatoires de plus en plus gros sont ajoutés au reste du graphe jusqu'à atteindre 1000 sommets.

Le graphe obtenu contient plusieurs composantes fortement connexes (approximativement le même nombre de graphes aléatoires utilisés pour construire le graphe) de différentes tailles (approximativement de même taille que les graphes aléatoires ajoutés, c'est-à-dire entre 1 et 200). On a ainsi des communautés de taille hétérogène (là où usuellement les algorithmes de clustering échouent).

Avec ce graphe (voir figure 5.17), après quelques itérations (approximativement 265), les particules fusionnent pour former des communautés. Le nombre de communautés est égal au nombre de graphes aléatoires utilisés dans la génération.

Qualité du partitionnement En utilisant la métrique définie en section 5.3.2, nous calculons à chaque itération la qualité du partitionnement. La qualité est nulle à l'état initial car chaque sommet est considéré comme une communauté.

Pour le graphe du Web (voir figure 5.18) et le graphe aléatoire clusterisé (voir figure 5.19), la qualité augmente rapidement atteignant son maximum (97%) après quelques centaines d'itérations (300), ensuite baisse lentement.

Pour le graphe aléatoire (voir figure 5.20), la qualité est très mauvaise (8%) et après seulement 100 itérations, les particules fusionnent formant une seule communauté avec une qualité de 0.5 %. Cette mauvaise qualité est due au fait que le graphe contient une composante géante fortement connexe et n'est pas clusterisé.

Plus précisément, on observe une phase de transition au bout de la $100^{\text{ème}}$ itération, la qualité s'effondre très rapidement. Cette phase de transition s'explique par le fait que les graphes aléatoires sont différents des réseaux d'interactions, notamment concernant le coefficient de re-

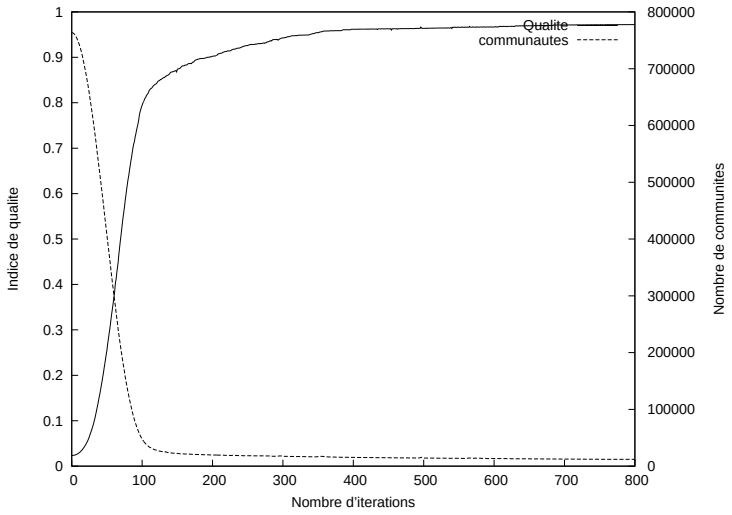


FIG. 5.18 – *Crawl du Web.*

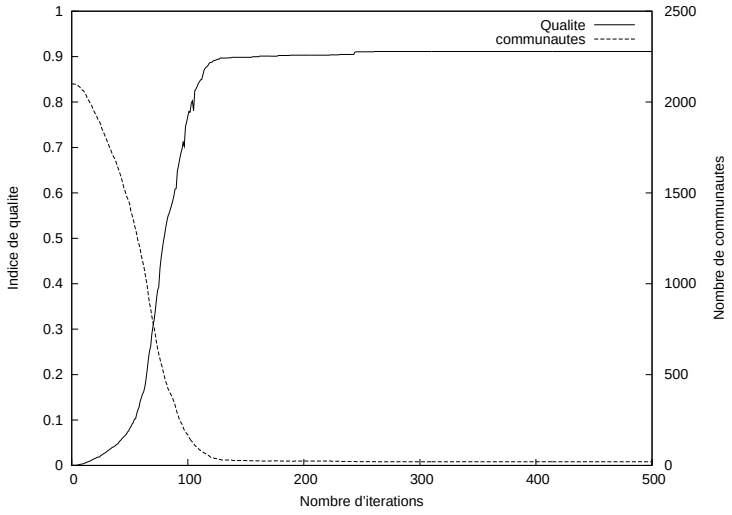


FIG. 5.19 – *Graphe aléatoire clusterisé.*

groupement : les réseaux d'interactions sont fortement regroupés contrairement aux graphes aléatoires. Les réseaux d'interactions ont également une distribution des degrés différente de celle des graphes aléatoires (voir section 1.3 page 19).

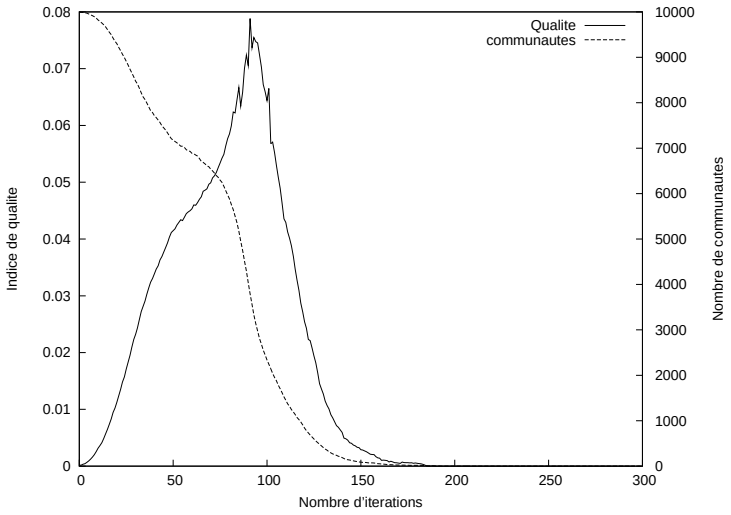


FIG. 5.20 – *Graphe aléatoire.*

5.6 Conclusion

Dans la deuxième partie de ce mémoire, nous avons d'abord présenté deux algorithmes de classification de pages Web : PageRank et HITS. Le premier assigne à chaque sommet du graphe un poids indépendemment de la requête de l'utilisateur. Le deuxième assigne à un sous-ensemble de pages Web deux poids : un poids d'autorité et un poids d'annuaire.

Le PageRank d'une page Web n'est rien d'autre que la probabilité qu'un surfeur aléatoire découvre la page Web lors de son parcours. L'algorithme PageRank est également utilisé pour orienter les crawlers lors de la phase de recherche de nouvelles pages Web. Il est calculé sur un sous-graphe du Web et les pages de fort PageRank sont crawlées en premier.

L'algorithme HITS est utilisé dans plusieurs domaines, notamment pour la recherche de communautés et pour la recherche de pages liées à une page Web donnée. Ces deux algorithmes sont importants dans le domaine de la recherche d'informations sur le Web et tous deux sont basés sur un calcul itératif de vecteurs propres.

Ensuite, nous avons présenté deux modèles pour la détection et l'extraction de communautés dans les réseaux d'interactions. Le modèle **particulaire** où les pages du réseau deviennent des *particules* évoluant dans un espace tridimensionnel. Les liens se traduisent en **forces** gravitationnelles s'exerçant sur les pages ; ainsi le mouvement de l'ensemble est induit par sa structure densément liée. Enfin, l'audience d'un page donne un **poids** à la particule. Dans le modèle intentionnel, les pages d'un réseau sont comme dans le modèle précédent, c'est-à-dire, des particules qui se déplacent dans un espace. Les particules sont positionnées sur la surface d'une sphère. Elles sont dotées d'un but (objectif) qui consiste à atteindre sa communauté. Ces deux modèles utilisent les liens entre particules pour faire émerger les communautés et le poids des particules

pour accélérer la formation des communautés.

Notre problématique est différente de celle des problèmes de détection de communautés (clustering). En effet, dans certains problèmes de clustering, le nombre de communautés est fixé. Notre approche ne fixe pas *a priori* le nombre de communautés et l'algorithme de clustering dépend seulement du graphe utilisé. En d'autres termes, dans nos modèles, les communautés émergent grâce aux rapports de force entre les différentes particules.

La taille des communautés n'apparaît pas dans le calcul de la qualité, seuls les arcs sont nécessaires. Notre métrique donne une bonne qualité pour la recherche de cliques et de noyaux (bicliques denses). Elle est également plus robuste au bruit contrairement aux autres méthodes.

Nous avons testé nos algorithmes sur des crawls du Web et nous avons obtenu des communautés mais il est impossible de dire si ces dernières ont du sens, sauf si une étude sémantique sur les clusters est faite. Vu la taille du graphe, cette solution paraît difficile à réaliser.

Notre algorithme de calcul de communauté par émergence est capable de manipuler de grands réseaux d'interactions et calcule les communautés au bout d'une centaine d'itérations. La complexité en temps de notre algorithme est en $O(m)$ pour chaque itérations. Cela est bien plus rapide que les algorithmes usuels [58, 10, 65].

Conclusion Générale

Nous avons abordé dans cette thèse deux des grandes problématiques de l'étude des grands réseaux d'interactions : la modélisation et le calcul d'une partition en communautés (clustering).

—

Nous avons ainsi décrit dans le premier chapitre un éventail relativement large de grand réseaux d'interactions rencontrés et étudiés en pratique. Il est surprenant de constater que malgré leur nature différentes, ils partagent des propriétés importantes à savoir une distribution des degrés en loi de puissance de paramètre compris entre 2 et 3, une distance moyenne faible, un fort coefficient de clustering, une densité faible, une composante connexe géante et une abondance de bicliques. Certaines de ces propriétés, telles que la distance moyenne faible et la composante connexe géante apparaissent dans beaucoup de graphes. Néanmoins, la densité faible, le fort coefficient de clustering et la distribution des degrés en loi de puissance sont des caractéristiques propres aux réseaux d'interactions. Soulignons que la plupart des réseaux étudiés ne sont obtenus qu'après une opération de collecte relativement complexe qui ne donne qu'une vue partielle de l'objet réel. La question de la représentativité de l'échantillon obtenu, et du biais introduit par la méthode de collecte, a pour l'instant été ignorée. Il est arbitrairement admis qu'un échantillon de taille maximale pouvait être considérée comme exhaustif.

—

Dans le deuxième chapitre, nous avons présenté plusieurs modèles existants permettant de générer des graphes avec une ou plusieurs caractéristiques des réseaux d'interactions. Il est ressorti de ce chapitre qu'il n'existe pas de modèles capables de regrouper toutes les propriétés communes des réseaux d'interactions. Les modèles présentés peuvent êtres regroupés en deux catégories. La première regroupe des modèles avec de nombreux paramètres tels qu'il devient difficile d'évaluer leurs pertinences expérimentalement. La deuxième catégorie contient des modèle avec un nombre de paramètres réduit permettant de générer des graphes avec quelques propriétés spécifiques, exemple avec une distribution des degrés en loi de puissance. Cette dernière catégorie génère des graphe peu réalistes et très peu utilisés en pratique.

—

Le chapitre trois décrit un nouveau modèle de crawls aléatoires permettant de générer des graphes artificiels ayant les mêmes propriétés que les crawls du Web. Il repose sur une observation faite sur la manière dont un crawl est obtenu, à savoir un parcours du Web. Notre modèle

simule tout simplement un parcours de graphe (en profondeur, en largeur, etc). Il ne nécessite que deux paramètres, mais génère des crawls aléatoires ayant beaucoup de propriétés communes avec des crawls réels du Web. En revanche, il est très difficile de l'analyser formellement. En effet, les arêtes sont dépendantes les unes des autres, ce qui fait échouer les méthodes classiques. À notre connaissance, c'est le premier modèle dont le but est de capturer non le graphe du Web, mais un crawl du Web.

—

Une description des algorithmes de classification de pages Web a été donnée dans le chapitre quatre. On y a présenté la forme simple du calcul du PageRank qui est une modélisation d'une marche aléatoire du graphe du Web. Ce calcul n'est possible que si le graphe possède certaines propriétés. Ensuite, a été présenté la forme pratique qui elle est utilisée pour tout graphe. L'algorithme de PageRank affecte à chaque page Web un rang qui ne dépend que de la structure hypertexte du Web. L'algorithme HITS, contrairement à l'algorithme PageRank, dépend de la requête de l'utilisateur et assigne à une page Web deux poids : d'autorité et d'annuaire. L'algorithme HITS est utilisé pour rechercher des pages Web liées à une page donnée dans un graphe de voisinage et calculer des communautés sur le Web.

—

Le dernier chapitre est consacré aux algorithmes de clustering permettant la décomposition d'un graphe en plusieurs parties, telle que chaque partie est dense en liens mais très peu de liens existent entre les parties. Nous avons proposé deux algorithmes de calcul de communautés par émergence. L'idée principale est de permettre à une pages Web de se déplacer dans un espace tridimensionnel dans le but de rejoindre sa communauté. Nous observons une convergence rapide de nos deux algorithmes; la qualité obtenue est toujours bonne et une observation "manuelle" des communautés obtenues est convaincante.

—

Malgré les différents modèles décrit dans le chapitre 2, Le problème de la modélisation des réseaux d'interactions est un domaine de recherche récent, ouvert et très actif. Ce travail mérite donc d'être continué.

Il est intéressant de développer des algorithmes capables de manipuler des grands réseaux d'interactions. Ces algorithmes pourraient tenir compte des propriétés communes des réseaux d'interactions.

Il est également intéressant de s'intéresser aux problèmes de visualisation des grands graphes. en effet, la visualisation des propriétés des réseaux d'interactions permet de faire une comparaison entre les différent réseaux réels ou simulés. Or notre algorithme de clustering peut donner un algorithme de visualisation.

Annexe A : Historique du Web

ARPANET est considéré comme le réseau précurseur d'INTERNET, créé en 1969 par ARPA (Advanced Research Project Agency) dans le but de relier efficacement les centres de recherches entre eux (le Stanford Institute, l'université de Californie à Los Angeles, l'université de Californie à Santa Barbara et l'université d'Utah). Ce réseau devait être capable d'une part de partager les ressources informatiques et d'autres part d'échanger du courrier électronique.

Le rôle principal d'ARPANET devait être de pouvoir résister à toutes sortes de pannes (pannes de machines, coupures de lignes de communication, etc.). Pour cela, le réseau ne devait présenter aucun point névralgique dont l'arrêt ou le dysfonctionnement pourraient avoir pour conséquence le blocage total du réseau. En cas de problème sur une partie du réseau, les données transitant par le réseau devait être redirigées automatiquement par un autre chemin. De plus les protocoles de communication devaient être simple.

Plusieurs chercheurs de différentes universités américaines ont été mobilisés pour réaliser un tel réseau. C'est en 1972, lors de la conférence ICCC (International Computer Communication Conference) que le réseau ARPANET fut présenté pour la première fois au grand public. Le terme «internetting» est utilise pour designer l'ARPANET, devenant ainsi un embryon d'internet.

Aujourd'hui,Internet[1,2] regroupe un ensemble de réseaux de nature plus ou moins hétérogène, disséminés aux quatre coins du monde. Tous ces réseaux sont interconnectés à l'aide de liaisons très diverses (RTC, ADSL, fibre optique, satellites, Wifi,etc.). Tous ces différents réseaux tissent une sorte de gigantesque toile d'araignée d'un bout à l'autre de la planète (voir figure 21).

Protocole TCP/IP

Pour que deux personnes communiquent, il est nécessaire d'utiliser le même langage que les deux personnes comprennent. De même pour les machines, pour que plusieurs machines puissent communiquer entre elles, il est nécessaire qu'elle utilisent un protocole de communication qui est un ensemble de règles respectées par les deux parties communicantes pour permettre le bon déroulement de l'échange d'information.

Le protocole permettant à deux entités d'un réseau de communiquer s'appelle le protocole

1. http://www.isoc.org/
2. http://www.ietf.org/

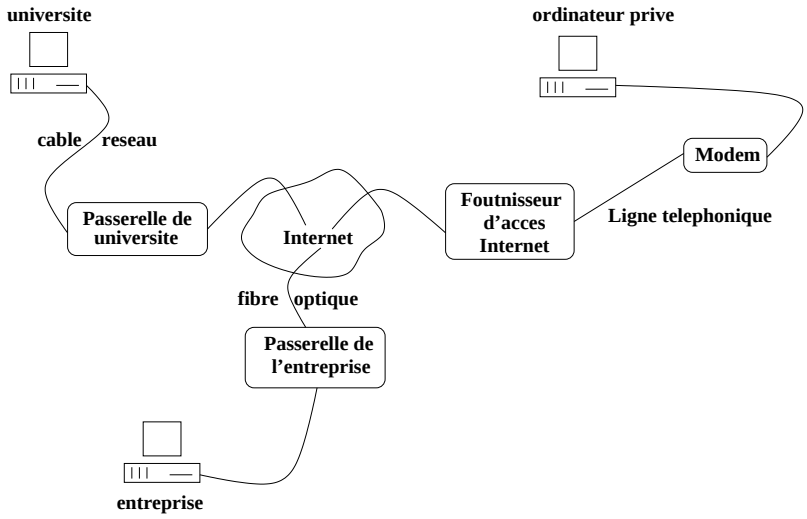


FIG. 21 – *Réseau Internet.*

TCP/IP (Transmission Control Protocol/Internet Protocol). Comme son nom l'indique, c'est une fusion de deux protocole : TCP et IP. Le protocole TCP/IP constitue une solution extrêmement pratique et fiable pour des environnements hétérogènes. L'emploi du protocole TCP/IP permet de connecter un grand nombre d'ordinateurs gérés par différents systèmes : Windows, Macintosh et Unix. Tous utilisent le protocole TCP/IP comme norme de communication pour la connexion Internet [96].

Le protocole IP

Il permet de gérer l'adressage et le routage des données, c'est-à-dire leur acheminement via le réseau, mais ne garantit pas qu'elles y arriveront toutes car des problèmes d'encombrement peuvent provoquer la destruction de paquets de données en attente et des erreurs de transmission peuvent entraîner leurs pertes [99].

Le protocole TCP

Il est chargé du transport des données, de leur découpage en paquets et de la gestion des éventuelles erreurs de transmission. Il est donc conçu pour empêcher ces erreurs et garantir une connexion fiable de bout en bout [100].

Les données sont découpées en petits paquets appelés datagrammes (unité de transport de l'information dans un réseau) dont la taille peut varier selon le type de réseau. Ils contient en plus des données : l'adresse de la station émettrice, celle de la station réceptrice et le numéro de port.

La plupart des applications sur Internet telles que Telnet (Terminal network Protocol), FTP (File Transfert Protocol), SMTP (Simple Message Transfert Protocol), HTTP (Hyper texte transfert protocole) utilisent le protocole TCP.

Les adresses IP

Chaque machines (hôte) sur Internet a une adresse *unique* de la forme : $V.W.X.Y$ où V,W,X,Y $\in [0,255]$ sur quatre octets. Elle permet d'identifier une machine sur le réseau. Si à partir d'une machine, on souhaite envoyer un message à une machine distante, il suffit de la designer par son adresse IP unique. Le protocole IP s'occupe d'acheminer le message vers la machine distante.

Une adresse IP se décompose en deux parties : la première représente le numéro du réseau auquel la machine appartient et la deuxième partie représente le numéro de la machine dans le réseau auquel elle appartient.

Suivant la taille du réseau, ce dernier peut être divisé en 5 classes : A, B, C, D et E. La classe d'un réseau est identifié en utilisant le premier octet de l'adresse IP (V), le reste ($W.X.Y$) vont être utilise pour identifier les machines dans le réseau. Le calcul est simple, un réseau de classe A peut contenir environs 255^3 machines.

Classe A : toute adresse dont $V \in [0,127]$, exemple $18.X.Y.Z$;
Classe B : toute adresse dont $V \in [128,191]$ exemple $149.X.Y.Z$;
Classe C : toute adresse dont $V \in [192,223]$ exemple $193.X.Y.Z$;
Classe D : toute adresse dont $V \in [224,239]$;
Classe E : toute adresse dont $V \in [240,255]$;

Avec une adresse IP sur 4 octets (IPv4), il est possible d'adresser un peu plus de 4 milliards de machines. Dans les années 90, ce nombre était largement suffisant. Mais avec l'avènement du Web, il s'avère aujourd'hui insuffisant. On assiste depuis quelques années a une saturation d'adresse IP. Pour résoudre ce problème, les adresses IP ont été rallongées passant de IPv4 vers IPv6.

Depuis 1995, les adresse IPv6 cohabitent avec les adresses IPv4. Une adresse IPv6 est longue de 128 bits, contre 32 pour IPv4. On dispose ainsi d'environ $3{,}4 \times 10^{38}$ adresses.

Les noms IP

Partant du principe qu'un nom est plus facile à retenir qu'une suite de chiffres, un système de nommage a été mis sur pied pour faire correspondre l'adresse IP numérique à un nom IP. La première structure hiérarchique, imaginée aux USA, a été basée sur la différenciation des catégories d'utilisateurs :

com Commercial
edu Education
gov Gouvernemental
int International
mil Militaire
net Réseau
org Organisation

Comme pour l'adresse IP, le nom se structure en groupes de caractères, de longueurs variables, séparés par un point (par exemple, ordinateur.subdomain.domain).

Ces catégories, appelées *top-level domains* sont vite apparues insuffisantes et il a été nécessaire de les compléter par les codes pays à 2 lettres des qu'Internet s'est fortement déployé hors des USA :

fr France
ca Canada
jp Japon
eg Égypte
cn Chine
de Allemagne
dz Algérie
...

Le serveur DNS

Le DNS (Domain Name Service) est le serveur des noms d'Internet. Comme nous l'avons souligné précédemment, toutes les machines connectes à Internet ont une adresse IP et donc un nom de domaine. Ces noms servent à trouver les adresses des machines et à construire les adresses IP [97, 98].

Le DNS permet aussi l'opération inverse, c'est-à-dire retrouver le nom de domaine correspondant à une adresse IP puisque les noms de domaines correspondent à une adresse IP. Les noms de domaines ont une structure hiérarchique. L'avantage principal du DNS est comme nous l'avons souligné d'offrir la possibilité à l'utilisateur de manipuler des formes «textuelles» plus faciles à mémoriser que l'adresses «numériques».

Le systeme de nommage DNS, utilise de nos jours, fut mis en œuvre en 1984, afin de pallier le manque de souplesse du nommage par table de nommage, demandant la mise à jour manuelle des correspondances entre le noms de machines et leur adresse sur des fichiers textes sur chacune des machines. Il est organisé de façon hiérarchique pour garantir l'unicité des noms [3] (voir figure 22).

3. http://www.renater.fr/Services/Procedures/Noms_Domaine2.htm

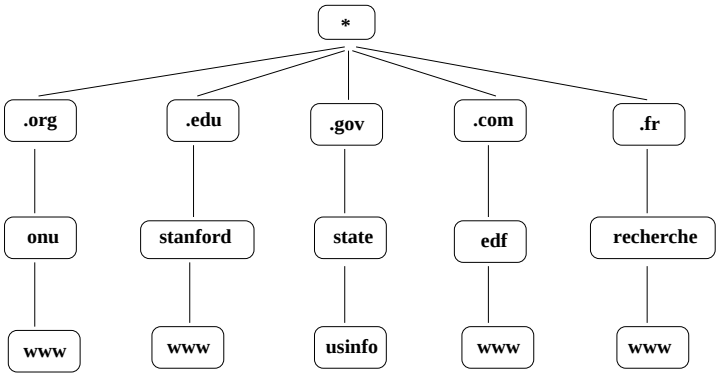


FIG. 22 – *Structure Hiérarchique du DNS.*

Le World Wide Web

En 1980, Tim Berners-Lee[4] proposa un projet de gestion de l'information dans le cadre d'une mission de conseil pour le compte du CERN (Organisation européenne pour la recherche nucléaire). Ce projet nommé *Enquire* contenait les prémices de l'hypertexte. Dans le cadre de ce projet, il développa un premier logiciel de stockage d'informations utilisant des associations aléatoires. Ce projet est la base du World Wide Web.

En 1989, il propose un projet d'hypertexte global qu'il appelle WorldWideWeb (il avait pense aussi à l'appeler Information Mesh (maillage d'informations), Mine of Information, ou encore Information Mine (mine d'informations)) et qui deviendra le World Wide Web d'aujourd'hui. Le but est de permettre aux gens de travailler ensemble en combinant leurs connaissances de façon à former une toile de documents hypertextes («toile»).

Fin 1990, Tim Berners-Lee met au point le protocole HTTP (Hyper Text Tranfer Protocol), le premier serveur HTTP (httpd) et le premier navigateur Web qu'il nomme «WorldWideWeb». Enfin, il crea le langage HTML (HyperText Markup Language) permettant de naviguer à l'aide de liens hypertextes, à travers les réseaux. Le World Wide Web est né. Le logiciel «World Wide Web» fut disponible pour la première fois en décembre au CERN pour usage interne, puis rendu disponible sur l'ensemble d'Internet au cours de l'été 1991. Tim Berners-Lee est à l'origine de standards parmi les plus utilisés comme HTTP, URL (Uniform Ressource Locator), et le langage HTML.

Le protocole HTTP

Le protocole HTTP permet de transférer des fichiers (essentiellement au format HTML) localisés grâce à son URL entre un navigateur (le client) et un serveur Web. La version 1.0 du

4. Directeur du Consortium W3C depuis 1994

protocole (la plus utilisée) permet également de transférer des messages avec des en-têtes décrivant le contenu du message en utilisant un codage de type MIME.

La communication entre le navigateur et le serveur se fait en deux temps : le navigateur effectue une requête HTTP. Le serveur traite la requête puis envoie une réponse HTTP.

Une requête HTTP est un ensemble de lignes envoyées au serveur par le navigateur. Elle comprend plusieurs champs dont :

- La méthode qui doit être appliquée ;
- L'URL du serveur ;
- La version du protocole utilisé par le client (généralement HTTP/1.0) ;

D'autres informations sont envoyées au serveur tel que le nom du navigateur et du système d'exploitation du client. Un exemple d'un requête HTTP :

```
GET http://www.lirmm.fr/xml/fr/lirmm.html HTTP/1.0
```

Dans l'exemple au dessus, on demande au serveur HTTP du lirmm, la page Web qui se situe dans `/xml/fr/` et qui porte le nom de `lirmm.html`. Il aurait été possible de demander que l'entête de l'URL en utilisant la méthode HEAD, ou envoyer des données a un programme grâce a POST, envoyer des données a une url avec PUT et enfin effacer une ressource en utilisant la méthode DELETE.

La réponse HTTP du serveur est constituée d'un ensemble de ligne contenant la version du protocole utilisée et l'état du traitement de la requête à l'aide d'un code et d'un texte explicatif. Voici donc un exemple de réponse HTTP :

```
GET http://www.lirmm.fr/xml/fr/lirmm.html –>200 OK
Connection: close
Date: Wed, 02 Nov 2005 12:52:40 GMT
Server: Apache-Coyote/1.1
Content-Length: 9032
Content-Type: text/html
Content-Type: text/html; charset=UTF-8
Last-Modified: Wed, 02 Nov 2005 06:04:01 GMT
Client-Date: Wed, 02 Nov 2005 12:52:40 GMT
Client-Response-Num: 1
Link: <../../logolirmm.ico>; rel="SHORTCUT ICON"
Title: Accueil
X-Cocoon-Version: 2.1.1
X-Meta-Keywords:
```

Ces lignes signifie que la ressource (la page Web `lirmm.html`) existe bien sur le serveur (200 ok) et qu'il va la transmettre. En effet après ces lignes la page Web est transférée au client. Le serveur envoie d'autres informations : le titre de la page Web, la date de la dernière modification, etc. Après l'envoi du contenu de la page Web, le serveur ferme la connection.

Langage HTML

Le HTML [5] («HyperText Mark-Up Language») est un langage permettant d'indiquer la façon dont doit être présente le document et les liens qu'il établir avec d'autres documents grâce aux balises de formatage. Il permet surtout la lecture de documents sur Internet à partir de machines différentes, grâce au protocole HTTP, permettant d'accéder via le réseau à des documents repérés par des adresses uniques : les URLs. On appelle World Wide Web (noté WWW) ou tout simplement Web (mot anglais signifiant toile) la «toile virtuelle» formée par les différents documents (appelés «pages Web») liés entre-eux par des hyperliens (liens hypertextes).

Les pages Web sont généralement organisées autour d'une page d'accueil, jouant un point central dans la navigation à l'aide des liens hypertextes. Cet ensemble cohérent de pages Web liées par des liens hypertextes et articulées autour d'une page d'accueil commune est appelé : site Web.

Le Web est ainsi une énorme archive composée d'un grand nombre de sites Web proposant des pages Web pouvant contenir du texte mis en forme, des images, des sons, des vidéo, etc. Il est composée de pages Web stockées sur des serveurs Web, c'est-à-dire des machines connectées à Internet en permanence et chargées de fournir les pages Web demandées. Chacune des pages Web, et plus généralement toute ressource en ligne (image, vidéo, musique, animation, etc.), est repérée par une adresse unique (URL).

Moteurs de recherche

Aujourd'hui les moteurs de recherche jouent un rôle important dans le processus de recherche d'information sur le Web. Plusieurs problèmes sont rencontres lors de la conception et la réalisation d'un moteur de recherche. La première étape pour la construction d'un bon moteur de recherche est la construction d'une base de donnes de pages Web. En effet, les requêtes des utilisateurs sont traduites par des requêtes vers la base de données du moteur de recherche. La base de données est alimentée grâce à un programme qui pqrcours le Web à la recherche de nouvelles pages Web (*crawler*). Il est important pour un bon moteur de recherche d'avoir un grand nombre de pages Web. Mais cela, n'est pas suffisant, il est nécessaire d'avoir une base de données contenant que des «bonne» pages Web (pertinentes). La construction d'un bon crawler avec des stratégies qui permettent de trouver des pages Web de qualité en premier est un domaine de recherche en plein expansion. Très peu de travaux ont été réalisés dans se contexte.

Après la construction de la base de données, la deuxième étape consiste a ordonner les pages Web par pertinence pour une requête utilisateur. En effet, les moteur de recherche ordonnent leurs résultats par ordre décroissant de la pertinence. La recherche de bon algorithmes pour la classification de pages Web est également un problème de recherche ouvert : trouver des algorithme rapides et pertinents. Enfin, il est important pour un moteur de recherche de connaître et prévoir l'évolution du Web. Comme le Web est dynamique, il est important d'essayer de prévoir

5. HTML est un standard publié par le consortium international : World Wide Web Consortium (W3C).

sa structure et sa forme dans le future. La modélisation des réseaux d'interaction est un domaine de recherche très important permettant de générer des graphes de plus en plus grand et de simuler différents scénarios qui peuvent de produire sur les réseaux d'interactions.

Annexe B : Méthode de la puissance itérée

LA méthode de la puissance itérée est une méthode itérative de calcul de la valeur propre *dominante* (de plus grand module) d'une matrice et du vecteur propre correspondant. Elle est utilisée dans l'algorithme de HITS (voir section (4.3)) pour l'extraction des autorités et des hubs.

Itération d'un vecteur

Multiplication itérée d'un vecteur arbitraire par la matrice A

Principe de la méthode

Définition 7 (Valeurs propres, vecteurs propres)
Un scalaire λ est une valeur propre de la matrice A si et seulement s'il existe :

$$AX = \lambda X \qquad X \neq 0$$

X est appelé vecteur propre de A de valeur propre λ. Le scalaire λ est appelé valeur propre associée à X.

Théorème 1
Soit A une matrice $n \times n$

1. *λ est une valeur propre de A si et seulement si le $det(\lambda E - A) = 0$ où E est la matrice unitaire (toutes les entrées sont à 1) et $det(A)$ représente le déterminant de la matrice A.*
2. *Si λ est une valeur propre de A, alors toute solution $X \neq 0$ de $(\lambda E - A)X = 0$ est un vecteur propre associé.*

Soit A une matrice carrée $n \times n$ et X_i $(i = 1, \ldots, n)$ ses vecteurs propres. Si les X_i sont linéairement indépendants, c'est-à-dire si le déterminant $||X_1 X_2 \ldots X_n||$ n'est pas nul, un vecteur arbitraire V_1 peut être développé suivant les X_i de la manière suivante :

$$V_1 = a_1 X_1 + a_2 X_2 + \ldots + a_n X_n. \tag{1}$$

Si on multiplie m fois l'expression (1) par A, on obtient

$$A^m V_1 = a_1 A^m X_1 + a_2 A^m X_2 + \ldots + a_n A^m X_n. \tag{2}$$

Sachant que $AX_i = \lambda_i X_i$,

$$A^m V_1 = a_1 \lambda_1^m X_1 + a_2 \lambda_2^m X_2 + \ldots + a_n \lambda_n^m X_n. \tag{3}$$

Théorème 2
Si $A = (m_{i,j})_{i,j=1,\ldots,n}$ est une matrice symétrique (telle que $m_{i,j} = m_{j,i}$,$\forall i,j = 1,\ldots,n$) alors les n valeurs propres de A sont toutes réelles.

Si les λ_i, sont réels et si $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$ quand $m \to \infty$ alors :

$$A^m V_1 \to a_1 \lambda_1^m X_1, \tag{4}$$

si on pose

$$V_{m+1} = A^m V_1, \tag{5}$$

on voit que l'on a, d'après l'expression (4),

$$\lambda_1 = \lim_{m \to \infty} \left(\frac{V_{m+1}}{V_m} \right), \tag{6}$$

le rapport qui figure dans l'expression (6) pouvant être calculé avec l'une quelconque des composantes de V_{m+1} et la composante correspondante de V_m.

Toujours d'après l'expression (4), V_{m+1} converge vers le vecteur propre X_1, à un coefficient près $a_1 \lambda_1^m$ qui est sans importance puisque les vecteurs propres ne sont définis qu'à un coefficient près; donc

$$V_{m+1} \to X_1 \qquad m \to \infty. \tag{7}$$

Exemple

$$\begin{array}{ccccccccc} & & & & \lambda_1 & = & 1 & \lambda_2 & = & 9 \\ A & = & \begin{vmatrix} 5 & 4 \\ 4 & 5 \end{vmatrix} & X_1 & = & \begin{vmatrix} 1 \\ -1 \end{vmatrix} & X_2 & = & \begin{vmatrix} 1 \\ 1 \end{vmatrix} \end{array} \tag{8}$$

Les valeurs successives de V_m sont

1	5	41	365	3281	29525	265721
0	4	40	364	3280	29524	265720
V_1	V_2	V_3	V_4	V_5	V_6	V_7

(9)

Les valeurs des rapports de deux composantes successives sont

$$\lambda \rightarrow \left| \begin{array}{c|c|c|c|c|c} 5 & 8{,}2 & 8{,}9 & 8{,}989 & 8{,}998 & 8{,}999 \\ \infty & 10 & 9{,}1 & 9{,}011 & 9{,}001 & 9{,}000 \end{array} \right| \tag{10}$$

chacune des deux suites tend vers la valeur 9 qui est la valeur propre de plus grand module.

Remarque 2
Le raisonnement précédent suppose toutefois que le vecteur arbitraire V_1 possède une composante non nulle suivant X_1, c'est-à-dire que a_1 est différent de zéro. Si l'on avait rigoureusement $a_1 = 0$ on obtiendrait λ_2 et X_2.

Le processus itératif Quand on n'a aucune indication *a priori* sur X_1, on part de $V_1 = (1{,}0{,}0{,}\ldots{,}0)$, comme nous l'avons fait dans l'exemple précédent (voir 9).

- La convergence est d'autant plus rapide que le quotient $\frac{|\lambda_2|}{|\lambda_1|}$ est petit;
- Si $|\lambda_2| \approx |\lambda_1|$ la méthode est très lente.
- Si A est réelle et symétrique, la convergence du rapport $\frac{V_{m+1}}{V_m}$ vers λ_1 est quadratique par rapport a $\left|\frac{\lambda_2}{\lambda_1}\right|$.

Pour calculer la suite $A^m V_1$ on ne calcule pas la suite des puissances de A, mais plutôt on passe d'un vecteur itéré au suivant comme suit :

$$V_{p+1} = AV_p \tag{11}$$

Le calcul de $V_2{,}V_3{,}\ldots{,}V_m$ nécessite mn^2 *multiplications*, alors qu'autrement il en faudrait n fois plus (voir l'algorithme (6)).

Définition 8
On appelle norme sur E une application de E dans $\mathbb{R}^+$:

$$x \rightarrow ||x||$$

qui possède les propriétés suivantes :

- $||x|| = 0 \leftrightarrow x = 0$
- $||\alpha x|| = |\alpha|||x|| \qquad \forall \alpha \in \mathbb{R}$
- $||x + y|| \leqslant ||x|| + ||y||$

Définition 9
On appelle un espace euclidien un espace vectoriel muni d'un produit scalaire qui permet de définir une norme par la relation :

$$||x||^2 = \langle x{,}x \rangle$$

Algorithme 6: Algorithme de la puissance itérée

Données : A : La matrice
$V_1 \in \mathbb{R}^n$: un vecteur quelconque
Résultat : $(V_m \to X_1, \rho_m \to |\lambda_1|$
Soit $m = 2$, $\epsilon \in \mathbb{R}$, $\epsilon << 1$
début
  tant que $\frac{|\mu_m - \mu_{m-1}|}{|\mu_{m-1}|} > \epsilon$ **faire**
    $V_m = AV_{m-1}$ (itération)
    $\mu_m = \|V_m\|$ (norme euclidienne)
    $V_m = V_m/\mu_m$ (normalisation)
    $\rho_m = \frac{V_m^i}{V_{m-1}^i}$ (i^{me} entrée du vecteur, i quelconque)
    $m = m + 1$
  fin
  retourner $(V_m \to X_1, \rho_m \to |\lambda_1|)$
fin

Définition 10 (Norme euclidienne sur $\mathbb{R}^n$)
Dans l'espace vectoriel $\mathbb{R}^n$, on défi nit la norme euclidienne d'un vecteur x par :

$$||x||^2 = \langle x,x \rangle = \sum_{i=1}^{n} x_i^2$$

Le produit scalaire associée étant :

$$\langle x,y \rangle = \sum_{i=1}^{n} x_i y_i$$

Sur l'espace vectoriel $\mathbb{R}^n$, d'autres normes que la norme euclidienne peuvent être défi nis. Par exemple, l'application suivante défi nit une norme :

$$x \to ||x||_p = \left(\sum_{i=1}^{n} x_i^p \right)^{\frac{1}{p}} \qquad \forall p \geqslant 1$$

les cas les plus usités étant :

$$||x||_1 = \sum_{i=1}^{n} |x_i| \qquad ||x||_\infty = \max_{i=1,\ldots,n} |x_i|$$

respectivement appelée norme 1 et norme infi nie.

Au lieu de laisser croître l'ensemble des composantes de V_m comme dans l'exemple précédent (voir tableau 9), on peut les normaliser; cela peut se faire de plusieurs manières; on peut diviser toutes les composantes de V_p par $\sqrt{V_p^t V_p}$ si la matrice est symétrique.

Annexe C : Une implémentation expérimentale d'un crawler

Le Web a évolué très rapidement depuis les années 90, passant de quelques millions de pages à plusieurs milliards. Depuis, les moteurs de recherche sont très sollicités pour la recherche d'information. Le succès d'un moteur de recherche dépend de la variété et de la pertinence des pages Web stockées (la probabilité de succès d'une recherche augmente évidemment avec la taille de la base de données). Pour constituer cette base, on utilise un *crawler*. Ce dernier parcours le Web en suivant les liens hypertextes et stocke les pages Web téléchargées dans une grande base de données où elles seront indexées pour répondre efficacement aux requêtes des utilisateurs.

Plusieurs travaux de recherches ont été menés dans plusieurs domaines liés à la technologie des moteurs de recherches, notamment les stratégies de parcours du Web, le stockage de la base de données, l'indexage, les techniques de tri, les techniques de compressions et l'analyse de la structure hypertexte du Web.

Étudier les stratégies de crawl est donc intéressant (ce que nous avons fait au chapitre 3). Mais, afin de disposer de données, nous avons été amenés à étudier et à implémenter un crawler réel.

Introduction

Les crawlers, très utilisés à nos jours, essayent de parcourir une grande partie du Web. Par exemple, le moteur de recherche Google indexe plus de 8 milliards de pages Web. Les crawls peuvent être spécialisés pour certains types d'informations tels que les adresses mails, images, vidéos, etc. Enfin, on peut trouver des crawlers personnalisés qui analysent des pages Web pour l'intérêt d'un utilisateur particulier. L'objectif du crawler personnalisé (recherche assistée) est d'anticiper le comportement de l'utilisateur en téléchargeant à priori des pages Web que l'utilisateur risque de vouloir visiter. Cela permet d'accélérer la navigation ou le téléchargement par modem.

Étant donné que le Web est gigantesque et dynamique, la conception d'un *bon* crawler est un vrai challenge. Plusieurs études montrent que la taille de la partie visible du Web est de plusieurs

milliards de pages Web[1] et que la taille moyenne d'une page Web est entre 5 et 10 kilo-octets. Une grande capacité de stockage (plusieurs téra-octets) est nécessaire pour stocker un crawl conséquent.

Lors de la conception d'un crawl, il est important de tenir compte :

1. **du choix des pages Web à visiter :** les crawlers ne sont pas en mesure de visiter tout le Web [67, 82]. Et c'est même impossible (Dynamicité,etc). Par conséquent, il est important de visiter les pages Web *pertinentes* en premier. Dans ce cas, la collection de pages Web est plus significative. La pertinence est généralement calculée en utilisant des algorithmes de tri simples tel que le degré sortant ou le degré entrant (très difficile à obtenir) ou en utilisant des algorithme plus compliqués tel que le PageRank ou HITS. Certains moteurs de recherche demandent même aux utilisateurs de soumettre des pages Web.
2. **du choix des pages Web à revisiter :** Il est nécessaire de mettre à jour régulièrement la collection de pages Web et donc de revisiter les pages Web de la collection. Les pages Web ont un taux de modifications très variable [32]. Par conséquent, le crawler doit décider quelles sont les pages Web à revisiter et celles qui peuvent attendre un délai plus long. Les pages à revisiter sont bien sûr les plus susceptibles de modification.
3. **du coût du crawling :** Quant un crawler collecte des pages du Web, il consomme des ressources appartenant à d'autres organisations. Un exemple, lorsqu'un crawler télécharge une page p du site S, le serveur Web du site S doit rechercher la page p dans son système de fichiers, cette opération nécessite du temps CPU et de l'accès à un support de stockage (disque dur, cd,etc.). Après la phase de recherche, la page p doit être transférée au crawler à travers le réseaux Internet qui est une ressource communes à toutes les organisations. Des crawler mal conçu peuvent bloquer des serveur Web (voir endommager) suite à de multiples téléchargement du même site Web. Pour palier à ce problème, généralement, les administrateurs de serveurs Web limitent l'accès aux crawlers, voir même leur interdisent complètement le parcours de leur site Web.
4. **de paralléliser le processus de crawl :** Vu la grande taille du Web, les crawlers souvent s'exécutent sur plusieurs machines et téléchargent des pages Web en parallèle [32, 27]. Ce parallélisme est nécessaire si l'on souhaite télécharger un grand nombre de pages Web en temps raisonnable. Il est nécessaire que les crawlers coopèrent afin de ne pas télécharger les mêmes pages en même temps. Cette coopération nécessite une communication (des envoies de messages) entre les crawlers limitant ainsi le nombre de crawlers en parallèle (plus le nombre de crawlers est grand et plus le temps et le volume de communication est grand).

Un crawler est un programme qui télécharge des pages Web pour un moteur de recherche. Plus précisément, un crawler part d'un ensemble d'URLs S_0. Ces S_0 premières URLs sont placées en queue de la liste des URLs à visiter (télécharger). À partir de cette liste, le crawler choisit

1. http://www.oclc.org/research/projects/archive/wcp/default.htm

une URL de la liste (suivant une stratégie de parcours), télécharge la page, extrait toutes les URLs contenues dans la page et enfin place les nouvelles URLs en queue de liste des URLs à visiter. Ce processus est réitéré tant que la liste n'est pas vide. Les pages Web téléchargées sont analysées (afin d'extraire les urls et les mots) et indexées (construction d'un dictionnaire de mots). Une page Web indexée signifie qu'à partir d'un mot du dictionnaire, il est possible de retrouver toutes les pages Web téléchargées contenant ce mot.

État de l'art

Le premier crawler a été conçu par Matthew Gray's Wanderer en 1993 [51]. Dans les deux premières conférences sur le Web (World Wide Web conference), plusieurs travaux sur le crawling (parcours du Web) ont été proposés [41, 76, 92]. Mais, on note qu'à cette époque la taille du Web était beaucoup plus petite qu'aujourd'hui.

Dans la littérature, très peu d'information existe sur l'architecture des crawlers. Il y a notemment deux crawlers industriels (partiellement) documentés : *Mercator* conçu par Najork et Heydon [54, 81], utilisé par le moteur de recherche *Altavista*. Et *WebBase*, le crawler du moteur de recherche *Google* où une premier version du crawl est donnée dans [26]. On peut également trouver quelques détails du crawler utilisé dans *Internet Archive* [29].

WebBase

Développé à l'université de Stanford, le Crawler *WebBase*, utilisé par le moteur de recherche Google, est constitué de cinq processus. Le processus *URL Server* lit les URLs et les transmet aux différents processus crawlers. Chaque processus *Crawler* s'exécute sur une machine, utilise des entrées/sorties asynchrone et peut se connecter à 300 serveurs Web en parallèle. Le processus Crawler transmet les pages Web téléchargées au processus *Store Server*. Ce dernier, compresse les pages Web et les stocke sur disque. Les pages sont alors lues par le processus *Indexer* qui extrait les liens des pages HTML et les stocke (les liens) dans différents fichiers. Le processus *URL Resolver* lit les fichiers contenant les liens hypertextes, les transforme en liens absolus [2]; et les stocke dans des fichiers afin que le processus URL Server les récupère. En général, trois à quatre crawlers sont utilisés, le programme nécessite en tout entre quatre et huit machines. Le moteur de recherche Google utilise une base de données répartie par conséquent, il utilise plusieurs «WebBase» répartie sur un grand nombre de machines.

Malgré la commercialisation du moteur de recherche Google, le projet WebBase de l'université de Stanford a permis l'implémentation d'un crawler distribué très performant capable de télécharger 50 à 100 documents par seconde [55]. Dans [34], des modèles de crawlers incrémentales sont proposés qui tiennent compte de la fréquence de mises à jours des documents. Un

2. par exemple, si dans la page Web `http://www.lirmm.fr/~bennouas/`, on trouve un lien hypertexte `<ahref="recherche.html">`, ce dernier est convertie en `http://www.lirmm.fr/~bennouas/recherche.html`

crawler incrémental est un crawler qui permet de mixer le processus de crawl (à la découverte de nouvelles pages Web) et le processus de rafraîchissement des pages Web.

Internet Archive

Le crawler *Internet archive* utilise également plusieurs machines pour parcourir le Web [29, 1]. 64 sites Web sont attribués à chaque processus *crawler*. Un site Web est assigné à un seul processus *crawler*. Le processus *crawl* parcours les 64 sites Web en parallèle et attribut pour chaque site Web S_i une liste L_i où seront stocké les URLs à visiter du site Web S_i. Le crawler choisit une url de la liste L_i, la télécharge, extrait les liens hypertextes contenu dans la page et insère les url du site S_i en queue de la liste L_i. Si le crawler rencontre une url d'un site qui ne lui est pas attribué, cette dernière est stockée sur disque où un autre processus les récupère pour les redistribuer ultérieurement sur les différents crawlers.

Mercator

Le crawler *Mercator* développé par Najork et Heydon [54, 81] en 2001, est un crawl très performant et distribué sur plusieurs machines. Il manipule une structure de données adaptées à un grand nombre de pages Web. Mercator à une capacité de téléchargement de 50 millions de pages Web par jour. Bien sûr, il ne cause que très peu de désagrément aux serveurs Web visités en respectant les consignes des administrateurs des serveurs Web, ces derniers pouvant restreindre l'accès à leurs sites Web ou encore l'interdire complètement. Il est incrémental, c'est-à-dire, permet la mise à jour des pages Web déjà téléchargées tout en recherchant de nouvelles pages Web. Enfin, le crawler est extensible et entièrement portable (écrit en Java).

Mercator est constitué principalement de cinq composants : un composant pour stocker la liste des urls à télécharger, un composant qui permet de déterminer l'adresse IP d'un site Web, un composant pour le téléchargement des pages HTML en utilisant le protocole HTTP et enfin un composant qui détermine si une url à déjà été rencontrée.

Autres crawlers

Dans [39], une description du crawler WebFountain est donnée. L'architecture de ce dernier est proche de celle de Mercator.

Enfin, Cho et Garcia-Molina [33] proposent une architecture d'un crawler parallélisé avec un taux de chevauchement très faible entre les processus et un taux de téléchargement très élevé. En effet, lorsque plusieurs crawlers parcours le Web en parallèle, il est fréquent qu'une même portion du Web soit parcourue par plusieurs crawlers, ce phénomène est appelé chevauchement. Pour une performance optimale, il est nécessaire de s'assurer que deux crawlers ne parcourent pas la même portion du Web.

Cho and Garcia-Molina tiennent compte donc du *Chevauchement*, mais aussi de la *bonne qualité* des pages Web téléchargées, cette dernière est difficile à obtenir avec un crawler parallé-

lisé. En effet, chaque processus ne peut avoir une vu globale du *sous Web* collecté par l'ensemble des crawlers (processus). Chaque processus agit selon sa propre *vue du Web* (les pages Web téléchargées par le processus lui même). Les auteurs proposent un crawler parallélisé avec un minimum de *communication entre processus*. Les processus doivent communiquer entre eux (coopérer) afin d'éviter le chevauchement et pour s'échanger les URLs à télécharger. Plus le nombre de processus est grand et plus de temps et de message sont nécessaires pour la coopération.

Nous rappelons que l'architecture des crawlers citées ici n'est pas donnée avec beaucoup de détails par leurs auteurs. Cela est dû principalement à des raisons purement commercial.

Pour concevoir un crawler performant, il est nécessaire d'utiliser des méthodes de compressions pour gérer la grande quantité d'information. En effet, sans aucune compression, la taille de la base de données devient rapidement grande et donc le traitement des requêtes assez lent. Plusieurs travaux sur la compression du graphe du Web et la représentation des pages HTML sont données dans [94, 7, 102, 6, 20, 103].

La représentation et compression du graphe du graphe du Web n'entre pas dans le cadre de cette thèse mais c'est un domaine qui a beaucoup d'intérêt.

Objectifs de notre implémentation

Notre objectif est double. Le premier est de s'initier dans le domaine du crawling qui est à nos jour un domaine très fermé du fait de l'intérêt commercial. A ce jour, très peu de recherche universitaires sont menées dans ce domaine. Le second est de se procurer une image du Web (une partie du Web) assez récente afin de l'utiliser dans nos expérimentations pour la détection de communautés dans notre modèle gravitationnel du Web (voir le chapitre 5). La majorité des bases de données existantes datent de plusieurs années et ne correspondent plus à l'image actuelle du Web.

Architecture du crawler

Nous avons utilisé l'interpréteur *Python* pour la réalisation de ce crawler avec une connexion à une base de données PostgreSQL. Le module de crawl est portable et extensible. L'interpréteur Python est est un langage orienté objet très utilisé dans la programmation Web.

Le crawler est composé de cinq processus :

Ordonnanceur : Ce processus contient la liste des urls à télécharger. Cette liste est ordonnée selon la stratégie de parcours adoptée. L'ordonnanceur dépile les urls de la liste et les transfert au téléchargeur.

Téléchargeur : Reçoit les urls de l'ordonnanceur, télécharge la page Web et transmet son contenu à l'analyseur.

Analyseur : Analyse la page Web, extrait les liens hypertextes et les transmet au résolveur.

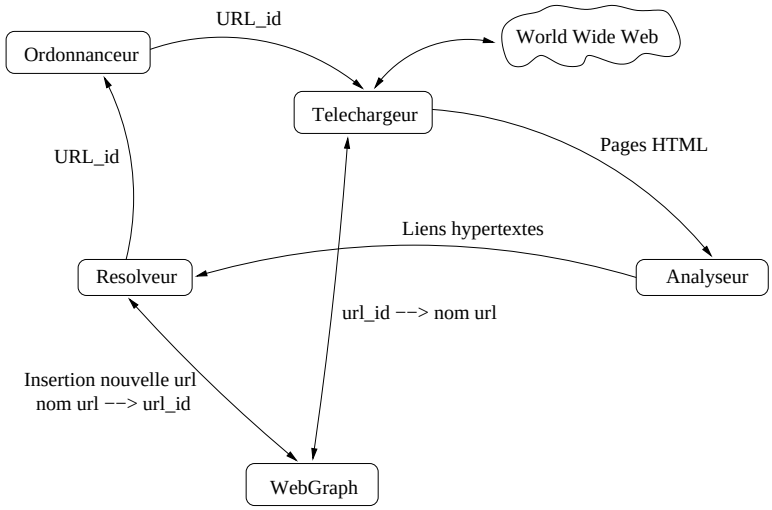


FIG. 23 – *Schema du crawler.*

Résolveur : Ce dernier transforme les liens hypertextes en liens absolus, vérifie l'existence des pages Web sur les serveurs et les transmet à l'ordonnanceur.

WebDataBase : C'est une base de données contenant trois tables : domaines, urls et liens. Ces tables contiennent respectivement des informations concernant les nom de domaines, les urls et les liens hypertextes entre pages Web rencontrés lors du parcours. Cette base de données est utilisé par le téléchargeur et le résolveur.

Dans tous le processus, seuls les numéros des urls sont manipulés, lors du téléchargement de la page Web, le téléchargeur accède à la base de données pour avoir l'url de la page Web à télécharger.

La liste des urls est ordonnée en utilisant plusieurs stratégie de parcours, Exemple, le parcours en largeur (BFS), le parcours en profondeur (DFS), le parcours aléatoire (un mélange de BFS et DFS). Il existe d'autre parcours plus complexes, par exemple, le parcours selon la valeur du PageRank où la liste des URLs à visiter est ordonnée dans l'ordre décroissant de leur PageRank (voir chapitre 4).

Résultats

Nous avons tenté deux expérimentations en mars et mai 2004. Le crawler c'est exécuté pendant 1 semaine lors des deux expérimentations. Le nombre de pages Web téléchargées et de l'ordre de 763876 pages Web et 4527019 liens hypertextes. Nous avons utilisé le parcours en largeur évitant ainsi de trop s'approfondir dans un site.

Algorithme 7: Algorithme de crawl

Données : L_d : La liste d'URLs de départ
début
1. Supprimer une URL de la liste des URLs à télécharger
2. Déterminer l'adresse IP du nom de domaine de l'URL
3. Télécharger la pages Web
4. Extraire tous les liens hypertextes de la page Web
début
pour chaque *lien de la page Web* **faire**
5. Si nécessaire, transformer le lien relatif en lien absolus
6. Si non encore rencontré, ajouter le lien dans la liste des URLs
fin
fin
7. Tant que la liste des URL non vide, aller à 1.
fin

Types et origines des pages Web : Les pages Web téléchargées proviennent de différents sites Web aussi bien commerciaux que gouvernementaux et appartiennent à des sites géographiquement distants (voir tableau 5.6).

Tableau 5.6 : Types et origines des pages Web crawlées.

prefixe	Nombre de pages	prefixe	Nombre de pages
.com	228553	.ca	198457
.org	80601	.fr	111655
.net	21811	.be	21813
.edu	20390	.de	20002
.gov	16686	.dz	2306
sous-total	368041	sous-total	354233

Distribution des degrés : La distribution des degrés entrants et sortants suivent une loi en puissance avec respectivement les paramètres $\lambda^- = 2.1$ et $\lambda^+ = 2.5$ (voir figure 24).

Distance moyenne et diamètre : La distance moyenne est de 18.293 et le diamètre de 503. On remarque que la distribution des distances suit une loi de poisson centré autour de la distance moyenne.

Composante géante fortement connexe : Le crawl contient bien une composante géante fortement connexe contenant plus de 30% des pages Web. De plus, la figure 26 montre que la

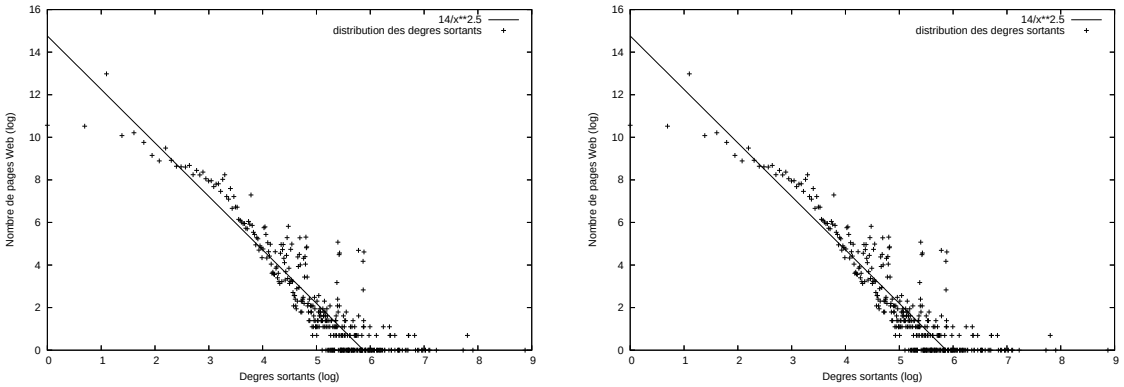


FIG. 24 – *Distribution des degrés du crawl du Web.*

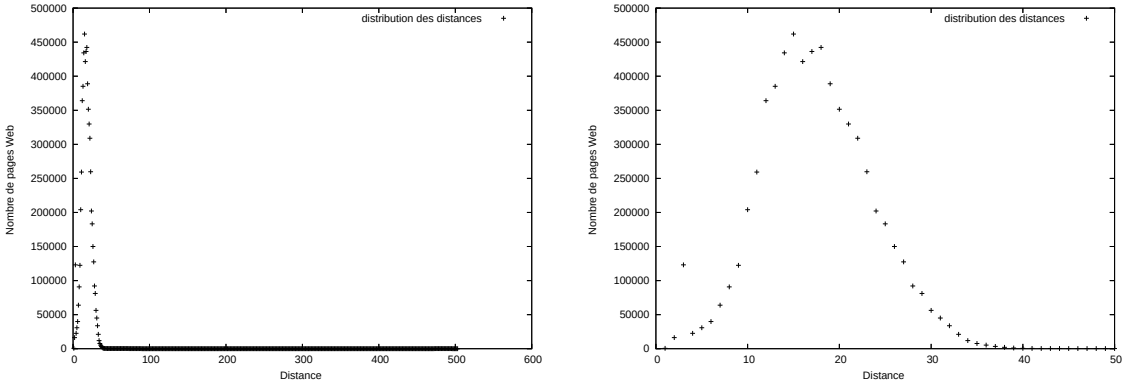


FIG. 25 – *Distribution des distances.*

distribution du nombre de composantes fortement connexe en fonction de leurs tailles suit une loi de puissance.

Coefficient de regroupement : Pour calculer le coefficient de regroupement, nous supprimons l'orientation des liens. Le nombre d'arêtes dans le graphe devient de $E = 2087268$ et le degré moyen de $d^+ = 5.46$. Le coefficient de regroupement du crawl est de $CC = 0.197858$.

On remarque que cette valeur est plus grande que celle d'un graphe aléatoire d'Ardös et Rényi de même taille.

En résumé, nous avons implémenté un crawler distribué avec des stratégies de parcours simples. Il nécessite l'utilisation d'une base de données pour stocker les urls et les liens entre les urls. Cette base de données peut être étendue pour stocker le contenue des pages Web.

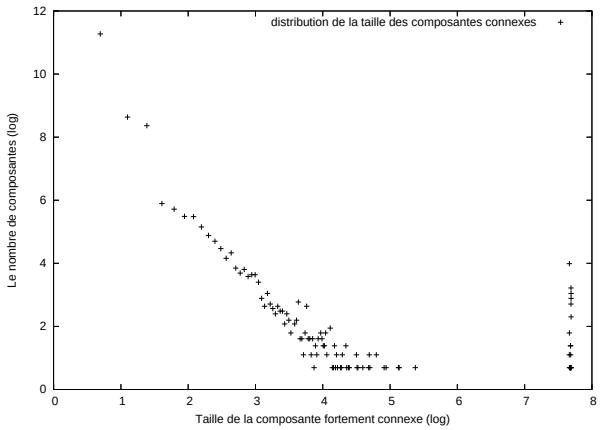


FIG. 26 – *Distribution de la taille des composantes connexes du crawl.*

Pour augmenter le nombre de pages Web crawlées, il est possible de lancer plusieurs crawls en parallèle a condition de s'assurer que les crawlers ne vont pas parcourir la même portion du Web (le recouvrement n'est pas géré pour le moment).

En implémentant ce crawler, nous avons fait face a différents problèmes liés à l'accès au ressources sur Internet, au traitement du contenue des pages Web (parser), au traitement des urls, etc.

Les crawls obtenus ont les mêmes propriétés que les autres crawls du Web à savoir une distribution des degrés en loi de puissance, une coefficient de regroupement fort, une distance moyenne faible et une composante géante fortement connexe (voir chapitre 1).

Enfin, il permet d'obtenir une image récente du Web d'une grande taille (plus de 1 million) après seulement quelques jours d'utilisation. L'intérêt de ce crawl est de pouvoir l'utiliser pour tester différentes stratégies de crawl sur le Web. Le crawler peut être adaptés facilement à de nouvelles stratégies de parcours.

Liste des Algorithmes

Table des figures

Bibliographie

[1] The internet archive. http://www.archive.org/. Cité page(s) 122

[2] *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, 1979. Cité page(s) 82

[3] *Social Network Analysis: A Handbook*. London, 2nd edition, 2000. Cité page(s) 82

[4] Serge Abiteboul, Mihai Preda, and Gregory Cobena. Adaptive on-line page importance computation. In *Proceedings of the twelfth international conference on World Wide Web*, pages 280–290. ACM Press, 2003. Cité page(s) 49, 67

[5] Lada A. Adamic. The small world web. In *Proceedings of the Third European Conference on Research and Advanced Technology for Digital Libraries*, pages 443–452. Springer-Verlag, 1999. Cité page(s) 17, 21, 22, 81

[6] M. Adler and M. Mitzenmacher. Towards compressing web graphs. Technical report, voir institution, 2000. Cité page(s) 123

[7] Micah Adler and Michael Mitzenmacher. Towards compressing web graphs. In *Proceedings of the Data Compression Conference (DCC '01)*, page 203. IEEE Computer Society, 2001. Cité page(s) 123

[8] William Aiello, Fan Chung, and Linyuan Lu. A random graph model for massive graphs. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 171–180. ACM Press, 2000. Cité page(s) 18, 20, 21, 30

[9] Réka Albert, Hawoong Jeong, and Albert-László Barabasi. Diameter of the world-wide web. *Nature*, 401:130–131, September 1999. Cité page(s) 17, 20

[10] Mark S. Aldenderfer and Roger K. Blashfield. *Cluster Analysis*. Sage Publications, 1984. Cité page(s) 84, 101

[11] Jose Ignacio Alvarez-Hamelin, Luca Dall'Asta, Alain Barrat, and Alessandro Vespignani. k-core decomposition: a tool for the visualization of large scale networks, 2005. Cité page(s) 41, 52, 53, 54, 55

[12] L. Amaral, A. Scala, M. Barthelemy, and H. Stanley. Classes of small-world networks. *Proceedings of National Academy of Science*, 97(21):11149–11152, 2000. Cité page(s) 17, 19, 84

[13] B. Amento, L. Terveen, and W. Hill. Does authority mean quality? predicting expert quality ratings of web documents. In *Proceedings of the 23rd Int'l ACM SIGIR Conference*, pages 296–303, Press, New York, 2000. Cité page(s) 74

[14] A. Arasu, J. Novak, A. Tomkins, and J. Tomlin. Pagerank computation and the structure of the web: Experiments and algorithms, 2001. Cité page(s) 72

[15] A.-L. Barabasi, H. Jeong, Z. Néda, E. Ravasz, and A. Schubert T. Vicsek. Evolution of the social network of scientific collaborations. In *Physica A 311*, pages 590–614, 2002. Cité page(s) 18, 20, 21, 22

[16] Albert-László Barabasi and Réka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999. Cité page(s) 8, 17, 20, 31, 37

[17] V Batagelj and M Zaversnik. Generalized cores, 2002. Cité page(s) 52

[18] T. Bennouas, M. Bouklit, and F. de Montgolfier. Un modèle gravitationnel du web. In *Algotel, 5émes Rencontres Francophones sur les aspects Algorithmiques des Télécommunications*, 2003. Cité page(s) 10, 80

[19] T. Bennouas, M. Bouklit, and F. de Montgolfier. Un modèle gravitationnel du web. In *Premières journées francophones de la Toile*, 2003. Cité page(s) 10, 80

[20] Krishna Bharat, Andrei Broder, Monika Henzinger, Puneet Kumar, and Suresh Venkatasubramanian. The connectivity server: fast access to linkage information on the web. In *Proceedings of the seventh international conference on World Wide Web 7*, pages 469–477. Elsevier Science Publishers B. V., 1998. Cité page(s) 123

[21] Béla Bollobás, Oliver Riordan, Joel Spencer, and Gábor E. Tusnády. The degree sequence of a scale-free random graph process. *Random Structures and Algorithms*, 18(3):279–290, May 2001. Cité page(s) 37, 38

[22] Anthony Bonato. A survey of models of the web graph. In *The Proceedings of Combinatorial and Algorithmic Aspects of Networking*, August 2004. Cité page(s) 31

[23] Y. Boufkhad and L. Viennot. The observable web. Technical Report RR-4790, INRIA, 2003. Cité page(s) 17

[24] Mohamed Bouklit and Alain Jean-Marie. Une analyse de pagerank, une mesure de popularité des pages web. In *Proceedings ALGOTEL'02*, Mèze, France, May 2002. Cité page(s) 69

[25] Sergey Brin. Extracting patterns and relations from the world wide web. In *WebDB '98: Selected papers from the International Workshop on The World Wide Web and Databases*, pages 172–183, London, UK, 1999. Springer-Verlag. Cité page(s) 69

[26] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, pages 107–117. Elsevier Science Publishers B. V., 1998. Cité page(s) 66, 67, 68, 121

[27] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998. Cité page(s) 120

[28] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, pages 309–320. North-Holland Publishing Co., 2000. Cité page(s) 17, 20, 21, 22, 24, 46, 49, 50

[29] Mike Burner. Crawling towards eternity: Building an archive of the world wide web. http://www.webtechniques.com/archives/1997/05/burner/, May 1997. Cité page(s) 121, 122

[30] CAIDA. The cooperative association for internet data analysis. http://www.caida.org/home/. Cité page(s) 56

[31] J. Carriere and R. Kazman. Webquery: Searching and visualizing the web through connectivity. In *Proceedings of the Sixth International World Wide Web Conference*, pages 701–711, Elsevier Science, New York, 1997. Cité page(s) 73

[32] Junghoo Cho and Hector Garcia-Molina. The evolution of the web and implications for an incremental crawler. In *Proceedings of the 26th International Conference on Very Large Data Bases*, pages 200–209. Morgan Kaufmann Publishers Inc., 2000. Cité page(s) 120

[33] Junghoo Cho and Hector Garcia-Molina. Parallel crawlers. In *Proceedings of the eleventh international conference on World Wide Web*, pages 124–135. ACM Press, 2002. Cité page(s) 122

[34] Junghoo Cho, Hector García-Molina, and Lawrence Page. Efficient crawling through URL ordering. *Computer Networks and ISDN Systems*, 30(1–7):161–172, 1998. Cité page(s) 121

[35] Fan Chung and Linyan Lu. Coupling on-line and off-line analyses for random power graphs. submitted. Cité page(s) 38

[36] Fan Chung and Linyuan Lu. The diameter of sparse random graphs. *Adv. Appl. Math.*, 26(4):257–279, 2001. Cité page(s) 29

[37] Colin Cooper and Alan Frieze. Random deletion in a scale free random graph process. to appear in Internet Mathematics.submitted. Cité page(s) 38

[38] Jeffrey Dean and Monika R. Henzinger. Finding related pages in the world wide web. In *Proceeding of the eighth international conference on World Wide Web*, pages 1467–1479. Elsevier North-Holland, Inc., 1999. Cité page(s) 77, 81

[39] Jenny Edwards, Kevin McCurley, and John Tomlin. An adaptive model for optimizing performance of an incremental web crawler. In *Proceedings of the tenth international conference on World Wide Web*, pages 106–113. ACM Press, 2001. Cité page(s) 122

[40] Kemal Efe, Vijay Raghavan, C. Henry Chu, Adrienne L. Broadwater, Levent Bolelli, and Seyda Ertekin. The shape of the Web and its implications for searching the Web, 31 – 6 2000. Cité page(s) 81

[41] David Eichmann. The rbse spider - balancing effective search against web load. In *Proceedings of the First International Conference on World Wide Web*, pages 113–120, 1994. Cité page(s) 121

[42] P. Erdös and A. Rényi. On random graphs. In *Publicationes of the Mathematicae*, volume 6, pages 290–297, 1959. Cité page(s) 8, 28, 29

[43] P. Erdös and A. Rényi. On the evolution of random graphs. In *Publications of the Mathematical Institute of Hungarian Academy of Sciences*, volume 5, pages 17–61, 1960. Cité page(s) 8, 28, 29

[44] P. Erdös and A. Rényi. On the strength of connectedness of random graphs. In *Acta Mathematica Scientia Hungary*, volume 12, pages 261–267, 1961. Cité page(s) 8, 28, 29

[45] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. *Computer Communication Review*, 29:251–262, 1999. Cité page(s) 16, 20, 21

[46] I. Ferrer, R. Cancho, and R. V. Solé. The small-world of human language. *Proceedings of the Royal Society of London Biol. Sc*, 268:2261–2265, 2001. Cité page(s) 19, 20, 21, 22

[47] Gary William Flake, Steve Lawrence, and C. Lee Giles. Efficient identification of web communities. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–160. ACM Press, 2000. Cité page(s) 81

[48] Bruno Gaume. Balades aléatoires dans les petits mondes lexicaux. *I3 : Information - Interaction - Intelligence*, 04(02), 2004. Cité page(s) 84

[49] David Gibson, Jon Kleinberg, and Prabhakar Raghavan. Inferring web communities from link topology. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems*, pages 225–234. ACM Press, 1998. Cité page(s) 80

[50] Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for internet map discovery. In *Proceedings of the 2000 IEEE INFOCOM Conference*, pages 1371–1380, Tel Aviv, Israel, March 2000. IEEE. Cité page(s) 16, 20, 21, 55

[51] Matthew Gray. Internet growth and statistics: Credits and background. Cité page(s) 121

[52] Jean-Loup Guillaume and Matthieu Latapy. Bipartite structure of all complex networks. *Information Processing Letters (IPL)*, 90:215–221, 15 June 2004. Cité page(s) 36

[53] S. W. Hawking. *A Brief History of Time*. Bantam, NY, 1988. Cité page(s) 86

[54] Allan Heydon and Marc Najork. Mercator: A scalable, extensible web crawler. *World Wide Web*, 2(4):219–229, 1999. Cité page(s) 121, 122

[55] Jun Hirai, Sriram Raghavan, Hector Garcia-Molina, and Andreas Paepcke. Webbase: A repository of web pages. In *Proceedings of the Ninth International Conference on World Wide Web*, pages 277–293, May 2000. Cité page(s) 121

[56] B. A. Huberman and L. A. Adamic. Growth dynamics of the world-wide-web. *Nature*, 401:131, 1999. Cité page(s) 17, 20

[57] H. Jeong, B. Tombor, R. Albert, Z.-N. Oltvai, and A.-L. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407(6804):651, Jun 2000. Cité page(s) 18

[58] B. W. Kernighan and S. Lin. A efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, (49):291–307, 1970. Cité page(s) 82, 84, 101

[59] Jon Kleinberg. The small-world phenomenon: an algorithm perspective. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 163–170. ACM Press, 2000. Cité page(s) 35

[60] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999. Cité page(s) 25, 66, 74, 75, 76, 77, 80

[61] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, page 57. IEEE Computer Society, 2000. Cité page(s) 32, 33

[62] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D. Sivakumar, Andrew Tompkins, and Eli Upfal. The web as a graph. In *Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–10. ACM Press, 2000. Cité page(s) 17, 20

[63] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Extracting large-scale knowledge bases from the web. In *Proceedings of the 25th International Conference on Very Large Data Bases*, pages 639–650. Morgan Kaufmann Publishers Inc., 1999. Cité page(s) 39

[64] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the web for emerging cyber-communities. In *Proceeding of the eighth international conference on World Wide Web*, pages 1481–1493. Elsevier North-Holland, Inc., 1999. Cité page(s) 25, 26, 80, 81, 132

[65] Matthieu Latapy and Pascal Pons. Computing communities in large networks using random walks. Cité page(s) 84, 85, 101

[66] S. Lawrence and C. L. Giles. Accessibility of information on the web. *Nature*, 400(6740):107–109, 1999. Cité page(s) 17

[67] Steve Lawrence and C. Lee Giles. Accessibility of information on the web. *Intelligence*, 11(1):32–39, 2000. Cité page(s) 120

[68] E. Lebhar and N. Schabanel. Close to optimal decentralized routing in long-range contact networks (contient la dimension d>1). *Invité au numéro spécial de Theoretical Computer Science pour ICALP'04*, 348(2-3), 2005. Cité page(s) 36

[69] S. LEVANT. http ://hipercom.inria.fr/soleil/, 2001. Cité page(s) 56, 57, 90, 132

[70] Mark Levene and Alexandra Poulovassilis. Web dynamics. *Software Focus*, 2(2):60–67, August 2001. Cité page(s) 29

[71] F. Liljeros, C. R. Edling, L. A. N. Amaral, H. E. Stanley, and Y. Aberg. The web of human sexual contacts. *Nature*, 411:907, 2001. Cité page(s) 18, 20, 21

[72] M. Marchiori and V. Latora. Harmony in the small-world. *Physica A*, (285):639–546, 2000. Cité page(s) 84

[73] F. MATHIEU and M. BOUKLIT. The effect of the back button in a random walk : application for pagerank. In *Alternate track papers & posters of the 13th international conference on World Wide Web*, pages 370–371. ACM Press, 2004. Cité page(s) 72

[74] F. Mathieu and L. Viennot. Local structure in the web. In *12-th international conference on the World Wide Web*, 2003. poster. Cité page(s) 58

[75] Fabien Mathieu. Graphes du web, mesures d'importance à la pagerank, 2004. Cité page(s) 24, 41, 50, 69, 72

[76] Olivier A. McBryan. Genvl and wwww: Tools for taming the web. In *Proceedings of the First International Conference on World Wide Web*, pages 79–90, 1994. Cité page(s) 121

[77] S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967. Cité page(s) 18, 21

[78] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. In *Random Graphs 93: Proceedings of the sixth international seminar on Random graphs and probabilistic methods in combinatorics and computer science*, pages 161–179, New York, NY, USA, 1995. John Wiley & Sons, Inc. Cité page(s) 30

[79] Michael Molloy and Bruce Reed. The size of the giant component of a random graph with a given degree sequence. In *Combinatorics, Probability and Computing 7*, pages 295–305, 1998. Cité page(s) 30

[80] Rajeev Motwani and Prabakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995. Cité page(s) 69

[81] Marc Najork and Allan Heydon. High-performance web crawling. pages 25–45, 2002. Cité page(s) 121, 122

[82] Marc Najork and Janet L. Wiener. Breadth-first crawling yields high-quality pages. In *Proceedings of the tenth international conference on World Wide Web*, pages 114–118. ACM Press, 2001. Cité page(s) 50, 120

[83] M. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003. Cité page(s) 31, 40, 61

[84] M. E. J. Newman. Scientific collaboration networks. i. network construction and fundamental results. In *Physical Review*, volume 64, page 016131, 2001. Cité page(s) 18

[85] M. E. J. Newman. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. In *Physical Review*, volume 64, page 016132, 2001. Cité page(s) 18

[86] M E J Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, 2004. Cité page(s) 40, 61, 85

[87] M E J Newman and M Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004. Cité page(s) 40, 61, 79, 85

[88] M. E. J. Newman, S. H. Strogatz, and D. Watts. Random graphs with arbitrary degree distribution and their applications. In *Physical Review*, volume 4, page 131, 1998. Cité page(s) 17, 18, 20, 21, 22

[89] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Computer Science Department, Stanford University, 1998. Cité page(s) 50, 66, 67, 68, 71, 72, 90

[90] Larbin Home Page. http://larbin.sourceforge.net/. Cité page(s) 90

[91] Gopal Pandurangan, Prabhakar Raghavan, and Eli Upfal. Using pagerank to characterize web structure. In *Proceedings of the 8th Annual International Conference on Computing and Combinatorics*, pages 330–339. Springer-Verlag, 2002. Cité page(s) 50

[92] Brian Pinkerton. Finding what people want: Experiences with the wencrawler. In *Proceedings of the First International Conference on World Wide Web*, 1994. Cité page(s) 121

[93] Pascal Pons. Algorithmique des grands réseaux d'interactions : détection de structures de communautés. Master's thesis, ENS - Paris 7, 2004. Cité page(s) 85

[94] Keith H. Randall, Raymie Stata, Janet L. Wiener, and Rajiv G. Wickremesinghe. The link database: Fast access to graphs of the web. In *Proceedings of the Data Compression Conference (DCC '02)*, page 122. IEEE Computer Society, 2002. Cité page(s) 123

[95] S. Redner. How popular is your paper? an empirical study of citation distribution. In *Eur Phys J. B.*, volume 4, pages 131–134, 1998. Cité page(s) 18, 20, 21

[96] RFC1122. Protocole tcp/ip. http://www.faqs.org/rfcs/rfc1122.html. Cité page(s) 108

[97] RFC71034. Domain name server. http://www.faqs.org/rfcs/rfc1034.html. Cité page(s) 110

[98] RFC71035. Domain name server. http://www.faqs.org/rfcs/rfc1035.html. Cité page(s) 110

[99] RFC791. Protocole ip. http://www.faqs.org/rfcs/rfc791.html. Cité page(s) 108

[100] RFC793. Protocole tcp. http://www.faqs.org/rfcs/rfc793.html. Cité page(s) 108

[101] G. Salton. The smart retrieval system. In Prentice Hall Inc, editor, *Experiments in Automatic Document Processing*, pages 207–208, 1971. Cité page(s) 66

[102] Torsten Suel and Jun Yuan. Compressing the graph structure of the web. In *Proceedings of the Data Compression Conference (DCC '01)*, page 213. IEEE Computer Society, 2001. Cité page(s) 123

[103] Raymond Wan and Alistair Moffat. Effective compression for the web: exploiting document linkages. In *Proceedings of the 12th Australasian conference on Database technologies*, pages 68–75. IEEE Computer Society, 2001. Cité page(s) 123

[104] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(1–7):440–442, 1998. Cité page(s) 8, 14, 19, 21, 22, 28, 34, 35, 81, 84, 131

[105] S.-H. Yook, H. Jeong, and A.-L. Barabasi. Modeling the internet's large-scale topology. *cond-mat/0106084*, 2001. Cité page(s) 16, 21, 22

Résumé

Le graphe du Web, plus précisément le crawl qui permet de l'obtenir et les communautés qu'il contient est le sujet de cette thèse, qui est divisée en deux parties.

La première partie fait une analyse des grand réseaux d'interactions et introduit un nouveau modèle de crawls du Web. Elle commence par définir les propriétés communes des réseaux d'interactions, puis donne quelques modèles graphes aléatoires générant des graphes semblables aux réseaux d'interactions. Pour finir, elle propose un nouveau modèle de crawls aléatoires.

La second partie propose deux modèles de calcul de communautés par émergence dans le graphe du Web. Après un rappel sur les mesures d'importances, PageRank et HITS est présenté le modèle gravitationnel dans lequel les noeuds d'un réseau sont mobile et interagissent entre eux grâce aux liens entre eux. Les communautés émergent rapidement au bout de quelques itérations. Le second modèle est une amélioration du premier, les noeuds du réseau sont dotés d'un objectif qui consiste à atteindre sa communautés.

Mots clés

Graphes - Web - Communautés - PageRank - HITS - Crawler - Regroupement - Modèles aléatoires - Loi de puissance - Bicliques - Noeud papillon - Petits Mondes - Autorités - Fans - Modèle gravitationnel - Modèle intentionnel - Parcours en largeur - Parcours en profondeur - Pertinence - Qualité.

Abstract

The modelization of the Web graph and the modelization and the extraction of communities in the graph of the Web are the subject of this thesis, which is divided into two parts. The first part makes an analysis of large graphs and introduced a new model of random crawls. We starts by defining the common properties of networks, then gives some random models for the generation of networks. To finish, we proposes a new model of random crawls.

Then, the second part proposes two models of emergence of community in the networks. After a remainder on the algorithms of classification: PageRank and HITS is presented the gravitational model in which the nodes of a network are mobile and interact to the links between them. The communities emerge quickly after some iterations. The second model is an improvement of the first, the nodes have now an objective which consists in reaching their communities.

Keywords

Graphs - Web - Communities - PageRank - HITS - Crawler - Clustering - Random models- Power Law - cores - Bowtie - Small World - Authorities - Hubs - Gravitational Model - Intentional Model - BFS - DFS - Relevance - Quality.

LIRMM —161 rue Ada —34392 Montpellier Cedex 5 —France