Un modèle de crawls aléatoires

Toufik Bennouas, Fabien de Montgolfier

Un modèle de crawls aléatoires

Toufik Bennouas et Fabien de Montgolfier

LIAFA, Université Denis Diderot, 175 rue du Chevaleret 75013 Paris *{bennouas,fm}@liafa.jussieu.fr*

Beaucoup d'auteurs ont discuté des propriétés du graphe du *Web* et proposé des modèles. Mais l'objet modélisé est dur à définir. Nous proposons à la place un modèle de *crawls* aléatoires, en modélisant l'échantillon plutôt que la population et en faisant des hypothèses simples sur la démarche utilisée pour construire un crawler, plutôt que des hypothèses complexes sur la sociologie des auteurs de pages Web. Il en ressort que certaines propriétés bien connues des crawls pourraient en réalité être des artefacts dûs aux outils utilisés lors de la capture.

Keywords: réseaux d'interactions, graphe du Web, Small Worlds, modèles de graphes aléatoires, émergence de structures, nœud papillon

1 Introduction

Il est généralement difficile d'obtenir un *grand réseau d'interaction* en entier. On doit souvent se contenter d'un échantillon de taille la plus grande possible afin de mener différentes études. Dans le cas de l'ensemble des pages hypertextes (Web), on utilise un *crawler*. Suite à l'analyse de différents *crawls* du Web, plusieurs modèles et analyses de propriétés ont été publiés. On peut reprocher à leurs auteurs une certaine confusion entre les *crawls du Web* sur lesquels ils travaillent, et le *Web lui-même*. Inférer d'un échantillon les propriétés d'une population ne peut se faire que si l'échantillonage est non biaisé, ce qui n'est bien sûr pas le cas. Par exemple, tous les crawlers tomberont rapidement sur la page de Google, même pour un échantillon de quelques milliers de pages Web parmi les milliards possibles. En plus du biais d'accessibilités, il y a un problème fondamental : le Web est potentiellement infini à cause des pages dynamiques ou générées à la requête, des mécanismes de session, etc. Chaque utilisateur en a une vue différente (penser à une page affichant l'IP du surfer) et la très forte dynamicité du Web le rend impossible à collecter avec une bande passante raisonnable. Il y a donc lieu d'essayer de modéliser ce que nous en connaissons : *les crawls*.

Tous les moteurs de recherche disposent d'un programme qui parcourt le Web pour archiver les pages Web, il est appelé *crawler*. Schématiquement, le crawler contient un module de stockage en relation avec un module de téléchargement incrémental : ce sont les liens sortants des pages stockées qui fournissent les nouvelles URLs à télécharger. Parmi les nombreuses problèmatiques soulevées est la stratégie de parcours à adopter : quelle page télécharger au sein de l'immense stock d'URL connues ? En général c'est le parcours en profondeur qui est utilisé, bien qu'une absence de stratégie (choix aléatoire) soit plus facile à implémenter. Bien entendu le processus ne s'arrête jamais.

Le *crawling* permet donc de construire un graphe, appelé *crawl* du Web, différent du «vrai» Web, objet théorique formé de «toutes» les pages et «tous» leurs hyperliens. Ces graphes ont des propriétés bien connues (voir § 2). Nous proposons dans cet article un modèle très simple de crawls, utilisant des hypothèses minimales sur ce qu'est un crawler. Puis, à l'aide de simulations, nous montrons que les propriétés des graphes que nous générons sont en fait quasiment identiques à des propriétés connues des graphes de crawls (attribuées au graphe du Web). Cela ouvre un débat : il n'est en effet pas possible d'affirmer que lesdites propriétés seraient intrinsèques au Web.

2 Description du modèle

Un *bon* modèle de crawl du Web doit reproduire les propriétés observées dans les crawls réels, dont un catalogue sommaire est :

1. fort coefficient de clustering ;

2. faible distance moyenne ;
3. degrés distribués suivant une loi de puissance ;
4. connexité (à cause du crawling incrémental)
5. source unique (si l'on part d'un seul sommet) ou peu de sources
6. forte ouverture sur l'inconnu (beaucoup de liens non-crawlés)
7. Ayant une structure de *nœud papillon* [BKM+00], controversé
8. tels que les sommets de forte connectivité (de fort PageRank) aient été découverts tôt.

Les trois premiers points sont des paradigmes classiques des réseaux d'interactions [New03] ; les cinq derniers sont typiques des crawls [BKM+00]. Nous avons décomposé le processus de génération en deux parties : pré-calcul et crawl (génération proprement dite). Les paramètres de notre modèle sont deux exposants pour les distributions de degrés, utilisés en phase de pré-calcul, et une stratégie de crawl utilisée en deuxième phase.

Phase de pré-calcul Nous posons que les degrés entrants comme sortants sont distribués en loi de puissance : la probabilité qu'un sommet soit de degré δ varie en δ^{λ} (renormalisé à 1). Dans notre phase de pré-calcul, nous produisons un pool de n sommets (l'univers des pages Web accessibles), chaque page recevant un degré entrant et un degré sortant. Le degré entrant suit une loi de puissance de paramètre λ_{in} et le degré sortant une loi de paramètre λ_{out} (des propriétés admises ici et pas considérées comme un artefact).

Ensuite est tirée la *liste de découverte*. C'est une liste où chaque sommet apparaît autant de fois que son degré entrant (une page *potentiellement* pointée par trois autres pages sera présente trois fois dans la liste, etc), permutée aléatoirement. Le premier élément de cette liste sera la page découverte en premier, le second élément la page découverte en second, etc. Ou plus exactement le premier élément est la première destination d'hyperlien trouvé au cours du crawl, etc. Donc avant de débuter le crawl est déjà connu le moment de découverte d'une page, mais non pas ses prédécesseurs, qui seront justement attribués en seconde phase.

Phase de crawl Le crawl débute à partir d'un sommet choisi aléatoirement. Chaque sommet est «téléchargé» (visité) une seule fois. Les n pages potentielles sont donc divisées en trois groupes : les pages *téléchargées*, les pages *à télécharger* (connues car déjà pointées) et les pages *inconnues* (non encore pointées).

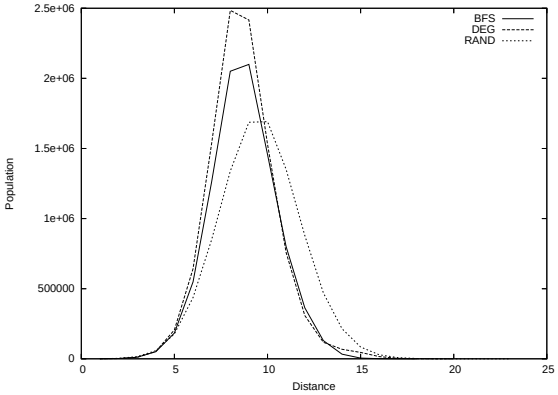
L'algorithme de crawl choisit une page p de l'ensemble des pages *à télécharger*. Le degré sortant $d^+(p)$ a déjà été calculé en première phase et correspond aux liens sortants de la page. Sont alors extrait $d^+(p)$ pages de la *liste de découverte*, ce qui revient à assigner une extrémité à ces hyperliens. Parmi ces pages pointées, celles qui sont non téléchargées sont insérées dans l'ensemble des pages à télécharger. Le seul choix à faire est, une fois pour toute, la stratégie de parcours adoptée, c'est-à-dire la façon dont la liste des pages à télécharger est gérée (et donc quelle page p sera extraite à la prochaine étape). Nous avons comparé quatre stratégies de parcours simples :

- 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 : le sommet le plus pointé est choisi. La structure de données est une file de priorité ;
- RANDOM : le sommet u est choisi de façon aléatoire et uniforme parmi les sommets marqués

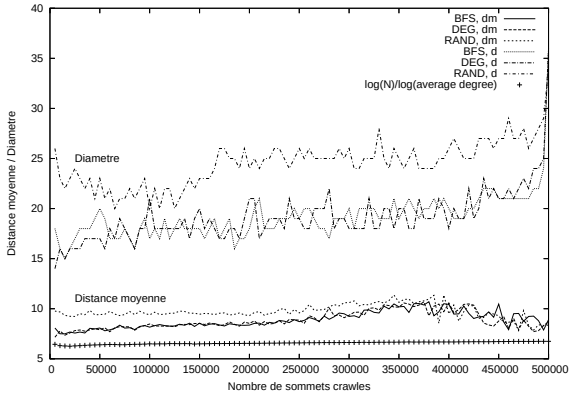
Le processus de génération ne s'arrête que lorsque la liste des sommets marqués est vide ou encore que tous les sommets sont visités et donc la liste de voisinage est vide. Noter que les graphes générés peuvent avoir quelques boucles et multi-arcs, facilement supprimés, et que le degré entrant calculé en phase de pré-calcul n'est qu'un degré maximal, correspondant au cas peu probable où l'arret serait causé par épuisement de la liste de découverte. En résumé, on ne parcours pas un graphe aléatoire mais on génère le graphe en le parcourant. La première phase construit des origines et des extrémités de liens, la deuxième les relie.

3 Résultats

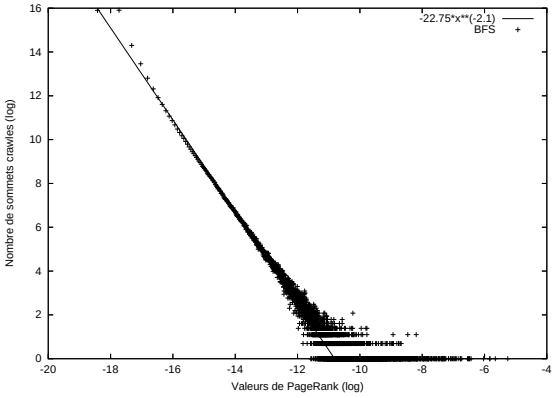
Nous présentons ici le résultat de simulations, où les distributions des degrés (données) suivent les paramètres couramment acceptés [BKM+00] de -2.1 comme exposant des degrés entrants et -2.72 pour les degrés sortants. À chaque instant du crawl, les degrés suivent ces lois par construction.



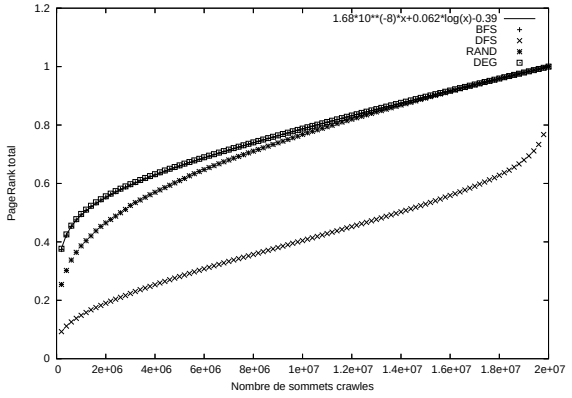
La **distribution des distances** entre tous couples de sommets suit une loi gaussienne pour les trois stratégies BFS, DEG et RANDOM. La distribution est ici calculée après un crawl de 150,000 sommets mais persiste jusqu'à la fin du crawl. Pour le DFS (non dessiné), les distances sont immenses et ne suivent pas de loi clairement identifiable.



Évolution de la **distance moyenne** (courbes du bas) et du **diamètre** (courbes du haut) au cours du crawl pour BFS, DEG et RANDOM. Ces deux paramètres évoluent très lentement et dépendent peu de la taille. On note que le diamètre est très petit par-rapport à la distance moyenne (cf. figure de gauche).



La **distribution des valeurs de PageRank** suit une loi de puissance de paramètre -2.1. Cette caractéristique a été observée dans [PRU02]. Le PageRank est utilisé par Google pour classer les pages Web [PBMW98] c'est un important (et utile) biais de crawling.

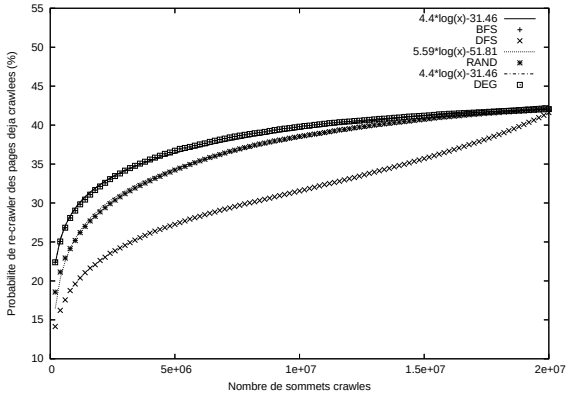


Le **PageRank total** accumulé au cours du temps. BFS permet d'obtenir des sommets *pertinents* rapidement. Cette caractéristique du BFS a été observée par [APC03, NW01]. [Mat04] note qu'il existe une assez forte corrélation entre le PageRank et les degrés entrants.

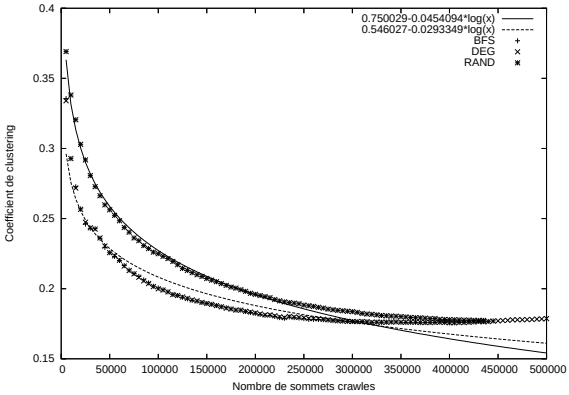
Nous nous sommes intéressés également aux différentes composantes du crawl au sens du **modèle en noeud papillon** [BKM+00]. Dans nos graphes la composante fortement connexe géante augmente jusqu'à 65% du graphe. Inversement, la taille de la composante OUT diminue mais n'est pas négligeable. On observe que nos crawls contiennent une très petite composante *IN* : elle est parfois réduite à un sommet quand on part d'une source unique !

Enfin, nous nous sommes intéressés à l'**énumération des bicliques**. Typiquement 3% des sommets font partie d'une biclique (4,4) ! Sans attachement préférentiel nous avons donc la présence de *cores*, bicliques constituées de *hubs* et d'*autorités*. [Ben05] donne un tableau plus complet de ces résultats.

La comparaison des quatre stratégies est très défavorable au DFS, qui n'a pas du tout les propriétés des crawls du Web. Les trois autres stratégies se valent. Sauf que en début de crawl BFS et DEG sont deux fois meilleures que RANDOM pour trouver les pages de fort PageRank ! C'est une mauvaise nouvelle pour les crawlers qui n'implémentent pas de stratégie.



Évolution de la **probabilité qu'une extrémité de lien soit déjà téléchargée** au cours du temps. On observe en accord avec [PBMW98] que le crawl s'étend vers l'inconnu.



Évolution du **coefficient de clustering** au cours du temps. Il semble converger vers une constante et reste nettement supérieur au coefficient de clustering d'un graphe aléatoire d'Erdös et Rényi peu dense.

4 Conclusion

Nous avons proposé un modèle de génération de crawl du Web simple, les seuls paramètres nécessaires étant les exposants des lois suivies par les degrés sortants et entrants. Dans ce papier, les lois de puissances sont admises, bien qu'il soit possible qu'elles soient elles aussi un artefact de crawling.

Le processus de génération quant à lui n'est rien d'autre qu'une simulation de parcours de graphe par une stratégie donnée, identique au processus de crawl du Web. Les résultats montrent que les crawls obtenus ont beaucoup de propriétés semblables à celles réputées possédées par le graphe du Web. 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. Peut-être que beaucoup des propriétés observées sont en fait un artefact de crawling...

Du côté prospectif, nous regrettons de n'avoir pu nous livrer à une étude formelle de notre modèle. Les dépendances nombreuses et complexes (temporelles, etc) rendent inopérantes les procédés de champ moyen et autres qui servent à prouver l'existence de propriétés simples dans certains modèles [Bol85, ABJ99, ACL00]. Bien entendu nos crawls générés n'ont pas toutes les propriétés répertoriées des crawls réels. Par exemple, les sommets de fort degrés sont très liés entre eux. Complexifier le modèle en apporterait davantage, mais nous pensons qu'en présentant un modèle simple produisant tant de propriétés, notre argumentation sur le caractère non probant des propriétés du Web a plus de force.

[ABJ99] R. Albert, A.-L. Barabasi, and H. Jeong. Diameter of the world wide web. *Nature*, 401, 1999.

[ACL00] 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.

[APC03] 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.

[Ben05] Toufik Bennouas. Modélisation de crawl du web et calcul de communautés par émergence. thèse de doctorat, université montpellier II, 2005.

[BKM+00] 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.

[Bol85] B. Bollobás. Random graphs. *Academic Press*, 1985.

[Mat04] Fabien Mathieu. Graphes du web, mesures d'importance à la pagerank. thèse de doctorat, université montpellier II, 2004.

[New03] M. Newman. The structure and function of complex networks. *SIAM Review*, 45(2) :167–256, 2003.

[NW01] 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.

[PBMW98] 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.

[PRU02] 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.