Filtrage, partitionnement et visualisation multi-échelles de graphes d'interactions à partir d'un focus

François Boutin

UNIVERSITE MONTPELLIER II
SCIENCES ET TECHNIQUES DU LANGUEDOC

T H E S E

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE MONTPELLIER II

Discipline : Informatique
Formation Doctorale : Informatique
Ecole Doctorale : Information, Structures, Systèmes

présentée et soutenue publiquement

par

François Boutin

le 28 novembre 2005

_____

Filtrage, partitionnement et visualisation multi-échelles de graphes d'interactions à partir d'un focus

_____

JURY

Mme Marianne Huchard,	Présidente
M. Guy Mélançon,	Directeur de Thèse
Mme Mountaz Hascoët	Codirectrice de Thèse
Mme Tamara Munzner	Rapporteur
Mme Maylis Delest	Rapporteur
Mme Marie-Luce Viaud	Examinateur

Avant-propos et remerciements

Cette thèse est le résultat de rencontres, hasard et travail…

Rentrée 2001, j'assiste par curiosité à un enseignement de programmation objet dispensé par Jacques Ferber aux étudiants du DESS informatique de Montpellier. Fasciné par cette approche nouvelle de la programmation, je demande d'assister en candidat libre à la formation. Enseignant depuis 1995 à l'Université Montpellier 1, je redécouvre ainsi avec joie les bancs de l'Université… en tant qu'étudiant !

C'est là que j'assiste au cours sur les interfaces homme-machine de Mountaz Hascoët. Jeune maître de conférences à l'Université Montpellier 2, et pleine de projets, elle me propose de rencontrer son équipe qui travaille sur des problématiques de visualisation d'information. Elle me présente ainsi Guy Mélançon et son thésard Fabien Jourdan avec qui elle collabore sur un projet de visualisation et d'exploration de grands graphes.

Le groupe de travail m'a d'emblée semblé intéressant : Mountaz venant du monde de la programmation et Guy de l'univers des graphes et de la combinatoire. Rapidement adopté par l'équipe, je participe avec plaisir aux discussions passionnées du vendredi matin, et me laisse insensiblement envahir par le goût de la recherche.

A la rentrée 2002, Mountaz Hascoët me propose un sujet de thèse très riche sur la visualisation multi-échelles de grands réseaux. Elle m'incite à lire beaucoup, à publier et à ne jamais rien considérer comme acquis. Le « métier » de chercheur commence à rentrer, fait de questionnements et de rebondissements.

Rentrée 2003, je m'apprête à être papa, les nuits sont courtes et je fais la connaissance de Maxime Collomb, gourou sympathique de l'informatique, nouveau thésard de Mountaz qui m'apporte une aide technique précieuse.

Rentrée 2004, je fais une nouvelle rencontre décisive : Jérôme Thièvre, autre thésard de Mountaz, avec qui je mène durant une année une collaboration active. Dieu de la visualisation et de l'interaction, il m'aide à concrétiser mes rêves de partitionnement automatique.

Rentrée 2005, aboutissement d'un long travail et impression que toutes les routes se rejoignent enfin ! La boucle est bouclée, les nuits redeviennent très courtes, en attendant d'être à nouveau papa…

Je dois énormément à Mountaz Hascoët pour ce qu'elle m'a apporté durant ces années et notamment pour tout le temps qu'elle m'a accordé chaque semaine ! Elle a su diriger, mes recherches sans jamais m'imposer une voie, acceptant de me guider dans le domaine des graphes qui n'était pas initialement le sien. Elle m'a été d'un soutien précieux lors de la rédaction d'articles, m'incitant à suivre une démarche scientifique rigoureuse. Ses remarques toujours pertinentes et constructives m'ont guidé dans mes travaux, m'obligeant à une perpétuelle remise en question. Lorsque j'étais un peu trop heureux de ma dernière « trouvaille », elle me disait avec justesse que je pouvais encore creuser, que je n'étais qu'au

début… Elle m'a ainsi aidé à réaliser qu'un chercheur est en quête perpétuelle : le bonheur de la « découverte » n'est qu'éphémère. Je voudrais enfin lui faire part de ma grande sympathie.

J'ai également eu un grand plaisir à travailler avec Guy Mélançon. Lors de nombreuses discussions, il m'a sensibilisé aux enjeux et problèmes de l'organisation multi-échelles de graphes. Par son enthousiasme, sa passion communicative et sa grande connaissance du sujet, il a su m'orienter vers de nouvelles pistes dans les périodes de creux. Pour tout cela, je tiens à lui faire part de ma sincère reconnaissance.

Lors du déjeuner de thèse de Fabien Jourdan en décembre 2004, j'ai eu l'occasion de discuter avec Tamara Munzner. Nous avons passé une heure à « graphonner » sur le coin de la nappe tout en mangeant ! J'ai apprécié une personne à la fois sympathique, curieuse scientifiquement et très à l'écoute. Particulièrement intéressé par ses travaux dans le domaine de la visualisation, j'ai été ravi qu'elle accepte de rapporter ma thèse.

J'ai eu l'occasion d'apprécier les articles de Maylis Delest notamment ceux écrits en collaboration avec Guy Mélançon dans le domaine des arbres et graphes. La vision éclairée d'une spécialiste de combinatoire et de graphes me semble très intéressante. Je remercie particulièrement Madame Delest pour le temps consacré à la lecture critique de cette thèse.

Marianne Huchard me fait l'honneur de bien vouloir présider ma thèse. Ayant beaucoup travaillé sur les structures de graphes, elle sera, je pense, intéressée par le sujet traité sur l'organisation et la visualisation de gros graphes.

Je remercie également Marie-Luce Viaud de participer à ce jury. Co-directrice de thèse de Jérôme Thièvre, elle a suivi de près notre collaboration depuis un an. Par ailleurs, de par son travail à l'Institut National de l'Audiovisuel, elle saura évaluer la portée pratique des nouvelles techniques proposées.

La rencontre avec Jérôme Thièvre, thésard à l'INA, fut très riche. Nous avons travaillé par téléphone durant un an sur un projet de visualisation et interaction de graphes partitionnés. Par sa grande maîtrise de la programmation, il a pris en charge la partie visualisation et interaction, alors que je m'occupais des algorithmes de partitionnement et filtrage de graphes. L'essentiel des vues proposées dans cette thèse ont été obtenues avec son API. J'ai aussi beaucoup apprécié sa grande gentillesse : quiconque le connaît sait que je n'exagère pas !

Je tiens à témoigner toute ma sympathie à Fabien Jourdan (anciennement thésard, depuis peu chercheur), Maxime Collomb (en deuxième année de thèse) et tous mes collègues et amis qui m'ont encouragé durant ces années.

Une tendre pensée pour mes parents qui ont consacré leur vie au bonheur de leurs enfants et les ont toujours encouragés dans leurs projets. Je tiens, sans aucun doute, mon goût pour la recherche de maman qui a toujours aimé mots croisés et casses tête mathématiques !

De grosses bises à toute ma famille qui m'a soutenu durant ces années et plus particulièrement à « grand-mère » Monique qui s'est occupée avec beaucoup d'amour de Julie, pendant que « papa » travaillait ! Je suis reconnaissant également à mon frère aîné, Eric, de m'avoir fait partager très tôt ses projets de recherche et encouragé à postuler à l'Université.

Je veux remercier bien sûr ma chère Sophie que j'aime et qui a accepté de sacrifier nombreuses soirées, pour que j'avance dans mon projet de thèse. J'espère ne pas avoir été trop « ailleurs » par moments, on ne fait malheureusement pas de recherche à mi-temps…

Je voudrais enfin embrasser ma petite Julie que j'adore et le bébé que l'on attend avec beaucoup de joie pour la fin de l'année.

Filtre d'amour …

A Sophie, Julie et Amandine

Table des matières

Introduction générale

Les réseaux d'interactions jouent un rôle clé dans de nombreux domaines scientifiques. Il s'agit de structures d'information (graphes) mettant en jeu des acteurs et des relations entre ces acteurs. On les rencontre notamment en informatique (graphe du web, réseau internet), en bibliométrie (graphe de citations, graphe de co-auteurs), en sociologie (graphe des relations personnelles ou professionnelles), en biologie (graphe d'interactions de protéines), en géographie ou économie (réseaux de communications ou d'échanges).

Les réseaux d'interactions réels sont souvent de nature dynamique. Initialement constitués d'un petit noyau d'acteurs, ils croissent à mesure que de nouveaux acteurs sont connectés. Les acteurs ayant déjà plusieurs connections ont généralement davantage de chance d'être connectés par les nouveaux acteurs (principe d'attachement préférentiel). Il a été montré (Barabási et Albert 1999) que ce processus de croissance entraîne une distribution des degrés proche d'une loi de puissance : peu d'acteurs assurent l'essentiel des relations. Les réseaux ayant cette propriété sont dits « sans échelle ». Ils possèdent souvent un noyau très dense interconnectant l'ensemble des acteurs.

Par ailleurs, les réseaux d'interactions issus du monde réel présentent habituellement les deux propriétés « petit monde » (Milgram 1967; Watts et Strogatz 1998) : le diamètre moyen du réseau est faible, et deux acteurs liés ont de fortes chances de partager des relations communes. Les réseaux « petit monde » sont constitués de communautés interconnectées.

Nous nous intéressons, dans cette thèse, aux réseaux « sans échelle » à comportement « petit monde ». Ces réseaux constituent la majorité des réseaux d'interactions réels. Ils possèdent souvent un noyau très dense assurant l'essentiel des relations entre communautés.

Les vues de ces réseaux d'interactions, obtenues avec des outils de dessin classiques, sont souvent surchargées et difficilement analysables. Par ailleurs, ces réseaux se prêtent généralement mal à l'application d'une quelconque technique de partitionnement.

L'objectif de ce travail est de proposer une nouvelle méthode permettant de structurer et visualiser les réseaux « sans échelle » à comportement « petit monde ». Il s'agit de faire apparaître une structure simple et des motifs pertinents quelle que soit la densité du noyau. Nous verrons que les techniques classiques de filtrage de graphes simplifient les réseaux d'interactions sans parvenir à dégager de structure intéressante.

Quelle structure construire pour organiser les réseaux d'interactions ?

L'arbre est une structure naturelle mais « simpliste » pour décrire des graphes « petit monde sans échelle ». Nous proposons de généraliser la structure d'arbre à la structure de graphe « arboré » : il s'agit d'un arbre connectant des communautés de nœuds.

Intuitivement, un graphe « arboré » est facilement visualisable (comme un arbre) par un algorithme de dessin de graphe classique. De plus, les communautés ont des chances d'être clairement séparées comme nous le verrons sur des exemples.

Les modèles « petit monde sans échelle » et « arborés » sont décrits au chapitre 1.

Comment filtrer un graphe « petit monde sans échelle » en un graphe « arboré » ?

Nous proposons, pour cela, l'application d'une nouvelle technique de filtrage d'arêtes. Cette technique extrait un graphe « arboré » du graphe initial en utilisant un focus utilisateur. L'idée est de préserver la connectivité du graphe autour du focus. En revanche, les arêtes « lointaines » et « faibles » auront davantage de risque d'être supprimées.

Cette technique extrait divers graphes « arborés » suivant le focus choisi. Les graphes filtrés sont d'autant plus « fidèles » au graphe traité que l'on est proche du focus. Ce filtre a l'avantage de conserver les propriétés « sans échelle » et « petit monde » du graphe initial.

Divers filtrages sont présentées au chapitre 2. Puis, la nouvelle technique est introduite et agrémentée d'exemples.

Quelle technique utiliser pour organiser un gros graphe « arboré » ?

Pour gérer de gros graphes « arborés », il est souhaitable d'effectuer un partitionnement. Nous proposons une nouvelle technique permettant la structuration d'un graphe « arboré » en un arbre de motifs appelé arbre de silhouettes emboîtées.

L'arbre de silhouettes est construit autour d'un focus éventuellement différent du focus de filtrage, produisant ainsi différentes perspectives du graphe filtré.

Diverses structures et techniques de partitionnement sont décrites au chapitre 3. Puis, la nouvelle technique de partitionnement contextuelle est exposée et illustrée par un exemple.

Quelle technique de visualisation pour les arbres de silhouettes emboîtées ?

La technique doit proposer une organisation arborescente et faciliter la navigation multi-échelles des silhouettes emboîtées. Nous proposons d'intégrer certaines caractéristiques des techniques de visualisation d'arbre et de graphe existantes.

Le chapitre 4 fait le point sur les techniques classiques de visualisation et d'interaction de graphe et présente une technique adaptée à la représentation de graphes d'interactions.

Comment évaluer la qualité du filtrage et du partitionnement ?

Les techniques de filtrage et de partitionnement introduites permettent d'organiser un réseau d'interactions en un « beau » graphe « arboré » puis un « bel » arbre de silhouettes. Encore faut-il s'entendre sur ce que l'on qualifie de « beau ». Par ailleurs, il ne suffit pas que les structures soient « belles », il faut qu'elles aient du sens.

Le chapitre 5 étudie les critères d'évaluation de partitionnement et en propose de nouveaux. Il offre également une évaluation pratique à travers divers cas d'étude ainsi qu'une évaluation théorique des techniques de filtrage et partitionnement proposées.

Nombreuses figures présentées dans cette thèse ont été obtenues avec l'API de dessin de structures multi-échelles de Jérôme Thièvre.

Les différents chapitres peuvent être lus « presque » indépendamment. Pour une aide à la lecture, nous proposons un glossaire des termes nouveaux en fin de document.

« Nous préférerions voir défiler, plutôt que la succession des nœuds ferroviaires, celle des paysages changeants de cheminements discrets, lents ou rapides, de sentiers boueux, de torrents fougueux… Le flot multiple des rivières vers la mer, en quelque sorte. »

Jean Claude Pecker

« L'univers exploré peu à peu expliqué »

Odile Jacob – Sciences

Chapitre 1 Graphes

1.1 Définitions préliminaires

Les réseaux d'interactions ont une structure de graphe. Aussi, nous présentons dans cette section les principales définitions issues de la théorie des graphes (Berge 1970) utiles à la compréhension de ce document. Cette théorie a fait l'objet de nombreux travaux au niveau algorithmique (Gibbons 1985; Evans et Mineka 1992; Jungnickel 1999).

1.1.1 Graphes

Définition 1. Un **graphe orienté** (directed graph) G = (V, E) est défini par un ensemble de sommets (ou nœuds) V = $\{v_i\}$ et un ensemble d'arêtes (appelées aussi arcs ou liens) E = $\{e_k\}$. Tout arc e_k est associé à un couple (ensemble ordonné) de sommets (v_i, v_j) où v_i est le sommet de départ et v_j celui d'arrivée.

Définition 2. Un **graphe non orienté** (undirected graph) G est défini par un ensemble de sommets (ou nœuds) V = $\{v_i\}$ et un ensemble d'arêtes (ou liens) E = $\{e_k\}$. Chaque arête e_k est caractérisée par une paire $\{v_i, v_j\}$ de sommets appelés extrémités. On note G = (V, E).

Définition 3. L'**ordre** d'un graphe fini est le cardinal de E (i.e. le nombre de sommets).

Définition 4. Une **boucle** est une arête liant un même sommet.

Définition 5. Deux arêtes sont dites **parallèles** si elles ont le même sommet de départ et le même sommet d'arrivée (mêmes extrémités pour un graphe ordonné).

Définition 6. Un graphe est dit **simple** s'il ne contient pas de boucle ni d'arêtes parallèles.

Définition 7. Deux sommets liés par une arête sont dits **adjacents**. L'arête est dite **incidente** aux deux sommets.

Définition 8. Le **degré** d'un sommet *v* est défini par le nombre d'arêtes incidentes (chaque boucle compte deux fois). Pour un graphe orienté le **degré entrant** est défini par le nombre d'arcs entrants (i.e. le nombre d'arcs ayant *v* pour arrivée). Le **degré sortant** est le nombre d'arcs sortants de *v*.

Définition 9. Une **feuille** d'un graphe est un sommet de degré 1.

Définition 10. Le **voisinage** d'un sommet est l'ensemble de ses sommets adjacents. Pour un graphe orienté on définit également le **voisinage entrant** et **sortant**.

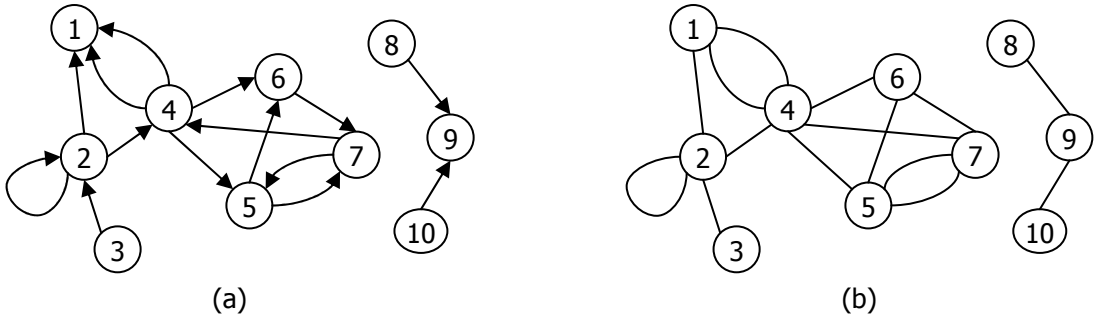


Figure 1 – (a) graphe orienté G, (b) graphe non orienté associé G'

Exemple 1. La Figure 1 présente un graphe orienté G et le graphe non orienté associé G'.

- Les graphes G et G' ont 10 sommets et 15 arêtes.
- Une boucle relie le sommet 2.
- Les arêtes liant les sommets 5 et 7 sont parallèles dans G', non parallèles dans G. Les arêtes liant les sommets 1 et 4 sont parallèles dans G et G'.
- Le degré du sommet 2 est 5. Son degré entrant vaut 2, son degré sortant vaut 3.
- Les sommets 3, 8 et 10 sont des feuilles.

Définition 11. G' = (V', E') est dit graphe **partiel** de G = (V, E) si V' = V et E' $\subseteq$ E.

Définition 12. G' = (V', E') est dit **sous graphe** de G = (V, E) si : V' $\subseteq$ V et E' $\subseteq$ E.

Définition 13. Soit G = (V, E) et V' un sous ensemble de V. Le **graphe induit** G'est défini par G' = (V', E') où E' = $\{\{u, v\} \in E \cap V'^2\}$. G' est un sous graphe de G.

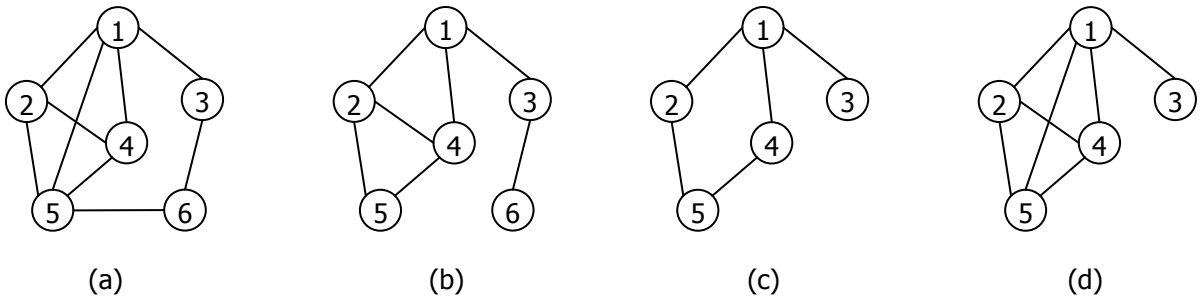


Figure 2 – (a) graphe initial, (b) graphe partiel, (c) sous graphe, (d) graphe induit

Exemple 2. Figure 2 présente (a) un graphe non orienté G, (b) un graphe partiel, (c) un sous graphe, (d) le graphe induit par l'ensemble {1, 2, 3, 4, 5}.

Définition 14. Un graphe est dit **complet** s'il est simple et si deux sommets du graphe sont toujours adjacents (i.e. tous les sommets sont connectés).

Définition 15. Un sous graphe complet est appelé **clique**.

Définition 16. Un graphe est dit **k-régulier** si tous ses sommets ont même degré k.

Définition 17. Un graphe **valué** (arêtes-valuées) est un graphe pour lequel on associe à chaque arête une valeur positive ou nulle appelée **poids**.

Définition 18. Un graphe **sommets-valués** est un graphe pour lequel on associe à chaque sommet une valeur positive ou nulle appelée **poids**.

Définition 19. Un graphe G = (V, E) est dit **biparti** si V est partitionné en deux ensembles V_1, V_2 tels que toute arête de E a une extrémité dans chaque ensemble.

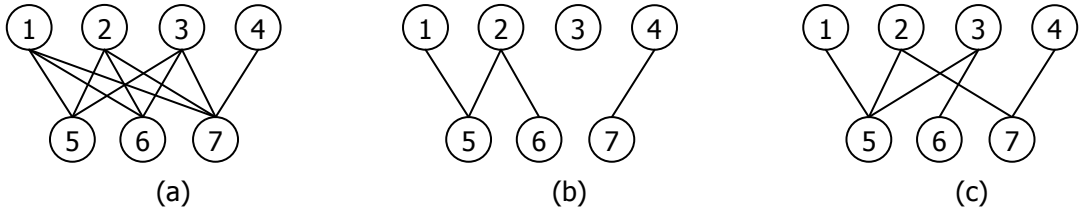


Figure 3 – (a) graphe biparti connexe (b) graphe biparti non connexe (c) arbre biparti

Exemple 3. Trois graphes bipartis sont représentés en Figure 3 : (a) un graphe connexe (voir section 1.1.2), (b) un graphe non connexe, (c) un graphe connexe à n-1 arêtes appelé arbre biparti (voir section 1.1.4).

Définition 20. Deux graphes G = (V, E) et G' = (V', E') sont dits isomorphes s'il existe une bijection f de E dans E' telle que : $(v_1, v_2) \in E \Leftrightarrow (f(v_1), f(v_2)) \in E'$

Définition 21. Un graphe est dit **planaire** s'il existe un dessin du graphe dans le plan tel qu'il n'y ait aucune intersection d'arêtes.

Théorème 1. Tout graphe isomorphe à un graphe planaire est également planaire.

Exemple 4. Le graphe Figure 2 (a) est planaire alors que le graphe Figure 3 (a) ne l'est pas. Les graphes Figure 4 sont isomorphes et planaires. La bijection de G_a vers G_b est l'identité. La bijection de G_b vers G_c est définie par : f(1)=3, f(2)=2, f(3)=4, f(4)=5, f(5)=6, f(6)=1, f(7)=7. Par composition la bijection de G_a vers G_c est f.

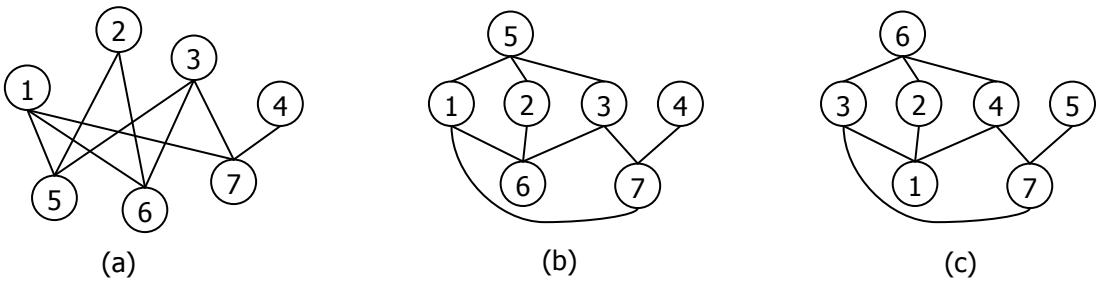


Figure 4 – trois graphes isomorphes : G_a, G_b, G_c

Définition 22. Un graphe non orienté acyclique à n sommets et n-1 arêtes est appelé **arbre** (voir section 1.1.4).

Définition 23. Un graphe orienté acyclique est appelé DAG (directed acyclic graph) (Jungnickel 1999).

Exemple 5. Un DAG est présenté en Figure 5.

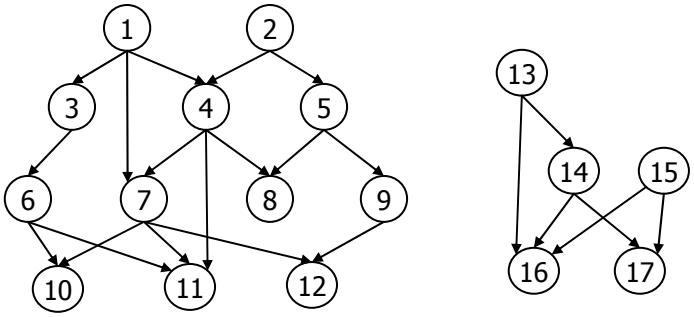


Figure 5 – DAG – graphe orienté acyclique

1.1.2 Connexité

Définition 24. Dans un graphe non orienté, une **chaîne** est constituée d'une suite de sommets adjacents. Elle est dite **simple** si elle ne contient pas deux fois la même arête. Elle est dite **élémentaire** si elle ne contient pas deux fois le même sommet (sauf les extrémités).

Définition 25. Un **cycle** est une chaîne simple fermée d'un graphe non orienté.

Définition 26. Dans un graphe orienté on définit un **chemin** comme une chaîne orientée de sommets, et un **circuit** comme un chemin simple fermé.

Exemple 6. Voir Figure 6, (a, b, c, a, e, d, a) est un cycle ainsi que (j, k, l, m, j)

Définition 27. Soit G un graphe non orienté, la relation « être liés par une chaîne » est une relation d'équivalence dont les classes sont les **composantes connexes** de G.

Définition 28. G est dit **connexe** s'il ne contient qu'une seule composante connexe.

Définition 29. Un sommet v est dit **point d'articulation** si le nombre de composantes connexes de G privé de v et de ses arêtes adjacentes est supérieur strictement au nombre de composantes connexes de G.

Définition 30. Un graphe non orienté connexe sans point d'articulation est dit non **séparable**, ou **inarticulé**.

Définition 31. Une arête e est appelée **isthme** si le nombre de composantes connexes de G \ $\{e\}$ est supérieur strictement au nombre de composantes connexes de G.

Définition 32. Un graphe connexe d'ordre supérieur strictement à k est dit **k-sommets-connexe** (k-connexe) s'il ne perd pas sa connexité en enlevant k-1 sommets.

Définition 33. Un graphe connexe d'ordre supérieur strictement à k est dit **k-arêtes-connexe** s'il ne perd pas sa connexité en enlevant k-1 arêtes.

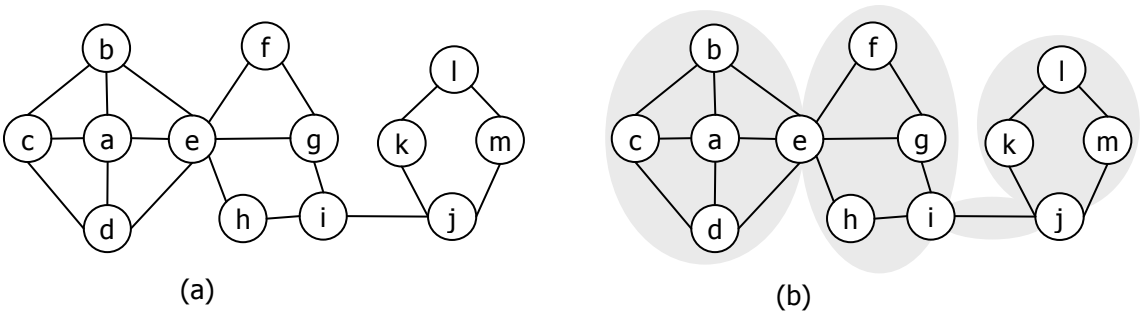


Figure 6 – (a) graphe connexe, (b) composantes 2-connexe, 3-connexe et triviale

Exemple 7. Un graphe connexe est présenté en Figure 6a. Il contient trois points d'articulation : e, i, j et un isthme (i, j). Les nœuds d'articulation séparent quatre composantes (les nœuds d'articulation sont les seuls nœuds pouvant appartenir à plusieurs composantes) :

- une composante triviale : {i, j}
- 2 composantes 2-connexe et 2-arêtes connexe {e, f, g, h, i} et {j, k, l, m}
- une composante 3-connexe et 3-arêtes connexe {a, b, c, d, e}.

Définition 34. Un graphe orienté G est dit **fortement connexe** si pour tout couple de sommets, il existe un chemin d'un sommet à l'autre.

Remarque 1. La relation « être liés par un chemin » n'est pas une relation d'équivalence car elle n'est pas symétrique.

1.1.3 Distances

Nous introduisons les principales distances sur les graphes (Buckley et Harary 1990). Elles vérifient les trois propriétés : symétrie, identité, inégalité triangulaire.

Définition 35. La **distance** entre deux sommets distincts d'un graphe non orienté est définie par la longueur de la plus courte chaîne entre sommets si elle existe, l'infini sinon. Elle vaut zéro si les sommets sont confondus.

Définition 36. Soient V_1 et V_2 deux sous-ensembles disjoints de sommets de G = (V, E). On définit plusieurs distances entre les ensembles V_1 et V_2 :

- La **distance maximale** (*complete distance*) des sous graphes de G : $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ est définie comme la plus grande distance dans G entre les sommets v_1 et v_2 où $v_1 \in V_1$ et $v_2 \in V_2$. Elle vaut l'infini s'il existe deux sommets $v_1 \in V_1$ et $v_2 \in V_2$ non reliés par une chaîne dans G.
- La **distance minimale** (*single distance*) des sous graphes de G : $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ est définie comme la plus petite distance dans G entre les sommets v_1 et v_2 où $v_1 \in V_1$ et $v_2 \in V_2$. Elle vaut l'infini si aucun couple de sommets (v_1, v_2) n'est relié par une chaîne.

- La **distance moyenne** (*average distance*) des sous graphes de G : $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ est définie comme la moyenne des distances dans G entre les sommets v_1 et v_2 où $v_1 \in V_1$ et $v_2 \in V_2$. Elle vaut l'infini s'il existe deux sommets $v_1 \in V_1$ et $v_2 \in V_2$ non reliés par une chaîne dans G.

Définition 37. Le **diamètre maximum** (ou diamètre) d'un graphe est le maximum des distances entre sommets (éventuellement l'infini).

Définition 38. Le **diamètre moyen** d'un graphe est la distance moyenne entre deux sommets quelconques du graphe (éventuellement l'infini).

Remarque 2. Dans le cas d'un graphe orienté, la longueur du plus court chemin ne définit pas une distance.

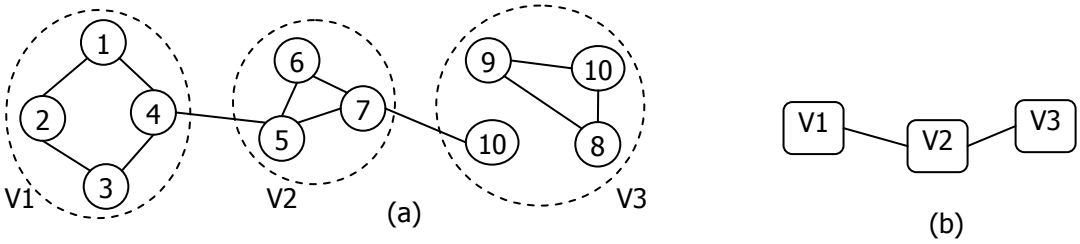


Figure 7 – (a) graphe partitionné (b) graphe quotient associé

Exemple 8. En Figure 7, on considère un graphe G = (V, E) et trois sous ensembles V_1, V_2, V_3 de V qui induisent trois sous graphes G_1, G_2 et G_3. Le graphe G_1 a pour diamètre maximal 2 et pour diamètre moyen 4/3. G_3 a pour diamètre moyen et maximal l'infini car G_3 n'est pas connexe. La distance minimale de G_1 à G_2 vaut 1, la distance maximale vaut 4, la distance moyenne vaut 8/3. La distance minimale de G_1 à G_3 vaut 3, les distances moyenne et maximale valent l'infini.

1.1.4 Arbres

Définition 39. Un **arbre** est défini par un graphe non orienté connexe acyclique. Cette définition est équivalente à celle présentée précédemment (Définition 22).

Définition 40. Une **forêt** est un graphe non orienté acyclique. Ses composantes connexes sont des arbres.

Définition 41. Une **arborescence** (*rooted tree*) est définie par un arbre ayant un sommet particulier appelé **racine**.

Définition 42. Dans une arborescence la **profondeur** d'un sommet est définie par sa distance à la racine.

Définition 43. La **hauteur d'une arborescence** (ou profondeur) est définie par la longueur de la plus longue chaîne simple partant de la racine.

Définition 44. La **hauteur d'un sommet** est définie par la hauteur de l'arborescence ayant pour racine le sommet (obtenue en « coupant » l'arbre au niveau du sommet).

Définition 45. Soit une arborescence de racine *r* et deux sommets *v* et *v'*. On dit que *v* a pour **descendant** *v'* si *v* appartient au plus court chemin de *r* à *v'*. Dans ce cas *v* est dit **ascendant** de *v'*. Si de plus, *v* et *v'* sont adjacents, *v'* est appelé **fils** de *v*, et *v* **père** de *v'*.

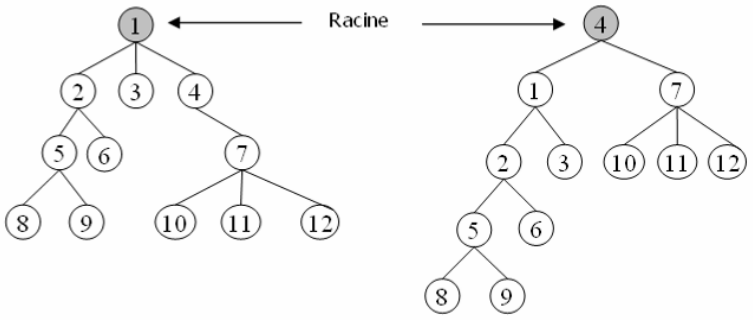


Figure 8 – deux arborescences T_1 et T_4 d'un même arbre T

Exemple 9. Figure 8, présente deux arborescences d'un même arbre.

- T_1 a pour hauteur 3. T_4 a pour hauteur 4.
- Le sommet 7 a pour profondeur 2 dans T_1 et 1 dans T_4.
- Le sommet 4 a pour hauteur 2 dans T_1 et 4 dans T_4.
- Dans T_4, le sommet 2 a pour descendants 5, 6, 8 et 9 et pour ascendants 1 et 4. Il a pour père le sommet 1 et pour fils les sommets 5 et 6.

Définition 46. Un graphe biparti connexe à n sommets et n-1 arêtes est appelé **arbre biparti** (voir Figure 3c).

Définition 47. Un arbre **couvrant** de G = (V, E) est un arbre construit à partir d'un ensemble d'arêtes de E qui relient tous les sommets de V.

Propriété 1. Tout graphe connexe possède un arbre couvrant.

Définition 48. Un **arbre couvrant minimal** d'un graphe valué est un arbre couvrant qui minimise la somme des poids des arêtes.

1.2 Modèles de graphes

Nous présentons (section 1.2.1) l'un des premiers modèle de graphe étudié : le modèle de graphe aléatoire (Erdös et Rényi 1959). D'autres modèles ont été introduits récemment pour tenter de modéliser les graphes du monde réel (graphes sociaux, biologiques, web…) : il s'agit des modèles « petit monde » (section 1.2.2) et « sans échelle » (section 1.2.3).

Pour mieux modéliser les graphes d'interactions réels, nous introduisons deux modèles : « petit monde sans échelle » (section 1.2.4) et « petit monde arboré » (section 1.2.5).

1.2.1 Graphes aléatoires

Le modèle de graphe aléatoire a été largement étudié (Erdös et Rényi 1959), révélant diverses propriétés. Deux écritures du modèle ont été proposées :

Définition 49. **Modèle G (n, m)** : un graphe aléatoire G = (V, E) à n sommets et m arêtes est caractérisé par la donnée de m couples aléatoires de sommets (v_i, v_j).

Définition 50. **Modèle G (n, p)** : un graphe aléatoire G = (V, E) de probabilité p est caractérisé par n sommets et un ensemble d'arêtes E contenant chaque couple de sommets (v_i, v_j) avec une probabilité p. Si n est l'ordre du graphe, on définir $\lambda = pn$.

Propriété 2. La distribution des degrés peut être approchée par une loi de Poisson de paramètre λ. Soit k un entier positif ou nul, la probabilité que le degré d'un nœud égale k vaut : $P_k = e^{-\lambda} \frac{\lambda^k}{k!}$ (voir Figure 9).

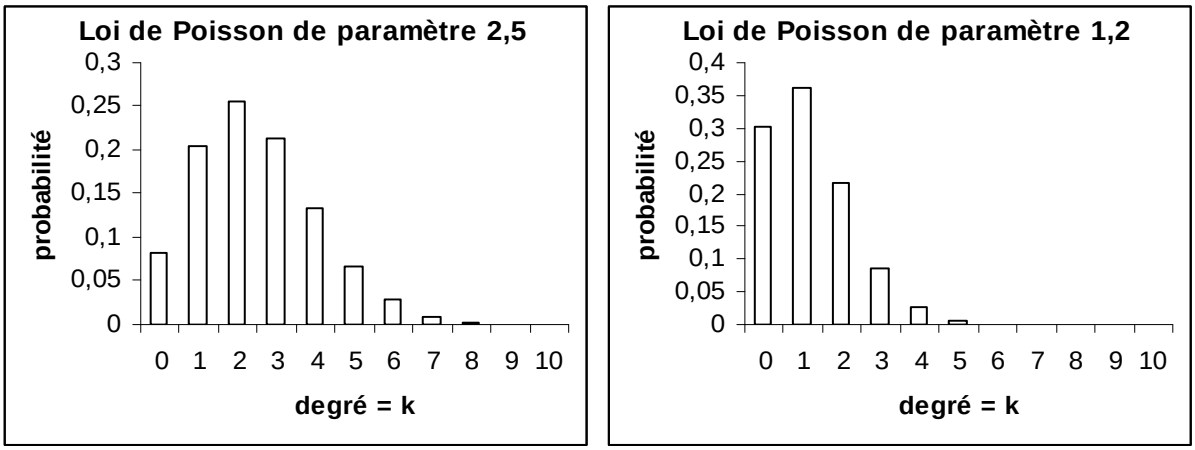


Figure 9 – lois de Poisson de paramètre $\lambda = 2,5$ et $\lambda = 1,2$

Propriété 3. Divers travaux (Erdös et Rényi 1961; Bollobás 1985; Janson, Luczak et al. 1999; Albert et Barabási 2002) ont étudié l'apparition de caractéristiques du graphe selon la valeur de λ :

- La présence d'une composante connexe géante (de taille > n/2) si $\lambda > 0,5$.
- La connexité du graphe si $\lambda > \log(n) / 2$.
- L'apparition de cliques de degré k si $\lambda > \frac{n^{(1-\frac{2}{k-1})}}{2}$ (triangles si $\lambda > ½$).

Exemple 10. Quatre graphes aléatoires sont présentés en Figure 10. Pour λ supérieur ou égal à 0,7 on observe bien une composante connexe principale.

Le modèle de graphe aléatoire est dit « démocratique » dans le sens où la probabilité de connexion d'un nœud est la même pour tous. Généralement, ce n'est pas le cas des graphes issus du monde réel (section 1.3) : deux nœuds liés ont souvent de grandes chances d'être liés à un voisin commun. Par ailleurs, certains nœuds (hubs) possèdent de très nombreuses connexions. Les modèles présentés section 1.2.2 et 1.2.3 prennent en compte ces propriétés.

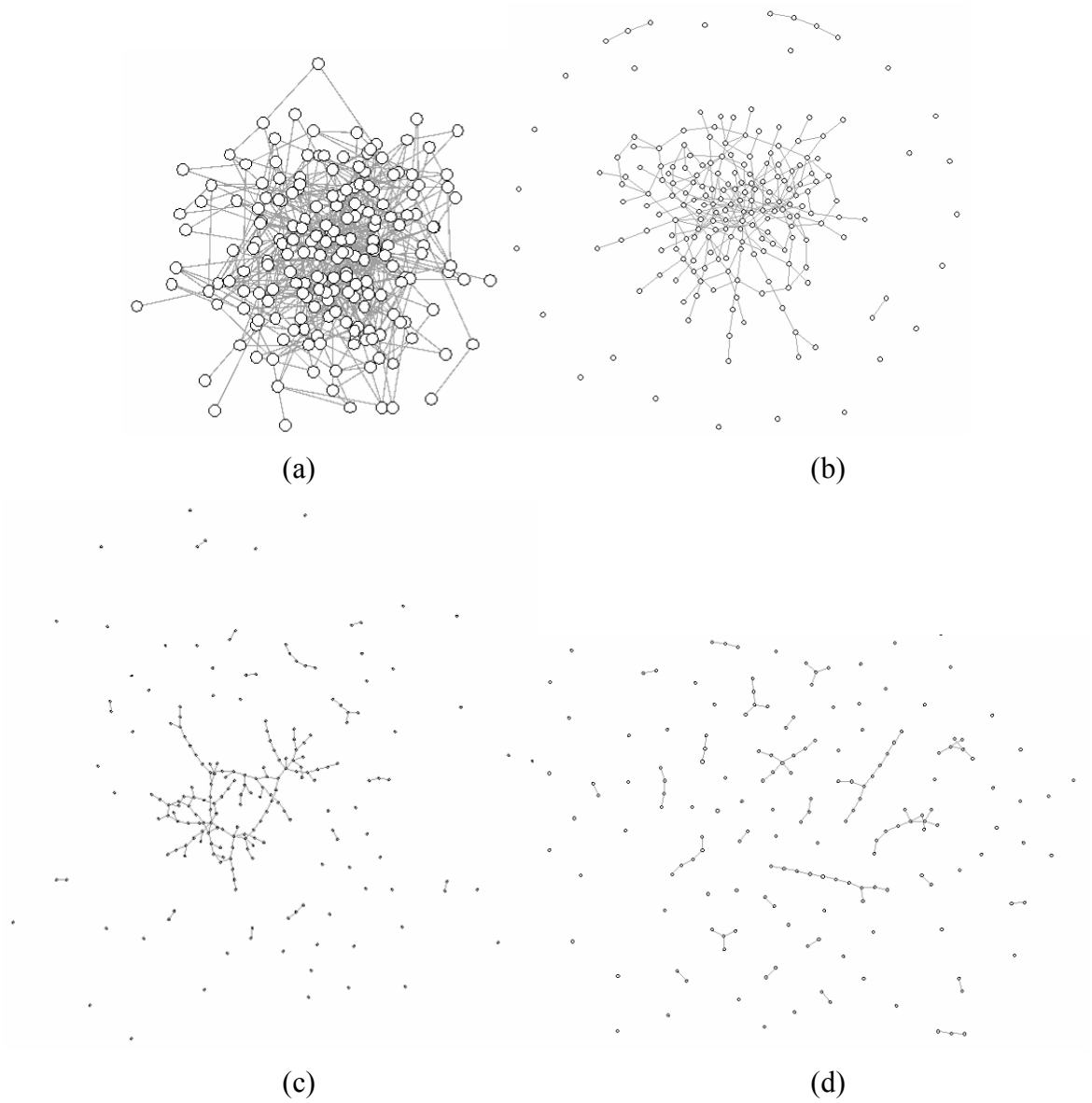


Figure 10 – graphe aléatoire à 200 noeuds (a) λ = 2,5 (b) λ= 1,2 (c) λ= 0,7 (d) λ= 0,4

1.2.2 Graphes « petit monde »

1.2.2.1 Définitions

Le phénomène « petit monde » a été mis en évidence par un psychologue (Milgram 1967), constatant expérimentalement que deux individus quelconques aux USA sont liés en moyenne par une chaîne de connaissances de longueur six. Les travaux théoriques de Pool et Kochen ont confirmés ces résultats empiriques (Pool et Kochen 1978; Kochen 1989).

Watts et Strogatz constatent que ce phénomène « petit monde » ne se limite pas aux réseaux sociaux (Watts et Strogatz 1998; Watts 1999). Par ailleurs ils montrent qu'il s'accompagne le plus souvent d'une autre propriété : deux nœuds ayant un voisin commun ont plus de chance d'être connectés que deux nœuds pris au hasard. En d'autres termes, deux nœuds adjacents d'un graphe « petit monde » ont davantage de chances de partager des

voisins communs que s'ils appartenaient à un graphe aléatoire de même taille (même nombre de nœuds et d'arêtes). Cette caractéristique, correspondant à l'adage populaire « les amis de mes amis sont mes amis ». Le coefficient de clustering permet de préciser cette caractéristique :

Définition 51. Soit G un graphe non orienté et v un sommet du graphe. Si on note N(v) l'ensemble des k voisins de v et m le nombre de liens entre ces k sommets, Watts et Strogatz définissent le **coefficient de clustering local** de v par le rapport du nombre d'arêtes reliant ses voisins sur le nombre maximum d'arêtes possibles : $C(v) = \frac{2m}{k(k-1)}$. Le **coefficient de clustering** C du graphe est la moyenne des coefficients de clustering locaux.

Exemple 11. La Figure 11 présente le voisinage de deux sommets :

- (a) le coefficient de clustering vaut $C(v)=\frac{2\times7}{6\times5}=\frac{14}{30}=47\%$
- (b) le coefficient de clustering vaut $C(v)=\frac{2\times3}{6\times5}=\frac{6}{30}=20\%$

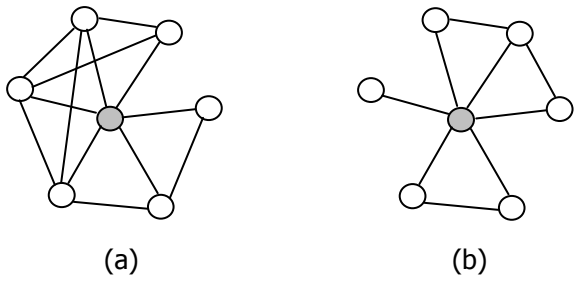


Figure 11 – coefficient de clustering

Propriété 4. Coefficient de clustering ainsi que diamètre d'un graphe complet valent 1.

Watts et Strogatz (Watts et Strogatz 1998) ont proposé une définition formelle des graphes petits monde basée sur l'utilisation conjointe du diamètre moyen L et du coefficient de clustering C :

Définition 52. Un graphe connexe est dit « petit monde » s'il possède les deux propriétés :

- Diamètre moyen L « faible » (en log(N) comme dans un graphe aléatoire, N étant le nombre de nœuds).
- Coefficient de clustering C « grand » (supérieur à celui d'un graphe aléatoire de même taille : même nombre de nœuds et même nombre d'arêtes).

1.2.2.2 Construction de Watts et Strogatz

Watts et Strogatz proposent une technique de construction de graphe « petit monde ». Pour construire un graphe à n nœuds et nk arêtes, n nœuds sont placés de façon homogène sur un cercle. Chaque nœud est alors lié à ses k plus proches voisins formant ainsi un graphe k-

régulier à une dimension (Figure 12.a). Chaque arête a ensuite une probabilité p d'être déconnectée à l'une de ses extrémités et reconnectée à un sommet aléatoire (Figure 12.b). A la limite lorsque p vaut 1, toutes les arêtes sont remplacées aléatoirement et on obtient un graphe aléatoire (Figure 12.c).

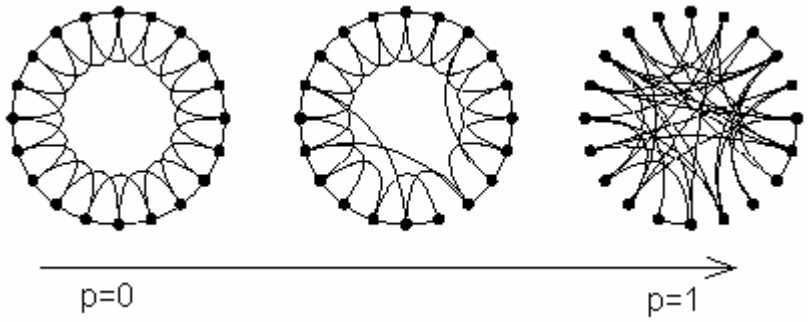


Figure 12 – (a) graphe 4-régulier (b) graphe « petit monde » (c) graphe aléatoire

Propriété 5. Le modèle « petit monde » de Watts et Strogatz est l'intermédiaire entre graphe régulier et graphe aléatoire. Il a un fort coefficient de clustering (comme un graphe régulier) et un diamètre faible (comme un graphe aléatoire).

Les variations du coefficient de clustering C et du diamètre moyen L en fonction de la probabilité p de remplacement d'une arête ont étudiés (Watts et Strogatz 1998). Il suffit de remplacer quelques arêtes dans un graphe régulier pour diminuer drastiquement le diamètre moyen L entre sommets tout en préservant le coefficient de clustering C.

Exemple 12. Figure 13, pour un graphe de 1000 nœuds, de degré moyen 10, la valeur seuil avoisine p = 1 %. En effet, pour cette valeur, le rapport C(p)/C(0) reste élevé (proche de 95%) et le rapport L(p)/L(0) est faible (proche de 20%).

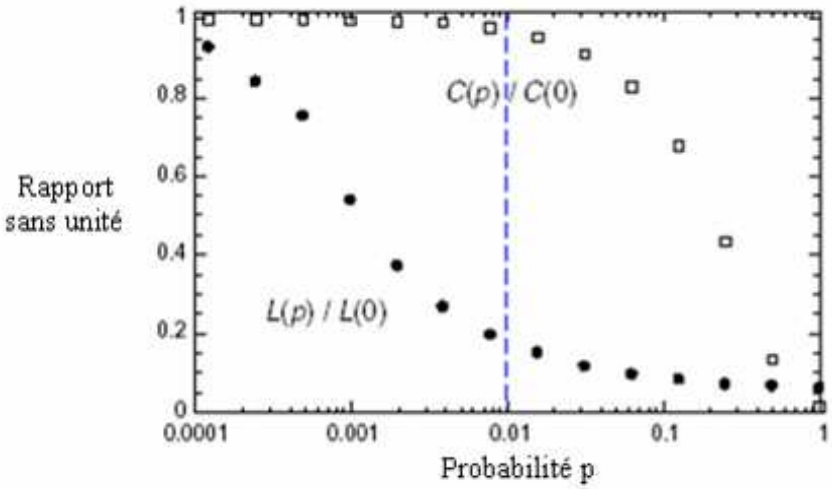


Figure 13 – variations de C et de L en fonction de la probabilité p

Le modèle de graphe de Watts et Strogatz (comme celui d'Erdös et Renyi) possède un diamètre logarithmique.

Contrairement au modèle de Kleinberg (section 1.2.2.3), le modèle de Watts et Strogatz est non navigable. C'est-à-dire que l'on ne peut pas trouver facilement de chemin court entre deux sommets sans connaissance globale du graphe.

1.2.2.3 Construction de Kleinberg

Kleinberg a proposé la construction d'un modèle de graphe « petit monde » navigable à l'aide d'un algorithme décentralisé. Ce modèle permet de lier deux nœuds en temps optimal avec la seule connaissance d'un voisinage local des nœuds et de quelques voisins lointains.

Partant de n sommets disposés sur une grille 2D, il associe à un sommet donné un ensemble de liens de courte portée (short ranges) et des liens de longue portée (long ranges).

Définition 53. Le modèle « petit monde » (Kleinberg 1999) est défini par p, q et r :

- Chaque sommet de la grille est connecté aux sommets de son p-voisinage (sommets à distance de Manhattan inférieure ou égale à p dans la grille). Il s'agit des liens de courte portée.
- Pour chaque sommet, q liens de longue portée sont choisis dans la grille.
- La probabilité d'existence d'une arête de longueur d est proportionnelle à d^{-r}.

Exemple 13. Le principe est expliqué en Figure 14, pour p = 2 et q = 3. Considérant un sommet v (en gris), les liens de courte portée sont matérialisés par des flèches pleines, les liens de longue portée par des flèches en pointillés.

Propriété 6. Kleinberg étudie pour quelle valeur de r, le routage est optimal (considérant p et q fixés) :

- Lorsque r = 0, la distribution des cibles des liens de longue portée est uniforme, comme dans le modèle proposé en section 1.2.2.2 (Watts et Strogatz 1998).
- Lorsque r augmente, on a davantage de liens de longue portée « proches » du sommet considéré.
- Pour r = 2, le routage est optimal c'est-à-dire le trajet moyen entre deux nœuds est minimal.

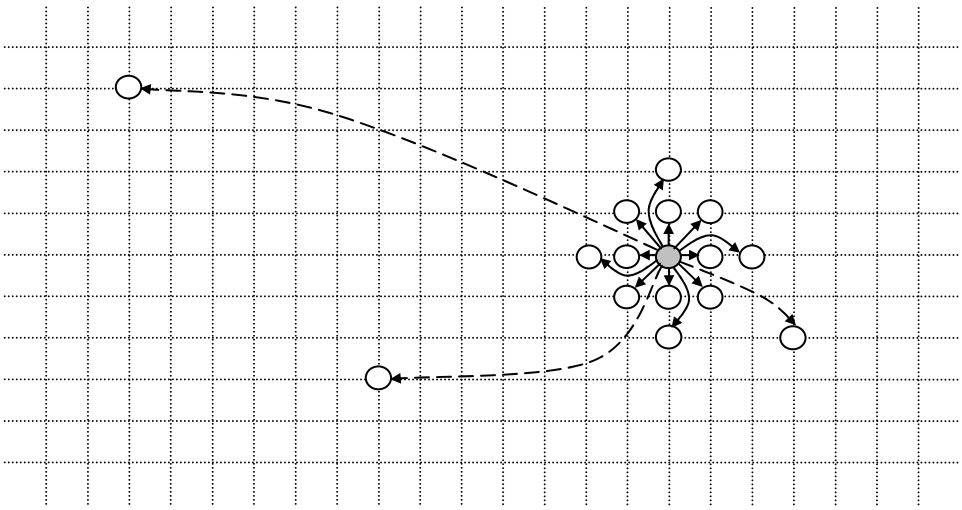


Figure 14 – liens de courte et longue portées d'un sommet

Les modèles de graphes « petit monde » proposés par Watts et Strogatz d'une part et Kleinberg d'autre part ont une distribution des degrés proche d'une loi Normale (Gaussienne). Ce résultat vient de ce que plusieurs arêtes sont déconnectées et reconnectées aléatoirement.

Or, nombreux graphes réels sont « petit monde » tout en ayant une distribution des degrés suivant une loi de puissance et non une loi Normale (section 1.3). Ils appartiennent ainsi à la catégorie des graphes « sans échelle » (section 1.2.3). Les modèles de Watts-Strogatz et Kleinberg ne permettent pas de simuler de tels graphes.

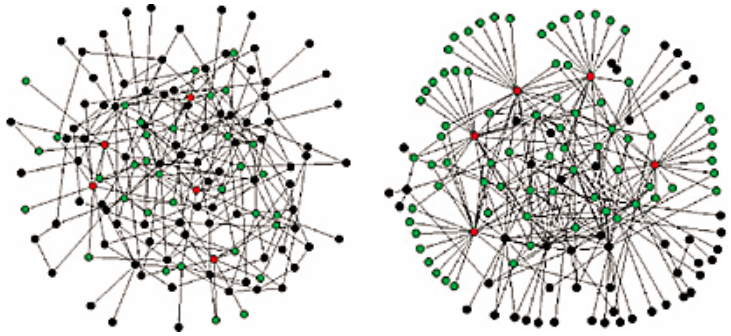
1.2.3 Graphes « sans échelle »

Dans les graphes issus du monde réel, la distribution des degrés suit rarement une loi de Poisson (voir graphes aléatoires d'Erdös et Renyi) ou une loi Normale (voir graphes « petit monde » de Watts-Strogatz ou Kleinberg), mais une loi de puissance (appelée loi de Pareto) : la probabilité qu'un nœud soit de degré k vaut $P_k = a.k^{-\gamma}$. De tels graphes sont dits « sans échelle » ou graphes de puissance (Barabási et Albert 1999; Albert et Barabási 2002).

Propriété 7. En passant au log on obtient : $\log(P_k) = \log(a) - \gamma \log(k)$. C'est l'équation d'une droite dont la pente est le coefficient γ de la loi de puissance.

Par définition, dans un graphe « sans échelle » la grande majorité des sommets a un degré faible alors qu'un petit nombre de sommets présente un degré élevé. La notion de degré moyen n'a pas d'intérêt d'où la terminologie de « sans échelle ».

Contrairement au modèle de graphe aléatoire proposé par (Erdös et Rényi 1959) dit « démocratique », le modèle de graphe « sans échelle » répond à une logique de type « Monopoly » : quelques nœuds se partagent l'essentiel des connexions (« rich get richer »).



(a) (b)

Figure 15 – (a) graphe aléatoire (b) graphe « sans échelle » (Albert, Jeong et al. 2000)

Exemple 14. La Figure 15, présente un graphe « sans échelle » et un graphe aléatoire de même taille (Barabási et Albert 1999). Sont représentés en rouge les 5 nœuds d'ordre maximum. Dans le graphe aléatoire ils sont liés à 27 % du graphe, contre 60 % dans le graphe « sans échelle ».

Exemple 15. La Figure 16 présente la distribution des degrés à deux échelles : l'échelle linéaire et l'échelle log log. Elle suit approximativement une loi de puissance. L'approximation est bonne puisque le coefficient de détermination R^2 est très proche de 1. La pente de la droite de régression (Figure 16b) correspond au paramètre $\gamma = 2{,}372$ de la loi de puissance.

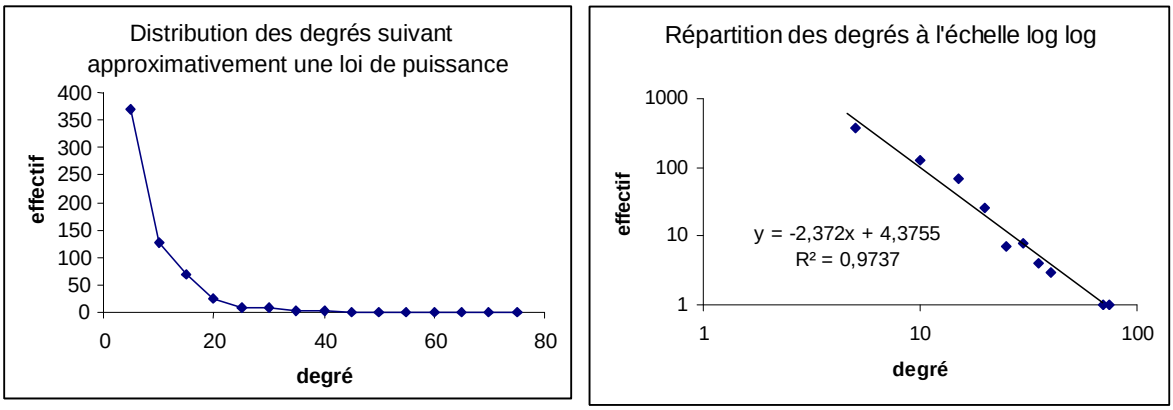


Figure 16 – distribution proche d'une loi de puissance (a) échelle linéaire (b) log log

Construction : Un graphe « sans échelle » peut être construit en simulant la **croissance** d'un graphe (Albert et Barabási 2002) : chaque nouveau nœud est connecté à m sommets du graphe existant en privilégiant les sommets de degrés importants (hubs). Ce principe d'**attachement préférentiel** consiste à associer à chaque nœud du graphe une probabilité de connexion proportionnelle à son degré. Ainsi, la distribution des degrés suit une loi de puissance (Albert et Barabási 2002). D'autres modèles de graphes « sans échelle » ont été proposés (Aiello et Chung 2001).

On retrouve les deux principes de construction : croissance et attachement préférentiel dans divers graphes sociaux dynamiques qui sont typiquement des graphes « sans échelle ».

Divers travaux (Albert, Jeong et al. 2000; Newman 2002) se sont intéressés au problème du routage dans un réseau « sans échelle » :

Propriété 8. Un graphe « sans échelle » est résistant aux pannes (comme un réseau aléatoire) mais vulnérable aux attaques de sommets à fort degré (hubs).

Une classification empirique des graphes « sans échelle » en fonction de la valeur γ est proposée dans (Goh, Oh et al. 2002). Lorsque $\gamma = 1$, le diamètre est en log n. Lorsque $\gamma \geq 2$ la distance entre deux sommets est en log n / log log n (Bollobás et Riordan 2004).

1.2.4 Nouveau modèle : graphe « petit monde sans échelle »

Nous verrons en section 1.3 que de nombreux réseaux d'interactions réels sont « sans échelle » à comportement « petit monde ». C'est le cas notamment du graphe d'interactions de protéines (Figure 17) étudié par (Jeong, Mason et al. 2001).

Or, le modèle de graphe « sans échelle » de (Albert et Barabási 2002) favorise l'attachement préférentiel aux hubs de fort degré mais il n'assure pas la formation de communautés. Ce modèle, tel quel, ne permet donc pas d'obtenir un graphe « petit monde ».

De plus, les graphes « petit monde » de Watts-Strogatz d'une part et Kleinberg d'autre part ne sont pas « sans échelle » car la distribution des degrés ne suit pas une loi de puissance.

Nous proposons dans cette section un nouveau modèle de graphe ayant à la fois les deux propriétés : « sans échelle » et « petit monde » :

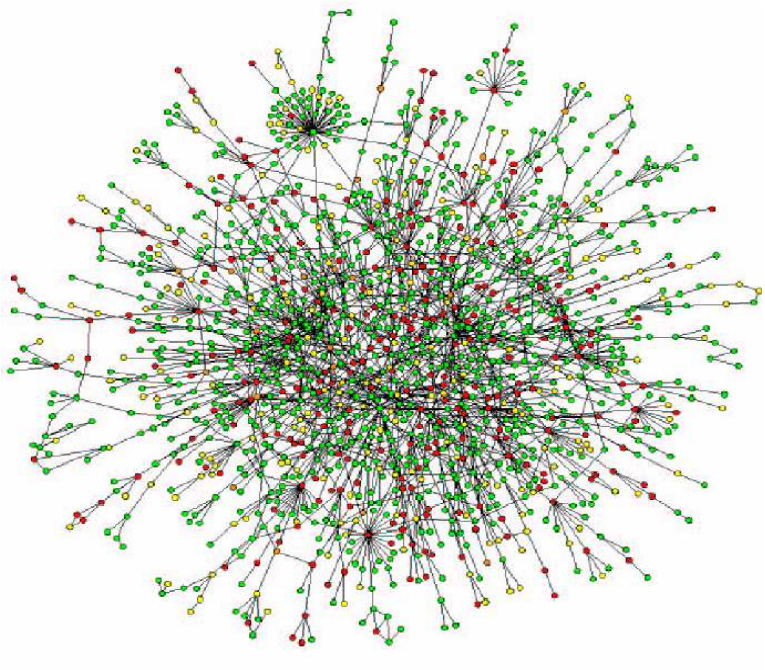


Figure 17 – graphe de « YEAST » (Jeong, Mason et al. 2001)

L'idée est de construire un graphe dynamique s'inspirant de (Albert et Barabási 2002) :

- Chaque nouveau nœud v est lié avec attachement préférentiel à m(v) nœuds existants.
- Puis v est lié avec attachement préférentiel à m'(v) nœuds adjacents des m(v) nœuds.

Un tel modèle prend en compte le caractère « sans échelle » car les liens sont toujours choisis avec attachement préférentiel. D'autre part, le fait de connecter les voisins de voisins permet de former des communautés assurant ainsi le comportement « petit monde ».

Le graphe « sans échelle » à comportement « petit monde » ainsi construit a des propriétés dépendant du choix des fonctions m(v) et m'(v). Dans un modèle simplifié on peut considérer des constantes : m(v) = m et m'(v) = m'.

Propriété 9. Lorsque m(v) est petit par rapport à m'(v), le graphe est « petit monde » avec un coefficient de clustering élevé.

Propriété 10. Pour obtenir un coefficient de clustering nul, il suffit de considérer m' nul et d'interdire à chaque nouveau nœud d'être lié à un voisin d'un de ses voisins.

Nous avons montré dans cette section, qu'il est possible de concevoir un modèle alliant à la fois les deux propriétés « sans échelle » et « petit monde ». D'autres travaux pourraient être menés pour étudier plus en détail ce modèle et voir son adéquation avec des graphes réels. Nous nous intéresserons davantage par la suite à des techniques de simplification et de visualisation des graphes « sans échelle » à comportement « petit monde ».

1.2.5 Nouveau modèle : graphe « arboré »

Propriété 11. Certains réseaux d'interactions ont l'apparence d'un arbre auquel ont été ajoutées des arêtes « courtes » entre nœuds « proches ». Il s'agit par exemple du graphe, Figure 18a (section 1.3). Nous les qualifions « d'arborés ».

Pour savoir si un graphe est « arboré », on doit extraire l'arbre couvrant « maximal » :

Construction : L'arbre couvrant « maximal » T du graphe est construit itérativement :

- L'arbre T_0 est constitué du focus (à défaut du nœud de plus fort degré).
- L'arbre T_n est créé en liant T_{n-1} à son voisin de plus fort degré.

On définit la longueur d'une arête comme la distance (dans l'arbre) entre ses extrémités.

Définition 54. Soit D le diamètre moyen de l'arbre couvrant « maximal ». Un réseau est dit « arboré » si les deux conditions suivantes sont réalisées :

- La longueur moyenne des arêtes (dans l'arbre) est petite devant D.
- Les arêtes ont *presque sûrement* une longueur inférieure à D

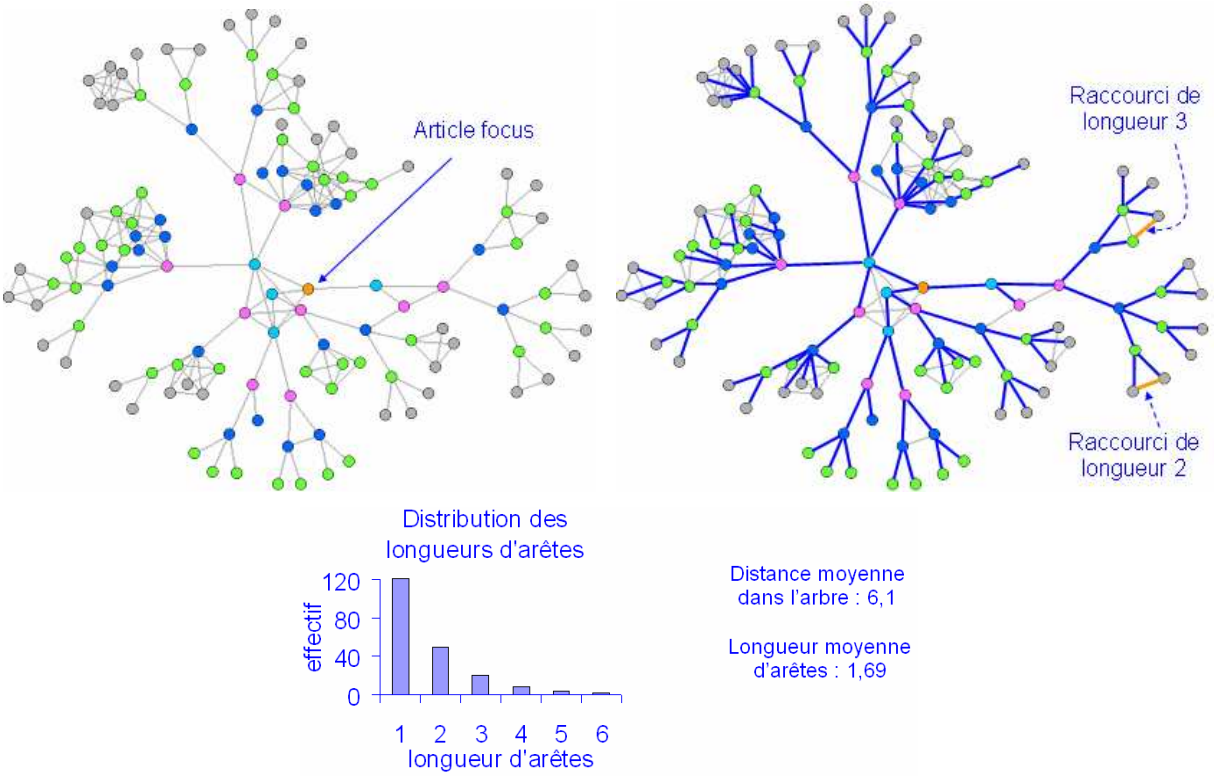


Figure 18 – (a) graphe « arboré » (b) arbre couvrant « maximal » (c) longueur des raccourcis

Exemple 16. En Figure 18c, la longueur moyenne des arêtes vaut 1,69 ce qui est faible devant le diamètre moyen du graphe qui vaut 6,1. De plus, toutes les arêtes ont une longueur inférieure au diamètre moyen, donc le graphe est « arboré ».

Modèle : Pour construire un graphe « arboré », il suffit de partir d'un arbre et d'ajouter des raccourcis dont la longueur est faible par rapport au diamètre moyen (voir section 5.3.7).

Nous verrons que les graphes « arborés » sont aisément visualisables et partitionnables (sections 3.6 et 4.3). Nous proposerons de filtrer les graphes « petit monde sans échelle » (difficilement visualisables et partitionnables) en graphes « arborés » (section 2.5).

1.3 Réseaux issus du monde réel

Nous présentons dans ce chapitre différents réseaux issus du monde réel : graphe de similarités (CiteSeer), graphe de co-auteurs (LIRMM), graphe de citations (InfoVis Contest), graphe des relations (amitiés et sympathies), graphe d'interactions de protéines.

Il s'agit de constituer un recueil de réseaux d'interactions réels permettant de tester nos différents algorithmes. Nous étudions leurs caractéristiques : distribution des degrés, liaison entre degré et coefficient de clustering d'un nœud.

Nous proposons une vue de ces graphes en utilisant un algorithme de dessin basé sur un modèle de forces (Eades 1984; Fruchterman et Reingold 1991; Walshaw 2000). Nous présentons le principe de l'algorithme en section 3.2.1.3. D'autres techniques peuvent être utilisées : elles sont recensées en section 4.1.

Certains graphes « petit monde » réels (peu en réalité – voir en section 1.3.1) sont « arborés » au sens défini en section 1.2.5 : ils s'apparentent à un arbre auquel ont été ajoutées des arêtes entre nœuds « proches ». Des vues « agréables » (au sens précisé en section 5.5.2) sont obtenues avec un algorithme de dessin à base de forces.

Les autres graphes étudiés sont « sans échelle » à comportement « petit monde ». Ils présentent un noyau très dense. Les algorithmes de dessin et de partitionnement (clustering) classiques sont difficilement exploitables. Nous introduisons en section 2.5 un nouvel algorithme de filtrage d'arêtes, transformant tout graphe connexe en un graphe « arboré » facilement visualisable et clusterisable.

1.3.1 Réseaux « petit monde arborés »

Nous proposons l'étude de graphes bibliographiques extraits de la bibliothèque numérique scientifique *ResearchIndex* (CiteSeer; Lee Giles, Bollacker et al. 1998; Lawrence, Bollacker et al. 1999). Différents articles, en ligne sur le web, sont organisés. Des liens entre articles sont établis en utilisant citations, co-citations ou similarités (Chen 1999).

Le graphe de citations de *CiteSeer* a été étudié expérimentalement (An, Janssen et al. 2002) avec quatre sous-graphes de 20 000 nœuds (en considérant des liens non orientés). De diamètre proche de 18, ils sont tous les quatre « sans échelle » ($\gamma \approx 1{,}7$) à comportement « petit monde ». De plus 90 % des articles forment une énorme composante connexe. Parmi ceux-ci 68,5 % n'ont pas été cités et 58 % forment une composante biconnexe géante.

Nous proposons ici l'étude de trois sous graphes de similarité de CiteSeer. Un premier graphe de taille moyenne (section 1.3.1.1), un second de taille réduite (section 1.3.1.3), enfin un dernier de taille importante (section 1.3.1.3). Ces graphes ont la particularité d'être plus ou moins « arborés ».

Nous analyserons ensuite en section 1.3.1.4 un autre graphe « arboré » : le graphe des amitiés entre étudiants d'une même promotion.

Nous discuterons de la propriété « arboré » en section 1.3.1.5.

1.3.1.1 Graphe de CiteSeer de taille moyenne

Le graphe présenté en Figure 19 contient 329 nœuds et 560 liens : un nœud correspond à un document bibliographique de CiteSeer. Un lien est établi entre documents s'ils sont liés (« related ») au sens de CiteSeer c'est-à-dire s'ils partagent des mots clés et/ou des citations.

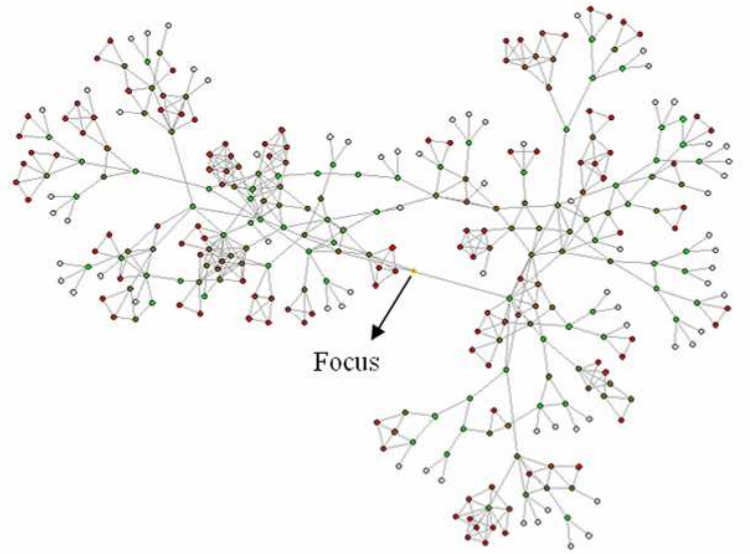


Figure 19 – vue générale du graphe de similarités issu de CiteSeer

Construction : partant d'un document focus « *Tree visualisation and navigation clues for information visualisation* » (Herman, Delest et al. 1990), nous utilisons le parseur *HTMLParser* afin de récupérer une première génération de « related documents ». Puis nous réitérons le processus pour récupérer une seconde génération de « related documents » associés à la première génération de documents. Par un parcours en profondeur, nous récupérons ainsi plusieurs générations de « related documents ».

Nous représentons en rouge en Figure 19 les nœuds à fort coefficient de clustering.

A chaque génération, la taille du graphe augmente de façon exponentielle. Le nombre moyen de « related documents » étant de 3,4 nous nous limitons à quelques générations.

La distribution des degrés en Figure 20 est proche de la loi de Poisson (si l'on fait abstraction des nœuds de degré 1 appartenant essentiellement à la dernière génération). Il ne s'agit donc pas d'un graphe « sans échelle ».

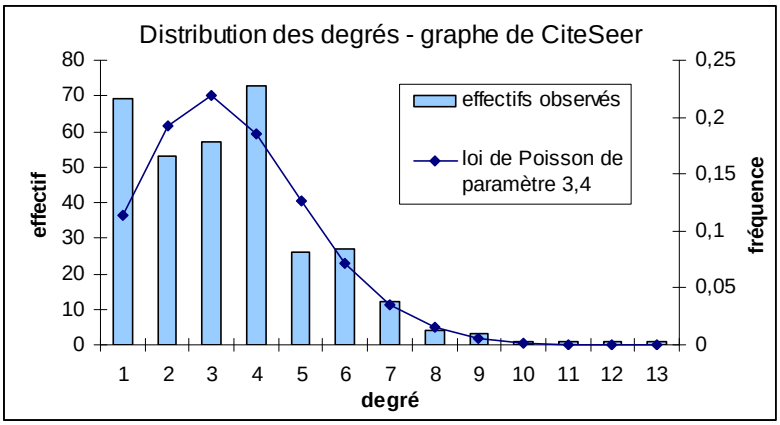


Figure 20 – distribution des degrés – graphe de similarités de CiteSeer

Le coefficient de clustering est important. Il vaut 0,4 alors qu'il vaudrait environ 0,02 avec un graphe aléatoire de même taille (simulation). C'est donc un graphe « petit monde ».

Le nuage de points de la Figure 21 révèle que les sommets de degré important ont un coefficient de clustering inférieur à la moyenne. Ce résultat s'explique par la définition même du coefficient de clustering : dans un graphe peu dense, si on considère un nœud de degré important (noté n), le nombre d'inter connexions de ses n voisins a de grandes chances d'être faible devant n(n-1)/2. Il en résulte un coefficient de clustering local faible.

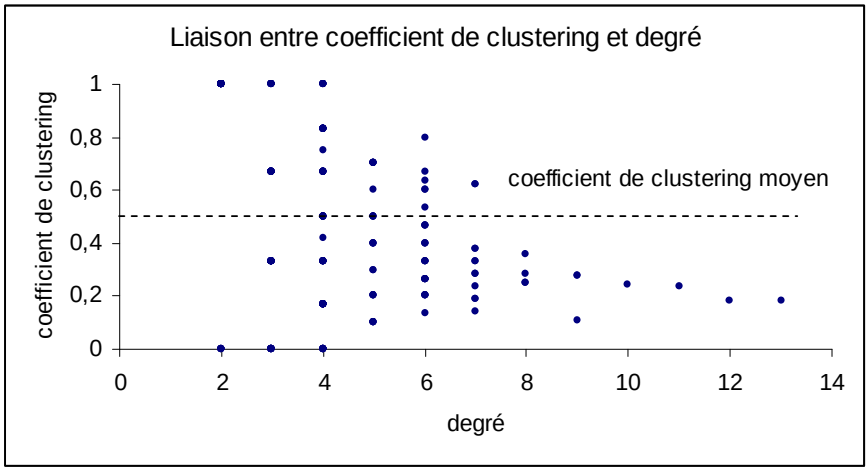


Figure 21 – liaison entre coefficient de clustering et degré – graphe de CiteSeer

Remarque 3. Ce graphe ressemble à un arbre dont certains nœuds « proches » auraient été liés. C'est un graphe « arboré » (section 1.3.1.5). Un algorithme de dessin de graphe adapté donne une vue agréable de ce graphe (Figure 19).

1.3.1.2 Petit graphe de CiteSeer

Exemple 17. Nous avons collecté un graphe avec la même technique, partant du focus : "Navigation and Interaction within Graphical Bookmarks" (Hascoët 1999). Il comprend 122 articles et 206 liens et présente aussi une allure « arborée ».

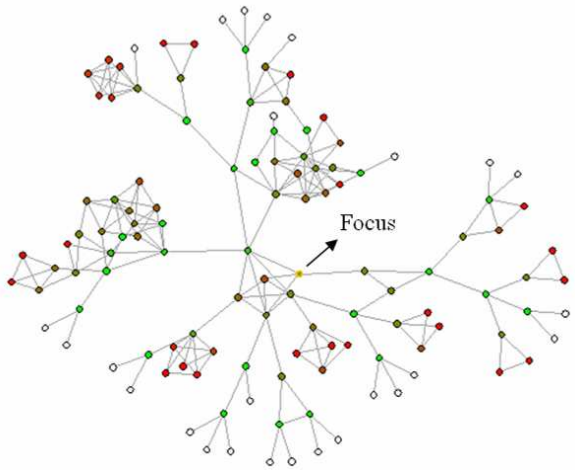


Figure 22 – petit graphe de CiteSeer « arboré »

Le graphe « arboré » est représenté en Figure 22. Les nœuds à coefficient de clustering important apparaissent en rouge.

Comme auparavant, la distribution des degrés (Figure 23) suit approximativement une loi de Poisson (si l'on supprime les nœuds de degré 1 appartenant à la dernière couche). Ce n'est donc pas un graphe « sans échelle ».

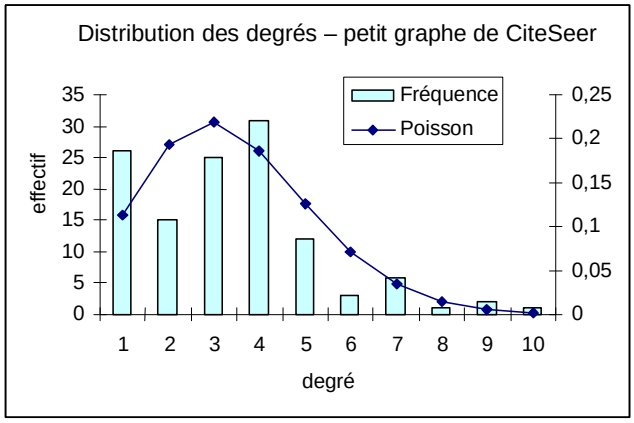


Figure 23 – distribution des degrés – petit graphe de CiteSeer

Le coefficient de clustering vaut 0,49 : c'est un graphe « petit monde ». Les sommets de degré fort ont un coefficient de clustering plus faible que la moyenne (Figure 24).

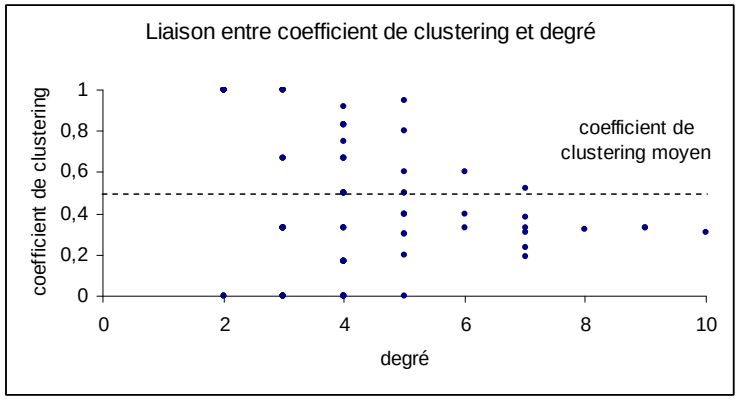


Figure 24 – liaison entre coefficient de clustering et degré – petit graphe de CiteSeer

1.3.1.3 Gros graphe de CiteSeer

Nous présentons un gros graphe issu de CiteSeer comprenant 2888 articles et 4985 liens de similarité (related) à partir du focus : « Graph Visualization and Navigation in Information Visualization: a Survey» (Herman, Mélançon et al. 2000) (au « centre » du graphe Figure 25).

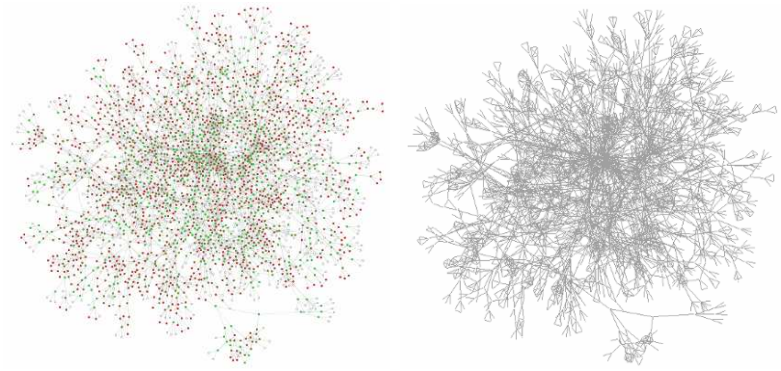


Figure 25 – gros graphe de CiteSeer (a) nœuds à fort clustering en rouge (b) arêtes

Le graphe a une allure « arborée touffue » (buisson…). Aussi, nous proposons deux vues du graphe : les nœuds à fort coefficient sont représentés en rouge (Figure 25a). Ils sont bien répartis dans le graphe et appartiennent généralement à de petites composantes denses. La représentation des arêtes seules (Figure 25b) souligne le côté « arboré touffu ».

La distribution des degrés est proche d'une loi de Poisson (Figure 26).

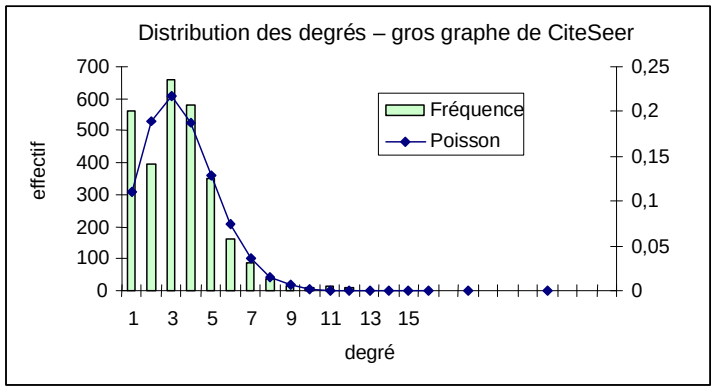


Figure 26 – distribution des degrés – gros graphe de CiteSeer

Le graphe est « petit monde » (coefficient de clustering : 0,46). Comme précédemment, les hubs ont un faible coefficient de clustering (Figure 27).

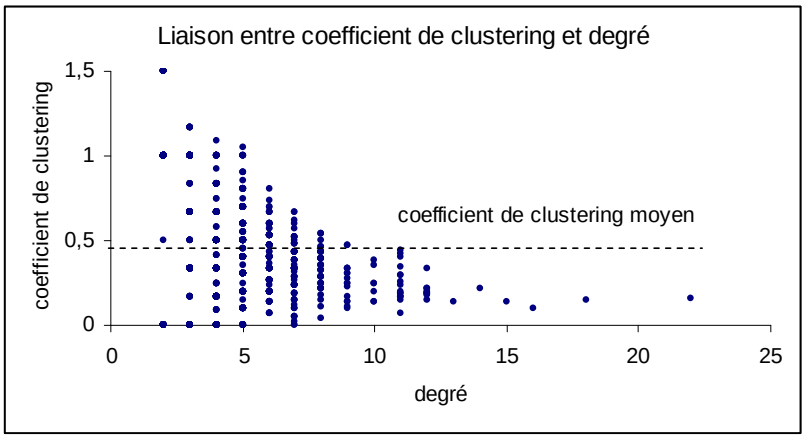


Figure 27 – liaison entre coefficient de clustering et degré – gros graphe de CiteSeer

1.3.1.4 Graphes des amitiés et sympathies entre étudiants

Nous considérons, dans cette section, le graphe des amitiés (bilatéral) et le graphe des sympathies (unilatéral) entre 56 étudiants d'une promotion (étude Boutin 2005).

Chaque étudiant devait cocher dans la liste des 56 étudiants ceux qu'il appréciait, sans autre consigne particulière.

Dans le graphe des amitiés (197 liens, Figure 28a), deux étudiants sont liés s'ils s'apprécient tous deux. Dans le graphe des sympathies (464 liens, Figure 28b), deux étudiants sont liés si l'un d'eux au moins apprécie l'autre.

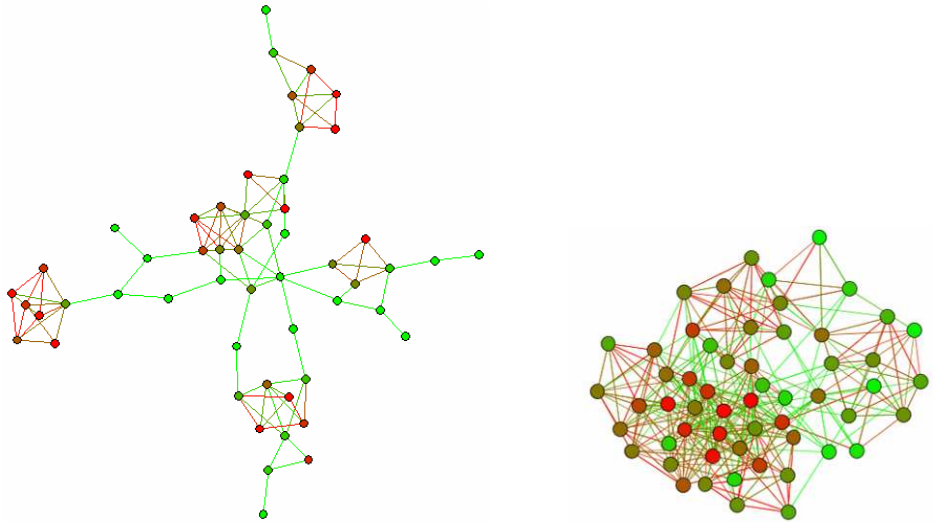


Figure 28 – (a) graphe d'amitiés (b) graphe de sympathies

Propriété 12. Notons quelques propriétés de ces graphes :

- Le graphe des amitiés est constitué de cinq composantes en interaction.
- Le graphe des amitiés a l'apparence d'un graphe « arboré ». Il contient des composantes qui se dégagent clairement d'une structure arborescente.
- Le graphe des sympathies beaucoup plus dense contient des composantes fortement interconnectées.
- Les histogrammes des degrés se rapprochent de lois de Poisson (voir Figure 29).

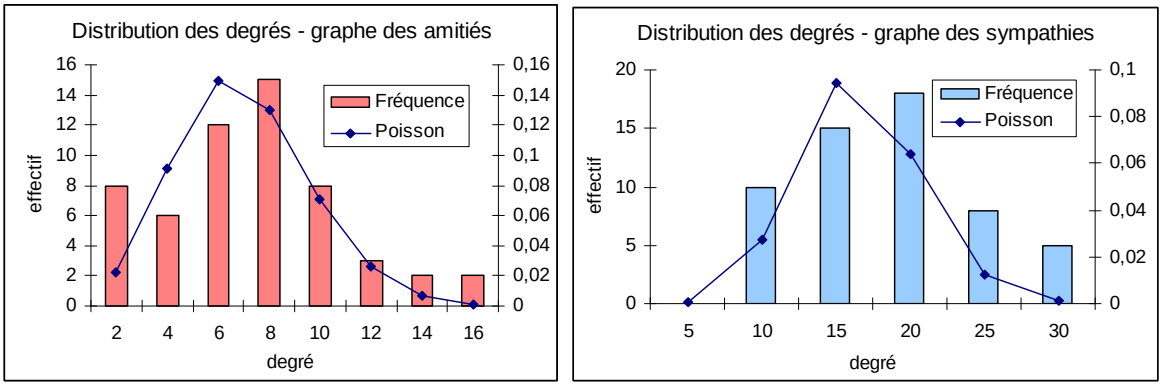


Figure 29 – répartition des degrés (a) graphe des amitiés (b) graphe des sympathies

Propriété 13. Nous présentons, Figure 30, la répartition des coefficients de clustering dans les graphes d'amitiés et de sympathies :

- Dans le graphe des amitiés, 15 étudiants ont un coefficient de clustering nul (pas d'amitiés triangulaires). Les autres font partie de petits groupes d'amis.
- Dans le graphe des sympathies, la répartition a l'allure d'une loi Normale (gaussienne) : peu d'individus ont un coefficient de clustering très fort ou très faible.

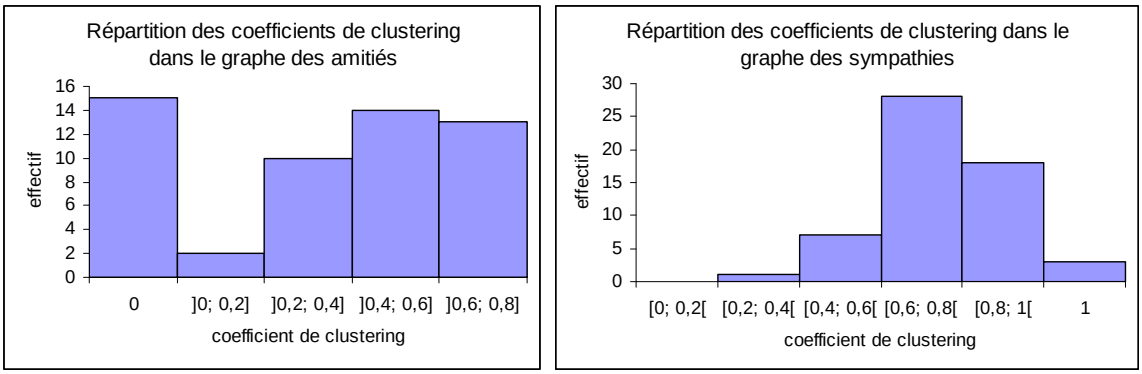


Figure 30 – répartition des coefficients de clustering (a) amitiés (b) sympathies

Propriété 14. Nous avons noté précédemment que le coefficient de clustering local est faible quand le degré est important (résultat provenant directement de la définition du coefficient de clustering).

- On retrouve ce résultat dans le graphe des sympathies (Figure 31b).
- On ne retrouve pas ce résultat pour le graphe d'amitiés (Figure 31a). En effet les groupes d'amis sont denses (amitiés fortes entre tous ses membres).

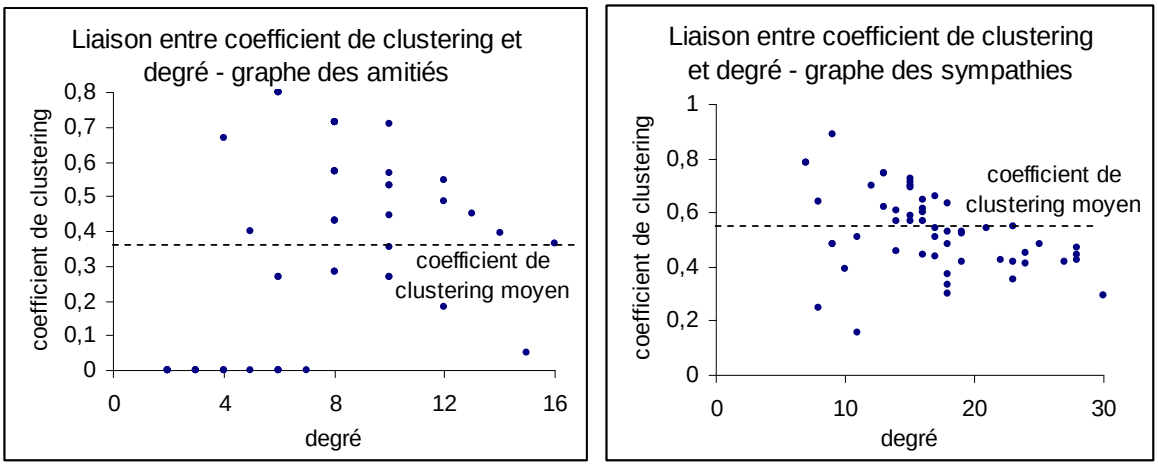


Figure 31 – liaison entre degré et coefficient de clustering (a) amitiés (b) sympathies

1.3.1.5 Commentaires sur les graphes « petit monde arborés »

Les graphes de similarité de CiteSeer ainsi que le graphe d'amitiés sont « arborés » au sens défini en section 1.2.5. Intuitivement, il s'agit d'arbres auxquels ont été ajoutées des liens entre nœuds « proches ».

Cette structure a été mise en évidence dans le graphe de CiteSeer de taille moyenne (Figure 19). Elle apparaît plus clairement encore dans le petit graphe de CiteSeer (Figure 22). Le caractère « arboré » du gros graphe de CiteSeer (Figure 25) est beaucoup moins évident. En effet la structure est très « touffue » et le graphe ressemble davantage à un « buisson » qu'à un arbre (c'est du moins l'impression que donne la vue obtenue avec modèle de forces).

En utilisant des liens de citations et non de similarités, la structure du graphe est bien différente de celle d'un graphe « arboré » (voir graphe de citations d'InfoVis, Figure 39).

En fait, la majorité des réseaux d'interactions réels que nous allons étudier ne sont pas « arborés ». Comment expliquer que les graphes de similarité et d'amitiés le soient ?

Ce résultat provient de la nature des relations étudiées :

Les relations de similarité et d'amitiés, sont réflexives et symétriques. De plus, si A et B sont reliés ainsi que B et C, alors A et C ont des chances de l'être (nous parlerons de « pseudo » transitivité). Nous qualifierons cette relation de « pseudo » relation d'équivalence.

Il est alors naturel qu'apparaissent des clusters comme composantes connexes de cette « pseudo » relation d'équivalence. Ces clusters correspondent intuitivement à des domaines.

Les clusters ont « tendance » à appartenir à une arborescence. C'est le cas du petit graphe de CiteSeer (Figure 22). En effet deux domaines distincts ont peu de chances d'être liés par plus d'un lien (au risque d'appartenir à un même grand domaine). Parfois, cependant, deux clusters peuvent appartenir à un même cycle. C'est le cas par exemple des deux clusters principaux du graphe moyen (Figure 19) qui appartiennent à un grand cycle englobant le focus. Dans le gros graphe de CiteSeer (Figure 25), le nombre de grands cycles reliant les clusters est certainement plus important. Ce qui explique cette vue « arborée touffue ».

Nous proposons, en section 2.5, une nouvelle technique de filtrage d'arêtes permettant de briser les grands cycles interconnectant les clusters. Le graphe résultant est « bien arboré ». Les différents domaines sont organisés en une arborescence de clusters. Nous verrons que cette technique fonctionne aussi bien sur des graphes « arborés touffus » (comme le gros graphe de CiteSeer) que sur des graphes d'interactions complexes ayant une structure non « arborée » comme la majorité des réseaux « sans échelle » à comportement « petit monde » (voir section 1.3.2).

1.3.2 Réseaux « petit monde sans échelle »

1.3.2.1 Graphe de co-auteurs du LIRMM

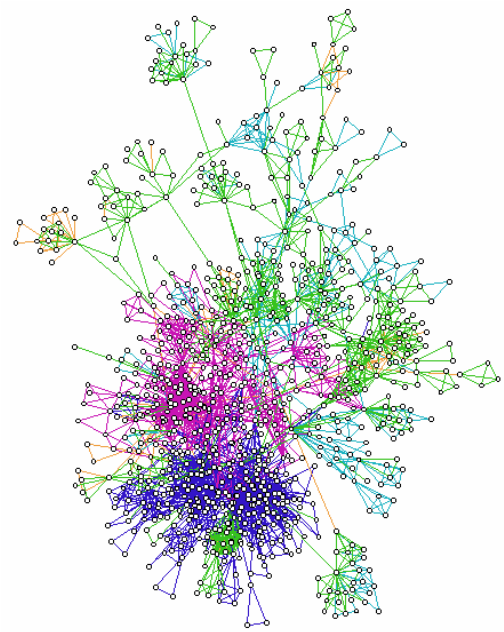


Figure 32 – graphes des co-auteurs du LIRMM

Nous étudions dans cette section le graphe de co-auteurs du LIRMM (Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier) contenant 1511 noeuds et 7902 liens. Chaque nœud représente un auteur ayant participé à une publication du LIRMM. Un lien est établi entre deux auteurs s'ils ont signés une publication en commun.

Le graphe comprend 294 petites composantes connexes et une composante géante (Figure 32). Cette composante est formée d'un noyau très dense et d'un halo de petites composantes.

La couleur des liens correspond au domaine de publication : bleu (micro électronique), rose (robotique), vert (informatique). On note une interconnexion des différents domaines.

La distribution des degrés suit approximativement une loi de puissance (voir Figure 33). Le graphe est donc typiquement un graphe « sans échelle ».

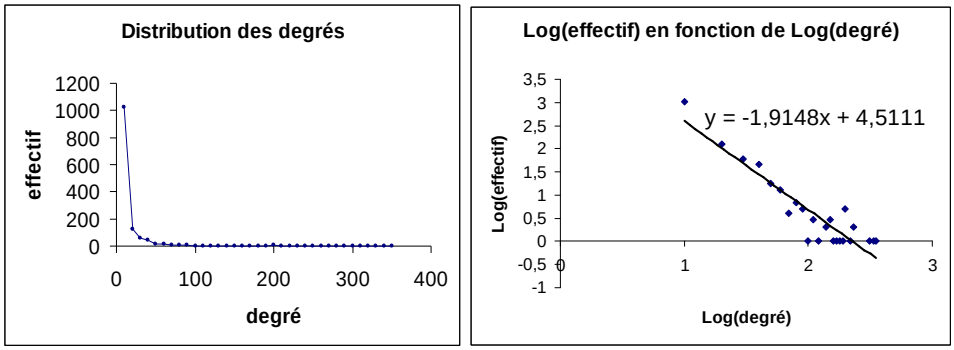


Figure 33 – distribution des degrés – (a) échelle linéaire (b) log log – co-auteurs du LIRMM

Ce graphe a la particularité d'être valué puisque deux auteurs peuvent avoir plusieurs publications en commun. Ainsi, nombreux coefficients de clustering locaux dépassent 1 et le coefficient de clustering moyen vaut 1,3. C'est donc un graphe « petit monde ».

Le graphe de co-auteurs est par conséquent un graphe « sans échelle » à comportement « petit monde ». Newman (Newman 2001) a également étudié expérimentalement des graphes de co-auteurs et montré leur caractère « sans échelle » et « petit monde ».

La distribution des coefficients de clustering suit aussi une loi de puissance (Figure 34).

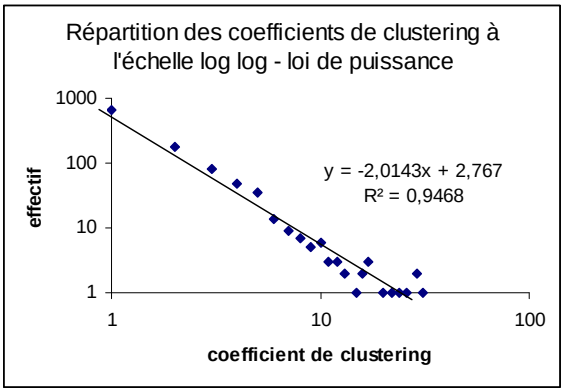


Figure 34 – répartition des coefficients de clustering à l'échelle log log – co-auteurs

Les nœuds de faible degré peuvent avoir un coefficient de clustering important, mais les nœuds de fort degré ne peuvent avoir qu'un coefficient de clustering faible (Figure 35). Ce résultat a déjà été obtenu pour divers graphes étudiés en section 1.3.1.

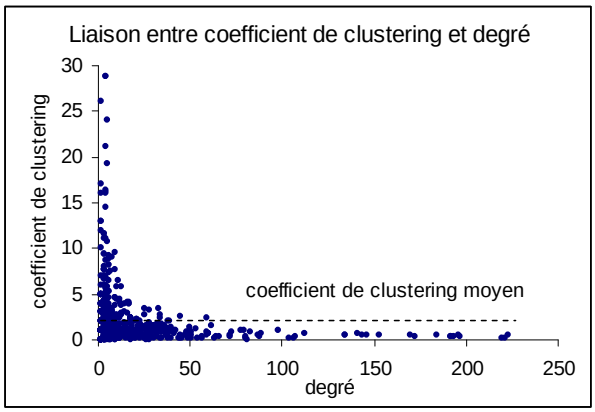


Figure 35 – liaison entre degré et coefficient de clustering local – graphe de co-auteurs

1.3.2.2 Graphe dual des co-publications du LIRMM

Le graphe des co-publications du LIRMM est le dual du graphe des co-auteurs. Les nœuds représentent les publications (en rouge les publications à coefficient de clustering important). Une arête associe deux publications écrites par un même auteur (Figure 36).

Ce graphe est « presque » « sans échelle » car la distribution de ses degrés suit « plus ou moins » une loi de Puissance (Figure 37). De plus, le coefficient de clustering moyen vaut 1,18 : le graphe est donc « petit monde ».

Une fois encore, les nœuds de très fort degré ont un coefficient de clustering local inférieur au coefficient de clustering moyen (voir Figure 38).

Remarque 4. On note différents groupes de publications correspondant le plus souvent à un même « grand auteur » et son équipe. Lorsque deux clusters sont liés par une publication, ça signifie que cette publication est commune à 2 grands auteurs.

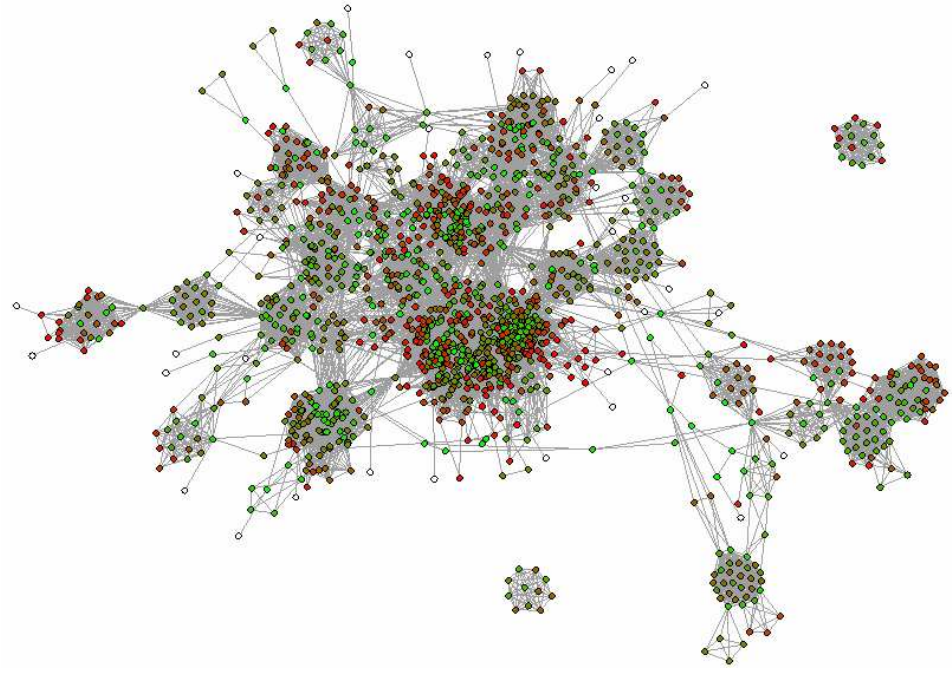


Figure 36 – graphe de co-publications du LIRMM

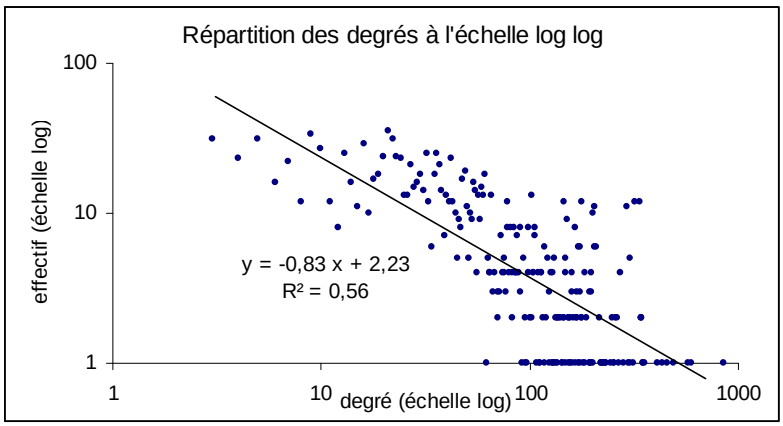


Figure 37 – répartition des coefficients de clustering à l'échelle log log – co-publications

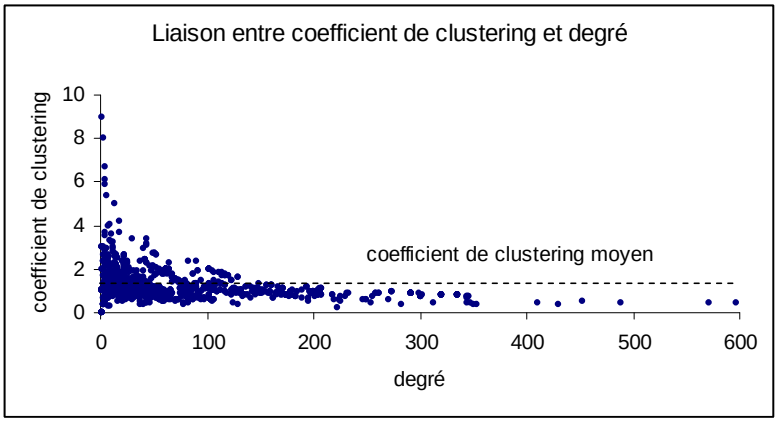


Figure 38 – liaison entre degré et coefficient de clustering local – graphe de co-publications

1.3.2.3 Graphe de citations d'InfoVis Contest 2004

Nous étudions dans ce paragraphe le graphe des 614 papiers publiés entre 1994 et 2004 à la conférence InfoVis. Ce graphe a été proposé comme jeu de test au concours d'InfoVis (InfoVis'Contest 2004). Les 1970 arêtes du graphe correspondent aux citations entre articles d'InfoVis.

Le graphe (Figure 39) présente un noyau très dense d'articles et quelques articles épars en périphérie. Il s'agit d'un réseau « sans échelle ». Il présente en effet une distribution des degrés suivant une loi de puissance (Figure 40).

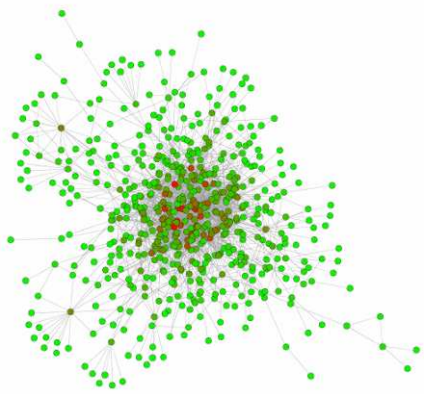


Figure 39 – graphe des citations – InfoVis Contest 2004

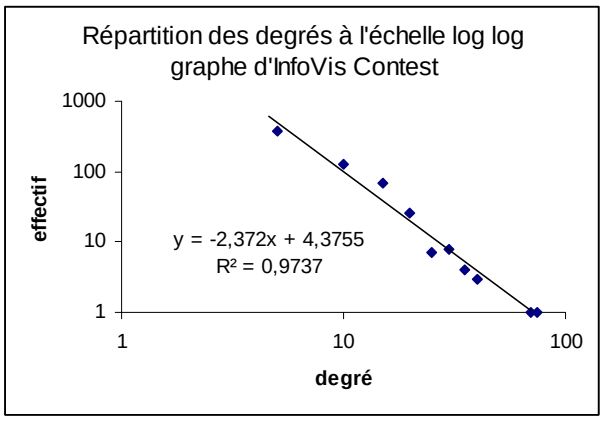


Figure 40 – répartition des degrés – échelle log log – InfoVis Contest 2004

Le coefficient de clustering moyen vaut 0,21. Il s'agit d'un réseau « petit monde ». Une fois encore, les coefficients de clustering importants correspondent aux nœuds de degré faible (Figure 41).

Le réseau d'InfoVis Contest est donc « sans échelle » à comportement « petit monde ».

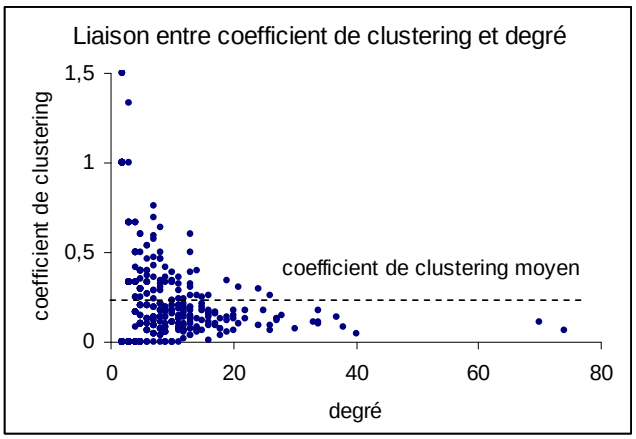


Figure 41 – liaison entre coefficient de clustering et degré – InfoVis Contest

1.3.2.4 Graphe d'interactions de protéines - « YEAST »

Les réseaux métaboliques (Jeong, B. Tombor et al. 2000), et les réseaux d'interactions de protéines (Barabási et Oltvai 2004) sont également le plus souvent « sans échelle ».

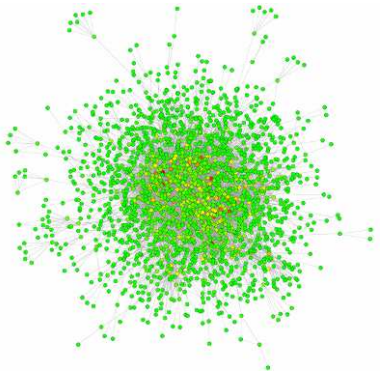


Figure 42 – Composante géante du graphe « YEAST »

Nous considérons ici le graphe d'interactions de protéines (nommé YEAST car il fait intervenir des levures). Il comprend 2361 protéines et 7182 interactions (Barabási et Oltvai 2004). La Figure 42 présente la composante connexe géante contenant 78 % des protéines (une autre vue, issue de la littérature, est proposée en Figure 17).

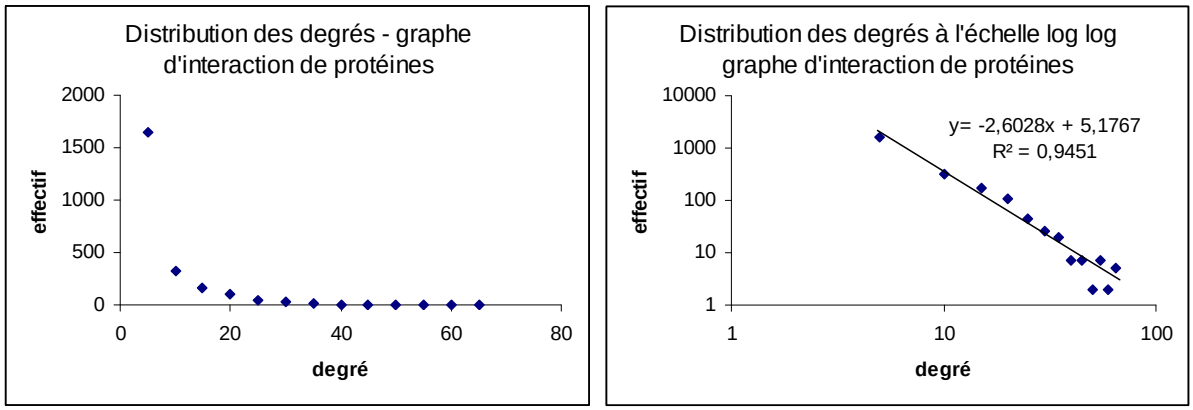


Figure 43 – répartition des degrés (a) échelle linéaire (b) log log – YEAST

C'est un graphe « sans échelle » de paramètre $\gamma = 2{,}6$ (Figure 43). Comme dans les graphes précédents, le coefficient de clustering local est faible pour un degré important (Figure 44).

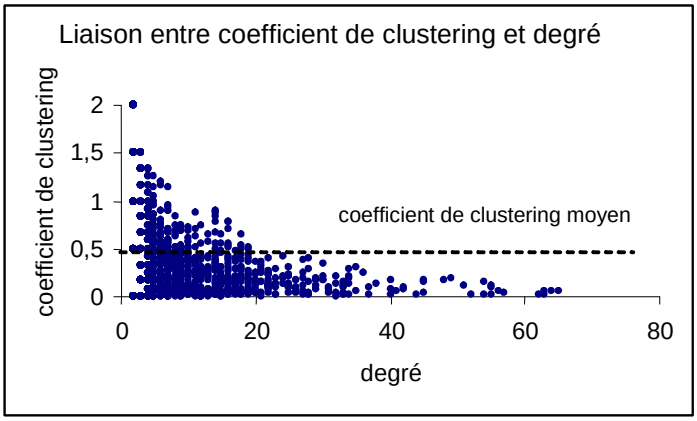


Figure 44 – coefficient de clustering en fonction du degré – YEAST

1.3.2.5 Graphes « petit monde sans échelle » : commentaires

Nous avons présenté dans cette section divers réseaux « sans échelle » à comportement « petit monde » issus du monde réel. Ils présentent généralement un noyau très dense entouré d'un amas de petites composantes.

Les vues classiques obtenues avec un algorithme à base de forces sont peu exploitables car trop denses.

Nous proposons en section 2.5 une nouvelle technique de filtrage d'arêtes permettant de transformer ces graphes complexes en graphes « arborés » facilement visualisables. Nous développons ensuite des techniques de partitionnement multi-échelles adaptées à la structure « arborée » (section 3.6). Nous proposons enfin une technique de visualisation multi-échelles d'un graphe « arboré » à partir d'un focus utilisateur.

Les divers exemples introduits dans cette section seront traités en section 5.3.

« Au fond, est-ce que ranger ça ne revient pas un peu à foutre le bordel dans son désordre ? »

Philipe Geluck

Chapitre 2 Filtrage de graphes

Les graphes « sans échelle » présentent le plus souvent une composante connexe principale à noyau très dense d'où il est difficile d'extraire des motifs pertinents (voir Figure 39 et Figure 42). Comme nous l'avons constaté en section 1.3, ils se prêtent généralement mal aux algorithmes de visualisation basés sur un modèle de forces.

Un nettoyage préalable du graphe s'impose, basé sur une technique de filtrage appropriée (Huang, Eades et al. 2005). Le graphe ainsi épuré peut alors révéler une structure et des motifs intéressants. La difficulté est de proposer un filtre qui révèle des caractéristiques sans pour autant dénaturer le graphe.

Nous présentons tout d'abord dans ce chapitre des techniques de filtrage globales, basées sur la suppression de nœuds ou d'arêtes en fonction d'une métrique. Nous décrivons ensuite des techniques de filtrage contextuelles (basées sur un focus).

Nous introduisons, enfin, deux nouvelles techniques de filtrage adaptées aux grands réseaux d'interactions. La première, basée sur la détection de plus petits cycles passant par une arête, favorise l'extraction de motifs (cycles). La seconde, particulièrement adaptée au traitement de gros graphes denses, repose sur la détection et la suppression de grands raccourcis dans un arbre couvrant. Elle fournit un graphe « arboré » (voir section 1.2.5) facilement visualisable avec un algorithme basé sur un modèle de forces.

2.1 Choix d'une métrique

Filtrer un graphe revient à filtrer ses nœuds ou ses arêtes selon certains critères (Huang, Eades et al. 2005).

Propriété 15. Un filtre sur l'ensemble des arêtes transforme un graphe $G = (V, E)$ en un graphe partiel $G' = (V, E')$ avec $E' \subseteq E$ (voir Définition 11).

Propriété 16. Un filtre sur l'ensemble de nœuds transforme le graphe $G = (V, E)$ en un sous graphe induit $G' = (V', E')$ avec $V' \subseteq V$ et $E' = \{(u, v) \in E \cap V'^2\}$.

Les critères de filtre sont basés sur les propriétés quantitatives ou qualitatives des sommets ou arêtes (Henry 1992; Huang, Eades et al. 2005). Une métrique doit donc être donnée ou calculée sur les sommets ou arêtes. On obtient, suivant le cas, un graphe sommet-valué ou arête-valué.

En pratique, diverses métriques sont utilisées : certains graphes disposent d'une métrique naturelle indépendante de leur structure. Si on considère par exemple un graphe d'amitiés, la force du lien entre deux personnes est relevée par l'expérimentateur et ne dépend

pas de la structure du graphe. De même pour l'âge des personnes (critère quantitatif) ou leur ville d'appartenance (critère qualitatif).

En revanche, certains graphes ne disposent d'aucune donnée supplémentaire. Il est alors possible de calculer une métrique dépendant uniquement de la structure du graphe. Nous nous intéressons particulièrement à ce type de graphe dans ce chapitre.

Nous avons considéré en section 1.2 divers graphes d'interactions présentant pour la plupart des propriétés « sans échelle » et « petit monde ». Ils sont ainsi caractérisés par :

- Une distribution des degrés suivant une loi de puissance
- Un coefficient de clustering supérieur à celui d'un graphe aléatoire
- Un diamètre moyen faible

Nous considérons, dans cette partie, divers filtres (basés sur diverses métriques) et leur incidence sur ces trois propriétés des graphes « petit monde sans échelle ».

2.2 Filtrage de nœuds

Nous présentons différentes techniques de filtrage de nœuds : filtrage aléatoire, filtrages suivant le degré, l'indice d'inter centralité ou le coefficient de clustering.

2.2.1 Filtrage aléatoire de noeuds

Un filtre élémentaire consiste à supprimer aléatoirement des nœuds.

Exemple 18. Dans le graphe d'interactions de protéines, nous proposons d'appliquer un filtre aléatoire sur les nœuds (voir Figure 45 et Figure 46). Parmi les 2361 nœuds, nous filtrons successivement 100, 500, 1000, 1500 puis 2000 nœuds. Remarquons que le noyau dense est conservé pour les quatre premiers filtres. Pour le dernier filtre le seuil de cohésion du noyau est franchi : le noyau est décomposé en plusieurs petites composantes.

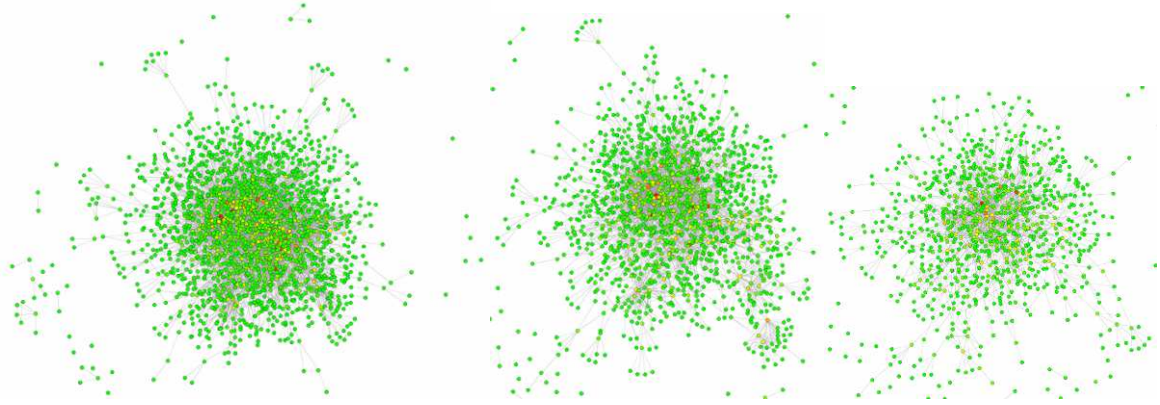


Figure 45 – filtre de 100, 500, 1000 nœuds – YEAST

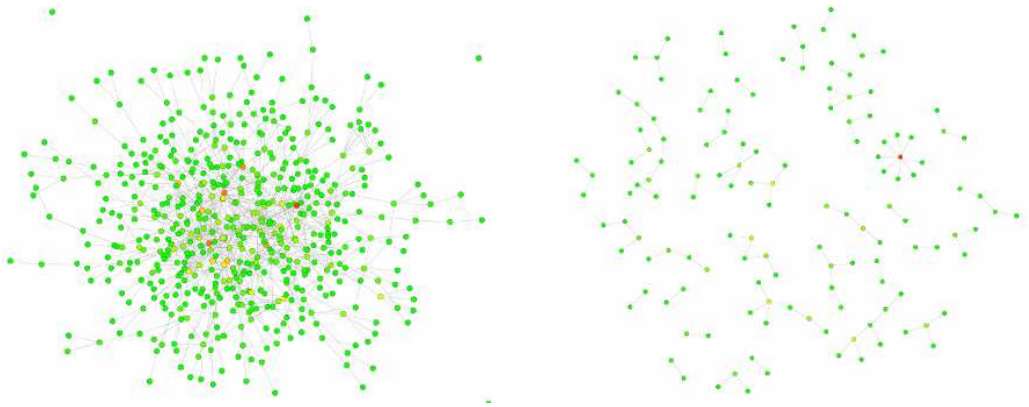


Figure 46 – filtre de 1500 et 2000 nœuds – YEAST

La technique de filtre aléatoire permet ainsi de simplifier un graphe dense tout en conservant son allure générale (si l'on reste au dessus d'un certain seuil de nœuds filtrés). Cependant une telle technique ne permet pas de révéler motifs et structure du graphe.

Propriété 17. Un filtre aléatoire a une incidence sur la structure du graphe :

- Un graphe « sans échelle » reste « sans échelle » après filtrage : la distribution des degrés suit encore une loi de puissance après suppression aléatoire de nœuds.
- Le diamètre moyen augmente avec les suppressions de nœuds.
- Le coefficient de clustering diminue : dans l'exemple décrit Figure 45 et Figure 46, il vaut respectivement 0.476, 0.477, 0.429, 0.327 et 0.063 si l'on supprime aléatoirement 100, 500, 1000, 1500 et 2000 nœuds.

Remarque 5. On peut procéder, de la même manière, à un filtrage aléatoire d'arêtes.

2.2.2 Filtrage de nœuds selon leur degré

La métrique du degré est certainement la plus naturelle et la plus simple à calculer puisqu'elle définit le poids d'un sommet par le nombre de ses voisins.

Filtrer les nœuds de degrés 0 et 1 revient à supprimer respectivement les sommets isolés et les feuilles du graphe initial. Notons que le graphe résultant peut lui-même contenir des sommets de degré 0 ou 1. Le filtre peut être alors appliqué itérativement sur le graphe résultant jusqu'à ce qu'il n'y ait plus de nœuds de degré 0 ou 1 (voir Figure 47).

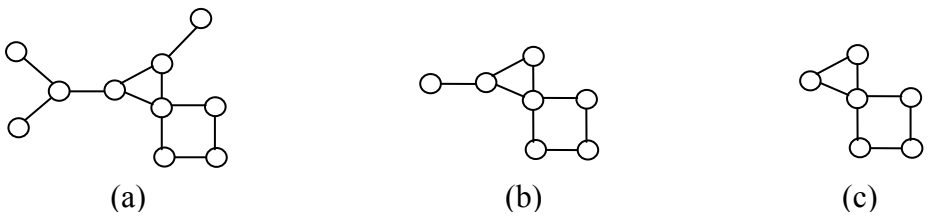


Figure 47 – filtre des noeuds de degré 1 (a) graphe initial (b) 1ère itération (c) 2ème itération

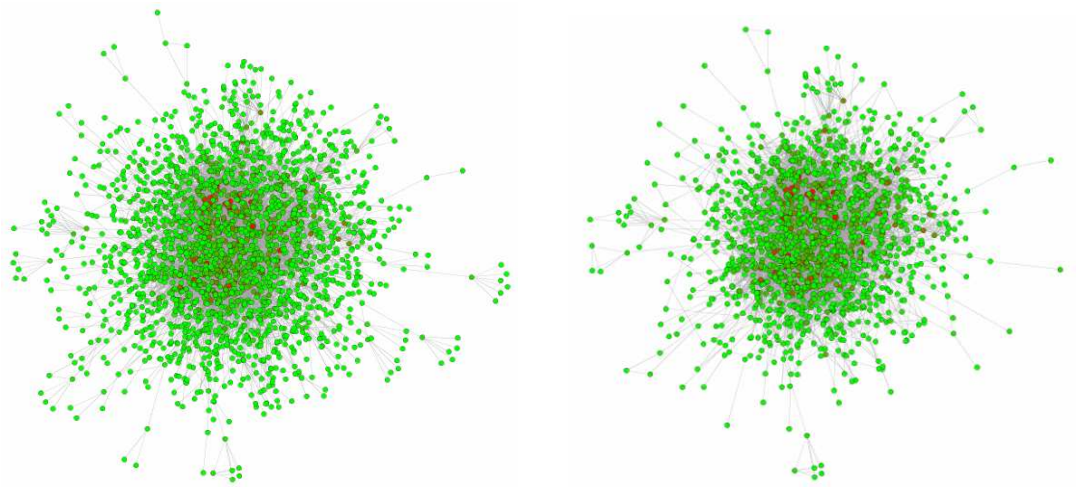


Figure 48 – (a) graphe initial « YEAST » (b) filtrage des nœuds de degré 1

Exemple 19. Nous présentons en Figure 48, la composante géante du graphe d'interactions de protéines avant et après filtrage des nœuds de degré 1 (768 nœuds filtrés sur 2361 nœuds). La structure du noyau principal a peu changé. L'intérêt de ce filtre, pour ce type de graphe, est uniquement de supprimer le bruit autour du noyau principal. Le filtre peut être réitéré sur le graphe résultant.

De façon générale, on peut effectuer le filtrage de sommets de degré inférieur à k de façon à ne garder que les nœuds de fort degré. Ce filtre peut être appliqué uniquement sur le graphe initial ou itérativement jusqu'à ce que tous les nœuds du graphe résultant aient un degré supérieur ou égal à k. On parle alors de « filtrage itéré ».

Exemple 20. Nous présentons en Figure 49, un filtrage itéré du graphe de co-citations du LIRMM ne conservant que les nœuds de degré supérieur ou égal à 6 : certaines petites composantes sont déconnectées. La composante principale reste très dense et difficilement analysable.

Figure 49 – (a) graphe des co-auteurs du LIRMM (b) filtrage des sommets de degré < 6

Exemple 21. Nous présentons sur le graphe de CiteSeer de taille moyenne (voir Figure 19) le filtrage itéré des nœuds de degré 1 et 2 (Figure 50). Les graphes résultants sont « arborés » (au sens défini en section 1.2.5) comme le graphe initial. La structure du graphe est plus lisible.

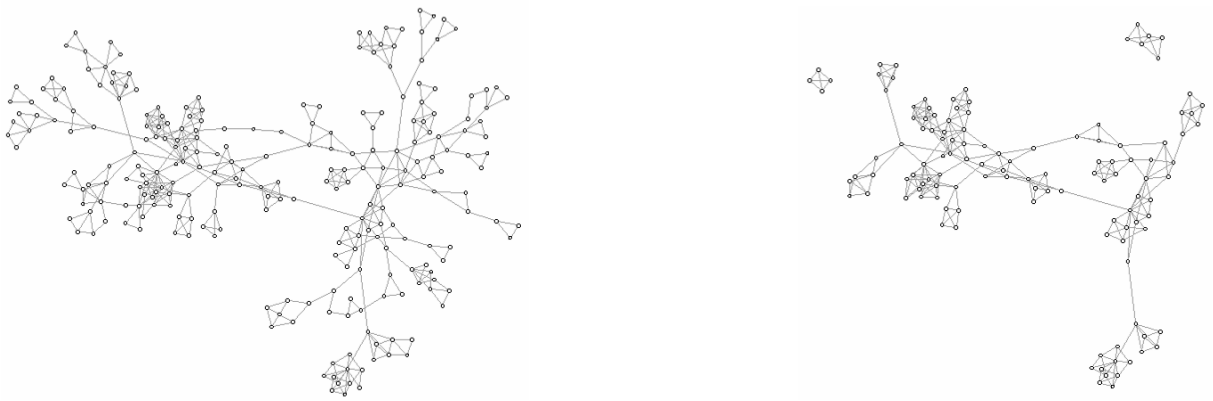


Figure 50 – CiteSeer de taille moyenne – (a) filtre des noeuds de degré 1 (b) 2

Réciproquement, d'autres techniques consistent à ne filtrer que les nœuds de degré fort. A partir d'un certain seuil, le noyau principal « explose » en de multiples composantes.

Exemple 22. Dans le graphe des co-auteurs du LIRMM, en filtrant les sommets de fort degré on supprime des hubs (« chefs » signant de nombreuses publications) pour privilégier les coopérations entre « petits » auteurs (voir Figure 51). En supprimant les nœuds de degré > 20, le graphe est « éclaté » en petites composantes sans relations entre elles.

Le filtrage des nœuds de fort degré transforme un graphe à noyau dense en un amas de petites composantes connexes. L'inconvénient est que l'on perd la connectivité du graphe.

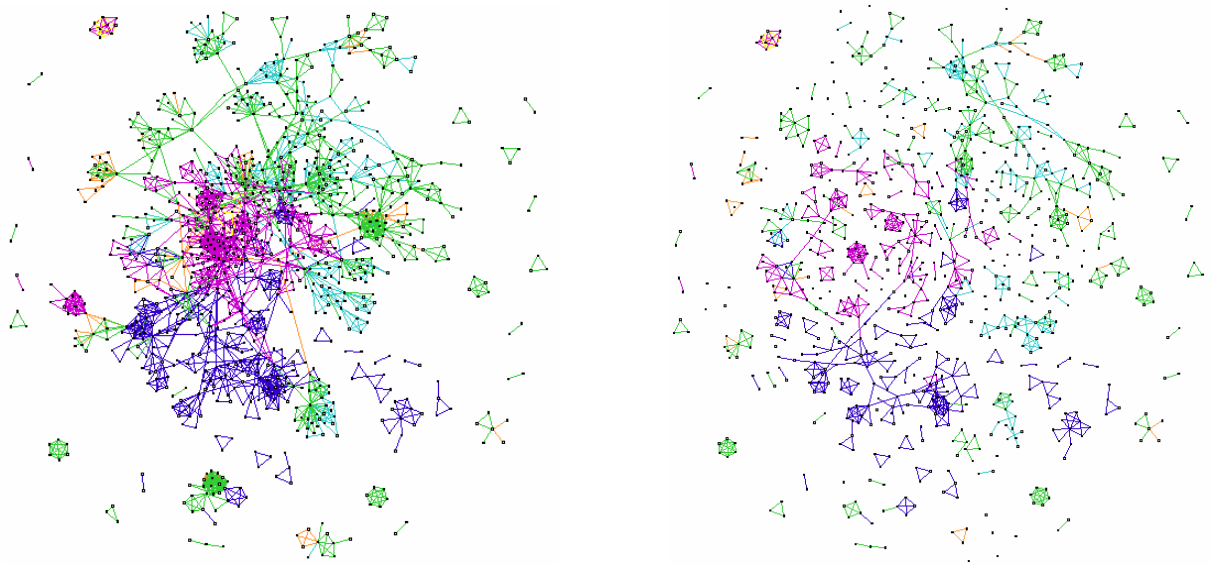


Figure 51 – graphe du LIRMM – filtre des nœuds de degré > 100 puis > 20

2.2.3 Filtre de nœuds basé sur un indice de centralité globale

Le degré d'un nœud peut être vu comme un indice de centralité locale puisque son calcul ne dépend que du voisinage direct du nœud considéré. Différents indices permettent de définir la centralité globale d'un nœud dans le graphe (Freeman 1979; Bonacich 1987; Friedkin 1991; Brandes 2001; Costenbader et Valente 2003). Il s'agit notamment de la centralité de proximité (closeness centrality), de l'inter centralité (betweenness centrality BC) et de la centralité par vecteur propre (eigen vector centrality) (Brandes et Willhalm 2002).

Définition 55. La centralité de proximité d'un nœud est définie par la distance moyenne de ce nœud aux autres nœuds du graphe.

Dans un graphe « sans échelle » à comportement « petit monde », l'utilisation d'un tel indice permet de filtrer les nœuds en périphérie du noyau central. Cependant il ne permet généralement pas de dégager des composantes intéressantes dans le noyau.

Définition 56. L'inter centralité d'un nœud (BC) (Freeman 1979) est la fraction de nombre de plus courts chemins entre deux nœuds passant par le nœud sélectionné.

Propriété 18. L'inter centralité demande un temps de calcul important même si la complexité de calcul a été ramenée à $O(n^2+nm)$ (Brandes 2001).

Intuitivement, un noeud a un fort indice d'inter centralité s'il est « central » (au sens ou de nombreux chemins passent par lui) et s'il s'apparente localement à un nœud d'articulation.

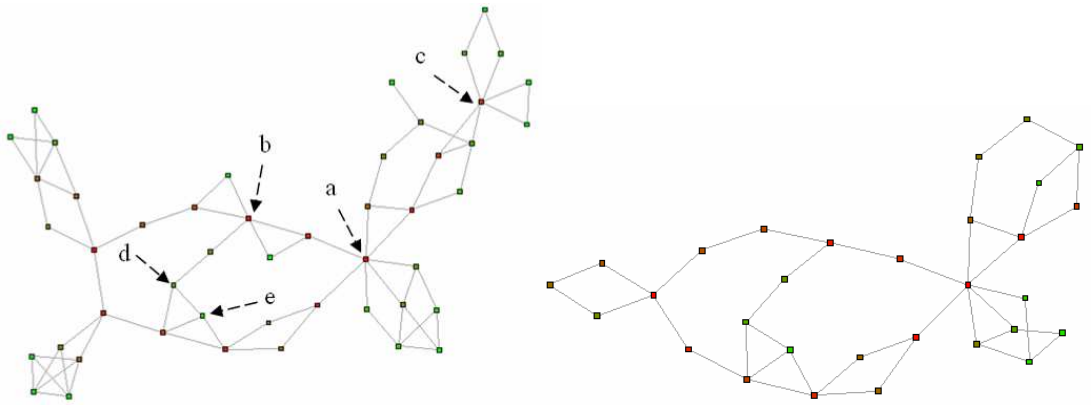


Figure 52 – cas d'école (a) graphe initial (b) filtrage itéré des nœuds d'indice BC = 0

Exemple 23. Dans le graphe « école », Figure 52a, nous étudions cinq nœuds particuliers :

- a est un nœud d'articulation « central » séparant 2 composantes de tailles voisines. Il présente l'inter centralité maximale : BC = 0,577.
- b est un nœud d'articulation local. BC = 0,278 ($5^{ème}$ plus grande).
- c est un nœud d'articulation « excentré » séparant une petite composante d'une grande : il présente la $9^{ème}$ plus grande inter centralité : BC = 0,207.
- d est un nœud d'articulation local excentré d'inter centralité faible : BC = 0,034.
- e n'est pas un nœud d'articulation local. L'inter centralité est faible : BC = 0,004.

Le résultat du filtrage itéré des nœuds d'inter centralité nulle est représenté Figure 52b. On retrouve la structure centrale du graphe (le squelette) contenant des nœuds « carrefour » par lesquels passent la majorité des chemins.

Exemple 24. La Figure 53a présente le graphe des co-auteurs où les nœuds à forte inter centralité sont en rouge. Ce sont des points d'articulation ou des nœuds « centraux ». Le filtrage des 42 nœuds d'inter centralité maximale (> 0,1) donne le graphe filtré Figure 53b. Le noyau est décomposé en composantes interconnectées et en un amas de petits clusters périphériques.

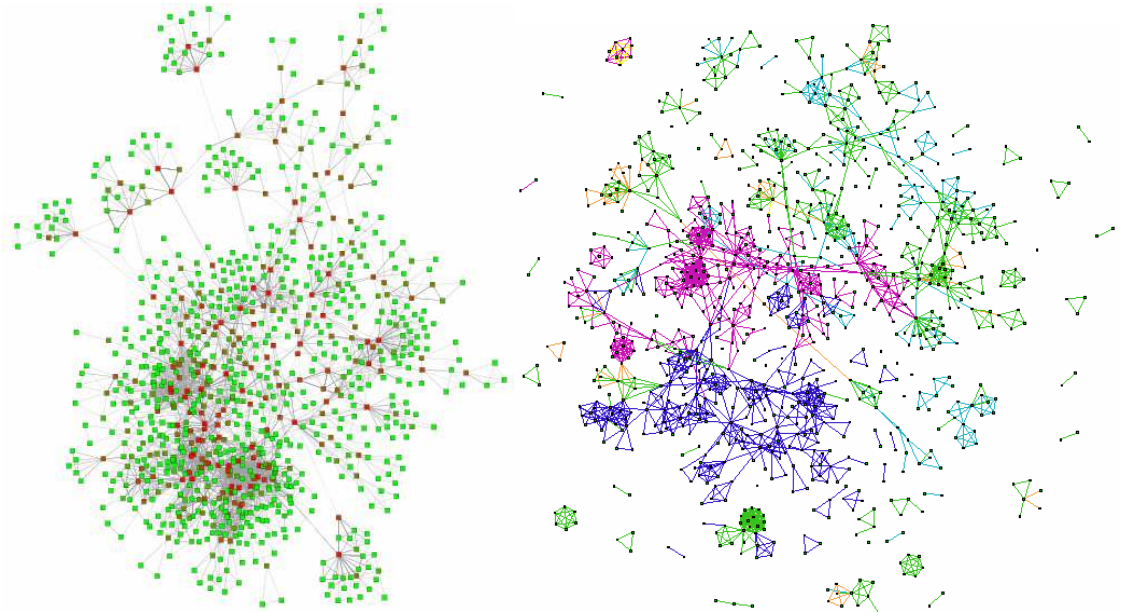


Figure 53 – graphe des co-auteurs (a) indices BC (b) filtrage des nœuds de BC > 0,1

Exemple 25. Le filtrage itéré des nœuds d'inter centralité nulle (Figure 54a) puis d'inter centralité inférieure à 0,05 (Figure 54b) simplifie la structure du graphe. Cependant les clusters restent fortement interconnectés.

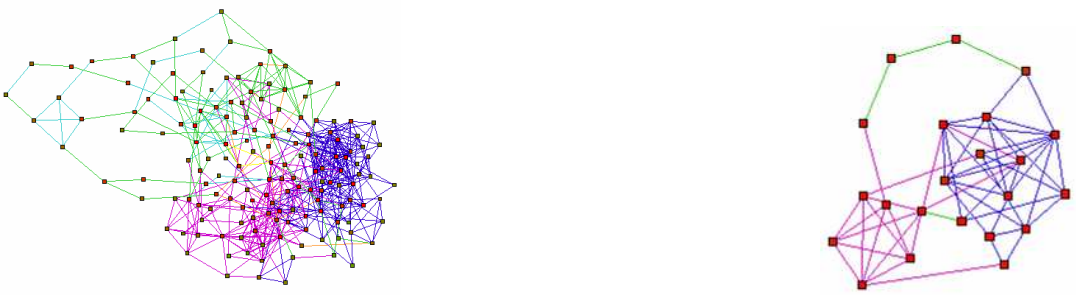


Figure 54 – (a) filtrage itéré des nœuds de BC = 0 (b) BC < 0,05

Un autre indice de centralité (Bonacich 1987) utilise des techniques matricielles :

Définition 57. Le score d'importance d'un nœud (node importance score : NIS) est une fonction affine de la moyenne des NIS de ses voisins, calculée en $O(n^{2,376})$ (Coppersmith et Winograd 1990). Le calcul de la NIS utilise un vecteur propre principal d'une matrice d'adjacence (Huang, Eades et al. 2005).

2.2.4 Filtrage de nœuds basé sur le coefficient de clustering

Le filtrage des nœuds de coefficient de clustering faible, peut révéler des motifs.

Exemple 26. La Figure 55a présente le graphe d'interactions de protéines : en blanc les nœuds de degré 1 dont on ne peut calculer le coefficient de clustering (essentiellement sur le pourtour). En vert les nœuds à faible coefficient de clustering (surtout au centre). En rouge les nœuds à fort coefficient de clustering (surtout autour du noyau).

Nous constatons que les motifs se trouvent essentiellement en périphérie, alors que les nœuds assurant l'interaction entre motifs se trouvent au centre. Ce résultat s'explique intuitivement par le théorème *central limit* : un nœud à faible coefficient de clustering mais degré important est attiré de manière quasi indépendante par ses voisins. Il en résulte qu'il a de grandes chances d'être proche du centre du nuage.

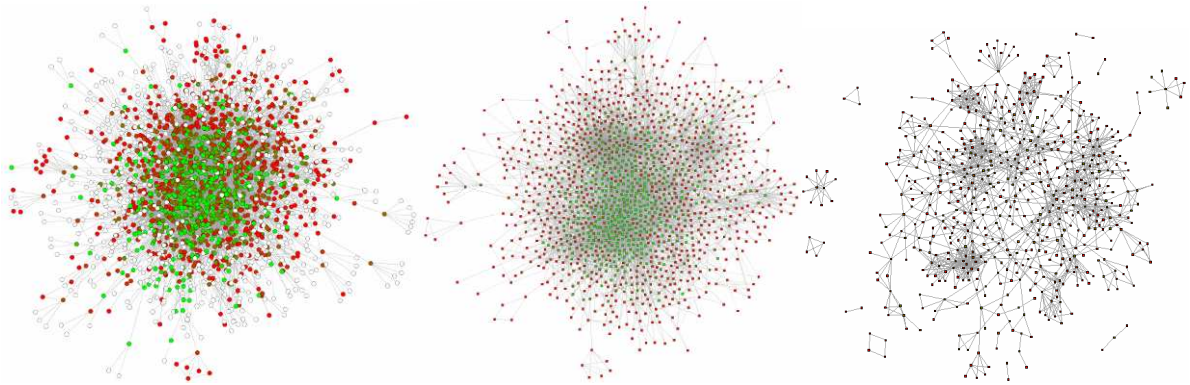


Figure 55 – (a) YEAST initial (b) filtrage noeuds de clustering = 0 (c) < 0,2

Exemple 27. Le graphe, présenté en Figure 55c, est obtenu en appliquant au graphe YEAST un filtre des nœuds de coefficients de clustering inférieur à 0,2. Le noyau initialement très dense (Figure 48) se décompose alors en une grosse composante connexe (contenant divers motifs) et une myriade de petites composantes périphériques.

La technique peut être appliquée sur le graphe initial en prenant successivement différentes valeurs seuils. Elle peut aussi être itérée, à partir d'une même valeur seuil, sur les différents graphes résultants du filtrage.

Exemple 28. La Figure 56 présente le graphe de co-auteurs du LIRMM après filtrage itéré des nœuds de clustering < 0,2, < 0,3 et < 0,4. Le graphe est initialement épuré puis décomposé en petites composantes (au seuil 0,4).

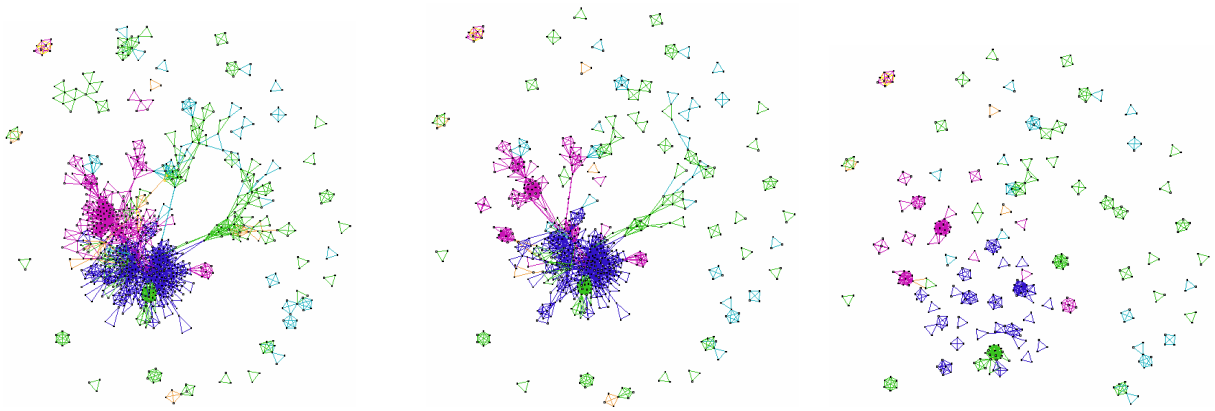


Figure 56 – filtrage des noeuds de clustering (a) < 0,2 (b) < 0,3 (c) < 0,4 – co-auteurs

2.2.5 Filtrage de nœuds – discussion

Nous avons étudié diverses techniques de filtrage de nœuds basées sur le degré, l'indice d'inter centralité (BC) et le coefficient de clustering.

Le degré et BC sont deux indices de centralité. Pour un réseau « sans échelle », les nœuds à faible degré ou BC sont souvent en périphérie du noyau dense. La suppression de ces noeuds simplifie le graphe mais le noyau n'est pas pour autant décomposé. Ce résultat est observé Figure 48, Figure 49 et Figure 54. Ces filtres peuvent être utilisés en pré traitement.

Au contraire, la suppression des nœuds de fort degré ou forte inter centralité décompose le noyau central en diverses composantes interconnectées et en un amas de clusters périphériques (Figure 51 et Figure 53). L'inconvénient de cette technique est de supprimer des nœuds « majeurs » (à fort degré ou BC) et de transformer ainsi la structure du graphe.

Pour faire apparaître les clusters, on peut également supprimer les nœuds à faibles coefficients de clustering (Figure 55 et Figure 56). Il faut pour ça choisir le bon seuil de filtrage, ce qui n'est pas chose évidente. Par ailleurs, une fois le noyau décomposé, on perd les relations entre composantes (assurées par les nœuds de faible coefficient de clustering).

Ainsi, deux cas se présentent lors du filtrage de nœuds d'un graphe « sans échelle » :

- Le graphe est épuré mais le noyau reste très dense. La structure du graphe est simplifiée, sans apparition de clusters.
- Le noyau est éclaté en diverses petites composantes sans interconnexions. Il y a apparition de clusters avec perte de la structure générale du graphe.

Dans les deux cas, ces techniques de filtrage de nœuds ne permettent pas d'extraire à la fois une structure simple du graphe et des clusters.

Nous allons introduire dans la suite de ce chapitre une technique de filtrage d'rêtes permettant la détection de clusters tout en faisant apparaître une structure simple du graphe.

2.3 Filtrage d'arêtes

2.3.1 Extraction de motifs du graphe

Dans cette section nous introduisons des techniques d'extraction de composantes reposant sur le filtrage d'arêtes. Nous décrivons ensuite une technique d'extraction d'un squelette du graphe.

2.3.1.1 Filtrage d'arêtes suivant leur force

La notion de coefficient de clustering peut être généralisée aux arêtes : la force d'une arête est définie dans (Chiricota, Jourdan et al. 2003). Le filtrage d'arêtes faibles peut faire apparaître des clusters (parfois fortement interconnectés).

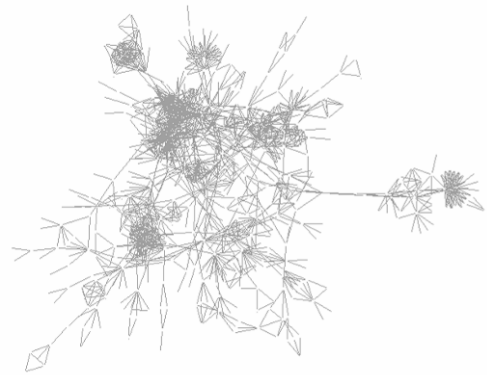


Figure 57 – filtrage des arêtes de force inférieure à 1,5 – YEAST

Exemple 29. La Figure 57 présente la composante centrale du graphe d'interactions de protéines après filtrage des 5295 arêtes de force inférieure à 1,5 (pour mieux révéler les différents motifs, nous proposons de ne visualiser que les arêtes).

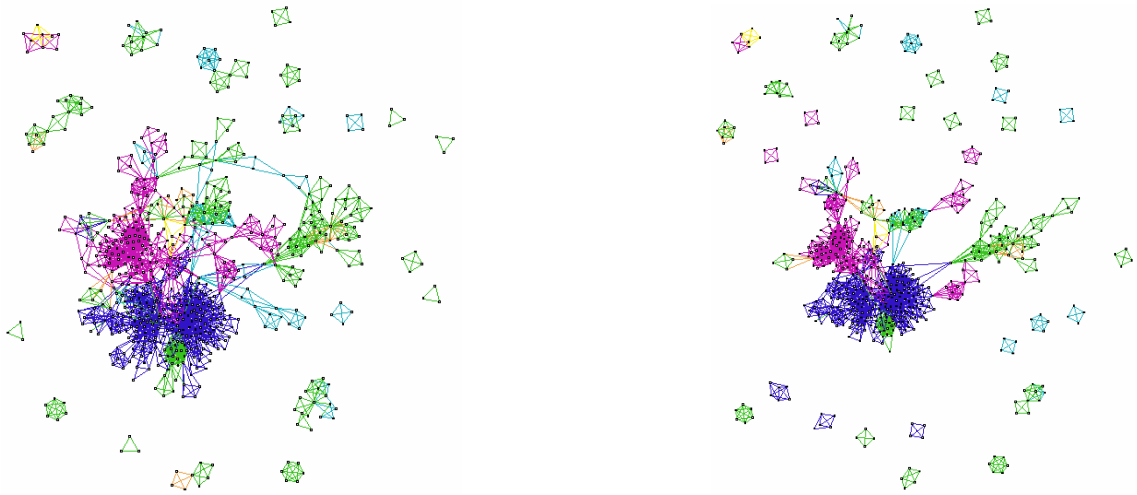


Figure 58 – filtrage itéré des arêtes de force (a) < 1 (b) < 1,5 – co-auteurs

Figure 59 – filtrage itéré des arêtes de force (a) < 2,5 (b) < 5 – co-auteurs

Exemple 30. Le filtrage des arêtes de force < 1 ; 1,5 ; 2,5 et 5 dans le graphe des co-auteurs (Figure 58, Figure 59) révèle la présence de clusters. Rappelons que la couleur des liens correspond au domaine de publication : bleu (micro électronique), rose (robotique), vert (informatique). Par soucis de clarté, les nœuds de degré nul ont été supprimés.

2.3.1.2 Filtrage d'arêtes suivant leur coefficient de centralité

L'inter centralité d'une arête (betweenness centrality BC) (Freeman 1979) peut être utilisée pour séparer des composantes (voir inter centralité des nœuds section 2.2.3). Une arête à inter centralité forte a des chances d'être centrale dans le graphe (nombreux chemins passent par elles) et de s'apparenter localement à un isthme.

2.3.2 Extraction d'un squelette du graphe

Extraire un squelette du graphe connexe revient à trouver un graphe simplifié (Reid, Fan et al. 2004) ou un arbre possédant certaines propriétés intéressantes du graphe (Botafogo, Rivlin et al. 1992; Huang, P. et al. 1998).

2.3.2.1 Extraction d'un graphe squelette

L'algorithme FADE (Quigley et Eades 2000) basé sur un clustering géométrique, propose l'extraction de divers squelettes d'un graphe suivant le niveau de détail choisi (Figure 60). Cette technique s'appuie sur un calcul géométrique à partir du placement.

La technique peut être satisfaisante pour un graphe « arboré » (Figure 60). Elle est cependant mal adaptée à la simplification de réseaux « sans échelle » très denses (de type : YEAST ou co-auteurs). En effet, le placement de ces graphes dépend fortement du placement initial.

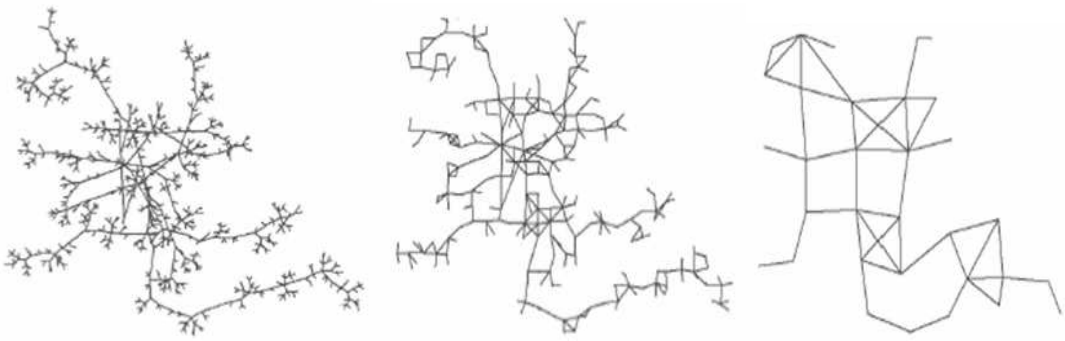


Figure 60 – (a) graphe initial (b) filtrage de niveau 1 (c) filtrage de niveau 2

2.3.2.2 Extraction d'un arbre squelette

Dans le cas d'un graphe connexe « sans échelle » à comportement « petit monde », il peut être intéressant d'extraire un arbre couvrant ayant également une distribution des degrés

suivant une loi de puissance. Une technique a été proposée (Kim, Noh et al. 2004) pour construire un tel squelette appelé aussi noyau de communication du graphe :

Définition 58. Le noyau de communication est un arbre couvrant dont l'ensemble des arêtes maximise la somme des indices d'inter centralité.

Algorithme : À l'étape 1, l'arbre T_1 est constitué de l'arête de plus fort indice. A l'étape k, l'arête adjacente de plus fort indice ne formant pas de cycle est ajoutée à l'arbre T_{k-1}.

L'arbre couvrant obtenu est « sans échelle » (Kim, Noh et al. 2004). Nous obtenons le même type de résultat empiriquement en utilisant non pas l'indice d'inter centralité mais le degré. Il s'ensuit un gain de performance par économie du calcul de l'indice d'inter centralité.

Exemple 31. Nous présentons en Figure 61, le graphe des co-auteurs du LIRMM et l'arbre couvrant associé (en utilisant le degré). Graphe initial et arbre couvrant ont une distribution des degrés suivant une loi de puissance (Figure 62).

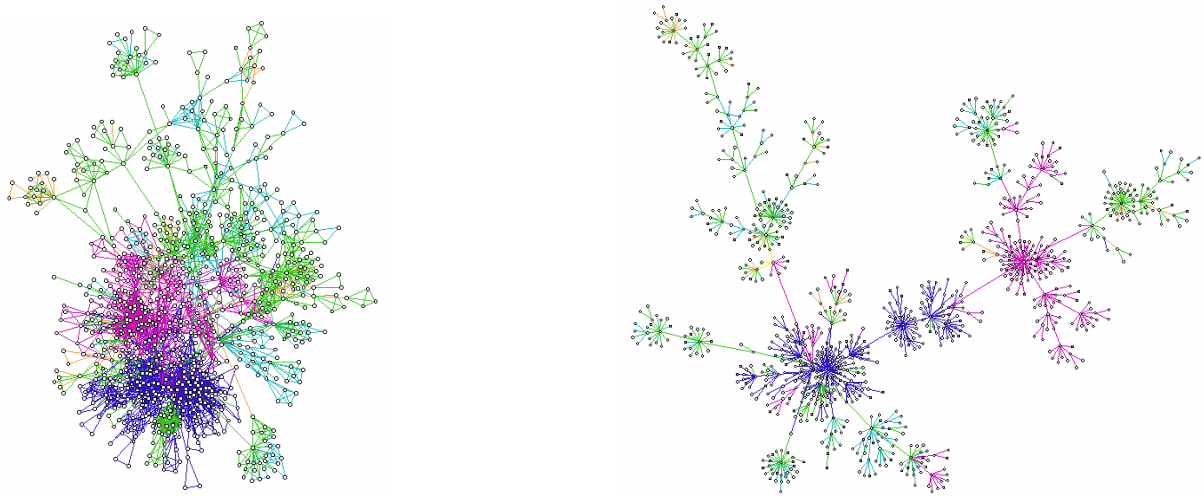


Figure 61 – (a) graphe G des co-auteurs du LIRMM (b) arbre couvrant associé T

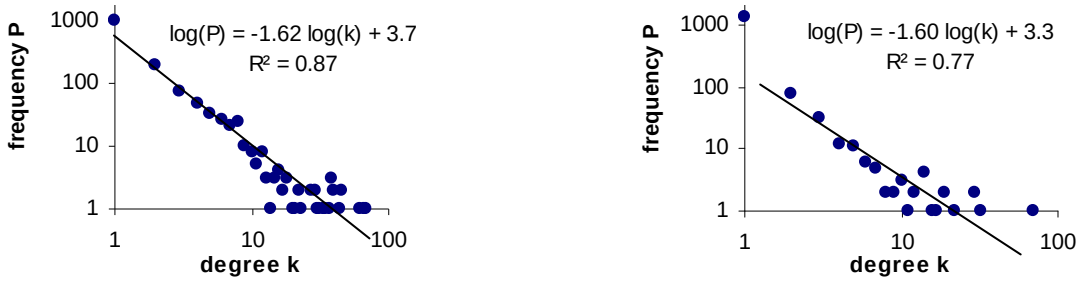


Figure 62 – distribution des degrés - (a) graphe initial, λ = 1.60 (b) arbre couvrant, λ = 1.62

2.4 Filtrage contextuel

Les techniques de filtrage décrites précédemment sont globales. Il peut être cependant intéressant d'avoir différentes perspectives d'un même graphe dépendant du focus choisi. Nous présentons dans cette section des techniques prenant en compte un focus utilisateur.

Une technique de filtrage contextuelle a été introduite (Furnas 1986) appelée « fisheye généralisé » : considérant un focus donné, le degré d'intérêt (degree of interest : DOI) d'un nœud est défini par son importance a priori et sa distance au focus. La technique de filtrage contextuelle consiste alors à visualiser les nœuds de DOI important. Ce qui permet de relativiser l'importance d'un nœud selon sa distance au focus. Ainsi, un nœud peu important mais proche du focus aura plus de poids qu'un nœud de la même importance mais éloigné du focus. De même un nœud lointain mais important aura peut-être moins de poids qu'un nœud moins important mais proche du focus.

L'objectif du filtre de Furnas est d'assurer une vue détaillée autour du focus tout en gardant le contexte du graphe. Ce filtre sémantique est différent du fisheye graphique de Sarkar (Sarkar et Brown 1992) qui propose une déformation géométrique de la vue autour du focus sans supprimer de nœuds.

Le fisheye généralisé de Furnas a été utilisé essentiellement pour visualiser des listes ou des arbres, mais peu de travaux se sont intéressés au filtrage sémantique de graphes. Cela s'explique par une difficulté d'appréhension de la suppression et de l'ajout de nœuds lors d'un changement de focus.

Une double technique utilisant un fisheye sémantique et un fisheye géométrique a été développée (Schaffer, Zhenping et al. 1996) pour faciliter la navigation d'un graphe clusterisé. Dans le graphe (Figure 63a), l'utilisateur peut sélectionner un cluster focus pour l'agrandir et avoir un niveau de détail supérieur (Figure 63b ou c). Les autres clusters sont alors réduits géométriquement ou sémantiquement.

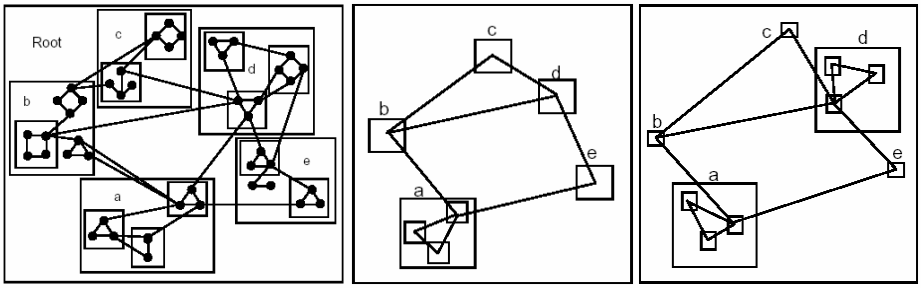


Figure 63 – (a) graphe clusterisé (b) focus a (c) focus a et d

Une technique de filtrage sémantique de graphe a été récemment proposée (van Ham et van Wijk 2004). Elle part d'un placement initial du graphe optimisé (Figure 64a) puis construit un partitionnement hiérarchique géométrique du graphe (section 3.3.1), groupant les nœuds en fonction de leur proximité dans le plan (Figure 64b). Lorsqu'on choisit un focus, les clusters proches du focus sont décomposés en sous clusters de degrés d'abstraction (DOA) appropriés (Figure 64c). Cette technique donne des résultats intéressants dans le cas de graphes « petit monde ».

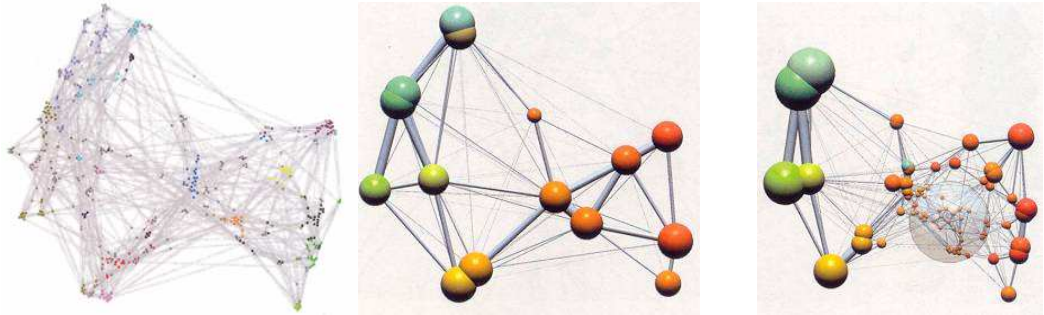


Figure 64 – (a) graphe placé avec modèle de forces (b) graphe clusterisé (c) + focus

2.5 Nouvelle technique d'extraction conjointe de motifs et squelette

Nous avons présenté, section 2.3, diverses techniques permettant l'extraction de motifs d'un graphe. Cependant ces techniques ne fournissent pas d'organisation structurée : les motifs sont souvent noyés dans un réseau d'interactions complexe ou totalement déconnectés.

Nous proposons une nouvelle technique de filtrage produisant un graphe « arboré » (voir section 1.2.5). Ce filtrage, dépendant d'un focus, est contextuel (section 2.4).

Cette technique est appelée « filtrage de contour » (« outline filtering ») car elle permet de détacher le contour du graphe. Le graphe « arboré » obtenu présente des caractéristiques analysées en section 2.5.2. Diverses variantes de l'algorithme sont proposées en section 2.5.3.

Cette section a fait l'objet d'une publication (Boutin, Thièvre et al. 2005).

2.5.1 Algorithme

Soit G un graphe connexe. Soit λ un entier choisi comme valeur seuil du filtre. Le filtre consiste à transformer G en un graphe « arboré » noté G_λ. Nous décrivons les deux étapes :

- **Construction de l'arbre couvrant T** : partant du nœud de plus fort degré dans G, T est construit itérativement par ajout de nœuds adjacents, prioritairement selon leur degré. Ces nœuds sont alors connectés à leur voisin de plus fort degré dans l'arbre.

- **Suppression des arêtes longues du graphe** : la taille d'une arête de G (autre qu'une arête d'arbre) est définie par la distance, dans l'arbre T, entre ses deux sommets. Les arêtes de taille supérieure à λ (dites longues) sont supprimées de G.

Remarque 6. Plutôt que de considérer le degré dans le calcul de l'arbre couvrant, on peut considérer une autre mesure de centralité comme l'indice BC (Brandes 2001). Les deux indices donnent des résultats très voisins (Figure 167). Comme nous l'avons évoqué précédemment (section 2.2.3), le coût de calcul de BC est élevé. Aussi utiliserons nous par la suite le degré.

2.5.2 Propriétés

Propriété 19. La complexité globale est en $O(m \log n)$, si on considère que le graphe G a n nœuds et m arêtes et que l'arbre couvrant T est construit à partir du degré.

Preuve :

- L'arbre couvrant est calculé en $O(n \log n)$ puisque basé sur le tri de n indices.
- La distance dans l'arbre entre deux nœuds est calculée en $O(\log n)$ car basée sur la détermination du plus petit ancêtre commun dans l'arbre. Donc la taille des m arêtes est calculée en $O(m \log n)$.

Propriété 20. L'algorithme présente d'intéressantes caractéristiques

- Si le graphe initial est « sans échelle », l'arbre couvrant associé l'est aussi.
- Si le graphe initial est « petit monde », le graphe filtré l'est également.
- Les degrés d'un nœud dans l'arbre et dans le graphe sont corrélés.
- Par construction, le graphe filtré est « arboré » (voir section 1.2.5). C'est un arbre de composantes facilement navigable.
- Contrairement aux techniques présentées précédemment, le filtre proposé est idempotent. Cela signifie qu'il donne le même résultat sur un graphe si on l'applique une ou plusieurs fois en utilisant le même seuil.
- Le graphe est rapidement dessiné par un algorithme de dessin basé sur un modèle de forces : il suffit de placer au préalable l'arbre couvrant puis de lancer l'algorithme de placement en utilisant toutes les arêtes. Ainsi, l'« allure » du graphe filtré est voisine de celle de l'arbre couvrant associé.

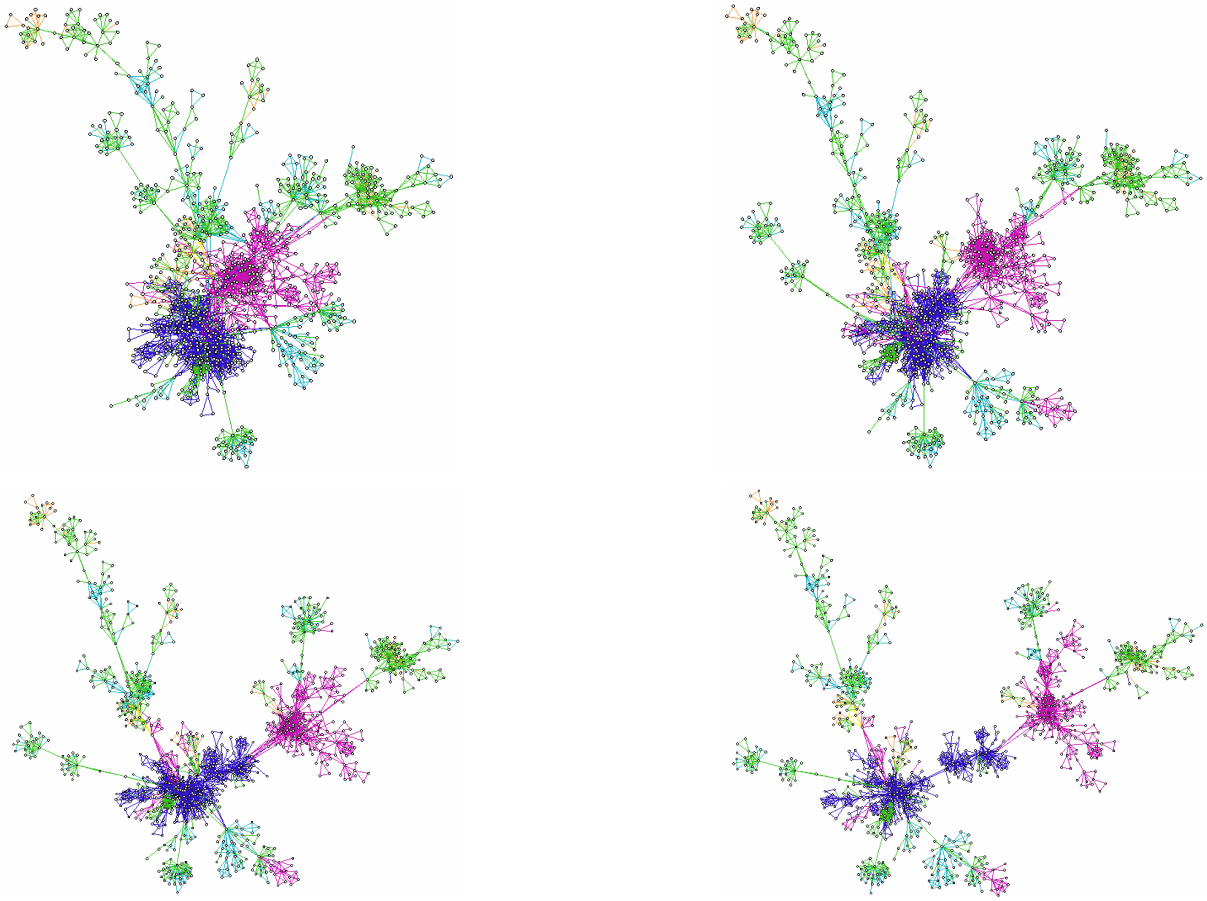


Figure 65 – graphes filtrés (a) G_8 (b) G_6 (c) G_4 (d) G_2

Exemple 32. L'algorithme est appliqué sur le graphe de co-auteurs (Figure 61a) :

– Dans l'arbre couvrant Figure 61b, les départements du LIRMM, identifiés par une couleur d'arête spécifique, sont naturellement regroupés.

– L'application de divers filtres de seuils λ = 8, 6, 4 puis 2 produit les vues Figure 65. Plus λ est faible, plus la séparation des départements est nette : Le département d'informatique est organisé en branches. Puis les départements de robotique (bleu) et micro électronique (violet) sont séparés. Enfin micro électronique est scindé en deux.

- Les graphes résultants sont « sans échelle » avec β proche de 1,6 (Figure 62).
- G, G_2, G_4, G_6 et G_8 ont un coefficient de clustering proche.
- A partir de λ = 6, le graphe filtré est « arboré » (Figure 65b) : sa structure s'apparente à celle de l'arbre couvrant (Figure 61b) contrairement à celle de G_8 (Figure 65a).
- **Temps de calcul** : Le graphe est filtré en 5 secondes (2,66 Mhz, 512 Mo RAM)

Exemple 33. Graphe des publications du LIRMM entre 2000 et 2004 (Figure 66) :

- 2 211 nœuds, 53 722 arêtes
- Chaque nœud représente une publication.
- Un lien est établi entre deux publications d'un même auteur.
- Les amas sont séparés en appliquant le filtre des arêtes de longueur supérieure à 2. Trois vues sont proposées du graphe filtré G_2: la vue Figure 67 est obtenue comme superposition des vues Figure 68 (que les arêtes) et Figure 69 (que les nœuds). Les départements sont clairement séparés (Figure 68). De plus, chaque amas correspond aux publications d'un auteur majeur et de son groupe de thésards (Figure 69).

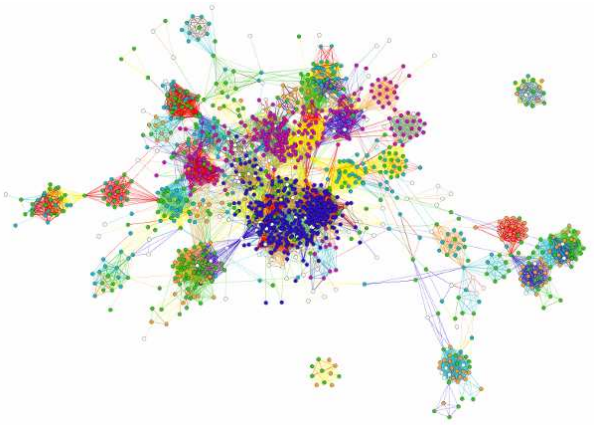


Figure 66 – graphe G des co-publications (nœuds) et auteurs (arêtes) du LIRMM

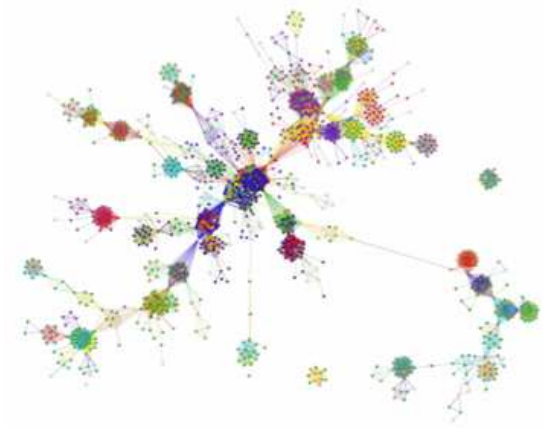


Figure 67 – G_2 : filtrage des arêtes de longueur > 2 dans le graphe des co-publications

- **Temps de calcul** : Le graphe est filtré en 50 secondes (2,66 Mhz, 512 Mo RAM)

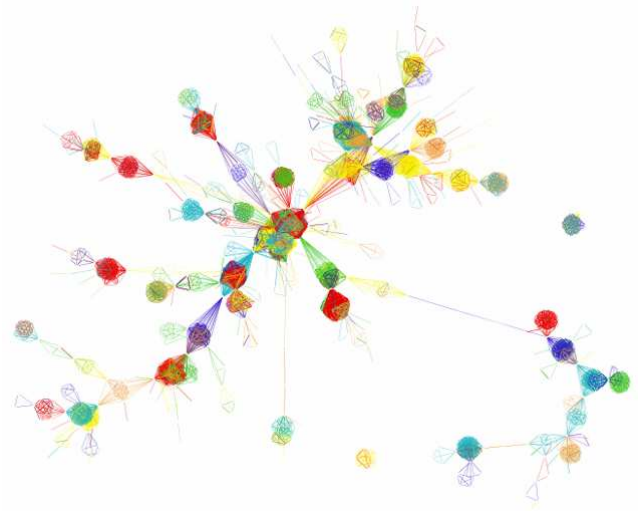


Figure 68 – graphe de co-publications filtré G_2 – nœuds non visualisés

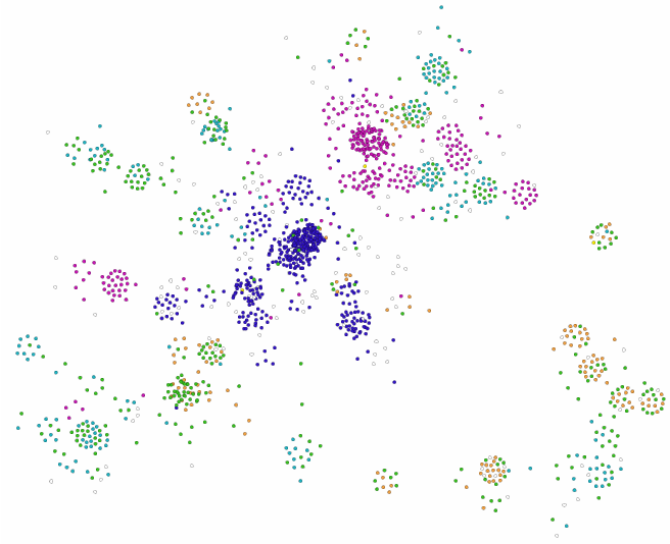


Figure 69 – graphe de co-publications filtré G_2 – arêtes non visualisés

Exemple 34. Le graphe filtré G_4 en Figure 70 est très proche visuellement du graphe G_2 Figure 68. Le noyau central est simplement un peu plus dense.

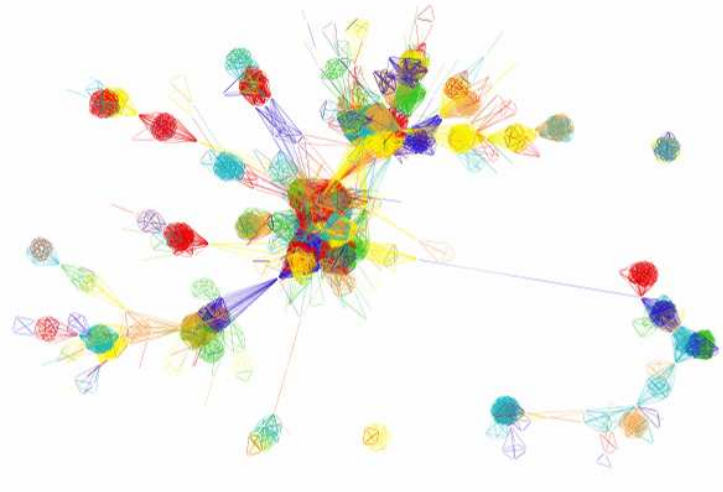


Figure 70 – graphe de co-publications filtré G_4 – nœuds non visualisés

Exemple 35. En appliquant l'algorithme avec $\lambda = 2$ sur le graphe des sympathies on obtient le graphe Figure 71. Ce graphe partage une composante avec le graphe d'amitiés Figure 28 (composante en bas à droite). Pour le reste, les groupes d'amis diffèrent des groupes des sympathies.

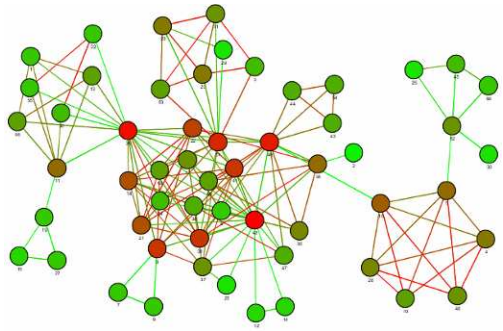


Figure 71 – G_2 : filtrage des arêtes de taille > 2 dans le graphe des sympathies

2.5.3 Variantes

2.5.3.1 Variante sans extraction d'arbre couvrant

La taille d'une arête peut être définie directement dans un graphe non orienté comme la longueur du plus petit cycle non élémentaire (s'il existe) contenant l'arête. Si l'arête n'appartient à aucun cycle élémentaire, l'arête est un isthme (de taille infinie).

La suppression d'arêtes de taille supérieure à une valeur limite peut permettre de révéler des motifs dans certains graphes. En particulier en considérant une taille limite de 3, on ne conserve que les arêtes appartenant à des triangles. Ces arêtes sont qualifiées de courte portée (Kleinberg 1999).

Exemple 36. Le graphe d'école (Figure 72a) se prête bien à l'identification de composantes (Figure 72b) car les composantes se séparent naturellement.

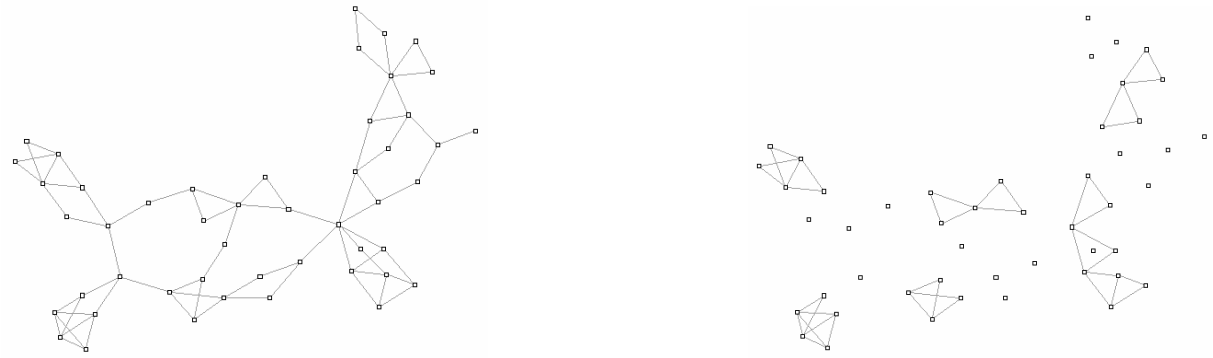


Figure 72 – (a) graphe d'école (b) cycle de longueur 3

Cette variante est souvent mal adaptée au traitement de gros graphes « sans échelle ». En effet, le graphe résultant présente alors un amas de motifs (triangles) difficilement discernables. Aucune structure a priori n'est extraite du graphe.

Contrairement à la technique principale présentée en section 2.5.1, la variante ne permet pas d'organiser le graphe en un « arbre de motifs ». Cette variante peut être toutefois utilisée en accompagnement de la technique principale (pré ou post traitement).

Exemple 37. Figure 73 présente le graphe de co-auteurs du LIRMM d'où n'ont été retenues que les arêtes appartenant à un cycle de longueur 3 (triangle). Le noyau ne parvient pas à se décomposer et reste très dense.

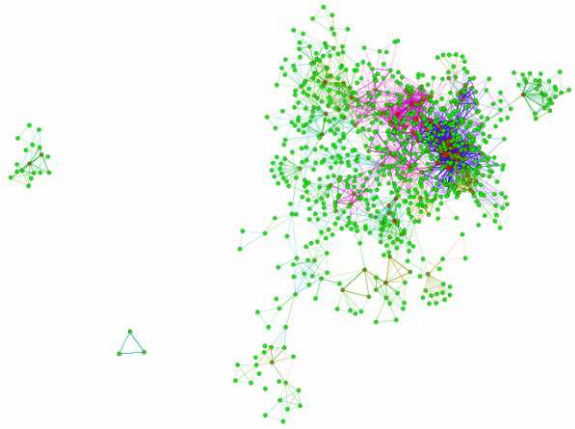


Figure 73 – cycles de longueur 3 dans le graphe de co-auteurs du LIRMM

2.5.3.2 Arbre couvrant reposant sur un focus utilisateur

Une variante de l'algorithme principal décrite en section 2.5.1, consiste à prendre en compte le choix d'un nœud focus pour la construction de l'arbre couvrant. Le principe consiste à calculer un arbre couvrant ayant pour racine le focus. Chaque niveau k de l'arbre contient les nœuds à distance k du nœud focus. Chaque nœud sur la couche k>0 est lié au nœud de plus fort indice (degré par exemple) sur la couche k-1.

Propriété 21. En utilisant ce nouvel arbre couvrant on obtient également un graphe filtré « arboré », présentant une organisation structurée autour du focus. Le graphe filtré est noté G'_λ pour le différentier du graphe filtré G_λ obtenu avec l'algorithme général (section 2.5.1).

Exemple 38. Nous présentons (Figure 74) le graphe du LIRMM filtré G'_3 à partir de deux focus (deux auteurs). Les principales composantes se détachent clairement.

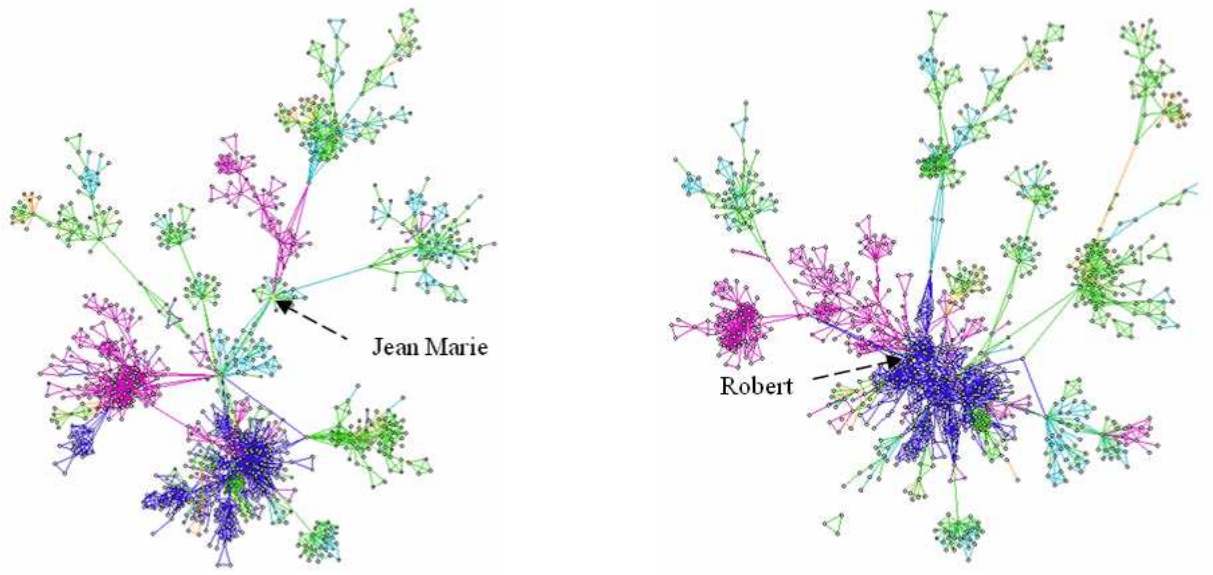


Figure 74 – G'_3 focus (a) Jean-Marie, (b) Robert

2.5.3.3 Méthodes mixtes

Pour tirer pleinement partie des techniques de filtrage, il convient de savoir les utiliser conjointement selon le type de graphe traité et la nature des propriétés étudiées.

Exemple 39. Considérons, par exemple, le graphe d'interactions de protéines en ne gardant que les arêtes appartenant à un triangle (Figure 75a). Pour faire apparaître des clusters, filtrons les arêtes de forces inférieures à 0,5 (Figure 75b). Enfin pour obtenir un arbre de clusters nous appliquons le filtrage G_3 (Figure 75c).

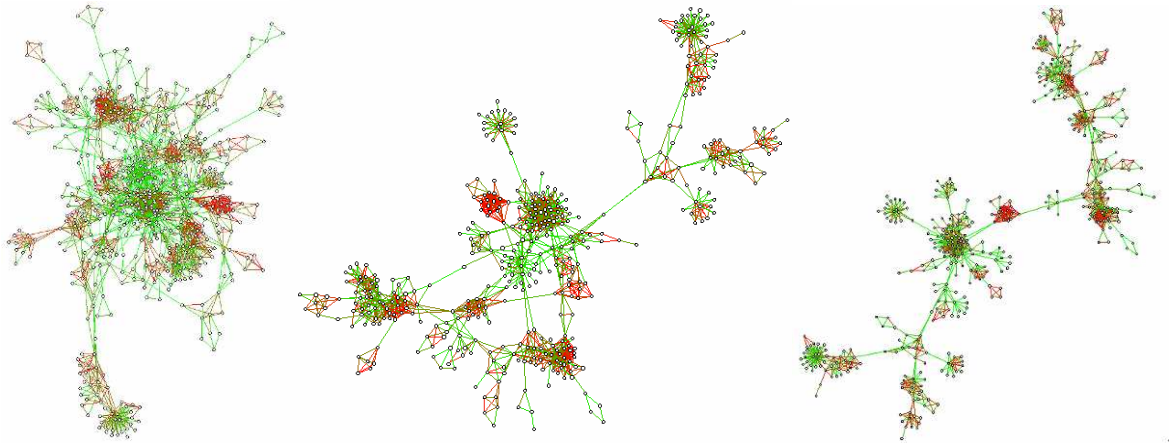


Figure 75 – (a) triangulation (b) suppression des arêtes de force < 0,5 (c) G_3

Exemple 40. Nous présentons en Figure 76, le graphe de citation d'InfoVis'Contest après extraction d'un arbre couvrant et suppression des arêtes de taille supérieure à deux dans l'arbre (graphe G_2). Les nœuds à fort coefficient de clustering local sont peints en rouge.

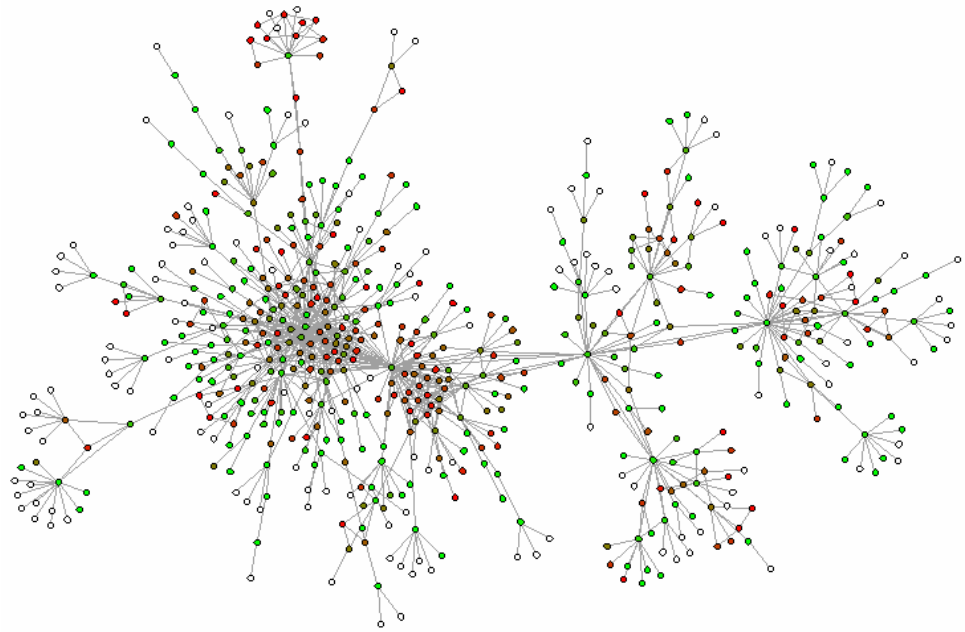


Figure 76 – filtrage G_2 du graphe de citations d'InfoVis

Exemple 41. Pour simplifier la vue, supprimons les nœuds de degré 0 et 1 (Figure 77). Ce graphe présente deux principaux noeuds d'articulation. Le premier correspond à l'article de Furnas : « Generalized Fisheye Views » (Furnas 1986). Il sépare deux composantes, l'une traitant de visualisation, l'autre d'interfaces graphiques. Le second correspond à l'article « The Eyes Have It : a Task by Data Type Taxonomy for Information Visualizations » (Shneiderman 1996) lui même lié à divers noeuds d'articulation.

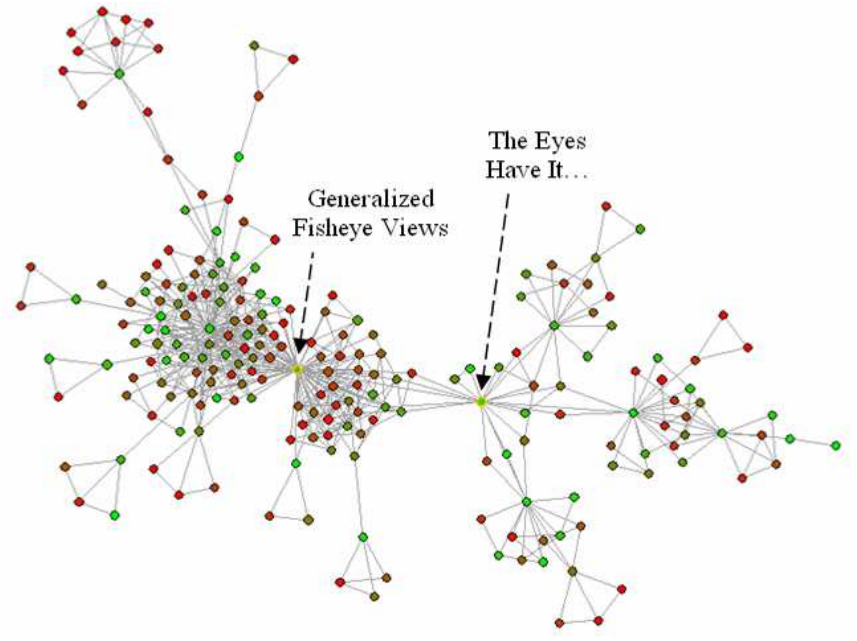


Figure 77 – G_2 – filtrage des noeuds de degré 0 et 1 – InfoVis Contest

En conclusion, nous avons proposé dans cette section, deux nouvelles techniques de filtrage construisant un « graphe arboré » à partir d'un arbre couvrant et de l'ajout d'arêtes « courtes ». Cette structure est facilement visualisable avec un algorithme de dessin de graphe basé sur un modèle de forces.

Toutefois il est souhaitable d'utiliser en plus une technique adaptée de partitionnement pour visualiser de gros graphes. Nous en proposons une, au chapitre suivant, tirant pleinement partie de cette structure « arborée ».

« L'ordre est le plaisir de la raison, mais le désordre est le délice de l'imagination. »

Paul Claudel
(Le soulier de satin)

Chapitre 3 Partitionnement de graphe

Nous recensons en section 3.1.1 différentes structures de partitionnement. Puis nous en introduisons de nouvelles en section 3.1.2 qui seront utilisées par la suite. Nous proposons une classification des techniques de partitionnement : géométrique (section 3.2), basé sur une métrique (section 3.3), structurel (section 3.4) et autres (section 3.5). Nous décrivons enfin de nouvelles techniques de partitionnement basées sur un focus en section 3.6.

3.1 Structures de partitionnement

Nous recensons les structures de partitionnement de graphes existantes (Sugiyama, Tagawa et al. 1981; Sugiyama et Misue 1991; Eades 1996; Brockenauer et Cornelsen 2001). Nous en proposons également de nouvelles qui seront utilisées dans ce document.

3.1.1 Etat de l'art

3.1.1.1 Partitionnement simple

Définition 59. Une **k-partition** d'un ensemble V est définie par une famille de sous ensembles $\{V_1,\ldots, V_k\}$ vérifiant : $\bigcup_{i=1}^{k} V_i = V$ et $V_i \cap V_j = \varnothing, \forall i \neq j$

Définition 60. Un **graphe clusterisé** est un graphe G = (V, E) pour lequel on dispose d'une partition $\{V_1,\ldots, V_k\}$ de l'ensemble des sommets où les V_i sont des **clusters**.

Définition 61. Soit un graphe G = (V, E) clusterisé et une partition $\{V_1,\ldots, V_k\}$ des sommets, toute arête reliant deux clusters V_i et V_j est appelée **pont** ou **coupe** (cut edge).

Définition 62. Soit $\{V_1,\ldots, V_k\}$ une k-partition de l'ensemble des sommets de G. Le **graphe quotient** $\Gamma = (V, E)$ est défini par :

- $V = \{V_1,\ldots,V_k\}$ où chaque sous ensemble V_i est considéré comme un noeud de Γ.
- V_i et V_j sont adjacents dans Γ si et seulement si les ensembles V_i et V_j contiennent deux sommets v_i dans V_i et v_j dans V_j qui sont adjacents dans G.

Exemple 42. La Figure 78a, présente un graphe clusterisé. La Figure 78b, présente le graphe quotient associé.

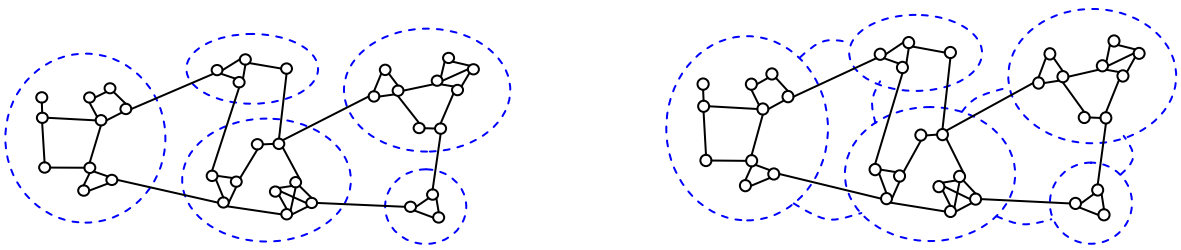


Figure 78 – (a) graphe clusterisé (b) graphe quotient associé

3.1.1.2 Graphe clusterisé hiérarchique

Définition 63. Un **graphe clusterisé hiérarchique** est défini (Eades 1996; Brockenauer et Cornelsen 2001) par un graphe G = (V, E) et une arborescence T tels que :

- Les feuilles de T sont les sommets de G.
- Chaque autre nœud de T représente un cluster de sommets de G (en bleu et orange).
- L'arborescence T (appelé arborescence d'inclusion) décrit une inclusion de clusters de sorte que chaque cluster (à l'exception des feuilles) contient ses clusters fils.
- Il n'y a pas de recouvrement de clusters de même niveau.

Exemple 43. La Figure 79 présente une vue 2D d'un graphe clusterisé à 2 niveaux. Une vue 3D est proposée en Figure 80.

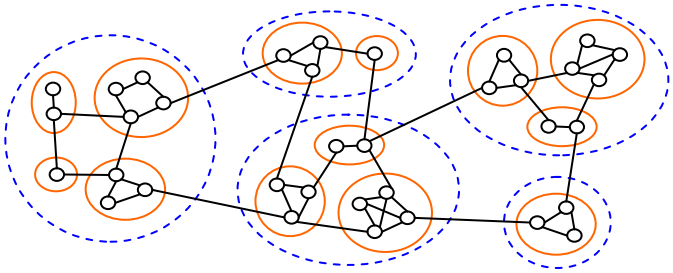


Figure 79 – graphe clusterisé hiérarchique 2D

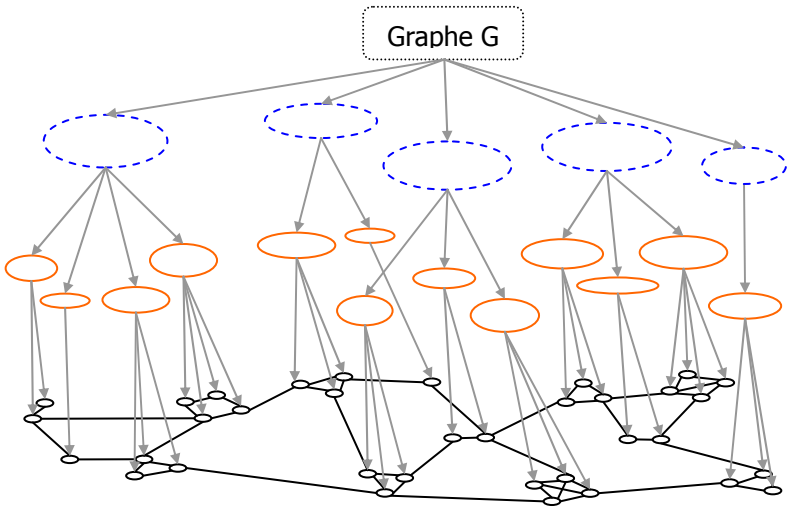


Figure 80 – graphe clusterisé hiérarchique 3D

3.1.1.3 Graphe composé

Définition 64. Un **graphe composé** est défini (Sugiyama et Misue 1991) par la donnée d'un triplet (V, A, I), tel que $G_A = (V, A)$ est un graphe d'adjacence et $G_I = (V, I)$ un graphe d'inclusion orienté acyclique (DAG) vérifiant :

- G_I décrit une inclusion de clusters telle que chaque cluster contient ses clusters fils.
- G_A est un graphe d'adjacence de clusters (et de sommets).

Cas particulier : Un graphe clusterisé hiérarchique est un graphe composé tel que :

- G_I est une arborescence d'inclusion (c'est-à-dire un DAG très particulier).
- Les arêtes de A relient uniquement les feuilles de G_I (et non tous les nœuds).

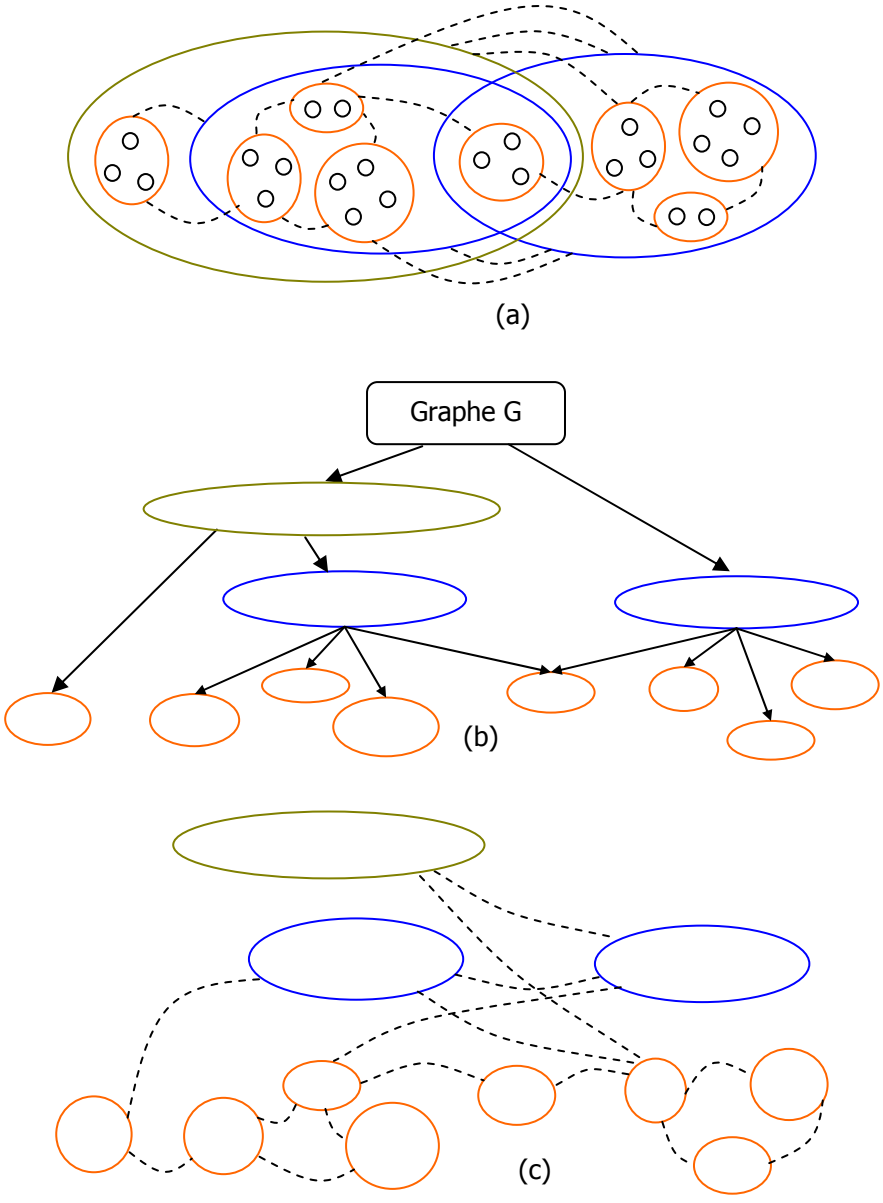


Figure 81 – (a) graphe composé (b) DAG d'inclusion (c) graphe d'adjacence de clusters

Exemple 44. Nous présentons en Figure 81, différentes vues d'un graphe composé :

- (a) : une vue globale en 2D, peu lisible.
- (b) : une vue 3D du DAG d'inclusion.
- (c) : une vue du graphe d'adjacence comprenant éventuellement des relations entre clusters de niveaux différents.

Dans l'exemple proposé, une arête d'adjacence est construite entre deux clusters A et B s'ils possèdent respectivement des nœuds adjacents v_A et v_B tels que $v_A \notin B$ et $v_B \notin A$. Un autre choix aurait pu être fait.

Un graphe composé est une structure complexe (voir Figure 81). Nous utiliserons par la suite des graphes composés particuliers, plus simples, ayant de bonnes propriétés décrites en section 3.1.2.

3.1.2 Nouvelles structures de partitionnement

Nous introduisons diverses structures de partitionnement que nous utiliserons dans la suite de ce document. Nous pensons qu'elles sont assez générales pour pouvoir être utilisées dans d'autres contextes.

3.1.2.1 Arbre d'ensembles

Nous définissons la structure d'arbre d'ensembles (Boutin et Hascoët 2003).

Définition 65. Un **arbre d'ensembles** est défini par un graphe quotient dont les arêtes entre ensembles définissent un arbre

Exemple 45. La Figure 82 présente un arbre d'ensembles. Les arêtes entre ensembles sont induites des arêtes entre nœuds.

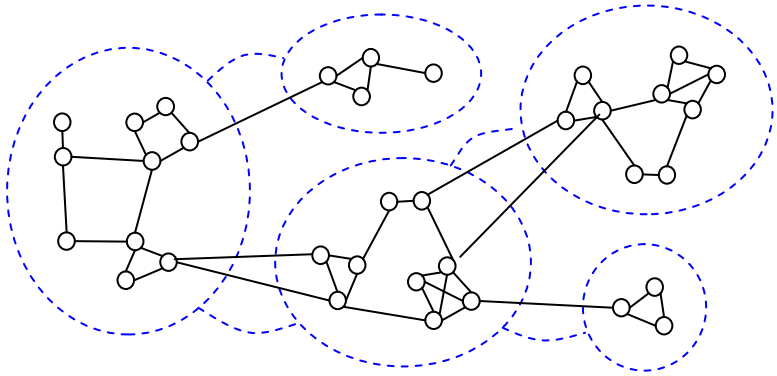


Figure 82 – arbre de clusters

Nous introduirons par la suite des exemples d'arbres d'ensembles : les arbres de clusters (section 3.6.1) et les arbres de silhouettes (section 3.6.2). Nous proposerons des algorithmes de construction et de placement de ces arbres.

3.1.2.2 Arbre composé simple

Nous introduisons une version multi-échelles de l'arbre d'ensembles (Boutin et Hascoët 2004). Il s'agit d'un graphe composé particulier nommé arbre composé simple :

Définition 66. Un **arbre composé simple** est défini par un graphe composé tel que :

- Les ensembles appartiennent à un arbre d'inclusion dont la racine est le graphe. Il n'y a pas de recouvrement contrairement aux graphes composés basés sur un DAG.
- Les seules relations d'adjacence sont entre ensembles de même niveau.
- Les ensembles englobants (au niveau 1 dans l'arbre d'inclusion) appartiennent à un arbre d'adjacence.

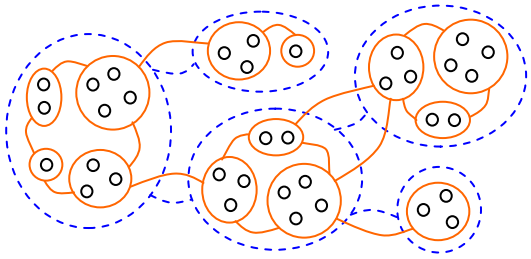


Figure 83 – arbre composé simple

Exemple 46. La Figure 83 montre l'arbre d'adjacence d'ensembles englobants (pointillés).

Les ensembles englobants peuvent être facilement placés avec un algorithme de dessin d'arbre, par contre ce n'est pas le cas des autres clusters.

3.1.2.3 Arbre composé multi niveaux

Nous introduisons enfin une structure particulière d'arbre composé (Boutin, Thièvre et al. 2005). D'autres structures ont été introduites pour la représentation d'arbres réels (Godin et Caraglio 1996).

Définition 67. Un **arbre composé multi niveaux** est un arbre composé dont chaque niveau est constitué d'un arbre d'adjacence d'ensembles.

Exemple 47. En Figure 84, les ensembles orange (tout comme les bleus) constituent un arbre d'adjacence.

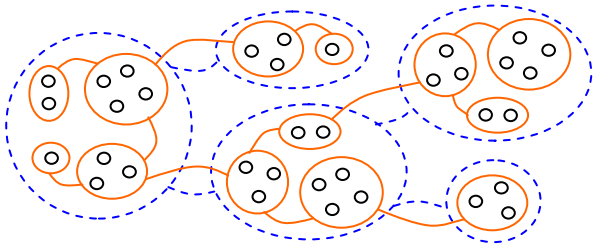


Figure 84 – arbre composé multi niveaux

Cette structure est simple à manipuler et à visualiser car il s'agit d'un arbre contenant des arbres. Elle sera largement utilisée par la suite (section 4.3.3).

3.2 Partitionnement géométrique

Les techniques de partitionnement géométriques ont été introduites en analyse de données et data mining (Jain et Dubes 1988; Jain, Murty et al. 1999). Elles proposent une classification de points dans un espace donné : droite (clustering temporel), plan ou sphère (clustering physique), ou plus généralement espace vectoriel euclidien. Par exemple, des documents textuels peuvent être définis par des vecteurs dans une base de mots clés (Steinbach, Karypis et al. 2000). Le placement de points dans un espace euclidien assure le calcul de barycentres et du produit scalaire utiles au partitionnement géométrique.

Procéder à un partitionnement géométrique de graphe suppose donc que ce graphe soit initialement placé dans un espace vectoriel le plus souvent euclidien. Différentes techniques de placement sont décrites en section 3.2.1. Pour des approches non euclidiennes voir (Lamping, Rao et al. 1995; Munzner 1998)

Une fois placé, le graphe peut être partitionné en utilisant une technique géométrique issue de l'analyse de données. Cette technique peut être adaptée au domaine des graphes en prenant en compte les arêtes du graphe et en cherchant par exemple à minimiser le nombre d'arêtes entre clusters. Nous décrirons dans cette section les principales techniques de partitionnement géométrique de graphe.

3.2.1 Placement du graphe dans un espace euclidien

3.2.1.1 Plongement naturel

Certains graphes sont naturellement placés dans un espace euclidien. Il s'agit de graphes incluant des informations supplémentaires sur les nœuds telles que :

- La position des sommets dans un espace physique : par exemple, le graphe ayant pour sommets les aéroports européens et pour arêtes les routes aériennes entre eux, possède un placement sur la sphère.
- La position des sommets dans un espace euclidien : si on considère le graphe du Web, on peut associer à toute page ses coordonnées dans une base de mots-clés.
- Tracé optimal des arêtes d'un graphe connaissant le placement des nœuds. On pourra consulter (Jourdan 2004) pour la représentation de voies métaboliques.

3.2.1.2 Réduction du nombre de facteurs – méthode factorielle

Avant d'appliquer un partitionnement on peut réduire le nombre de dimensions de l'espace vectoriel en utilisant une technique factorielle comme l'ACP (analyse en composantes principales). Le principe est de calculer un nombre réduit de vecteurs orthogonaux, appelés facteurs principaux, définissant une base de l'espace factoriel (Bouroche et Saporta 2002; Lebart, Morineau et al. 2005) et de projeter les points sur cet espace.

Définition 68. Le plan factoriel de dimension k est le sous espace de dimension k tel que la somme des carrés des distances entre les individus et leur projection sur l'espace factoriel soit minimale.

Propriété 22. C'est également l'espace de dimension k tel que les projections des points aient la dispersion (l'inertie ou la variance) maximale (voir Figure 85)

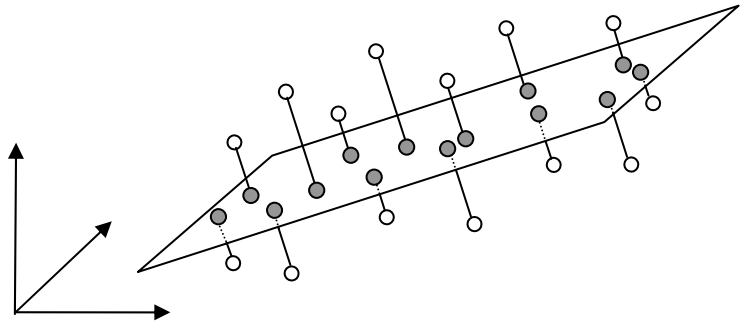


Figure 85 – plan factoriel – points et leurs projetés

Théorème 2. Les facteurs principaux v_1, …, v_p sont obtenus comme vecteurs propres de MV (M : métrique, V : matrice de variances) associés aux valeurs propres écrites en ordre décroissant : λ_1, …, λ_p. Pour construire le plan factoriel d'ordre k on utilise v_1, …, v_k.

La projection des points dans un espace à deux ou trois dimensions permet une représentation graphique du nuage et l'identification visuelle éventuelle de clusters.

3.2.1.3 Placement de graphes utilisant un modèle de forces

Dans les techniques de placement de graphes utilisant des algorithmes basés sur un modèle de forces, les nœuds sont initialement placés dans l'espace (de façon aléatoire ou en utilisant une heuristique particulière) puis soumis à deux types de forces :

- Une force d'attraction f qui attire deux sommets adjacents.
- Une force de répulsion g repoussant deux sommets en fonction de leur distance.

Définition 69. Modèle d'énergie proposé par (Fruchterman et Reingold 1991) défini par :

$$\sum_{\{u,v\}\in E} f\left(\|p_v - p_u\|\right) + \sum_{\{u,v\}\in V^2} g\left(\|p_v - p_u\|\right)$$

où p_u et p_v représente les vecteurs position des nœuds u et v dans l'espace.

Différents modèles ont été proposés. Il a été prouvé que le modèle énergétique LinLog (Noack 2003) est le plus adapté pour la séparation de clusters. La distance entre deux clusters est (approximativement) inversement proportionnelle au nombre d'arêtes qui les séparent.

Après avoir décrit diverses techniques de placement de graphe, nous proposons de recenser les techniques de partitionnement géométrique les plus courantes.

3.2.2 Segmentation géométrique par un hyperplan

La technique de segmentation décrite (Elsner 1997; Chamberlain 1998) consiste à déterminer un hyperplan perpendiculaire à un axe qui partage l'ensemble des sommets en deux ensembles égaux en minimisant le nombre de ponts (coupes) entre composantes. Dans un espace à n dimensions, n hyperplans sont sélectionnés et comparés.

Les sous graphes sont ensuite partitionnés récursivement en considérant la même technique pour chaque sous graphe.

Une autre technique est proposée (Duncan 2000) pour un partage équilibré des nœuds.

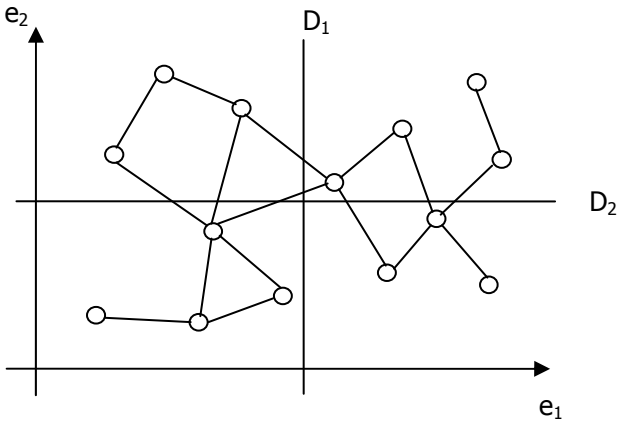


Figure 86 – segmentation de G par deux hyperplans D_1 et D_2

Exemple 48. En Figure 86, D_1 est choisi pour segmenter le graphe G car il intercepte deux arêtes alors que D_2 en intercepte six.

3.2.3 Segmentation basée sur un calcul d'inertie

Plutôt que de choisir un hyperplan perpendiculaire à un axe de coordonnées, on le choisit perpendiculaire à un axe d'inertie (Elsner 1997). Cet axe passe par le centre de gravité du nuage de points. Il minimise le moment d'inertie. Il est déterminé de sorte que la somme des carrés des distances des points à l'axe soit minimale.

Exemple 49. En Figure 87, le plan perpendiculaire à l'axe d'inertie sépare le nuage en deux ensembles de points.

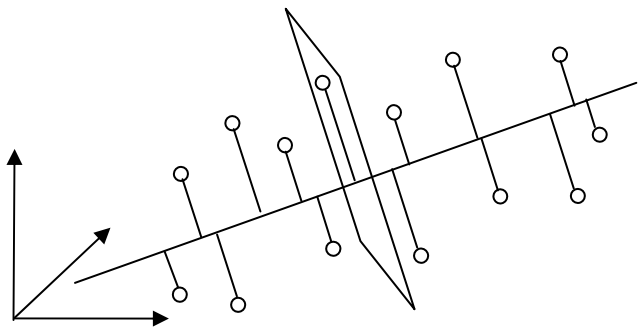


Figure 87 – axe d'inertie

3.2.4 Segmentation géométrique par cercle ou sphère

Les sommets du graphe initialement placés dans l'espace vectoriel IR^p sont projetés sur la sphère unité de IR^{p+1} par projection stéréographique à partir du pôle (Elsner 1997; Chamberlain 1998).

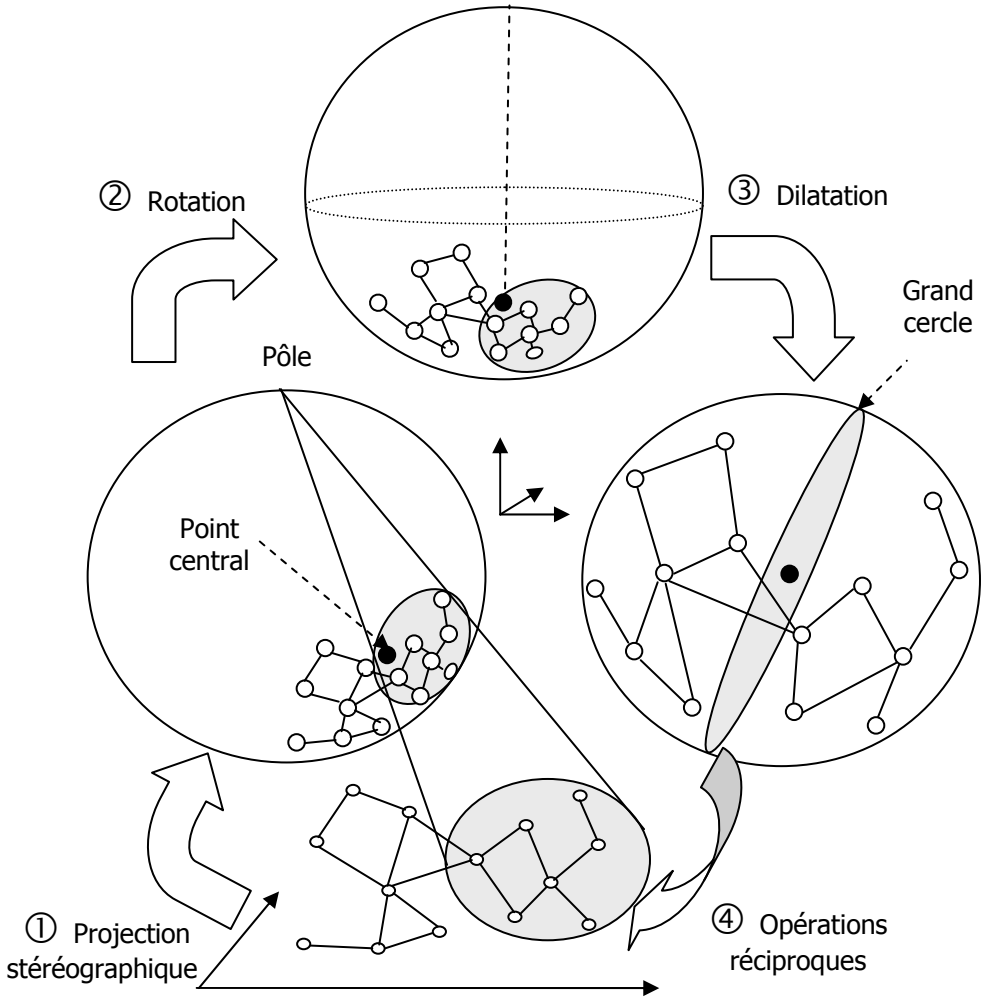


Figure 88 – segmentation de G par un cercle

Un point central est calculé pour l'ensemble des projetés. Il s'agit d'un point tel que tout hyperplan passant par ce point sépare le nuage en deux parties non vides (un tel point est moins sensible aux valeurs aberrantes qu'un point moyen). Le nuage est ensuite pivoté de façon à ce que le point central ait pour coordonnées (0,…, 0, r) dans IR^{p+1}. Puis le nuage est dilaté afin que le point central ait pour position l'origine. Un grand cercle (intersection de la sphère et d'un hyperplan passant par l'origine) est sélectionné qui sépare le nuage en deux ensembles en minimisant (par exemple) le nombre d'arêtes interceptées. En effectuant les opérations inverses (dilatation, rotation, projection stéréographique), on définit un cercle dans IR^p (ou une droite dans le cas dégénéré) qui partitionne l'ensemble de points.

Exemple 50. Les différentes étapes sont décrites à travers un exemple (voir Figure 88).

3.2.5 Méthode de partitionnement des centres mobiles

La méthode des centres mobiles « K-Means » (Alpert et Kahng 1995; Han et Karypis 2000; Lebart, Morineau et al. 2005) est utilisée pour obtenir une partition en k classes d'un ensemble de n points d'un espace vectoriel euclidien. Le critère retenu pour mesurer la qualité du partitionnement est la minimisation de l'inertie intra classe (pour les k classes). L'inertie d'une classe étant la moyenne des carrés des distances au barycentre de la classe. L'inertie intra classe globale est définie par $I = \Sigma I_j$ où I_j est l'inertie de la classe j.

Les différentes étapes de l'algorithme sont décrites ci-dessous :

- Etape 0 : tirage au sort de k centres mobiles.
- Etape 1 : regroupement des points autour du centre mobile le plus proche.
- Etape 2 : calcul du barycentre des points associés à chaque centre mobile.
- Etape 3 : retour à l'étape 1 en considérant les barycentres comme centres mobiles.

Remarque : la convergence de l'algorithme est assurée par le fait que l'inertie intra classe décroît à chaque itération de l'algorithme. Si on note I^m et I^{m+1} les inerties intra classes aux itérations m et m+1, on va montrer que $I^m \geq I^{m+1}$: la somme des carrés des distances des points d'une classe au centre mobile est supérieure ou égale à l'inertie de la classe (théorème de Huygens). Par ailleurs la composition d'une classe peut changer d'une itération à l'autre. Chaque point est associé au centre mobile le plus proche donc l'inertie intra classe diminue.

L'algorithme converge vers un optimal local dépendant du choix initial des centres mobiles. En répétant l'algorithme plusieurs fois en tirant au sort d'autres centres mobiles, on peut mettre en évidence des groupements stables (ensembles de points ayant toujours été affectés à une même classe dans chacune des partitions).

L'algorithme suppose initialement fixé le nombre k. Si le choix de k est parfois naturel, il est souvent arbitraire et le partitionnement peut alors ne pas expliquer la véritable structure géométrique des données. La détection de groupements stables peut toutefois aider à déterminer le nombre de classes du nuage.

Exemple 51. En Figure 89, on considère en (a) un nuage de points et deux centres mobiles aléatoires. Le graphe est partitionné en deux classes après quatre étapes. Les étapes (c) et (d) proposent le même partitionnement. Il s'agit d'une situation d'équilibre correspondant à la fin de l'algorithme.

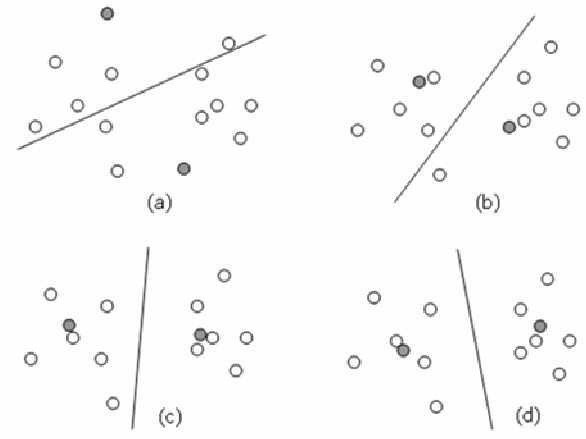


Figure 89 – méthode des centres mobiles

3.2.6 Partitionnement par suppression des arêtes « longues »

Nous proposons une technique géométrique consistant à filtrer les arêtes « longues », c'est-à-dire de distance géométrique importante.

Exemple 52. Nous proposons, en Figure 90a, la suppression de 1% des arêtes les plus longues du graphe des co-auteurs du LIRMM présenté en Figure 32. Quelques petites composantes apparaissent. Mais la technique ne permet pas de décomposer le noyau du graphe « sans échelle ».

Exemple 53. Nous proposons, en Figure 90b, la suppression de 1% des arêtes les plus longues du graphe filtré G'_3 de focus « Jean-Marie » présenté en Figure 74a. Le graphe « arboré » est « proprement » découpé en composantes.

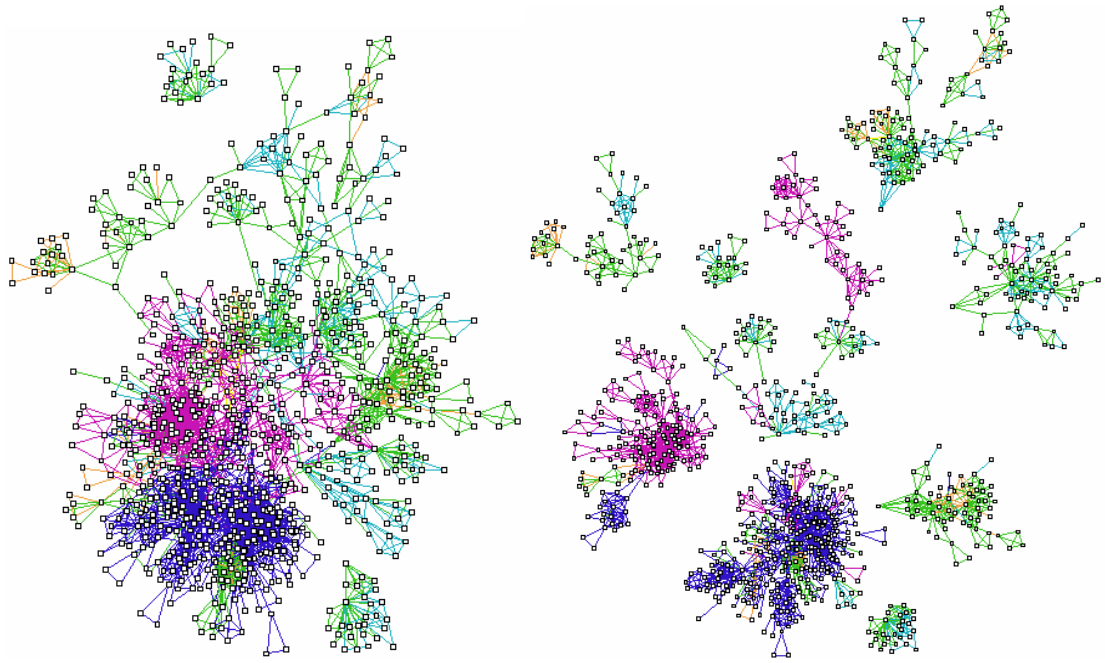


Figure 90 – suppression des arêtes longues (a) co-auteurs initial (b) G'_3 focus : Jean-Marie

3.2.7 Partitionnement géométrique de graphes « arborés »

La structure de graphe « arboré » se prête mal à un partitionnement géométrique à l'aide de plans (sections 3.2.2 et 3.2.3) cercles ou sphères (section 3.2.4).

Si l'on connaît a priori le nombre de clusters à construire, on peut utiliser la méthode des centres mobiles (section 3.2.5). On n'a malheureusement pas toujours cette information.

Parmi les diverses techniques de partitionnement géométrique présentées en section 3.2, la suppression des arêtes « longues » (section 3.2.6) semble la plus adaptée aux graphes « arborés ». Elle fait apparaître clairement les diverses composantes. Comme toutes les techniques géométriques, elle est toutefois entièrement dépendante du placement du graphe.

Nous décrivons, en section 3.3, des techniques de partitionnement basé sur la donnée d'une métrique (pas nécessairement la distance géométrique).

3.3 Partitionnement basé sur une métrique

Certains partitionnements s'appuient sur le calcul d'une métrique. Pour un graphe placé dans un espace euclidien, il est naturel d'utiliser la métrique euclidienne (Buckley et Harary 1990). Une métrique de graphe peut également être utilisée (section 1.1.3). Nous introduisons dans cette section les techniques de clustering les plus utilisées basées sur une métrique.

3.3.1 Partitionnement ascendant hiérarchique

Cette technique de partitionnement issue de l'analyse de données (Alpert et Kahng 1995; Chamberlain 1998; Karypis, Han et al. 1999; Lebart, Morineau et al. 2005) s'adapte naturellement au partitionnement de graphes puisqu'elle nécessite uniquement la définition d'une métrique (Buckley et Harary 1990). Elle peut prendre en compte des graphes valués.

Considérant un graphe d'ordre n, le principe de l'algorithme repose sur le groupement itératif en n-1 itérations des sommets ou ensembles de sommets les plus proches au sens de la distance définie dans le graphe. Le partitionnement obtenu est dit hiérarchique ascendant car il construit par agrégation une structure arborescente appelée dendrogramme.

En considérant l'histogramme des distances d'agrégation on définit un niveau idéal de coupe du dendrogramme correspondant au plus haut palier. Chaque coupe du dendrogramme produit un partitionnement du graphe. En effectuant différentes coupes on obtient une structure de graphe clusterisé hiérarchique.

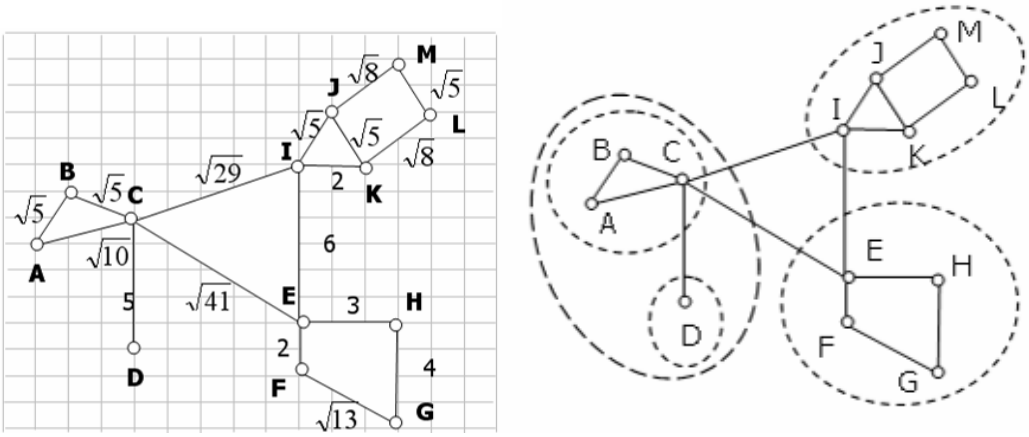


Figure 91 – graphe valué (a) distances euclidiennes (b) vue clusterisée

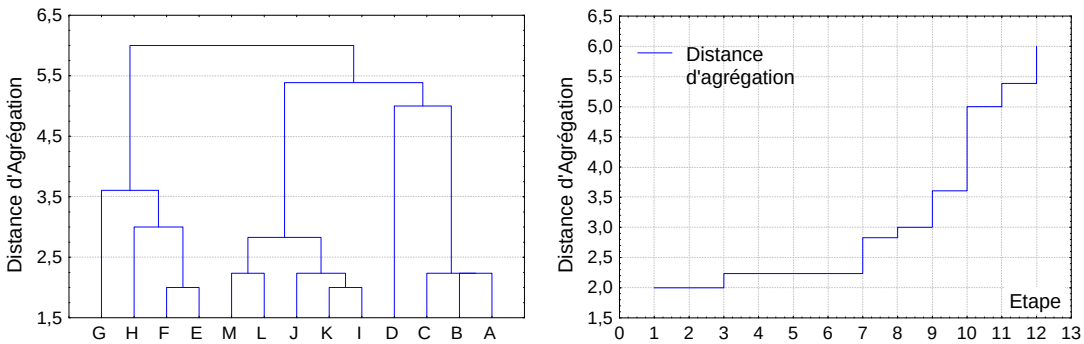


Figure 92 – distance min : dendrogramme et histogramme des distances d'agrégation

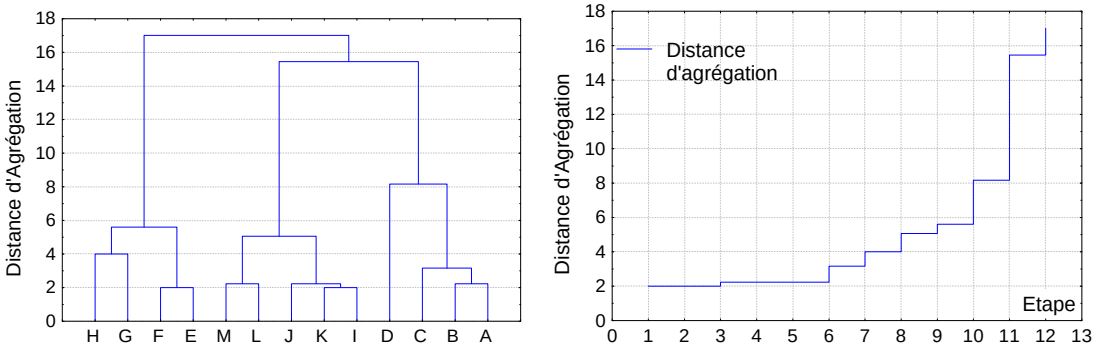


Figure 93 – distance max : dendrogramme et histogramme des distances d'agrégation

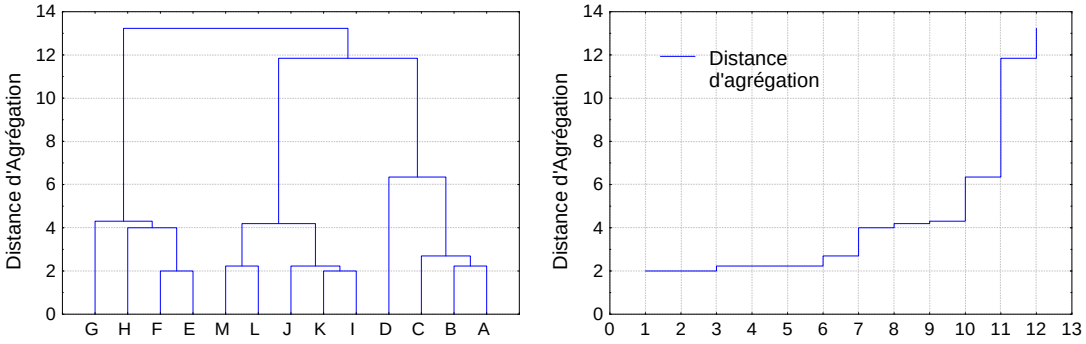


Figure 94 – distance moyenne : dendrogramme, histogramme de distances d'agrégation

Exemple 54. Un partitionnement hiérarchique est appliqué au graphe valué (Figure 91a). Le poids des arêtes représentant la distance géométrique entre sommets, la distance entre A et E est la longueur du plus court chemin : $\sqrt{10}+\sqrt{41}$. Dendrogrammes et histogrammes des distances d'agrégation sont obtenus avec (Statistica) (Figure 92, Figure 93, Figure 94) pour les distances min, max et moyenne. Un palier est obtenu à la $10^{ème}$ étape pour d_{min} (coupe entre 3,6 et 5) et à la $11^{ème}$ étape pour d_{max} et d_{moy}. L'ensemble des sommets est partitionné en {A, B, C}, {D}, {E, F, G, H}, {I, J, K, L, M} pour d_{min} et {A, B, C, D}, {E, F, G, H}, {I, J, K, L, M} pour d_{max} et d_{moy} (Figure 91 b).

Une technique de partitionnement hiérarchique basée sur la distance euclidienne a été récemment développée par (van Ham et van Wijk 2004). Cette technique a été également implémentée par (Koenig et Mélançon 2005) avec InfoVis Toolkit (Fekete 2004) (Figure 95).

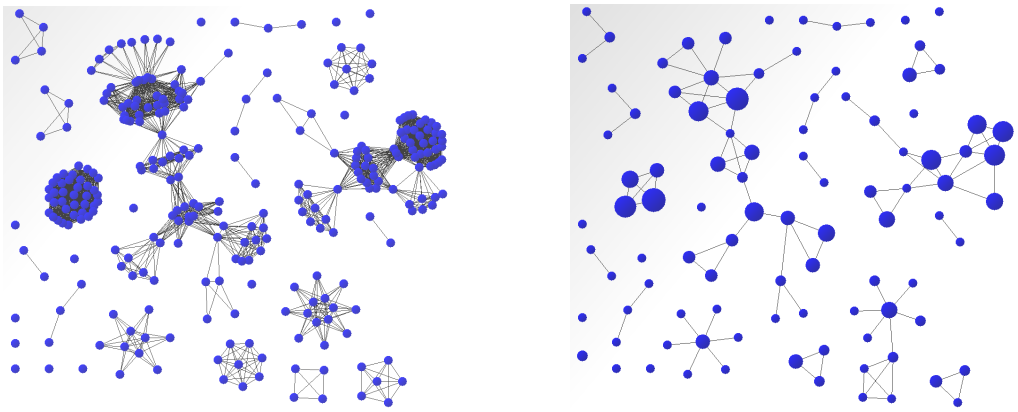


Figure 95 – clustering hiérarchique (a) avant (b) après

3.3.2 Partitionnement mixte

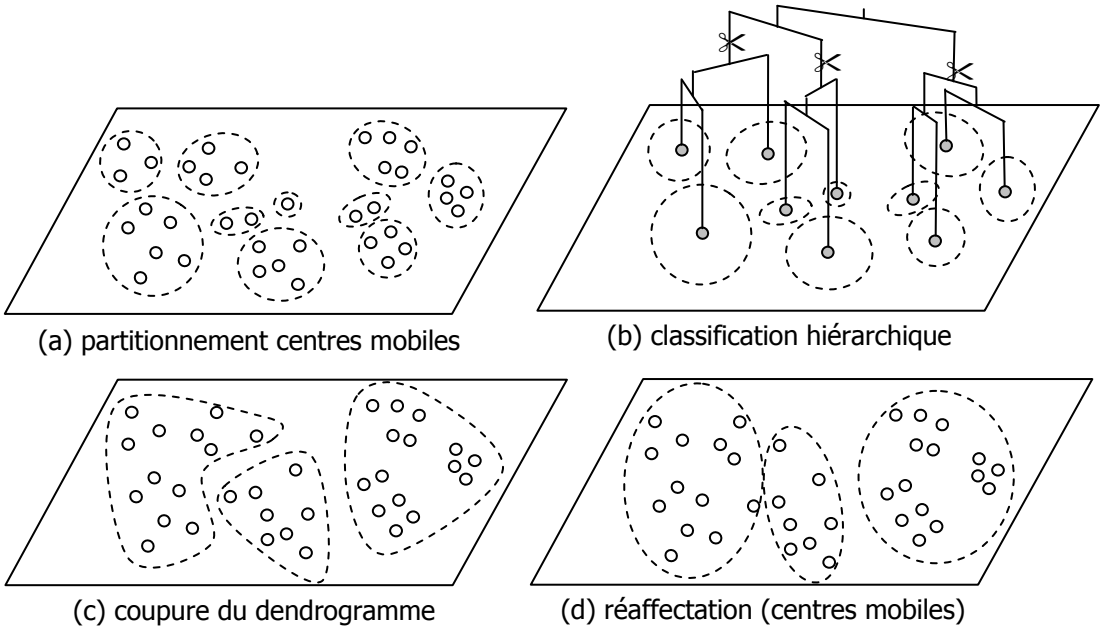


Figure 96 – classification mixte

Il est parfois judicieux de coupler deux méthodes de partitionnement (Karypis, Han et al. 1999; Steinbach, Karypis et al. 2000; Lebart, Morineau et al. 2005). Si le graphe est placé dans un espace euclidien, on peut utiliser une méthode mixte basée à la fois sur la technique de classification hiérarchique et sur celle des centres mobiles.

La technique des centres mobiles partitionne un ensemble important de données à faible coût. Elle présente toutefois deux inconvénients majeurs : elle dépend des centres mobiles choisis initialement et suppose prédéfini le nombre de classes.

La technique de partitionnement hiérarchique est moins adaptée à la classification d'un ensemble important de données mais présente l'avantage d'être déterministe et de ne pas fixer a priori le nombre de classes.

L'algorithme de classification mixte se décompose en trois phases :

- Partitionnement des sommets en un nombre fixé de classes (centres mobiles).
- Classification hiérarchique des classes précédentes. Choix d'un niveau de coupe du dendrogramme. Création de nouvelles classes.
- Réaffectation des individus au sein des classes (centres mobiles).

Exemple 55. Les différentes étapes sont décrites en Figure 96.

3.3.3 Partitionnement descendant par filtrage d'arêtes

La technique de partitionnement descendant proposée par (Auber, Chiricota et al. 2003; Chiricota, Jourdan et al. 2003) est basée sur le calcul d'une métrique sur arête appelée force d'arête (voir section 2.3). L'algorithme supprime successivement les arêtes de force minimale pour constituer des clusters (Figure 97). D'autres métriques peuvent être utilisées comme l'inter centralité (voir section 2.3.1.2). D'après (Auber, Chiricota et al. 2003), cette technique donne de bons résultats pour les graphes « petit monde » à fort coefficient de clustering.

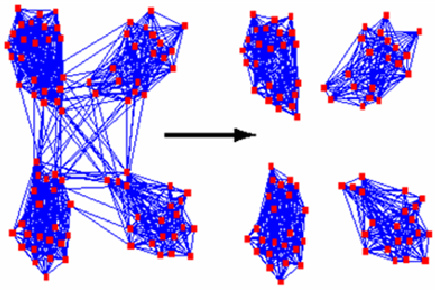


Figure 97 – extraction de clusters par suppression d'arêtes

Remarque 7. Dans l'exemple présenté en Figure 97, la suppression des arêtes « longues » (voir section 3.2.6) aurait donné un résultat très proche.

3.3.4 Partitionnement avec métrique de graphes « arborés »

Une technique de partitionnement hiérarchique ou mixte peut donner des résultats satisfaisants avec un graphe « arboré » (en considérant la distance euclidienne) si on utilise un algorithme de dessin basé sur un modèle de forces. En effet, dans ce cas, les composantes sont naturellement séparées.

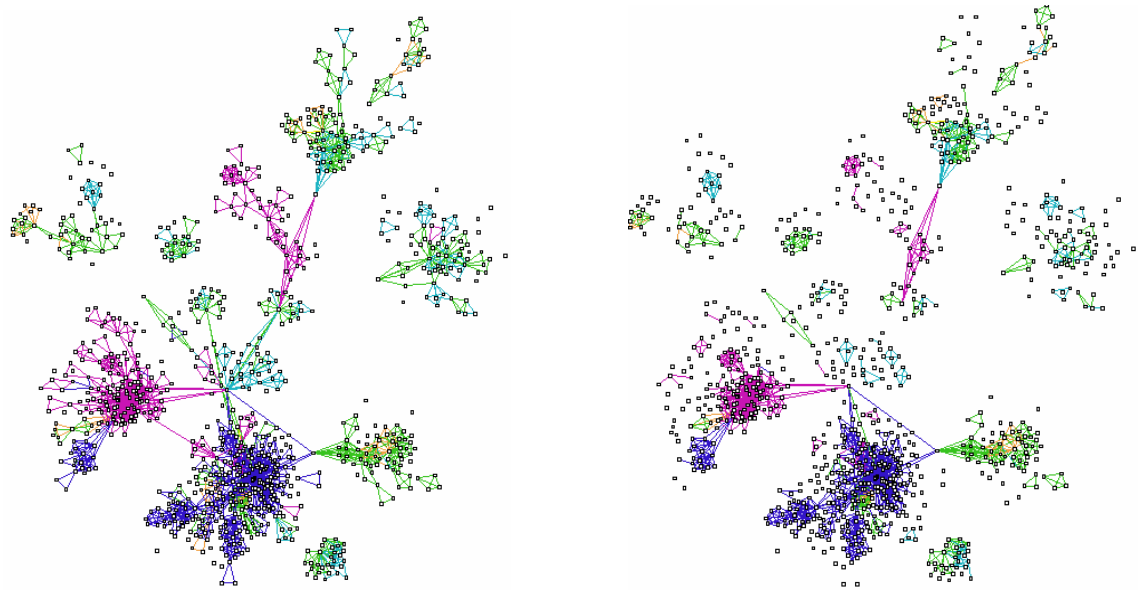


Figure 98 – suppression (a) des arêtes de force nulle (b) 20% des arêtes les plus faibles

L'utilisation d'une métrique autre que la métrique euclidienne ne donne pas toujours des résultats satisfaisants (voir Figure 98) : les trois principales composantes rose, bleue et verte restent connectées après suppression de 20% des arêtes les plus faibles.

3.4 Partitionnement structurel

Nous avons décrit des méthodes de partitionnement utilisant des propriétés géométriques du graphe ou des distances. Les autres méthodes sont dites structurelles.

3.4.1 Partitionnement basé sur un parcours du graphe

L'algorithme (Alpert et Kahng 1995; Chamberlain 1998) détermine deux sommets du graphe à distance « presque » maximale. Partant de l'un des sommets et effectuant un parcours en largeur, la moitié des sommets est affectée à la première classe. Les autres sommets sont assignés à la seconde. L'algorithme est appliqué récursivement à chaque classe.

Une variante de l'algorithme (Karypis et Kumar 1998) propose un parcours en largeur sélectif, ne gardant que les nœuds participant à une diminution du nombre d'arêtes de coupe.

Exemple 56. En Figure 99, les sommets les plus éloignés sont A et B. Les sommets en gris obtenus par un parcours en largeur à partir de A font partie de la classe de A. Les autres appartiennent à la classe de B. L'algorithme n'est pas optimal : en effet le sommet C appartient à la classe de B alors qu'il n'est adjacent à aucun sommet de la classe de B mais à deux sommets de la classe de A.

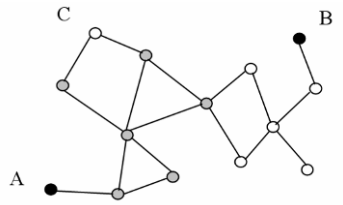


Figure 99 – parcours du graphe à partir du sommet A

3.4.2 Echange de sommets et Min Cut

L'une des techniques de partitionnement de graphe les plus anciennes nommée KL (Kernighan et Lin 1970), construit un partitionnement optimal à partir d'un partitionnement initial en échangeant successivement des sommets entre composantes (Alpert et Kahng 1995; Chamberlain 1998; Brandes, Gaertler et al. 2003). L'objectif est de minimiser une fonction de coût intégrant le nombre de ponts entre composantes.

Définition 70. Le **gain** de l'échange est défini par la différence entre le nombre de ponts entre composantes avant et après échange.

L'algorithme consiste à effectuer une permutation des sommets des deux classes en cherchant à maximiser le gain de l'échange : les sommets associés au gain maximal sont échangés puis bloqués. L'algorithme est appliqué itérativement avec les sommets restants même si le gain est négatif : un tel choix permet de sortir d'un puits de convergence à la recherche d'un optimal global. La solution correspondant au gain maximum est retenue.

L'algorithme peut alors être réitéré en utilisant cette solution comme partition initiale. Il stope lorsqu'aucune solution meilleure que la solution initiale n'est trouvée.

Une variante de l'algorithme, proposée par Fiduccia-Mattheyses (FM) (Chamberlain 1998) choisit itérativement un seul sommet à déplacer associé à un gain maximal. Contrairement à KL, l'algorithme FM modifie ainsi la taille des classes à chaque itération.

3.4.3 Méthode de flux

La technique de partitionnement de Markov (MCL) (van Dongen 2000) repose sur une idée simple : un parcours aléatoire d'une partie dense d'un graphe a peu de risque de quitter cette partie dense avant d'avoir visité bon nombre de ses sommets.

Plutôt que de simuler des marches aléatoires, l'algorithme propose d'étudier le flux du graphe en se basant sur un processus de Markov. La matrice de transition est successivement élevée à la puissance e (simulant e marches) puis normalisée.

L'algorithme proposé converge vers un point fixe ou vers un état récurrent. Les composantes connexes du graphe induit par la matrice finale sont les classes de la partition.

Soit M la matrice d'adjacence du graphe G = (V, E), e le facteur d'expansion et r le facteur d'inflation. Les différentes étapes de l'algorithme sont décrites ci-dessous :

- M est élevée à la puissance e.
- Chaque élément de la matrice (poids) est élevé à la puissance r.
- Chaque poids est ensuite divisé par le poids total de la ligne (normalisation).
- L'algorithme est réitéré tant qu'aucun point fixe ou état récurrent n'est atteint.

3.4.4 Méthode spectrale

Nous étudions une technique (Alpert et Kahng 1995; Elsner 1997; Chamberlain 1998; Brandes, Gaertler et al. 2003) permettant de minimiser le nombre de coupes (ponts) lors d'une bi partition en deux classes A et B d'un ensemble V = (v_1, …, v_n).

Considérons pour cela la matrice Laplacienne L définie par :

$$L_{ij} = \begin{cases} \deg(v_i) \;\; si \;\; i = j \\ -1 \;\; si \;\; v_i \;\; et \;\; v_j \;\; sont \;\; adjacents \\ 0 \;\; ailleurs \end{cases}$$

A chaque bi partition, on associe un vecteur X(x_1,…, x_n) tel que x_i vaut 1 si v_i appartient à la classe A, et vaut -1 sinon.

Théorème 3. le produit $X^t \cdot L \cdot X$ est égal à quatre fois le nombre de coupes (c'est-à-dire quatre fois le nombre d'arêtes entre A et B).

Preuve : Si on écrit L sous la forme D-M avec D matrice diagonale tel que $d_i = \deg(v_i)$:

$$X^t \cdot L \cdot X = X^t \cdot D \cdot X - X^t \cdot M \cdot X = \sum_i d_i x_i^2 - 2 \sum_{(vi,vj) \in E} x_i x_j$$

$$\text{Donc } X^t \cdot L \cdot X = \sum_{(vi,vj) \in E} (x_i - x_j)^2 = \sum_{\substack{vi \in A, vj \in B \\ (vi,vj) \in E}} (x_i - x_j)^2 = 4\delta(A,B)$$

Exemple 57. Considérons trois partitionnements du graphe en Figure 100 a, b et c :

- (a) : aucune coupe, le produit $X^t \cdot L \cdot X$ est nul.
- (b) : deux coupes, le produit $X^t \cdot L \cdot X$ vaut 8.
- (c) : trois coupes, le produit $X^t \cdot L \cdot X$ vaut 12.

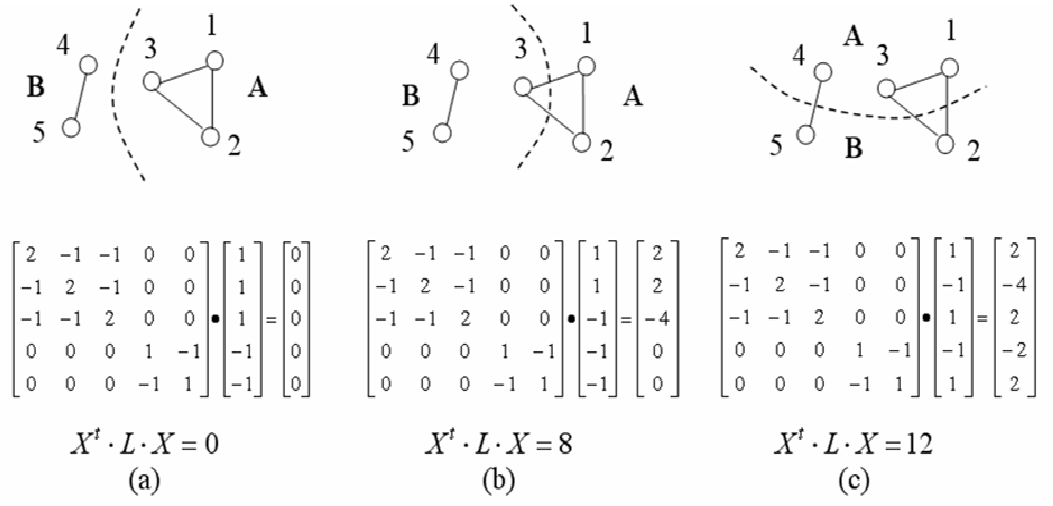


Figure 100 – trois partitionnement (a) 0 coupe (b) 2 coupes (c) 3 coupes

Minimiser le nombre de coupes revient à minimiser l'expression quadratique $X^t \cdot L \cdot X$ où x_i vaut 1 ou -1 et $\sum x_i = 0$. La technique décrite peut être généralisée à un partitionnement multiple (Alpert et Kahng 1995).

Théorème 4. Le problème étant *NP*-complet, une approximation est donnée en considérant que x_i peut varier entre $-\sqrt{n}$ et $\sqrt{n}$: X est alors le vecteur propre de L associé à la plus petite valeur propre strictement positive.

Les sommets v_i ayant une valeur x_i importante sont associés à A, les autres à B :

- La valeur limite peut être fixée à zéro.
- Pour des groupes A et B de même taille, la limite est choisie comme médiane des x_i.

3.4.5 Partitionnement structurel de graphes « arborés »

La méthode de parcours en largeur (section 3.4.1) donne un découpage « basique » du graphe « arboré ». Cependant certaines composantes peuvent être incidemment scindées en deux.

La technique d'échange de sommets (section 3.4.2) suppose un partitionnement initial du graphe. Elle peut alors permettre d'affiner chaque composante.

Les méthodes de flux (section 3.4.3) sont susceptibles de donner de bons résultats avec des graphes « arborés » puisque les composantes sont naturellement organisées. La marche aléatoire a alors de grandes chances de rester « piégée » dans la composante.

La méthode spectrale décrite en section 3.4.4 a une complexité trop élevée pour pouvoir être appliquée à de gros graphes.

3.5 Techniques complémentaires

Les techniques présentées ici peuvent être couplées à un partitionnement structurel.

3.5.1 Méthode de recuit simulé

Cette méthode probabiliste issue de la physique est utilisée pour sortir de puits de convergence (Johnson, Aragon et al. 1989; Alpert et Kahng 1995; Davidson et Harel 1996; Chamberlain 1998). Considérant une partition du graphe retenue à l'étape n, une nouvelle partition est choisie aléatoirement (exemple : par échange de sommets) :

- Si le gain associé à cette transformation est positif, la nouvelle partition est conservée, et le processus réitéré.
- Sinon, on note Δ la perte, la partition est retenue avec une probabilité $e^{\frac{-\Delta}{T}}$ ou T appelée température est une fonction décroissante de n (tendant vers 0).

Propriété 23. Cette méthode a plusieurs avantages :

- Aucune solution n'est rejetée définitivement.
- Les partitions associées à une trop grosse perte ont une probabilité très faible.
- Après un certain nombre d'étapes, la température du « système » est proche de zéro ainsi que la probabilité de retenir une solution de gain négatif.

La méthode de recuit simulé est trop lente pour être appliquée à de gros graphes (Johnson, Aragon et al. 1989).

3.5.2 Recherche tabou

Certaines techniques de partitionnement structurelles peuvent être améliorées en prenant en compte une liste de sommets (ou de couples) tabous (Glover 1989). Il s'agit, par exemple, des k derniers sommets échangés qui ne pourront être déplacés à la prochaine étape.

Une telle méthode permet d'éviter de stagner dans un minimum local en utilisant, par exemple, les mêmes déplacements de façon cyclique.

3.5.3 Algorithme génétique

Le principe des algorithmes génétiques (Alpert et Kahng 1995; Chamberlain 1998) est de construire, à partir d'une population de partitions, de nouvelles générations de partitions obtenues par « croisement » ou « mutation » de partitions de la génération précédente. Les « bonnes solutions » ont statistiquement plus de chance d'être retenues.

La qualité des résultats obtenus dépend fortement des méthodes de croisement et de mutation retenues (pas nécessairement évidentes à choisir).

3.6 Nouvelles techniques de partitionnement dépendant d'un focus

Nous avons développé en section 2.5 une nouvelle technique de filtrage de graphe prenant en compte un focus utilisateur. Le graphe résultant, dit « arboré », présente une arborescence intuitive de composantes.

Les clusters sont susceptibles d'être facilement identifiés par un algorithme adapté de partitionnement : ascendant hiérarchique (section 3.3.1), par suppression d'arêtes longues (section 3.2.6), échange de sommets (section 3.4.2) ou calcul de flux (section 3.4.3). Cependant, ces algorithmes n'assurent pas une organisation structurée des clusters. Par ailleurs, ils ne permettent pas la prise en compte d'un focus.

L'objectif du chapitre est de proposer une technique de clustering basée sur un focus assurant le calcul des composantes du graphe « arboré » et leur structuration multi-échelles. Il en résulte une organisation arborescente contextuelle nommée « contour » du graphe.

Le contour simple d'un graphe peut être représenté par un arbre de clusters (section 3.6.1) ou un arbre de silhouettes (section 3.6.2). Il s'agit d'arbres composés (section 3.1.2). Nous étudierons le lien étroit entre ces deux structures. Nous introduirons ensuite une technique de génération de contours emboîtés : arbre de clusters emboîtés (section 3.6.3) et arbre de silhouettes emboîtées (section 3.6.4). Il s'agit d'arbres composés multi niveaux (section 3.1.2). Ces structures multi-échelles facilitent la représentation et l'exploration de gros graphes à partir d'un focus.

3.6.1 Arbre de clusters

Le contenu de cette section a fait l'objet d'un article (Boutin et Hascoët 2003).

3.6.1.1 Définitions préliminaires et notations

Nous appliquons notre nouvelle technique de partitionnement de contour à un graphe connexe non orienté G = (V, E) où V décrit l'ensemble $\{X_0, \ldots, X_n\}$ et E l'ensemble des arêtes (X_i, X_j). Par convention, nous prendrons X_0 comme focus utilisateur. La distance entre deux nœuds du graphe est définie comme la longueur du plus court chemin entre ces nœuds.

Exemple 58. Nous décrirons l'algorithme sur le graphe présenté en Figure 101.

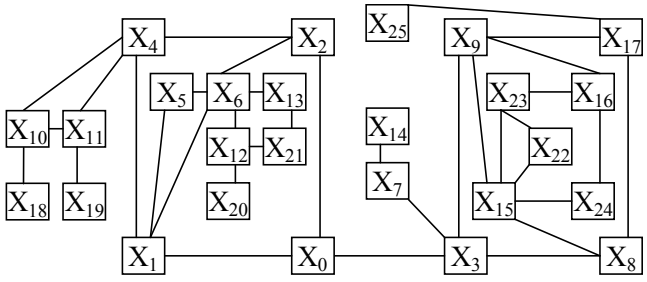


Figure 101 – graphe connexe G

Définition 71. Le niveau d'un nœud X_i est défini par la distance entre X_0 et X_i.

Définition 72. Nous définissons G_n comme le sous graphe de G induit par les nœuds de niveau supérieur ou égal à n (Figure 102).

Définition 73. La composante $C_{n,k}$ est défini comme la $k^{ème}$ composante connexe de G_n.

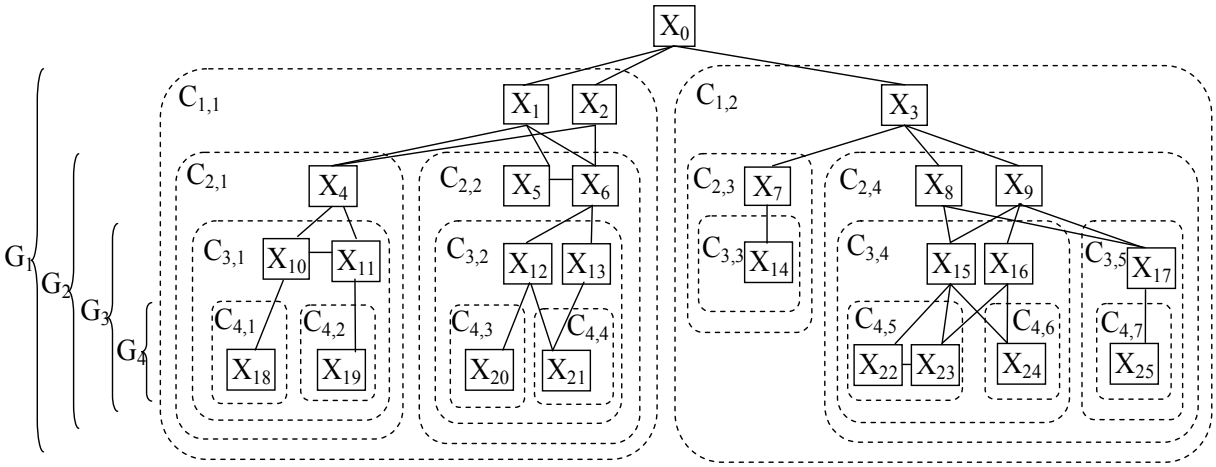


Figure 102 – décomposition des G_k en composantes connexes

Le théorème intuitif suivant est à la base de plusieurs résultats obtenus :

Théorème 5. Considérant les graphes G et G' tels que G' soit un sous graphe de G, toute composante connexe de G' est incluse dans une composante connexe de G.

Grâce au théorème précédent on déduit le résultat suivant :

Propriété 24. Il existe une relation d'inclusion entre $C_{n,k}$ décrite par un arbre d'inclusion : ses nœuds représentent les $C_{n,k}$, ses arêtes orientées représentent les relations d'inclusion entre les $C_{n,k}$ (Figure 103). Les feuilles de cet arbre sont appelées composantes finales (elles ne contiennent aucune composante).

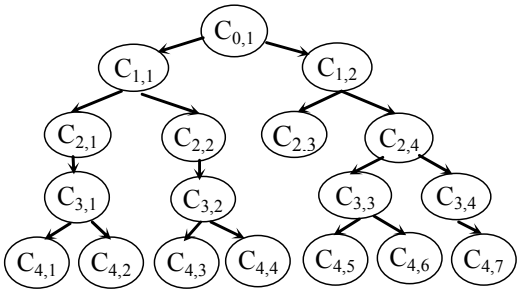


Figure 103 – arbre d'inclusion des composantes connexes

3.6.1.2 Cluster : ensemble d'articulation du graphe

Définition 74. Nous définissons l'ensemble d'articulation $S_{n,k}$ (appelé cluster) par le sous ensemble des nœuds de $C_{n,k}$ de niveau n (voir Figure 104).

Propriété 25. Toute composante non finale $C_{n-1,k}$ se décompose (Figure 104) en :

- Un ensemble d'articulation $S_{n-1,k}$.
- Un ensemble de composantes $\{C_{n,h}\}$.

Soit un ensemble d'articulation $S_{n-1,k}$ lié à différentes composantes $C_{n,h}$. Sa suppression déconnecte ces composantes d'où l'appellation : « ensemble d'articulation ».

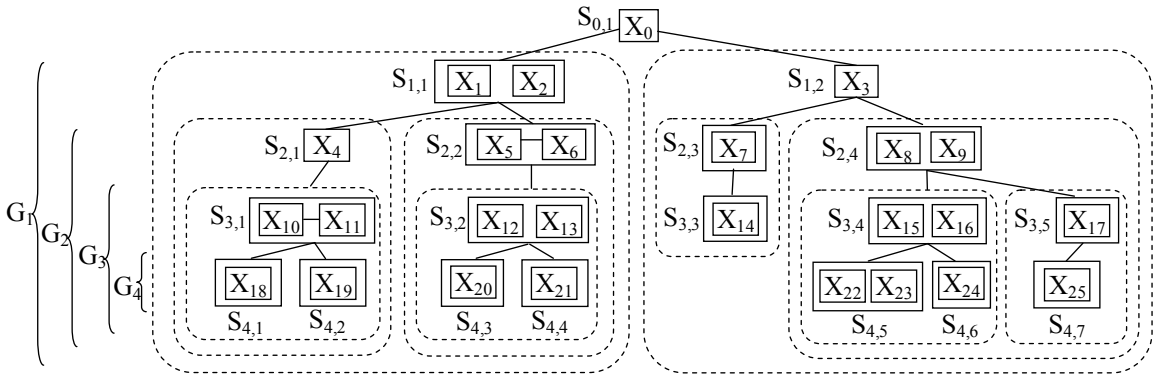


Figure 104 – création des clusters par niveau

Définition 75. Un **lien d'articulation** est défini entre ensembles d'articulation (clusters) s'ils contiennent des nœuds adjacents.

Théorème 6. Le graphe constitué des ensembles d'articulation (clusters) et des liens d'articulation est un **arbre de contour** appelé **arbre de clusters** du graphe noté **T**. Cet arbre est isomorphe à l'arbre d'inclusion (voir Figure 103 et Figure 104). Il a une structure d'arbre d'ensembles (section 3.1.2.1).

3.6.1.3 Optimisation du calcul des clusters

Nous procédons récursivement pour créer les ensembles d'articulation $S_{n,k}$ en introduisant le graphe biparti G'_n composé des nœuds X_i de niveau n et des ensembles d'articulation de niveau n+1 (voir G'_1 : Figure 105).

- Nous déterminons d'abord les composantes connexes de G_{max} (de niveau maximum). Elles définissent les ensembles d'articulation $S_{max,k}$.
- Supposons connus les ensembles d'articulation de niveau n, nous déterminons les composantes connexes $G'_{n-1,k}$ de G'_{n-1} (voir pour n = 2, G'_1 : Figure 105). Puis nous définissons $S_{n-1,k}$ comme l'ensemble des nœuds de $G'_{n-1,k}$ de niveau n-1.

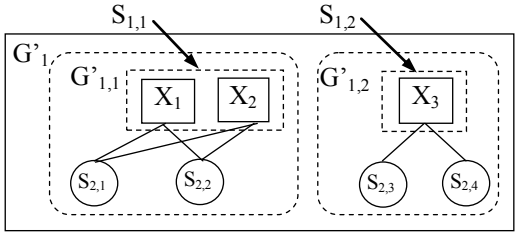


Figure 105 – décomposition de G'$_1$

Cet algorithme calcule ainsi les ensembles d'articulation en temps linéaire : O(|V|+|E|).

3.6.1.4 Description de l'algorithme sur un exemple

Nous proposons l'organisation de 26 pages web collectées sur le site www.nature.com (Figure 106a) à partir d'une page focus de titre « Painkillers show Alzheimer's promise » (Figure 106b). Nous procédons ensuite au partitionnement (Figure 106, Figure 107).

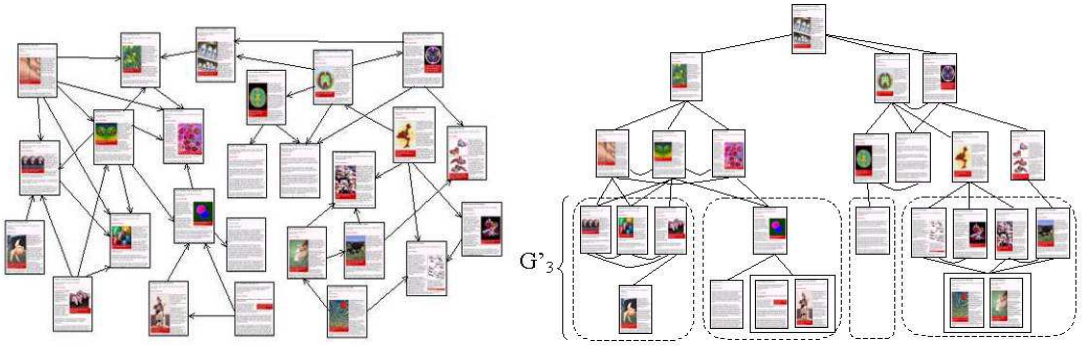


Figure 106 – (a) graphe des pages de Nature (b) clustering – étape 1

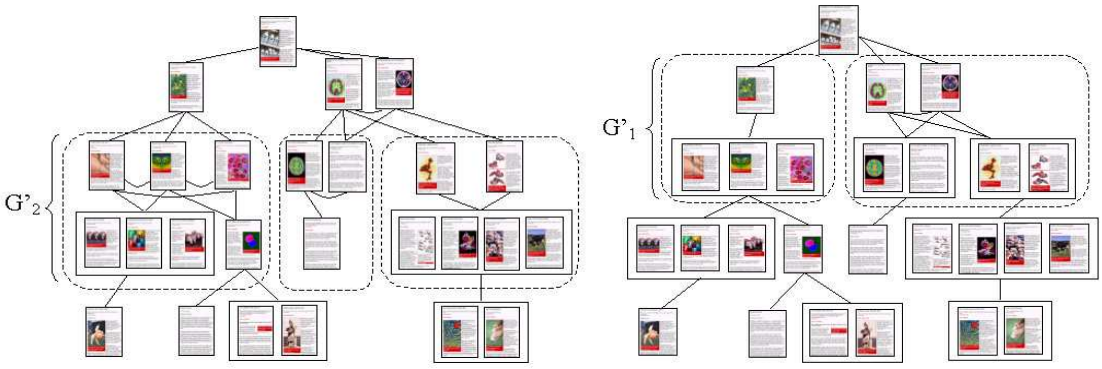


Figure 107 – clustering (a) étape 2 (b) étape 3

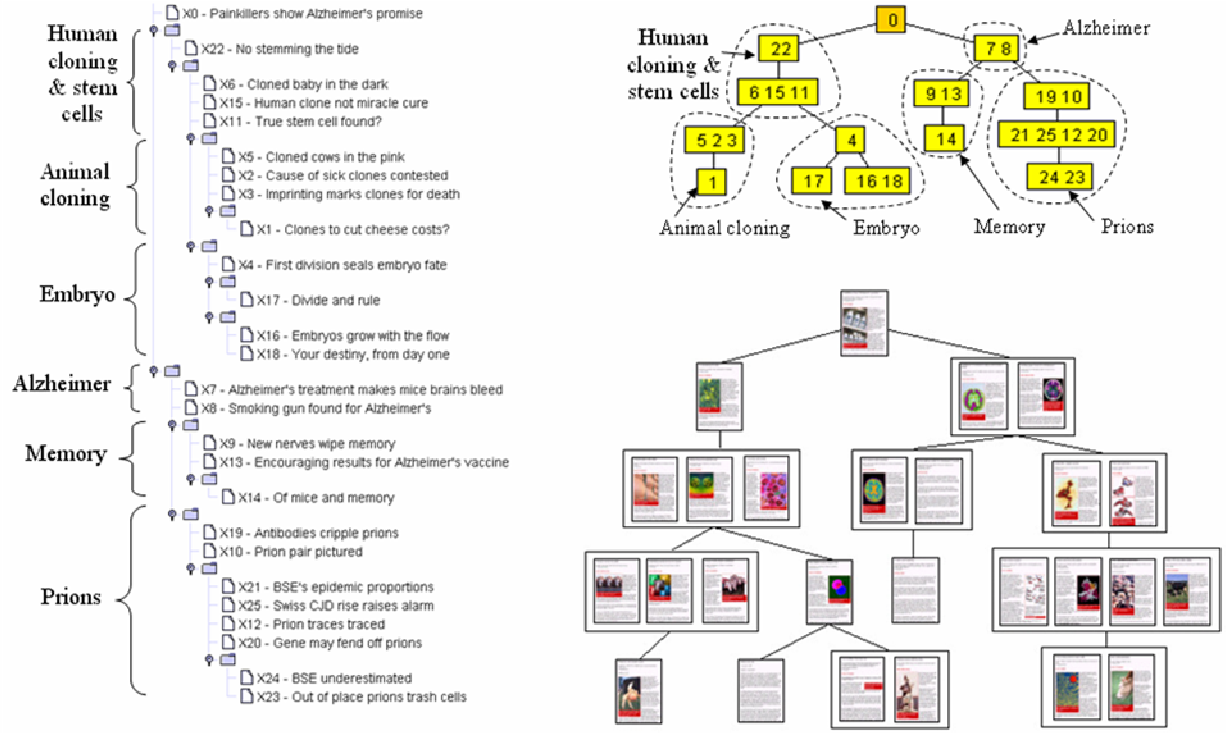


Figure 108 – arbre de clusters – repérage a posteriori de catégories

Nous présentons des vues « pédagogiques » en Figure 106 et Figure 107 obtenues manuellement. Des vues automatiques sont proposées au Chapitre 4.

Une étude a posteriori du partitionnement par analyse du contenu, révèle un groupement des pages selon des thèmes (Figure 108), certains pouvant regrouper plusieurs clusters.

Nous allons introduire en section 3.6.2 la notion de silhouette regroupant plusieurs clusters. Une silhouette est mieux à même de recouvrir un thème.

3.6.2 Arbre de silhouettes

Nous avons mis en évidence (Boutin et Hascoët 2004) des composantes invariantes lors du changement de focus dans l'arbre de clusters (section 3.6.2.1). Ces composantes forment un recouvrement du graphe (et non une partition) rendant ainsi difficile leur dessin.

Nous construisons en section 3.6.2.2 des ensembles de noeuds appelés silhouettes comme sous-ensembles des composantes invariantes. Les silhouettes forment une partition du graphe sans recouvrement (Boutin, Thièvre et al. 2005). Nous proposons en section 4.3.3 des techniques de visualisation d'arbre de silhouettes.

3.6.2.1 Recouvrement par composantes biconnexes et triviales

Rappelons qu'un nœud d'un graphe connexe est noeud d'articulation si sa suppression déconnecte le graphe (Définition 29).

Définition 76. Une composante connexe est dite **biconnexe** (2-connexe) si elle ne peut être déconnectée en supprimant un nœud (voir section 1.1.2).

Définition 77. Une composante est dite **triviale** si elle est formée uniquement de 2 nœuds d'articulation interconnectés (ou d'un nœud d'articulation et d'une feuille).

Propriété 26. Tout graphe connexe G peut être considéré comme un arbre biparti liant nœuds d'articulation et composantes biconnexes maximales ou triviales de G.

Preuve : Si les composantes biconnexes maximales (ou triviales) étaient liées par un cycle, elles formeraient une composante biconnexe (impossible car composante maximale).

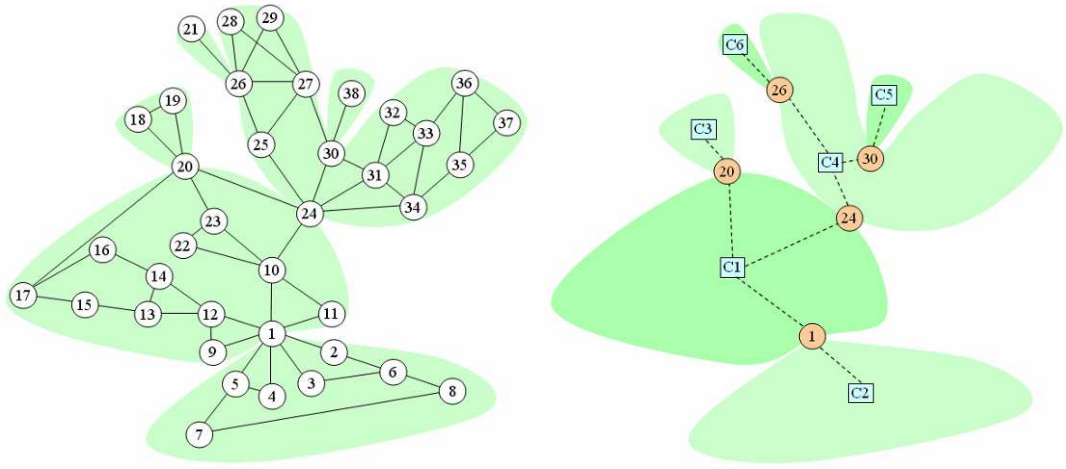


Figure 109 – arbre biparti des nœuds d'articulation et composantes biconnexes (ou triviales)

Exemple 59. La Figure 109, présente l'arbre biparti associé au graphe d'école (Figure 52). Certaines composantes sont triviales comme la composante C5 comprenant deux noeuds. Le nœud d'articulation 30 appartient à la fois à C4 et C5.

Propriété 27. Un nœud d'articulation appartient à plusieurs composantes biconnexes ou triviales. Ainsi l'ensemble des composantes biconnexes et triviales ne constitue pas une partition du graphe mais un recouvrement du graphe.

Nous proposons dans la section suivante une technique permettant de supprimer dans une composante biconnexe (ou triviale) les nœuds d'articulation déjà affectés à une autre composante. Les composantes résultantes appelées silhouettes forment une partition.

3.6.2.2 Partition en silhouettes

Pour choisir à quelle composante affecter un nœud d'articulation, nous proposons un parcours en largeur du graphe à partir d'un focus : chaque nœud d'articulation est affecté à la première composante biconnexe (ou triviale) rencontrée. Les composantes ainsi constituées, sont appelées silhouettes. Elles forment un **arbre de silhouettes** noté **T'** qui a la structure d'arbre d'ensembles définie en 3.1.2.1.

Exemple 60. La Figure 110 présente deux arbres de silhouettes d'un même graphe (Figure 109), basés sur deux focus différents : les nœuds 1 et 24.

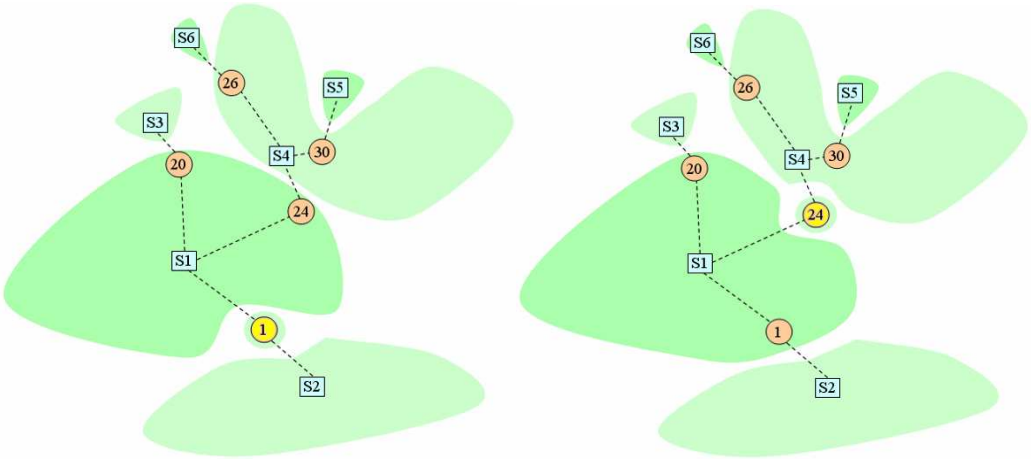


Figure 110 – arbres de silhouettes basés sur les focus (a) 1 (b) 24

3.6.2.3 Relations entre clusters et silhouettes

Nous allons montrer dans cette section qu'une silhouette est constituée de clusters et plus précisément, une silhouette est un arbre de clusters.

Théorème 7. Considérons l'arbre de clusters T et l'arbre de silhouettes T', tous les deux associés au graphe G et à un même focus f. Un cluster C de T (de niveau n) appartient nécessairement à une silhouette.

Preuve : si C est trivial, C ne contient qu'un nœud qui est nœud d'articulation (ou nœud feuille). Ce nœud appartient naturellement à une silhouette. Si C est non trivial on considère deux sommets x et y de C. Ils sont connectés dans G_n car C appartient à une composante

connexe de G_n (Définition 74). Il existe donc un chemin entre x et y passant uniquement par des sommets de niveau supérieur ou égal à n. De plus, par construction, il existe un chemin entre f et x d'une part et f et y d'autre part. Donc x et y sont connectés dans $G \setminus G_n$ (G privé de G_n). C'est-à-dire qu'il existe un chemin entre x et y passant uniquement par des nœuds de niveau inférieur à n. Ainsi, il existe deux chemins indépendants (sans nœuds communs) liant x et y. Par conséquent x et y appartiennent à une composante biconnexe. Le cluster C appartient donc à une seule silhouette.

Théorème 8. Toute silhouette est un arbre d'adjacence de clusters.

Preuve : nous avons montré précédemment que tout cluster est contenu dans une silhouette. Les clusters et les silhouettes formant deux partitions du graphe, on en déduit que toute silhouette est constituée d'une partition de clusters. Une silhouette étant par construction connexe, elle contient un sous ensemble connexe de clusters c'est-à-dire un sous arbre de T.

3.6.3 Arbre de clusters emboîtés

Nous avons introduit au paragraphe 3.6.1 un arbre de clusters T définissant le contour d'un graphe G. Nous proposons, à présent, une organisation multi-échelles de ce contour.

L'idée (Boutin et Hascoët 2004) consiste à « superposer » les arbres de clusters issus des sous graphes de G contenant les nœuds à distance du focus inférieure ou égale à p. Nous obtenons ainsi plusieurs niveaux d'organisation du graphe dépendant du voisinage considéré.

Définition 78. G^p définit le sous graphe de G contenant les nœuds à distance au plus p du focus. T^p définit l'arbre de clusters associé à G^p et au focus.

Théorème 9. Considérons les deux arbres de clusters T^{p-1} et T^p correspondant aux graphes G^{p-1} et G^p. Tout cluster de T^{p-1} est contenu dans un cluster de T^p.

Preuve : Soient deux nœuds appartenant à un cluster de T^{p-1} de la couche $m < p$. Par définition, ils appartiennent à une composante connexe de $G^{p-1}{}_m$, donc à une composante connexe de $G^p{}_m$ (Théorème 5). Ainsi tout cluster de T^{p-1} est contenu dans un cluster de T^p.

Propriété 28. On définit ainsi une relation d'inclusion entre les divers arbres de clusters : $T^0 \subset \ldots \subset T^{p-1} \subset T^p \subset \ldots \subset T$.

Propriété 29. Chaque cluster de l'arbre T est à l'origine d'un arbre d'inclusion de clusters.

Propriété 30. La structure complexe composée des différents arbres d'adjacence de clusters T^p et des arbres d'inclusion est un arbre composé multi niveaux appelé **arbre de clusters emboîtés** (voir section 3.1.2.3).

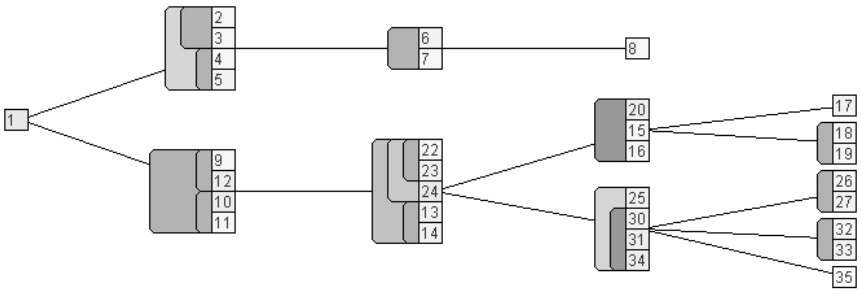


Figure 111 – arbre de clusters emboîtés – focus 1

Exemple 61. Nous proposons (Figure 111), une représentation de l'arbre de clusters emboîtés correspondant au graphe (Figure 109 a) et au focus 1. Il s'agit bien d'un arbre de clusters. Chaque cluster peut englober d'autres clusters. Nous omettons de visualiser les relations intra cluster par soucis de clarté.

Nous reviendrons au Chapitre 4 sur cette vue et proposerons d'autres visualisations multi-échelles de cette structure.

3.6.4 Arbre de silhouettes emboîtées

Nous avons défini au paragraphe 3.6.2 le partitionnement d'un graphe connexe non orienté en un arbre de silhouettes T'. Chaque silhouette contient une arborescence de clusters. Nous allons, comme au paragraphe 3.6.3, proposer une organisation multi-échelles de l'arbre de silhouettes. Pour cela nous appliquons l'algorithme aux sous graphes G^p (Définition 78).

L'idée introduite (Boutin, Thièvre et al. 2005) consiste à « superposer » les arbres de silhouettes T'^p des sous graphes G^p (contenant les nœuds de G à distance inférieure ou égale à p du focus).

Théorème 10. Considérons les arbres de silhouettes T'^{p-1} et T'^p correspondant aux graphes : G^{p-1} et G^p. Toute silhouette de T'^{p-1} est contenue dans une silhouette de T'^p.

Preuve : en effet, toute composante biconnexe de G^{p-1} est incluse dans une composante biconnexe de G^p car G^{p-1} est sous graphe de G^p (d'après Théorème 5).

Propriété 31. On peut ainsi définir une relation d'inclusion entre les arbres de silhouettes : $T'^0 \subset \ldots \subset T'^{p-1} \subset T'^p \subset \ldots \subset T'$.

Propriété 32. Chaque silhouette de T' est à l'origine d'un arbre d'inclusion de silhouettes.

Propriété 33. La structure complexe composée des divers arbres d'adjacence de silhouettes et des arbres d'inclusion est un arbre composé multi niveaux (voir section 3.1.2.3) appelé **arbre de silhouettes emboîtées**.

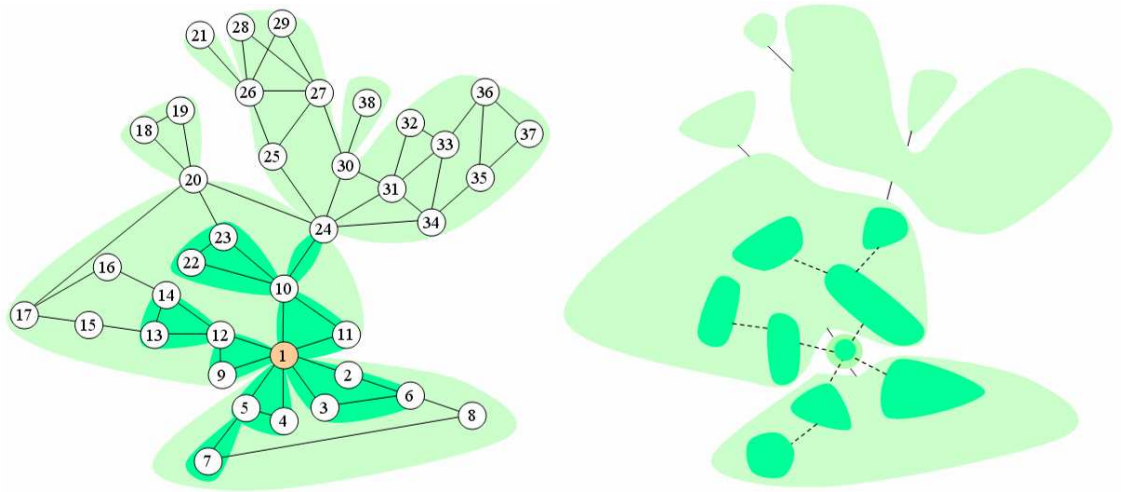


Figure 112 – arbres de composantes biconnexes emboîtées et silhouettes emboîtées

Exemple 62. Nous représentons (Figure 112b) les arbres de silhouettes emboîtées T' et T'^2 obtenus à partir des composantes biconnexes (Figure 112 a) et du focus 1.

3.6.5 Optimisation du calcul de contours emboîtés

Nous allons montrer dans ce paragraphe que les arbres de silhouettes T'^{p} peuvent être calculés itérativement en temps linéaire.

- L'arbre silhouette initial T'^{0} est trivial : il ne contient que le focus.
- Supposons l'arbre de silhouettes T'^{p-1} déjà créé, construisons l'arbre T'^{P}. La $p^{ème}$ couche est initialement organisée en composantes connexes. Si une composante connexe est liée à un unique nœud v sur la couche p-1, v est un nœud d'articulation pour T'^{p} et la composante connexe constitue une nouvelle silhouette de l'arbre T'^{p}. Sinon la composante appartient à une large silhouette englobant la ou les silhouettes de T'^{p-1} auxquelles elle est connectée.

L'algorithme précédant est décrit sur un exemple :

Exemple 63. La Figure 113a, représente l'arbre de silhouettes T'^{p-1} à l'étape p-1. Les composantes connexes de la couche p sont : *{ab}, {cd}, {e}, {f}, {g}, {hij}, {k}, {lm}, {no}*. La Figure 113b présente l'arbre de silhouettes T'^{p-1} à l'étape p. Les composantes *{ab}, {cd}, {e}, {f}, {g}, {lm}* constituent une nouvelle silhouette. Les composantes *{hij}, {k}, {no}* appartiennent à une large silhouette englobant une ou plusieurs silhouettes de T'^{p-1}.

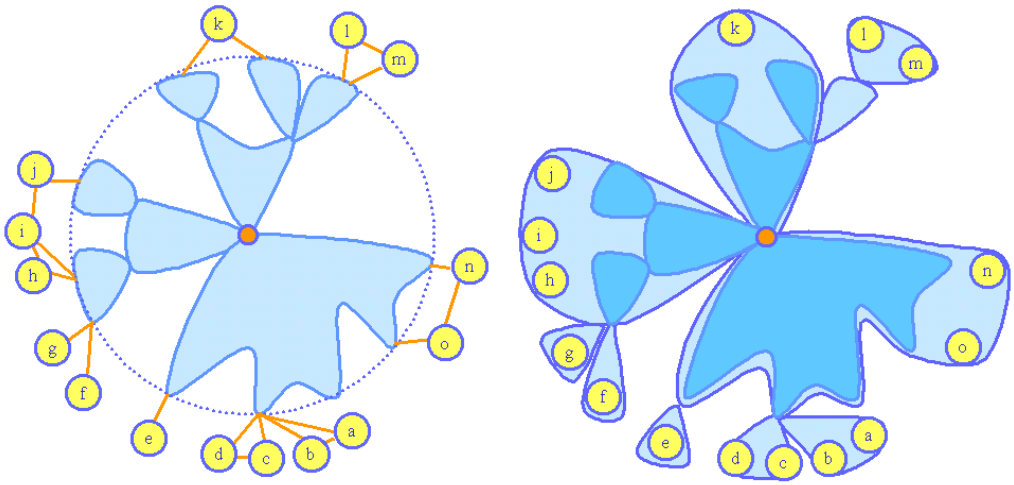


Figure 113 – construction de silhouettes (a) étape p-1 (b) étape p

Propriété 34. L'algorithme est calculé en temps linéaire.

Nous avons décrit dans cette section une technique de partitionnement multi-échelles basée sur un focus. La structure résultante est un arbre de silhouettes emboîtées, chaque silhouette pouvant contenir d'autres silhouettes de niveau inférieur. De plus, une silhouette peut être considéré comme un arbre de clusters (ce qui lui donne son apparence « gantée »).

Nous introduirons en section 4.3 plusieurs techniques de visualisation adaptées. Nous les comparerons sur des exemples en section 5.3.

« J'ai mis toute ma vie à savoir dessiner comme un enfant. »

Pablo Picasso

Chapitre 4 Visualisation de graphes et interaction

Le but premier de la visualisation d'information est de faciliter la compréhension de quantités importantes de données abstraites. Contrairement à la visualisation scientifique, la visualisation d'information s'attache à la représentation de données pour lesquelles il n'y a pas de représentation implicite. Ainsi, dans le domaine de la visualisation de graphes, plusieurs techniques de dessin peuvent être utilisées en fonction de la nature du graphe et du type d'interaction souhaité.

Nous proposons initialement dans ce chapitre un survol des techniques de dessin de graphe et des techniques d'interaction proposées dans la littérature, selon la nature du graphe traité. L'objectif n'est pas de décrire en détail chaque méthode mais d'étudier les caractéristiques intéressantes de chacune.

Nous proposons, en section 4.3, deux techniques de visualisation d'arbres de silhouettes et d'arbres de clusters tirant partie, au mieux, des techniques de visualisation exposées en section 4.1.

Pour un état de l'art plus complet du domaine, on pourra consulter (Battista et Eades 1994; Battista, Eades et al. 1999; Herman, Mélançon et al. 2000).

4.1 Visualisation de graphes : état de l'art

Certains graphes sont caractérisés par des contraintes géométriques : il peut s'agir de graphes de composants d'une carte électronique (VLSI), de graphes d'interactions géographiques (graphe des routes aériennes entre capitales). Les nœuds et arêtes ont alors un placement connu ou un placement à optimiser en fonction de contraintes.

Les autres graphes sont uniquement définis par leur structure (interconnexion des nœuds). Différents placements peuvent alors être proposés suivant les contraintes structurelles fixées et la technique de dessin utilisée.

Nous étudions, tout d'abord, les principales techniques de dessin d'arbres existantes. Certaines utilisent des segments ou des courbes pour représenter les liens entre nœuds (vues horizontale, radiale, Cone Tree et hyperbolique). D'autres représentent les liens de parenté par une relation d'inclusion d'aires (rectangles, cercles ou secteurs angulaires emboîtés).

Nous présentons ensuite diverses techniques de dessin de graphes. Parmi elles, une technique largement utlisée basée sur un modèle de forces. Une autre technique populaire est basée sur le placement préalable d'un arbre couvrant. Le dessin de graphes spécifiques, comme les graphes orientés acycliques (DAG), les graphes planaires ou les graphes composés, doit prendre en compte des contraintes particulières.

4.1.1 Les arbres

4.1.1.1 Vue verticale (ou horizontale)

Une approche « classique » de dessin d'arbre vertical consiste à représenter les arêtes par des segments de droite entre les nœuds. Les nœuds sont placés sur des couches horizontales en fonction de leur distance à la racine. Chaque noeud père est placé au dessus de ses fils. L'algorithme (Reingold et Tilford 1981; Walker 1990) optimise le placement des nœuds de façon à minimiser la largeur de l'arbre (Figure 114). Cette représentation verticale d'un arbre est largement utilisée pour de petits arbres. Elle présente un inconvénient majeur pour les gros arbres : les nœuds des couches supérieures (proche de la racine) ont autant de place que ceux des couches inférieures alors qu'ils sont souvent beaucoup moins nombreux.

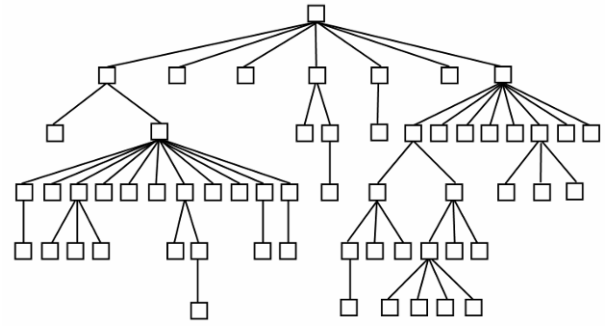


Figure 114 – algorithme de Reingold et Tilford

4.1.1.2 Vue radiale

L'utilisation d'une vue radiale permet d'optimiser l'occupation de l'espace (Eades 1992). Les nœuds à même distance de la racine sont disposés sur un cercle autour de la racine. En considérant des cercles concentriques de rayon R, 2R, 3R… on peut placer un arbre binaire régulier en conservant le même espacement entre nœuds. Pour un arbre dont les nœuds possèdent de nombreux fils, on ne peut garder cette propriété d'équidistance : les nœuds sur les couches périphériques ont alors moins de place que les nœuds proches du focus. Une vue radiale leur accorde toutefois davantage de place qu'une vue verticale (section 4.1.1.1).

Pour éviter le recouvrement de branches, le principe de base consiste à accorder un secteur angulaire optimal à chaque branche (Figure 115).

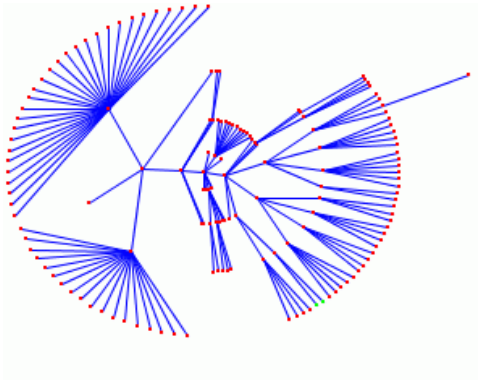


Figure 115 – vue radiale

4.1.1.3 Vue 3D

Des astuces graphiques peuvent être utilisées pour visualiser des arbres dont le degré moyen des nœuds est important. Cone Tree (Robertson, Mackinlay et al. 1993) propose par exemple une vue 3D où chaque branche correspond à un cône pouvant pivoter sur son axe (Figure 116). Cette représentation permet de représenter de façon synthétique des arbres assez gros en cachant à l'utilisateur la moitié des nœuds. Toutefois au-delà de 1000 nœuds, l'arbre est difficilement utilisable.

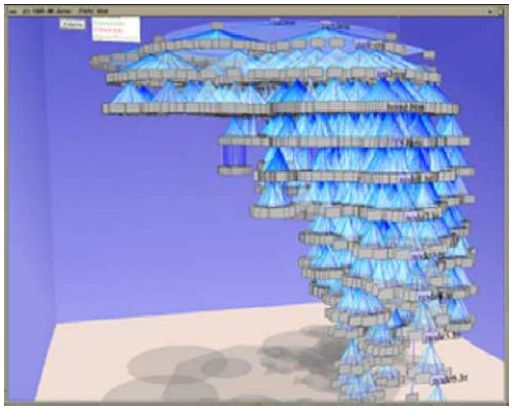


Figure 116 – Cone Tree (Robertson 1991)

4.1.1.4 Vue hyperbolique

Nous présentons une technique de dessin d'arbres basée sur la géométrie hyperbolique (Lamping, Rao et al. 1995; Munzner 1998). L'arbre est initialement placé dans le plan hyperbolique, puis projeté sur un disque visualisé à l'écran (Figure 117). Lors du changement de focus, le placement de l'arbre est conservé, seul le plan hyperbolique est transformé.

Cette technique produit naturellement une vue « focus+contexte » (voir section 4.2.1.3). Le calcul d'une nouvelle vue est ainsi rapide et fluide, favorisant l'exploration de gros arbres. Cependant l'utilisation du navigateur hyperbolique peut demander un certain travail intellectuel. En effet, une transformation simple telle que la translation peut entraîner la rotation de certaines branches (dans l'espace euclidien).

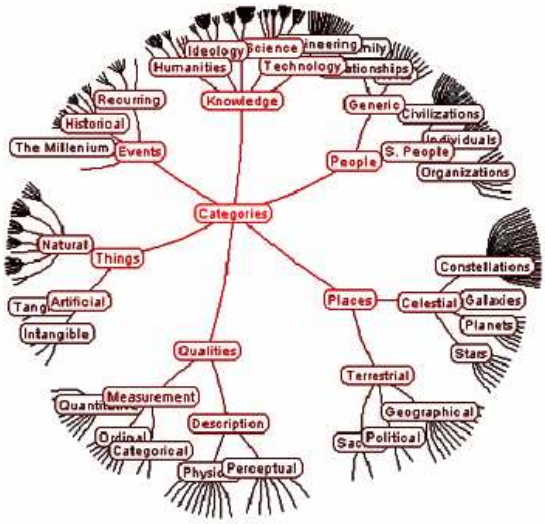


Figure 117 – arbre hyperbolique

4.1.1.5 Intérêt des surfaces emboîtées

Nous présentons, dans la suite de cette section, différentes techniques permettant de visualiser un arbre comme un ensemble de surfaces emboîtées. Les avantages sont multiples :

- Occupation optimisée de l'espace
- Calcul rapide de la vue
- Aucun problème de recouvrement
- Aucune arête mais des inclusions de surfaces
- L'importance d'un nœud est représentée par sa taille
- La coloration des surfaces permet de représenter une autre propriété

Différentes techniques d'interaction seront également développées en section 4.2.

4.1.1.6 Rectangles emboîtés

Les TreeMaps (Johnson et Shneiderman 1991) (Figure 118) constituent une technique de dessin d'arbre telle que chaque noeud est représenté par un rectangle englobant d'autres rectangles (ses fils). Une telle vue permet de privilégier certaines branches en jouant sur la taille des rectangles associés.

Par construction, une vue TreeMaps permet une optimisation de l'espace d'affichage en en utilisant la totalité du rectangle attribué. Elle diffère ainsi des vues classiques qui cherchent (de manière parfois coûteuse) à optimiser la place utilisée et le non recoupement d'arêtes.

Tous les nœuds de l'arbre sont représentés par des rectangles emboîtés. Seules les feuilles sont visualisées par des aires. Aussi, il peut être parfois difficile de disposer de place suffisante pour nommer un nœud intermédiaire sans réduire la branche dont il est le père.

Nous verrons, (section 4.2) que l'utilisation de rectangles emboîtés facilite l'usage et la compréhension de techniques « focus + contexte », faisant ainsi la force des TreeMaps.

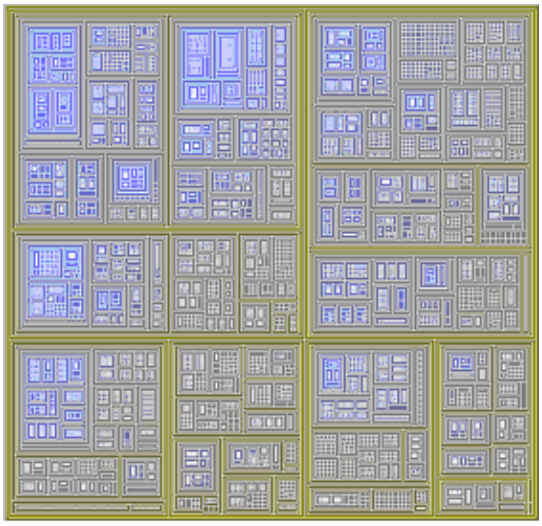


Figure 118 – TreeMaps

4.1.1.7 Cercles emboîtés

D'autres techniques de dessin d'arbres (Grokker; Grivet, Auber et al. 2004) remplissent l'espace par des disques concentriques. A un niveau de l'arbre, l'aire d'un disque est proportionnelle à l'importance du nœud correspondant. Les disques fils sont répartis, de façon optimale, sur le pourtour intérieur du disque père. La couleur peut être utilisée pour représenter toute autre propriété (Figure 119).

Ces techniques sont adaptées à la visualisation d'arbres dont les nœuds ont un degré ni trop faible ni trop grand (entre 3 et 15). En dehors de ces limites, l'occupation de l'espace n'est pas optimisée et il est préférable d'utiliser une autre vue.

La vue par cercles concentriques libère une zone centrale qui peut être utilisée pour ajouter des informations propres au nœud (ce qui est plus difficile avec les TreeMaps).

Nous verrons en section 4.2 que l'utilisation adéquate de techniques de zoom facilite l'exploration multi-échelles de l'arbre.

Figure 119 – vues obtenues avec Grokker

4.1.1.8 Secteurs angulaires emboîtés

L'utilisation de rectangles (sections 4.1.1.6) ou de cercles emboîtés (section 4.1.1.7) a plusieurs avantages (section 4.1.1.5) mais présente l'inconvénient de désorienter un utilisateur habitué aux représentations classiques d'arbres avec nœuds et arêtes.

Les vues proposées par (Stasko et Zhang 2000; Yang, Ward et al. 2003) présentent l'avantage de combiner à la fois une représentation radiale de l'arbre et une affectation de zones (secteurs d'anneaux) aux nœuds.

L'espace est découpé en secteurs angulaires autour du focus mais également en anneaux concentriques (comme une cible). L'intersection d'un secteur angulaire et d'un anneau définit un secteur d'anneau. A chaque branche correspond un secteur angulaire et à sa racine correspond un secteur d'anneau.

Exemple 64. Sur la vue (Figure 120), seuls les secteurs d'anneaux sont représentés.

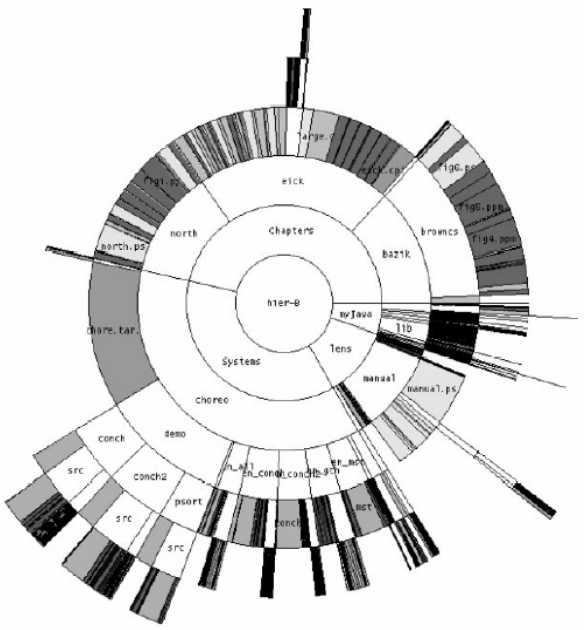


Figure 120 – arbre en secteurs angulaires et secteurs d'anneaux

4.1.2 Les graphes

Un arbre est un graphe particulier facilement visualisable. Le dessin d'un graphe quelconque est plus délicat. Différentes heuristiques de dessin ont été proposées.

Nous décrivons dans cette section deux approches classiques de dessin de graphes. La première repose sur l'extraction et le placement préalable d'un arbre couvrant pour chaque composante connexe. La seconde consiste à utiliser une technique d'optimisation basée sur un modèle de forces. Nous présenterons, enfin, d'autres techniques pour des graphes spécifiques.

4.1.2.1 Placement radial d'un graphe

Certains algorithmes de dessin de graphe reposent sur l'extraction et le placement d'un arbre couvrant pour chaque composante connexe. Différentes techniques d'extraction peuvent être utilisées (Jungnickel 1999). Le placement de l'arbre couvrant a généralement un temps de calcul linéaire.

Les arêtes n'appartenant pas à l'arbre couvrant sont ensuite ajoutées. La difficulté est de choisir un bon ordonnancement des noeuds sur chaque couche afin de limiter les croisements d'arêtes. Le problème étant *NP* complet, il faut utiliser des méthodes d'optimisation (Marti et Laguna 2003).

Par construction cette technique prend en compte un focus par composante : racine de l'arbre couvrant. Elle donne de bons résultats pour des graphes « arborés » (section 1.2.5) minimisant le temps de calcul et les croisements d'arêtes. Par contre, elle n'est pas toujours adaptée à la visualisation de gros graphes denses.

Par ailleurs cette technique peut nécessiter le choix de plusieurs focus dans le cas d'un graphe non connexe.

Exemple 65. La Figure 121a donne une vue radiale du graphe d'école (Figure 52) de focus 1 placé avec l'algorithme (Yee, Fisher et al. 2001) (section 4.1.1.2).

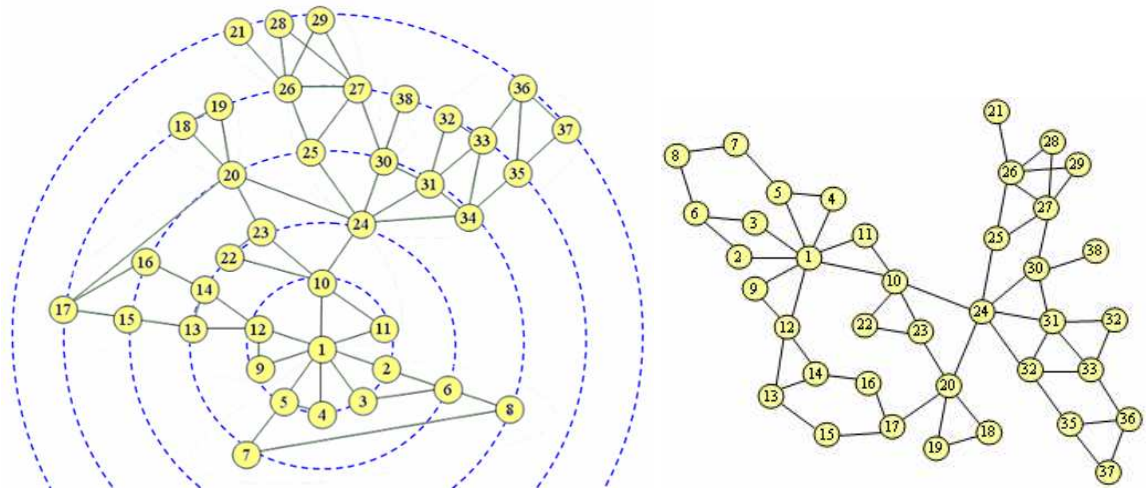


Figure 121 – graphe d'école (a) vue radiale (b) modèle de forces

Lors du changement de focus, la trajectoire des nœuds est optimisée (Yee, Fisher et al. 2001) de façon à limiter les croisements de nœuds et et faciliter ainsi la compréhension de la transformation (Figure 122).

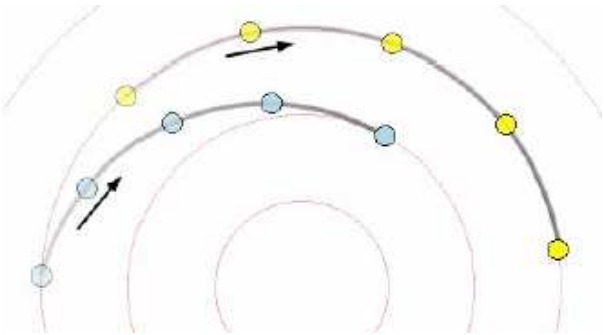


Figure 122 – vue radiale - déplacement optimisé des nœuds

Le placement radial est intéressant lorsqu'on veut prendre en compte un focus. Par contre, les nœuds ont la contrainte forte d'être sur un cercle concentrique. Cette contrainte peut entraîner des intersections d'arêtes que l'on n'obtient pas toujours avec un algorithme basé sur un modèle physique (voir section 4.1.2.2).

4.1.2.2 Algorithmes basés sur un modèle de forces

Une méthode de dessin basée sur un modèle physique est proposée par (Eades 1984) et affinée par (Fruchterman et Reingold 1991). Elle présente des résultats satisfaisants pour des graphes « peu » denses comme les graphes « arborés » introduits en section 1.2.5. Le principe de l'algorithme est exposé en section 3.2.1.3.

Exemple 66. La Figure 121b est une vue du graphe école obtenue avec modèle de forces. Il s'agit d'un petit graphe planaire. Sa représentation est voisine de celle du graphe radial (Figure 121a) mais l'occupation de l'espace est meilleure.

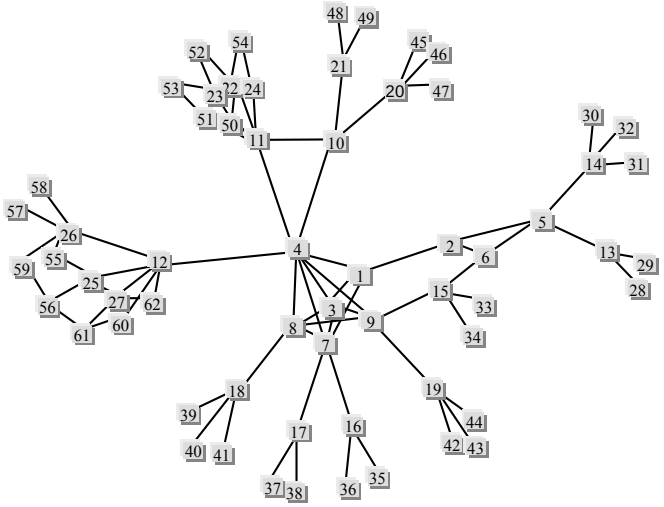


Figure 123 – dessin de graphe obtenu par modèle de forces – graphe de CiteSeer

Exemple 67. La Figure 123 présente une vue avec modèle de forces du petit graphe de CiteSeer introduit en section 1.3.1.2.

Diverses fonctions d'énergie peuvent être utilisées : Noack en a proposé une, optimisée pour le placement de graphes « petit monde » (Noack 2003).

Exemple 68. La Figure 124 présente un pseudo graphe aléatoire G (4, 100, 25 %, 10 %) comprenant 4 clusters de 100 nœuds chacun, tel que deux sommets u et v aléatoires ont 25 % de chance d'être liés s'ils appartiennent à un même cluster, 10 % sinon. Le modèle proposé par Noack donne de meilleurs résultats que celui de Fruchterman et Reingold (voir Figure 124a et b).

Figure 124 – G (4, 100, 0.25, 0.1) modèles de (a) Noack (b) Fruchterman et Reingold

Ces techniques de dessin de graphe basées sur un modèle physique peuvent toujours être utilisées lorsqu'aucun algorithme spécifique ne peut être appliqué. Basées sur la minimisation d'une fonction d'énergie, ces méthodes ont une complexité en $O(n^3)$.

Pour de gros graphes, il est possible d'utiliser une technique multi-échelles consistant à placer initialement les clusters, puis les nœuds à l'intérieur des clusters. La complexité tombe ainsi à $O(n^2 \log n)$ (Walshaw 2000; Koren, Carmel et al. 2002). Une autre technique de dessin multi-échelles à base de forces est introduite dans (Gajer, Goodrich et al. 2000).

4.1.2.3 Technique de recuit simulé : optimisation avec contraintes

Davidson et Harel ont proposé un algorithme de recuit simulé (Davidson et Harel 1996) optimisant des critères de qualité du graphe. Cette méthode, très gourmande en temps de calcul, donne des résultats intéressants pour de petits graphes.

Exemple 69. Le dessin de graphe présenté en Figure 125 optimise le placement des nœuds en prenant en compte deux contraintes : une distribution uniforme des nœuds dans le plan et l'égalité des longueurs d'arêtes.

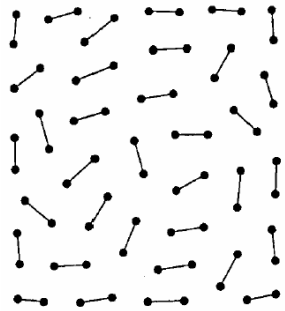


Figure 125 – distribution uniforme des nœuds et même longueur d'arête

Exemple 70. En Figure 126, deux vues d'un même graphe sont présentées. La seconde plus « esthétique » et régulière minimise la distance entre nœuds et arêtes.

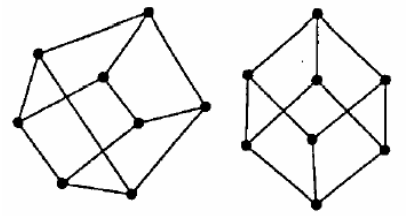


Figure 126 – minimisation de la distance entre nœuds et arêtes

4.1.2.4 Dessin de graphes acycliques orientés (DAG)

Les graphes acycliques orientés (DAG) sont une généralisation des arbres : leurs nœuds, assignés à des couches parallèles, sont reliés par des arêtes de même sens, de sorte qu'il n'y ait pas de cycle.

Deux principales techniques proposent un placement de DAG en minimisant les intersections d'arêtes. La première repose sur le placement d'un arbre couvrant et la réorganisation successive des nœuds sur les couches tant que la configuration n'est pas satisfaisante (Sugiyama, Tagawa et al. 1981; Gansner, Koutsofios et al. 1993) (Figure 127). La seconde est basée sur l'heuristique du « barycentre » (Battista, Eades et al. 1999; Herman et Mélançon 2000) consistant à affecter à chaque nœud la moyenne des coordonnées de ses voisins (calcul en temps linéaire).

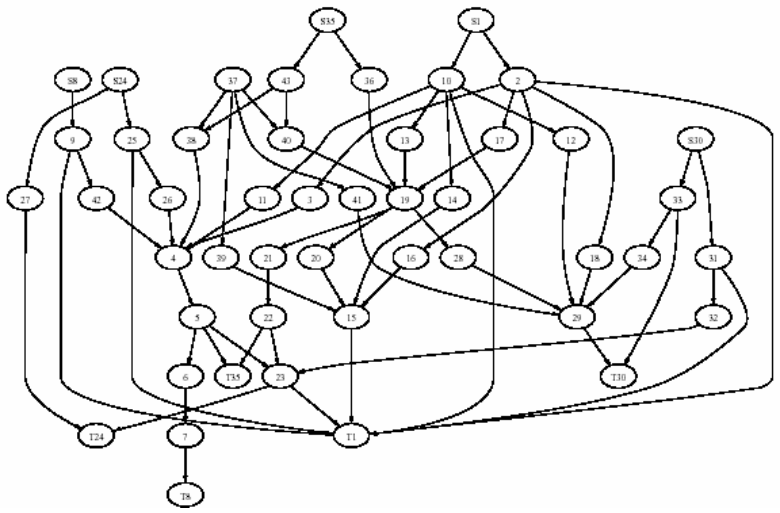


Figure 127 – dessin de DAG

Les graphes bipartis sont des DAG particuliers à deux couches. Différents travaux ont étudiés la minimisation des intersections d'arêtes dans un graphe biparti (Jünger et Mutzel 1997). Une comparaison des techniques est proposée dans (Marti et Laguna 2003).

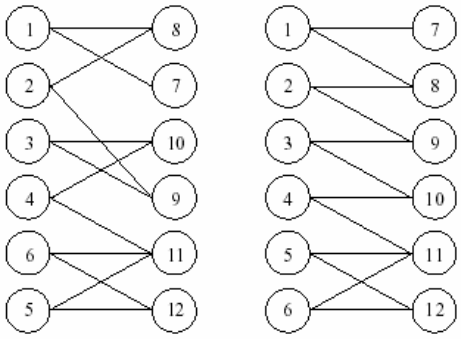


Figure 128 – ordonnancement des nœuds minimisant les intersections d'arêtes

4.1.2.5 Dessin de graphes planaires

Plusieurs techniques assurent le dessin de graphes planaires (Kant 1996; Hong et Eades 2005) (Figure 129). La majorité des graphes d'interactions réels n'étant pas planaires, nous n'étudierons pas ces techniques.

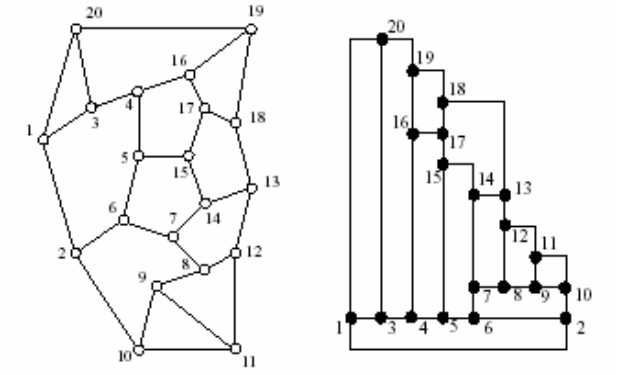


Figure 129 – dessin orthogonal de graphe planaire 3-connexe

4.1.3 Les structures de graphes multi-échelles

4.1.3.1 Vues générales

Exemple 71. Des vues 3D de graphes composés (section 3.1.1.3) sont proposées (Feng 1997) évitant le recoupement d'arêtes d'un même niveau (Figure 130a).

Figure 130 – représentation 3D de graphe composé (a) global (b) par niveau

4.1.3.2 Vues par niveau de partitionnement

Certaines vues superposées de graphes composés (Quigley 2000) ne présentent pas de liens entre niveaux (Figure 130b).

4.1.4 Graphes composés orientés

Dans un graphe composé orienté, les nœuds sont placés sur des couches et groupés en clusters (Figure 131a). La difficulté principale du dessin est de minimiser les intersections d'arêtes. Diverses solutions sont proposées. Certaines sont issues des techniques de placement de DAG (Sugiyama, Tagawa et al. 1981) (section 4.1.2.4). D'autres, plus spécifiques aux graphes composés (Sugiyama et Misue 1991; Sander 1996), proposent une organisation des nœuds sur chaque couche évitant le « découpage » de clusters.

Forster s'intéresse au dessin de graphes composés bipartis (Forster 2002) (Figure 131b) minimisant les croisements d'arêtes sous la contrainte de garder des clusters rectangulaires sans recouvrement.

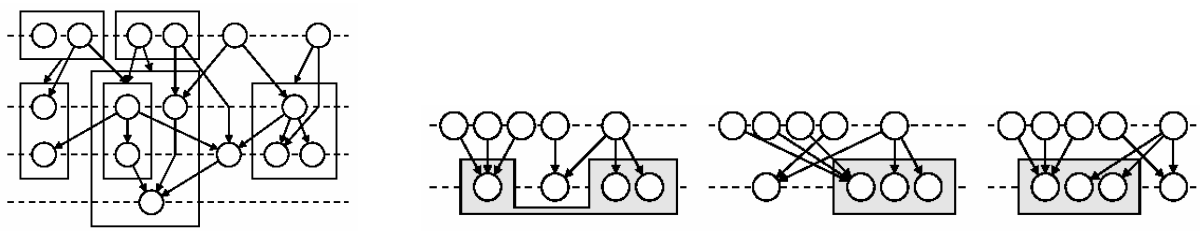


Figure 131 – (a) graphe composé orienté (b) réorganisation des nœuds par couche

4.2 Focus + contexte : état de l'art

La majorité des techniques de visualisation, décrites précédemment, assurent un affichage rapide d'arbres (ou graphes) de quelques milliers de nœuds. Toutefois, en raison de la taille réduite des écrans, l'exploitation des vues est limitée sans l'utilisation de techniques d'exploration adaptées.

Nous étudions dans cette section des techniques de visualisation « focus + contexte ». Elles proposent à l'utilisateur une vue détaillée autour du focus sans perte du contexte. Deux grandes familles de techniques sont utilisées :

La première famille regroupe les techniques basées sur une distorsion géométrique (zoom géométrique) donnant davantage de place aux nœuds proches du focus. Chaque nœud est visualisé. La taille d'un nœud dépend de sa distance au focus.

La seconde famille correspond aux techniques de zoom sémantique. Le niveau de détail de l'élément à afficher (cluster, branche d'arbre) dépend de sa distance au focus. Suivant le focus choisi certains éléments peuvent être détaillés, simplifiés ou simplement supprimés.

Des techniques mixtes alliant zoom géométrique et sémantique peuvent être utilisées.

4.2.1 Zoom géométrique

L'utilisation d'un zoom géométrique (Sarkar et Brown 1992) permet d'avoir une vue locale tout en préservant le contexte général. Nous allons décrire différentes techniques.

4.2.1.1 Distorsion 2D

La technique du fisheye géométrique (Sarkar et Brown 1992) (Figure 132) permet de déformer une vue 2D indépendamment de la technique de dessin utilisée. Elle s'applique ainsi à une image 2D mais également à une vue de graphe, arbre ou toute autre structure représentée dans le plan. L'utilisateur peut changer de focus ainsi que de facteur de distorsion pour explorer la vue.

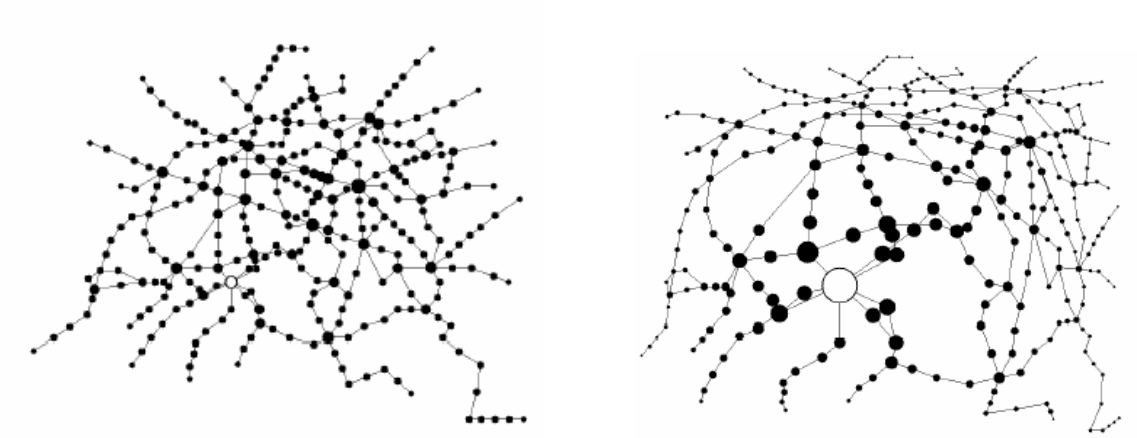


Figure 132 – application d'un zoom géométrique

4.2.1.2 Distorsion 1D

L'exploration de gros arbres peut être simplifiée par une déformation géométrique : TreeJuxtaposer (Munzner, Guimbretiere et al. 2003) est un outil de comparaison de gros arbres phylogénétiques pouvant contenir un million de nœuds. L'utilisation de techniques « focus + contexte » facilite la comparaison des différents arbres. L'utilisation de couleurs pour les branches sélectionnées assure une bonne lisibilité des vues (Figure 133).

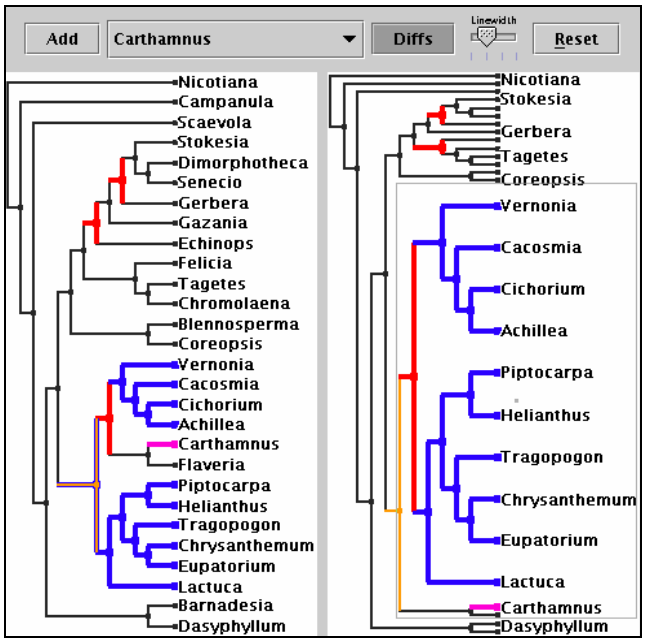


Figure 133 – comparaison de deux arbres phylogénétiques

4.2.1.3 Distorsion hyperbolique

La géométrie hyperbolique (Lamping, Rao et al. 1995; Munzner 1998) permet de zoomer « naturellement » sur certaines zones tout en conservant le contexte général (section 4.1.1.4). En effet la distorsion est une caractéristique de cette géométrie non euclidienne.

La Figure 134 présente une vue hyperbolique favorisant un focus tout en conservant le contexte (Lamping, Rao et al. 1995).

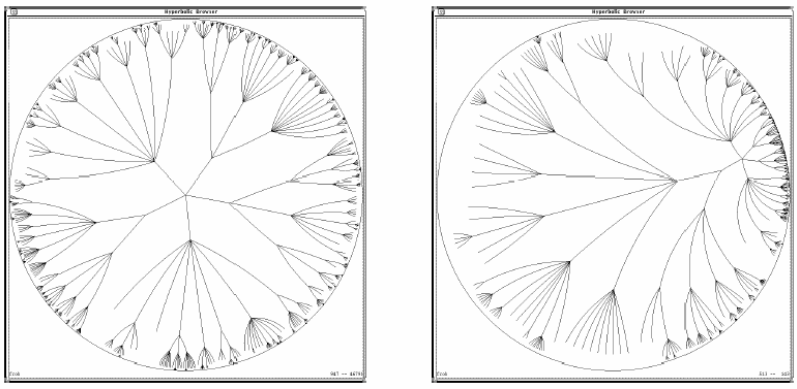


Figure 134 – vue hyperbolique d'un arbre de 1004 nœuds – changement de focus

4.2.1.4 Distorsion circulaire et radiale

Nous avons présenté en Figure 120 une technique de visualisation d'arbres en secteurs d'anneaux (Ring Tree). Deux techniques de distorsion ont été proposées pour explorer un Ring Tree (Yang, Ward et al. 2003) :

Une distorsion circulaire obtenue par « cliquer glisser », donne davantage d'importance à une branche en augmentant le secteur angulaire qui lui est associé (Figure 135).

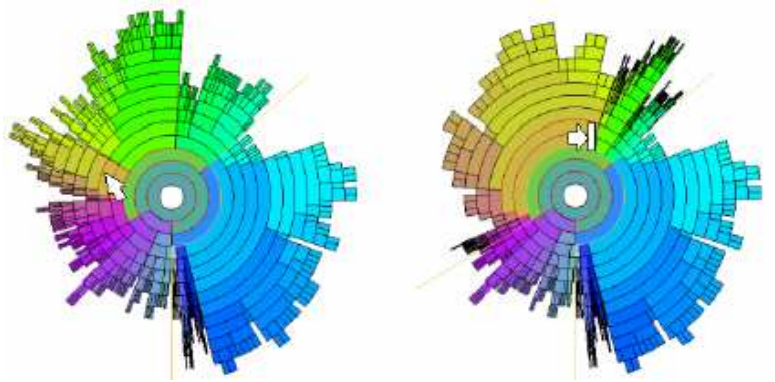


Figure 135 – Ring Tree – distorsion circulaire

Une distorsion radiale élargit une certaine couche à une distance choisie du focus (Figure 136) (par cliquer glisser). L'utilisation de couleurs affectées aux secteurs aide à la compréhension de la transformation.

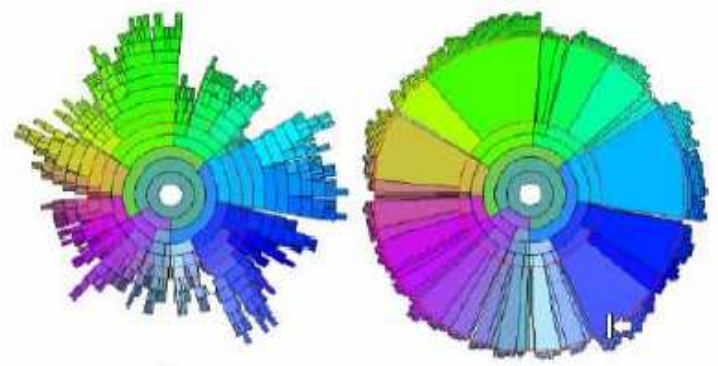


Figure 136 – Ring Tree – distorsion radiale

4.2.1.5 Deux espaces : détail + contexte

Certaines techniques séparent l'espace en deux : contexte et détail, comme le navigateur bifocal (Freitas, Luzzardi et al. 2002).

Cet outil de visualisation d'arbres, divise l'arbre en deux sous-arborescences à partir d'un focus utilisateur (Figure 137) : le détail (sous-arborescence de racine le focus) et le contexte (sous-arborescence de racine le nœud père du focus).

Une animation adaptée facilite la compréhension lors du changement de focus.

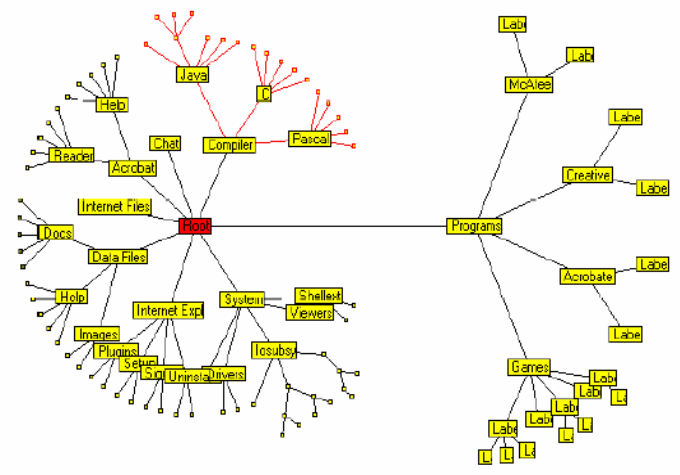


Figure 137 – navigateur bifocal (a) contexte (b) détail autour du focus

4.2.2 Zoom sémantique

L'utilisation d'un zoom sémantique est basé sur le calcul du degré d'intérêt (DOI) (Furnas 1986). Le degré d'intérêt d'un nœud dépend de l'importance du nœud et de sa distance au focus.

Le zoom sémantique est appliqué indépendamment de la technique de dessin utilisée.

4.2.2.1 Exploration d'arbres

Pour faciliter l'exploration de gros arbres (Figure 116), on peut utiliser un filtrage de branches prenant en compte l'importance des branches et leur distance au focus (Figure 138).

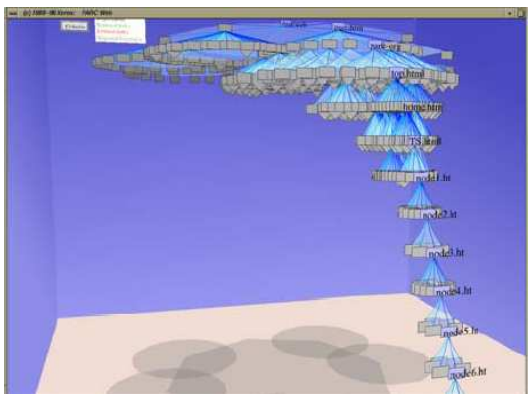


Figure 138 – Cone Tree – zoom sur une branche

Space Tree optimise le placement de l'arbre tout en facilitant l'ouverture et la fermeture de branches (Plaisant, Grosjean et al. 2002) (Figure 139). Une animation des nœuds permet de passer en douceur d'une vue à une autre.

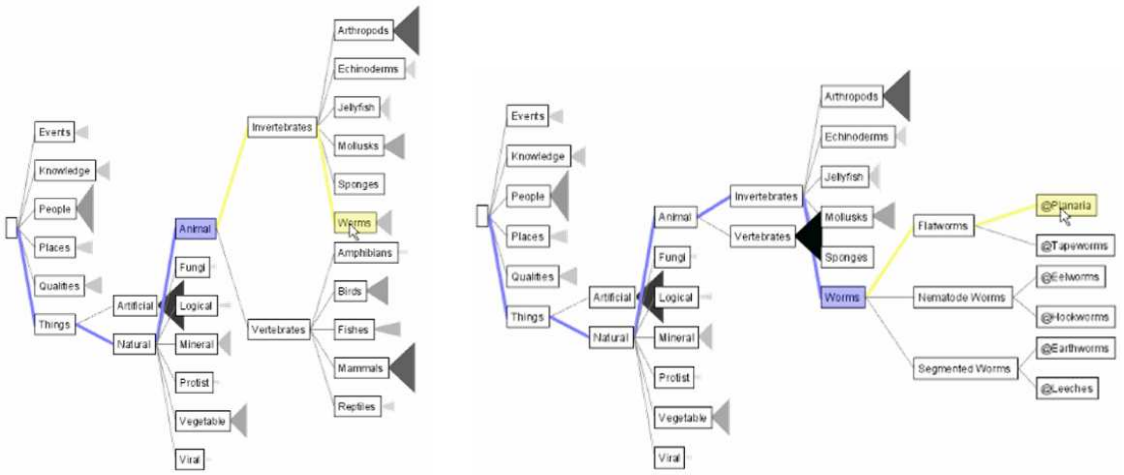


Figure 139 – Space Tree (a) avant sélection (b) après sélection

4.2.2.2 Exploration de graphes

L'utilisation conjointe d'une distorsion géométrique et structurelle facilite la navigation dans un graphe multi-échelles (Schaffer, Zhenping et al. 1996; van Ham et van Wijk 2004) (Figure 140).

Un cluster trop peu important ou trop loin du focus est « caché » dans son cluster père.

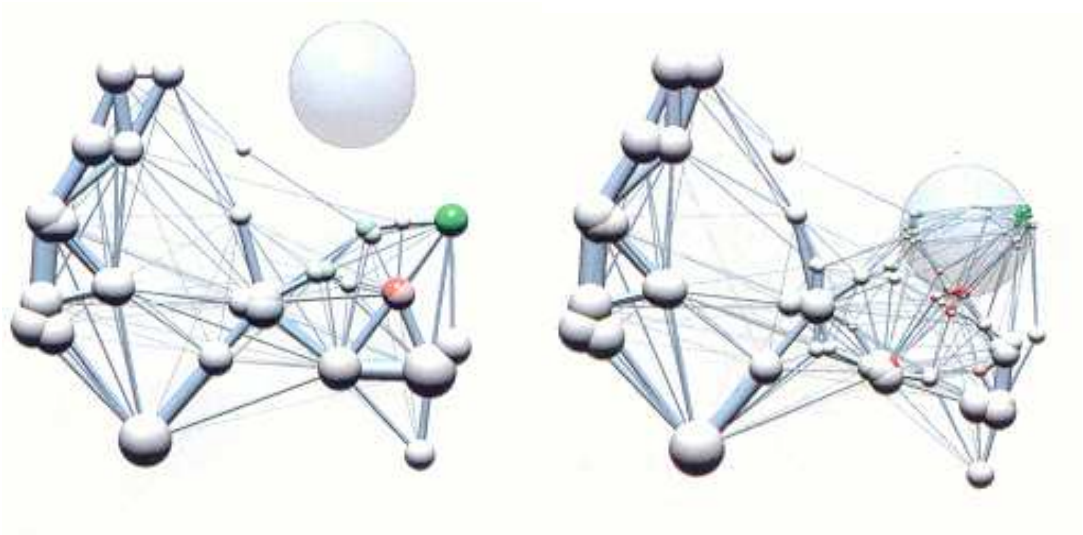


Figure 140 – zoom sémantique + géométrique dans graphe « petit monde »

4.3 Nouvelles techniques de visualisation multi-échelles

Nous proposons, dans cette section, deux techniques de visualisation d'arbres de silhouettes et d'arbres de clusters emboîtés. Elles reposent sur des techniques de visualisation d'arbres existantes possédant de bonnes propriétés que nous énonçons en section suivante.

4.3.1 Caractéristiques attendues

Nous avons introduit en section 3.6 une technique de clustering produisant des arbres de silhouettes (et de clusters) emboîtés ayant une structure d'arbre composé multi niveaux.

Cette structure générique, introduite en section 3.1.2.3, est un cas particulier de graphe composé. En ce sens, elle peut être visualisée à l'aide des techniques décrites en section 4.1.3.

Cependant les techniques classiques de visualisation multi-échelles de graphes ne prennent pas en compte toute la complexité d'un arbre de silhouettes. Notamment, chaque silhouette a une forme « gantée » déterminée par l'arbre de clusters qu'elle contient.

Propriété 35. Nous recensons les caractéristiques d'une technique de visualisation et d'interaction adaptée aux arbres de silhouettes emboîtées.

- **Vue centrée sur un focus de partitionnement** : l'arbre de silhouettes emboîtées est construit à partir d'un focus de partitionnement (éventuellement le même que le focus de filtrage). Il parait ainsi logique de proposer une visualisation autour de ce focus.

- **Vue hiérarchique** : lors de la construction de l'arbre de silhouettes et de l'arbre de clusters associé, les nœuds sont initialement organisés suivant leur distance au focus. Chaque cluster contient des nœuds à même distance du focus. Il paraît alors naturel de proposer une vue hiérarchique.

- **Vues emboîtées** : pour visualiser la superposition des différents arbres de silhouettes, il est souhaitable d'utiliser une visualisation en zones emboîtées (sans recouvrement). Le lien entre une silhouette et sa silhouette père est alors représenté visuellement par une appartenance (et non une arête comme dans les autres visualisations).

La vue horizontale décrite en section 4.1.1.1 est hiérarchique et basée sur un focus. Nous montrerons en section 4.3.2 comment la modifier pour intégrer des clusters emboîtés.

La vue Ring Tree (section 4.2.1.4) possède les trois caractéristiques. Nous verrons en section 4.3.3 qu'elle peut être adaptée pour visualiser des arbres de silhouettes emboîtées.

Nous utiliserons également des techniques proches de celles décrites en section 4.1.4 pour éviter le recouvrement de clusters.

4.3.2 Arbres de clusters emboîtés

Nous avons développé initialement (Boutin et Hascoët 2004) un algorithme de dessin d'arbre horizontal dont chaque nœud (appelé cluster englobant) est lui-même un arbre d'inclusion de clusters (onglets emboîtés). Pour simplifier la vue, les clusters singletons ne possèdent pas d'onglet associé.

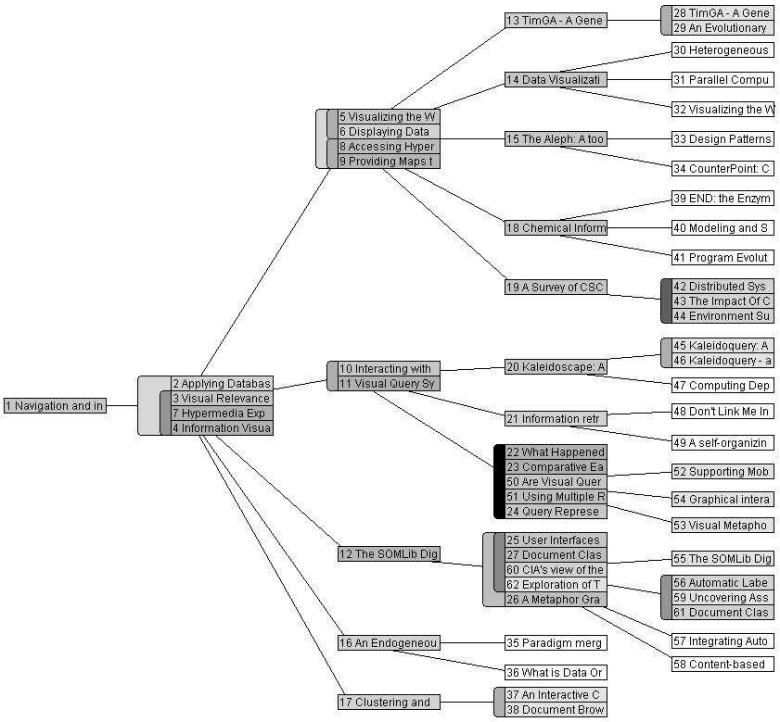


Figure 141 – arbre composé multi niveaux – petit graphe CiteSeer – vue horizontale

Exemple 72. Nous considérons le petit graphe de CiteSeer (section 1.3.1.2) incluant les 62 pages à distance au plus 4 du focus : « Navigation and interaction within graphical bookmarks » (Hascoët 1999). La Figure 123 présente une vue obtenue avec un algorithme basé sur un modèle de forces. La Figure 141 présente l'arbre de clusters emboîtés correspondant.

Propriété 36. Une vue horizontale a différents atouts :

- Pris en compte de la distance des clusters au focus et de leur niveau d'emboîtement.
- Affichage sans recouvrement du titre (ou d'une partie du titre) des nœuds.
- Uniquement des arêtes entre clusters englobants : pas de problème de recouvrement.
- Calcul rapide de la vue (comparé à un algorithme basé sur un modèle de forces).
- Réduction / expansion possible de clusters ou de branches de l'arbre.
- Mise en surbrillance des nœuds ou clusters adjacents du nœud ou cluster sélectionné.

Une telle vue ne permet cependant pas d'optimiser le placement des nœuds et clusters lorsque le graphe est de taille importante.

Nous verrons au paragraphe suivant qu'une vue radiale permet à la fois de représenter un arbre de clusters et un arbre de silhouettes.

Il semble naturel de considérer des silhouettes (connexes) plutôt que des clusters (éventuellement non connexe) surtout si l'on veut définir la notion de « cohésion ».

Il est intéressant d'affecter une couleur à un cluster (voir Figure 141) ou une silhouette dépendant de sa cohésion. Cependant le calcul n'est pas naturel pour un cluster non connexe.

4.3.3 Arbre de silhouettes emboîtées

Pour dessiner un arbre de silhouettes à partir d'un focus, nous choisissons naturellement un algorithme de dessin radial (section 4.1.2.1). Nous créons ensuite des silhouettes en utilisant une technique proche de celle utilisée pour les Ring Trees (section 4.2.1.4).

Diverses API de dessin de graphes peuvent être utilisées : Tulip (Auber 2002; Auber 2003), Prefuse (Heer, Card et al. 2005), InfoVis Toolkit (Fekete 2004).

L'un des algorithmes disponible avec l'API de Prefuse assure une représentation radiale du graphe et une animation lors d'un changement de focus : les nœuds suivant un chemin intuitif entre leurs positions de départ et d'arrivée (Heer, Card et al. 2005).

Exemple 73. Nous montrons en Figure 142 un graphe placé à partir d'un arbre couvrant. A chaque nœud est associé un secteur angulaire englobant tous ses descendants dans l'arbre couvrant. Cette vue a été obtenue à partir de l'API de Prefuse.

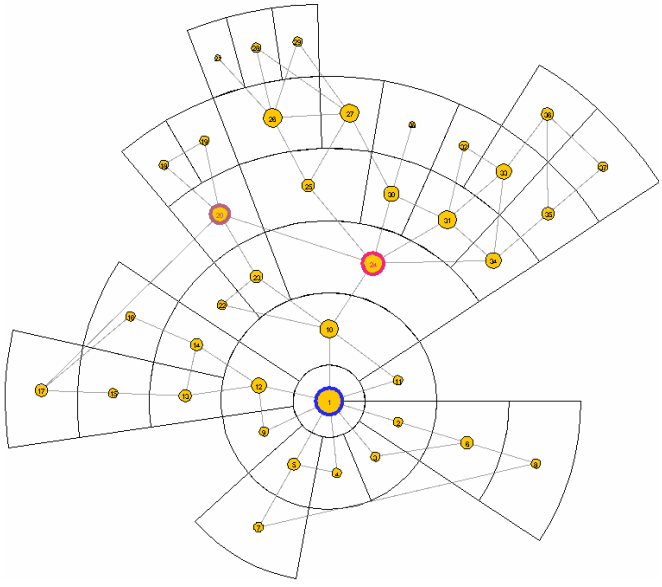


Figure 142 – API Prefuse – dessin des secteurs angulaires

Jérôme Thièvre a implémenté (Boutin, Thièvre et al. 2005) une adaptation de l'algorithme de Prefuse (Heer, Card et al. 2005), incluant la représentation automatique de clusters et de silhouettes.

Les nœuds sont initialement ordonnés par couche à partir d'un focus en tenant compte du niveau d'emboîtement des clusters (dans divers arbres de clusters T^p). Un arbre couvrant est ensuite extrait du graphe en utilisant l'ordre des nœuds et les relations entre clusters. Le choix adéquat de l'arbre couvrant conditionne la cohésion et le non recouvrement des clusters e silhouettets.

Une fois les nœuds placés, chaque cluster (ou silhouette) est dessiné comme forme englobante de ses nœuds, à l'exception des clusters (ou silhouettes) triviaux, pour ne pas surcharger la vue. A chaque silhouette est associée une couleur. Ainsi, les clusters ont la couleur de leur silhouette.

Exemple 74. Nous représentons (Figure 143) l'arbre de clusters et l'arbre de silhouettes obtenus à partir du focus 1 dans le graphe école.

Le passage de l'arbre de clusters à l'arbre de silhouettes se fait par simple « continuité » géométrique. La silhouette résultante est connexe car c'est un arbre de clusters (Figure 143).

Propriété 37. Par construction de l'arbre de silhouettes, il n'est pas indispensable de visualiser les arêtes du graphe pour comprendre sa structure. C'est un atout pour visualiser de gros graphes clustérisés.

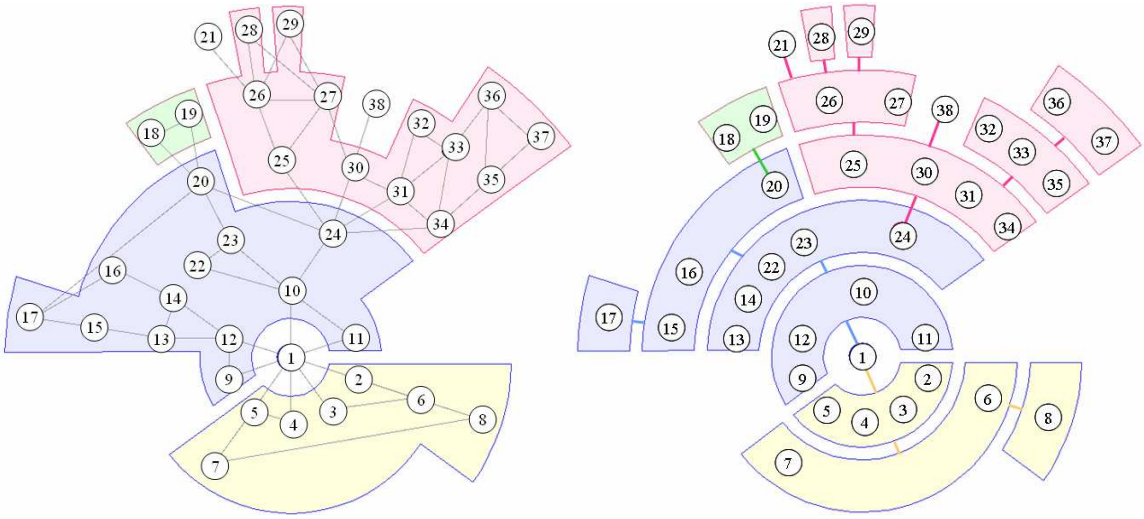


Figure 143 – arbre de silhouettes et arbre de clusters du graphe école

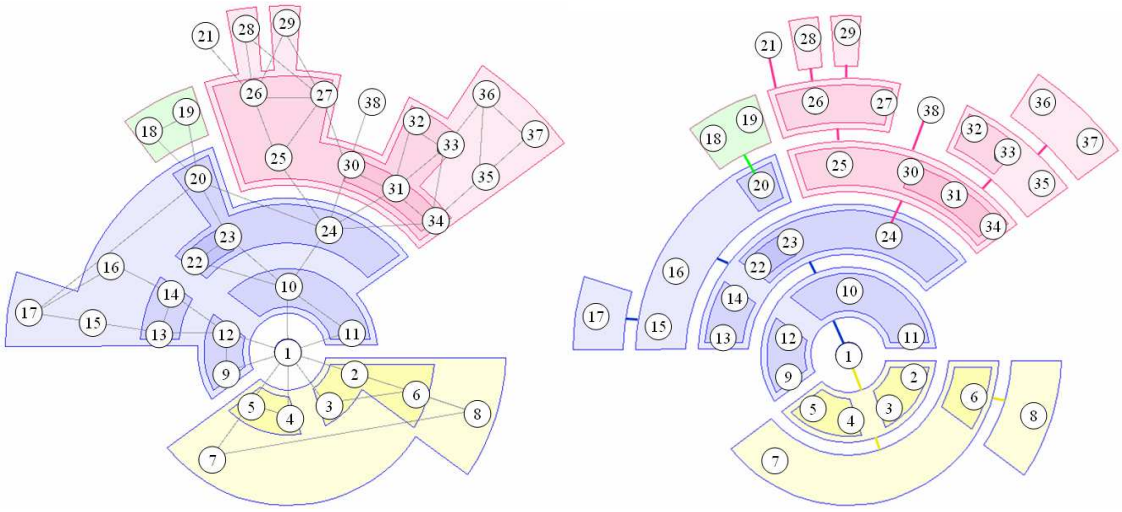


Figure 144 – arbre de silhouettes et arbre de clusters emboîtés du graphe école

L'API de Jérôme Thièvre permet la visualisation de silhouettes et clusters emboîtés. L'utilisation de couleurs transparentes permet de percevoir des niveaux d'emboîtement dans l'arbre d'inclusion de clusters ou silhouettes.

Exemple 75. La Figure 144 présente une vue de clusters emboîtés et silhouettes emboîtées.

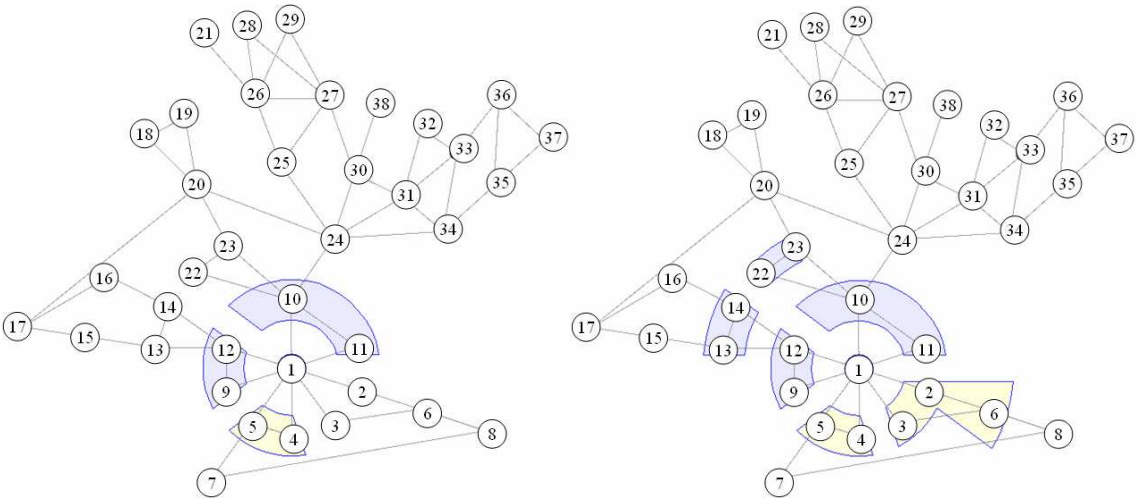


Figure 145 – arbre de silhouettes (a) rayon 1 (b) rayon 2

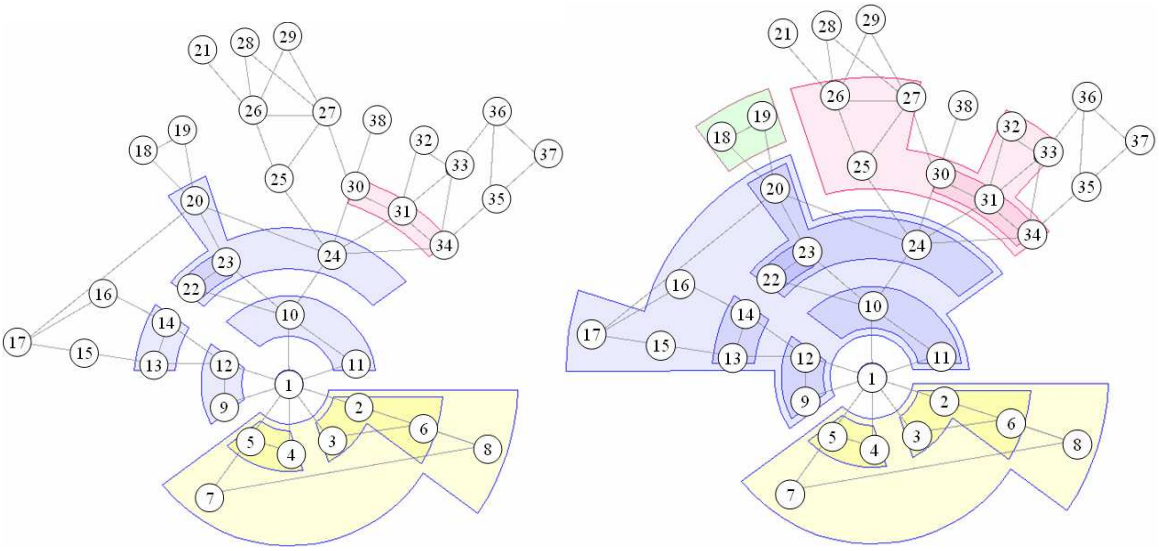


Figure 146 – arbre de silhouettes emboîtées (a) rayon 3 (b) rayon 4

Exemple 76. Nous présentons (Figure 145, Figure 146) l'arbre silhouette multi-échelles en considérant différents voisinages autour du focus de rayon : 1, 2, 3 et 4. La Figure 147b présente le MuSi Tree (global) de rayon 5.

4.3.4 Changement de focus

Exemple 77. Nous présentons le graphe d'école avec cinq focus potentiels (Figure 147a), et donnons une vue pour chaque focus (Figure 147, Figure 148, Figure 149).

Propriété 38. Le changement de focus ne modifie pas l'arbre biparti des composantes biconnexes et points d'articulation (voir Figure 109). Les silhouettes sont retrouvées après changement de focus même si certains nœuds d'articulation peuvent changer de silhouette (voir section 3.6.2.2).

Une animation des nœuds permet de comprendre la transformation de la vue lors du changement de focus (Heer, Card et al. 2005).

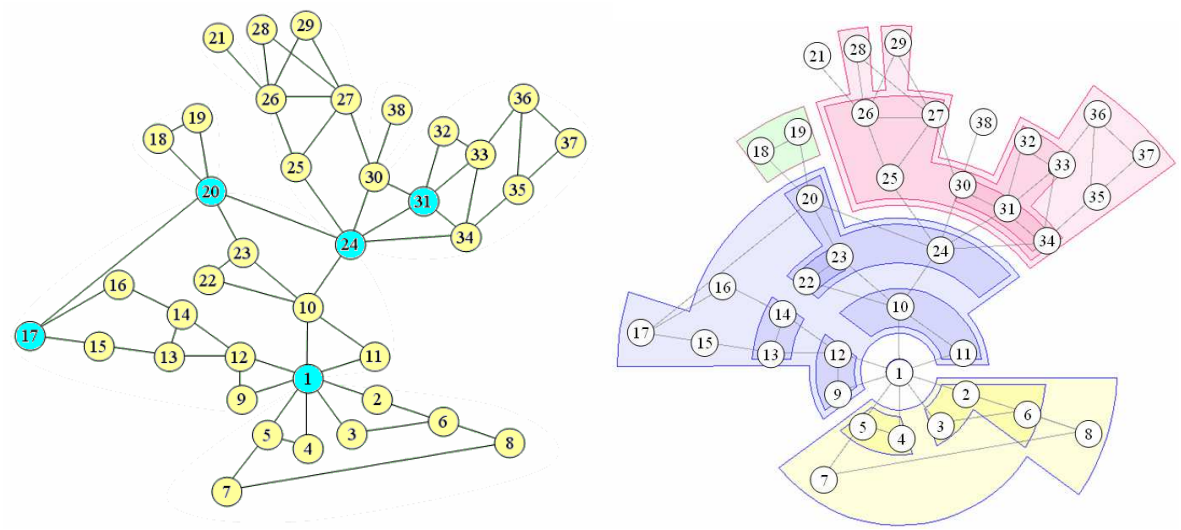


Figure 147 – (a) différents focus (b) focus 1

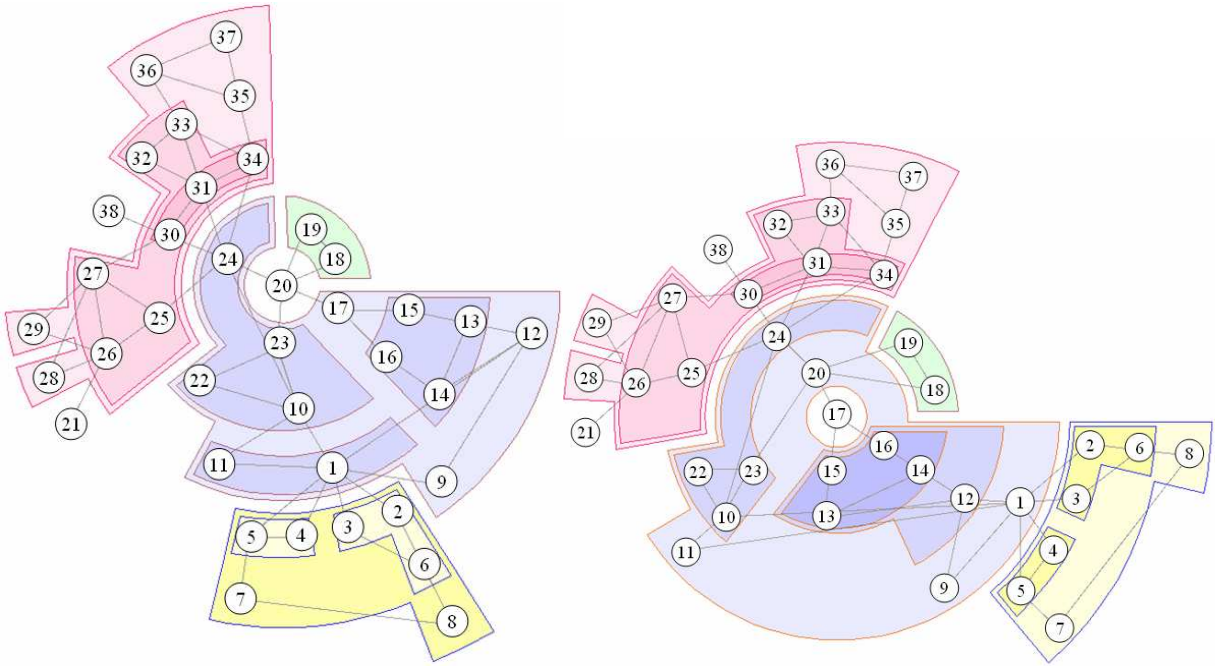


Figure 148 – (a) focus 20 (b) focus 17

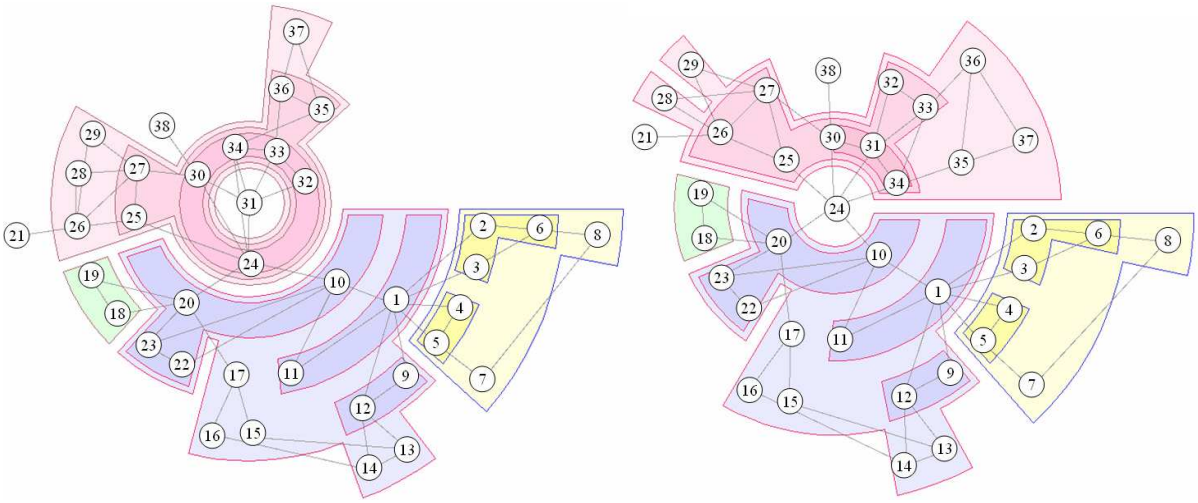


Figure 149 – (a) focus 31 (b) focus 24

4.3.5 Techniques « focus + contexte » – perspectives

Nous avons essentiellement intégré (à l'API de Jérôme Thièvre) des techniques d'interactions prenant en compte un changement de focus.

Pour la gestion de gros graphes, il peut être souhaitable d'inclure des techniques de zoom géométrique et sémantique adaptées à la structure d'arbre de silhouettes emboîtées. Nous énonçons succinctement quelques pistes :

Les techniques de distorsion radiales et circulaires décrites en section 4.2.1.4 peuvent être appliquées pour réduire ou agrandir certaines silhouettes ou branches de clusters.

Il est également possible d'utiliser une technique de fisheye géométrique (section 4.2.1.1) pour faciliter l'exploration de silhouettes (notamment à la périphérie).

Dans le cas de gros graphes comprenant une architecture complexe de silhouettes emboîtées, il est intéressant de faire appel à un fisheye sémantique prenant en compte l'importance de la silhouette et sa distance au focus (section 4.2.2). Les silhouettes périphériques de faible importance peuvent éventuellement être supprimées. Les silhouettes emboîtées complexes, éloignées du focus, peuvent aussi être réduites à une silhouette simple.

L'application conjointe d'un zoom géométrique et sémantique peut assurer une exploration simplifiée de la structure.

Ces différentes pistes feront l'objet de travaux ultérieurs.

« La théorie, c'est quand on sait tout et que rien ne fonctionne.
La pratique, c'est quand tout fonctionne et que personne ne sait pourquoi. »

Albert Einstein

Chapitre 5 Evaluation

Nous présentons, dans ce chapitre, une classification des critères d'évaluation des techniques de partitionnement. Nous introduisons également de nouveaux critères assurant une normalisation des critères existants.

Nous proposons ensuite une évaluation empirique des nouvelles techniques de filtrage et partitionnement à partir des réseaux d'interactions réels filtrés en section 1.3.

Enfin nous étudierons des critères théoriques de qualité afin de mieux définir :

- un « bon » filtrage
- un « bon » partitionnement
- un « beau » graphe « arboré »
- un « bel » arbre de silhouettes emboîtées

5.1 Critères d'évaluation d'un partitionnement

Cette section a fait l'objet d'un article (Boutin et Hascoët 2004).

Les techniques de clustering de graphe consistent en un partitionnement des nœuds du graphe en clusters. L'objectif est le plus souvent de maximiser la connectivité intra cluster (compacité) et de minimiser la connectivité inter clusters (séparabilité). Ce principe repose sur le théorème de Huygens : « La variabilité globale se décompose en une variabilité inter groupes et une variabilité intra groupe ».

Différents indices de compacité et de séparabilité sont calculés en analyse de données. Malheureusement ces calculs supposent généralement que les points (nœuds) ont des coordonnées dans un espace euclidien ce qui n'est pas le cas a priori pour les nœuds d'un graphe. Nombreux indices ont ainsi été transformés et adaptés à l'usage des graphes. Peu de travaux présentent des indices de validité de clustering spécifiques aux graphes.

Dans ce chapitre nous proposons une synthèse des divers indices caractérisant un clustering de graphe. Nous décrivons des indices de compacité, puis de séparabilité de clusters. Nous exposons ensuite des indices de clustering locaux. Enfin nous citons des indices externes permettant de comparer deux résultats de clustering.

Nous proposons également en section 5.2 certaines améliorations des indices existants.

Pour faciliter la compréhension de cette section, il nous a fallu revoir les notations d'origine des auteurs de sorte à faire apparaître tous les indices dans un cadre uniforme.

5.1.1 Notations et définitions préliminaires

On considère un graphe G non orienté à N nœuds v_i et E arêtes $e_{ij} = \{v_i , v_j\}$ tel que :

- G est partitionné en $\{C_1,\ldots, C_K\}$.
- N_i est le nombre de nœuds de C_i.
- E_i est le nombre d'arêtes de C_i.
- E_{ij} est le nombre d'arêtes entre C_i et C_j.
- E_i' est le nombre d'arêtes entre C_i et les autres clusters.

Nous noterons d (v_i, v_j) la distance dans le graphe G entre les sommets v_i et v_j, et d la distance moyenne dans le graphe. Nous noterons diam le diamètre moyen (voir section 1.1.3).

5.1.2 Compacité de graphe

5.1.2.1 Compacité basée sur la densité des arêtes

Les indices de compacité les plus simples sont basés uniquement sur le nombre de nœuds et d'arêtes. Par exemple : $\frac{E}{N}$ ou $\frac{E}{N^2}$.

De tels indices sont faciles à calculer mais ils ne prennent pas en compte la structure réelle du graphe. En effet deux graphes peuvent avoir une structure différente mais présenter le même nombre de nœuds et d'arêtes.

Exemple 78. En Figure 150, G_1 et G_2 ont même compacité E/N = 13/8.

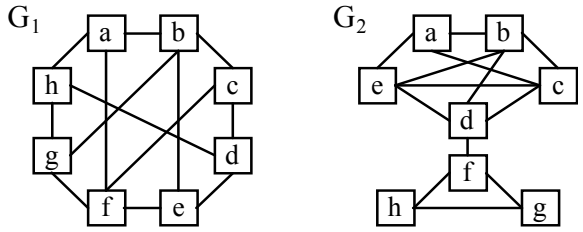


Figure 150 – compacité de graphes

5.1.2.2 Indice de compacité Cp

Définition 79. Un indice de compacité noté Cp a été introduit (Botafogo, Rivlin et al. 1992) prenant en compte la connectivité du graphe. Pour un graphe connexe G de diamètre maximal Q, C_p est calculé en temps quadratique par :

$$Cp = \frac{Max - \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d(v_i,v_j)}{Max - Min}$$

où Min = N(N-1)/2 et Max = Q.N(N-1)/2 sont minorant et majorant de $\sum_i \sum_j d(v_i,v_j)$.

En effet, il y a N(N-1)/2 couples (v_i, v_j) distincts et d (v_i, v_j) varie entre 1 et Q.

Exemple 79. Pour les graphes présentés en Figure 150, Cp (G_1) = 0,48 et Cp (G_2) = 0,69.

Pour calculer la compacité d'un graphe non connexe, on peut définir la distance entre deux nœuds non connectés par une valeur arbitrairement grande Q. Cependant l'indice résultant dépend du choix de la valeur Q.

5.1.3 Séparabilité

Pour évaluer la qualité d'un partitionnement de graphe nous comparons le plus souvent inter et intra connectivité des clusters. L'intra et l'inter connectivité peuvent être définis de diverses manières. Nous présentons dans cette section les différents indices correspondants.

5.1.3.1 Indices basés sur diamètre et distance

Indice de Dunn :

Pour définir cet indice nous supposons :

- C_i et C_j sont les clusters les plus proches au sens de la distance moyenne d.
- C_h est le cluster de plus grand diamètre.

Définition 80. L'indice de Dunn (Dunn 1974; Eades 1996; Bezdek et Pal 1998; Halkidi, Batistakis et al. 2001; Bolshakova et Azuaje 2003) est défini par :

$$D(C)=\frac{d(C_i,C_j)}{diam(C_h)}$$

Où d (C_k, C_j) et diam (C_k) définissent respectivement l'inter et l'intra connectivité.

Propriété 39. D n'est pas robuste car il ne prend en compte qu'un nombre réduit de clusters et de relations entre ces clusters. Il est sensible aux clusters aberrants.

Exemple 80. En Figure 151 sont représentés deux graphes de même indice de Dunn mais de connectivité différente. En effet, dans les graphes G_3 et G_4, les clusters les plus proches sont C_2 et C_3 et le cluster de plus grand diamètre est C_1.

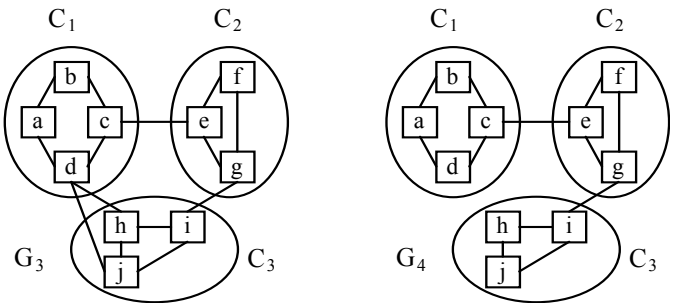


Figure 151 – indices de Dunn identiques pour les deux graphes

Indice de Davies Bouldin :

L'indice DB (Davies et Bouldin 1979; Bolshakova et Azuaje 2003) est défini par :

$$DB=\frac{1}{K}\sum_{i=1}^{K}\max_{j\neq i}\left[\frac{diam(C_i)+diam(C_j)}{d(C_i,C_j)}\right]$$

Propriété 40. Chacun des K clusters intervient en relation avec tous les autres. Ainsi le calcul est plus robuste (et plus long) que celui proposé par Dunn.

5.1.3.2 Indices basés sur arêtes intra et inter clusters

Un indice de coupe appelé MinMaxCut :

Définition 81. Les liens entre deux clusters C_i et C_j définissent des coupes (cut).

Définition 82. MinMaxCut est défini par (Ding, Xiaofeng et al. 2001) :

$$MinMaxCut=\sum_{i=1}^{K}\frac{E_i'}{E_i}.$$

L'objectif est de minimiser la fonction MinMaxCut (Zhao et Karypis 2001) c'est-à-dire le nombre de coupes. Malheureusement cet indice ne prend pas en compte le nombre de nœuds N_i du cluster C_i.

Exemple 81. En Figure 152, MinMaxCut vaut 0,5 pour les deux graphes G_5 et G_6. Pourtant G_6 contient des composantes de plus petit diamètre.

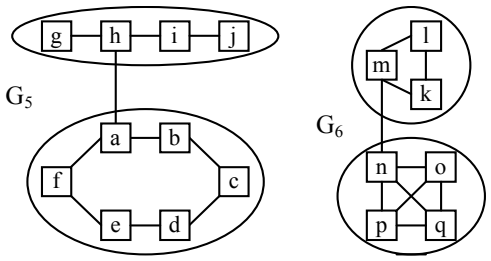


Figure 152 – MinMaxCut identiques pour conductance et couverture

Couductance d'une coupe :

Définition 83. Conductance d'une coupe entre C_i et C_j (Brandes, Gaertler et al. 2003) :

$$Conductance\ (C_i,C_j)=\frac{E_{ij}}{\min\left(E_i,E_j\right)}$$

Propriété 41. Les clusters à faible conductance sont bien séparés.

Exemple 82. En Figure 152, la conductance vaut 1/3 pour les deux graphes G_5 et G_6.

Couverture d'une partition de graphe :

Définition 84. La couverture (coverage) d'une partition de graphe C est défini par la fraction des arêtes intra cluster sur l'ensemble des arêtes du graphe (Brandes, Gaertler et al. 2003) :

$$Cov(C)=\frac{\sum_{i=1}^{K} E_i}{E}$$

Propriété 42. Ainsi plus la couverture est importante, meilleur est le partitionnement. Cet indice est facile à calculer mais il ne prend pas en compte le nombre de nœuds N_i des clusters C_i.

Exemple 83. En Figure 152, la couverture vaut 9/10 pour les deux graphes G_5 et G_6.

5.1.3.3 Indices basés sur le nombre de nœuds et d'arêtes

Performance d'un partitionnement :

Performance d'un partitionnement (van Dongen 2000; Brandes, Gaertler et al. 2003) :

$$Perf(G) = 1 - \frac{\sum_{i<j}^{K} E_{ij} + \sum_{i=1}^{K}\left(\frac{N_i(N_i-1)}{2} - E_i\right)}{\frac{N(N-1)}{2}} \text{ ou } Perf(G) = 1 - \frac{\|False+\| + \|False-\|}{\frac{N(N-1)}{2}}$$

- $\|False-\|$ est le nombre d'arêtes inter clusters.
- $\|False+\|$ est le nombre de couples (v_i, v_j) non adjacents d'un même cluster.

La performance peut être calculée par l'expression :

$$Perf(C) = 1 - \frac{E\left(1-2\,\mathrm{cov}(C)\right) + \sum_{i=1}^{K}\frac{N_i(N_i-1)}{2}}{\frac{N\cdot(N-1)}{2}}$$

Qualité de modularité MQ :

Définition 85. L'indice de qualité de modularité MQ est défini par la différence entre la connectivité intra et inter clusters (Mancoridis, Mitchell et al. 1998) :

- La connectivité intra cluster est définie par : $intra\ (C_i) = \frac{E_i}{N_i(N_i-1)/2}$ où $N_i(N_i\text{-}1)/2$ est le nombre maximum d'arêtes intra cluster.

– Connectivité moyenne intra cluster : $intra = \dfrac{\sum_{i=1}^{K} \dfrac{E_i}{N_i(N_i-1)/2}}{K}$

– Connectivité inter clusters : $inter\ (C_i, C_j) = \dfrac{E_{ij}}{N_i N_j}$

– Connectivité moyenne inter clusters : $inter = \dfrac{\sum_{i<j}^{K} \dfrac{E_{ij}}{N_i N_j}}{K(K-1)/2}$

$$MQ = intra - inter = \frac{\sum_{i=1}^{K} \frac{E_i}{N_i(N_i-1)/2}}{K} - \frac{\sum_{i<j}^{K} \frac{E_{ij}}{N_i N_j}}{K(K-1)/2}$$

Malheureusement cet indice calcule des moyennes simples (non pondérées) alors que les clusters peuvent être de taille différente.

Exemple 84. En Figure 153, MQ = 0,64. Le calcul ignore que C_1 est plus grand que C_2.

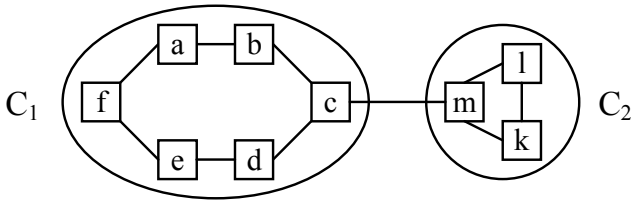


Figure 153 – clusters de taille différente

5.1.4 Indices de clustering locaux

Nous avons précédemment décrit des indices de compacité globaux étudiant relations entre clusters. Nous introduisons à présent des indices de clustering locaux pour chaque nœud. De tels indices permettent notamment de définir l'attachement d'un nœud à son cluster. Les nœuds ayant peu de relations avec leur cluster peuvent alors être supprimés ou déplacés.

5.1.4.1 L'indice de silhouette

Précisons que le terme silhouette ici n'a rien à voir avec le terme défini en section 3.6.2.

Indice GS :

Considérons un nœud v_i appartenant à un cluster C_j. Notons C_h le cluster le plus proche du nœud v_i (distance moyenne). L'indice de silhouette est défini par (Rousseeuw 1987; Bolshakova et Azuaje 2003) :

$$s(v_i) = \frac{d(v_i,C_h) - d(v_i,C_j)}{\max\left(d(v_i,C_j), d(v_i,C_h)\right)}$$

Propriété 43. L'indice de silhouette est borné : $-1 \leq s(v_i) \leq 1$. De plus, lorsque $s(v_i)$ est proche de 1, v_i est dit « bien clustérisé ». Quand $s(v_i)$ est inférieur à 0, v_i doit être assigné au cluster le plus proche.

Exemple 85. En Figure 153, s(m) = (2,5-1) / 2,5 = 0,6 et s(k) = (3,5-1) / 3,5 = 0,71.

Pour un cluster donné C_i, l'indice de silhouette S_i est définie par :

$$S_j = \frac{\sum_{i=1}^{Nj} s(v_i)}{N_j}$$

L'indice de silhouette global GS est définie par :

$$GS = \frac{\sum_{j=1}^{K} S_j}{K}$$

5.1.4.2 Mesures de couverture

Mesure de couverture naïve :

Soit v_i un nœud appartenant au cluster C_j. Soit $N(v_i)$ le voisinage de v_i.

Définition 86. L'ensemble des faux positifs noté $False_i+$ est défini comme l'ensemble des nœuds de C_i n'appartenant pas à $N(v_i)$. L'ensemble des faux négatifs noté $False_i-$ est défini comme l'ensemble des nœuds appartenant à $N(v_i)$ mais pas à C_i. La mesure de couverture naïve de v_i est définie (van Dongen 2000; Ramaswamy, Iyengar et al. 2003) par :

$$Cov(v_i) = 1 - \frac{\|False_i+\| + \|False_i-\|}{N-1}$$

Propriété 44. L'indice de performance défini précédemment est la moyenne des $Cov(v_i)$.

Mesure de couverture normalisée :

Définition de la mesure de couverture normalisée (Ramaswamy, Iyengar et al. 2003) :

$$ScalCov(v_i) = 1 - \frac{\|False_i+\| + \|False_i-\|}{\|C_j \cup N(v_i)\|}$$

Si $||True_i+|| = ||True_i-|| = 0$ alors $||C_j \cup N(v_i)|| = ||False_i+|| + ||False_i-||$ et $ScalCov(v_i) = 0$ contrairement à $Cov(v_i)$.

5.1.4.3 Indice de clustering dans les graphes « petit monde »

Un indice de clustering spécifique aux graphes « petit monde » a été défini (Watts et Strogatz 1998). La métrique mesure la densité d'arêtes au voisinage d'un nœud v_i. Si on note $N(v_i)$ le voisinage d'un nœud v_i comprenant n nœuds et e arêtes entre ces nœuds (sans compter les arêtes adjacentes à v_i), on détermine :

$$c(v_i)=\frac{e}{\frac{n(n-1)}{2}}$$

Exemple 86. En Figure 153, c(m) = 1/3, c(k)=1, c(c)=0.

L'indice de clustering global du graphe G est obtenu en moyennant les indices $c(v_i)$:

$$c(G)=\frac{\sum_{i=1}^{N} c(v_i)}{N}$$

5.1.5 Indices externes pour comparer deux partitions

Dans cette section, nous présentons différents indices permettant de comparer deux partitions $P = \{C_1,\ldots, C_K\}$ et $P' = \{C'_1,\ldots, C'_T\}$.

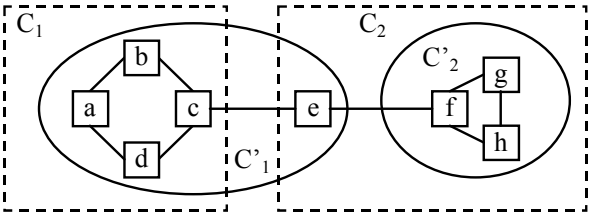


Figure 154 – comparaison des partitions P et P'

5.1.5.1 Mesures de co-partitionnement

Nous décrivons des indices de co-partitionnement (Halkidi, Batistakis et al. 2001; Law et Jain 2003) comparant deux partitions P et P' de N(N-1)/2 couples de nœuds (v_i, v_j).

La répartition des couples (v_i, v_j) est la suivante :

Partition de $\{(v_i, v_j)\}$	même cluster dans P'	Différents clusters dans P'
même cluster dans P	a	c
Différents clusters dans P	b	d

Coefficient de Jaccard :

Le coefficient de Jaccard est un indice de similarité entre les partitions P et P' :

$$J = \frac{a}{a+b+c}$$

J définit la probabilité que deux nœuds appartenant à un même cluster dans une partition appartiennent également à un même cluster dans l'autre partition.

Indice de Folkes et Mallows :

Folkes et Mallows ont introduit un autre indice :

$$FM = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}$$

a/(a+b) est la probabilité que deux nœuds appartiennent à un même cluster dans P s'ils appartiennent à un même cluster dans P'. De même, a/(a+c) est la probabilité que deux nœuds appartiennent à un même cluster dans P' s'ils appartiennent à un même cluster dans P.

Statistique Rand :

La statistique Rand mesure la similarité entre deux partitions P et P' :

$$R = \frac{a+d}{a+b+c+d}$$

Contrairement à J et FM, la statistique Rand inclus d dans son calcul. Elle calcule la probabilité que deux nœuds appartiennent soit à un même cluster soit à deux clusters différents à la fois dans P et P'.

Statistique de Hubert et Arabie (Rand normalisée) :

La statistique de Rand a été modifiée (Hubert et Arabie 1985) de façon à ce que sa valeur soit majorée par 1 et qu'elle vale 0 pour un graphe aléatoire :

$$Hubert = \frac{a \cdot d - b \cdot c}{\sqrt{(a+b)\cdot(c+d)\cdot(a+c)\cdot(b+d)}}$$

Propriété 45. Remarquons que cette statistique est également appelée statistique Phi. Elle est en fait équivalente au coefficient de corrélation de Pearson (Pearson 1900). Phi peut être calculé à partir de la statistique du Chi deux :

$$\text{Hubert}^2 = \varphi^2 = \frac{X^2}{n} \text{ avec } n = a+b+c+d$$

Exemple 87. En Figure 154, a = 9, b = 4, c = 3, d = 12. Les indices valent J = 0,56 ; FM = 0,72 ; R = 0,75 et Hubert = 0,5.

5.1.5.2 Indices basés sur des mesures de probabilités

Nous décrivons des méthodes basées sur un calcul de probabilité pour comparer les partitions $P = \{C_1,\ldots, C_K\}$ et $P' = \{C'_1,\ldots, C'_T\}$. Si on note f_{ij} le pourcentage de nœuds du cluster C_i appartenant au cluster C'_j on obtient :

$$\forall i \in \{1,\ldots, K\}, \sum_{j=1}^{T} f_{ij} = 1$$

Coefficient de partitionnement :

(Bezdek et Pal 1998) ont introduit le coefficient de partitionnement PC.

Considérant un cluster C_i :

$$PC(C_i)=\sum_{j=1}^{T} f_{ij}^2$$

PC (C_i) est une valeur entre 1/T et 1. Si presque tous les nœuds de C_i appartiennent à un même cluster de C'$_j$ alors PC (C_i) est proche de 1. Si les nœuds de C_i sont répartis aléatoirement dans les différents clusters de P', alors PC (C_i) est proche de 1/T.

Le coefficient de partitionnement global est déterminé par :

$$PC(P/P') = \frac{1}{K}\sum_{i=1}^{K} PC(C_i) = \frac{1}{K}\sum_{i=1}^{K}\sum_{j=1}^{T} f_{ij}^2$$

PC (P/P') est également une valeur entre 1/T et 1. Si PC (P/P') est proche de 1/T, P et P' sont presque indépendants. De plus si PC (P/P') est proche de 1, alors P est « proche » de P'.

Exemple 88. En Figure 154, PC (P/P') = 0,84

Entropie de partitionnement :

(Bezdek et Pal 1998) ont défini l'entropie d'un partitionnement. Soit un cluster C_i :

$$Entropie(C_i) = -\sum_{j=1}^{T} f_{ij} \cdot \log(f_{ij})$$

en posant 0.log(0) = 0 c'est-à-dire la valeur limite de x.log (x) lorsque x tend vers 0.

L'entropie de C_i est une valeur entre 0 et log (T). Si presque tous les nœuds de C_i appartiennent à un même cluster C'$_k$ alors f_{ij} est proche de 0 pour $j \neq k$ et f_{ij} est proche de 1 pour j = k. Ainsi l'entropie de C_i est proche de 0 puisque 0.log (0) = 1.log (1) = 0. Par contre, si les nœuds de C_i sont répartis aléatoirement dans les clusters de P', alors f_{ij} est proche de 1/T et l'entropie de C_i est proche de log (T).

Une mesure globale d'entropie est définie par :

$$Entropie\ (P/P') = \sum_{i=1}^{K} \frac{N_i}{N} Entropy\ (C_i)$$

L'entropie globale est également une valeur entre 0 et log (T). Si Entropie (P/P') est proche de log (T) alors P et P' sont presque indépendants. Si l'entropie est proche de 1, P est « proche » de P'.

Exemple 89. Figure 154, entropie (C_1)=0,72 et entropie (C_2)=0 donc entropie (P/P')=0,45.

5.2 Nouveaux indices de qualité de partitionnement

Les indices Cp, MQ et GS introduits en section 5.1 ne prennent pas en compte la taille des clusters. Nous proposons de nouveaux indices Cp*, MQ* et GS* standardisés, palliant ce problème, ainsi que plusieurs autres variantes de MQ.

5.2.1 Nouvel indice de compacité standardisé Cp*

Nous proposons un nouvel indice de compacité normalisé noté Cp* indépendant de toute constante Q. Nous considérons en effet une mesure de similarité au lieu d'une distance.

Considérant deux nœuds v_i et v_j, nous définissons leur similarité $sim(v_i, v_j)$ par l'inverse de la distance $d(v_i, v_j)$ si les nœuds sont connectés, 0 sinon. La similarité varie entre 0 et 1.

Définition 87. Le nouvel indice Cp* est défini ainsi :

$$Cp^* = \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} sim(v_i, v_j)}{N(N-1)/2}$$

Propriété 46. Cp* est borné par 0 et 1. Si le graphe est complètement déconnecté, Cp* vaut 0. Si G est un graphe complet, Cp* vaut 1.

Exemple 90. En Figure 150, $Cp^*(G_1) = 0{,}73$ et $Cp^*(G_2) = 0{,}69$.

5.2.2 Nouvel indice de silhouette standardisé GS*

Nous proposons un nouvel indice GS* prenant en compte la taille des clusters :

$$GS^* = \frac{\sum_{j=1}^{K} N_j S_j}{\sum_{j=1}^{K} N_j} = \frac{\sum_{i=1}^{N} s(v_i)}{N}$$

5.2.3 Nouvelles mesures, variantes de MQ

Nous introduisons dans cette section différentes variantes de MQ prenant en compte la taille des clusters (section 5.2.3.1) la valuation des arêtes (section 5.2.3.2). Nous proposons également des valeurs de MQ normalisées (section 5.2.3.4) ou toujours positives comme la statistique F de Fisher (voir section 5.2.3.5).

5.2.3.1 Un nouvel indice pondéré noté MQ* :

L'indice MQ peut être modifié pour prendre en compte la taille des clusters. L'indice résultant noté MQ* est dit pondéré :

$$MQ^* = \frac{\sum_i E_i}{\sum_i \frac{N_i(N_i-1)}{2}} - \frac{\sum_{i<j} E_{ij}}{\sum_{i<j} N_i N_j}$$

Propriété 47. Contrairement à MQ, MQ* prend en compte connectivité et taille des clusters.

Exemple 91. En Figure 153, la taille de C_1 est supérieure à celle de C_2. C_1 a une plus petite connectivité intra cluster ainsi MQ* = 0,44 est inférieur à MQ = 0,64.

5.2.3.2 Nouvelle mesure MQ pour un graphe valué :

Nous proposons une nouvelle mesure MQ pour un graphe valué :

$$MQ = \frac{\sum_{i=1}^{K} \frac{p_i E_i}{N_i(N_i-1)/2}}{\sum_{i=1}^{K} p_i} - \frac{\sum_{1\le i<j}^{K} \frac{p_{ij} E_{ij}}{N_i N_j}}{\sum_{1\le i<j}^{K} p_{ij}}$$

Où p_i représente le poids moyen des arêtes du cluster i et p_{ij} représente le poids moyen des arêtes inter clusters C_i et C_j.

Propriété 48. Pour un graphe non valué, les p_i et p_{ij} ont pour valeur 1 et on retrouve la formule de MQ.

5.2.3.3 Nouvelle mesure standardisée MQ pour un graphe valué :

En suivant la même technique, on peut modifier MQ* dans le cas d'un graphe valué :

$$MQ^* = \frac{\sum_{i=1}^{K} p_i E_i}{\sum_{i=1}^{K} N_i(N_i-1)/2} - \frac{\sum_{1\le i<j}^{K} p_{ij} E_{ij}}{\sum_{1\le i<j}^{K} N_i N_j}$$

Propriété 49. On obtient bien l'expression précédente de MQ* lorsque p_i et p_{ij} valent 1.

En notant p le poids moyen des arêtes, on peut normaliser MQ* en utilisant :

$$MQ^* = \frac{\sum_{i=1}^{K} \frac{p_i}{p} E_i}{\sum_{i=1}^{K} N_i(N_i-1)/2} - \frac{\sum_{1\le i<j}^{K} \frac{p_{ij}}{p} E_{ij}}{\sum_{1\le i<j}^{K} N_i N_j}$$

5.2.3.4 Nouvelle mesure de MQ entre 0 et 1

Les différentes expressions de MQ proposées ci-dessus sont obtenues comme différence d'un terme intra-cluster et d'un terme inter-clusters.

Par exemple, pour le calcul de l'expression standardisée de MQ (voir section 5.2.3.3) :

$$\text{intra} = \frac{\sum_{i=1}^{K} \frac{p_i}{p} E_i}{\sum_{i=1}^{K} N_i(N_i - 1)/2} \text{ et inter} = \frac{\sum_{1 \le i < j}^{K} \frac{p_{ij}}{p} E_{ij}}{\sum_{1 \le i < j}^{K} N_i N_j} \text{ et MQ} = \text{intra} - \text{inter}$$

Ainsi, MQ peut prendre toute valeur positive ou négative.

Nous proposons une autre variante de MQ :

MQ = intra / (intra + inter)

- MQ est compris entre 0 et 1.
- MQ est proche de 1 pour des clusters bien séparés (avec peu d'interactions).
- MQ est proche de 0 pour un graphe non clusterisé (clusters de taille 1).
- Plus il y a d'interactions entre clusters, plus MQ est faible.

Cette expression de MQ a le mérite d'être normalisée. Elle peut ainsi être comparée avec d'autres mesures effectuées sur d'autres graphes.

5.2.3.5 Expression de MQ positive

Nous avons remarqué en section 5.2.3.4 que l'expression initiale de MQ s'écrit comme différence d'une composante intra cluster et d'une composante inter clusters :

MQ = intra – inter

Le F de Fisher largement utilisé en statistique (Saporta 1990), est basé sur le rapport des composantes (appelées variances) inter et intra clusters.

On peut de même définir une variante de MQ par :

MQ = inter / intra

- MQ est une valeur positive.
- Si MQ est proche de 0, il y a peu d'interactions entre composantes.
- Pour un graphe très peu clusterisé, MQ est très grand.

Après avoir recensé les divers critères de qualité d'un partitionnement et introduit de nouveaux critères, nous proposons, en section suivante, une évaluation empirique de nos techniques de partitionnement.

5.3 Evaluation empirique : cas d'étude

Nous proposons une évaluation empirique des nouvelles techniques de partitionnement proposées à partir des graphes issus du réel présentés en section 1.3.

Pour tester la qualité du clustering nous analysons, a posteriori, le contenu de chaque silhouette. Les résultats sont donnés pour chaque graphe étudié (à l'exception du graphe d'interactions de protéines difficilement analysable par un non spécialiste…).

5.3.1 Graphes de CiteSeer

Partant d'une page focus, nous collectons automatiquement trois graphes de similarités d'articles sur le site bibliographique (CiteSeer). Le principe consiste à inclure de proche en proche dans le graphe les pages les plus similaires (related) au sens de CiteSeer.

5.3.1.1 Petit graphe

Nous choisissons comme page focus l'article de titre : "Navigation and Interaction within Graphical Bookmarks" (Hascoët 1999) et comme profondeur du graphe 5.

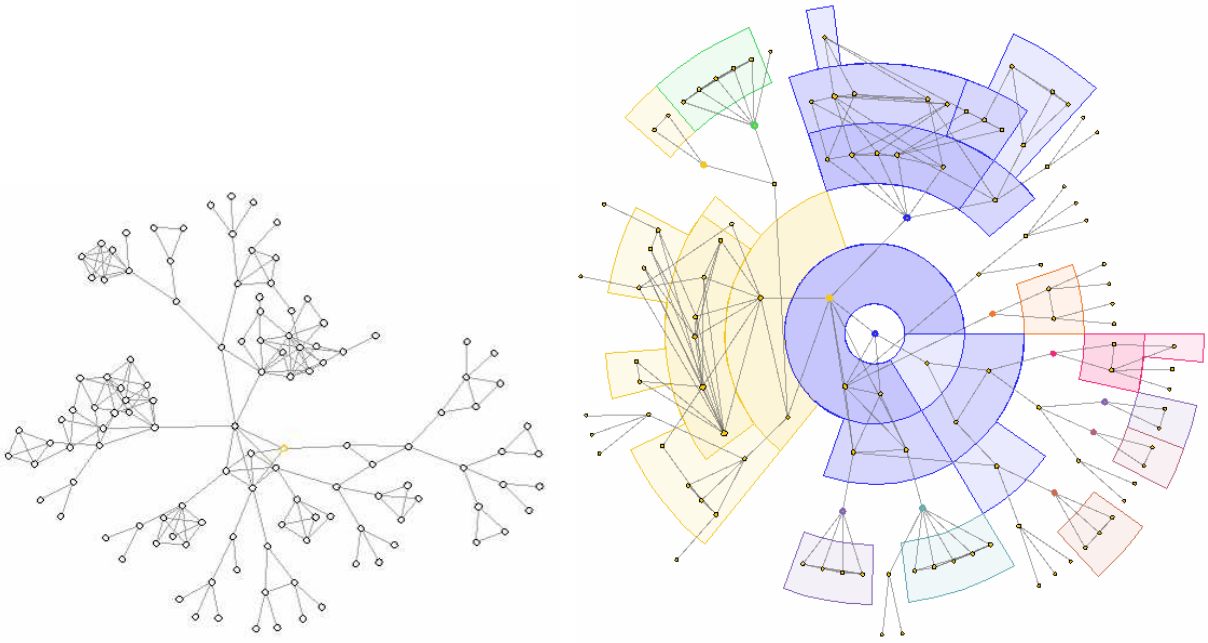


Figure 155 – petit graphe de CiteSeer

Ce graphe comprend 122 articles et 206 relations entre articles. C'est un graphe « arboré » (section 1.3.1.2) facilement représentable avec un algorithme utilisant un modèle de forces (Figure 155a). L'arbre de silhouettes centré sur le même focus en orange (Hascoët 1999) révèle la structure du graphe (Figure 155b). Des catégories sont associées a posteriori aux silhouettes (Figure 156). Elles reflètent bien l'organisation réelle des articles.

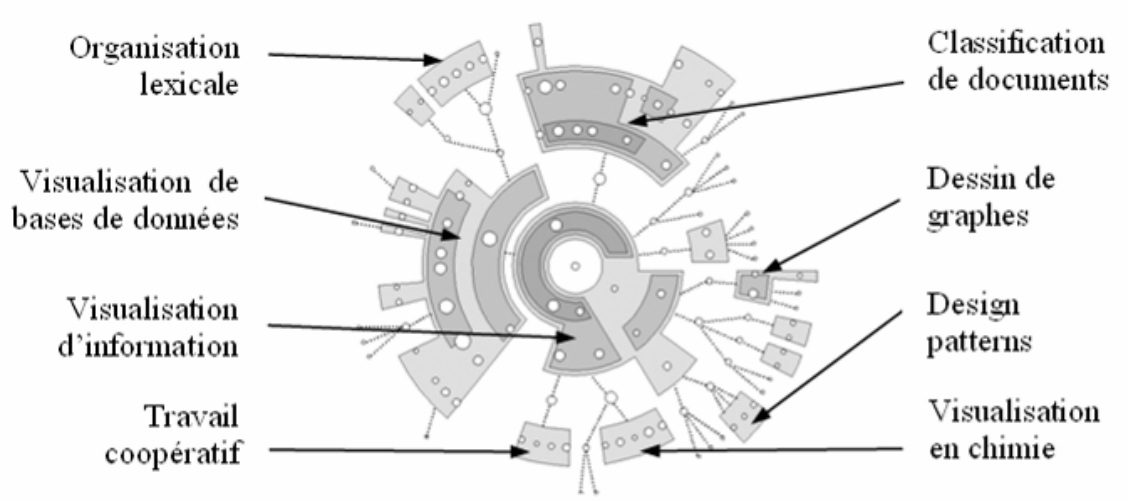


Figure 156 – reconnaissance a posteriori de catégories associées aux silhouettes

5.3.1.2 Gros graphe de CiteSeer

Considérons le gros graphe de Citeseer (section 1.3.1.3) à partir du focus : « Graph Visualisation in Information Visualisation: a Survey » (Herman, Mélançon et al. 2000) (nœud orange). Ce graphe est trop fortement connecté pour révéler des composantes intéressantes (Figure 157a). De même l'arbre de silhouettes résultant (Figure 157b) n'est pas satisfaisant.

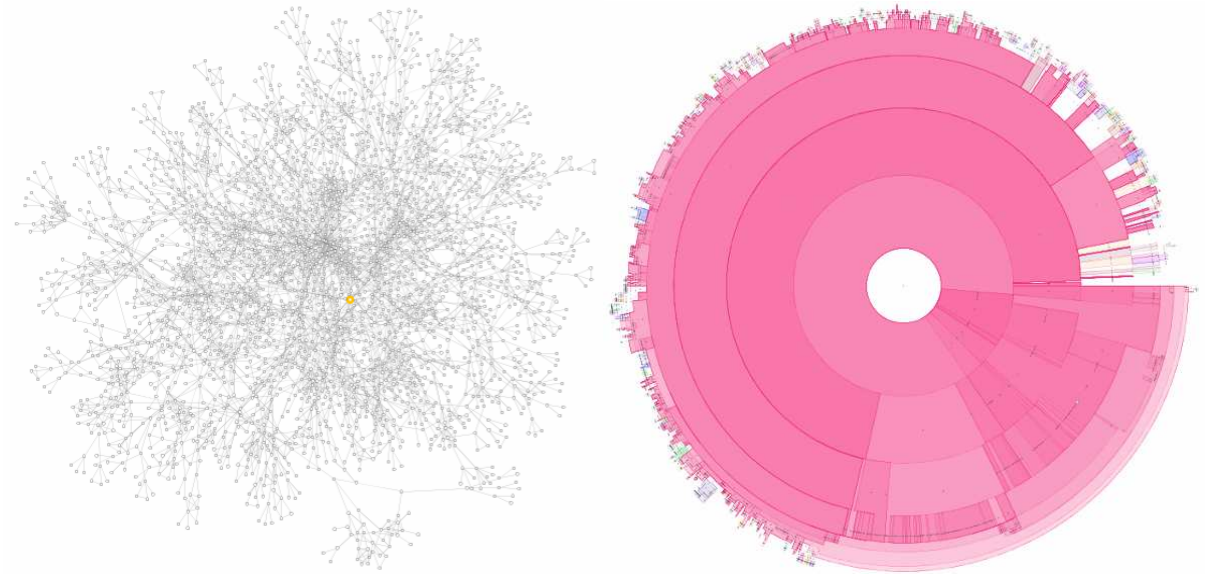


Figure 157 – gros graphe de CiteSeer non filtré

En supprimant les 321 arêtes de longueur supérieure à 4 dans l'arbre couvrant général, on obtient le graphe G_4 (Figure 158) : la visualisation par silhouettes révèle des composantes non visualisables avec un algorithme basé sur un modèle de forces.

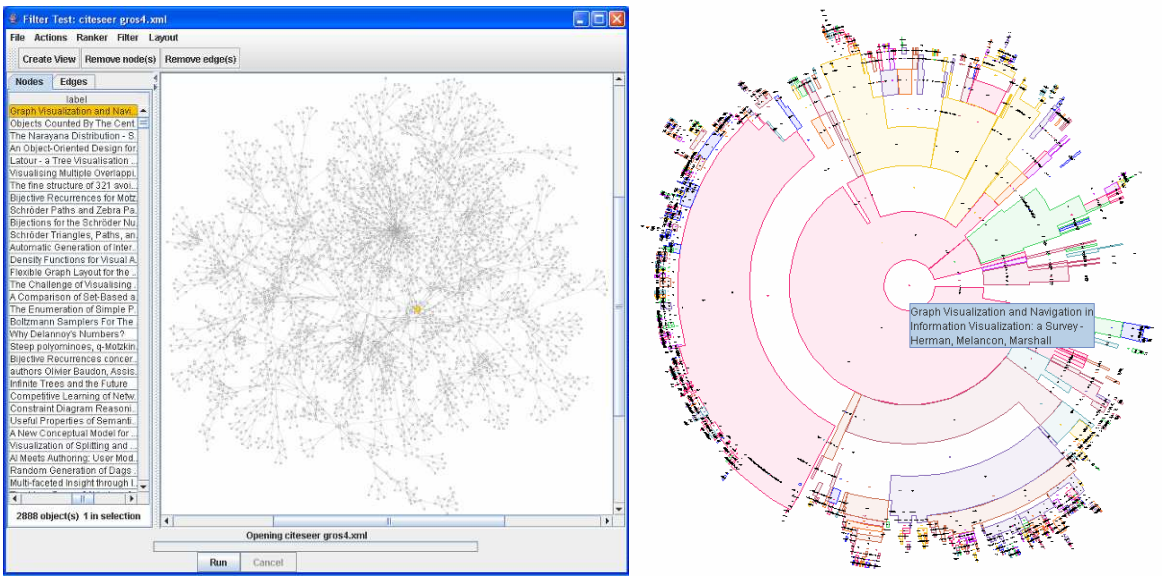


Figure 158 – gros graphe de CiteSeer filtré – G_4

Nous proposons une visualisation avec silhouettes emboîtées du graphe G_3 (488 suppressions d'arêtes). En terme de structure il est très proche du graphe G_4 (Figure 158). Cela montre une certaine robustesse de la technique de filtrage.

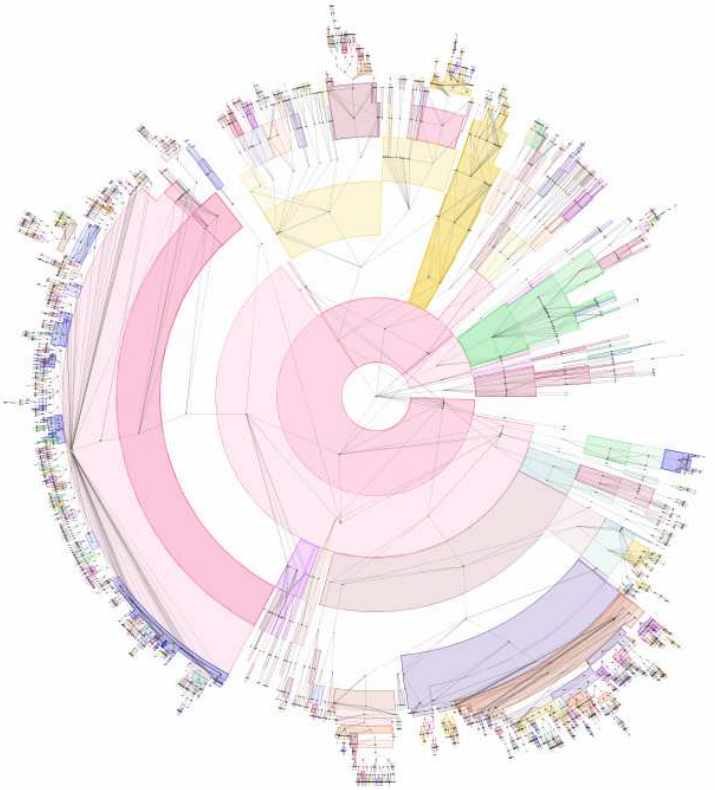


Figure 159 – gros graphe de CiteSeer filtré – G_3

Nous présentons une vue de G'_3 (Figure 160) construite à partir de l'arbre couvrant de focus « Graph Visualization and Navigation in Information Visualization: a Survey »

(Herman, Mélançon et al. 2000) (suppression de 573 arêtes de longueur supérieure à 3). La structure est voisine de celle de G_3 (Figure 159), même si la couleur des silhouettes change.

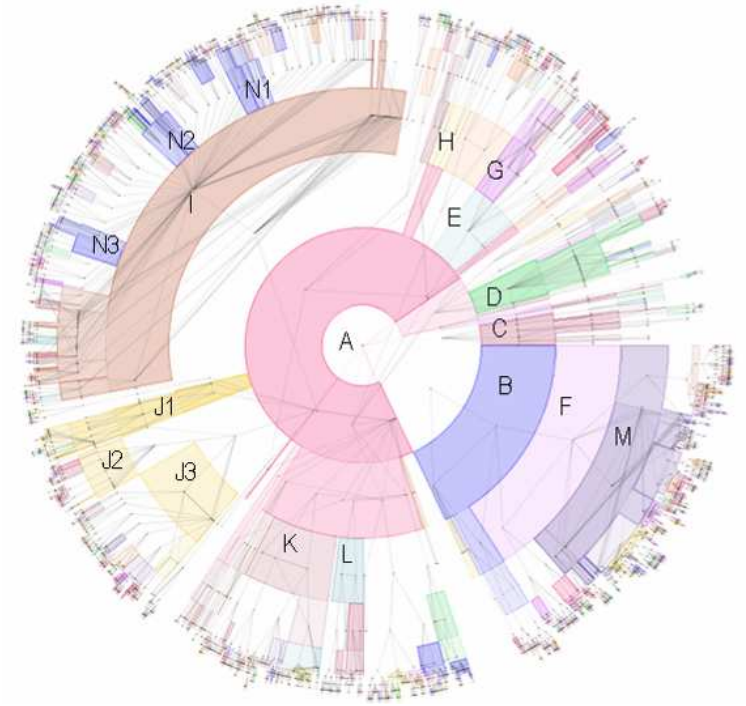


Figure 160 – gros graphe de CiteSeer filtré à partir du focus – G'$_3$

Les silhouettes dessinées (Figure 160) ont été étiquetées par leur nœud d'articulation :

- A : Graph Visualization and Navigation in Information Visualization: a Survey
- B : Tree visualisation and navigation clues for information visualisation
- C : Visualizing the Evolution of a Subject Domain: a Case Study
- D : Skeletal Animation for the Exploration of Graphs
- E : Visualising Multiple Overlapping Classification Hierarchies
- F : Objects Counted By The Central Delannoy Numbers
- G : Interactive Visualisation Techniques for Ontology Development
- H : a New Conceptual Model for Large-Scale Hypermedia
- I : a cognitive framework for describing & evaluating software exploration tools
- J : Flexible Graph Layout for the Web + Constraint Diagram Reasoning
- K : Random Generation of DAGs for Graph Drawing
- L : Visualization of Splitting and Merging Processes
- M : Generating Functions for Generating Trees
- N : Cognitive Support In Software Engineering Tools: a Distributed Framework

L'étiquetage présente l'avantage d'être facilement implémentable. Cependant, il ne permet pas de distinguer plusieurs silhouettes ayant même nœud d'articulation (exemple : N1, N2, N3 ou J1, J2, J3). D'autres techniques peuvent être utilisées basées sur l'analyse du contenu (fréquence d'apparition de termes dans le titre, le résumé ou les mots clés).

5.3.2 Graphe des relations entre étudiants

5.3.2.1 Graphe des amitiés

Nous présentons le graphe des 197 relations d'amitié (bilatérales) entre 56 étudiants :

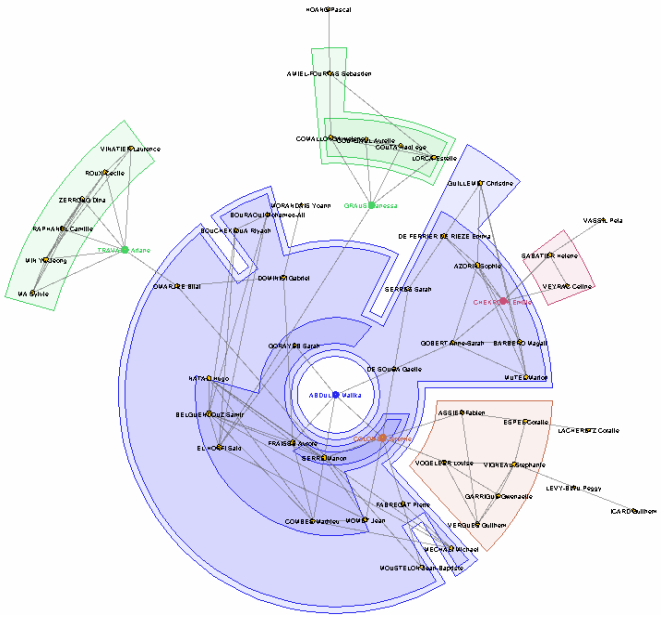


Figure 161 – graphe des relations d'amitié

Ce graphe présente une silhouette principale, et quatre silhouettes périphériques associées à de réels groupes d'amis au sein de la promotion d'étudiants (cette confirmation a été apportée par les étudiants après présentation de diverses vues).

5.3.2.2 Graphe des sympathies

Nous étudions le graphe des 464 sympathies (unilatérales) entre 56 étudiants. Ce graphe est trop dense pour que l'algorithme proposé détecte des composantes (Figure 162a). L'application du filtre G_2 permet de révéler des groupes de sympathies (Figure 162b) en ne conservant que 234 relations.

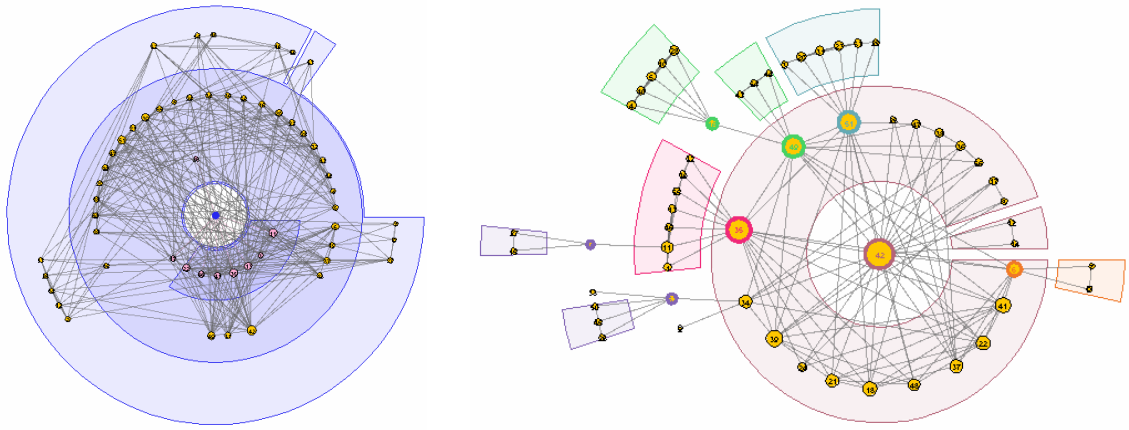


Figure 162 – graphe des sympathies avant et après filtrage G'_2

5.3.3 Graphe d'interactions de protéines

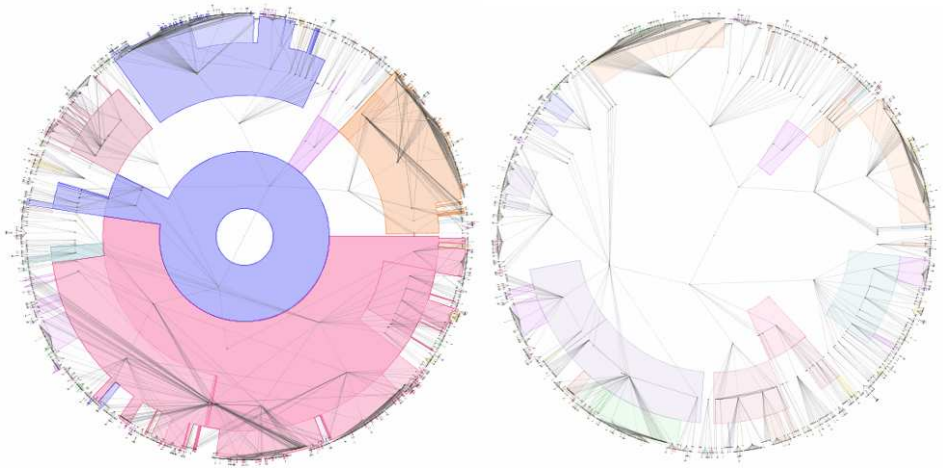


Figure 163 – graphes filtrés à partir du focus YBR236C (a) G'_3 (b) G'_2

Pour obtenir une visualisation intéressante, il faut procéder à un filtrage du graphe d'interactions de protéines. Nous proposons d'utiliser un filtrage basé sur le focus YBR236C. Nous présentons en Figure 163 les graphes filtrés G'_3 et G'_2. Ces graphes ont été obtenus en filtrant près de la moitié des arêtes. L'interprétation des résultats est donc toute relative. Notons quelques similarités entre les visualisations. Une fois encore la difficulté vient du fait que les couleurs des silhouettes ne peuvent être conservées. En effet suivant le niveau de filtre utilisé, la composition des silhouettes peut changer radicalement.

5.3.4 Graphe des conférences

Nous présentons le graphe des conférences en relation avec la conférence Interact 2005. Les données ont été collectées en utilisant itérativement les « related » de Google. On distingue une grande composante centrale sur la visualisation d'information et diverses composantes périphériques (Figure 164). Aucun filtre n'a été appliqué à ce graphe « arboré ».

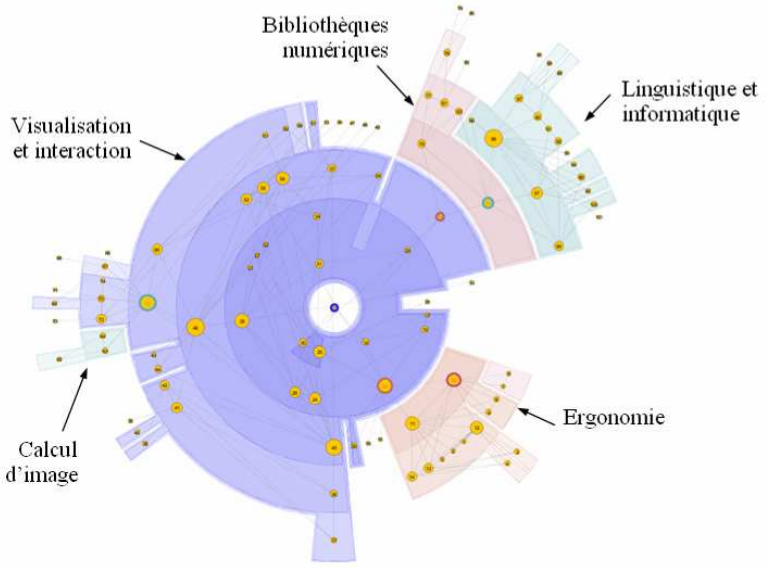


Figure 164 – graphe des conférences – focus : « Interact 2005 »

5.3.5 Graphe des citations d'InfoVis

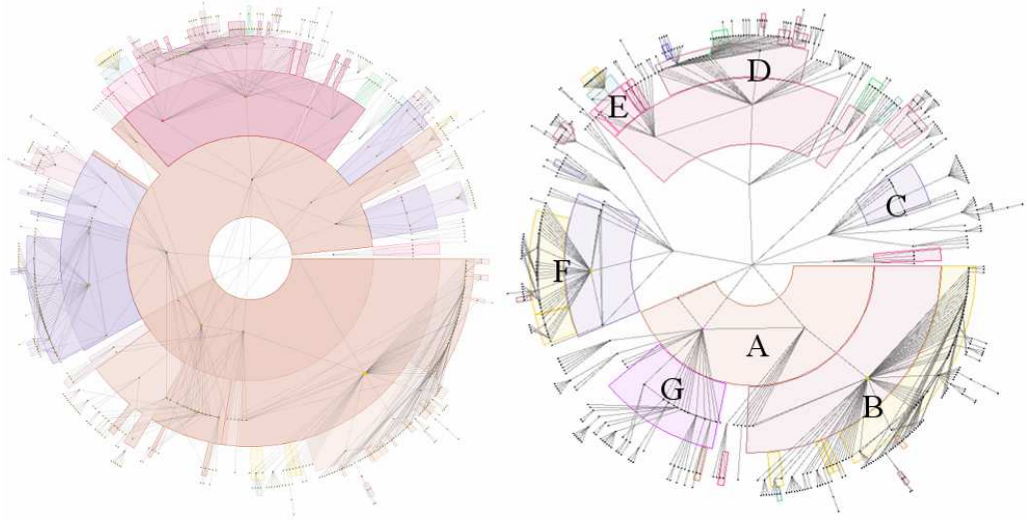


Figure 165 – graphe des citations d'InfoVis – focus : acm 618538 – (a) G'_3 (b) G'_2

Le graphe des citations d'InfoVis est très dense. Nous proposons un filtrage à partir du focus : « Exploring large graphs in 3D hyperbolic space, T. Munzner » (Munzner 1998). Les graphes G'_3 et G'_2 sont visualisés en Figure 165.

Les grandes zones (englobant une ou deux silhouettes) notées de A à F correspondent aux domaines suivants :

- A : espace hyperbolique
- B : navigation hiérarchique
- C : clustering de graphe
- D : taxonomie de la visualisation d'information
- E : animation interactive 3D
- F : fisheye généralisé
- G : fisheye graphique

5.3.6 Graphe du LIRMM

5.3.6.1 Graphe des co-auteurs du LIRMM

Nous présentons le filtre G'_3 sur le graphe des co-auteurs du LIRMM (Figure 166) puis le graphe des co-auteurs du laboratoire d'informatique (composante du LIRMM) (Figure 167a). Pour l'un et l'autre, nous prenons pour focus : Guy Mélançon. Les différentes silhouettes sont mises en évidence par des auteurs à fort degré.

L'arbre de silhouettes emboîtées présenté en Figure 166 est calculé en 1,5 secondes avec un Pentium II, 266 MHz, 52 Mo de RAM.

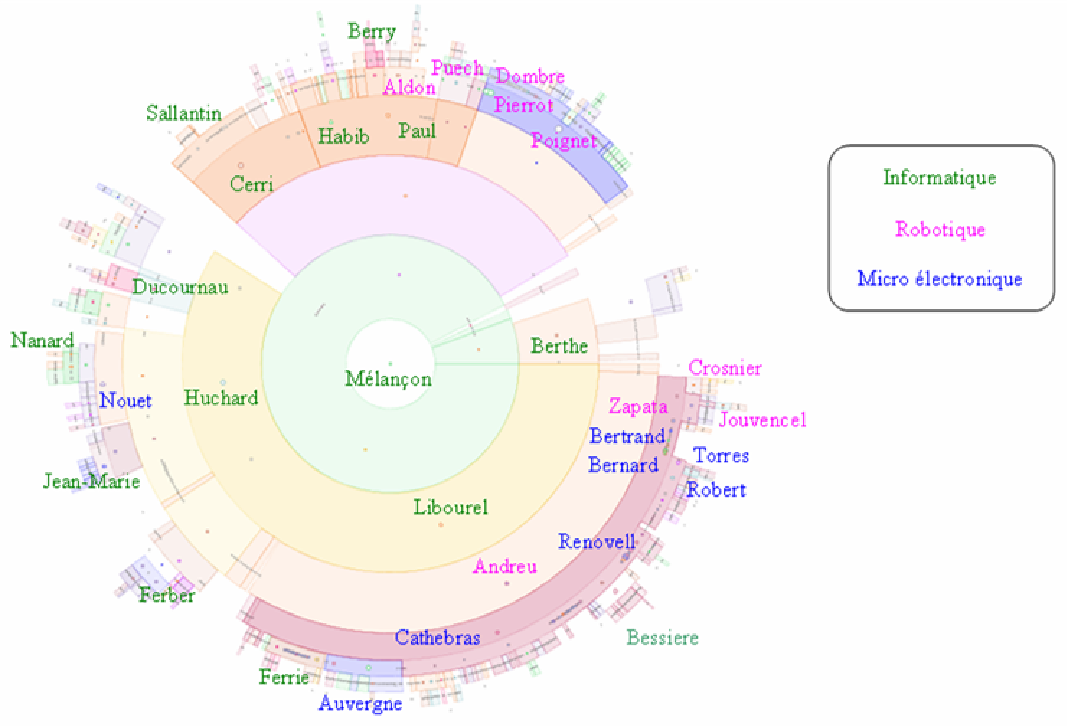


Figure 166 – graphe des co-auteurs du LIRMM – focus : Guy Mélançon – G'$_3$

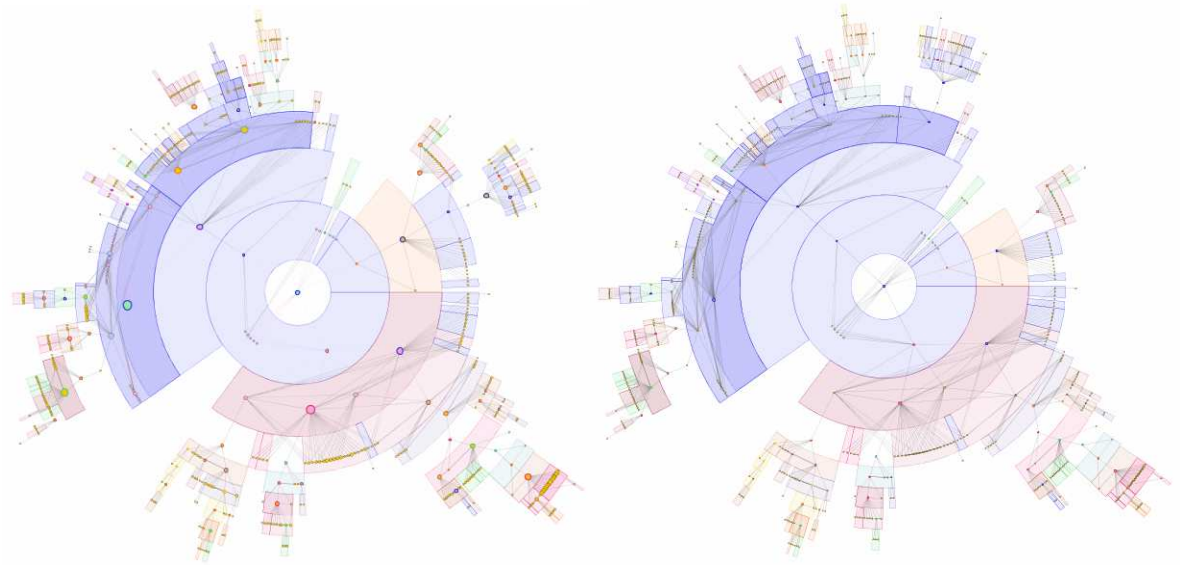


Figure 167 – co-auteurs du labo d'informatique filtré G'$_3$ basé sur (a) degré (b) BC

Nous présentons (Figure 167a et b) deux graphes filtrés G'$_3$ très proches. Le premier est obtenu avec l'arbre couvrant classique utilisant le degré. Le second utilise l'indice BC d'inter centralité (section 2.5.1). L'indice BC étant long à calculer (Brandes 2001), il est préférable d'utiliser le degré qui donne des résultats similaires.

5.3.6.2 Graphe des co-publications du LIRMM

Nous représentons en Figure 168 le graphe filtré G'$_2$ des co-publications du LIRMM centré sur l'article « Software Components Capture using Graph Clustering » (Chiricota, Jourdan et al. 2003).

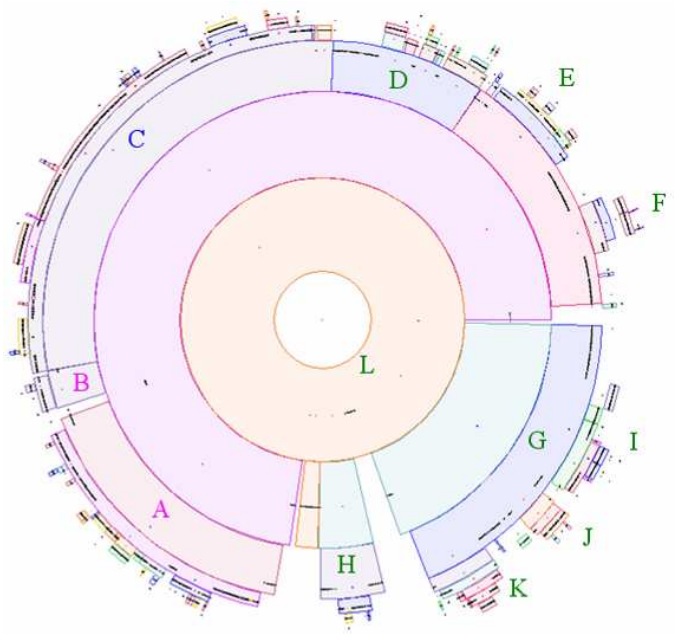


Figure 168 – arbre de silhouettes des co-publications du LIRMM – G'_2

Chaque silhouette est associée à une catégorie :

- A : robotique (1) (médical)
- B : robotique (2)
- C : micro électronique
- D: théorie des graphes
- E : apprentissage
- F : objet et agents
- G : bases de données
- H : arithmétique et combinatoire
- I : robotique (3)
- J : interface homme machine
- K : traitement algorithmique du langage
- L : dessin de graphe

5.3.7 Graphes simulés

Nous avons vu précédemment que les graphes du monde réel doivent souvent être préalablement filtrés avant d'appliquer le clustering. Nous proposons de générer des graphes « arborés » directement clusterisables sans filtrage préalable (voir modèle en section 1.2.5).

Pour créer de tels graphes, il suffit de partir d'un arbre et de lui ajouter des arêtes « courtes » (short ranges) entre ses nœuds. Différentes techniques peuvent être utilisées :

Exemple 92. Nous créons un arbre aléatoire à 1 000 nœuds : partant du noeud racine nous ajoutons des nœuds à l'arbre en liant tout nouveau nœud à un nœud existant. Nous créons ensuite 500 arêtes « courtes » liant un nœud source aléatoire et son nœud cible : à partir d'un nœud source, une marche aléatoire est simulée dans l'arbre. Elle a 50 % de chances de s'arrêter sur le nœud cible et 50 % de chances de continuer. Le graphe résultant est présenté en Figure 169.

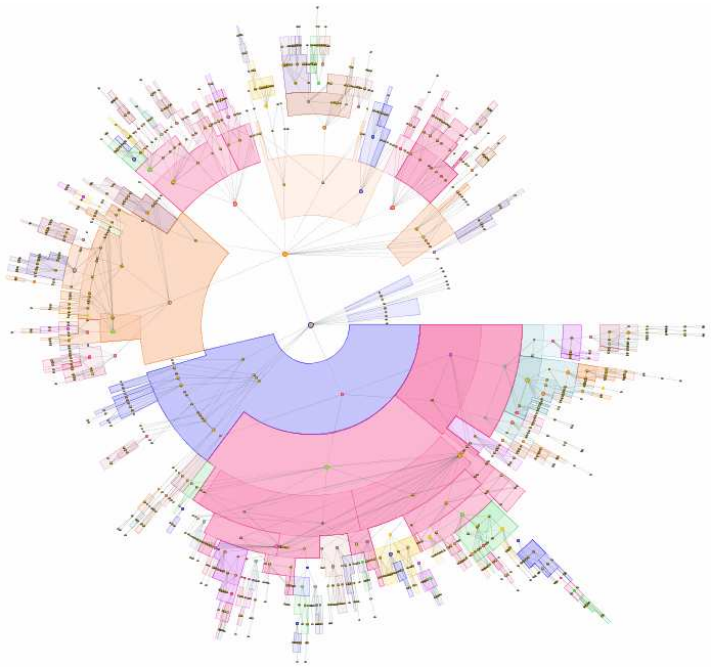


Figure 169 – graphe simulé : 1 000 nœuds, 1 500 arêtes

Exemple 93. Nous utilisons la même technique que précédemment (Exemple 92). La seule différence consiste à construire l'arbre initial en favorisant les liens partant de nœuds à fort degré. On parle alors d'attachement préférentiel. Le graphe résultant est présenté en Figure 170.

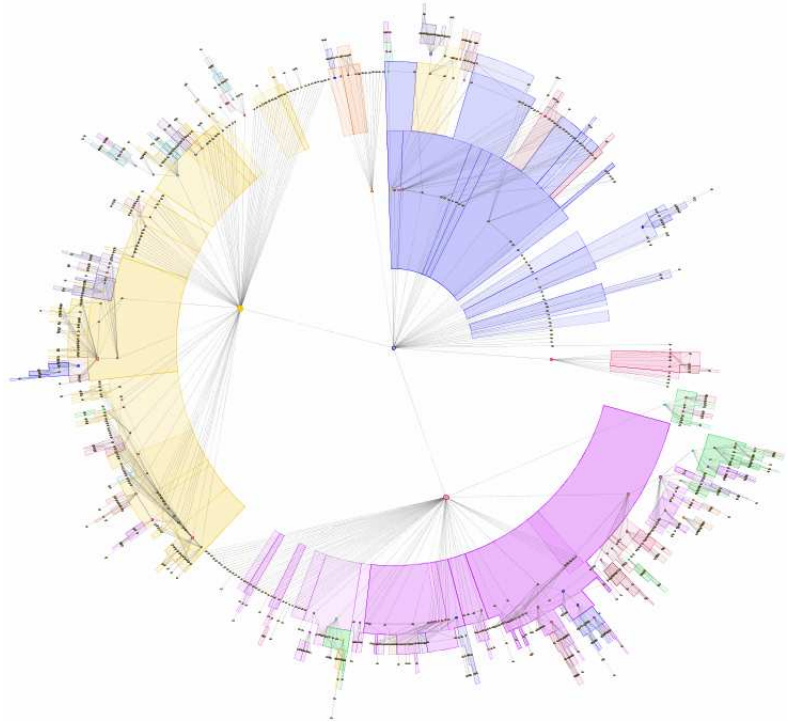


Figure 170 – graphe simulé avec attachement préférentiel : 1 000 nœuds, 1 500 arêtes

Exemple 94. A partir d'une arborescence aléatoire de 10 000 nœuds (construite en largeur à partir d'un focus), nous ajoutant 20 000 arêtes aléatoires. Pour cela nous considérons 20 000 nœuds aléatoires, chacun ayant une chance sur deux d'être lié à un nœud frère, une chance sur quatre d'être lié à un cousin, une

chance sur huit d'être lié à un petit cousin … à condition que de tels nœuds existent dans l'arborescence. L'arbre de silhouettes emboîtées résultant est présenté en Figure 171. Il est calculé et visualisé en 15 secondes avec un Pentium II, 266 MHz, 52 Mo de RAM.

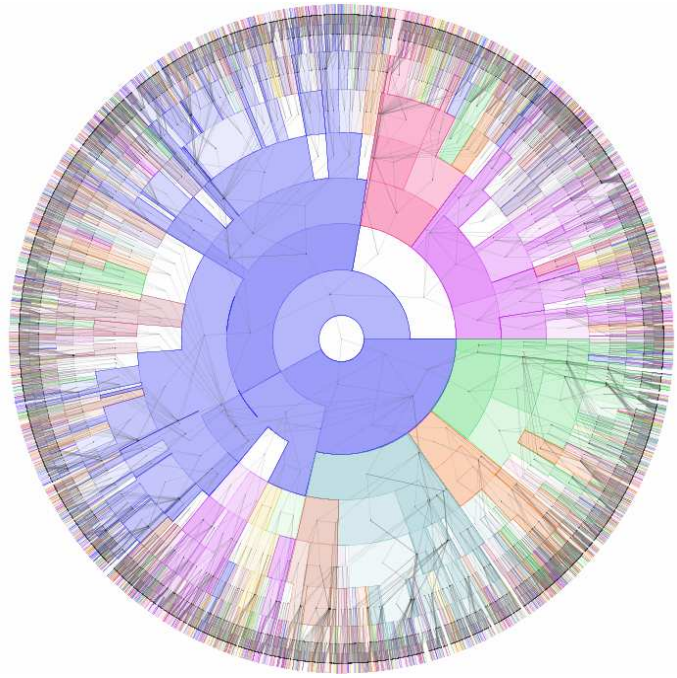


Figure 171 – graphe simulé : 10 000 nœuds, 30 000 arêtes

5.3.8 Résultats de l'évaluation empirique

Nous recensons les principaux résultats obtenus sur les graphes issus du monde réel :

- Les deux arbres de silhouettes de CiteSeer (Figure 156 et Figure 160) présentent des silhouettes représentant différents domaines de l'informatique.
- La pertinence du découpage de l'arbre de silhouettes en groupes d'amis (Figure 161) a été confirmée a posteriori par les étudiants interrogés.
- L'arbre de silhouettes des conférences liées à Interact 2005 (Figure 164) correspond à un découpage réel en terme de champs d'application.
- Le graphe des citations d'InfoVis (Figure 165) a été également organisé en grands domaines de la visualisation d'information.
- Les graphes des co-auteurs (Figure 166, Figure 167) et de co-publications du LIRMM (Figure 168) présentent un découpage naturel en domaines d'intérêt.
- Seul le graphe d'interactions de protéines (Figure 163) n'a pas été interprété par méconnaissance du sujet biologique.

Nous proposons dans les sections suvantes de nous appuyer sur les résultats empiriques obtenus pour définir des critères de qualité des nouvelles techniques et structures manipulées.

5.4 Evaluation analytique du nouveau filtrage

La qualité du filtrage dépend de la lisibilité du graphe « arboré » et de sa pertinence.

5.4.1 Lisibilité du graphe « arboré »

Les graphes « arborés » ont été introduits en section 1.2.5. Nous avons proposé une méthode de construction par filtrage (section 2.5) et un modèle de construction (section 5.3.7). Ces méthodes utilisent un arbre couvrant « maximal » et identifient des arêtes « courtes » entre nœuds proches. Cependant, la définition donnée en section 1.2.5 ne permet pas de s'assurer que l'on obtient bien un « beau » graphe « arboré ». Nous énonçons ce que nous attendons intuitivement d'un tel graphe :

Propriété 50. Un « beau » graphe « arboré » a l'allure d'un « bel » arbre interconnectant des clusters « bien » séparés.

L'utilisation d'un outil de dessin de graphes basé sur un modèle de forces permet d'identifier les « beaux » graphes « arborés ». En effet, avec un tel algorithme, les clusters sont naturellement séparés et organisés en une structure ayant « l'allure » d'une arborescence (Figure 22, Figure 28a, Figure 65, Figure 67). Toutefois, pour un gros graphe, suivant son placement initial, l'allure « arborée » peut être difficile à visualiser (Figure 158).

Comment identifier un « beau » graphe « arboré » sans visualisation graphique ?

Nous répondons à la question en trois étapes en reprenant les termes de la Propriété 50 :

5.4.1.1 Evaluation de la séparation des clusters

Pour déterminer si le graphe se décompose en clusters bien séparés, il faut d'abord les construire à l'aide d'une technique de partitionnement adaptée : ascendant hiérarchique (section 3.3.1), basé sur la suppression d'arêtes « faibles » (section 3.2.6), le calcul de flux (section 3.4.3), la construction de « silhouettes » emboîtées (section 3.6), ou une toute autre technique.

La séparation des clusters est alors estimée par un indice de séparabilité comparant la connectivité intra et inter clusters (section 5.1.3).

5.4.1.2 Estimation de « l'allure arborée » d'un graphe connexe

Les clusters doivent être non seulement bien séparés, mais appartenir de plus à une structure ayant « l'allure » d'un arbre de clusters.

Nous allons définir ce que l'on entend par « allure » d'un graphe « arboré ». Puis nous proposerons des critères pour qu'un graphe connexe quelconque ait une « allure arborée ».

- **« Allure » d'un graphe « arboré »**

Il est rare d'obtenir exactement un arbre interconnectant les clusters. Pour cela, il faut souvent supprimer certaines arêtes inter clusters.

Exemple 95. Le graphe des amis est un « beau » graphe « arboré » (Figure 172a). En supprimant uniquement trois arêtes (en jaune) on obtient un arbre qui interconnecte les clusters (Figure 172b). Ainsi, le graphe initial a « l'allure » d'un arbre de clusters.

Pour obtenir un arbre de clusters, on doit supprimer les arêtes n'appartenant ni à l'arbre couvrant ni à aucun cluster.

Définition 88. Le taux d'arêtes supprimées définit « l'allure » d'un graphe « arboré ».

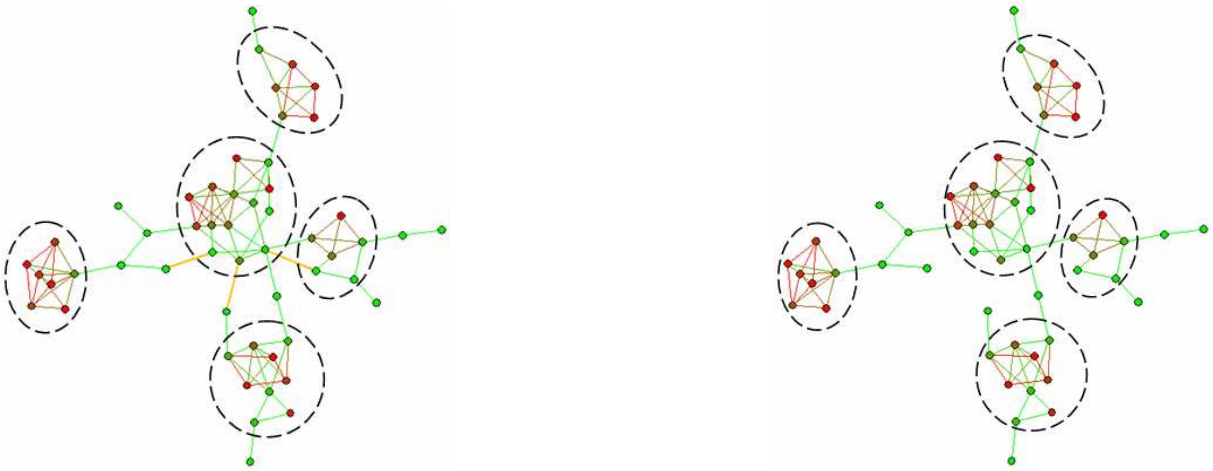


Figure 172 – graphe « arboré » des amis (a) clustering (b) filtrage d'arêtes inter clusters

Lorsqu'on considère un graphe connexe avant filtrage, il peut apparaître une arête « longue » qui « court circuite » le graphe (voir l'arête en pointillés en Figure 174a). Une telle arête risque davantage de détériorer « l'allure arborée » du graphe qu'une arête « courte » (en jaune, Figure 174a).

Pour estimer « l'allure arborée » d'un graphe connexe quelconque (pas nécessairement « arboré »), on va déterminer « l'allure » du graphe quotient.

- **« Allure arborée » d'un graphe connexe quelconque**

Pour déterminer « l'allure arborée » d'un graphe connexe quelconque nous construisons le graphe quotient constitué des clusters et meta-arêtes entre clusters. Notons que dans le cas d'un graphe « arboré », le graphe quotient obtenu est un arbre (voir l'arbre quotient et le graphe « arboré » simplifié présenté en Figure 173).

Nous définissons la « taille » des meta-arêtes du graphe quotient avec la technique introduite en section 2.5.1 (basée sur l'extraction d'un arbre couvrant maximal).

Remarque 8. Les arêtes « longues » (associées à une meta-arête de taille supérieure à un) détériorent « l'allure arborée » du graphe initial.

Exemple 96. L'ajout d'une arête « longue » au graphe des amis (pointillés, Figure 174a), produit le graphe quotient en Figure 174b. La meta-arête a pour taille 2.

Figure 173 – graphe des amis (a) graphe « arboré » simplifié (b) arbre quotient

Définition 89. « L'allure » arborée d'un graphe connexe quelconque dépend :

- du « poids » des clusters (à maximiser)
- du poids des meta-arêtes de taille 1 (celles de l'arbre couvrant du graphe quotient)
- du poids des meta-arêtes de taille supérieure à 1 (à minimiser)

Notons que le « poids » des clusters et meta-arêtes peut être défini en utilisant l'un des indices de compacité et séparabilité recensés en section 5.1.

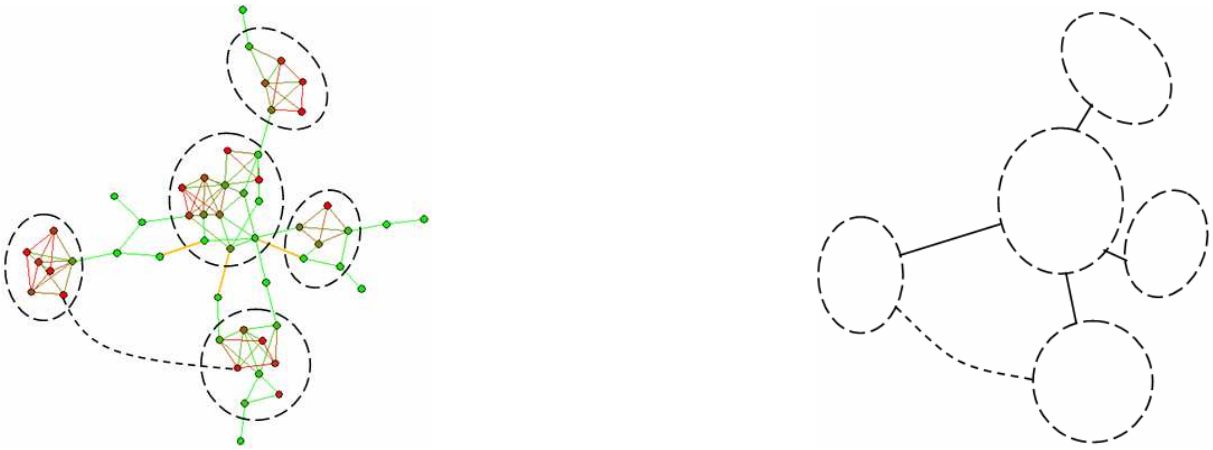


Figure 174 – graphe des amis (a) ajout d'une arête « longue » (b) graphe quotient

5.4.1.3 Caractérisation d'un « bel » arbre de clusters

Nombreux travaux se sont penchés sur la « beauté » des arbres (Battista et Eades 1994; Davidson et Harel 1996; Battista, Eades et al. 1999).

Nous avons postulé (Propriété 50) qu'un « beau » graphe « arboré » a l'allure d'un « bel » arbre de clusters. Qu'entend-on par « bel » arbre ?

- L'arborescence ne doit pas être triviale c'est-à-dire réduite à un cluster (c'est souvent le cas quand le seuil de filtrage est trop élevé).
- L'arborescence doit être homogène relativement à la taille des branches, le diamètre des nœuds-clusters, et le degré des nœuds-clusters de l'arbre.

Exemple 97. En Figure 172a, le graphe présente 5 clusters homogènes et bien organisés.

5.4.2 Pertinence du graphe « arboré »

5.4.2.1 Optimisation du seuil de filtrage choisi

Nous avons vu en section 5.4.1 que la présence d'arêtes « longues » nuit à la lisibilité du graphe « arboré ». Il est ainsi naturel de vouloir appliquer un filtrage de seuil assez bas. L'inconvénient est qu'en filtrant trop d'arêtes on perd les caractéristiques du graphe initial.

Le problème est donc de définir un seuil optimal c'est-à-dire assez faible pour obtenir un « beau » graphe « arboré », mais pas trop faible de façon à supprimer le minimum d'arêtes.

Nous étudions le pourcentage d'arêtes supprimées en fonction du seuil choisi pour trois graphes réels introduits précédemment.

Pour le graphe de CiteSeer de taille moyenne (Figure 175a) et le graphe de co-auteurs du LIRMM (Figure 175b), moins de 10 % des arêtes présentent une taille supérieure à trois (seuil de filtrage) contre 42 % pour le graphe YEAST (Figure 175c). L'interprétation du filtrage du graphe YEAST au seuil 3 est par conséquent beaucoup plus délicate.

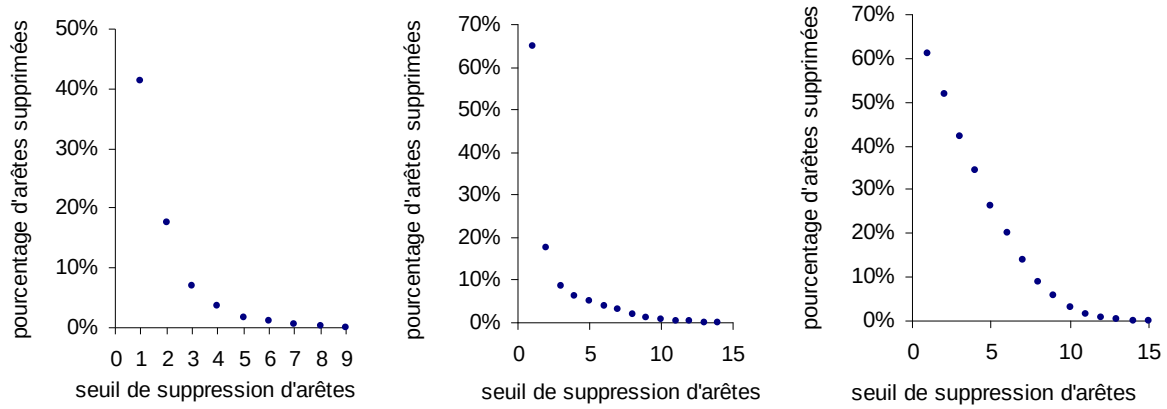


Figure 175 – seuil de filtrage – (a) CiteSeer (b) co-auteurs du LIRMM (c) YEAST

5.4.2.2 Suppression d'arêtes non « pertinentes »

Il faut non seulement supprimer peu d'arêtes, mais les arêtes non « pertinentes ».

Par construction du graphe « arboré », les arêtes supprimées se trouvent pour la plupart éloignées du focus. Elles peuvent ainsi être considérées comme non « pertinentes ». En effet, leur suppression ne modifie pas le graphe localement autour du focus. Remarquons que la « pertinence » d'une arête dépend également du choix du focus de filtrage.

5.4.2.3 Conservation des propriétés du graphe

Le filtrage utilisé augmente le diamètre du graphe (pour une meilleure lisibilité) tout en conservant ses principales propriétés :

Propriété 51. Un graphe « sans échelle » le reste après filtrage (Kim, Noh et al. 2004).

Cette propriété s'explique par le mode de construction de l'arbre couvrant favorisant les relations entre nœuds à fort degré (section 2.3.2.2).

Propriété 52. Après filtrage des arêtes « longues », le coefficient de clustering augmente. Ainsi un graphe « petit monde » reste « petit monde ».

Exemple 98. Le coefficient de clustering du graphe de CiteSeer (Figure 157) vaut initialement 0,46 puis 0,51 après filtrage (Figure 159 et Figure 160). Pour le graphe d'InfoVis il vaut initialement 0,21 puis augmente à 0,33 avec G'_4 (Figure 165a) et 0,41 avec G'_3 (Figure 165b).

5.5 Evaluation analytique du nouveau partitionnement

La qualité d'un partitionnement dépend de la bonne lisibilité de l'arbre de silhouettes emboîtées et de sa pertinence.

5.5.1 Lisibilité de l'arbre de silhouettes emboîtées

L'arbre de silhouettes emboîtées est une structure multi-échelles facilitant l'organisation contextuelle du graphe à partir d'un focus. Cette structure doit être facilement visualisable. De ce fait, elle doit présenter diverses caractéristiques de lisibilité que nous allons étudier.

5.5.1.1 Structure non triviale

L'arbre doit présenter plusieurs silhouettes englobantes (et nœuds d'articulation).

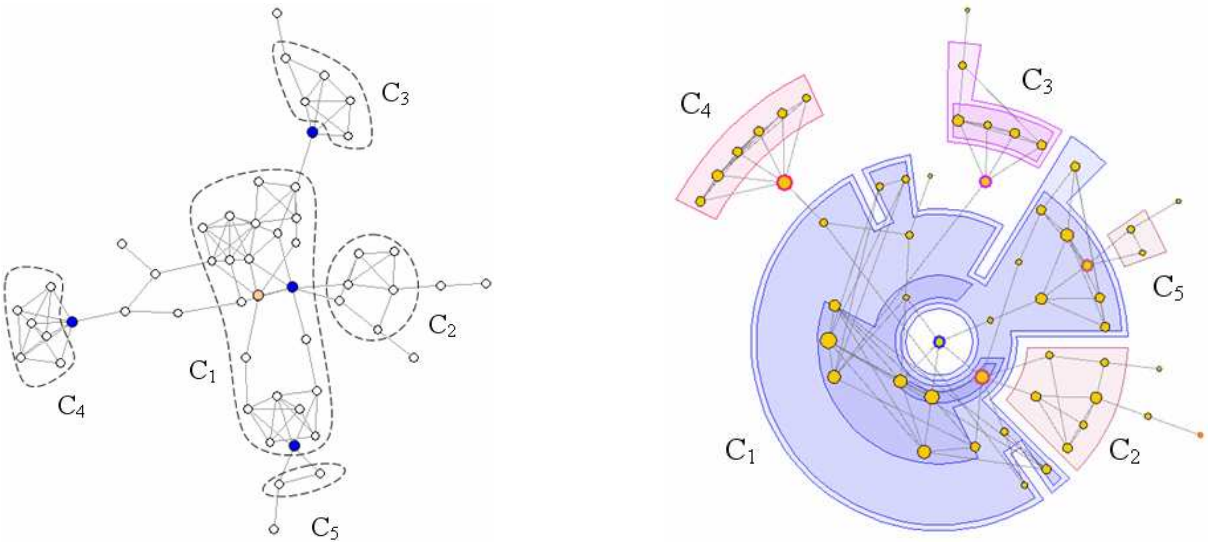


Figure 176 – graphes des amis (a) identification des silhouettes (b) vue radiale

Le graphe des amis (Figure 176) présente 4 nœuds d'articulation et 5 silhouettes. Tandis que le graphe des sympathies (Figure 162) et le gros graphe de CiteSeer (Figure 157b) qui sont non « arborés » ne présentent qu'une seule grosse silhouette.

5.5.1.2 Lisibilité des silhouettes emboîtées

Pour éviter de surcharger les vues d'arbre de silhouettes emboîtées, nous proposons de ne visualiser que les silhouettes non triviales c'est-à-dire contenant au moins deux nœuds.

En Figure 176b, par exemple, la silhouette C_1 contient une sous-silhouette non triviale constituée de 9 nœuds biconnectés à une distance 2 du focus. Cette sous-silhouette contient elle-même une silhouette constituée de 3 nœuds biconnectés à une distance 1 du focus.

L'utilisation de couleurs transparentes permet de refléter le niveau de connectivité d'une silhouette et de ses sous-silhouettes emboîtées (couleur plus ou moins foncée).

5.5.1.3 Découpage « ganté » des silhouettes

Chaque silhouette doit être organisée en un « bel » arbre de clusters favorisant ainsi un « beau » découpage « ganté » de la silhouette. Ce découpage « ganté » apparaît clairement lorsqu'on utilise une vue radiale comme celle introduite en section 4.3.

Dans le graphe des amis, la construction des silhouettes ne permet pas de scinder C_1 (contrairement au partitionnement proposé en Figure 172). Cette silhouette présente toutefois un « découpage ganté » (Figure 176b) mettant en valeur les deux parties de la silhouette C_1.

Les arbres de silhouettes emboîtées obtenus à partir de « beaux » graphes « arborés » présentent généralement les caractéristiques décrites dans cette section : ils sont ainsi qualifiés de « beaux » arbres de silhouettes emboîtées.

5.5.2 Pertinence de l'arbre de silhouettes emboîtées

Il ne suffit pas que l'arbre de silhouettes emboîtées soit « beau », il faut aussi qu'il soit « pertinent ». Pour cela il doit vérifier deux principaux critères étudiés dans cette section.

5.5.2.1 Séparation optimale des composantes

La connectivité inter silhouettes est minimisée (selon les critères définis section 5.1.3). En effet, les silhouettes sont construites à partir de composantes biconnexes. Ainsi, seuls les nœuds d'articulation assurent la connexion entre silhouettes.

Pour déterminer un indice de séparabilité des silhouettes on pourra utiliser la technique (exposée en section 5.4.1.2) relative à « l'allure » d'un graphe « arboré ». Si on note N le nombre de silhouettes, M le nombre d'arêtes du graphe et K le nombre d'arêtes inter-silhouettes on peut définir la séparation des silhouettes par le taux d'arêtes à supprimer :

$$\frac{K - N + 1}{M}$$

Chaque nœud doit être plus fortement lié à son cluster d'affectation qu'aux autres. Pour le tester, il est possible d'utiliser un indice de clustering local défini en section 5.1.4.

Cette propriété est, par définition, vérifiée pour tous les nœuds du graphe à l'exception de certains nœuds d'articulation. En effet, ces derniers (appartenant au moins à 2 composantes biconnexes) sont arbitrairement affectés à la première silhouette rencontrée à partir du focus. Ceci simplifie le dessin des silhouettes et permet une homogénéité de lecture de la vue.

Il reste toutefois possible d'affecter un nœud d'articulation à sa composante la plus proche (afin de diminuer l'interconnectivité) en utilisant les critères définis en section 5.1.4.

5.5.2.2 Qualité sémantique des silhouettes

Les silhouettes étant construites à partir de composantes biconnexes, elles constituent un ensemble de nœuds interconnectés tel qu'il existe toujours un cycle passant par deux nœuds quelconques. Ainsi ses nœuds ont peu de risques d'être connectés par hasard (d'autant moins que la technique de filtrage en graphe « arboré » ne conserve que les cycles « courts »).

Ainsi les silhouettes contiennent des nœuds fortement liés. Par ailleurs, elles ne sont interconnectées que par des nœuds d'articulation.

Il paraît ainsi naturel de considérer les silhouettes emboîtées comme des unités de sens : il s'agit de domaines (et sous domaines emboîtés) bien séparés et organisés en arbre.

Nous ne pouvons pas évaluer de façon théorique la qualité sémantique des silhouettes. Toutefois, l'étude du contenu des silhouettes menée sur des réseaux d'interactions réels en section 5.3 montre la pertinence de la construction. A condition de choisir un seuil de filtrage adapté, les réseaux s'organisent naturellement en arbres de silhouettes ayant un sens.

Le changement de focus de filtrage permet éventuellement d'identifier des silhouettes associées à d'autres significations. C'est tout l'intérêt d'utiliser des techniques contextuelles.

La qualité sémantique de l'arbre de silhouettes emboîtées dépend beaucoup de la qualité du graphe « arboré » utilisé (voir section 5.4.2).

Il n'y a pas de « bon » partitionnement en arbre de silhouettes emboîtées sans « bon » filtrage préalable en graphe « arboré ». Ces deux structures sont en effet intimement liées. Nous avons défini des critères permettant de les évaluer.

Cette analyse complète l'étude empirique menée en section 5.3 : moyennant l'utilisation d'un filtrage adapté, les structures obtenues sont non seulement facilement visualisables et navigables, mais elles sont également pertinentes et riches de sens.

« Ce n'est pas le difficile, c'est le beau que je cherche. »

Fénelon

Chapitre 6 Conclusion et annexes

6.1 Conclusion générale

Diverses techniques de filtrage et de partitionnement de graphe ont été développées, mais peu s'appliquent aux graphes « sans échelle » à comportement « petit monde ». Il s'agit pourtant de propriétés caractéristiques des principaux graphes d'interactions du monde réel.

La propriété « sans échelle » vient du fait que ces graphes d'interactions sont souvent dans un processus de croissance. Les nouveaux nœuds établissent, de préférence, des liens avec des nœuds à fort degré (attachement préférentiel).

La propriété « petit monde » s'interprète par un faible diamètre du graphe et l'adage : « les amis de mes amis sont mes amis » que l'on retrouve souvent dans les graphes sociaux.

Ces graphes présentent généralement un noyau très dense. Aussi, les algorithmes de filtrage et de partitionnement classiques parviennent rarement à extraire structure et motifs intéressants. De plus, les outils de dessin de graphes sont souvent peu performants pour ce type de graphe.

Nous avons introduit, dans cette thèse, de nouvelles techniques permettant de filtrer, partitionner et visualiser des réseaux d'interactions « petit monde sans échelle » tout en conservant leurs principales propriétés.

Les principales caractéristiques de ces techniques sont résumées ci-dessous :

Approche contextuelle reposant sur la donnée d'un focus utilisateur

Le fil conducteur de ces techniques est d'utiliser un point de vue contextuel : le graphe est initialement filtré à partir d'un « focus de filtrage » en un graphe « arboré ». Puis, il est organisé autour d'un « focus de partitionnement » en un arbre de silhouettes emboîtées. Enfin la technique de visualisation radiale permet une exploration contextuelle de la structure.

Construction et manipulation de structures multi-échelles

Les techniques de partitionnement et visualisation proposées permettent de construire et représenter des structures de données multi-échelles à la fois simples (basées sur la structure d'arbre) et riches : dans un arbre de silhouettes emboîtées, chaque silhouette englobante est un arbre d'inclusion de silhouettes. La structure résultante est une superposition d'arbres de silhouettes. Par ailleurs, chaque silhouette se décompose en un arbre d'adjacence de « clusters » donnant la forme « gantée » de la silhouette.

Changement de perspective facilité

La conservation de composantes invariantes lors d'un changement de focus permet de naviguer dans l'arbre de silhouettes emboîtées sans être perdu. En effet, les silhouettes sont déplacées, éventuellement déformées mais gardent leur contenu (à l'exception éventuelle des nœuds d'articulation). Par ailleurs, l'utilisation de techniques d'animation de nœuds facilite la compréhension de la vue lors du changement de perspective (Heer, Card et al. 2005).

Un partitionnement qui a du sens

L'analyse a posteriori du contenu des silhouettes révèle que le partitionnement a un sens. Suivant la perspective considérée, les nœuds peuvent être organisés différemment. Cela assure une organisation contextuelle du graphe.

Faible complexité des algorithmes

Les techniques de filtrage partitionnement et visualisation de graphes d'interactions que nous avons introduites ont une faible complexité. Elles peuvent ainsi être appliquées en temps réel favorisant les interactions avec l'utilisateur.

Interdépendance des techniques

Les techniques de partitionnement et de visualisation proposées ne donnent des résultats intéressants qu'avec de « beaux » graphes « arborés ». Ainsi, il est nécessaire d'utiliser un pré filtrage, à moins que le graphe ne soit naturellement « arboré ».

Perspectives :

Autour des graphes « arborés »

Nous avons défini la structure de graphe « arboré », puis proposé deux techniques pour extraire une telle structure d'un réseau d'interactions réel. Il peut être souhaitable de rechercher des variantes de cette structure, mais aussi de nouvelles techniques d'extraction adaptées.

Construction d'autres partitionnements basés sur un graphe « arboré »

Cette thèse décrit une technique de partitionnement de graphe « arboré » basée sur un focus utilisateur. Toutefois, d'autres techniques pourraient être utilisées. Il serait intéressant d'étudier ces techniques et de les comparer avec celle proposée.

Techniques avancées de visualisation et d'interaction d'arbre de silhouettes

Nous avons proposé une visualisation utilisant des zones emboîtées. Nous pourrions intégrer de nouvelles techniques d'interaction et étudier d'autres visualisations adaptées.

6.2 Glossaire des termes nouveaux

Nous proposons dans cette section une brève description des principales expressions et structures introduites dans ce document. Nous considérons G un graphe connexe donné :

- **Graphe « arboré »** : graphe possédant un arbre couvrant « maximal » tel que :
  - La longueur moyenne des arêtes dans l'arbre soit petite devant le diamètre de l'arbre.
  - Et la longueur des arêtes soit presque sûrement inférieure au diamètre de l'arbre.

- **Filtrage de contour** : technique permettant d'obtenir un graphe « arboré » à partir de G :
  - **G_λ :** filtre des arêtes de longueur supérieure à λ en utilisant un arbre couvrant ayant pour racine le nœud de plus fort degré.
  - **G'_λ :** même filtre que G_λ utilisant un arbre couvrant construit à partir d'un focus.

Le filtrage de contour G_4 du graphe du LIRMM est présenté Figure 177 (voir section 2.5).

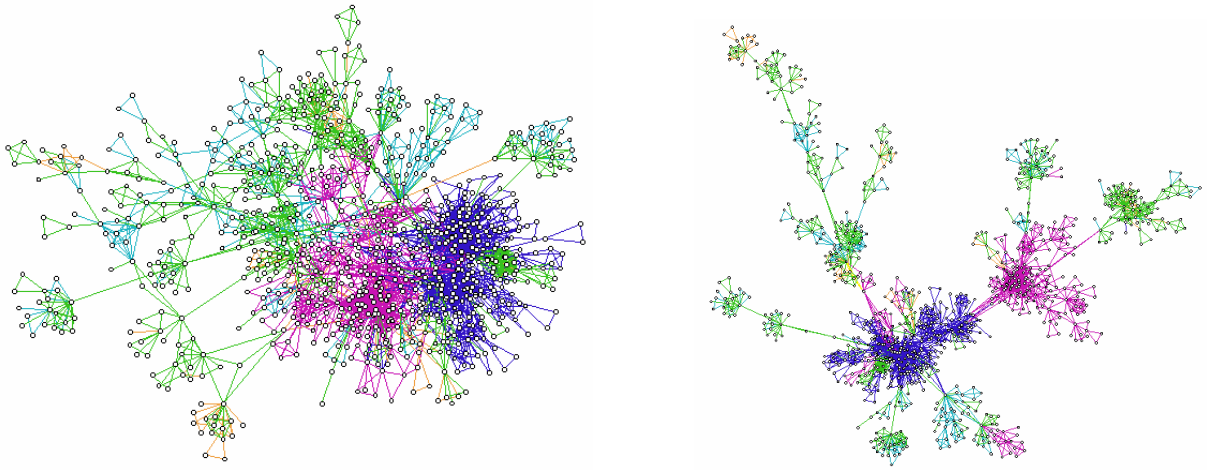


Figure 177 – filtrage de contour (a) graphe G (b) graphe « arboré » – filtre G_4

- **Clustering de contour** : organisation contextuelle des nœuds en structures de partitionnement multi-échelles :
  - **Arbre composé :** graphe composé dont les clusters englobants forment un arbre.
  - **Arbre composé multi niveaux :** graphe composé dont chaque niveau de clusters est organisé en arbre.

Nous présentons (Figure 178) ces deux structures (voir section 3.1.2)

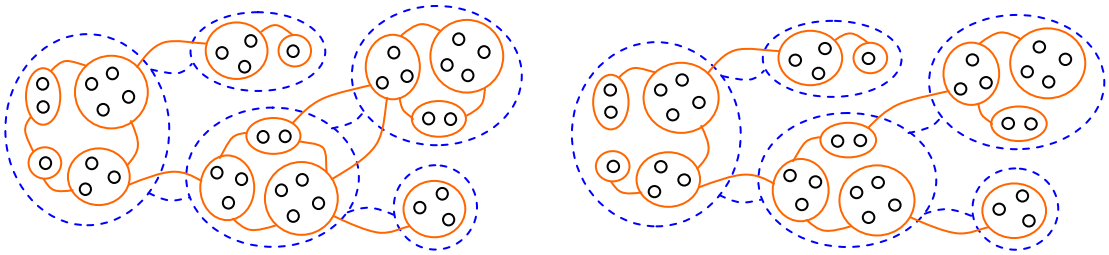


Figure 178 – (a) arbre composé (b) arbre composé multi niveaux

- **Visualisation de contour** : technique produisant une vue clusterisée d'un graphe arboré :
  - **Arbre de clusters** : chaque cluster est constitué de nœuds à même distance du focus.
  - **Arbre de silhouettes** : chaque silhouette contient un arbre de clusters.
  - **Arbre de clusters emboîtés** : arbre composé multi niveaux dont chaque cluster englobant est à l'origine d'un arbre d'inclusion de clusters.
  - **Arbre de silhouettes emboîtées** : arbre composé multi niveaux dont chaque silhouette englobante est à l'origine d'un arbre d'inclusion de silhouettes.

Nous présentons Figure 179 et Figure 180 les diverses vues (sections 3.6 et 4.3) :

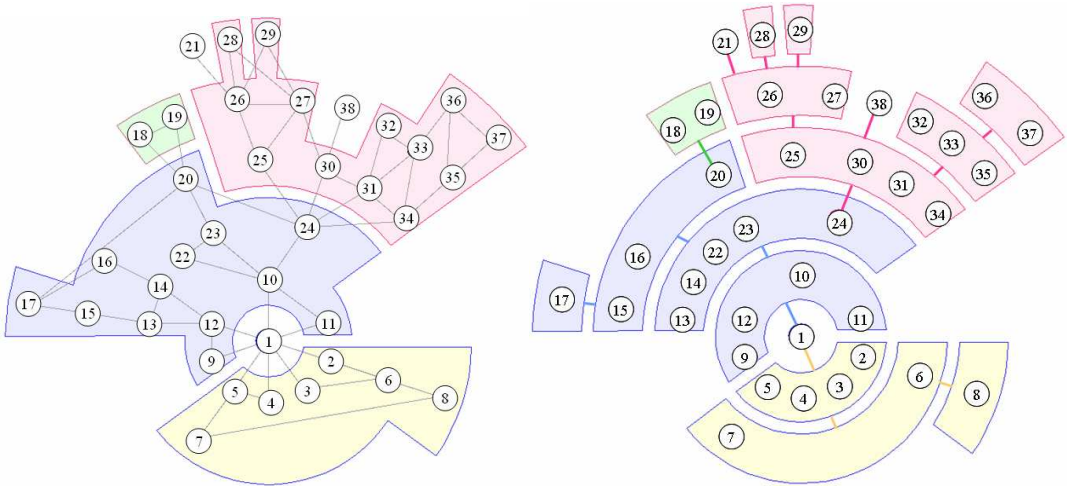


Figure 179 – (a) arbre de silhouettes (b) arbre de clusters

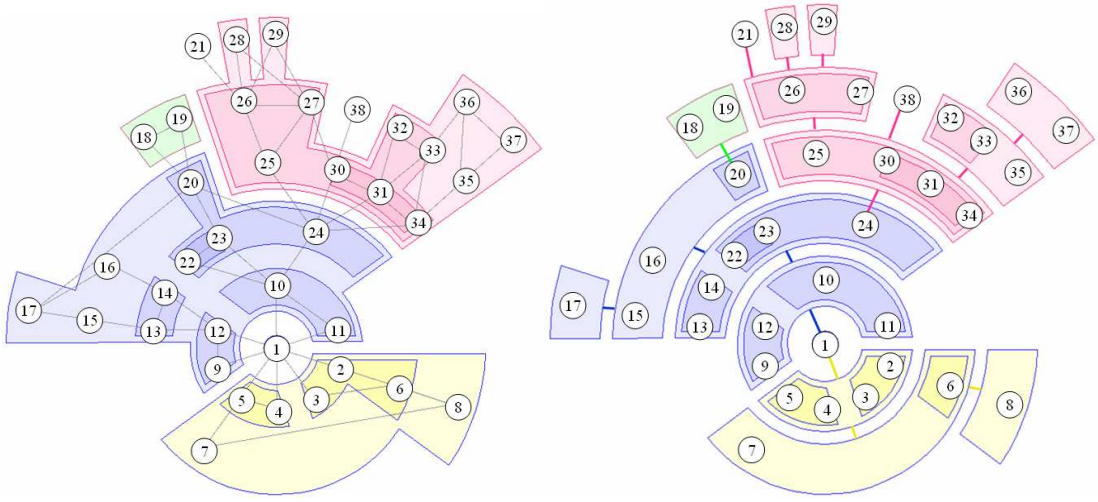


Figure 180 – (a) arbre de silhouettes emboîtées (b) arbre de clusters emboîtés

6.3 Table des figures

6.4 Bibliographie

Aiello, W. et F. Chung (2001). Random Evolution in Massive Graphs. 42nd IEEE symposium on Foundations of Computer Science.

Albert, R. et A. L. Barabási (2002). "Statistical mechanics of complex networks." Reviews of modern physics 74(1): 47-97.

Albert, R., H. Jeong, et al. (2000). "Attack and error tolerance of complex networks." Nature 406: 378-382.

Alpert, C. J. et A. B. Kahng (1995). "Recent Developments in Netlist Partitioning: A Survey, Integration." VLSI Journal 19: 1-81.

An, Y., J. Janssen, et al. (2002). Characterizing the Citation Graph as a Self-Organizing Networked Information Space. Second International Workshop on Innovative Internet Computing Systems, LNCS.

Auber, D. (2002). Outils de visualisation de larges structures de données, University Bordeaux I.

Auber, D. (2003). "Tulip : A huge graph visualisation framework." Graph Drawing Softwares, Mathematics and Visualization. Springer-Verlag: 105-126.

Auber, D., Y. Chiricota, et al. (2003). "Multiscale Visualization of Small World Networks." InfoVis: 75-81.

Barabási, A. L. et R. Albert (1999). "Emergence of scaling in random networks." Science 286: 509-512.

Barabási, A. L. et Z. N. Oltvai (2004). "Network biology: understanding the cell's functional organization." Nature Reviews Genetics 5: 101-113.

Battista, G. D. et P. Eades (1994). "Algorithms for Drawing Graphs: an Annotated Bibliography." Computational Geometry: Theory and Applications 4: 235-282.

Battista, G. D., P. Eades, et al. (1999). Graph Drawing: Algorithms for the Visualization of Graphs, Prentice Hall.

Berge, C. (1970). Graphes et Hypergraphes, Dunod.

Bezdek, J. C. et N. R. Pal (1998). "Some new indexes of cluster validity." IEEE Transactions on Systems, Man and Cybernetics 28: 301-315.

Bollobás, B. (1985). Random Graphs. New York, Academic Press.

Bollobás, B. et O. Riordan (2004). "The diameter of scale-free random graphs." Combinatorica 24: 5-34.

Bolshakova, N. et F. Azuaje (2003). "Cluster Validation Techniques For Genome Expression Data." Signal Processing 83(4): 825-833.

Bonacich, P. (1987). "Power and centrality—a family of measures." American Journal of Sociology 92(5): 1170-1182.

Botafogo, R. A., E. Rivlin, et al. (1992). "Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics." ACM Transactions on Information Systems 10(2): 142-180.

Bouroche, J. M. et G. Saporta (2002). L'analyse de données, PUF.

Boutin, F. et M. Hascoët (2003). Focus-Based Clustering for Multi-Scale Visualization. IV'2003.

Boutin, F. et M. Hascoët (2004). Cluster Validity Indices for Graph Partitioning. Information Visualisation, London.

Boutin, F. et M. Hascoët (2004). Focus Dependent Multi-level Graph Clustering. Advanced Visual Interfaces, AVI 2004.

Boutin, F. et M. Hascoët (2004). Multi-Level Exploration of Citation Graphs European Conference on Digital Library (ECDL), LNCS.

Boutin, F., J. Thièvre, et al. (2005). Focus-based filtering + clustering technique for power-law networks with small world phenomenon. vda 2006, SPIE.

Boutin, F., J. Thièvre, et al. (2005). Multilevel Compound Tree - Construction Visualization and Interaction. Interact, Rome, LNCS.

Brandes, U. (2001). "A faster algorithm for betweenness centrality." Journal of Mathematical Sociology 25: 163-177.

Brandes, U., M. Gaertler, et al. (2003). Experiments on Graph Clustering Algorithms. ESA'03, LNCS.

Brandes, U. et T. Willhalm (2002). Visualization of bibliographic networks with a reshaped landscape metaphor. 4th Joint Eurographics and IEEE TCVG Symposium on Visualization (VisSym '02), ACM.

Brockenauer, R. et S. Cornelsen (2001). "Drawing Clusters and Hierarchies." LNCS Springer Verlag 2025: 194-228.

Buckley, F. et F. Harary (1990). Distance in graphs. Redwood City.

Chamberlain, B. L. (1998). Graph Partitioning Algorithms for Distributing Workloads of Parallel Computations. Washington, University.

Chen, C. (1999). "Visualizing Semantic Spaces and Author Co-Citation Networks in Digital Libraries." Information Processing & Management 35: 401-420.

Chiricota, Y., F. Jourdan, et al. (2003). "Software component capture using graph clustering." 11th International Workshop on Program Comprehension: 217-226.

CiteSeer "Scientific Literature Digital Library." http://citeseer.ist.psu.edu/.

Coppersmith, D. et S. Winograd (1990). "Matrix Multiplication via Arithmetic Programming." Journal of Symbolic Computing 9: 251-280.

Costenbader, E. et T. W. Valente (2003). "The stability of centrality measures when networks are sampled." Social Networks 25: 283-307.

Davidson, R. et D. Harel (1996). "Drawing graphs nicely using simulated annealing." ACM Transactions on Graphics 15(4): 301-331.

Davies, D. L. et D. W. Bouldin (1979). "A cluster separation measure." IEEE Transactions on Pattern Recognition and Machine Intelligence 1(2): 224-227.

Ding, C., H. Xiaofeng, et al. (2001). Spectral min-max cut for graph partitioning and data clustering. Berkeley, California University.

Duncan, C. A. (2000). "Balanced Aspect Ratio Trees and Their Use for Drawing Large Graphs." Graph Algorithms and Applications 4(3): 19-46.

Dunn, J. (1974). "Well separated clusters and optimal fuzzy partitions." Journal of Cybernetics 4: 95-104.

Eades, P. (1984). "A heuristic for graph drawing." Congressus Numerantium 42: 146-160.

Eades, P. (1992). "Drawing Free Trees." Bulletin of the Institute for Combinatorics and its Applications: 10-36.

Eades, P. (1996). Multilevel Visualization of Clustered Graphs. Graph Drawing, Berkeley, California, Springer Verlag.

Elsner, U. (1997). Graph partitioning - A survey. Chemnitz, Technische Universitat.

Erdös, P. et A. Rényi (1959). "On random graphs." Ubl. Math. Debrecen 6: 290-297.

Erdös, P. et A. Rényi (1961). "On the strength of connectedness of a random graph." Acta Mathematica Scienta Hungaria 12: 261-267.

Evans, J. T. et E. Mineka (1992). Optimization algorithms for networks and graphs. New York, Marcel Dekker inc.

Fekete, J. D. (2004). The InfoVis Toolkit. InfoVis.

Feng, Q. (1997). Algorithms for Drawing Clustered Graphs, Newcastle.

Forster, M. (2002). Applying Crossing Reduction Strategies to Layered Compound Graphs. Graph Drawing.

Freeman, L. C. (1979). "Centrality in social networks: conceptual clarification." Social Networks 1: 215-239.

Freitas, C. M. D. S., P. R. G. Luzzardi, et al. (2002). Usability issues in the evaluation of information visualization techniques. AVI.

Friedkin, N. E. (1991). "Theoretical foundations for centrality measures." American Journal of Sociology 96(6): 1478-1504.

Fruchterman, T. M. J. et E. M. Reingold (1991). "Graph drawing by force-directed placement." Software – Practice and Experience 21(11): 1129-1164.

Furnas, G. W. (1986). Generalized Fisheye Views. CHI.

Gajer, P., M. T. Goodrich, et al. (2000). A Fast Multi-Dimensional Algorithm for Drawing Large Graphs. Graph Drawing.

Gansner, E. R., E. Koutsofios, et al. (1993). "A technique for drawing directed graphs." Software Engineering 19(3): 214-230.

Gibbons, A. (1985). Algorithm graph theory, Cambridge university press.

Glover, F. (1989). "Tabu Search." ORSA Journal of Computing 1: 190-206.

Godin, C. et Y. Caraglio (1996). A multiscale model of plant topological structures. Montpellier, CIRAD, Laboratoire de modélisation des plantes.

Goh, K. I., E. S. Oh, et al. (2002). Classification of scale free networks. National Academy of Sciences, USA.

Grivet, S., D. Auber, et al. (2004). Bubble tree drawing algorithm. International Conference on Computer Vision and Graphics.

Grokker http://www.grokker.com/.

Halkidi, M., Y. Batistakis, et al. (2001). "On clustering validation techniques." JIIS 17: 107-145.

Han, E. H. et G. Karypis (2000). Centroid-Based Document Classification: Analysis and Experimental Results. 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD).

Hascoët, M. (1999). Navigation and interaction within graphical bookmarks, LRI.

Heer, J., S. K. Card, et al. (2005). Prefuse: a toolkit for interactive information visualization. CHI, Human Factors in Computing Systems.

Henry, T. R. (1992). Interactive Graph Layout: The Exploration of Large Graphs. Department of Computer Science. Tucson, University of Arizona**:** 101.

Herman, I., M. Delest, et al. (1990). "Tree visualization and navigation clues for information visualization." Computer Graphics Forum 17(2): 153-165.

Herman, I. et G. Mélançon (2000). DAG drawing from an information visualization perspective Eurographics & IEEE TCVG Symposium on Visualization.

Herman, I., G. Mélançon, et al. (2000). "Graph Visualisation in Information Visualisation: a Survey." IEEE Transactions on Visualization and Computer Graphics 6(1): 24-44.

Hong, S. H. et P. Eades (2005). "Drawing Planar Graphs Symmetrically, II: Biconnected Planar Graphs." Algorithmica 42(2): 159-197.

Huang, M. L., E. P., et al. (1998). "WebOFDAV - Navigating and Visualizing the Web On-line with Animated Context Swapping." 7th World Wide Web Conference Elsevier Science: 636-638.

Huang, X., P. Eades, et al. (2005). "A Framework of Filtering, Clustering and Dynamic Layout Graphs for Visualization." ACSC 2005: 87-96.

Hubert, L. J. et P. Arabie (1985). "Comparing partitions." Journal of Classification 2: 193-218.

InfoVis'Contest (2004). http://www.cs.umd.edu/hcil/iv04contest.

Jain, A. K. et R. C. Dubes (1988). Algorithms for Clustering Data, Prentice-Hall, Inc.

Jain, A. K., M. N. Murty, et al. (1999). "Data clustering: A Review." ACM Computing Surveys 31(3): 264-323.

Janson, S., T. Luczak, et al. (1999). Random Graphs. New York, John Wiley.

Jeong, H., B. Tombor, et al. (2000). "The large-scale organization of metabolic networks." Nature 407: 651 - 654.

Jeong, H., S. P. Mason, et al. (2001). "Lethality and centrality in protein networks." Nature 411(41).

Johnson, B. et B. Shneiderman (1991). Tree–maps: a Space–filling Approach to the Visualization of Hierarchical Information Structures. Visualisation.

Johnson, D. S., C. R. Aragon, et al. (1989). "Optimization by simulated annealing: An experimental evaluation; part I, graph partitioning." Operations Research 37(6): 865-892.

Jourdan, F. (2004). Visualisation d'information : dessin, indices structuraux et navigation. Applications aux réseaux biologiques et aux réseaux sociaux. University Montpellier II.

Jünger, M. et P. Mutzel (1997). "2-layer straightline crossing minimization: Performance of exact and heuristic algorithms." Graph Algorithms and Applications 1(1): 1-25.

Jungnickel, D. (1999). Graphs, Networks and Algorithms. Berlin.

Kant, G. (1996). "Drawing planar graphs using the canonical ordering." Algorithmica 16(1): 4-32.

Karypis, G., E.-H. Han, et al. (1999). "CHAMELEON: A hierarchical clustering algorithm using dynamic modeling." IEEE Computer 32(8): 68-75.

Karypis, G. et V. Kumar (1998). "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs." SIAM Journal on Scientific Computing 20(1): 359-392.

Kernighan, B. W. et S. Lin (1970). "An efficient heuristic procedure for partitioning graphs." Bell System Technical Journal 49(2): 291-308.

Kim, D. H., J. D. Noh, et al. (2004). "Scale-free trees: the skeletons of complex networks." Physical Review E 70(046126).

Kleinberg, J. (1999). "The Small-World Phenomenon: An Algorithmic Perspective." Cornell Computer Science Technical Report 99-1776.

Kochen, M. (1989). The Small World. Norwood, Ablex.

Koenig, P. Y. et G. Mélançon (2005). Mémoire de DEA, Visualisation focus + contexte et navigation multi échelle de données clusterisées, Université Montpellier II.

Koren, Y., L. Carmel, et al. (2002). ACE: A Fast Multiscale Eigenvectors Computation for Drawing Huge Graphs. Information Visualization.

Lamping, J., R. Rao, et al. (1995). A Focus+context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. CHI.

Law, M. H. et A. K. Jain (2003). Cluster validity by bootstrapping partition, Michigan State University.

Lawrence, L., K. Bollacker, et al. (1999). Indexing and retrieval of scientific literature. CIKM, Kansas City, Missouri.

Lebart, L., A. Morineau, et al. (2005). Statistique exploratoire multidimensionnelle, Dunod.

Lee Giles, C., K. D. Bollacker, et al. (1998). CiteSeer: An Automatic Citation Indexing System. Digital library.

Mancoridis, S., B. S. Mitchell, et al. (1998). Using Automatic Clustering to Produce High-Level System Organizations of Source Code. 6th International Workshop on Program Comprehension, IEEE.

Marti, R. et M. Laguna (2003). "Heuristics and Meta-Heuristics for 2-Layer Straight Line Crossing Minimization." Discrete Applied Mathematics 127(3): 665-678.

Milgram, S. (1967). "The Small World Problem." Psychology Today: 60-67.

Munzner, T. (1998). Drawing Large Graphs with H3Viewer and Site Manager. Graph Drawing.

Munzner, T. (1998). "Exploring large graphs in 3D hyperbolic space." IEEE Computer Graphics and Applications 18(4): 18-23.

Munzner, T., F. Guimbretiere, et al. (2003). "TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility." ACM Transaction on Graphics (TOG) 22(3): 453-462.

Newman, M. E. J. (2001). "Clustering and preferential attachment in growing networks." oai:arXiv.org:cond-mat/0104209.

Newman, M. E. J. (2002). "Random graphs as models of networks." eprint arXiv:cond-mat/0202208.

Noack, A. (2003). An energy model for visual graph clustering. Graph Drawing, LNCS.

Pearson, K. (1900). "On a criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can reasonably be supposed to have arisen from random sampling." Philosophical Magazine 50: 157-175.

Plaisant, C., J. Grosjean, et al. (2002). SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. InfoVis, Boston.

Pool, I. et M. Kochen (1978). "Contacts and influence." Social Networks 1: 1-48.

Quigley, A. J. (2000). Large Scale 3D Clustering and Abstraction. Pan-Sydney workshop on Visualisation, Sydney.

Quigley, A. J. et P. Eades (2000). Graph Drawing, Clustering, and Visual Abstraction. Graph Drawing, LNCS.

Ramaswamy, L., A. Iyengar, et al. (2003). "Connectivity based node clustering in decentralized peer-to-peer networks." 3rd International Conference on Peer-to-Peer Computing: 66-73.

Reid, A., C. Fan, et al. (2004). "Drawing power law graphs." LNCS Springer Verlag 3383: 12-17.

Reingold, E. M. et J. S. Tilford (1981). "Tidier Drawing of Trees." IEEE Transaction on Software Engineering SE-7(2): 223-228.

Robertson, G. G., J. D. Mackinlay, et al. (1993). Cone Trees: Animated 3D Visualizations of Hierarchical Information. CHI.

Rousseeuw, P. J. (1987). "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis." J. Comp App. Mat 20: 53-65.

Sander, G. (1996). Layout of compound directed graphs, Universität Saarbrücken,.

Saporta, G. (1990). Probabilités, analyse des données et statistique.

Sarkar, M. et M. H. Brown (1992). Graphical Fisheye Views of Graphs. CHI.

Schaffer, D., Z. Zhenping, et al. (1996). "Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods." ACM Transactions on Computer-Human Interaction 3(2): 162-188.

Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualization. IEEE Symposium on Visual Languages.

Stasko, J. et E. Zhang (2000). Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations. InfoVis.

Statistica "logiciel de statistiques." http://www.statsoft.com.

Steinbach, M., G. Karypis, et al. (2000). A Comparison of Document Clustering Techniques. TextMining Workshop, KDD. University of Minnesota.

Sugiyama, K. et V. Misue (1991). "Visualization of structural information: automatic drawing of compound graphs." IEEE Trans. System Man Cybernetics 21: 876-892.

Sugiyama, K., S. Tagawa, et al. (1981). "Methods for Visual Understanding of Hierarchical System Structures." IEEE Transactions on Systems Man and Cybernetics 11(2): 109-125.

van Dongen, S. (2000). Performance criteria for graph clustering and Markov cluster experiments. Amsterdam, National Research Institute for Mathematics and Computer Science in the Netherlands.

van Dongen, S. M. (2000). Graph Clustering by Flow Simulation, Utrecht.

van Ham, F. et J. J. van Wijk (2004). Interactive Visualization of Small World Graphs. InfoVis.

Walker, J. Q. (1990). "A Node-Positioning Algorithm for General Trees." Software – Practice and Experience 20(7): 685-705.

Walshaw, C. (2000). A multilevel algorithm for force-directed graph drawing. 8th International Symposium on Graph Drawing.

Watts, D. J. (1999). Small Worlds. Princeton, Princeton University Press.

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

Yang, J., M. O. Ward, et al. (2003). InterRing: An Interactive Tool for Visually Navigating and Manipulating Hierarchical Structures. InfoVis 2002.

Yee, K. P., D. Fisher, et al. (2001). Animated Exploration of Dynamic Graphs with Radial Layout. Information Visualization.

Zhao, Y. et G. Karypis (2001). Criterion functions for document clustering: experiments and analysis, University Minnesota.

6.5 Sommaire général

Thèse d'informatique

Résumé

Cette thèse étudie les caractéristiques de réseaux d'interactions réels, notamment les propriétés « petit monde » et « sans échelle ». Elle présente diverses techniques de filtrage, partitionnement et visualisation de ces réseaux.

La propriété « sans échelle » provient du fait que ces réseaux d'interactions sont le plus souvent dans un processus de croissance : les nouveaux noeuds établissent, de préférence, des liens avec des nœuds existants à fort degré. On parle d'« attachement préférentiel ». La propriété « petit monde » s'explique par l'adage : « les amis de mes amis sont mes amis ».

Les réseaux d'interaction présentent souvent un noyau dense difficilement analysable et visualisable à l'aide de techniques de partitionnement et de dessin classiques.

Cette étude introduit une nouvelle technique de filtrage permettant l'extraction d'une structure dite « arborée » ayant également les propriétés « petit monde » et « sans échelle ». Le réseau, ainsi filtré, est organisé en un arbre de silhouettes emboîtées. Cette structure multi-échelles, facilement visualisable et navigable, présente une organisation contextuelle du réseau autour d'un focus utilisateur.

La nouvelle technique de partitionnement optimise les critères de qualité recensés et introduits dans cette thèse. Par ailleurs, l'étude du contenu des silhouettes, menée a posteriori, souligne la qualité de l'utilisation conjointe du filtrage et du partitionnement.

Mots clés : visualisation d'information, réseaux d'interactions, graphe « sans échelle », graphe « petit monde », filtrage, partitionnement, structure multi-échelles, focus utilisateur

Abstract

The thesis is focused on the features of real interaction networks, especially on the "small world" and "scale free" properties. It presents various filtering, clustering and visualization techniques of these networks.

The "scale free" property comes from these interaction networks being most often in growth process: the new nodes tend to connect to high degree nodes. This process is called "preferential attachment". The "small world" property recall the saying: "my friends' friends are my friends".

Interaction networks usually have a dense core that is difficult to analyse and to visualize with classical clustering and drawing techniques.

This study develops a new filtering technique allowing extracting a "tree like" structure also having "small world" and "scale free" properties. The resulting network is organized into a tree of nested silhouettes. This multi scaling structure is easily drawn and explored. It presents a contextual organization around user focus.

The new clustering technique optimizes the quality criteria listed and presented in the thesis. Moreover, the silhouettes'content interpretation emphasizes the quality of the joint use of filtering and clustering techniques.

Title: Network multi-scaling filtering, clustering and visualisation techniques from a focus

Keywords: information visualization, interaction networks, scale free graph, small world graph, filtering, clustering, multi scale structure, user focus

Adresse du laboratoire

LIRMM, UMR 5506, 161 rue Ada 34392 Montpellier Cedex 5 - France