Evaluation rapide du diamètre d'un graphe

Christian Belbeze[*,**], Max Chevalier[*,***]
Chantal Soule-Dupuy[*,**]

[*]Institut de Recherche en Informatique de Toulouse
[**]Université Toulouse 1 Capitole, 2 rue du Doyen Gabriel Marty, 31042 Toulouse Cedex 9
[***]Université Paul Sabatier, 118 Route de Narbonne, 31062 Toulouse Cedex 9
christian@belbeze.com, chevalier@irit.fr, soule@irit.fr

Résumé. Lors de l'analyse de graphes, il est important de connaître leurs propriétés afin de pouvoir par exemple identifier leur structure et les comparer. Une des caractérisations importante de ces graphes repose sur le fait de déterminer s'il s'agit ou non d'un "petit monde". Pour ce faire, la valeur du diamètre du graphe est essentielle. Or la mesure du diamètre est pour un très grand graphe, une opération extrêmement longue. Nous proposons un algorithme en deux phases qui permet d'obtenir rapidement une estimation du diamètre d'un graphe avec une proportion d'erreur faible. En réduisant cet algorithme à une seule phase et en acceptant une marge d'erreur plus élevée, nous obtenons une estimation très rapide du diamètre. Nous testons cet algorithme sur deux grands graphes de terrain (plus d'un million de nœuds) et comparons ses performances avec celles d'un algorithme de référence BFS (Breadth-First Search). Les résultats obtenus sont décrits et commentés.

1 Introduction

Depuis quelques années, la recherche sur les grands graphes a soulevé de plus en plus d'intérêt. Quelle que soit la nature des réseaux représentés, il est indéniable que le nombre d'éléments (noeuds et liaisons) les constituant et pris en compte dans leurs études n'a cessé d'augmenter. L'identification de communautés dans les réseaux sociaux est un de ces domaines d'étude sur lequel nous nous sommes focalisés dans nos travaux et qui nous a amené à étudier les caractéristiques des graphes sous-jacents à ces réseaux (Belbeze et al., 2009).

Bien que de nature différente, les réseaux de "grande taille" en général (réseaux d'ordinateurs, réseaux sociaux, réseaux de mots, pages Web reliées par des hyperliens et autres réseaux d'échange), ont des caractéristiques qui comportent un grand nombre de similitudes. Ces similitudes ont permis de créer un type de graphe nommé "Grand graphe de terrain". Par définition, ces graphes venus du monde réel ne sont donc pas issus d'une formule mathématique. Ils existent sur le terrain et les noeuds se doivent d'avoir une existence physique. La phrase de Watts et Al. en 1998 (Watts et Strogatz, 1998) "la plupart des graphes de terrain ont des propriétés non-triviales en commun" peut être considérée comme la consécration de leur domaine d'étude. Parmi ces propriétés "non-triviales en commun", la valeur du diamètre d'un graphe

est essentielle. Le diamètre est défini comme la distance la plus élevée entre deux noeuds en utilisant le chemin le plus direct. Il est l'élément qui permet de classer un graphe en "petit monde". Les grands graphes de terrain et les graphes générés aléatoirement sont généralement de type petit monde et présentent un diamètre faible au regard du nombre de noeuds. Il reste cependant important de pouvoir valider cette caractéristique pour s'assurer que le graphe étudié ne fait pas exception à la règle. Toutefois, le coût en temps CPU d'un calcul exact avec les algorithmes classiques est prohibitif sur des réseaux de plusieurs millions de noeuds.

Dans cet article, nous présentons un algorithme spécifique permettant d'obtenir une évaluation rapide du diamètre d'un graphe et cela indépendamment de sa complexité. Nous appliquerons cet algorithme à quatre familles de graphes générés aléatoirement et à deux exemples de grands graphes de terrain contenant plus d'un millions de noeuds. Nous comparons les résultats obtenus avec le diamètre retourné par un algorithme exhaustif de type BFS (Breadth-First Search) (Najork, 2001) recherchant les noeuds les plus éloignés pour chaque noeud du graphe.

2 Contexte et état de l'art

Avec la notion de "petit monde" apparue en 1969 dans les travaux de Stanley Milgran, et plus particulièrement avec le fameux concept des six degrés de séparation (Travers et Milgram, 1969), le diamètre est devenu un élément primordial de la caractérisation d'un graphe. On retrouvera des niveaux de séparation limités (faible diamètre) dans des réseaux aussi variés que des réseaux de téléphones (Wasserman et Faust, 1994), d'ordinateurs (comme Internet) (Magnien et al., 2009) et de pages sur le Web (Faloutsos et al., 1999) (Huberman et Adamic, 1999) (Tauro et al., 2001).

Les réseaux de type "petit monde" ont été principalement analysés dans l'univers des réseaux sociaux. Le sociologue Duncan Watts a montré que l'on pouvait comparer des réseaux en normalisant leurs paramètres et en les comparant aux valeurs issues de réseaux aléatoires (Watts et Strogatz, 1998) (Watts, 1999). Grâce à ces paramètres, il est également possible de mieux comprendre les propriétés dynamiques des réseaux sociaux. Les réseaux sont alors plus ou moins sensibles selon la nature de leurs objets et leurs caractéristiques à différentes attaques : contagion de l'information, risque financier, etc... Ils présentent aussi des architectures organisationnelles et managériales comparables. Ces réseaux, une fois représentés comme des graphes, sont de manière quasi générale classés comme appartenant au type "petit monde". Cette appartenance est définie par une faible distance maximale entre deux nIJuds (diamètre) et par un fort coefficient de "clustering". Le coefficient de clustering global (Watts et Strogatz, 1998) peut être défini comme le rapport entre le nombre de triads (ensemble de trois nœuds tous connectés deux à deux) et le nombre d'ensembles de trois noeuds connectés (deux à deux ou pas). La classification d'un réseau en "petit monde" permet donc de mieux en cerner ses propriétés et ses comportements éventuels. Le diamètre faisant partie de ces propriétés.

Cependant, les méthodes exhaustives de mesure de l'ensemble des chemins sont parfois inutilisables. Ainsi, la mise en oeuvre d'algorithmes de recherche basés sur leur ré-itération sur chacun des noeuds (par exemple l'algorithme de Dijkstra (Dijkstra, 1959) et BFS (Najork, 2001) est bien sur impossible. D'un coût CPU complètement prohibitif, cette méthode ne répond pas au problème. Ces algorithmes sont néanmoins encore utilisés (Bader et Madduri, 2006) pour évaluer différentes méthodes de programmation parallèle sur des supercalculateurs.

La mesure du diamètre par une exploration de l'ensemble des chemins et leurs comparaisons (Bader et Madduri, 2006), quelle que soit la méthode utilisée, reste une méthode trop lourde pour traiter les très grands graphes de terrain. Plusieurs travaux ont tenté de résoudre la difficulté de la mesure du diamètre d'un grand graphe par la mesure de bornes supérieure et inférieure comme par exemple la méthode proposée par C. Magien et al. (Magnien et al., 2009) qui est comparable à notre proposition dans son objectif. Cependant le positionnement des deux méthodes est différent. La méthode de C. Magnien et al. propose de borner le diamètre par un ensemble d'algorithmes . Une valeur haute est obtenue par une simplification préalable du graphe en arbre. La valeur basse correspond à la distance maximale parcourable depuis des noeuds sélectionnés aléatoirement. Notre démarche est différente, nous considérons le graphe dans son entière complexité, mais avançons l'hypothèse que certains noeuds possèdent, par leurs caractéristiques, une plus haute probabilité d'être les extrémités du diamètre du graphe. En utilisant ces nœuds comme noeuds de départ d'une exploration, nous pouvons rapidement obtenir une estimation minorée ou exacte de la valeur du diamètre.

3 Algorithme du "Bout du Monde"

L'algorithme que l'on propose est basé sur l'intuition que chaque noeud d'un graphe, en fonction de sa position, n'a pas la même probabilité d'être l'extrémité du diamètre du graphe. Supposons que nous devions le plus rapidement possible, et cela de manière manuelle, définir le diamètre d'un graphe.

Dans l'exemple de la carte de France, figure 1, l'oeil cherche rapidement à localiser des villes qui se positionnent comme les plus "extrèmes". Dunkerque et Nice ou Strasbourg et Pau sont sans aucun doute plus susceptibles d'être les extrémités du diamètre de ce graphe que Limoges, Clermont-Ferrand ou Orléans.

Dans l'algorithme présenté dans les sections suivantes, les noeuds choisis comme candidats sont soit des noeuds présentant la plus forte distance à un noeud de départ, soit des noeuds ayant un degré unitaire (le degré est définit dans des graphes non orientés et non pondérés, comme ceux étudiés ici, comme le nombre de liaisons que le noeud possède). Nous nommerons ces noeuds, qui sont les extrémités du diamètre du graphe, "noeuds du bout du monde".

La méthode proposée a été construite empiriquement en choisissant un nombre minimal de noeuds à considérer comme candidats pour être l'extrémité du diamètre tout en conservant un taux et des valeurs d'erreurs acceptables.

Préliminaire : classement des noeuds par nombre de liaisons. Avant d'exécuter à proprement parler l'algorithme, une opération préliminaire est nécessaire. Elle consiste à rechercher le noeud qui possède le degré le plus élevé. Ce noeud est utilisé comme "Noeud de départ". Le choix s'explique simplement par le fait qu'étant en contact avec le plus grand nombre de noeuds possible, il est susceptible de nous mener aux noeuds du bout du monde rapidement. Un algorithme de classement quelconque peut être utilisé ici. Cette partie ne sera donc pas détaillée dans cet article. Cependant le temps lié à cette opération préparatoire sera comptabilisé dans nos expérimentations.

Phase I : recherche de noeuds du bout du monde en partant du noeud de degré maximal. Cette phase peut être comparée à un système qui tenterait de mesurer le diamètre d'une mare

FIG. 1 – *Détection intuitive des extrémités du diamètre d'un graphe : Dunkerque, Nice.*

en comptant des ronds dans l'eau. On lance un caillou dans le centre de la mare et on recherche l'emplacement où se forme l'ondulation la plus distante (en ronds dans l'eau) du point central. On jette ensuite un caillou à ce dernier emplacement et on recommence jusqu'à revenir sur un endroit d'où l'on n'a pas déjà jeté un caillou. Le nombre maximal de ronds dans l'eau mesuré est notre estimation du diamètre de la mare.

L'algorithme de la phase I (figure 2) présente un caractère récursif par l'appel de la fonction ***Mesure_distance_Max*** . Cette fonction est réutilisée pour trouver les noeuds du bout du monde à partir du noeud de départ et réutilisée ensuite pour trouver les noeuds les plus éloignés de ces noeuds. Le point d'arrêt de la récursivité étant donné par le fait que le ou les points trouvés comme le ou les plus distants du point de départ ont déjà été utilisé(s) comme point de départ. Cette recherche des noeuds "les plus éloignés" a été nommée "Recherche du bout du monde".

L'algorithme de "recherche du bout du monde" utilise trois ensembles :

- l'ensemble ***Noeuds_déjà_testés_comme_extrémités*** pour stocker les noeuds déjà traités comme points de départ de la recherche Mesure-distance-Max (afin d'éviter ainsi de repartir dans une exploration inutile) ;
- l'ensemble ***Derniers_noeuds_trouvés*** pour stocker les derniers noeuds parcourus dans l'exploration ;
- l'ensemble ***Noeuds_en_cours*** pour stocker l'ensemble des noeuds déjà parcourus par l'algorithme.

Les ***Derniers_noeuds_trouvés*** viennent rejoindre, à chaque début de nouvelle exploration, les ***Noeuds_en_Cours***.

Trois variables sont utilisées :

```
[Initialisation]
Diamètre = 0 ;
Distance = 0 ;
Nœud_de_départ = nœud ayant le plus grand degré

//---------------------------------------------------------------------------------------- Phase I
Diamètre = Mesure_distance_Max (Nœud_de_départ, 0)
//---------------------------------------------------------------------------------------- Phase II
Pour chaque nœud ayant un degré unitaire et non présent dans Nœuds_déjà_testé_comme_extrémités
faire
    Distance = Mesure_distance_Max (nœud, 0)
    Si Diamètre < Distance alors
      Diamètre = Distance
    Fin de si
Fin de pour

// Fonction récursive de recherche du diamètre
Fonction Mesure_distance_Max (NœudDépart, distanceActuelle)
    Vider Nœuds_en_cours
    Vider  Derniers_nœuds_trouvés
    Placer NœudDépart dans Nœuds_en_cours
    Dist = distanceActuelle
    Ajouter NœudDépart à Nœuds_déjà_testés_comme_extrémités
    Ajouter NœudDépart à Nœuds_en_cours
    S'il existe un nœud voisin du Nœuds_en_cours alors
        Placer tous les nœuds en liaisons avec Nœuds_en_cours dans Derniers_nœuds_trouvés
        Dist = Dist +1
        Placer les nœuds de Derniers_nœuds_trouvés dans Nœuds_en_cours
        Tant que l'on trouve des nœuds dans Derniers_nœuds_trouvés faire
          Dist = Dist + 1
          Placer les nœuds du  Derniers_nœuds_trouvés dans Nœuds_en_cours
          Vider Derniers_nœuds_trouvés
          Placer tous les nœuds en liaisons avec Nœuds_en_cours dans Derniers_nœuds_trouvés
        Fin de tant que
    Fin de si
    // on est positionné sur les nœuds les plus éloignés du nœud de départ
    Pour chaque nœud de Nœuds_en_cours faire
        Si le nœud n'est pas dans le vecteur Nœuds_déjà_testés_comme_extrémités alors
            Stocker le nœud dans le vecteur Nœuds_déjà_testés_comme_extrémités
            Si Diamètre < = Dist alors
                Diamètre = Dist
                Dist = Mesure_distance_Max (nœud, Dist)
                Si Diamètre < Dist alors
                  Diamètre  = Dist
                Fin de si
            Fin de si
        Fin de si
    Fin de pour
    Retourner Diamètre
Fin de fonction
```

FIG. 2 – *Algorithme "Recherche du bout du monde" - Evaluation du diamètre d'un graphe.*

- ***Dist*** : représente la distance entre les noeuds de départ et les noeuds en cours dans une exploration. Cette variable est interne à la fonction *Mesure_distance_Max* ;
- ***Distance*** : représente la valeur maximale retournée par une exploration complète ;
- ***Diamètre*** : représente la valeur maximale de la distance rencontrée dans l'ensemble de l'exploration et sera retournée comme valeur du diamètre effectif.

Phase II : recherche de noeuds du bout du monde en partant de noeuds excentrés. Si la phase I permet de parcourir rapidement le graphe, dans certain cas, elle ne permet pas de retourner une valeur exacte. Le plus souvent, ces "erreurs" sont dues à la non prise en compte de noeuds qui possèdent un degré unitaire (reliés aux autres nœuds par un seul lien). La phase I donne alors un diamètre minoré.

Pour contourner ce problème, la phase II de l'algorithme va ensuite rejouer la même fonction récursive sur les nœuds de la composante connexe en train d'être mesurée en prenant comme nœuds de départ les nœuds de degré unitaire.

La particularité de l'algorithme repose sur le fait que l'on ne teste les "distances" qu'entre certains nœuds.

4 Validation fonctionnelle de l'algorithme de "Recherche du bout du monde"

4.1 L'algorithme de référence

Pour valider l'évaluation du diamètre d'un graphe, et ce quelles que soient les combinaisons des nœuds entre eux, nous avons comparé sur plusieurs milliers de graphes aléatoires les résultats de cet algorithme avec ceux d'un algorithme exhaustif BFS (Najork, 2001). L'algorithme BFS pour Breadth-First Search, utilisé comme élément référent, calcule récursivement pour chaque nœud la distance maximale vers tous les autres nœuds, le diamètre du graphe représentant alors la valeur maximale de ces distances impliquant un usage prohibitif des ressources.

4.2 Les graphes de test

Des graphes aléatoires ont été créés pour représenter des exemples de graphes et notamment des exemples de graphes de terrain et de petits mondes. Les dimensions et le nombre de graphes testés ont été choisis de façon à ce que l'ensemble de la campagne de test puisse s'effectuer dans des temps CPU inférieurs à un mois. Les graphes sont créés selon trois méthodes différentes :

Méthode 1 : "Une composante connexe". Dans ce premier algorithme, chaque nœud rajouté est lié à un seul nœud précédent de manière aléatoire. Une fois la composante connexe terminée, un nombre aléatoire de liaisons sont éventuellement rajoutées. Ce nombre de liaisons supplémentaires pouvant être compris entre 0 et le nombre maximal que le graphe peut porter. Cette méthode génère des graphes constitués d'une seule composante connexe. Elle servira à créer deux échantillons de test :

– Un échantillon nommé "Petite Mono-Composante" contenant 1 000 000 de graphes de 10 à 100 nœuds ;
– Un échantillon nommé "Mono-Composante" contenant 10 000 graphes contenant de 100 à 1 000 nœuds.

Méthode 2 : "erdos-renyi-graph". Les graphes sont générés par l'utilisation de la fonction : "networkx.générators.random-graphs.erdos-renyi-graph" dans la librairie NetworkX. Le premier paramètre est généré aléatoirement entre 10 et 1 000 et le second entre 0 et 1 (fonction décrite en détail sur le site http ://networkx.lanl.gov/).

Cette fonction génère des graphes selon deux paramètres : le premier paramètre est le nombre de nœuds, le second la probabilité de présence de liaison. Cette fonction génère un échantillon de test nommé "erdos-renyi-graph" contenant 100 000 graphes contenant de 10 à 1 000 nœuds.

Méthode 3 : "newman-watts-strogatz". Les graphes sont générés par l'utilisation de la fonction : networkx.générators.random-graphs.newman-watts-strogatz-graph de la librairie NetworkX.

Cette fonction génère des graphes selon deux paramètres. Le premier paramètre est le nombre de nœuds, le second la probabilité de présence de liaison. Le premier paramètre est généré aléatoirement entre 10 et 1000 et le second entre 0 et 1 (fonction décrite en détail sur le site http ://networkx.lanl.gov).

Cette méthode nous servira à générer un échantillon de test nommé "newman-watts-strogatz" contenant 100 000 graphes contenant de 10 à 1000 nœuds.

Le tableau 1 présente les caractéristiques des différents graphes étudiés.

	Moy nb nœuds	Max nb nœuds	Min nb nœuds	Moy diam	Max diam	Min diam	Ecart type diam
Petite Mono-composante	59	100	10	2	9	1	0.96
Mono-composante	595	1000	100	13	34	2	7.98
Erdos-renyi-graph	507	1000	10	4	66	0*	4.20
Newman-watts-strogatz	347	735	64	5	9	2	1.93

TAB. 1 – *Détails des différents échantillons de graphes. *La valeur de diamètre à 0 signifie l'absence de liaison dans le graphe.*

4.3 Les résultats

L'algorithme de "Recherche du bout du monde" n'est pas exhaustif. Il est donc susceptible d'évaluer le diamètre d'un graphe de manière erronée. Ces erreurs d'estimation doivent être à la fois évaluées en importance et en fréquence. Il faut aussi pouvoir estimer le gain éventuel de temps que cet algorithme peut faire gagner.

Dans la figure 3, le diamètre est effectivement de 4. Les nœuds 5 et 13 sont les extrémités. En commençant par le nœud 12, l'algorithme de "Recherche de bout du monde" effectue un premier saut sur 6, 10, 2, 15, 14, 15, 4, 9, puis un second saut sur 5, 3, 16, 1, 13 ,17 , 11 et enfin

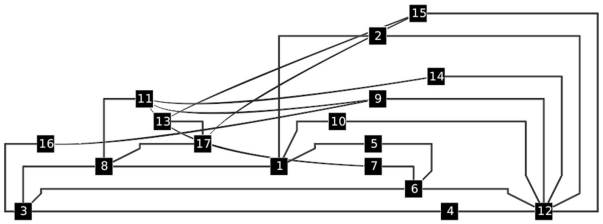

FIG. 3 – *Exemple de graphe sur lequel l'algorithme de "Recherche du bout du monde" renvoie un diamètre inférieur de 1 à la valeur réelle renvoyée par BFS.*

un troisième saut sur 8. La distance est alors de 3. On repart du nœud 8 pour un premier saut vers 1, 3, 17, 11, puis un second saut vers 2, 10, 5, 4, 6, 16, 4, 15, 13, 7, 14 et enfin retombe sur le nœud 12. Le nœud 12 ayant déjà été testé et le graphe ne comportant pas de nœud ayant un degré unitaire, l'algorithme renvoie un diamètre de 3 qui est faux.

Les Erreurs. Une erreur est comptabilisée chaque fois que l'algorithme "Recherche du bout du monde" ne donne pas le même résultat que BFS. La proportion d'erreurs est extrêmement différente selon le type de graphe généré. Dans le tableau 2, l'importance maximale de l'erreur est l'écart maximal rencontré entre la valeur retournée par BFS et celle retournée par l'algorithme de "Recherche du bout du monde". Si D_{BFS} représente le diamètre retourné par BFS et D_{BDM} le diamètre retourné par l'algorithme de recherche du bout du monde, l'importance maximale de l'erreur est notée :
Max($|D_{BFS}$-$D_{BDM}|$)

	% erreurs	Nb erreurs	Nb graphes testés	Valeur max. de l'erreur	%. d'erreur moyen	Ecart type de l'erreur
Petite Mono-composante	0.0001	1	1 000 000	1	0.00005	0.001
Mono-composante	0	0	100 000	N/A	0	0
Erdos-renyi-graph	0	0	100 000	N/A	0	0
Newman-watts-strogatz	0.01	10	100 000	1	0.002	0.01029

TAB. 2 – *Détails des différents graphes testés et des erreurs rencontrées.*

La probabilité d'erreur semble extrêmement variable selon les générateurs de graphe employés (cf. tableau 2) . Avec 1 seule erreur sur 1 000 000, les mono-composantes semblent peu affectées. Les graphes newman-watts-strogatz avec 1 erreur pour 10 000 graphes sont plus impactés.

Dans tous les cas d'erreur rencontrés, l'algorithme de "Recherche de bout du monde" donne un écart de 1 par rapport au diamètre réel renvoyé par BFS. En cas d'erreur, le diamètre renvoyé par BFS est toujours supérieur.

Les performances Le gain en performance est lui aussi très différent selon les caractéristiques du graphe. L'algorithme de "Recherche de bout du monde" peut même être plus lent que

BFS sur des graphes de tout petit diamètre. Cependant, sur des graphes présentant un nombre de composantes connexes raisonnables (inférieur au nombre de nœuds divisé par 10) et des diamètres suffisants (supérieur à 6) il peut être notablement plus rapide. Sur l'échantillon "Petite Mono-Composante", pour un diamètre de 9, le gain est en moyenne de 95% du temps CPU (cf. figure 4).

L'algorithme de "Recherche de bout du monde" est en effet moins efficace sur des graphes présentant de petites composantes connexes. Dans le cas où une composante connexe est constituée, par exemple, de deux ou trois nœuds, tous les nœuds seront alors testés comme candidats pour être l'extrémité du diamètre. De même, si le diamètre du graphe est très faible en partant du nœud le plus connecté, un très nombre de candidats seront testés.

En conclusion, l'erreur semble contenue (pourcentage moyen d'erreur à 0,002) et l'algorithme offre une amélioration significative en terme de performance (supérieure à 60% à partir d'un diamètre de 7).

5 Algorithme simplifié

Sur des grands graphes de terrains contenant un grand nombre de nœuds de degré unitaire (ayant une seule liaison), la mise en œuvre de la Phase II de l'algorithme proposé peut représenter un temps de calcul trop important. En effet, la Phase II rejoint en coût de temps CPU, pour les nœuds de degré unitaire, celui de l'algorithme BFS.

En partant d'échantillons similaires à ceux du précédent chapitre, nous évaluons le risque d'erreur pris par la suppression de la Phase II de l'algorithme.

	% erreurs	Nb erreurs	Nb graphes testés	Valeur max erreur	% max erreur	% moy erreur	Ecart type moy erreur
Petite Mono-composante	0.0415	415	1 000 000	2	20	0.0034	0.0214
Mono-composante	0.0540	54	100 000	2	11.1	0.0057	0.0303
Erdos-renyi-graph	0.1070	107	100 000	5	18.5	0.0345	0.0614
Newman-watts-strogatz	0.0110	11	100 000	1	12.5	0.0020	0.0104

TAB. 3 – *Détails des différents graphes testés et des erreurs rencontrées en utilisant l'algorithme simplifié (uniquement Phase I).*

Comme on peut le constater en comparant le contenu des tableaux 2 et 3, la suppression de la Phase II de l'algorithme introduit un risque d'erreur plus élevé. L'erreur, en valeur, peut ainsi atteindre -5. Cette erreur est rencontrée sur un graphe de diamètre 27, soit un taux d'erreur de 18.5%. En cas d'erreur, le diamètre renvoyé par BFS est toujours supérieur.

Pour mesurer le gain de temps, nous allons expérimenter l'algorithme simplifié sur deux graphes de terrain. Le premier possède 2 800 000 nœuds et l'autre 1 294 245 nœuds. Les deux graphes ont été placés dans des bases de données de type SQL server 2008. La machine responsable du calcul du diamètre est un ordinateur sous Windows 2008 server 32 bits avec 4 Giga de RAM, un processeur 4 cIJurs à 2.6 Giga hertz. Le code est développé avec Microsoft Visual Studio 10.

Le premier graphe est issu des données des recherches effectuées par les utilisateurs du réseau de "Pear to Pear" eDonkey et correspond à 10 semaines de logs de requêtes (Wasserman

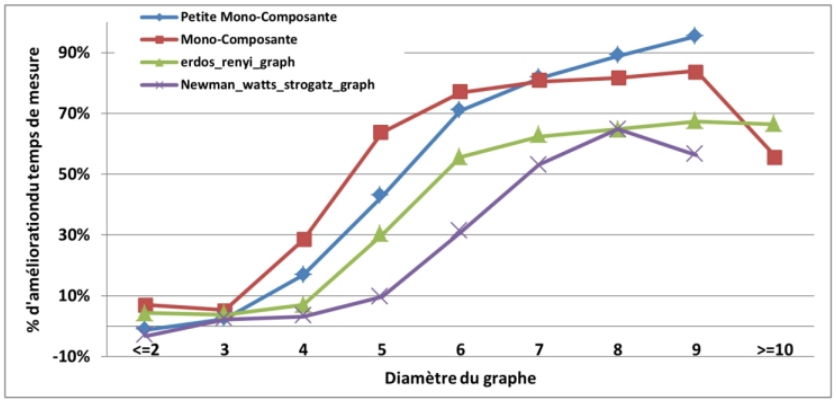

FIG. 4 – *Gain moyen en temps de l'algorithme de Recherche du bout du monde par rapport à BFS.*

et Faust, 1994)[13]. Les nœuds représentent donc des mots et les liaisons représentent les utilisations couplées des mots dans une même requête par un utilisateur (co-occurrence). Les fichiers sont disponibles à l'adresse : http ://antipaedo.lip6.fr/Ten-Weeks.pdf.

Le second graphe est issu des fichiers de logs de recherches d'utilisateurs sur AOL.com durant deux mois en 2004. Comme dans l'exemple précédent, les nœuds représentent les mots et les liaisons leurs utilisations couplées. Les fichiers sont disponibles à l'adresse : http ://gregsadetsky.com/aol-data.

Les deux graphes sont constitués d'une composante connexe qui contient plus de 90% des nœuds et plus de 99% des liaisons.

Les résultats Nous avons effectué en premier lieu l'estimation sur cette composante principale seule. L'algorithme a très rapidement fourni une estimation du diamètre de cette composante.

No.	Nœuds	Paires	Diamètre	Temps de recherche BFS estimé	Temps d'estimation algorithme simplifié	Nb composantes connexes testées
1	2 741 251	33 291 281	13	8 350 jours*	7.01 minutes	1
2	1 239 001	12 480 298	13	1 739 jours*	3.60 minutes	1

TAB. 4 – *Temps des estimations sur la composante connexe principale. * les estimations sont basées sur le temps nécessaire pour une exploration du type BFS à partir de 100 nœuds.*

Sur plus de 99% des liaisons du graphe (cf. tableau 4) et plus de 95% des nœuds, le diamètre du graphe est estimé en quelques minutes. L'algorithme de "Recherche du bout du monde" - Phase I est donc extrêmement efficace sur cette partie du graphe.

L'estimation sur l'ensemble du graphe est moins rapide (cf. tableau 5). Le nombre de composantes connexes de plus petite taille étant relativement important, l'algorithme est alors pénalisé.

No.	Nœuds	Paires	Diamètre	Temps de recherche BFS estimé	Temps d'estimation algorithme simplifié	Nb composantes connexes testées
1	2 833 164	33 318 555	13	8 625 jours*	8 heures 45 mn	21 788
2	1 294 245	12 511 959	13	1 834 jours*	8 heures	25 063

TAB. 5 – *Comparaison des temps de mesure sur l'ensemble du graphe. * les estimations sont basées sur le temps nécessaire pour une exploration du type BFS à partir de 100 nœuds.*

L'algorithme simplifié de "Recherche de bout du monde" permet un gain de temps supérieur à 99.98% par rapport à BFS sur les deux grands graphes de terrain testés. Le temps d'estimation du diamètre sur la composante connexe principale avec l'algorithme simplifié "Recherche de bout du monde" - Phase I est sans commune mesure avec celui nécessaire à BFS.

6 Conclusion

L'algorithme de "Recherche du bout du monde" est particulièrement efficace sur les graphes présentant une composante connexe majoritaire. Dans ce cas-là, et si les diamètres sont suffisants (supérieur à 5), les estimations sont réellement rapides avec un taux d'erreur moyen faible.

Pour des besoins d'estimation rapide (caractérisation de l'évolution du diamètre des graphes de grandes tailles), si les caractéristiques du graphe sont adaptées à l'usage de l'algorithme (diamètre suffisant et proportion de nœuds unitaires faible) celui-ci est bien adapté tant en terme de temps de calcul, qu'en terme d'erreurs (faible relativement à l'approche de référence BFS). Si une valeur intégrant une fourchette d'erreur plus large est acceptée et que les graphes contiennent une proportion considérable de nœuds de degré unitaire, il est possible de se contenter de l'exécution de l'algorithme simplifié. Particulièrement véloce sur les composantes connexes importantes, l'algorithme permet d'évaluer le diamètre d'un grand graphe rapidement. La proportion d'erreurs en fonction du type de graphe est un élément important à considérer pour estimer la qualité des évaluations. En effet, le taux d'erreur est variable selon le type de graphe. Il conviendra donc d'estimer plus précisément, et en fonction du taux d'erreur accepté, la proportion d'erreurs sur la nature des graphes étudiés.

Il est possible, par l'étude des cas particuliers créant des erreurs, d'enrichir l'algorithme de "Recherche du bout du monde" de phases supplémentaires et d'avoir ainsi une panoplie d'algorithmes permettant de trouver le juste équilibre entre erreur acceptée et vitesse attendue. Il sera ainsi possible d'observer en temps réel l'évolution d'un réseau. Sur un réseau social par exemple, on pourrait mesurer dynamiquement l'évolution des communautés d'utilisateurs et de leurs comportements.

Références

Bader, D. et K. Madduri (2006). Parallel algorithms for evaluating centrality indices in real-world networks. In *Parallel Processing, 2006. ICPP 2006. International Conference on*, pp. 539 –550.

Belbeze, C., M. Chevalier, et C. Soulé-Dupuy (2009). Agrégats de mots-clés validés sémantiquement : pour de nouveaux services d'accès à l'information sur Internet. *Document numérique, Documents annotés et langages d'indexation 12*(1), 81–105.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik 1*, 269–271. 10.1007/BF01386390.

Faloutsos, M., P. Faloutsos, et C. Faloutsos (1999). On power-law relationships of the internet topology. *SIGCOMM Comput. Commun. Rev. 29*, 251–262.

Huberman, B. A. et L. A. Adamic (1999). Growth dynamics of the World-Wide Web. *Nature 401*(6749), 131.

Magnien, C., M. Latapy, et M. Habib (2009). Fast computation of empirically tight bounds for the diameter of massive graphs. *J. Exp. Algorithmics 13*, 10 :1.10–10 :1.9.

Najork, M. (2001). Breadth-first search crawling yields high-quality pages. In *WWW '01 : Proc. 10th International World Wide Web Conference*, pp. 114–118.

Tauro, S., C. Palmer, G. Siganos, et M. Faloutsos (2001). A simple conceptual model for the Internet topology.

Travers, J. et S. Milgram (1969). An Experimental Study of the Small World Problem. *Sociometry 32*(4), 425–443.

Wasserman, S. et K. Faust (1994). *Social Network Analysis : Methods and Applications* (1 ed.). Number 8 in Structural analysis in the social sciences. Cambridge University Press.

Watts, D. J. (1999). Networks, dynamics, and the small-world phenomenon. *The American Journal of Sociology 105*(2), 493–527.

Watts, D. J. et S. H. Strogatz (1998). Collective dynamics of small-world networks. *Nature 393*(6684), 440–442.

Summary

To analyze graphs, it is necessary to know their properties so for example identifying their structure and comparing them. One of the significant characterizations of these graphs rests on the fact of determining if it is a "small world" or not. With this intention, the value of the diameter of the graph is essential. However the measurement of the diameter is an extremely long operation for a very large graph. So we propose an algorithm in two phases which makes it possible to quickly obtain an estimation of the diameter of a graph with a small proportion of error. By reducing this algorithm to only one phase and by accepting higher margin of error, we obtain a very fast estimation of the diameter. We test this algorithm on two large real-life graphs (more than one million nodes) and compare its performances with those of an algorithm of reference BFS (Breadth-First Search). The results obtained are described and commented on.