Partitionnement de grands graphes : mesures, algorithmes et visualisation

François Queyroi

N^o d'ordre : 4863

THÈSE

présentée à

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET INFORMATIQUE

par François QUEYROI

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : Informatique

Partitionnement de grands graphes : Mesures, Algorithmes et Visualisation

---

Soutenue le : 10/10/2013

Après avis de :

MMe.	Pascale Kuntz-Cosperec	Professeur	**Rapportrice**
M.	Matthieu Latapy	Directeur de Recherche	**Rapporteur**

Devant la Commission d'Examen composée de :

M.	James Abello	Professeur	**Examinateur invité**
M.	Romain Bourqui	Maître de conférences .	**Co-directeur de thèse**
MMe.	Maylis Delest	Professeur	**Directrice de thèse**
MMe.	Laurence Duval	Maître de conférences .	**Examinatrice invitée**
M.	Cyril Gavoille	Professeur	**Président**
MMe.	Pascale Kuntz-Cosperec	Professeur	**Rapportrice**
M.	Matthieu Latapy	Directeur de Recherche	**Rapporteur**

- 2013 -

Remerciements

Mes remerciements vont d'abord à l'Université de Bordeaux et au Laboratoire Bordelais de Recherche en Informatique (LaBRI) grâce auxquels j'ai pu réaliser cette thèse mais également à l'Institut Polytechnique Bordelais (IPB) au sein duquel j'ai découvert le métier d'enseignant.

Je remercie mes rapporteurs, Pascale Kuntz-Cosperec et Matthieu Latapy, ainsi que les membres de mon jury pour l'intérêt porté à mon travail et leurs remarques sur le présent manuscrit.

Un grand merci à mes directeurs de thèse, Maylis Delest et Romain Bourqui, qui m'ont fait découvrir l'univers de la recherche et un domaine d'étude nouveau et passionnant, à savoir l'analyse des réseaux. Travailler avec eux fût une très bonne expérience tant sur le plan professionnel que sur le plan humain. Je souhaite par la même occasion remercier l'ensemble des membres du thème *Visualisation de grandes masses de données* du LaBRI où l'entraide et l'amitié ont fait que ces trois dernières années furent agréables. Je remercie également l'ensemble des chercheurs que j'ai rencontré et avec qui j'ai collaboré pendant cette thèse, en particulier les professeurs James Abello et Yves Chiricota pour leur accueil dans leur pays et au sein de leur laboratoire.

Enfin, je remercie ma famille et en particulier ma mère Claudine pour m'avoir encouragé dans la poursuite de mes études. Merci à ma copine, Clélie, pour son soutien et sa patience, ces années auraient été difficiles sans elle.

Cette thèse est dédiée à mon grand-père, Louis, qui nous a quitté durant la rédaction de ce manuscrit.

Partitionnement de grands graphes : Mesures, Algorithmes et Visualisation

Résumé : L'analyse de réseaux (représentés par des graphes) est une composante importante dans la compréhension de systèmes complexes issus de nombreuses disciplines telles que la biologie, la géographie ou la sociologie. Nous nous intéressons dans cette thèse aux décompositions de ces réseaux. Ces décompositions sont utiles pour la compression des données, la détection de communautés ou la visualisation de graphes.

Une décomposition possible est un partitionnement hiérarchique des sommets du graphe. Nous traitons de l'évaluation de la qualité de telles structures (leur capacité à bien capturer la topologie du graphe) par le biais de mesures de qualité. Nous discutons ensuite l'utilisation de ces mesures en tant que fonctions objectives à maximiser dans le cadre d'algorithmes de partitionnement. Enfin, nous nous intéressons à la définition de métaphores visuelles efficaces permettant de représenter différentes décompositions de graphes.

Mots-clés : détection de communautés, partitionnement, visualisation de graphes.
Discipline : Informatique.

Graph Partitioning : Measures, Algorithms and Visualization

Abstract : Network analysis is an important step in the understanding of complex systems studied in various areas such as biology, geography or sociology. This thesis focuses on the problems related to the decomposition of those networks when they are modeled by graphs. Graph decomposition methods are useful for data compression, community detection or network visualisation.

One possible decomposition is a hierarchical partition of the set of vertices. We propose a method to evaluate the quality of such structures using quality measures and algorithms to maximise those measures. We also discuss the design of effective visual metaphors to represent various graph decompositions.

Keywords : community detection, partitioning, graph visualization.
Field : Computer Science.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1
Introduction

Les *réseaux complexes* sont au cœur des sciences naturelles et humaines car ils permettent de représenter les interactions entre les différents éléments d'un système, que ce soit un échange de photos entre amis sur un réseau social, des interactions entre protéines dans une cellule ou la circulation de personnes au sein d'un pays. Le terme *complexe* (formé du latin *con-* "avec" *plexus* "entrelacement") est presque redondant dans ce contexte puisqu'un réseau suppose l'existence de relations entre ses membres.

Ces réseaux sont souvent modélisés par des graphes, une structure permettant l'encodage de données relationnelles. Quel que soit le domaine applicatif, cette modélisation sert à l'étude de la structure émergeant des entrelacements entre individus. Ainsi, les utilisateurs d'un réseau social auront tendance à former des groupes plus fortement connectés entre eux qu'avec le reste du réseau : ils formeront des *communautés*. La forme et la taille de ces communautés sont *a priori* inconnues, de même que la façon dont elles s'imbriquent.

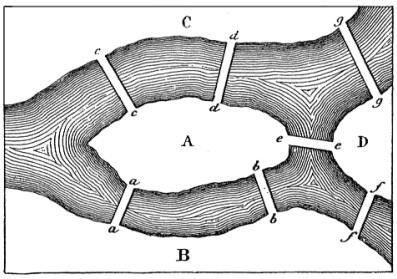
L'identification de ce type de structure est un défi majeur dans le domaine de l'analyse de données. En effet, une *décomposition* du réseau reflétant sa topologie est une étape primordiale pour la compréhension du système et son exploration. Nous traitons de l'évaluation de la qualité de telles structures (leur capacité à bien capturer la topologie du graphe) par le biais de mesures de qualité. Nous discutons ensuite l'utilisation de ces mesures en tant que fonctions objectives à maximiser dans le cadre d'algorithmes de partitionnement. Enfin, nous nous intéressons à la définition de métaphores visuelles efficaces permettant de représenter différentes décompositions de graphes.

1.1 Les graphes : un modèle pour l'étude d'interactions

Un graphe désigne un objet mathématique formé d'entités, appelées *sommets*, interagissant entre elles. Ces interactions sont représentées par des *arêtes* et peuvent correspondre à des contraintes, des dépendances ou des échanges entre les entités.

L'utilisation de graphes pour modéliser remonte à 1741 lorsque Leonhard Euler proposa le problème des *sept ponts de Königsberg* [49]. Il consiste à déterminer l'existence d'une promenade passant une et une seule fois par chaque pont de la ville (voir Figure 1.1(a)). Ce type de problème est une question posée sur la *topologie* du graphe. La topologie correspond à l'espace formé par les relations du type "est voisin de" ou "est accessible à partir de". L'enjeu est souvent d'extraire des propriétés globales à partir de ces relations locales.

Un autre exemple célèbre en *théorie des graphes* est le problème de *coloration* [81] : combien de couleurs sont nécessaires au minimum pour pouvoir colorier une carte de façon à ce que deux régions adjacentes n'aient pas la même couleur (voir Figure 1.1(b)) ?



(a) Illustration du problème des sept ponts de Königsberg

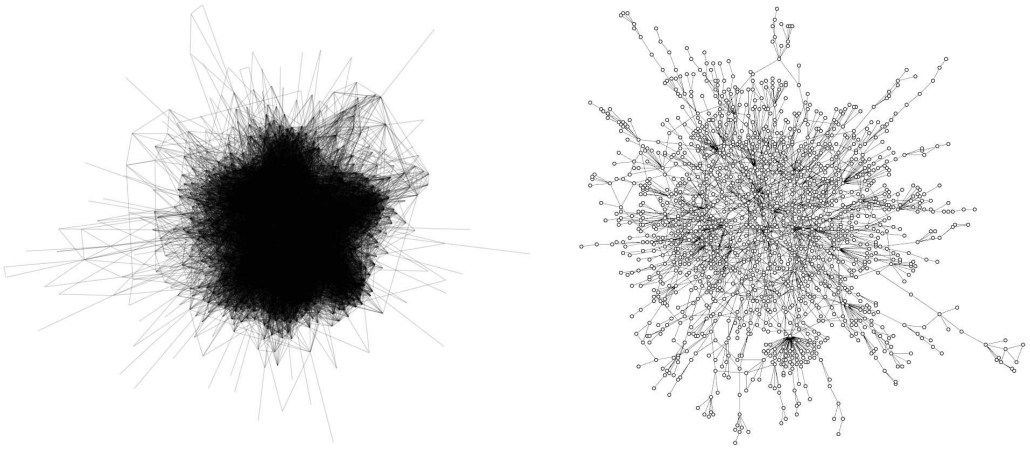
(b) Illustration du problème de coloration sur la carte de l'Europe

FIGURE 1.1: Illustration de certains problèmes issus de la théorie des graphes

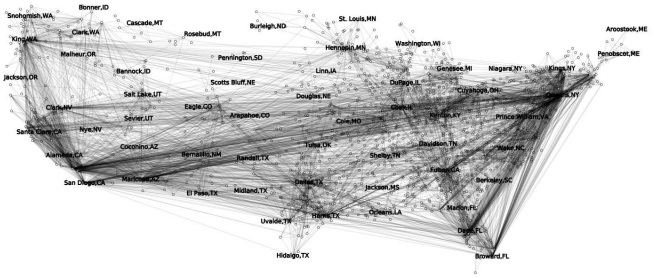
Dans les exemples donnés ici, les problèmes sont résolus en identifiant certaines propriétés des graphes. Pour la coloration de cartes, le graphe représentant les différentes contraintes est *planaire*. Ce type de propriété aide à la résolution du problème. Cependant les graphes peuvent être utilisés pour modéliser des systèmes beaucoup plus complexes issus de domaines variés. Cette modélisation est alors utile pour répondre à des problèmes d'*analyse de données*. Les graphes générés ne vérifient généralement pas les propriétés permettant de résoudre aisément ces problèmes.

L'analyse de données regroupe les méthodes permettant d'extraire des connaissances à partir d'un ensemble d'observations. Dans ce cadre, les statistiques servent à étudier des variables décrivant une population et tester des hypothèses telles que "le temps passé par une personne sur son ordinateur dépend de son âge". Un prédicat souvent utilisé est que les individus sont indépendants. Cette contrainte implique que l'état d'un individu ne dépend pas de l'état d'un autre individu dans la population. Dans ce cadre, l'hypothèse est testée en utilisant une régression linaire [96]. Une autre hypothèse pour cet exemple pourrait être "le temps passé par une personne sur son ordinateur dépend du temps passé par ses amis sur un ordinateur". L'approche du problème est différente car on cherche

maintenant à tester l'indépendance entre les individus. Ce problème peut être abordé en tenant compte des relations entre eux, c'est ce que les graphes permettent de faire. Cette hypothèse peut être testé en utilisant des indices de corrélations spatiales (topologiques) venant de la géographie quantitative [90]. Les approches possibles dépendent d'une prise en compte simulatannée de variables exogènes (issues des données brutes du réseau) et de variables endogènes (issues de la topologie du réseaux).



(a) Réseau social *Facebook* de l'université Caltech (b) Réseau d'interactions de protéines de *Yeast*



(c) Réseau migratoire aux États-Unis

FIGURE 1.2: Illustration de différents réseaux complexes

Les graphes sont ainsi utilisés pour modéliser des interactions entre individus (au sens statistique). Des méthodes et algorithmes, issus de la théorie des graphes, permettent de répondre à des problèmes venant d'autres disciplines. Les domaines concernés sont nombreux, on peut évoquer la sociologie [132], la géographie [17] ou la biologie [40]. Les système concernés sont appelés *réseaux complexes*. Ce sont par exemple les réseaux sociaux issus du Web, les flux migratoires entre les villes d'un pays ou les réseaux d'interactions entre protéines. Une illustration de chaque exemple est disponible dans la Figure 1.2.

L'étude de ces réseaux (appelée *analyse de réseaux*) a connu un essor très important

durant les deux dernières décennies. Parmi les résultats importants, on peut citer les travaux sur l'identification de constantes structurelles dans les réseaux complexes.
Un premier résultat est l'observation d'une *invariance d'échelle* dans la distribution du nombre de connections [16, 5] qui peut être approximée par une loi de puissance. Cela implique qu'il est facile de trouver des sommets ayant un nombre de connections beaucoup plus grand que la moyenne. Ce résultat est néanmoins vivement contesté pour sa validité statistique (voir la contre-étude de Perline [107]). Un second résultat montre que les distances (en terme de plus court chemin) dans les réseaux réels sont relativement courtes comparées à la taille du réseau. Ce phénomène est connu sous le nom de *petit-monde* [149].
Une dernière propriété intéressante des réseaux complexes est la présence d'une structure de communautés. Ce point est plus amplement discuté dans la section 1.2.
L'identification de ces propriétés a fait l'objet de nombreuses vulgarisations (voir à ce sujet le documentaire *"Connected : The Power of Six Degrees (How Kevin Bacon Cured Cancer)"*, BBC, 2008).

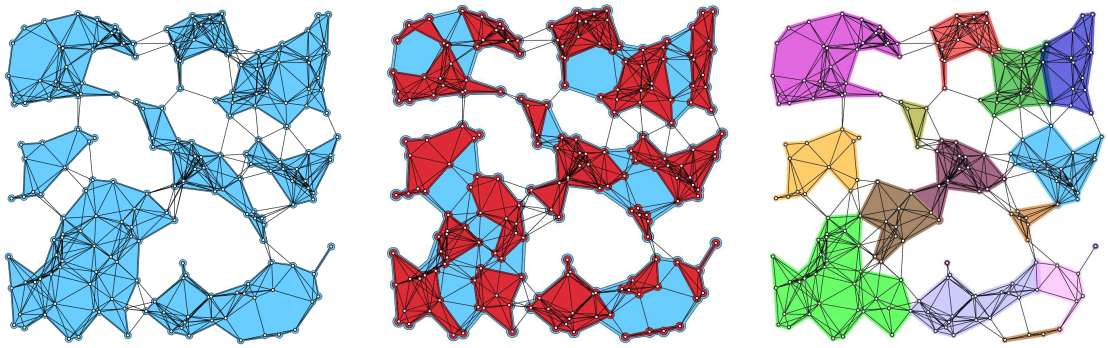
1.2 Partitionnement de graphes et détection de communautés

D'après une récente étude [59], la taille totale des données générées dans le monde s'élèverait à 2.8 zettaoctets (mille milliards de gigaoctets) en 2012 pour atteindre 40 zettaoctets en 2020. Dans ce cadre, la taille des réseaux à analyser va également augmenter ou, du moins, l'analyse systématique de réseaux de grande taille sera nécessaire. Cet enjeu met en lumière l'importance du partitionnement de données. En effet, le partitionnement de données rassemble les méthodes permettant de découper une population en groupes. Les critères guidant ce processus peuvent varier mais les objectifs sont connus. Un premier objectif est de compresser les données en entrée et faciliter leur traitement en considérant chaque groupe comme un sous-ensemble indépendant. Cette approche correspond au paradigme *"diviser pour régner"*. Un deuxième est d'extraire de la connaissance à partir de ces groupes. Ces besoins se retrouvent naturellement dans l'analyse de réseaux et se traduisent par les problèmes de *partitionnement de graphes* ou, plus généralement, de *détection de communautés*.

Nous avons déjà mentionné plusieurs problèmes célèbres en théorie des graphes. Le partitionnement de graphes est l'un d'eux. Il consiste à trouver une partition des sommets respectant une ou plusieurs propriétés. Ces propriétés sont de nature très diverses. On peut, par exemple, chercher à minimiser la *capacité* des arêtes externes, c'est-à-dire le nombre d'arêtes ou la somme totale des poids des arêtes reliant deux groupes. Cette seule contrainte définit le problème de la *coupe minimum* qui peut être résolu par des

algorithmes polynomiaux [29]. Le problème devient difficile si on cherche à diviser le graphe en k groupes de taille équilibrée car il n'existe alors pas d'algorithme polynomial pour y répondre. Le partitionnement équilibré d'un graphe intervient par exemple dans des problèmes d'ordonnancement de tâches pour des systèmes multi-cœurs [11].

Le problème de partitionnement de graphes s'est généralisé avec l'étude des réseaux complexes. En effet, une caractéristique importante de certains réseaux est que les densités locales de certains sous-graphes sont relativement fortes comparées à la densité globale du graphe. Par *densité*, on entend ici le rapport entre le nombre d'arêtes observé et le nombre potentiel d'arêtes. Cette dernière observation est liée à la présence de sous-ensembles d'individus qui sont très connectés entre eux et relativement peu connectés avec le reste du réseau. L'ensemble de ces groupes correspond à la *structure de communauté*. Nous désignons ainsi par *détection de communautés*, le problème d'extraction automatique de ces groupes en se basant principalement sur la topologie du graphe.



(a) Partition plate (b) Partition hiérarchique (c) Fragmentation

FIGURE 1.3: Illustration de trois types de décomposition d'un graphe en utilisant des enveloppes concaves [85]

La détection de communautés est un enjeu important dans l'analyse de réseau. En biologie, un ensemble de protéines interagissant fortement entre elles forment généralement un module fonctionnel [71]. En géographie, une régionalisation du territoire peut être formée en détectant des ensembles de villes s'échangeant des *commuters* (les personnes vivant et travaillant dans deux villes différentes) [41]. Enfin, en sociologie, des personnes inter-connectées sont susceptibles de partager les mêmes caractéristiques sociales ou culturelles [148]. Nous détaillons ci-dessous trois types de décomposition (voir Figure 1.3).

Une première approche pour détecter les communautés est de chercher une partition des sommets du graphe. Toutefois, le nombre de groupes ou leur taille sont *a priori* inconnus. Ce problème apparaît donc plus "complexe" que ceux évoqués en début de cette section car l'espace de recherche est plus grand. La détection de communautés à travers

le partitionnement de graphes a attiré l'attention de nombreux chercheurs issus de disciplines variées. Les approches utilisées pour répondre à ce problème sont nombreuses. Un état de l'art couvrant la plupart de ces approches est donné dans le chapitre 5. Beaucoup de méthodes se basent sur l'optimisation d'une fonction objective, appelée *mesure de qualité*. La définition de ces mesures est un point crucial dans ce contexte car ce sont elles qui permettent d'explorer efficacement un espace de solutions très large. La mesure la plus utilisée actuellement est la *modularité* Q [101, 100]. Cette mesure souffre cependant de nombreux défauts [55, 66]. Nous discutons, dans le chapitre 3, des approches alternatives pour évaluer la qualité d'une partition. En particulier, nous généralisons une mesure basée sur la densité nommée MQ (définie par Mancoridis *et al.* [93]) en y incluant une pondération des groupes. Nous montrons que cette mesure possède, sous cette forme, de bonnes propriétés statistiques.

Une partition des sommets d'un graphe est une modélisation possible de la structure de communautés d'un réseau complexe. Cependant, de nombreuses études suggèrent la présence d'une organisation *hiérarchique* dans la structure de communauté des réseaux complexes. Par *hiérarchie*, on entend une situation dans laquelle une communauté est récursivement découpée en plusieurs sous-communautés. Ce type de structure peut alors être modélisé par une partition multi-niveaux des sommets d'un graphe.

L'emploi du terme "hiérarchie" peut mener à des confusions car il n'implique pas (forcément) une notion de subordination entre les individus du réseau. Ce concept a notamment été théorisé par Herbert A. Simon [135] pour décrire la dynamique de systèmes complexes. Cette approche se retrouve dans les sciences sociales ou naturelles [112] comme en linguistique [61]. Certains modèles issus de la biologie essaient de capturer la structure hiérarchique présente dans des réseaux formés par diverses entités biologiques [143]. Un autre exemple peut se trouver dans les réseaux d'échanges entre des entités géographiques [18].

Beaucoup d'algorithmes de partitionnement de graphes tentent de reproduire l'organisation hiérarchique du réseau afin d'en extraire une partition pertinente des sommets. Ces méthodes sont soit *divisives* soit *agglomératives*. Toutefois, les travaux réalisés sur le problème d'extraction d'une structure de communauté hiérarchique sont relativement peu nombreux. En particulier, l'évaluation de la qualité d'une partition hiérarchique ou de la distance (similarité) entre ces structures sont des problèmes peu étudiés. Nous proposons dans le chapitre 4 une mesure permettant d'évaluer la qualité des partitions hiérarchiques [116] qui généralise les mesures de qualité des partitions plates telles que la modularité Q ou MQ. Nous montrons ensuite dans le chapitre 5 comment ce nouveau type de mesure peut être utilisé dans le cadre d'algorithmes de partitionnement multi-niveaux [115].

Le partitionnement des sommets d'un graphe n'est pas toujours adapté pour modéliser

une structure de communauté. En effet, dans une partition, un élément ne peut appartenir qu'à un unique groupe. Or dans certains exemples, cette contrainte devrait pouvoir être relâchée. Le cas le plus illustratif est celui des réseaux sociaux issu du Web, un individu peut en effet appartenir à plusieurs "cercles" comme celui de son travail, ses amis ou sa famille et il ne semble pas raisonnable de l'assigner dans une seule de ces trois communautés. Pour modéliser cette situation, une *fragmentation* de graphes, c'est-à-dire une décomposition du réseau en groupes d'individus pouvant se chevaucher, est utilisée. Ce nouveau problème est très étudié et de nombreux algorithmes de fragmentation existent (voir notamment [104, 3, 89, 36]). Il n'est pas abordé dans le présent document. Utiliser une décomposition chevauchante pour détecter des communautés peut sembler plus naturel, d'autant que l'espace des fragmentations de graphes inclut les partitions à un ou plusieurs niveaux. Cependant nous pensons que les trois types de décomposition de graphes décrits ici correspondent à des problèmes qui ne sont pas équivalents. De plus, dans certains cas d'études, le partitionnement peut être vu comme une *classification* des individus. Dans ce cas, la fragmentation avec chevauchements est *a priori* inadaptée.

1.3 Analyse visuelle de graphes

Les progrès récents en analyse de l'information vont vers l'aide au processus de création de connaissances. Dans ce contexte, la visualisation d'informations est une voie de développement qui cherche à intégrer la capacité cognitive de l'être humain. Ce domaine inter-disciplinaire se base ainsi sur "des techniques de représentations visuelles et interactives tirant parti de la bande passante existante entre l'œil humain et son cerveau pour permettre à l'utilisateur de voir, d'explorer et de comprendre de grandes quantités d'information à la fois. La visualisation d'informations est axée sur la création d'approches pour transmettre des données abstraites de manière intuitive" [138].

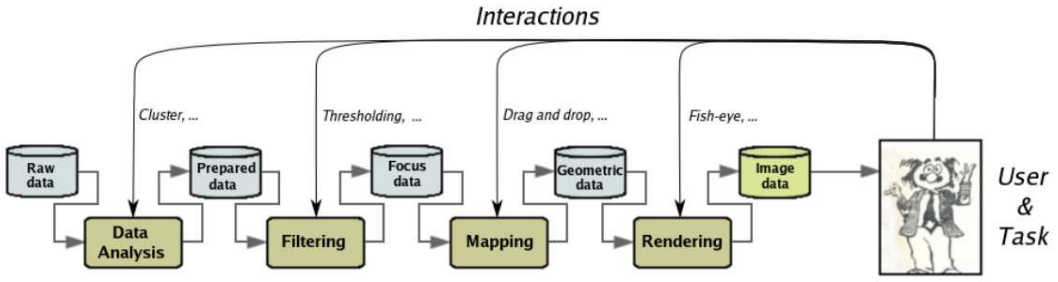


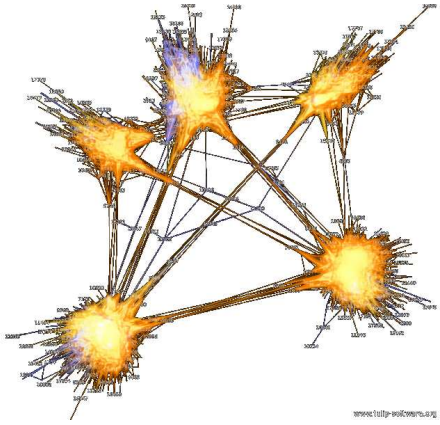
FIGURE 1.4: Illustration du *visualization pipeline* proposé par [44] et adapté par [24]

Les travaux présentés ici s'inscrivent dans la chaîne de traitement de données qui permet à l'utilisateur l'extraction de connaissances à partir de données brutes (ce *"visualization pipeline"* est illustré dans la Figure 1.4). Les données traitées ici sont des abstractions de réseaux complexes modélisées par des graphes. Dans ce cadre, l'étape appelée *"Data Analysis"* comprend le partitionnement de graphes qui peut être vu comme une simplification des données en entrée. Cette étape va permettre la construction de représentations visuelles (l'étape *"Mapping"*) utiles pour l'utilisateur final [2].

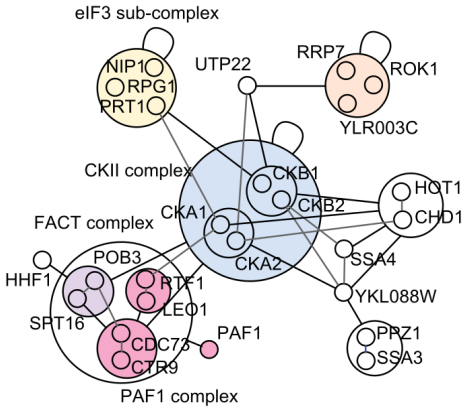
La représentation visuelle de graphes forme un domaine de recherche à part entière. Une abstraction visuelle couramment utilisée (c'est le cas dans ce document) est celle du *diagramme nœuds-liens* bien que d'autres abstractions existent comme la représentation matricielle [62]. La construction de ces métaphores visuelles nécessite la résolution de problèmes algorithmiques comme par exemple minimiser le nombre de croisements d'arêtes dans un plongement d'un graphe en deux dimensions. Des partitionnements ou des décompositions des éléments du graphes sont souvent utilisé pour en faciliter la représentation. La figure 1.5 illustre trois exemples de méthodes facilitant la compréhension des réseaux données en figure 1.2.

Une "bonne" représentation visuelle d'un graphe a l'objectif de fournir des intuitions à l'expert quant à la topologie du graphe. Des mesures *esthétiques* [114] calculées sur un dessin de graphe permettent d'évaluer le respect de certains critères comme, par exemple, la minimisation du nombre de croisements d'arêtes. Une étude récente [141] suggère que ces critères ne sont pas adaptés au cas où le réseau possède une structure de communauté. Nous proposons ainsi dans le chapitre 6 de nouvelles mesures esthétiques prenant directement en compte cette structure de communauté. Nous les utilisons par la suite pour comparer différents algorithmes de dessins au travers d'une analyse statistique. Cette comparaison permet d'évaluer la capacité des algorithmes à représenter un graphe partitionné.

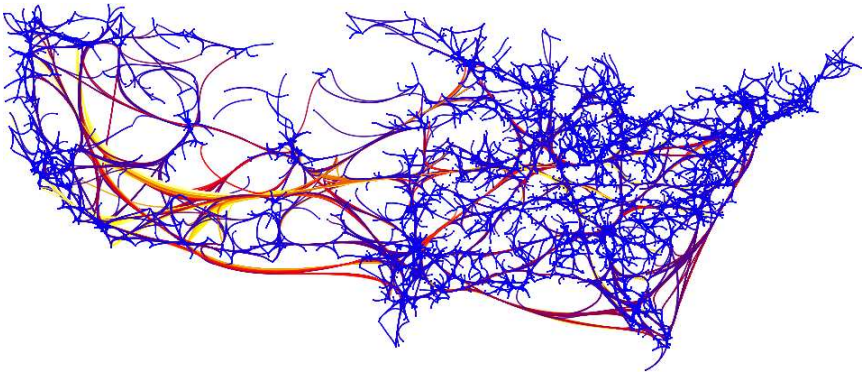
La visualisation d'une structure de communautés peut également être réalisée par l'utilisation de métaphores visuelles représentant explicitement les communautés. Dans le cas où cette structure est modélisée par une fragmentation du graphe, ce problème correspond à la visualisation d'ensembles chevauchants. Il existe de nombreuses méthodes issues du domaine de la visualisation d'information permettant de traiter ce problème [34, 136, 120, 8]. Nous proposons dans le chapitre 7 une solution tirant parti de la topologie du graphe pour générer des enveloppes délimitant les communautés [85]. Notre méthode ne modifie pas le plongement du graphe dans le plan et tente de minimiser l'occlusion visuelle liée à l'appartenance d'éléments (sommets ou arêtes) à différentes communautés.



(a) Réseau social *Facebook* dessiné en utilisant un partitionnement des sommets (`www.tulip-software.org`)



(b) Visualisation d'un réseau d'interactions de protéines basée sur une décomposition en *power graphs* [128]



(c) Réseau migratoire aux États-Unis dessiné avec un regroupement en faiseaux des arêtes [84]

FIGURE 1.5: Utilisation de différents type de regroupements pour faciliter la compréhension des réseaux

1.4 Travaux réalisés et organisation du manuscrit

Les travaux présentés ici contribuent aux domaines du partitionnement et de la visualisation de graphes. En particulier, nous proposons une généralisation de certaines mesures de qualité aux partitions multi-niveaux. Nous utilisons ensuite ce nouveau type de mesures comme base pour des algorithmes de partitionnement. Nous contribuons également aux problèmes de visualisation de graphes partitionnés.

Ces recherches ont donné lieu à l'implémentation de différents algorithmes tant pour le partitionnement de graphes que pour la visualisation au sein du logiciel Tulip [15] (`www.tulip-software.org`). Les expérimentations et les images contenues dans ce manuscrit ont été en grande partie réalisées en utilisant ce logiciel.

Ce document est organisé autour de trois thèmes majeurs qui sont les *mesures* de

qualité de partitionnement, les *algorithmes* pour le partitionnement hiérarchique et la visualisation de graphes possédant une structure de communauté.

Les concepts et les notations utilisés dans ce document sont définis dans le chapitre 2. Nous faisons en outre un état de l'art sur les modèles de graphes aléatoires qui seront utilisés enfin de valider les mesures et les algorithmes que nous proposons.

Les chapitres 3 et 4 sont consacrés à l'évaluation de la qualité des partitions de graphes. Nous discutons dans le chapitre 3 des différents moyens pour évaluer une partition à un niveau des sommets d'un graphe au travers de mesures *additives*. Dans ce cadre, nous donnons une nouvelle définition de la mesure de qualité MQ [93] se basant sur les densités d'arêtes internes et externes aux groupes. Cette mesure permet de répondre à certains problèmes de la *modularité* Q. Dans le chapitre 4, nous proposons une généralisation des mesures de qualité additives à l'évaluation de partitions hiérarchiques de graphes. Notre approche correspond à l'application récursive d'une mesure de qualité classique sur l'*arbre de partition* en faisant intervenir une variable q permettant de garder trace de la profondeur à laquelle un groupe se situe dans la hiérarchie. Nous validons cette approche sur des exemples simples qui confirment l'apport de ce nouveau type de mesure. Ces travaux ont fait l'objet d'une publication dans le journal *Data Mining and Knowledge Discovery* [116].

Dans le chapitre 5, nous présentons différents algorithmes pour partitionner hiérarchiquement un graphe en utilisant notre mesure de qualité multi-niveaux comme fonction objective à maximiser. En particulier, nous proposons une procédure gloutonne de filtrage d'une hiérarchie. Cette approche est mise en pratique avec succès sur les partitions multi-niveaux produites indirectement par un algorithme d'optimisation de la modularité [23]. Ces travaux ont été publié dans le journal *Advances in Knowledge Discovery and Management* [115]. Ce papier a, en outre, été désigné comme meilleur papier académique à la conférence EGC 2012 (Extraction et Gestion de Connaissances).

Les chapitres 6 et 7 traitent du problème de visualisation de graphes avec une structure de communauté dessinés à l'aide de diagrammes nœuds-liens. Une comparaison statistique de différents algorithmes de dessin est réalisée dans le chapitre 6. Cette comparaison s'appuie sur la capacité d'un dessin à bien retranscrire la structure de communauté du réseau. Dans ce cadre, nous proposons des nouvelles *mesures esthétiques* permettant de quantifier la capacité des algorithmes à respecter certains critères. Nous nous intéressons à la représentation explicite de la structure de communauté dans le chapitre 7. Nous y proposons une méthode pour visualiser efficacement une fragmentation de graphes en utilisant des enveloppes concaves. Ce travail a été présenté à la conférence *Information Visualisation* (IV2012) [85].

Une synthèse globale est donnée dans le chapitre 8. Nous reviendrons sur les éléments clés discutés dans chacun des chapitres précédents et nous détaillerons plusieurs perspectives et travaux en cours se basant sur le contenu de ce document.

Chapitre 2
Définitions et Notations

2.1 Ensembles et Partitions

Nous introduisons ici différentes notations relatives aux ensembles et au partitionnement d'ensembles.

Définition 2.1 ***(Ensemble)*** *Un ensemble $S = (s_1, \ldots, s_n)$ désigne une collection d'éléments.*

Les ensembles utilisés dans ce document sont finis et leur taille est notée $|S|$. On note $\emptyset$ l'ensemble vide.

Définition 2.2 ***(Fragmentation)*** *Soit un ensemble S, l'ensemble $\mathcal{C} = (C_1, C_2, \ldots, C_k)$ est une fragmentation de S si et seulement si*

- *$C_i \subseteq S$, $\forall i \in [1, k]$*
- *$\bigcup_{i=1}^{k} C_i = S$*

Les éléments de $\mathcal{C}$ sont alors appelés *fragments*.

Définition 2.3 ***(Partition)*** *Soit un ensemble S, l'ensemble $\mathcal{C} = (C_1, C_2, \ldots, C_k)$ est une partition de S si et seulement si C est une fragmentation et $C_i \cup C_j = \emptyset$, $\forall i \neq j \in [1, k]$.*

Ainsi une partition est un cas particulier de fragmentation.

Définition 2.4 ***(Sous-Partition)*** *Soit un ensemble S et deux partitions $\mathcal{C}_1$ et $\mathcal{C}_2$ de S, on dit que $\mathcal{C}_2$ est une sous-partition de $\mathcal{C}_1$ si et seulement si pour tout $C_i \in \mathcal{C}_1$, il existe un sous-ensemble $\mathcal{C}'_2 \subseteq \mathcal{C}_2$ tel que $\mathcal{C}'_2$ est une partition de l'ensemble C_i.*

Notons que la notion de sous-partition est transitive : pour tout triplet de partitions $\mathcal{C}_1$, $\mathcal{C}_2$ et $\mathcal{C}_3$ de S tels que $\mathcal{C}_2$ est une sous-partition de $\mathcal{C}_1$ et $\mathcal{C}_3$ est une sous-partition de $\mathcal{C}_2$ alors $\mathcal{C}_3$ est une sous-partition de $\mathcal{C}_1$.

Définition 2.5 ***(Partition hiérarchique)*** *Soit un ensemble S, l'ensemble noté $T = (N_1, N_2, \ldots, N_l)$ est une partition hiérarchique de S si et seulement si*

- *N_1 est une partition de S.*
- *N_i est une sous-partition de N_{i-1} pour $i \in [2, l]$.*

Dans ce cadre, un élément N_i de T est appelé niveau. *La quantité l désigne la* hauteur *de la partition hiérarchique T.*

Une partition est donc également un cas particulier de partition hiérarchique : c'est une partition hiérarchique de hauteur 1. Une partition hiérarchique est également un cas particulier de fragmentation.

Définition 2.6 ***(Distance entre partitions)*** *Soit $\mathcal{C}, \mathcal{C}'$ deux partitions d'un même ensemble S. Une fonction $d(\mathcal{C}, \mathcal{C}')$ est appelée une* distance *si et seulement si les propriétés suivantes sont vérifiées :*

- ***Symétrie :*** $\forall \mathcal{C}, \mathcal{C}'$, $d(\mathcal{C}, \mathcal{C}') = d(\mathcal{C}', \mathcal{C})$
- ***Séparation :*** $\forall \mathcal{C}, \mathcal{C}'$, $d(\mathcal{C}, \mathcal{C}') = 0 \Leftrightarrow \mathcal{C} = \mathcal{C}'$
- ***Inégalité triangulaire :*** $\forall \mathcal{C}, \mathcal{C}', \mathcal{C}''$, $d(\mathcal{C}, \mathcal{C}'') \leq d(\mathcal{C}, \mathcal{C}') + d(\mathcal{C}', \mathcal{C}'')$

Définition 2.7 ***(Distance par suppression [7])*** *Soit $\mathcal{C}, \mathcal{C}'$ deux partitions d'un même ensemble S. Pour $Z \subseteq S$, on note $\mathcal{C}(S \setminus Z)$ la partition de $(S \setminus Z)$ obtenue en supprimant les éléments de Z dans chaque groupe $C \in \mathcal{C}$.*
La distance par suppression *entre $\mathcal{C}$ et $\mathcal{C}'$, notée $DS(\mathcal{C}, \mathcal{C}')$, correspond à la taille de l'ensemble $Z \subseteq S$ minimum tel que $\mathcal{C}(S \setminus Z) = \mathcal{C}'(S \setminus Z)$.*

La distance $DS(\mathcal{C}, \mathcal{C}')$ correspond également au nombre minimum d'éléments devant être échangés entre les groupes de $\mathcal{C}$ pour obtenir la partition $\mathcal{C}'$. Cette mesure est bien une distance (voir définition 2.6). Son calcul revient à résoudre un problème d'affectation [28, 82] ou le problème de couverture-sommet dans un graphe parfait [68]. Le calcul se fait donc en temps polynomial par rapport à $|S|$ [111].

Définition 2.8 ***(Information mutuelle [94])*** *Soit $\mathcal{C}, \mathcal{C}'$ deux partitions d'un même ensemble S. L'*information mutuelle *entre $\mathcal{C}$ et $\mathcal{C}'$, notée $MI(\mathcal{C}, \mathcal{C}')$, est donnée par*

$$MI(\mathcal{C}, \mathcal{C}') = \frac{-1}{|S|} \sum_{i=1}^{k} \sum_{j=1}^{k'} |C_i \cap C'_j| \log_2 \left(\frac{|C_i \cap C'_j|^2}{|C_i||C'_j|} \right) \tag{1}$$

L'information mutuelle $MI(\mathcal{C}, \mathcal{C}')$ indique si il existe une dépendance entre $\mathcal{C}$ et $\mathcal{C}'$ en évaluant dans quelle mesure l'observation " les éléments $x, y \in S$ appartiennent (ou non) à un même groupe de $\mathcal{C}$" donne des informations sur la relation entre x et y dans $\mathcal{C}'$. Cette mesure n'est toutefois pas une distance contrairement à la *variation d'information*.

Définition 2.9 ***(Variation d'information [80])*** *Soit $\mathcal{C}, \mathcal{C}'$ deux partitions d'un même ensemble S. La* variation d'information *entre $\mathcal{C}$ et $\mathcal{C}'$, notée $VI(\mathcal{C}, \mathcal{C}')$, est donnée par*

$$VI(\mathcal{C}, \mathcal{C}') = H(\mathcal{C}) + H(\mathcal{C}') - 2MI(\mathcal{C}, \mathcal{C}') \tag{2}$$

*où $H(\mathcal{C}) = -\sum_{i=1}^{k} \frac{|C_i|}{|S|} \log_2 \left(\frac{|C_i|}{|S|} \right)$ est l'*entropie *de la séquence $\left(\frac{|C_1|}{|S|}, \frac{|C_2|}{|S|}, \ldots, \frac{|C_k|}{|S|} \right)$ (voir définition 2.53).*

La mesure VI est bien une distance entre partitions (voir définition 2.6). Elle est bornée par $\log_2(|V|)$.

2.2 Graphes

Définition 2.10 ***(Graphe)*** *Un graphe $G = (V, E, f)$ est une structure formée d'un ensemble V d'éléments appelés* sommets*, d'un ensemble E d'éléments appelés* arêtes *et d'une application surjective $f : E \to V \times V$ associant à chaque arête un couple de sommets.*

Pour une arête e, les deux sommets $f(e) = (u, v)$ sont appelés les extrémités de e. On note $n = |V|$ le nombre de sommets du graphe et $m = |E|$ le nombre d'arêtes du graphe. Dans ce travail, on suppose que les sommets et les arêtes de G sont toujours indexés par un entier de l'intervalle $[1 \dots n]$ et $[1, \dots, m]$ respectivement. On note également $V(G)$ (respectivement $E(G)$) l'ensemble des sommets (resp. arêtes) du graphe G.

Définition 2.11 ***(Graphe simple)*** *Un graphe $G = (V, E, f)$ est dit simple si et seulement si f est une fonction bijective.*

Un graphe qui n'est pas simple est dit multiple. Par la suite, si la mention *multiple* n'est pas présente un graphe sera supposé simple. Dans ce cadre, l'application f est omise et on supposera que $E \subseteq \{(u, v) | u \in V, v \in V\}$. Un graphe G sera donc noté $G = (V, E)$.

Définition 2.12 ***(Graphe non-orienté)*** *Un graphe $G = (V, E, f)$ est non-orienté si l'application f est restreinte aux ensembles de sommets.*

Par la suite si la mention *orienté* n'est pas présente, on considérera que le graphe est non-orienté.

Définition 2.13 ***(Graphe pondéré)*** *Un graphe $G = (V, E, w)$ est dit pondéré si la fonction w associe à chaque arête $e \in E$ un réel $w(e)$. On appelle $w(e)$ le poids de l'arête e.*

Si la fonction de pondération w n'est pas explicitement notée on considérera que le graphe est non-pondéré.

Définition 2.14 ***(Boucle)*** *Soit $G = (V, E)$ un graphe et $e \in E$ une arête avec pour extrémités $u, v \in V$. L'arête e est une boucle si et seulement si $u = v$.*

On supposera qu'un graphe ne contient pas de boucles si la mention "avec boucles" n'est pas explicitement donnée.

Définition 2.15 ***(Sous-graphe)*** *Soit $G = (V, E)$ un graphe. On dit que $G' = (V', E')$ est un sous-graphe de G si et seulement si $V' \subseteq V$ et $E' \subseteq E$.*

Définition 2.16 ***(Sous-graphe induit)*** *Soit $G = (V, E)$ un graphe et un sous-ensemble de sommets $S \subseteq V$. Le sous-graphe noté $G[S] = (S, E')$ est induit par S si et seulement si $E' = \{(u, v) \in E, u \in S, v \in S\}$.*

Définition 2.17 ***(Sous-graphe formé)*** *Soit $G = (V, E)$ un graphe et un sous-ensemble d'arêtes $E' \subseteq E$. Le sous-graphe noté $G(E') = (S, E')$ est formé par E' si et seulement si $S = \{u \in V, (u, v) \in E'\}$.*

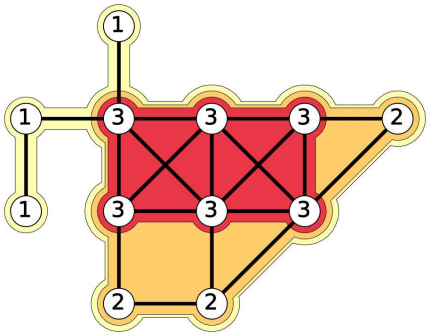


FIGURE 2.1: Illustration de la *Coreness* : le label des sommets correspond à la valeur de *coreness*. Les enveloppes correspondent au trois différents k-cores du graphe (1-core : jaune, 2-core : orange, 3-core : rouge).

Définition 2.18 ***(Graphe biparti)*** *Un graphe $G = (V, E)$ est dit biparti si et seulement si il existe une partition (V_1, V_2) de V telle que toute arête $e \in E$ a une extrémité dans V_1 et l'autre dans V_2.*

Un graphe biparti sera noté sous la forme $G = (V_1 + V_2, E)$. Notons que les sous-graphes induits $G[V_1]$ et $G[V_2]$ ne contiennent pas d'arêtes.

Définition 2.19 ***(Voisinage d'un sommet)*** *Soit $G = (V, E)$ un graphe et $u \in V$ un sommet. On appelle voisinage de u dans G, noté $N_G(u)$ l'ensemble $\{v \in V, (u, v) \in E\}$.*

Définition 2.20 ***(Degré d'un sommet)*** *Soit $G = (V, E)$ un graphe et $u \in V$ un sommet. On appelle degré de u dans G, noté $d_G(u)$ la quantité $|N_G(u)|$.*

Définition 2.21 ***(Voisinage entrant et sortant d'un sommet)*** *Soit $G = (V, A)$ un graphe orienté et $u \in V$ un sommet. On appelle voisinage entrant (respectivement sortant) de u dans G, noté $N_G^-(u)$ (resp. $N_G^+(u)$) l'ensemble $\{v \in V | (v, u) \in A\}$ (resp. $\{v \in V | (u, v) \in A\}$).*

Définition 2.22 ***(Degré entrant et sortant d'un sommet)*** *Soit $G = (V, A)$ un graphe orienté et $u \in V$ un sommet. On appelle degré entrant (respectivement sortant) de u dans G, noté $d_G^-(u)$ (resp. $d_G^+(u)$) la quantité $|N_G^-(u)|$ (resp. $|N_G^+(u)|$).*

Définition 2.23 ***(Coreness)*** *Soit un graphe $G = (V, E)$, la coreness d'un sommet $u \in V$, notée $core_G(u)$, est la plus grande valeur de $i \in [0, d_G(u)]$ pour laquelle il existe un sous-graphe G' de degré minimum i contenant u.*

Définition 2.24 ***(k-core)*** *Soit un graphe $G = (V, E)$, le k-core de G est le sous-graphe induit par l'ensemble $\{v \in V, core_G(v) \geq k\}$.*

On appelle la *dégénérescence* de G la valeur maximum k pour laquelle le k-core de G est non-vide. Une illustration du concept de k-core est donnée en Figure 2.1.

Définition 2.25 ***(Chemin orienté)*** *Soit $G = (V, A)$ un graphe orienté. On appelle chemin orienté dans G une séquence $(v_1, e_1, v_2, e_2, \ldots, e_{l-1}, v_l)$ avec :*

- $\forall i \in [1..l], v_i \in V$
- $\forall i \in [1..l-1], e_i = \{v_i, v_{i+1}\} \in A$

– $\forall i, j \in [1..l], i \neq j, v_i \neq v_j$ *et* $e_i \neq e_j$.

La *longueur* d'un chemin désigne le nombre d'arêtes de la séquence. Dans le cas de graphes pondérés, la longueur peut désigner la somme des poids des arêtes de la séquence, on parlera alors de *longueur pondérée.*

Définition 2.26 ***(Chemin non-orienté)*** *Soit $G = (V, E)$ un graphe. On appelle chemin non-orienté dans G une séquence $(v_1, e_1, v_2, e_2, \ldots, e_{l-1}, v_l)$ avec :*
– $\forall i \in [1..l], v_i \in V$
– $\forall i \in [1..l-1], e_i = \{v_i, v_{i+1}\} \in E$
– $\forall i, j \in [1..l[, i \neq j, v_i \neq v_j$ *et*
– $\forall i, j \in [1..l], i \neq j, e_i \neq e_j$.

Un chemin ne passant pas deux fois par une même arête est dit *simple.* On dira qu'un sommet u est accessible à partir d'un autre sommet v si il existe un chemin (orienté ou non selon le cas) entre v et u.

Définition 2.27 ***(Distance)*** *Soit $G = (V, E)$ un graphe et deux sommets distincts $s, t \in V$, la distance entre s et t notée $\delta_G(s, t)$ correspond à la longueur d'un plus court chemin entre s et t dans G.*

Définition 2.28 ***(Centralité-chemin [57])*** *Soit $G = (V, E)$ un graphe et un sommet $u \in V$, la* centralité-chemin *est le nombre de plus courts chemins passant par u pour tout couple de sommets (s, t) différents de u.*

La centralité-chemin peut être calculée en $\mathcal{O}(|E||V|)$ [26].

Définition 2.29 ***(Diamètre)*** *Soit $G = (V, E)$ un graphe. Le diamètre de G est égal à la distance maximum entre deux sommets de G.*

Définition 2.30 ***(Cycle et Graphe acyclique)*** *Soit $G = (V, E)$ un graphe orienté (respectivement non-orienté) et $u \in V$ un sommet. On appelle cycle tout chemin orienté (resp. non-orienté simple) de u à u. Si il n'existe pas de tel chemin dans G alors le graphe G est un graphe orienté (resp. non-orienté) acyclique.*

Définition 2.31 ***(Graphe connexe)*** *Soit $G = (V, E)$ un graphe. Le graphe G est connexe s'il existe un chemin entre toute paire de sommets dans G.*

L'ensemble des sous-graphes connexes maximaux de G sont appelés les composantes connexes de G.

Définition 2.32 ***(k-sommet-connexe)*** *Soit $G = (V, E)$ un graphe. Le graphe G est k-sommet-connexe si et seulement si $|V| > k$ et qu'il n'existe pas de sous-ensemble $V' \subset V$ de taille $|V'| < k$ tel que le sous-graphe induit $G[V \setminus V']$ n'est pas connexe.*

La connexité-sommet d'un graphe G est le k maximum tel que G est k-sommet-connexe.

Définition 2.33 ***(k-arête-connexe)*** *Soit $G = (V, E)$ un graphe. Le graphe G est k-arête-connexe si et seulement si il n'existe pas de sous-ensemble $E' \subset E$ de taille $|E'| < k$ tel que le sous-graphe $G' = (V, E \setminus E')$ n'est pas connexe.*

La connexité-arête d'un graphe G est le k maximum tel que G est k-arête-connexe.

Définition 2.34 ***(Graphe complet)*** *Soit $G = (V, E)$ un graphe. Le graphe G est un graphe complet si et seulement si $\forall u \in V$ et $\forall v \in V$, $(u, v) \in E$.*

Un graphe complet possède $m = \binom{n}{2} = \frac{n(n-1)}{2}$ arêtes. Un graphe orienté complet possède $m = n^2$ arcs.

Définition 2.35 ***(Densité)*** *Soit $G = (V, E)$ un graphe. La densité de G correspond à la proportion d'arêtes sur le nombre maximal d'arêtes dans un graphe de même type possédant $|V|$ sommets.*

Définition 2.36 ***(Matrice d'adjacence)*** *Soit G un graphe à n sommets. La matrice d'adjacence A de G est une matrice de taille $n \times n$ dont les éléments a_{uv} correspondent au nombre d'arêtes ayant pour extrémités les sommets d'indices u et v.*

Dans le cas non-orienté, A est une matrice symétrique. Dans le cas où G est sans-boucle, la diagonale de A est nulle.

2.3 Arbres

Définition 2.37 ***(Arbre)*** *On dit que T est un arbre si et seulement si T est un graphe acyclique et connexe.*

De manière à éviter de possibles ambiguïtés, les sommets d'un arbre seront appelés *nœuds* si T est clairement défini comme étant un arbre. Les nœuds $u \in V$ tels que $d_T(v) = 1$ sont appelés feuilles de l'arbre. L'ensemble des feuilles est noté $\mathcal{F}_T$. Si un nœud u de T n'est pas une feuille on dit que u est interne à T.

Définition 2.38 ***(Arbre enraciné)*** *Soit $T = (V, E)$ un graphe orienté. On dit que T est un arbre enraciné en un nœud $r \in V$ (appelée la* racine*) si et seulement si il existe un unique chemin orienté de r à tout autre nœud de G.*

Dans un arbre enraciné, le voisin entrant $u \neq r$ est appelé son parent (noté $p_T(u)$). Pour un nœud $u \in V$ l'ensemble de ses voisins sortants forme les successeurs de u (noté $\sigma_T(u)$). Dans le cas d'un arbre enraciné T, les feuilles de l'arbre vérifient $d_T^+(v) = 0$ et donc $\sigma_T(v) = \emptyset$.

Définition 2.39 ***(Arbre binaire)*** *Soit $T = (V, A)$ un arbre enraciné. On dit que T est un arbre binaire si et seulement si $d_T^+(u) = 2$ pour tout $u \in (V \setminus \mathcal{F}_T)$.*

Définition 2.40 ***(Sous-Arbre)*** *Soit $T = (V, E)$ un arbre, tout sous-graphe connexe de T est appelé un sous-arbre de T.*

Pour un graphe enraciné T, le sous-arbre enraciné en $u \in V$ est noté T_u et correspond au sous-graphe induit formé par u et l'ensemble des nœuds $v \in V$ accessibles à partir de u.

Définition 2.41 ***(Hauteur)*** *Soit $T = (V, E)$ un arbre enraciné en r, la hauteur d'un nœud $u \in V$ notée $h_T(u)$ correspond à la distance entre r et u.*

Définition 2.42 ***(Niveau)*** *Soit $T = (V, E)$ un arbre enraciné en r, le i-ème niveau de G noté $N_i(T)$ est l'ensemble de feuilles dans le sous-arbre induit par l'ensemble $\{u \in V, h_G(u) \leq i\}$.*

Définition 2.43 ***(Arbre de partition)*** *Soit un ensemble S, un arbre de partition T correspond à un couple (T, f) où T est un arbre enraciné en r et f est une application injective définie dans $V(T)$ et à valeur dans $\mathcal{P}(S)$ telle que*

- $f(r) = S$
- $f(u) = \bigcup_{v \in \sigma_T(v)} f(v)$ *pour tout nœud interne u.*
- $f(u) \cap f(v) = \emptyset$ *si* $p_T(u) = p_T(v)$

Par souci de clarté, on omettra l'application f et on notera $f(u) = S_u$. Toutes les notations relatives aux arbres enracinés pourront être appliquées directement à T. Un arbre de partition binaire est aussi appelé un *dendrogramme*. Un arbre de partition T d'un ensemble S permet d'encoder une partition hiérarchique de S puisqu'il existe une application bijective entre les deux. En particulier, notons que l'ensemble formé par $\{S_u, u \in N_i(T)\}$ est une partition de S.

Définition 2.44 ***(Arbre de partition canonique)*** *Soit un ensemble S, un arbre de partition T est canonique si et seulement si il ne contient pas de nœud u tel que $d_T^+(u) = 1$.*

La *réduction canonique* d'un arbre de partition T est un arbre de partition canonique T' obtenu en supprimant itérativement chaque nœud u de degré sortant 1 dans T et ajoutant une arête allant de $p_T(u)$ à l'unique successeur de u. Le résultat T' est un arbre de partition canonique de S. Tout arbre de partition admet une seule et unique réduction canonique. Par la suite, on supposera qu'un arbre de partition (on parlera aussi de *hiérarchie*) est toujours canonique sauf mention contraire.

2.4 Partitionnement de graphes

On présente ici différents concepts relatifs au partitionnement de graphe. Un partitionnement de G correspond à une partition des sommets de G. On parlera également de partitionnement des arêtes d'un graphe.

Définition 2.45 ***(Clique)*** *Soit $G = (V, E)$ un graphe et V' un sous-ensemble de sommets de G. Les sommets de V' forment une clique si et seulement si le sous-graphe induit $G[V']$ est un graphe complet.*

Définition 2.46 ***(Clique Maximale)*** *Soit $G = (V, E)$ un graphe et V' un sous-ensemble de sommets de G. Les sommets de V' forment une clique maximale si et seulement si V' est une clique et qu'il n'existe pas de clique $V'' \subseteq V$ tel que $V' \subset V''$.*

Une clique V' est dite maximum si $|V'|$ est maximum parmi l'ensemble des cliques de G. Déterminer la taille de la clique maximum est un problème NP-Complet [35].

Définition 2.47 ***(Arêtes internes/externes)*** *Soit $G = (V, E)$ un graphe et V' un sous-ensemble de sommets de G. L'arête $(u, v) \in E$ est dite* interne *si et seulement si $u \in V'$ et $v \in V'$. Soit V_1, V_2 deux sous-ensembles disjoints de V, l'arête $(u, v) \in E$ est dite* externe *si et seulement si $u \in V_1$ et $v \in V_2$.*

Pour une partition de $C = (C_1, \ldots, C_k)$, l'ensemble E_{ij} désigne l'ensemble des arêtes externes entre C_i et C_j. Ainsi E_{ii} est l'ensemble des arêtes internes à C_i. On note également e_i (respectivement e_{ij}) le nombre d'arêtes internes au groupe C_i (resp. externes entre les groupes C_i et C_j pour $i \neq j$). Si G est considéré pondéré, la quantité e_i (resp. e_{ij}) est la somme des poids des arêtes internes au groupe C_i (resp. externes entre les deux groupes).

Définition 2.48 ***(Coupe minimum)*** *Soit $G = (V, E)$ un graphe, une partition $(V', V - V')$ de taille 2 est appelée une coupe de G. La taille d'une coupe $(V', V - V')$ notée $|(V', V - V')|$ correspond au nombre d'arêtes externes entre les deux groupes.*

Si G est pondéré, la taille de la coupe est la somme des poids des arêtes traversant la coupe. Une coupe est minimum si et seulement si $|(V', V - V')|$ est minimum parmi l'ensemble des coupes de G. D'après le théorème de Menger [151], la connexité-arête d'un graphe G est égale à la taille d'une coupe minimum de G.

Définition 2.49 ***(Graphe quotient)*** *Soit un graphe G et une partition $\mathcal{C} = (C_1, \ldots, C_k)$ des sommets V. Le graphe quotient $Q = (V, V \times V, f)$ est un graphe dont les sommets $(v_1, \ldots, v_k)$ correspondent, via l'application bijective f, aux groupes $(C_1, \ldots, C_k)$.*

Le graphe Q peut être pondéré par la quantité e_{ij} correspondante et peut comprendre des boucles pondérées par la quantité e_i.

Définition 2.50 ***(Mesure de qualité)*** *Une mesure de qualité $\Phi(G, \mathcal{C})$ est une fonction définie sur l'ensemble des partitions $\mathcal{C}$ des sommets V d'un graphe G et à valeur dans l'ensemble des réels.*

Pour deux partitions $\mathcal{C}, \mathcal{C}'$ de V on dit que $\mathcal{C}$ est *meilleure* que $\mathcal{C}'$ si $\Phi(G, \mathcal{C}) > \Phi(G, \mathcal{C}')$. Pour un seuil τ fixé, on dira que $\mathcal{C}$ est une *bonne* partition si $\Phi(G, \mathcal{C}) > \tau$.

Définition 2.51 ***(Mesure de qualité additive)*** *Une mesure de qualité $\Phi(G, \mathcal{C})$ d'une partition $\mathcal{C} = (C_1, \ldots, C_k)$ des sommets V d'un graphe G est dite* additive *si elle peut être formulée sous la forme*

$$\Phi(G, \mathcal{C}) = \sum_{i=1}^{k} \phi(G, \mathcal{C}, C_i) \tag{3}$$

où $\phi(G, \mathcal{C}, C_i)$ correspond au gain *du groupe C_i au sein de la partition $\mathcal{C}$.*

Définition 2.52 ***(Mesure de qualité fortement additive)*** *Une mesure de qualité $\Phi(G, \mathcal{C})$ d'une partition $\mathcal{C} = (C_1, \ldots, C_k)$ des sommets V d'un graphe G est dite* fortement additive *si elle est additive et que pour tout $V' \subset V$, on a $\phi(G, \mathcal{C}(V'), V') = \phi(G, V')$ pour toute partition $\mathcal{C}(V')$ de V contenant V'.*

Le fait qu'une mesure soit fortement additive indique que le gain d'un groupe V' ne dépend pas de la façon dont les sommets appartenant à $(V \setminus V')$ sont partitionnés. Ainsi, si Φ est fortement additive, pour deux partitions $\mathcal{C}_1, \mathcal{C}_2$ de $V(G)$, on a

$$\Phi(G, \mathcal{C}_1) - \Phi(G, \mathcal{C}_2) = \sum_{C \in (\mathcal{C}_1 - \mathcal{C}_2)} \phi(G, C) - \sum_{C \in (\mathcal{C}_2 - \mathcal{C}_1)} \phi(G, C) \tag{4}$$

puisque les contributions des groupes $\mathcal{C}_1 \cap \mathcal{C}_2$ s'annulent. Disposer d'une mesure fortement additive est donc utile dans le cadre d'algorithme de partitionnement. Par exemple, pour une partition $\mathcal{C}$ donnée, la variation de qualité obtenue en regroupant les sommets de deux groupes $C_1, C_2 \in \mathcal{C}$ est égale à $\phi(G, C_1 \cup C_2) - (\phi(G, C_1) + \phi(G, C_2))$.

2.5 Probabilité et statistiques

Définition 2.53 ***(Entropie)*** *Soit X une variable aléatoire discrète à l états $(x_1, \ldots, x_l)$, on note $p_i = P(X = x_i)$ pour $i \in [1, l]$ la probabilité que X soit dans l'état x_i. L'entropie de X, notée $H(X)$, est donnée par la formule suivante :*

$$H(X) = -\sum_{i=1}^{l} p_i \log(p_i) \tag{5}$$

Définition 2.54 ***(Vraisemblance)*** *Soit $S = (s_1, \ldots, s_N)$ un N-échantillon d'une variable aléatoire X, la vraisemblance notée $\mathcal{L}(S, P)$ est la probabilité d'observer le N-échantillon S partant d'une distribution P de la variable X :*

$$\mathcal{L}(S, P) = \prod_{i=1}^{N} P(X = s_i) \tag{6}$$

La vraisemblance peut être utilisée pour estimer les éventuels paramètres d'une distribution P_X, la méthode est alors appelée *maximum de vraisemblance* car on va chercher le jeu de paramètres maximisant la vraisemblance ou, de façon équivalente, le logarithme de la vraisemblance.

Définition 2.55 ***(Loi de Bernoulli)*** *Soit X une variable aléatoire discrète à valeur dans l'ensemble $(0, 1)$. On dit que la variable X suit une loi de Bernoulli de paramètre p ou $X \sim \mathcal{B}(p)$ si et seulement si*

$$P(X = x) = \begin{cases} p & si\ x = 1 \\ 1 - p & si\ x = 0 \end{cases} \tag{7}$$

Définition 2.56 ***(Loi binomiale)*** *Soit X une variable aléatoire discrète à valeur dans l'ensemble $(1, \ldots, n)$. On dit que la variable X suit une loi binomiale de paramètres (n, p) ou $X \sim \mathcal{B}(n, p)$ si et seulement si X s'exprime sous la forme d'une somme de variables aléatoires $(X_1, \ldots, X_n)$ où $X_i \sim \mathcal{B}(p)$, pour tout $i \in [1, n]$.*

Définition 2.57 ***(Loi de puissance)*** *Soit X une variable aléatoire réelle. On dit que la variable X suit une loi de puissance de paramètres (x_{min}, α) ou $X \sim \mathcal{PL}(n,p)$ si et seulement si*

$$P(X \leq x) = \int_{x_{min}}^{x} cy^{-\alpha}dy \tag{8}$$

où c est une constante de normalisation.

2.6 Modèles de graphes aléatoires

Un graphe peut être considéré comme une réalisation d'un processus stochastique complexe *a priori* inconnu. Dans ce cadre, il est possible d'inférer des modèles paramétrés en utilisant des observations issues des réseaux réels. La *modélisation de graphes* permet ainsi de mieux comprendre la topologie et la dynamique d'un réseau. L'objet de cette section est de décrire différents modèles plus ou moins complexes qui seront utilisés dans le reste de ce document.

Pour un graphe $G = (V, E)$ on suppose ici que l'ensemble E des arêtes du graphe est sélectionné aléatoirement parmi tous les sous-ensembles de $V \times V$. Différents modèles existent à ce jour, nous allons en détailler quelques-uns ici. Pour cela, nous introduisons différentes notions relatives aux graphes aléatoires.

Définition 2.58 ***(Probabilité de lien)*** *Soit $G = (V, E)$ un graphe et deux sommets $u, v \in V$. On note p_{uv} la probabilité qu'une arête ayant pour extrémités u et v existe dans E.*

On a $p_{uv} = P(\{u, v\} \in E)$. Dans ce cadre, $(\{u, v\} \in E) \sim \mathcal{B}(p_{uv})$. Dans le cas non-orienté, on a $p_{uv} = p_{vu}$. Dans le cas où G est sans boucle, on a $p_{uu} = 0$, $\forall u \in V$.

Définition 2.59 ***(Probabilité de lien interne/externe)*** *Soit $G = (V, E)$ un graphe et une partition $\mathcal{C}$ de G. On note p_{ij} la proportion théorique d'arêtes reliant les sommets de C_i et les sommets de C_j. Dans ce cas p_{ii} est la proportion théorique de liens internes à C_i.*

Dans ce cadre, on suppose donc que $e_{ij} \sim \mathcal{B}(|C_i||C_j|, p_{ij})$.

2.6.1 Modèle de Erdős-Rényi

Ce modèle fut proposé par Erdős et Rényi [48], il est à la base de la théorie des graphes aléatoires. La probabilité p_{uv} est fixée pour toute paire de sommets appartenant à $V \times V$, on notera cette probabilité p. La génération consiste donc à poser $(u, v) \in E$ avec une probabilité p ainsi on a $m \sim \mathcal{B}(\binom{n}{2}, p)$ dans le cas de graphes non-orientés et sans boucles.

Ce modèle génère l'ensemble des graphes de taille n avec une probabilité non-nulle. Toutefois, il est rarement adapté à la modélisation de réseaux réels. En effet, l'observation de la présence ou l'absence d'une arête entre deux sommets est indépendante du reste du réseau, ce qui est rare en pratique.

L'estimation du modèle repose sur l'estimation de p, un estimateur direct est donné par la densité d'arête dans G. Cet estimateur est celui maximisant la vraisemblance du modèle.

2.6.2 Modèle ERMG

Une généralisation du modèle de Erdős-Rényi est proposée en introduisant la notion de partitionnement : les propriétés de connectivité entre les sommets dépendent des groupes auxquels ces sommets appartiennent. Ce modèle, nommé *Erdős-Rényi mixture for graph* [38] (ERMG), permet ainsi la modélisation de structures variées telles que les graphes bipartis ou les réseaux en étoiles.

Formellement, dans ce contexte, la probabilité de l'événement :

$$((u,v) \in E) | \{u \in C_i, v \in C_j\} \sim \mathcal{B}(p_{ij}) \tag{9}$$

Le statut de deux arêtes (absentes ou présentes) est donc conditionnellement indépendant. La simulation de graphes aléatoires n'est pas plus compliquée que dans le modèle de Erdős-Rényi si on dispose de la partition $\mathcal{C}$ et des valeurs des p_{ij}. Ce modèle, ainsi que le précédent, ont l'avantage de pouvoir prédire les liens pour de nouvelles entrées dans le système (de nouveaux sommets).

L'estimation du modèle consiste à déterminer à la fois la partition et les probabilités de liens internes et externes maximisant la vraisemblance du modèle (voir définition 2.54). On note $\mathcal{P} = \{p_{ij}\}_{1 \leq i,j \leq k}$ l'ensemble de ces probabilités pour une partition $\mathcal{C}$ avec k groupes. La vraisemblance du modèle est donc la probabilité d'observer $E(G)$ en fonction du couple $(\mathcal{C}, \mathcal{P})$.

$$\mathcal{L}(E(G), (\mathcal{C}, \mathcal{P})) \quad = \quad \prod_{i=1}^{k} \prod_{j=i}^{k} p_{ij}^{e_{ij}} (1 - p_{ij})^{t(C_i, C_j) - e_{ij}} \tag{10}$$

où $t(C_i, C_j) = \begin{cases} \binom{n_i}{2} & \text{si } i = j \\ n_i n_j & \text{sinon} \end{cases}$.

2.6.3 Modèle à distribution arbitraire de degré

Ce modèle est également une généralisation du modèle de Erdős-Rényi [5] : on considère que le degré des sommets est connu. Un graphe aléatoire serait dans ce cas obtenu en sélectionnant un graphe disposant du même nombre de sommets et de la même séquence de degré. Il permet également de modéliser un graphe dont les degrés sont simulés à partir d'une loi de probabilité connue.

Cette approche est issue de l'étude des réseaux à invariance d'échelle [16], où la loi de probabilité des degrés des sommets peut être approchée par une loi de puissance (voir Définition 2.57). La génération aléatoire repose, dans ce cadre, sur l'ajout successif de sommets qui se lient préférentiellement aux sommets les plus anciens.

De nombreux algorithmes furent développés dans le cas général, lorsque que la séquence de degrés ou la loi de probabilité sont connues [98]. Les plus récents se basent sur des algorithmes de Monte-Carlo par Chaîne de Markov (MCMC) [121] ou sur des algorithmes d'échantillonnage pondéré séquentiel (*Sequential Importance Sampling* [22]). Dans chacun de ces algorithmes, une arête est ajoutée entre deux sommets si le résultat de cette opération satisfait certains critères. Cependant, la façon de choisir un couple de sommets plutôt qu'un autre peut varier d'un algorithme à un autre. En général, dans la littérature, cette probabilité dépend du degré des deux sommets. De même que pour le modèle de Erdős-Rényi, de nombreuses propriétés statistiques ont été étudiées dans le cadre de ce modèle [5, 75].

2.6.4 Modèle hiérarchique

Une approche proposée par Clauset *et al.* [31] repose sur l'hypothèse de l'existence d'une structure hiérarchique inhérente à certains types de réseaux et, à l'instar du modèle ERMG, de l'existence de communautés plus ou moins liées entres elles.

Soit G un graphe de n sommets et D un dendrogramme dont les n feuilles correspondent aux sommets de G. A chacune des $n-1$ séparations de D (indexé de façon quelconque par un entier noté r) on assigne une probabilité notée p_r. Par ailleurs, on note r_{uv} le plus proche ancêtre commun (*least common ancestor*) dans D de $u, v \in V$. Ainsi deux sommets $\{u, v\}$ sont reliés par une arête avec une probabilité $p_{r_{uv}}$. Le modèle de *graphes aléatoires hiérarchiques* (noté HRG) correspond au couple formé par $(D, \{p_r\}_{r \in D})$.

Pour déterminer quel couple $(D, \{p_r\}_{r\in D})$ forme le "meilleur" modèle pour un graphe donné, les auteurs proposent d'utiliser des méthodes stochastiques de maximisation de la vraisemblance. On pose L_r (respectivement R_r) le nombre de feuilles à gauche (resp. à droite) de la séparation r et E_r le nombre d'arêtes dans G dont les extrémités ont pour plus proche ancêtre commun r. Pour un couple $(D, \{p_r\}_{r\in D})$, la vraisemblance du modèle est donnée par la formule suivante.

$$\mathcal{L}(E(G), (D, \{p_r\}_{r\in D})) \quad = \quad \prod_{r\in D} p_r^{E_r}(1-p_r)^{L_r R_r - E_r} \tag{11}$$

Les p_r maximisant $\mathcal{L}(E(G), (D, \{p_r\}_{r\in D}))$ notés $\widehat{p_r}$ correspondent à la proportion observée d'arêtes passant dans r :

$$\widehat{p_r} = \frac{E_r}{L_r R_r} \tag{12}$$

où $L_r R_r$ est égale au nombre total d'arêtes susceptibles de traverser la séparation r.
Pour estimer le modèle, il reste donc à déterminer une configuration D notée $\widehat{D}$ maximisant le logarithme de la vraisemblance après remplacement des p_r :

$$log(\mathcal{L}(E(G), D)) \quad = \quad \sum_{r\in D} L_r R_r(\widehat{p_r}\log(\widehat{p_r}) + (1-\widehat{p_r})\log(1-\widehat{p_r})) \tag{13}$$

La formule $h(p) = -p\log(p) - (1-p)\log(1-p)$ est la fonction d'entropie de Shannon (voir définition 2.53). Cette quantité sera faible dans des cas où les séparations correspondent soit à des probabilités faibles, soit à des probabilités fortes, c'est-à-dire dans le cas où ces dernières apportent peu d'information. Par convention, on pose $h(0) = h(1) = 0$ ce qui correspond aux limites de l'entropie. Le dendrogramme $\widehat{D}$ peut être déterminé par des algorithmes de type Monte-Carlo par Chaîne de Markov (MCMC) [121] qui offrent une convergence relativement rapide. Il faut cependant noter que rien n'indique l'existence d'un maximum unique.

Ce modèle, de par sa flexibilité, a de nombreux avantages.

- Il permet, à l'instar des précédents, de simuler de nouvelles instances de graphe possédant une structure proche [31].
- On peut obtenir une partition des sommets de G par élagage de $\widehat{D}$ et prendre les nouvelles feuilles obtenues pour partitionner le graphe. De ce point de vue, le modèle HRG généralise le modèle ERMG.

– L'existence d'une structure hiérarchique implique une notion de préférence entre les groupes et donc entre les sommets pris deux à deux. Il est ainsi possible de prédire l'absence de lien entre des sommets si, par exemple, on est dans le cas où on ne dispose que d'un graphe partiel [32], c'est-à-dire un échantillon de l'ensemble E.

2.6.5 Modèle LFR

Nous présentons maintenant le modèle LFR proposé par [88] permettant de générer des graphes aléatoires à partir d'une partition des sommets. Nous présentons également une extension de ce modèle, nommée LFR hiérarchique, aux partitions hiérarchiques [89]. Ces modèles ont été conçus de façon à pouvoir générer des graphes et tester différents algorithmes de partitionnement. L'optique n'est donc pas directement l'inférence de graphes.

Modèle LFR La construction de graphes en utilisant le modèle LFR correspond aux étapes suivantes :

1. Génération d'une séquence de degré suivant une loi de puissance $\mathcal{PL}(d_{min}, \alpha)$ où α est donné en paramètre et où d_{min} (le degré minimum) est calculé en fonction du nombre de sommets n et du degré moyen d donnés en paramètres. En utilisant ces paramètres, une séquence $(\lambda_1, \ldots, \lambda_n) \in [0,1]^n$ est générée. La génération du graphe G à partir de cette séquence repose sur la méthode de Molloy-Reed [98]. Ainsi, pour tout sommet v_i de G on peut garantir que $d_G(v_i) \simeq n\lambda_i$.
2. Génération du nombre et de la taille des groupes. La taille des groupes suit également une loi de puissance $\mathcal{PL}(s_{min}, \beta)$ où s_{min} et β sont deux paramètres du modèle.
3. Affectation aléatoire des sommets à chaque groupe en respectant la contrainte qu'un sommet u ne peut être assigné qu'à un groupe de taille supérieure ou égale à $d_G(u)$.
4. Routage des arêtes adjacentes à un sommet u de façon à garantir qu'environ $\mu d_G(u)$ arêtes aient leur opposé dans un groupe différent du groupe de u. Cette proportion μ est un paramètre du modèle, il va correspondre, au final, à la proportion d'arêtes externes dans le graphe.

Le graphe est donc généré à partir d'une partition, générée elle aussi de façon aléatoire. A la différence du modèle ERMG, les sommets d'un même groupe ne sont pas tous similaires puisque selon la valeur des paramètres, le degré des sommets au sein d'un groupe peut significativement varier. On peut cependant regretter l'absence d'attachement préférentiel entre les différents groupes à la différence du modèle ERMG ou HRG. La version hiérarchique de ce modèle, présentée ci-dessous, permet de combler partiellement cette lacune.

Modèle LFR Hiérarchique Ce modèle se rapproche beaucoup du modèle LFR mais propose que le graphe soit généré à partir d'une partition à deux niveaux. Le premier

niveau est celui des *macro-communautés*, il est généré de la même façon que la partition du modèle LFR où le paramètre μ_1 définit la proportion d'arêtes entre macro-communautés. Chaque macro-communauté C est ensuite découpée en différentes *micro-communautés*, ce découpage se fait de la même façon que la partition du modèle LFR mais cette fois en s'appliquant sur le sous-graphe induit $G[C]$. Le paramètre μ_2 définit la proportion d'arêtes entre les micro-communautés faisant partie d'une même macro-communauté. Pour tout sommet u, la proportion d'arêtes adjacentes à u externe à une micro-communauté est donc $\mu_1 + \mu_2$ en moyenne. Un exemple de graphe généré à partir de ce modèle est donné en Figure 2.2.

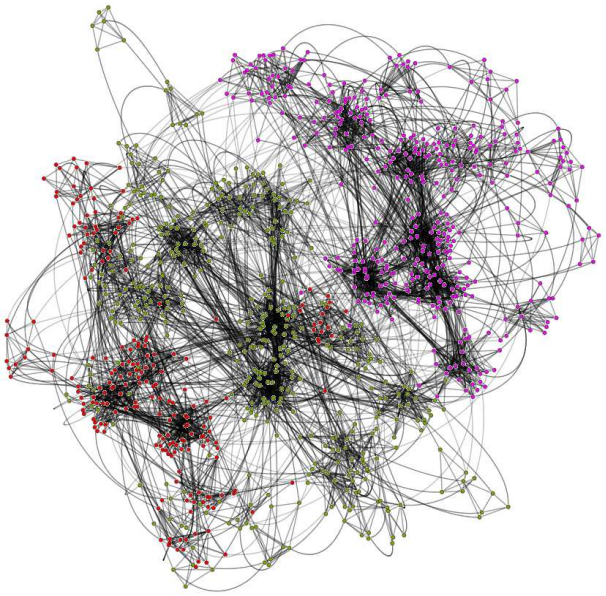


FIGURE 2.2: Exemple d'un graphe généré en utilisant le modèle LFR hiérarchique.

Chapitre 3

Mesures de qualité pour le partitionnement de graphes

Une mesure de qualité peut être utilisée pour comparer différentes partitions et choisir la meilleure. En effet, les mesures de qualité associent à un objet mathématique *a priori* dépourvu d'une relation d'ordre, une valeur numérique standardisée. Ainsi, ces mesures sont utilisées comme fonction objective à maximiser dans le cadre d'algorithmes de partitionnement. Malgré l'utilisation de ce type d'outil, cette tâche correspond à des problèmes difficiles. En effet, le nombre de partitions d'un ensemble de n éléments est le n-ième nombre de Bell [77]

$$\begin{aligned} B_0 &= 1 \\ B_n &= \sum_{k=0}^{n-1} \binom{n}{k} B_k \end{aligned} \tag{1}$$

Cela explique l'emploi de mesures de qualité permettant d'explorer plus facilement cet espace.

Les différentes mesures de qualité existantes à ce jour se basent sur des critères différents. Comparer les mesures par rapport aux critères revient à comparer plusieurs définitions de ce qu'est une "bonne" partition ou ce qui définit une communauté dans un réseau. Ces critères dépendent du domaine d'application du partitionnement de graphe.

On se place ici dans un cadre d'exploration où il n'y a pas d'*a priori* sur la forme, la taille des groupes ni sur leur nombre. Dans ce cadre, étudier le comportement des mesures de qualité est important. C'est l'objet de cette section où nous nous intéressons à l'analyse de la mesure MQ proposée par Mancoridis [93]. En particulier, nous comparons le comportement de cette mesure à celui de la modularité [101].

Après avoir présenté les mesures de qualité utilisées dans la section 3.1, nous proposons une généralisation de la mesure MQ dans la section 3.2. Cette mesure est analysée et comparée à la modularité Q dans la section 3.3.

3.1 État de l'art

Nous présentons ici différentes mesures de qualité pour une partition des sommets d'un graphe. Un grand nombre de mesures existantes sont présentées dans [25]. Nous détaillons ici les mesures qui seront utilisées dans la suite de ce manuscrit. Nous nous concentrons sur les mesures additives utilisant le nombre d'arêtes internes et/ou le nombre d'arêtes externes. Il s'agit de la *modularité* Q [101], la *mesure de de Mancoridis* MQ [93] et la *qualité de compression* L [126].

Notons qu'il existe cependant certaines mesures se basant sur différents critères. Par exemple White et Harary [152] proposent d'évaluer la cohésion d'un groupe comme le nombre minimum d'acteurs à retirer du réseau pour déconnecter le groupe. Ceci correspond à la connexité-sommet du sous-graphe induit par ce groupe (voir définition 2.32). Ils suggèrent par la suite que les groupes peuvent être évalués en se basant sur le nombre d'arêtes supplémentaires par rapport au nombre minimum d'arêtes nécessaire pour atteindre cette connexité. Une autre mesure introduite par [153] se base, en partie, sur la longueur moyenne des plus courts chemins au sein des groupes pour évaluer leur qualité.

3.1.1 Modularité Q

La mesure de modularité Q est très populaire dans le domaine du partitionnement de graphes [101, 100]. Elle repose sur la différence entre la part d'arêtes observées et la part d'arêtes théorique dans l'hypothèse où les sommets ne sont pas liés de manière préférentielle si ils appartiennent au même groupe.

Définition 3.1 *Soit $G = (V, E)$ un graphe et $\mathcal{C} = (C_1, \ldots, C_k)$ une partition de ces sommets. La* modularité *de la partition $\mathcal{C}$, notée $Q(G, \mathcal{C})$ est donnée par la formule suivante :*

$$Q(G,\mathcal{C}) \quad = \quad \sum_{i=1}^{k} \left(\frac{e_{ii}}{|E|} - \left(\frac{d_i}{2|E|} \right)^2 \right) \tag{2}$$

où $d_i = 2e_{ii} + \sum_{j \neq i} e_{ij}$ est la somme des degrés des sommets appartenant au groupe C_i.

La modularité telle que donnée par l'équation (2) est une mesure fortement additive (voir définition 2.52) à valeur dans $[-\frac{1}{2}, 1]$. En effet, pour chaque groupe, on calcule la différence entre la proportion observée d'arêtes internes au groupe et la proportion théorique d'arêtes internes dans le cas d'un graphe aléatoire avec la même distribution des degrés (voir section 2.6.3). La proportion d'arêtes incidente à un groupe C_i dans G est $\frac{d_i}{2|E|}$. C'est la probabilité en prenant une arête quelconque de G d'observer qu'au moins une de ses extrémités est dans C_i. Si on considère qu'il n'y a pas de corrélation entre l'appartenance de deux sommets à C_i et le fait qu'ils sont connectés alors la probabilité de tirer une arête interne est égale au produit des deux probabilités, à savoir $\left(\frac{d_i}{2|E|}\right)^2$.

Brandes *et al* [27] ont démontré que, pour une valeur τ donnée, vérifier l'existence d'une partition des sommets $\mathcal{C}$ d'un graphe G pour laquelle $Q \geq \tau$ est un problème NP-complet.
La modularité, telle que définie ici, souffre du problème de *limite de résolution* [55]. En effet, pour un ensemble d'arêtes E grand, une partition regroupant des petits groupes très faiblement connectés sera préférée aux partitions séparant ces groupes. Ce phénomène se retrouve sur des cas d'études simples [66] ou sur des *benchmarks* plus complexes [86]. Plusieurs illustrations de ce problème sont données dans la section 3.3.
Afin de répondre au problème de limite de résolution, d'autre formulations de la modularité introduisant un *paramètre de résolution* ont été proposées (voir notamment [119, 13]). L'équation 3 correspond à la formulation utilisée par [119].

$$Q_\lambda(G,\mathcal{C}) \quad = \quad \sum_{i=1}^{k} \left(\frac{e_{ii}}{|E|} - \lambda \left(\frac{d_i}{2|E|} \right)^2 \right) \tag{3}$$

où λ est un paramètre réel qui, lorsqu'il est grand, favorise les groupes de petite taille. Cette solution ne résout pas le problème dans tous les cas d'après [87].

Une mesure nommée *Surprise* a été proposée récemment dans [6]. À l'instar de la modularité, cette mesure évalue l'improbabilité d'observer la quantité d'arêtes internes dans une partition. La *Surprise* n'est pas une mesure additive. Des tests menaient sur le *benchmarks* LFR (voir section 2.6.5) suggèrent que des algorithmes de maximisation de cette mesure fournissent de meilleurs résultats que ceux maximisant la modularité. Toutefois, il n'a pas été démontré que cette mesure n'est pas sujette à certains bais telle que la *limite de résolution*.

3.1.2 Mesure de Mancoridis MQ

Mancoridis *et al.* [93] ont proposé une mesure de qualité appellée MQ (qui signifie *"Modularity Quality"*) afin d'évaluer la qualité d'un partition. Elle fut mise en œuvre dans le cadre d'un algorithme de partitionnement de graphes modélisant les relations entre différentes parties d'un logiciel. La mesure s'applique donc aux graphes orientés pouvant contenir des boucles.

La mesure MQ repose sur la prise en compte de densités d'arêtes. Elle s'exprime comme la somme des différences entre deux ratios de connectivité, calculés pour chaque groupe $\{C_i\}_{1 \leq i \leq k}$. Le premier est le *ratio de connectivité interne* et correspond au rapport entre le nombre de liens observé dans un groupe sur le nombre total de liens possibles. Le second est le *ratio de connectivité externe* et correspond à la somme des rapports entre le nombre de liens observé entre les groupes C_i et C_j sur le nombre total de liens possibles entre ces deux groupes.

Définition 3.2 *Soit $G = (V, E)$ un graphe orienté avec boucles et $\mathcal{C} = (C_1, \ldots, C_k)$ une partition de ces sommets. On note*

$$MQ(G, \mathcal{C}) = \frac{1}{k} \sum_{i=1}^{k} \left(\frac{e_{ii}}{|C_i|^2} - \frac{1}{k-1} \sum_{j \neq i} \frac{e_{ij}}{2|C_i||C_j|} \right) \quad (4)$$

L'idée de la mesure MQ est de comparer le nombre de liens internes et externes d'un groupe à un idéal. Ainsi, la mesure va privilégier un groupe C se rapprochant des graphes orientés complets ayant $|C|^2$ arcs pour le même nombre de sommets. Elle va également pénaliser la présence de graphes orientés bipartis complets entre deux groupes C_1 et C_2.

On vérifie facilement que MQ est une mesure additive (voir définition 2.51) sans être fortement additive. En effet, le ratio de connectivité externe et le facteur de standardisation k dépendent de la taille des autres groupes et du nombre de groupes respectivement. La mesure MQ est à valeur dans $[-1, 1]$. Tel que définie ici, la mesure MQ n'est adaptée qu'à un type de graphe bien particulier (orienté avec boucles). Une redéfinition permettant de prendre en compte différents types de graphes est proposée dans la section 3.2. Cette redéfinition permet également de rendre la mesure fortement additive.

La *Performance* [140] est un autre exemple de mesure basée sur la densité d'arêtes. Cette mesure correspond à la proportion de "bons couples" dans le partitionnement. Un bon couple étant soit deux sommets connectés et appartenant au même groupe soit deux sommets non-connectés appartenant à deux groupes différents. La *Performance* est une mesure fortement additive.

3.1.3 Qualité de compression L

La qualité de compression (originalement appelée *"Map Equation"* [126, 125]) d'une partition des sommets d'un graphe est basée sur l'idée que les groupes simplifient l'écriture sous forme codée d'un parcours aléatoire du graphe, de la même façon que l'utilisation des noms de villes simplifie l'écriture des adresses postales. On cherche ici à évaluer quantitativement la réduction de la complexité due aux groupes. Cette mesure a été généralisée au partitionnement hiérarchique (voir section 4.1).

Une marche aléatoire de graphes ne place pas tous les sommets sur un pied d'égalité. En effet, des sommets de plus fort degré ou ayant une *centralité* importante (voir définition 2.28) ont plus de chances d'être visités. De même, si le sommet courant appartient à une *communauté* alors il est probable que le prochain sommet visité appartienne à cette même communauté. L'ensemble de sommets de cette communauté forme alors une sorte de puits pour le marcheur aléatoire dont il est très difficile de sortir.

Un parcours de graphe (tel que la marche aléatoire) est décrit par la succession des sommets visités, certains sommets pouvant être visités plusieurs fois. Une façon de coder

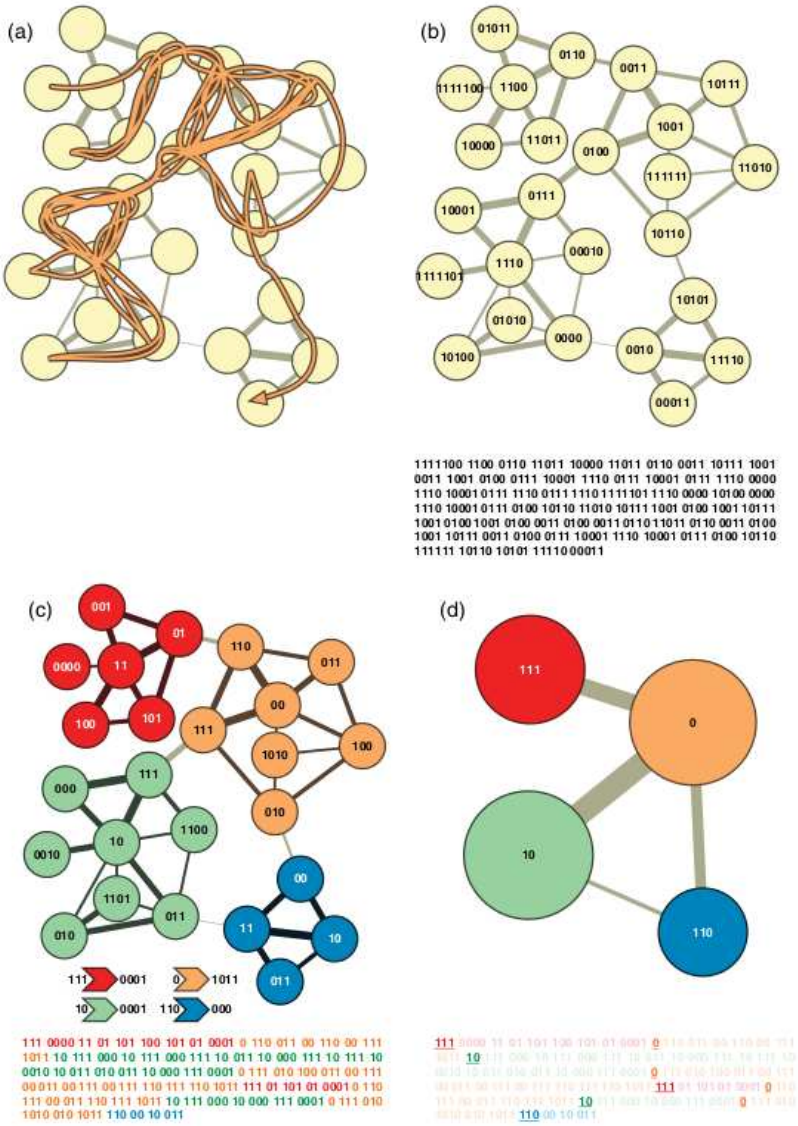


FIGURE 3.1: Illustration du principe de la qualité de compression d'une partition (tirée de [126]). **a)** Différents résultats de marche aléatoire sur un graphe. **b)** Un codage binaire encodant l'ensemble des parcours avec le moins de caractères possibles. **c)** Un codage à deux niveaux (sur les groupes puis sur les sommets) permettant à un même code d'être utilisé pour des sommets différents. Chaque groupe est associé à un code d'entrée et un code de sortie. **d)** Une visualisation des groupes sous forme de graphe quotient.

ce parcours est de disposer d'un identifiant unique par sommet en utilisant, par exemple, une notation binaire. Dans ce cadre, on peut attribuer un code de même taille à chaque sommet et il faut alors $p\lceil log_2(n)\rceil$ bits pour encoder le parcours si p désigne sa longueur et n désigne le nombre de sommets du graphe. Ce codage n'est pas optimal en terme de taille, une taille minimale est donnée par le théorème de Shannon [133] qui stipule que le

nombre de caractères nécessaire à décrire les états d'une variable aléatoire ne peut être inférieur à l'entropie (voir définition 2.53) de cette variable. La variable aléatoire est ici le sommet visité à un temps t de la marche aléatoire. La taille minimale pour coder un parcours est donné par la formule (5) sachant que la probabilité pour un sommet v d'être visité est estimée par $P_G(v) = \frac{d_G(v)}{2|E(G)|}$ dans le cas d'un graphe simple non-pondéré.

$$H(G) = -\sum_{v \in V(G)} P_G(v) log_2(P_G(v)) \tag{5}$$

La borne donnée par l'équation (5) peut être encore diminuée en utilisant un partitionnement du graphe. Pour cela, on associe à chaque groupe un code d'entrée et un code de sortie. Une marche aléatoire peut alors être décrite sans perte d'information. Un exemple illustrant cette idée se trouve en figure 3.1.

La probabilité de sortir d'un groupe C_i est notée $q_i^{out}(G)$, c'est la proportion d'arêtes ayant une seule extrémité dans C_i (voir équation (6)).

$$q_i^{out}(G) = \sum_{j \neq i} \frac{e_{ij}}{2|E(G)|} \tag{6}$$

Ainsi un code de sortie (et donc d'entrée) d'un groupe sera utilisé avec une probabilité notée $q^{out}(G) = \sum_{i=1}^{k} q_i^{out}(G)$. La compression maximale pour les codes des groupes est donc

$$H(\mathcal{C}, G) = -\sum_{i=1}^{k} \frac{q_i^{out}(G)}{q^{out}(G)} \log_2 \frac{q_i^{out}(G)}{q^{out}(G)} \tag{7}$$

D'un autre côté, le codage associé à un groupe C_i sera utilisé $p_i^{in}(G) = q_i^{out}(G) + \sum_{u \in C_i} P_G(u)$ fois en moyenne. La compression maximale pour les codes des sommets au sein d'un groupe C_i est donc donnée par

$$H(C_i, G) = -\frac{q_i^{out}(G)}{p_i^{in}(G)} \log_2 \frac{q_i^{out}(G)}{p_i^{in}(G)} - \sum_{v \in C_i} \frac{P_G(v)}{p_i^{in}(G)} \log_2 \frac{P_G(v)}{p_i^{in}(G)} \tag{8}$$

La taille de l'encodage minimum d'une marche aléatoire sur le graphe en utilisant la partition $\mathcal{C}$ est donnée dans la formule 9, en remplaçant les formules et après simplification on obtient l'équation 10.

$$\begin{aligned} L(\mathcal{C}) &= q^{out}(G) H(\mathcal{C}, G) + \sum_{i=1}^{k} p_i^{in}(G) H(C_i, G) \qquad (9) \\ &= \left(\sum_{i=1}^{k} q_i^{out} \right) \log_2 \left(\sum_{i=1}^{k} q_i^{out} \right) - 2 \sum_{i=1}^{k} q_i^{out}(G) \log_2(q_i^{out}(G)) \\ &\quad - \sum_{v \in V(G)} P_G(v) \log_2(P_G(v)) + \sum_{i=1}^{k} \left(q_i^{out} + \sum_{v \in C_i} P_G(v) \right) \log_2 \left(q_i^{out} + \sum_{v \in C_i} P_G(v) \right) \end{aligned}$$

Il faut noter que, pour deux partitions de G, on va préférer celle qui minimise cette quantité. Pour rester cohérent avec notre définition des mesures de qualité, nous utilisons une formulation équivalente donnée ci-dessous.

Définition 3.3 *Soit $G = (V, E)$ un graphe et $\mathcal{C} = (C_1, \ldots, C_k)$ une partition de ses sommets. Le* taux de compression $\Delta L(\mathcal{C})$ *est donné par*

$$\Delta L(\mathcal{C}) = \frac{H(G) - L(\mathcal{C})}{H(G)} \tag{10}$$

La mesure $\Delta L(\mathcal{C})$ définie dans l'équation 10 correspond au gain en terme de compression qu'apporte la partition $\mathcal{C}$ comparée à l'absence de partition.

Notons que la mesure $\Delta L(\mathcal{C})$ est une mesure de qualité additive, toutefois ni $\Delta L(\mathcal{C})$ ni $L(\mathcal{C})$ ne sont fortement additives. En effet, le gain de chaque groupe dépend de la quantité $q^{out}(G)$ qui correspond à la proportion d'arêtes externes de la partition $\mathcal{C}$.

3.2 Redéfinition de la mesure MQ

Dans cette partie, nous présentons une version modifiée de la mesure de Mancoridis [93] nommée MQ (voir définition 3.2). Comme noté précédemment, cette mesure repose sur l'évaluation de densité d'arêtes en comparant, pour chaque groupe, un ratio de connectivité interne et un ratio de connectivité externe. Ces ratios seront notés α_i et β_i respectivement. Ils dépendent tous deux d'un modèle de référence (ou idéal) comme, par exemple, le graphe orienté complet pour la densité. D'autres exemples seront ici proposés.

De plus, nous associons à chaque groupe un poids x_i et posons $X = \sum_{i=1}^{k} x_i$ comme le poids total des groupes. Nous introduisons ces poids de façon à faire de MQ un barycentre (moyenne pondérée) plutôt qu'une simple moyenne. En effet, on peut par exemple considérer que plus un groupe a une grande cardinalité, plus son poids dans la mesure doit être important. Toutefois, pour conserver la propriété d'additivité de la mesure, nous imposons que ces poids soient additifs *i.e.* le poids de l'union des parties doit être égal à la somme des poids des parties.

Définition 3.4 *Le* ratio de connectivité interne *d'un groupe $C_i \in \mathcal{C}$ est défini comme la proportion d'arêtes internes à C_i sur le nombre d'arêtes du graphe de référence ayant $|C_i|$ sommets.*

$$\alpha_i = \frac{e_{ii}}{\delta_i} \tag{11}$$

On considère que $e_{ii} \leq \delta_i$. De plus, par convention, on pose $\alpha_i = 0$ si $e_{ii} = 0$.

Lorsque le graphe étudié est simple et sans boucle, le modèle de référence "naturel" est le graphe complet sans boucle avec $|C_i|$ sommets. Dans ce cas, on a $\delta_i = \binom{|C_i|}{2}$. On

compare donc le sous-graphe induit par le graphe à une clique de même taille. Le cas d'un groupe ne contenant qu'un seul sommet impose un choix. Nous considérons ici qu'un tel groupe a un ratio de connectivité nul.

Définition 3.5 *Le* ratio de connectivité externe *d'un groupe $C_i \in \mathcal{C}$ est défini comme la moyenne pondérée des proportions d'arêtes liant les groupes C_i et C_j sur le nombre d'arêtes du graphe biparti de référence ayant deux ensembles de sommets de taille $|C_i|$ et $|C_j|$ respectivement.*

$$\beta_i = \frac{1}{X - x_i} \sum_{j \neq i} \frac{x_j e_{ij}}{\delta_{ij}} \tag{12}$$

Lorsque le graphe étudié est simple et sans boucle, le modèle de référence "naturel" est le graphe biparti complet sans boucle. Dans ce cas, on a $\delta_i = |C_i||C_j|$.

Définition 3.6 *Soit $G = (V, E)$ un graphe et $\mathcal{C} = (C_1, \ldots, C_k)$ une partition de ses sommets. La mesure MQ généralisée est donnée par*

$$MQ(G, \mathcal{C}) = \begin{cases} \frac{1}{X} \sum_{i=1}^{k} x_i(\alpha_i - \beta_i) & si\ |\mathcal{C}| > 1 \\ \alpha_1 & sinon \end{cases} \tag{13}$$

La mesure définie par l'équation (13) correspond à un barycentre sur la différence des ratios de connectivité (donné par les équations (3.4) et (3.5)) sur l'ensemble des groupes. En particulier, les groupes dont le rapport $\frac{x_i}{X}$ est grand auront un plus grand impact sur la valeur finale de la mesure.

Remarquons que la mesure originale de Mancoridis *et al.* (voir équation (4)) peut être retrouvée ici. Pour cela, on considère un poids uniforme sur l'ensemble des groupes $x_i = 1$, pour tout $i \in [1, \ldots, k]$. Ensuite, les graphes de références sont les graphes complets orientés avec boucles et les graphes bipartis complets.

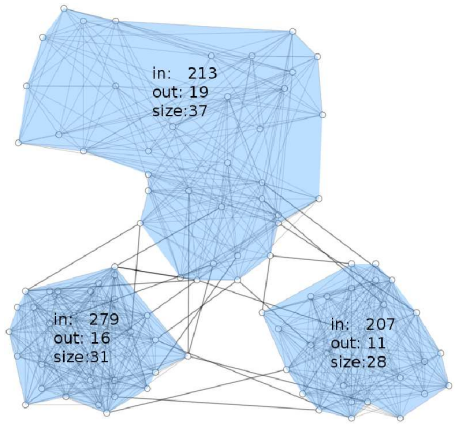


FIGURE 3.2: Partition d'un graphe de 96 sommets. Les trois groupes sont dessinés en utilisant des enveloppes concaves. Les quantités notées *in*, *out* et *size* correspondent respectivement à e_{ii}, $\sum_{j \neq i} e_{ij}$ et $|C_i|$.

Dans le cas de graphes non-orientés et sans boucle, nous pouvons prendre en tant que références les graphes complets et bipartis complets non-orientés. Ainsi, on aura $\delta_i = \binom{|C_i|}{2}$ et $\delta_{ij} = |C_i||C_j|$. Si on pondère chaque groupe avec sa taille $x_i = |C_i|$, on obtient la mesure définie dans l'équation 14.

$$\begin{aligned} MQ(G,\mathcal{C}) &= \frac{1}{|V|}\sum_{i=1}^{k}|C_i|\left(\frac{e_{ii}}{\binom{|C_i|}{2}} - \frac{\sum_{j\neq i}e_{ij}}{|C_i|(|V|-|C_i|)}\right) \\ &= \frac{1}{|V|}\sum_{i=1}^{k}\left(\frac{2e_{ii}}{|C_i|-1} - \frac{\sum_{j\neq i}e_{ij}}{|V|-|C_i|}\right) \end{aligned} \tag{14}$$

Nous illustrons le calcul de cette mesure par l'exemple donné en Figure 3.2. Dans ce cas, on a

$$\begin{aligned} MQ(G;\mathbf{C}) &= \frac{1}{96}\left(\frac{2\times 213}{37-1} - \frac{19}{96-37}\right) \\ &+ \frac{1}{96}\left(\frac{2\times 279}{31-1} - \frac{16}{96-31}\right) \\ &+ \frac{1}{96}\left(\frac{2\times 207}{28-1} - \frac{11}{96-28}\right) \\ &\simeq 0.47 \end{aligned}$$

La prise en compte de la taille des groupes a été suggérée dans [25]. Remarquons que dans le cas développé ici, le ratio de connectivité externe β_i équivaut au rapport entre la taille de la coupe $(C_i, V-C_i)$ sur la taille maximum de cette coupe, à savoir $|C_i|(|V|-|C_i|)$ (voir équation (14)). Puisque le gain d'un groupe ne dépend plus de la façon dont le reste des sommets est proportionné, la mesure MQ est sous cette forme fortement additive. Notons également qu'elle est à valeur dans $[-1, 1]$.

3.3 Analyse comparative des mesures MQ et Q

La mesure MQ, telle que définie dans la section 3.2, possède de bonnes propriétés. Dans le cas simple où la taille des groupes est utilisée et où les graphes complets et bipartis complets sont pris en graphes de références, la mesure est fortement additive à l'instar de la modularité Q.

Un autre avantage de la mesure MQ est de pouvoir évaluer le gain apporté par le fait d'utiliser une partition. En effet, pour la partition $\mathcal{C} = \{V\}$, la mesure $MQ(G,\mathcal{C})$ correspond à la densité d'arêtes dans G et est donc supérieure à 0 à partir du moment où $E(G)$ est non vide. Pour la modularité Q, l'absence de partition (ou partition en un groupe) correspond à une valeur de 0. Notons que la mesure $\Delta L(\mathcal{C})$ permet également de comparer un partitionnement à l'absence de partition.

Nous présentons dans cette section différentes analyses de la mesure MQ en la comparant avec la mesure de modularité Q. Le but est ici d'étudier le comportement de MQ et identifier ses biais éventuels. La plupart des expériences ou des exemples utilisés sont inspirés de travaux réalisés sur l'analyse de la mesure Q (voir notamment [87] et [66]).

3.3.1 Analyse Probabiliste sur le modèle d'Erdős-Rényi

Nous détaillons ici quelques résultats intéressants sur la mesure MQ lorsque l'on cherche à évaluer une partition d'un graphe aléatoire. Plusieurs modèles de graphes aléatoires sont détaillés dans la section 2.6. Nous nous intéressons ici aux graphes aléatoires de type Erdős-Rényi (voir section 2.6.1). Dans [67], les auteurs démontrent l'existence de partitions de sommets avec une forte modularité Q. Cette situation est problématique car l'apparition d'une structure de communauté est peu probable avec ce modèle.

On considère ici que la composante aléatoire d'un graphe G est l'ensemble de ses arêtes E, l'ensemble de sommets V étant fixé. Pour une partition $\mathcal{C} = (C_1, \ldots, C_k)$ de V, on va s'intéresser à la distribution de $MQ(G, \mathcal{C})$ pondéré (voir équation 14) en utilisant le graphe complet et le graphe biparti complet en tant que graphes de références. Afin d'éviter des situations singulières, on va poser comme contraintes $|\mathcal{C}| > 1$ et $|C_i| > 1$ pour tout $i \in [1, k]$.
On se place dans le cas où G peut être modélisé par un modèle de Erdős-Rényi. Deux sommets (u, v) sont connectés (on note cet événement $a_{uv} = 1$) avec une probabilité p.

Propriété 3.1 *Soit G un graphe aléatoire de type Erdős-Rényi avec une probabilité de lien p et $\mathcal{C}$ une partition de $V(G)$ telle que $|\mathcal{C}| > 1$ et $|C_i| > 1$ pour tout $i \in [1, k]$. On a*

$$E[MQ(G, \mathcal{C})] = 0 \tag{15}$$

$$V[MQ(G, \mathcal{C})] = \frac{p(1-p)}{2|V|^2} \sum_{i=1}^{k} \left[\frac{4|C_i|}{|C_i| - 1} + \sum_{j \neq i} |C_i||C_j| \left(\frac{1}{|V| - |C_i|} + \frac{1}{|V| - |C_j|} \right)^2 \right] \tag{16}$$

La propriété 3.1 s'explique par le fait que les variables $\{a_{uv}\}_{u<v}$ sont, sous cette hypothèse, indépendantes et identiquement distribuées et suivent une loi de Bernoulli de paramètre p. Dans ce cadre, la mesure MQ est bien une variable aléatoire en tant que somme pondérée de variables aléatoires. En effet, on peut reformuler MQ de la manière suivante :

$$MQ(G, \mathcal{C}) = \frac{1}{|V|} \sum_{v>u} S_{uv} a_{uv} \tag{17}$$

avec

$$S_{uv} = \begin{cases} \frac{2}{|C_i|-1} & \text{si } u, v \in C_i \\ -\left(\frac{1}{n-|C_i|} + \frac{1}{n-|C_j|} \right) & \text{si } u \in C_i, v \in C_j \end{cases}$$

L'espérance de MQ se calcule en remarquant que, pour un groupe C_i, on a en moyenne $\binom{|C_i|}{2}p$ arêtes internes ayant une contribution positive de $\frac{2}{|C_i|-1}$. Le total des contributions positives est donc de $|C_i|p$. Pour ce même groupe C_i, il y a en moyenne $|C_i|(|V|-|C_i|)$ arêtes externes avec une contribution négative de $\frac{-1}{|V|-|C_i|}$. Le total des contributions négatives est donc de $-|C_i|p$. Les contributions des arêtes internes et externes s'annulent donc dans le cas où $1 < |C_i| < |V|$ ce qui est notre hypothèse ici.

L'espérance nulle de MQ est aisément interprétable : si on fait l'hypothèse que le graphe que l'on partitionne est purement aléatoire, alors on ne peut espérer obtenir qu'une mesure de qualité moyenne (ici représentée par 0) pour la partition en question. Notons également que la variance σ^2_{ER} tend vers 0 lorsque $|V|$ tend vers l'infini.

3.3.2 Gain lié à la fusion de groupes

Nous allons maintenant étudier le comportement de MQ de façon analytique sur un exemple simple. Ce dernier est inspiré de celui utilisé dans [87] pour illustrer le problème de la résolution limite de la modularité (voir section 3.1) avec ou sans paramètre de résolution.

Pour deux groupes disjoints $A, B \subset V$ d'un graphe $G = (V, E)$ on cherche à savoir dans quelles situations il est préférable de regrouper ou non les sommets de A et de B au sein d'un même groupe noté $A \cup B$. Pour cela, on va utiliser différents paramètres qui sont illustrés dans la Figure 3.3. Les groupes A et B sont de taille t et contiennent $\binom{t}{2}p_{in}$ arêtes internes pour $p_{in} \in [0, 1]$. De plus, on suppose que A et B sont reliés par $t^2 p_{AB}$ arêtes pour $p_{AB} \in [0, 1]$ et que chacun a également une arête le reliant aux sommets de $(V \setminus (A \cup B))$. On note également n (respectivement m) le nombre de sommets (resp. arêtes) de G.

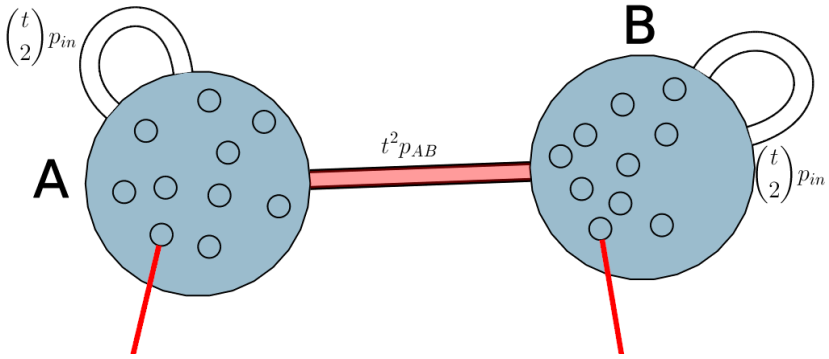


FIGURE 3.3: Deux groupes A et B de taille t et contenant $\binom{t}{2}p_{in}$ arêtes internes. Ils sont reliés par $t^2 p_{AB}$ arêtes et chacun par une arête avec le reste du graphe (exemple inspiré de [87]).

On cherche à analyser le gain de qualité obtenu en regroupant A et B. Pour une mesure fortement additive $\Phi(G, \mathcal{C})$, le gain obtenu est noté $\Delta_{A\cup B}\Phi$ et on a

$$\Delta_{A\cup B}\Phi = \phi(G, A \cup B) - (\phi(G, A) + \phi(G, B)) \tag{18}$$

d'après la propriété de forte additivité (voir définition 2.52).

Propriété 3.2 *En utilisant la modularité Q, le gain lié à la fusion des groupes A et B est*

$$\Delta_{A\cup B}Q \quad = \quad \frac{t^2 p_{AB}}{m} - \frac{(\binom{t}{2}p_{in} + t^2 p_{AB} + 1)^2}{2m^2} \tag{19}$$

La quantité $\Delta_{A\cup B}Q$ est fonction du nombre total d'arêtes m. Elle illustre bien le problème de limite de résolution discuté dans la section 3.1. En effet, le second terme décroît bien plus vite en fonction de m que le premier terme. La quantité $\Delta_{A\cup B}Q$ sera donc facilement positive si le graphe G est grand (en terme d'arêtes) bien que la proportion d'arêtes entre A et B soit très faible. Cela signifie que le regroupement de A et B sera presque toujours préférable pour un nombre d'arêtes suffisant. En particulier, la quantité $\Delta_{A\cup B}Q$ est de même signe que $2m\Delta_{A\cup B}Q$ qui tend vers $2t^2 p_{AB}$ lorsque m tend vers l'infini. Cette quantité sera positive même si p_{AB} est très proche de 0.

Propriété 3.3 *En utilisant la mesure MQ, le gain lié à la fusion des groupes A et B est*

$$\Delta_{A\cup B}MQ \quad = \quad \frac{4\binom{t}{2}p_{in} + 2t^2 p_{AB}}{2\,t - 1} - \frac{2}{n - 2\,t} - \frac{4\binom{t}{2}p_{in}}{t - 1} + \frac{2(t^2 p_{AB} + 1)}{n - t} \tag{20}$$

Dans le cas extrême où $n \to \infty$, on a

$$\Delta_{A\cup B}MQ \to \frac{2t^2(p_{AB} - p_{in})}{2t - 1} \tag{21}$$

Cette quantité est positive si et seulement si $p_{AB} > p_{in}$. Cela est dû au fait que lorsque n est très grand, la contribution négative des ratios de connectivité externe devient moins importante (et nulle lorsque n tend vers l'infini). Dans ce cadre, évaluer le gain lié à la fusion revient à comparer le ratio de connectivité interne de $A \cup B$ à ceux de A et B.

Ainsi, dans le cas où n est très grand et que A et B sont deux cliques de petite taille (avec $p_{in} = 1$), le gain lié à la fusion de A et B sera négatif ou nul ; ce dernier cas apparaissant lorsque $p_{AB} = 1$. Ainsi, dans cet exemple, regrouper deux cliques non-maximales en une clique maximale n'améliore pas la mesure. D'autres valeurs de p_{AB} pour lesquelles le gain de la fusion est nul en fonction de t et de n sont données dans la Figure 3.4.

Le comportement observé pour la modularité Q ne se retrouve pas en analysant la quantité $\Delta_{A\cup B}MQ$. La mesure MQ ne va pas systématiquement favoriser la fusion de

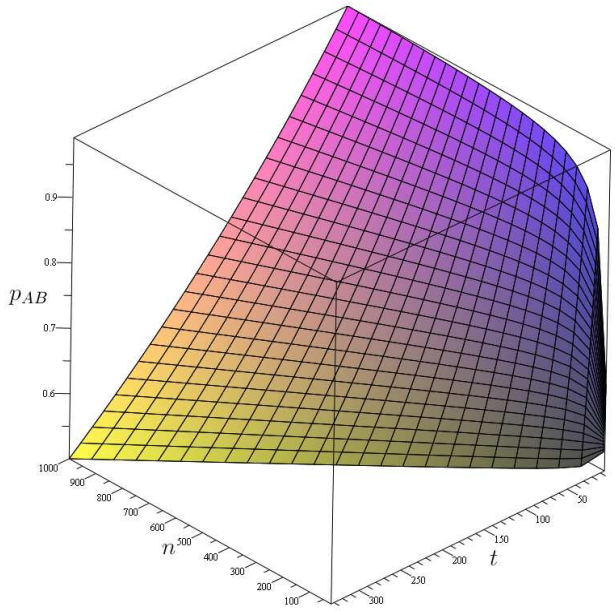


FIGURE 3.4: Pour $p_{in} = 1$, le plan représente les valeurs de p_{AB} à partir desquelles $\Delta_{A \cup B} MQ \geq 0$ pour différentes valeurs t et n.

groupes faiblement connectés. Cependant, on peut constater un autre type de dégénérescence lié à l'utilisation de la mesure MQ. Celle-ci dépend du nombre de sommets du graphe et non pas du nombre d'arêtes. Utilisée en tant que fonction objective dans un algorithme, la mesure MQ aura tendance à retourner des partitions contenant plus de groupes que les partitions maximisant la mesure Q.

3.3.3 Analyse Expérimentale

Nous allons ici étudier le comportement de la mesure MQ en utilisant la procédure détaillée dans [66]. Cette procédure est la suivante :

1. Extraire un M-échantillon $(C^i, MQ(G, \mathcal{C}^i))_{1 \leq i \leq M}$ couvrant au mieux l'espace.
2. Calcul de la distance entre chaque couple $\mathcal{C}^i, \mathcal{C}^j$ présent dans l'échantillon.
3. Analyse des résultats.

Le parcours exhaustif de l'ensemble des partitions de V n'est pas une démarche réaliste même si on étudie des exemples de grande taille. Tirer uniformément un certain nombre de partitions a un inconvénient majeur : on risque de ne couvrir que peu la (ou les) zone(s) où la mesure de qualité est forte. La technique utilisée est comparable à un *recuit simulé* [1]. Nous allons appliquer à une partition C une modification et valider cette dernière si elle a conduit à une amélioration de la qualité MQ. Toutefois, une modification diminuant MQ peut également être retenue avec une probabilité dépendant de la perte engendrée et d'un paramètre décroissant (la température) avec le nombre d'itérations de l'algorithme à la

manière d'un algorithme de Metropolis-Hasting [122].
L'expression *modification* correspond à une sélection aléatoire parmi trois mouvements possibles : affecter un sommet quelconque à un groupe différent, découper en deux un groupe ou regrouper deux groupes. Chacune de ces options est réalisée avec une probabilité différente. Le processus est ergodique : le nombre de modifications nécessaires pour aller d'un état à n'importe quel autre est fini.

Notons que cet algorithme est, sous cette forme, plus adapté à la recherche d'un maximum global qu'à la collecte d'un échantillon. Pour achever ce dernier objectif plusieurs modifications sont apportées :

- à chaque étape, on enregistre C ainsi que $MQ(G, \mathcal{C})$.
- selon un pas déterminé au préalable on réinitialise la température T à T_0 ce qui permet à l'algorithme de parcourir un ensemble de partitions plus éloignées, particulièrement dans le cas où la valeur de MQ est forte.
- le critère d'arrêt n'est pas explicité ici. On cherche simplement à recueillir un M-échantillon. On arrêtera donc l'algorithme après M itérations.

La distance entre les différentes partitions extraites par cette méthode est calculée en utilisant la variation d'information VI (voir définition 2.9). Afin de pouvoir visualiser les résultats de manière globale, en considérant toutes les partitions échantillonnées, nous utilisons une méthode de positionnement multidimensionnel [37]. Elle va permettre d'associer à chaque partition $(C^i)_{1 \leq i \leq M}$ des coordonnées à deux dimensions (x_i, y_i) de manière à ce que les distances (définies par VI) soient respectées. La méthode choisie est l'analyse en composantes curvilignes (*Curvilinear Distance Analysis* [91]).

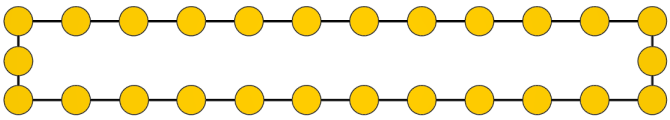
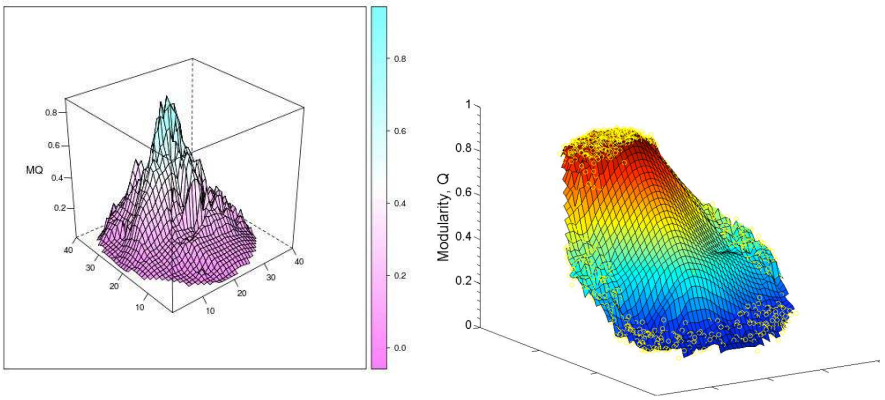


FIGURE 3.5: Réseau en anneau avec 24 cliques (les cercles) de 5 sommets chacune pour un total de 120 sommets et 264 arêtes.

Nous analysons les résultats sur le *réseau en anneau* (voir Figure 3.5) dans lequel k *cliques* sont reliées à leurs voisines par une unique arête. Good *et al.* [66] ont constaté une dégénérescence de la mesure de modularité Q dans ce cadre. En particulier, les configurations regroupant deux cliques voisines donnent des valeurs de Q supérieures à celles données par la configuration où chaque clique est isolée. On a ici cherché si un tel comportement se retrouve avec la mesure MQ (telle que définie par la formule 14). On génère pour cela un échantillon de 1200 couples $(\mathcal{C}, MQ(G, \mathcal{C}))$.

La partition maximisant MQ est bien celle qui isole chaque clique. Dans le cas de la configuration optimale, il semble que MQ décroît linéairement à mesure que l'on s'éloigne



(a) (b)

FIGURE 3.6: **(a)** Reconstruction 3D de la mesure MQ pour un réseau en anneau. **(b)** Reconstruction 3D pour le même exemple avec la mesure Q (image tirée de [66])

en terme de variation d'information. Cette observation est confirmée sur d'autres exemples bien que la partition correspondant à la qualité maximum ne soit pas toujours la vérité terrain. Une vue en trois dimensions est disponible dans la Figure 3.6. La surface est obtenue en utilisant l'interpolation linéaire d'Akima [4]. On constate une rapide progression vers de fortes valeurs de MQ (Figure 3.6(a)) avec un nombre restreint de "plateaux". Ce n'est pas le cas pour la modularité Figure 3.6(a)), en effet on constate un large plateau correspondant à des valeurs de Q proches du maximum. De plus, sur cet exemple, la modularité admet plusieurs maxima globaux qui ne correspondent pas à la partition attendue.

3.3.4 Conclusion sur les analyses

On peut conclure de ces analyses que MQ semble répondre à certains problèmes que posent la modularité. En effet, la mesure ne semble pas être affectée par la *résolution limite* car elle pénalise le regroupement de groupes très faiblement connectés. Des expérimentations indiquent qu'ils n'existent pas un grand nombre de partitions à la fois éloignées entre elles et avec un MQ proche de de l'optimum.

Toutefois, cette mesure, sous sa forme pondérée, pose d'autres problèmes car le regroupement de petits groupes très denses et très fortement connectés entre eux n'est pas toujours optimal si le nombre de sommets est important. Ces remarques sont valides dans le cas où les graphes complets sont utilisés pour déterminer le nombre maximum d'arêtes dans et entre chaque groupe.

3.4 Discussion et Perspectives

Dans ce chapitre, nous avons introduit différentes mesures permettant d'évaluer la qualité d'une partition des sommets d'un graphe. Ces mesures dépendent essentiellement du nombres d'arêtes internes et externes à chaque groupe. En particulier, nous avons proposé une extension de la mesure MQ en introduisant une pondération sur chaque groupe. De plus, nous avons introduit la notion de "graphe de références" c'est-à-dire la situation idéale à laquelle on se compare pour évaluer les relations intra et inter groupes.

Nous n'avons pas couvert de manière exhaustive l'ensemble des moyens de mesurer la qualité d'une partition. Cependant les mesures MQ et Q sont bien représentatives de deux approches différentes. D'un côté, la modularité compare une quantité observée à une quantité théorique prise dans le cas où le partitionnement n'explique pas la topologie du graphe. D'un autre côté, la mesure MQ teste l'adéquation entre les quantités observées et un cas optimal.

Comme on l'a vu sur des exemples simples, les deux mesures produisent des résultats différents. Toutefois, cette différence ne vient pas forcément de l'approche choisie mais peut-être des quantités observés. En effet, la modularité utilise la proportion d'arêtes internes et MQ utilise la densité d'arêtes internes. Remarquons qu'il est possible de modifier la modularité pour prendre en compte la densité. Des premiers tests semblent montrer que dans ce cas, la modularité se comporte de manière similaire à MQ.

Il serait intéressant de définir des maxima pour le nombre d'arêtes internes/externes qui soient des cas idéaux tout en tenant compte de la densité naturelle du graphe (comme le font les mesures de type *modularité*). Par exemple, connaissant λ la taille de la clique maximum de G, on sait qu'un groupe C avec $|C| > \lambda$ ne pourra jamais être de densité $\binom{|C|}{2}$ par définition. Toutefois, déterminer la taille de la clique maximum (voir section 2.4) est NP-Difficile.
Une piste possible est d'utiliser des bornes basées sur la *coreness* des sommets (voir Définition 2.23). En effet, la *coreness* permet, dans le cadre de l'analyse de réseau, de déterminer l'échelle à laquelle un sommet intervient et est souvent considérée comme une statistique plus *robuste* que le degré [9]. En particulier, les valeurs de $core_G(u)$ pour $u \in V(G)$, fournissent une borne supérieure à $|E(G)|$ [19] donnée dans l'équation suivante.

$$|E(G)| \leq \sum_{u \in V(G)} core_G(u) - \binom{c+1}{2} \tag{22}$$

L'inéquation 22 est valide si G est simple et non-orienté avec une dégénérescence c. Elle permet de définir d'une nouvelle façon la densité de G. Il est également possible de définir de nouveaux ratios de connectivité interne et externe pour un sous-graphe $C \subset V(G)$. Des expérimentations sont toutefois nécessaires pour valider cette approche.

Chapitre 4

Mesurer la qualité de partitions hiérarchiques de graphes

Une approche simple pour produire une partition hiérarchique de graphes est l'application itérative d'un algorithme de partitionnement non-hiérarchique soit à chaque sous-graphe induit soit au graphe quotient formé par les groupes à l'itération précédente. Selon que les méthodes soit agglomératives ou divisives, les itérations vont consister à découper des groupes en sous-groupes ou à agréger ensemble plusieurs groupes.

Bien que cette procédure produise une hiérarchie, chaque niveau créé résulte d'une décision locale que l'on espère optimale. La hiérarchie produite est donc une combinaison d'optima locaux et rien ne garantit la construction d'un optimum global. Pour juger cela, il faut être en mesure d'évaluer la qualité de la structure produite dans le cas où une évaluation extérieure ou une vérité terrain ne sont pas disponibles. Nous pensons que cette évaluation doit tenir compte du caractère hiérarchique de la structure, c'est-à-dire un modèle d'emboîtement entre groupes, et non simplement une séquence de partitions indépendantes entre elles.

La construction de telles mesures est l'objet de cette section. Nous proposons en effet une généralisation des mesures de qualité additives (voir Définition 2.51) aux partitions hiérarchiques [116]. L'idée est d'appliquer récursivement la mesure de qualité à chaque sous-division d'un groupe en introduisant une variable q permettant de tenir compte de la profondeur (dans la hiérarchie) à laquelle un groupe apparaît. Le résultat est un polynôme en q.
Un état de l'art comprenant deux travaux existants est proposé dans la section 4.1. Notre solution à cette problématique est détaillée en section 4.2 : nous introduisons une mesure qualité pour des arbres de partitions qui généralise les mesures de qualité additives. Une évaluation analytique et expérimentale de notre contribution est disponible dans la section 4.3.

4.1 État de l'art

Nous présentons ici deux travaux fortement reliés à notre problématique. Le premier [110] propose une méthode s'appuyant sur les mesures de qualité additives pour évaluer la qualité d'un filtrage appliqué sur un dendrogramme. Le second [124] est une généralisation de la *Qualité de compression* qui a été définie dans la section 3.1 proposée par les mêmes auteurs.

4.1.1 Qualité de résolution multi-échelles

Pons *et al.* [110, 109] s'intéressent à l'extraction de partitions à partir d'un dendrogramme dont les feuilles correspondent aux sommets du graphe G. Partant de l'observation que la plupart des mesures de qualité sont *additives* (voir Définition 2.51) et que le gain de chaque groupe $C \in \mathcal{C}$ est la somme d'une contribution positive notée $h(C)$ et négative notée $l(C)$, les auteurs introduisent un paramètre de résolution α permettant de moduler l'impact de ces deux fonctions.

Définition 4.1 ***(Qualité de résolution multi-échelles)*** *Soit un graphe G muni d'une partition $\mathcal{C} = (C_1, C_2, \ldots, C_k)$ de l'ensemble V. La* qualité de résolution multi-échelles *de la partition $\mathcal{C}$ est donnée par*

$$Q_\alpha(\mathcal{C}) = \sum_{i=1}^{k} \alpha h(C_i) + (1-\alpha) l(C_i) \tag{1}$$

où $\alpha \in [0,1]$ est le paramètre de résolution et les fonctions h et l sont telles que pour toute paire de sous-ensembles disjoints C, C' de V, on a $h(C \cup C') \geq h(C) + h(C')$ et $l(C \cup C') \leq l(C) + l(C')$.

Soit $(\mathcal{C}_\alpha)_{0\leq\alpha\leq1}$ la suite de partitions de V où chaque élément correspond à la partition obtenue par une coupe du dendrogramme qui maximise la qualité de résolution multi-échelles. Cette suite contient au plus n partitions distinctes et peut être calculée avec une complexité moyenne de $\mathcal{O}(n\sqrt{n})$ où n est le nombre de sommets du graphe.
Dans ce cadre, notons qu'une communauté C correspondant à un nœud du dendrogramme pourra apparaître dans plusieurs (ou aucune) partitions de $(\mathcal{C}_\alpha)_{0\leq\alpha\leq1}$. On peut considérer que plus une communauté apparaîtra souvent, plus celle-ci sera pertinente. De plus, pour un paramètre de résolution $\alpha^* \in [0,1]$, on peut évaluer la pertinence moyenne des communautés appartenant à $\mathcal{C}_{\alpha^*}$, on va donc pouvoir explorer le dendrogramme pour en retirer les paramètres de résolution et les partitions correspondantes les plus pertinentes.

Définition 4.2 ***(Pertinence de la résolution α)*** *Soit un graphe G muni d'un dendrogramme T et $(\mathcal{C}_\alpha)_{0\leq\alpha\leq1}$ la suite de partitions issues de T et maximisant une fonction de qualité de résolution multi-échelles Q_α.*
Pour un groupe $C \in T$, on pose $\alpha_{min}(C)$ (respectivement $\alpha_{max}(C)$) la première (resp. la

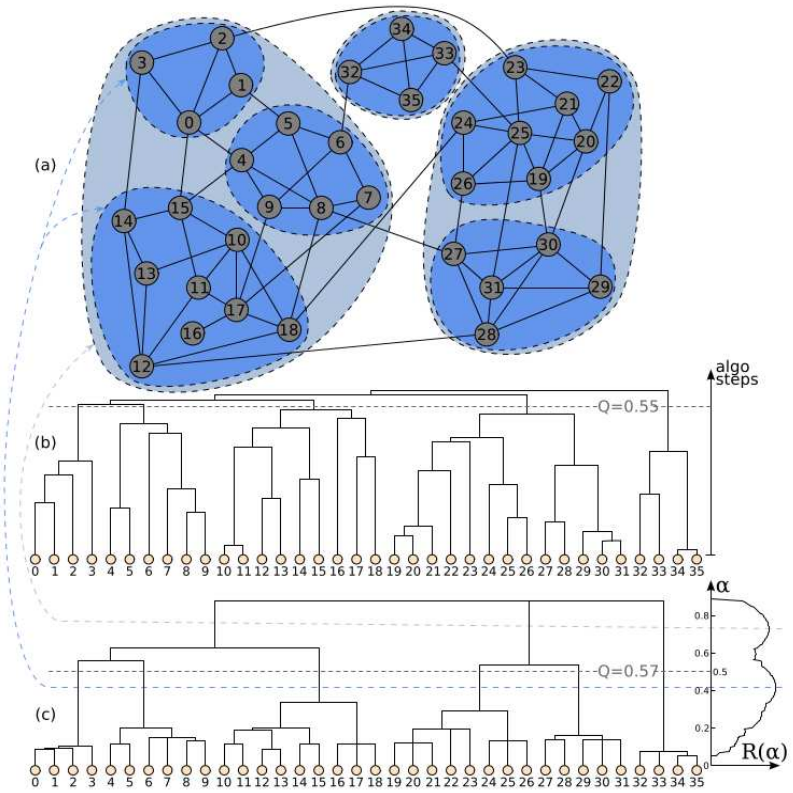


FIGURE 4.1: Illustration (tirée de [109]) d'une procédure d'extraction d'une partition hiérarchique optimale à partir d'un dendrogramme. **(a)** Le graphe partitionné. **(b)** Un dendrogramme des sommets du graphe. La ligne pointillée indique la coupe maximisant la modularité (voir définition 3.1). **(c)** Partition hiérarchique du graphe formé par la liste des partitions maximisant la modularité généralisée Q_α (voir définition 4.1). Les lignes bleues pointillées correspondent à deux maxima locaux de $R(\alpha)$.

dernière) valeur de α pour laquelle C appartient à $\mathcal{C}_\alpha$.
La pertinence de C pour le paramètre α est donnée par la formule suivante

$$R_\alpha(C) = \frac{\alpha_{max}(C) - \alpha_{min}(C)}{2} + \frac{2(\alpha_{max}(C) - \alpha)(\alpha - \alpha_{min}(C))}{\alpha_{max}(C) - \alpha_{min}(C)} \tag{2}$$

La pertinence de la résolution α, *notée* $R(\alpha)$, *correspond à*

$$R(\alpha) = \sum_{C \in \mathcal{C}_\alpha} \frac{|C| R_\alpha(C)}{n} \tag{3}$$

La fonction $R(\alpha)$ peut typiquement contenir plusieurs maxima locaux (voir Figure 4.1). Ces maxima locaux correspondent à des partitions pertinentes du graphe. Dans l'exemple utilisé ici, on peut en effet sélectionner deux coupes pour obtenir un partitionnement hiérarchique à deux niveaux.

Les différentes mesures développées ici permettent d'évaluer des partitions plates au sein d'une hiérarchie (considérée comme un dendrogramme) en utilisant une mesure de

qualité généralisée introduisant un paramètre de résolution α. Cette méthode ne permet donc pas directement d'évaluer la qualité d'une partition hiérarchique quelconque. Toutefois, la méthode est générique et peut être appliquée en utilisant différentes mesures de qualité en fonction de préférences ou de connaissances *a priori* sur la forme des communautés à identifier. Les exemples utilisés par les auteurs (comme celui de la Figure 4.1) illustrent un point important : la meilleure partition hiérarchique n'est pas toujours une combinaison des meilleurs partitions plates globales.

4.1.2 Qualité de compression hiérarchique

Cette mesure est proposée par Rosvall *et al.* [124], l'idée utilisée est un prolongement de celle qui définit la qualité de compression d'une partition à un niveau : l'emploi de codes d'entrée et de sortie pour chaque groupe. Ce codage permet de réduire la taille (en bits) nécessaire pour décrire une marche aléatoire sur le graphe.

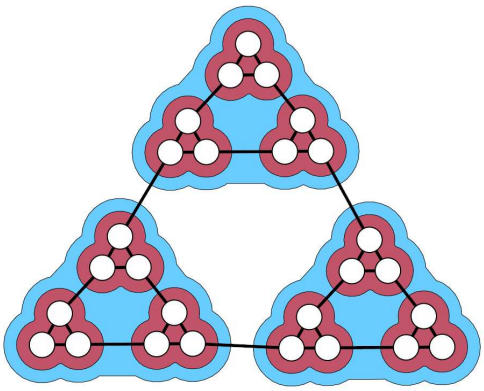


FIGURE 4.2: Illustration du calcul de la qualité de compression d'une partition hiérarchique (exemple tiré de [124]). Le graphe a 27 sommets et un degré total de 78 arêtes, la longueur minimale d'un code est $H(G) = 4.75$ bits. Les enveloppes délimitent des groupes, le graphe est donc ici partitionné sur deux niveaux.

Si on dispose d'un arbre de partition T, on peut évaluer le gain obtenu en affectant un code d'entrée et de sortie à chaque nœud interne de la hiérarchie. Si on reprend la métaphore des adresses postales, cela revient à considérer qu'on peut utiliser non seulement des noms de rues mais également des noms de pays, de départements, de villes *etc.* A grande échelle, ceci entraîne une réduction significative du nombre d'entrées différentes à utiliser pour coder une adresse.

Définition 4.3 *Soit G un graphe et T un arbre de partition de $V(G)$ enraciné en r. La* qualité de compression hiérarchique *de T notée $L(T)$ est donnée par la formule*

$$L(T) = \begin{cases} q_r^{out}(G)H(N_1(T_r), G) + \sum_{t\in\sigma(r)} L(G[V_t], T_t) & si\ \sigma(r) \neq \emptyset \\ p_r^{in}(G)H(V_r, G) & sinon \end{cases} \quad (4)$$

en reprenant les notations utilisées dans la définition 3.3.

Ici, V_t est le sous-ensemble de V associé au nœud t et $N_1(T_r)$ désigne la partition de V_r induite par les successeurs de r dans T. Comme pour la qualité de compression simple, on utilise ici le gain apporté par l'arbre de partition T par rapport à la situation où aucune partition n'est utilisée.

Définition 4.4 *Soit G un graphe et T un arbre de partition de $V(G)$ enraciné en r. Le* taux de compression hiérarchique *de T noté $\Delta L(T)$ est*

$$\Delta L(T) = \frac{H(G) - L(T)}{H(G)} \tag{5}$$

Le calcul de cette mesure est illustré sur un exemple simple (voir Figure 4.1.2). Si t est la racine de la partition, on a $q_t^{out} = \frac{6}{78}$ et $H(N_1(T_t), G)$ est l'entropie de la séquence $(\frac{2}{6}, \frac{2}{6}, \frac{2}{6})$, en effet il y a un degré total de 6 entre les enveloppes bleues. Si $t \in T$ correspond à une enveloppe bleue, on a $q_t^{out} = \frac{10}{78}$ et $H(N_1(T_t), G)$ est l'entropie de la séquence $(\frac{2}{10}, \frac{3}{10}, \frac{3}{10}, \frac{2}{10})$. Notons que le quatrième terme correspond à l'utilisation du code de sortie de l'enveloppe bleue. Enfin si t est une enveloppe rouge (qui n'est pas une des trois aux extrémités), on a $p_r^{in}(G) = \frac{10}{78}$. On peut remarquer que la contribution des enveloppes rouges est la même peu importe si les enveloppes bleues font, ou non, partie de la partition. Dans cet exemple, le taux de compression obtenu en utilisant la partition à deux niveaux est de 26% environ. En utilisant seulement le dernier niveau (les enveloppes rouges), on a un taux de compression d'environ 24%.

4.2 Généralisation des mesures de qualité additives aux partitions hiérarchiques

Nous proposons ici une généralisation des mesures de qualité additives (voir Définition 2.51) aux partitions hiérarchiques. Nous présentons d'abord le raisonnement nous ayant permis d'établir la mesure. Dans un second temps, nous définissons celle-ci avec deux formulations équivalentes.

L'approche présentée est nouvelle dans le sens où aucune mesure d'évaluation de la qualité d'une partition hiérarchique n'a été proposée jusqu'à présent exception faite de la mesure introduite par Rosvall *et al.* [124] (voir section 4.1) qui fut développée indépendamment durant la même période. Nous discuterons de la différence entre les deux approches dans la partie 4.3.

La généralisation d'une mesure de qualité additive Φ aux partitions hiérarchiques correspond à une définition récursive des gains impliquant une variable notée q. Le résultat est un polynôme en q. La plupart des mesures de qualité additives telles que MQ ou Q

peuvent être calculées en passant sur chaque arête et en regardant si les extrémités appartiennent ou non au même groupe. Après ce test, un poids (positif ou négatif) peut être associé à l'arête. En laissant de côté les constantes de normalisation, ces poids correspondent à une valeur ± 1. Dans le cas de partitions hiérarchiques, notre but est de prendre en compte la hauteur à laquelle une arête change de statut. En effet, une arête peut rester interne sur plusieurs niveaux. Nous allons donc associer un poids positif $1 + q + \cdots + q^r$ qui dépend de la hauteur maximale r à laquelle l'arête est interne. De la même manière, une arête liant deux groupes différents se voit assigner un poids négatif q^{r+1}.

Les motivations pour l'emploi de cette formulation de Φ proviennent du domaine de la combinatoire énumérative pour l'analyse récursive d'objets discrets par des *q-analogues*. La variable q permet de garder une trace de la profondeur intrinsèque de l'objet. Dans la plupart des cas, les objets peuvent être décrits par des langages formels générés par des grammaires algébriques, appelées *grammaires attribuées* avec l'introduction d'une variable de comptage q (voir [42] et [97]). On peut ainsi prendre en compte dans une même série énumératrice à la fois un paramètre algébrique tel que la longueur du mot et un paramètre non-algébrique tel que les positions d'une lettre dans un mot.

4.2.1 Définition de la mesure multi-niveaux

Nous pouvons maintenant présenter formellement les formules permettant de répondre à notre problème. La première, donnée par l'équation 6, est une formulation récursive.

Définition 4.5 ***(Mesure de qualité multi-niveaux)*** *Soit $\Phi(G, \mathcal{C})$ une mesure de qualité additive. La généralisation de $\Phi(G, \mathcal{C})$ en* mesure de qualité multi-niveaux *notée $\Phi(G, T; q)$ pour un arbre de partition T de racine r appliqué au graphe G s'exprime, pour $q \in [0, 1]$, par*

$$\Phi(G, T; q) = \begin{cases} \sum_{t \in \sigma_T(r)} \phi(G, N_1(T), V_t) \left(1 + q \times \Phi(G_t, T_t; q)\right) & \textit{si } \sigma_T(r) \neq \emptyset \\ 0 & \textit{sinon} \end{cases} \tag{6}$$

Dans le cas où T_t est une partition plate de G, la fonction $\Phi(G, T; q)$ correspond à la mesure de qualité classique. Nous avons donc bien une généralisation au cas hiérarchique.

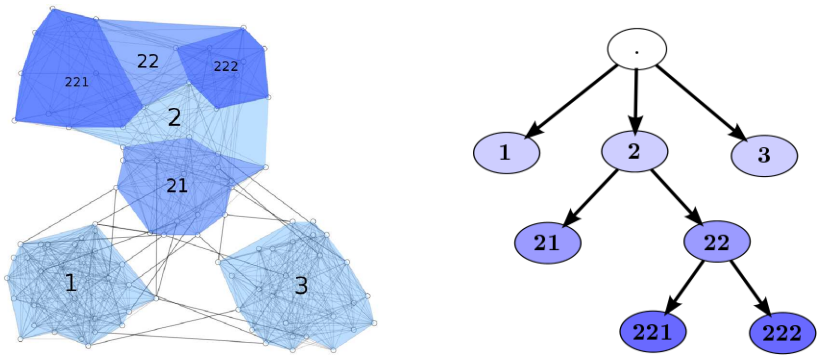
Le paramètre q appartient à l'intervalle $[0, 1]$. En effet, si $q < 0$, les arête internes ont un poids négatif dans la mesure tandis que les arêtes externes peuvent affecter positivement la mesure. De la même manière, prendre une valeur pour q supérieure à 1 correspond à la situation où plus un groupe est profond dans la hiérarchie plus la contribution de celui-ci est forte. Le risque est d'occulter complètement la contribution des premiers niveaux si le polynôme est de fort degré (*i.e.* si l'arbre est profond).

Examinons maintenant la complexité du calcul du polynôme $\Phi(G, T; q)$. On va se limiter au cas où $\Phi(G, \mathcal{C})$ est une mesure fortement additive (voir définition 2.52) où le

gain de chaque groupe dépend du nombre d'arêtes internes et/ou externes à ce groupe. Rappelons que la modularité Q (équation 2) et le MQ pondéré (équation 14) sont dans ce cas.
Le calcul revient à faire un parcours de l'arbre de partition, ce qui a une complexité en temps de $\mathcal{O}(|T|)$. A chaque nœud visité, le gain du groupe de sommet correspondant doit être calculé. Ces deux quantités peuvent être calculées efficacement si on considère que la hiérarchie est encodée dans un tableau qui à chaque sommet $u \in V$ associe une liste d'identifiants indiquant le groupe auquel u appartient dans les différents niveaux. La taille maximum d'une liste est alors h, la hauteur de l'arbre de partition. Le calcul de tous les nombres d'arêtes internes/externes peut être réalisé en $\mathcal{O}(h|E|)$.

4.2.2 Formulation en termes de chemins

La définition 4.5 utilise une récursion dans l'arbre de partition. La fonction $\Phi(G, T; q)$ est un polynôme en q. Cependant, comme nous allons le montrer ci-dessous, les coefficients de ce polynôme peuvent être obtenus directement.
Nous considérons que les descendants d'un nœud t de T sont étiquetés par les entiers $1, 2, \ldots$. Tout chemin allant de la racine à un nœud se situant à une hauteur r s'écrit alors comme une séquence d'entiers $w = i_1 \ldots i_r$. Nous appelons une telle séquence un *mot* sur l'alphabet $\{1, 2, \ldots\}$ et $|w|$ est la longueur de ce mot. La Figure 4.3 illustre ces notations : le mot codant le chemin issu de la racine est indiqué pour chaque nœud de l'arbre. Dans ce cadre la notation $u \prec w$ indique que u est un préfixe de v (le nœud u est un ancêtre de v dans T).



(a) Une partition hiérarchique d'un graphe.

(b) Représentation de l'arbre de partition.

FIGURE 4.3: Un arbre (à droite) encodant la partition hiérarchique du graphe (à gauche).

Nous pouvons réécrire la mesure de qualité multi-niveaux en utilisant ces notations. En effet, pour obtenir la contribution d'un nœud t de T de hauteur $p + 1$, il suffit de

multiplier les gains des groupes associés à chaque nœud se situant sur l'unique chemin partant de la racine au nœud t (on considère la racine comme ayant une contribution de 1). Le coefficient du polynôme $\Phi(G, T; q)$, noté $[\Phi(G, T; q); q^p]$ est donné par l'équation 7.

$$[\Phi(G, T; q); q^p] = \sum_{w \in \mathcal{D}_p} \prod_{u \prec w} \phi(G, \{C_v\}_{v \in \sigma_T(p(u))}, u) \tag{7}$$

où $\mathcal{D}_p = \{w \in T, |w| = p + 1\}$ est l'ensemble des nœuds ayant une hauteur p dans T.

L'équation 7 fournit une formulation alternative pour $\Phi(G, T; q)$. Il suffit alors d'itérer sur chaque niveau de la hiérarchie *i.e.* toute hauteur p telle que $\mathcal{D}_p \neq \emptyset$.

4.2.3 Interprétation et Utilisation

Notre généralisation au cas multi-niveaux des mesures de qualité additives répond à nos attentes car une même arête va être visitée plusieurs fois : une fois en tant qu'arête interne à $G[C_i]$, puis à $G[C_{ij}]$ et ainsi de suite. A mesure que l'on effectue la récursion, différentes puissances de q sont collectées. Cette approche qu'on peut appeler "le poids dépendant de la hauteur" est également valable pour les arêtes externes. Il faut également noter que le gain d'un nœud t à $\Phi(G, T; q)$ est pondéré par le produit des gains de ces ancêtres dans T. Ainsi, un nœud correspondant à un groupe de mauvaise qualité ne pourra engendrer qu'une sous-hiérarchie de mauvaise qualité.

Le cas où q est proche de 1 correspond à la situation extrême où le poids d'une arête (interne) est égale à sa hauteur maximale dans la hiérarchie. De la même manière, pour $q = 0$, la fonction $\Phi(G, T; q)$ est égale à la mesure de qualité sur le premier niveau de T. Comme nous le verrons dans la partie 4.3, la valeur donnée au paramètre q joue un rôle en permettant de favoriser les hiérarchies plus ou moins profondes.

Comparer deux partitions hiérarchiques en se basant sur des polynômes est difficile pour prendre une décision. Dans le cas où il n'y a pas de raison *a priori* de privilégier une décomposition plus ou moins haute, une solution possible est de prendre la valeur moyenne de $\Phi(G, T; q)$ sur $q \in [0, 1]$. Cette mesure est notée $\overline{\Phi}(G, T)$. Pour la calculer, il suffit de prendre l'intégrale de la mesure, que l'on peut formuler de la façon suivante en utilisant l'équation 7 :

$$\begin{aligned} \overline{\Phi}(G, T) &= \int_0^1 \Phi(G, T; q) dq \\ &= \sum_{p=0}^{h(T)-1} \frac{1}{p+1} \sum_{w \in \mathcal{D}_p} \prod_{u \prec w} \phi(G, \{C_v\}_{v \in \sigma_T(p(u))}, u) \end{aligned} \tag{8}$$

La mesure définie par l'équation 8 est à valeur dans $[0, 1]$. La contribution de chaque niveau $\mathcal{D}_p$ est inversement proportionnelle à la hauteur à laquelle les groupes de ce niveau se situent.

4.3 Validation expérimentale de la généralisation de MQ

Dans cette partie, une validation expérimentale de la généralisation multi-niveaux de MQ (voir Définition 3.6) est proposée. Cette généralisation de MQ pour un graphe G et pour un arbre de partition T est notée $MQ(G, T; q)$. Les conclusions faites dans cette partie se limitent donc à cette mesure. Nous montrons également que cette mesure permet de comparer différentes partitions hiérarchiques sur un exemple réel muni d'une vérité terrain.

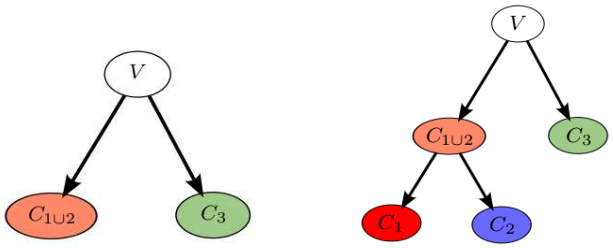
Les graphes traités dans les deux premiers exemples sont simples non-orientés et sans boucle. Nous utilisons le graphe complet et biparti complet comme référence, prenant ainsi $\delta_i = \binom{|C_i|}{2}$ et $\delta_{ij} = |C_i| \cdot |C_j|$ (voir Section 3.2). Nous utiliserons également la taille d'un groupe C_i pour définir son poids ($x_i = |C_i|$).

Les exemples utilisés par la suite suivent une construction simple et peuvent être vus comme une adaptation du modèle ERMG (voir description dans la Section 2.6.2) pour des graphes avec une structure de communautés multi-niveaux. Nous utilisons des cliques (graphes complets) comme briques élémentaires de la hiérarchie. Ces groupes sont associés aux feuilles dans les différents arbres de partitions étudiés. Ainsi, la différence entre ces hiérarchies dépend de la façon dont ces cliques sont regroupées. En effet, découper plus en profondeur une clique n'améliorera pas la valeur de MQ (voir partie 3.3). En outre, les exemples sont conçus de façon à ce que l'on puisse facilement prédire la "meilleure hiérarchie".

4.3.1 Un cas simple avec trois cliques

Bien que très simpliste, notre premier exemple illustre la difficulté de trouver une hiérarchie appropriée en utilisant des mesures de partitions "plates".
On considère que G est formé de trois cliques distinctes C_1, C_2, C_3 chacune de taille n. Les cliques C_1 et C_2 sont connectées par bn^2 arêtes avec $0 \leq b \leq 1$. Il n'y aucune arête entre C_3 et le reste du graphe. Cet exemple nous permet de comparer analytiquement différentes partitions hiérarchiques du graphe en utilisant la généralisation multi-niveaux de MQ. Nous commençons par comparer les arbres $T = [C_{1\cup 2}, C_3]$ et $T' = [[C_1, C_2], C_3]$ (voir Figure 4.4). Nous utilisons ici des expressions parenthésées : T est une partition à un

niveau dont le premier contient l'union de C_1 et C_2 tandis que T' divise dans un second niveau $(C_1 \cup C_2)$ en $[C_1, C_2]$.



(a) T (b) T'

FIGURE 4.4: Deux partitions hiérarchiques d'un graphe formé de trois cliques.

Les deux arbres coïncident sur le premier niveau, comparer leur qualité revient donc à décider si il est profitable de diviser $C_{1\cup2}$ en $[C_1, C_2]$. Le ratio de connectivité interne de $C_{1\cup2}$ est (voir l'équation 11)

$$\alpha_{1\cup2} = \frac{2n(1+b)-2}{2n-1}$$

L'arbre T est plat, la mesure MQ est donc constante (en tant que fonction de q). Nous allons comparer asymptotiquement les deux mesures par rapport à la taille des cliques n afin d'obtenir des expressions, notées $MQ^\infty(G,T;q)$, qui ne dépendent que des variables q et b.

$$\begin{aligned} MQ^\infty(G,T;q) &= \frac{2}{3} + \frac{b}{3} \\ MQ^\infty(G,T';q) &= \frac{1}{3}\left(1 + (1+b)[1+q(1-b)]\right) = \frac{1}{3}\left(2 + b + q(1-b^2)\right) \end{aligned}$$

Notons qu'on retrouve bien $MQ^\infty(G,T;0) = MQ^\infty(G,T';0)$ en raison de l'égalité des premiers niveaux. La comparaison des deux partitions repose donc sur le coefficient :

$$[MQ^\infty(G,T';q), q] = \frac{q(1-b^2)}{3}$$

Cette quantité est positive et fonction décroissante de b. Ceci confirme un phénomène évident : tant que C_1 et C_2 ne sont pas fortement connectés, il est raisonnable de diviser $C_{1\cup2}$ en deux groupes alors qu'ils devraient rester groupés lorsque le ratio b devient grand.

Nous allons également comparer ces deux arbres de partition avec la configuration $T'' = [C_1, C_2, C_3]$. On a

$$MQ^\infty(G,T'';q) = 1 - \frac{b}{3} \tag{8}$$

qui est évidemment une fonction décroissante de b. Si $b = 0.5$, T et T'' ont des valeurs de MQ égales. Remarquons que les qualités de T' et T'' se chevauchent pour $b \in [0, 0.5]$. Dans ce cas, une grande valeur de q rend la partition hiérarchique préférable. On peut constater que

$$\forall b \in [0, 0.5],\ MQ^{\infty}(G, T'; q) = MQ^{\infty}(G, T''; q) \Leftrightarrow q = \frac{2b - 1}{b^2 - 1}, \tag{8}$$

Cette fonction décroît de façon quasi-linéaire par rapport à b. Cela signifie que plus b est grand, moins il est nécessaire d'augmenter q pour choisir T' comme la meilleure partition.

Pour des valeurs de b comprises entre 0.25 et 0.5, T' peut être considérée comme la meilleure partition. C'est effectivement le cas si on considère l'intégrale de MQ telle que définie par l'équation 8. Cette mesure donne T' comme la meilleure configuration pour $b \in [0.25, 0.5]$ alors qu'on a $MQ^{\infty}(G, T''; 0) > MQ^{\infty}(G, T'; 0)$. Ainsi, pour ces valeurs de b, tout algorithme permettant de maximiser la mesure MQ pour une partition plate donnera pour résultat T'' comme meilleure configuration. L'agrégation de C_1 et C_2 sera rejetée car $MQ^{\infty}(G, T; 0) < MQ^{\infty}(G, T''; 0)$. Dans ce cadre, l'utilisation itérative d'un algorithme de maximisation de MQ n'aboutirait pas à la construction de l'arbre T' bien que celui-ci soit la meilleure configuration (selon les critères sous-jacents à MQ).

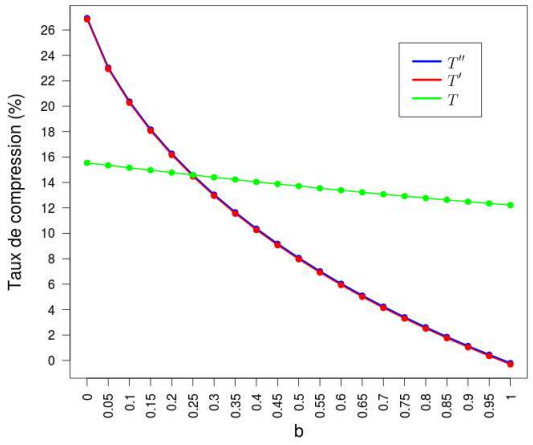


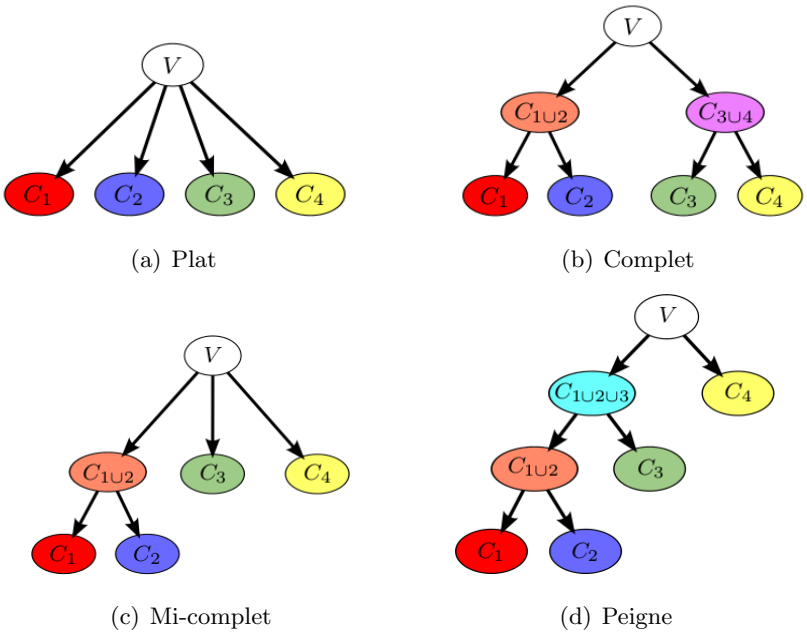
FIGURE 4.5: Valeurs de $\Delta L(T)$ pour $n = 100$ calculées pour différentes valeurs de b.

Le taux de compression ΔL (voir Définition 4.4) ne classe jamais la configuration T' comme la meilleure partition sur cet exemple (voir Figure 4.5). Deux remarques peuvent être faites quant à l'utilisation de ΔL. Premièrement, $\Delta L(T')$ a des valeurs très proches de $\Delta L(T'')$ quelque soit la valeur de b cependant on a toujours $\Delta L(T'') > \Delta L(T')$. L'agrégation de C_1 et C_2 ne semble donc pas avoir un impact important si ces deux cliques sont,

au final, séparées dans la hiérarchie. Deuxièmement, le taux de compression de la configuration T décroit à mesure que b augmente, ce qui est contre-intuitif. La configuration T obtient le meilleur taux de compression à partir de $b \simeq 0.25$. On peut donc dire que la mesure ΔL ne fournit pas des comparaisons cohérentes sur cet exemple simple.

4.3.2 Un exemple plus complexe

Nous étudions maintenant différents arbres de partition pour des graphes comprenant quatre cliques et montrons que MQ prédit correctement la meilleure configuration, dépendant des ratios de connectivité entre les groupes.
Nous comparons quatre arbres de partition : l'arbre *plat*, l'arbre *mi-complet*, l'arbre *complet* et l'arbre *peigne*. Une illustration de ces arbres est disponible en Figure 4.6. Dans cette exemple, les feuilles des quatre arbres correspondent à quatre cliques C_1, C_2, C_3 et C_4 de taille n. On pose b_{ij} comme étant le ratio de connectivité entre les groupes C_i et C_j. Nous allons étudier quatre cas différents pour lesquelles on posera toujours $b_{14} = 0 = b_{24} = 0$ (le groupe C_4 est toujours déconnecté de C_1 et C_2).

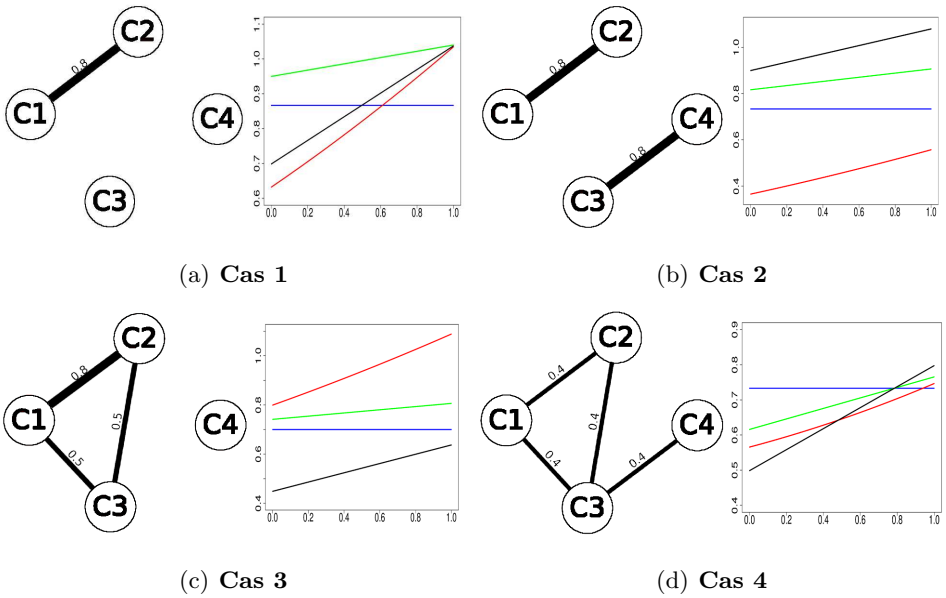


(a) Plat (b) Complet

(c) Mi-complet (d) Peigne

FIGURE 4.6: Différentes partitions hiérarchiques de quatre cliques.

La Figure 4.7 présente les courbes pour les quatre cas que nous étudions. Pour chaque cas, l'image de gauche est une représentation du graphe quotient associé à la partition en quatre cliques. Les étiquettes des arêtes indiquent les ratios d'inter-connectivité entre les deux cliques qu'elles connectent (il n'y pas d'arête lorsque le ratio est de 0). Les graphiques de droite illustrent les polynômes de MQ pour la partition plate (en bleue), mi-complète



(a) **Cas 1** (b) **Cas 2**

(c) **Cas 3** (d) **Cas 4**

FIGURE 4.7: Comparaison de différentes partitions hiérarchiques pour des graphes composés de quatre cliques.

(verte), complète (noire) et oblique (rouge). Notons que le polynôme associé à la partition plate est constant, ceux associés aux partitions complètes et mi-complètes sont de degré un et de degré deux pour la partition peigne. Différentes conclusions peuvent être faites sur cette expérience :

- **Cas 1 :** Quand b_{12} est plus grand que tout autre b_{ij}, la mesure MQ classe la partition mi-complète comme la meilleure.
- **Cas 2 :** Quand b_{12} et b_{34} sont plus grands que tout autre b_{ij}, alors la partition complète apparaît comme la meilleure configuration.
- **Cas 3 :** La partition peigne devient la meilleure candidate quand les ratios de connectivité vérifient $b_{12} \gg b_{13} \simeq b_{23} \gg$ les autres b_{ij}.

Dans le **cas 4**, les courbes des polynômes se chevauchent. Ce cas correspond en réalité à une situation où la meilleure partition n'est pas évidente. La variable q peut être utilisée ici pour favoriser (quand proche de 1) ou pénaliser (quand proche de 0) les hiérarchies profondes. Prendre un q entre 0 et 0.75 revient à classer la partition plate comme la meilleure. D'un autre côté, une valeur proche de 1 mène au choix de la partition complète. En utilisant l'intégrale de MQ (voir équation 8), la partition plate est donnée comme celle de meilleure qualité.

Les différents cas développés ici montrent que la mesure MQ multi-niveaux permet de déterminer la meilleure configuration possible.

4.3.3 Comparaison de différentes hiérarchies sur un réseau réel

Nous allons maintenant montrer que la mesure MQ multi-niveaux permet de comparer différentes partitions hiérarchiques d'un réseau réel. Le graphe décrit les rencontres entre les équipes de football américain universitaire issues de la division I-A sur la saison 2000 [64]. Ce jeu de données est régulièrement utilisé pour tester les algorithmes de partitionnement. Il contient 115 sommets (les équipes) et 613 arêtes (les rencontres) avec un degré moyen de 10.66.

L'intérêt de ce réseau est qu'il dispose d'une partition hiérarchique connue qui peut être considérée comme une vérité terrain. En effet, les équipes universitaires sont partagées en 11 conférences différentes. Trois d'entre elles (à savoir la *"Big Twelve"*, la *"South Eastern"* et la *"Mid-American"*) se décomposent chacune en deux sous-divisions. Nous disposons donc d'une partition hiérarchique à deux niveaux qui n'est pas équilibrée (certaines conférences sont sous-divisées et d'autres non). Bien que les rencontres sont plus susceptibles d'apparaître au sein d'une même conférence, elle dépendent également de la proximité géographique des équipes.

Nous comparons, en utilisant la mesure MQ, cette vérité terrain à trois autres partitions hiérarchiques obtenues en utilisant trois algorithmes de partitionnement de graphes différents. Les deux premiers algorithmes fournissent seulement des partitions plates de l'ensemble des sommets. Pour obtenir des partitions hiérarchiques, on rappelle les algorithmes récursivement sur chaque groupe détecté à l'étape précédente. Cette solution ainsi que les algorithmes utilisés sont détaillés plus en détail dans la Section 5.1.
Le premier algorithme correspond à un *single-linkage clustering* [54] où l'indice de Jaccard [73] est utilisé pour déterminer le poids d'une arête : il va correspondre à la proportion de triangles contenant l'arête en question. Le seuil optimal (tel que la partition correspondante maximise la qualité) est déterminé en utilisant la mesure MQ. On nommera cet algorithme **Single-Linkage**
Le second algorithme **MLR-MCL** [130] est une version modifiée de l'algorithme *MCL* [140].
Finalement, le troisième algorithme est l'algorithme **Louvain** [23] qui produit une partition hiérarchique bien que seul le premier niveau de la hiérarchie soit habituellement utilisé.
Nous utilisons pour les deux derniers algorithmes des implémentations proposées par les auteurs respectifs. Toutes les méthodes utilisées ici sont non-supervisée et ne requièrent pas de paramétrage.

Ici encore nous utilisons le graphe complet et le graphe biparti complet comme graphes de références pour les ratios de connectivité interne et externe. Le poids des groupes sera ici unitaire *i.e.* ne dépendra pas de la taille des groupes (nombre d'équipes). Toutefois,

pour garantir l'additivité des poids, on utilise les valeurs suivantes :

$$x_w = \begin{cases} 1 \text{ si } w \in \mathcal{F}(T) \\ |\mathcal{F}(T_w)| \text{ sinon} \end{cases}$$

où $\mathcal{F}(T_w)$ est l'ensemble des feuilles de T ayant w comme ancêtre commun.

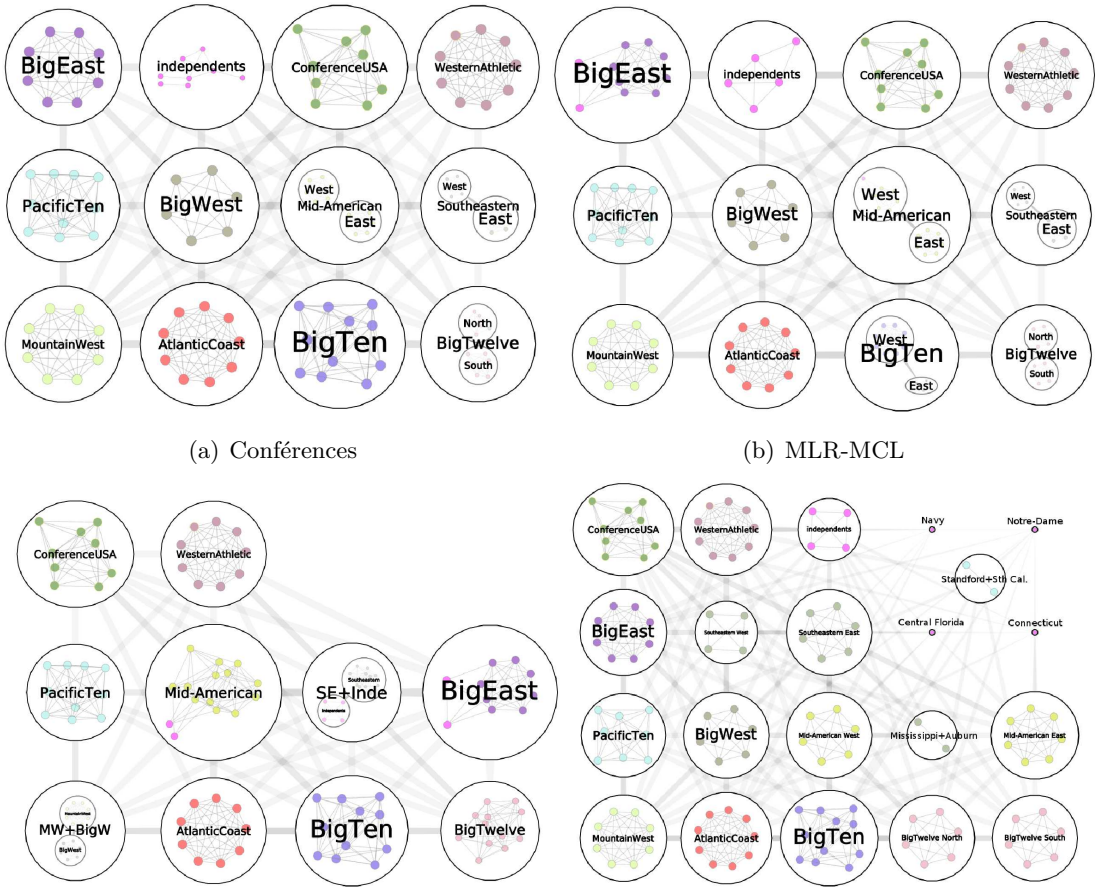


FIGURE 4.8: Visualisation des quatre partitions hiérarchiques du réseau de football universitaire basée sur des graphes quotients. Chaque méta-nœud correspond à un groupe de la partition. Les étiquettes des sommets indiquent les conférences d'origine des sommets se trouvant dans le groupe correspondant.

La Figure 4.8 permet de comparer les différentes partitions. Comme on peut s'y attendre, les algorithmes s'accordent sur une majorité de groupes qui correspondent aux vraies conférences. Cela s'explique par le fait que les équipes jouent beaucoup plus avec les autres équipes issues de la même conférence au cours d'une saison. Ainsi, la connectivité interne des conférences est relativement forte.

L'algorithme **Louvain** (voir Figure 4.8(c)) tend à grouper différentes conférences. Ces regroupements sont pertinents car ils reflètent la proximité géographique entre ces conférences. Ainsi, les équipes de la *"Mountain West"* et la *"Big West"* appartiennent au même

groupe sur le premier niveau de la hiérarchie.
Le partitionnement obtenu en utilisant l'algorithme **Single-Linkage** ne contient qu'un seul niveau (voir Figure 4.8(d)). Celui-ci comprend un nombre relativement important de groupes qui correspondent dans certains cas aux conférences et dans d'autres à des sous-divisions de conférences.
L'algorithme **MLR-MCL** fournit une partition à deux niveaux (voir Figure 4.8(b)) qui est proche de la vérité terrain (voir Figure 4.8(a)). En effet, le premier niveau correspond aux conférences (hormis pour les équipes indépendantes qui n'appartiennent pas à une conférence) et plusieurs de ces groupes sont eux-mêmes divisés. En particulier, on retrouve le bon découpage pour les conférences contenant vraiment des sous-divisions.

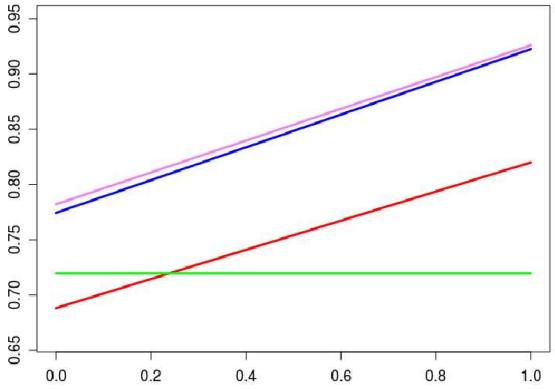


FIGURE 4.9: Courbes de MQ pour le partitionnement en conférences (bleu), Single-Linkage (vert), MLR-MCL (violet) et Louvain (rouge).

Algorithme	$MQ(G,T;q)$	$\overline{MQ}(G,T)$	Taux Compression $\Delta L(T)$
MLR-MCL	$0.782 + 0.144q$	0.854	15.569
Conférences	$0.774 + 0.148q$	0.848	16.379
Louvain	$0.688 + 0.132q$	0.754	18.528
Single-linkage	0.72	0.72	13.384

TABLE 4.1: Comparaison des résultats sur le réseau de football universitaire par rapport aux polynômes de MQ (équation 6) et leur intégrale $\overline{MQ}$ (équation 8) ainsi que le taux de compression $\Delta L(T)$ (voir Définition 4.4). Les valeurs sont données au millième près.

Les courbes des polynômes MQ obtenus pour chaque partition sont données dans la Figure 4.9. Le Tableau 4.1 fournit un classement des quatre partitions en fonction des valeur moyennes $\overline{MQ}$ obtenues en utilisant l'équation 8.

Comme indiqué ci-dessus, le résultat fourni par **MLR-MCL** est proche de la vérité terrain. Ceci se reflète dans les valeurs de MQ multi-niveaux pour ces deux partitions.

En effet, les courbes (voir Figure 4.9) sont très proches. Il apparaît cependant que **MLR-MCL** fournit une partition de meilleure qualité.
Le dernier niveau dans la hiérarchie produite par l'algorithme **Louvain** correspond bien à la partition en conférences des équipes. Toutefois, regrouper ces conférences à un niveau supérieur est fortement pénalisé par MQ. On peut cependant remarquer que cette partition est préférable au partitionnement obtenu en utilisant le **Single-Linkage** pour une valeur de q supérieure à 0.22 environ.

Nous pouvons également comparer les différentes partitions en utilisant une autre mesure multi-niveaux, le taux de compression $\Delta(L(G,T))$ (voir troisième colonne du Tableau 4.1). Le classement induit par l'utilisation de cette mesure est **Louvain**, **Conférences**, **MLR-MCL** et **Single-Linkage**. Les trois premières positions sont donc inversées. Ce critère associe une plus forte valeur à une partition qui ne sous-divise pas des conférences, ce qui explique pourquoi **Single-Linkage** est toujours classé dernier. On peut en conclure que les deux mesures sont différentes sur cet exemple bien que la vérité terrain semble dans les deux cas être un bon compromis.

4.4 Discussion et Perspectives

Nous avons présenté une mesure de qualité multi-niveaux permettant d'évaluer l'adéquation entre une partition hiérarchique des sommets du graphe et sa topologie. Cette mesure a l'avantage de généraliser des mesures de qualité existantes qui étaient réservées à l'évaluation de partitions plates. Nous associons à un arbre de partition un polynôme à une variable dont le degré correspond à la hauteur de l'arbre. Nous avons également proposé deux formulations équivalentes de cette mesure en se basant sur la notion de chemin pondéré dans l'arbre.

Nous avons ainsi validé la généralisation hiérarchique de la mesure MQ qui tient compte de la connectivité interne et externe de chaque groupe. Les exemples utilisés illustrent un point important malgré leur simplicité. On a en effet vu que la recherche de la meilleure partition hiérarchique par des procédures itératives pouvait dans certains cas échouer. Ceci est particulièrement vrai lorsque certains niveaux de la meilleure hiérarchie ne sont pas de bonne qualité lorsque évalués indépendamment. Ceci justifie l'emploi d'une mesure de qualité hiérarchique. En utilisant un réseau réel muni d'une structure de communautés à deux niveaux, nous avons comparé différentes partitions hiérarchiques et montré que notre mesure permettait de choisir une partition très proche de la vérité terrain.

Les définitions que nous avons proposées ici sont génériques à l'instar de la mesure de résolution multi-niveaux proposée par Pons *et al.* [110]. Nous avons également comparé la

généralisation hiérarchique de la mesure MQ à la mesure proposée par Rosvall *et al.* [124], les deux mesures ne donnent pas les mêmes résultats lorsque l'on cherche à comparer différentes partitions hiérarchiques. Toutefois, cette différence s'explique peut-être par le choix de MQ. On peut s'attendre à obtenir des résultats plus proches en utilisant une version multi-niveaux de la modularité.

En effet, nous nous sommes ici surtout focalisés sur la définition multi-niveaux de la mesure MQ. D'autres mesures de qualité peuvent être adaptées au cas multi-niveaux (comme la modularité) et on peut effectivement utiliser ces généralisations dans le cadre d'algorithmes de partitionnement de graphe. Ceci est l'objet du prochain chapitre.

Chapitre 5

Algorithmes pour le partitionnement hiérarchique de graphes

Les problèmes liés au partitionnement plat des sommets d'un graphe sont souvent difficiles car l'espace de recherche augmente de manière exponentielle par rapport aux nombre de sommets. La recherche de partitions hiérarchiques ne facilite donc pas les choses. En effet, le nombre de dendrogrammes de n feuilles est de l'ordre de $(2n-3)!!$ où !! désigne la double factorielle. Flajolet et Sedgewick [51] proposent une formule pour le nombre d'arbres de partitions à n feuilles, qui évolue encore plus rapidement (voir OEIS **A000311**). Les arbres de partitions contiennent bien sûr les partitions plates et les dendrogrammes.

De la même façon que la recherche d'un partitionnement plat peut être réalisée en parcourant des partitions induites par une structure hiérarchique, nous allons ici étudier des méthodes pour obtenir une partition hiérarchique (exploitable par un utilisateur) extraite d'une partition hiérarchique de plus grande taille. Pour guider cette exploration nous allons utiliser la généralisation multi-niveaux des mesures de qualité additives présentée dans le chapitre 4.

Ce chapitre est organisé de la façon suivante. Dans la section 5.1 nous présentons différentes méthodes basées sur un partitionnement hiérarchique des sommets. Dans cet état de l'art, nous montrons notamment que les différents algorithmes peuvent être séparés en plusieurs familles. La section 5.2 présente une application des mesures de qualité multi-niveaux. L'algorithme proposé permet d'optimiser une hiérarchie des sommets donnée en supprimant itérativement des nœuds internes de la hiérarchie [115]. Une application de cette méthode à la hiérarchie produite par l'algorithme *Louvain* [23] est donnée dans la section 5.3. Nous étudions par la suite une technique basée sur le filtrage d'arêtes et appliquée à un réseau de flux domicile-travail. Enfin, une discussion sur les méthodes proposées et la présentation de différentes perspectives sont faites dans la section 5.5.

5.1 État de l'art

Dans cette section, nous présentons différents algorithmes de partitionnement de graphes qui se basent sur une structure hiérarchique. Dans la plupart des cas, les méthodes proposées n'ont pas pour finalité de renvoyer une hiérarchie mais une partition à un niveau des sommets du graphe. Le partitionnement hiérarchique est ainsi souvent utilisé comme étape intermédiaire. Ce n'est toutefois pas le cas de tous les algorithmes créés pour répondre au problème du partitionnement de graphes. Cette section ne couvre donc pas de façon exhaustive les méthodes connues pour répondre à ce problème. Des états de l'art plus complets sur ce sujet existent (voir entre autres [131, 54, 105]).

Les approches présentées ici peuvent être regroupées en différentes familles : les méthodes agglomératives (section 5.1.1), divisives (section 5.1.2) et hybrides (section 5.1.3) qui empruntent aux deux premières. Enfin d'autres approches intéressantes mais utilisées pour répondre à des problèmes légèrement différents sont détaillées dans la section 5.1.4.

5.1.1 Approches agglomératives

Les approches agglomératives ont en commun l'idée qu'un groupe d'individus peut être considéré comme un nouvel individu, qui peut être à son tour groupé à d'autres individus. La procédure générique correspondante est détaillée dans l'algorithme 1.

Algorithme 1: *AggloGen*(G) un algorithme agglomératif générique pour la partition hiérarchique des sommets d'un graphe.

Entrées : $G = (V, E)$, un graphe
Sorties : T, une partition hiérarchique de V
$G' \leftarrow G$;
$T \leftarrow \emptyset$;
$\mathcal{C} \leftarrow partition(G')$;
tant que $\neg$*arret(G',C)* **faire**
  $T \leftarrow ajouterPremierNiveau(T, \mathcal{C})$;
  $G' \leftarrow grapheQuotient(G', \mathcal{C})$;
  $\mathcal{C} \leftarrow partition(G')$;
fin
retourner T;

Dans le cas du partitionnement de graphes, un groupe de sommets C est remplacé par un *méta-sommet*. Un tel méta-sommet peut être obtenu en contractant les arêtes internes

à un groupe C, le nouveau graphe G' obtenu est un graphe quotient (voir définition 2.49). La génération de ce graphe est faite en utilisant la fonction $grapheQuotient(G', \mathcal{C})$. Selon les cas, ce graphe peut être associé à une fonction de pondération sur les méta-arêtes (par exemple la somme des poids des arêtes de G correspondante) et/ou une fonction de pondération sur les méta-sommets (par exemple le nombre de sommets de G correspondant). Notons que ce graphe quotient est susceptible de contenir des boucles pondérées permettant de garder en mémoire le nombre d'arêtes internes à chaque groupe.
Un algorithme de partitionnement (correspondant ici à la fonction $partition(G')$) est appelé itérativement jusqu'à ce qu'un critère d'arrêt (fonction $arret(G', \mathcal{C})$) soit vérifié. À chaque itération, la partition des méta-sommets de G est ajoutée en tant que premier niveau ($N_1(T)$) de l'arbre de partition T.

Nous allons maintenant détailler les travaux présentant des méthodes qui suivent ce paradigme.

Compression multi-échelle Un algorithme agglomératif peut être utilisé pour réduire significativement le temps de calcul d'une partition des sommets. Cette phase de pré-traitement est utilisée dans l'algorithme *Metis* [76]. L'idée est de contracter itérativement des ensembles d'arêtes jusqu'à obtenir un graphe quotient de petite taille. Une partition des méta-sommets sur ce dernier graphe est réalisée, cette partition va également correspondre à une partition des sommets "de base" du graphe.
La même méthode est utilisée par l'algorithme *MLR-MCL* [130] qui se base sur l'algorithme de partitionnement *MCL* [140]. Ce dernier algorithme est très performant mais avec un coût algorithmique important. En effet, il repose sur des multiplications de matrices stochastiques.
Pour réaliser cette compression multi-échelle, $partition(G)$ correspond, à chaque étape, aux composantes connexes du sous-graphe $G(L)$ formé par un ensemble d'arêtes L. L'ensemble L forme un couplage de G (chaque sommet de G est connecté à au plus une arête de L). Il peut être choisi selon différents critères : aléatoirement, selon un poids, selon la taille de la clique maximale à laquelle les arêtes appartiennent *etc.* Le critère d'arrêt dans ce cas dépend de la taille de G'. Cette méthode est généralement très rapide puisque l'objectif est de partitionner des graphes de grande taille (qui ne tiennent pas en RAM). La complexité de la procédure dépend de la façon dont les ensembles L sont choisis.

Notons qu'une version simplifiée de cette méthode est la contraction d'une seule arête tirée aléatoirement à chaque étape. C'est exactement la procédure qui est mise en œuvre dans l'algorithme stochastique de recherche d'une coupe minimum de Karger [74]. En effet, si on contracte une arête $e \in E(G)$ tirée uniformément, la probabilité d'augmenter la taille de la coupe minimum de G (voir définition 2.48) est très faible si G est de grande taille. Dans l'algorithme de Karger, les compressions sont répétées jusqu'à ce que G' ne contienne

plus que deux sommets. La valeur de la coupe minimum est alors donnée par le nombre d'arêtes entre ces deux sommets. Les méthodes de compression multi-échelles développées ici sont ainsi justifiées par le fait que les contractions réalisées à chaque étape sont, avec une forte probabilité, des contractions d'arêtes internes aux communautés. Toutefois, ne connaissant pas *a priori* la forme ou la taille des communautés, la définition d'un critère d'arrêt pose problème.

***Clustering* hiérarchique** Cette approche n'est pas spécifique aux données relationnelles comme les graphes. Pour un ensemble d'individus (dans notre cas, les sommets), on peut définir des *distances* entre eux. Dans ce cadre, $partition(G)$ contient $|V|-1$ groupes car seuls les deux sommets les plus proches sont regroupés. L'algorithme s'arrête lorsque G' ne contient qu'un unique sommet. La partition T retournée correspond à un arbre de partition binaire (un dendrogramme). La complexité de ces méthodes est $\mathcal{O}(n^2)$ en utilisant les structures de données appropriées [134].

La proximité (ou la distance) entre deux groupes de sommets (ou deux méta-sommets) peut être définie de plusieurs façons : le minimum des distances entre les éléments du groupe (on parle alors de *single-linkage clustering*), le maximum des distances (*complete-linkage clustering*) ou encore la moyenne des distances (*average-linkage clustering*). Un critère très utilisé en analyse de données est le critère de Ward [146] qui cherche à minimiser la somme des carrés des distances entre les individus (les sommets du graphe ici) et le *centroïd* (le méta-sommet) dont la position est définie par ces mêmes distances. Dans le cas du partitionnement de graphes, Donetti *et al.* [43] utilisent par exemple une distance basée sur les vecteurs propres de la matrice laplacienne du graphe.

Optimisation de la modularité Les algorithmes décrits ici ont été conçus pour optimiser la modularité d'une partition des sommets d'un graphe. Cette mesure de qualité est présentée dans la section 3.1. On peut toutefois remarquer que, pour chacune de ces méthodes, d'autres fonctions objectives pourraient être utilisées.

Dans [99], Newman propose un algorithme glouton d'optimisation de la modularité en cherchant, à chaque étape, les deux groupes (méta-sommets) dont la fusion augmente le plus (ou diminue le moins) la modularité. La procédure s'arrête ici lorsqu'il ne reste qu'un unique groupe et l'arbre de partition est un dendrogramme.
La complexité de cet algorithme est $\mathcal{O}(|E||V|)$. Des améliorations de cette méthode, permettant d'en réduire le coût algorithmique, ont été proposées dans [33] et [144]. Une partition des sommets peut être obtenue en cherchant la coupe dans le dendrogramme généré qui maximise la modularité.

Une heuristique pour la maximisation de la modularité a été proposée par Blondel *et al.* [23]. Cet algorithme est généralement connu sous le nom d'*algorithme de Louvain*. La

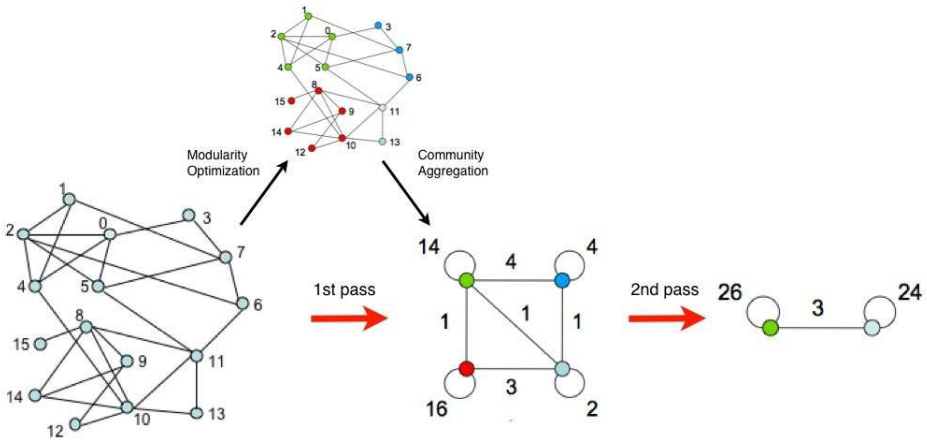


FIGURE 5.1: Illustration de l'approche agglomérative pour l'optimisation de la modularité dans l'algorithme *Louvain* (image tirée de [23]).

figure 5.1 illustre bien l'approche agglomérative utilisée par les auteurs.
La fonction $partition(G')$ correspond ici à un maximum local de la modularité. Cette partition est obtenue en traitant chaque sommet (ou méta-sommet) dans un ordre aléatoire et en l'affectant au groupe présent dans son voisinage pour lequel cette opération augmente le plus la modularité. La modularité étant une mesure fortement additive, le gain lié à ce changement de groupe peut être calculé rapidement.
L'algorithme *Louvain* s'arrête lorsque la dernière partition découverte a une modularité plus faible que la précédente. Au final, l'arbre de partition T est dans ce cadre un arbre de partition quelconque et pas forcément un dendrogramme. La partition plate retournée par l'algorithme correspond au premier niveau ($N_1(T)$) de cette hiérarchie car c'est le niveau pour lequel la modularité est maximale.

La fonction $partition(G')$ employée par l'algorithme *Louvain* est similaire aux méthodes basées sur la *propagation d'étiquettes* [118, 92] dans lesquelles un sommet est affecté au groupe dans son voisinage avec lequel il a le plus de connections. Ces heuristiques, ainsi que celle de [23], ont connu un grand succès ces dernières années en grande partie grâce à leur rapidité dans les temps de calcul. À notre connaissance, ce type d'algorithme a été proposé pour la première fois par Biemann [20] sous le nom de *"téléphone arabe"* (*Chinese whispers* en anglais).

5.1.2 Approches divisives

Une autre manière intuitive pour obtenir un partitionnement hiérarchique de graphes est d'appliquer récursivement un algorithme de partitionnement plat à chaque groupe détecté à l'étape précédente (voir l'algorithme générique 2). Notons que ce type d'approche

est différent de l'approche agglomérative. En effet, chaque sous-graphe induit $G[C]$ est considéré comme un nouveau graphe à part entière. En particulier, les connections entre $(V \setminus C)$ ne sont plus prises en compte. C'est cette procédure qui est utilisée pour générer des partitions hiérarchiques dans la section 4.3.

Algorithme 2: *AggloDiv*(G) un algorithme divisif générique pour la partition hiérarchique des sommets d'un graphe.

Entrées : $G = (V, E)$, un graphe
Sorties : T, une partition hiérarchique de V
$T \leftarrow (r, \emptyset)$;
$\mathcal{C} \leftarrow partition(G)$;
si $\neg$*arret(G,C)***alors**
  pour tous les $C \in \mathcal{C}$ **faire**
    $T' \leftarrow AggloDiv(G[C])$;
    $T \leftarrow ajouterSousArbre(T, T')$;
  fin
fin
retourner T;

La majorité des algorithmes dit "divisifs" que l'on trouve dans la littérature sont basés sur une approche commune. Premièrement, une métrique est calculée sur les arêtes. Cette métrique a pour but de séparer les arêtes qui sont susceptibles d'appartenir à une communauté et les autres. Deuxièmement, on retire du graphe l'arête la moins susceptible d'appartenir à une communauté. On applique ces deux étapes sur chaque composante connexe jusqu'à avoir des sommets isolés. Le résultat de ce type de procédure est donc un dendrogramme. Les algorithmes divisifs se différencient principalement sur la mesure employée permettant de déterminer la prochaine arête à retirer.

Mesure de Centralité Girvan et Newman [63, 101] proposent une approche basée sur la centralité-chemin (voir définition 2.28). En effet, une arête avec une forte centralité relie probablement deux communautés. D'autre types de "centralité" ont été également utilisés tels que la *centralité d'information* [56]. Cette mesure évalue, pour une arête donnée, la diminution dans les distances moyennes entre toutes les paires de sommets liées à la suppression de cette arête.
Ces mesures de centralité capturent l'importance globale de l'arête. La mise à jour des valeurs pour chaque arête a un coup important : $\mathcal{O}(|E||V|)$ pour la centralité-chemin et $\mathcal{O}(|E|^2|V|)$ pour la centralité d'information.

Mesure sur les cycles Des mesures utilisant des critères locaux ont été proposées dans

le cadre d'algorithmes divisifs. Dans [117], les auteurs généralisent la notion de coefficient de *clustering* aux arêtes. Cette mesure correspond à la proportion observée de triangles qui contiennent l'arête en question. La mise à jour des valeurs est plus rapide que pour les mesures de centralité puisque la suppression d'une arête affecte seulement les arêtes ayant une extrémité en commun avec l'arête en question.

Auber *et al.* [14] utilisent une mesure basée non seulement sur la proportion de triangles (cycle de longueur 3) mais aussi sur la proportion de cycles de longueur 4. Toutefois, l'algorithme proposé diffère des autres algorithmes divisifs présentés ici. Il consiste à trouver le seuil t tel que la partition obtenue en supprimant les arêtes dont la mesure est inférieure à t maximise la fonction de qualité MQ présentée dans la section 3. Dans ce cadre, la procédure équivaut à chercher le meilleur niveau dans la hiérarchie obtenue par l'utilisation d'un *single-linkage clustering*. Cette méthode correspond à la fonction *partition* de l'algorithme 2. Les auteurs produisent une partition hiérarchique en l'appliquant récursivement.

5.1.3 Approches hybrides

Infomap Rosvall *et al.* [124] ont introduit une mesure permettant d'évaluer la qualité d'une partition hiérarchique de graphe. Cette mesure est présentée en détail dans la section 3.1.
La procédure proposée par les auteurs pour obtenir une partition hiérarchique de graphe peut être qualifiée d'*hybride* dans le sens où elle alterne les phases d'agrégations (telle que définies dans l'algorithme 1) et de divisions (algorithme 2). La phase agglomérative est proche des algorithmes de *Louvain* ou de *propagation d'étiquettes* présentés plus haut. Chaque groupe est ensuite traité comme un graphe à part entière et est redivisé. À l'issue de cette procédure, des groupes entiers peuvent être déplacés dans la hiérarchie afin d'augmenter la qualité de la partition.

5.1.4 Autres approches

Modèle Statistique Hiérarchique Le modèle probabiliste présenté dans la section 2.6.4 peut être utilisé pour obtenir un arbre de partition binaire. Le dendrogramme reflétant la topologie du graphe est obtenu en maximisant la vraisemblance du modèle. L'algorithme proposé dans [31] est une heuristique de type Monte-Carlo par Chaîne de Markov (MCMC).

Qualité avec un paramètre de résolution Dans la section 4.1, nous avons présenté la méthode proposée dans [110, 109] permettant de filtrer un dendrogramme. Les auteurs utilisent des mesures incluant un paramètre de résolution α permettant de privilégier ou

pénaliser la présence de groupes de grande taille. Ce type de mesure permet d'identifier les groupes les plus significatifs de la hiérarchie (pour un paramètre α fixé) et donc obtenir un arbre de partitionnement moins grand. Les auteurs évaluent ensuite la pertinence des paramètres de résolution en utilisant la mesure $R(\alpha)$ (voir formule 3). La partition finale est obtenue en considérant les partitions correspondantes aux maxima locaux de cette fonction.

5.2 Optimisation d'une partition hiérarchique donnée

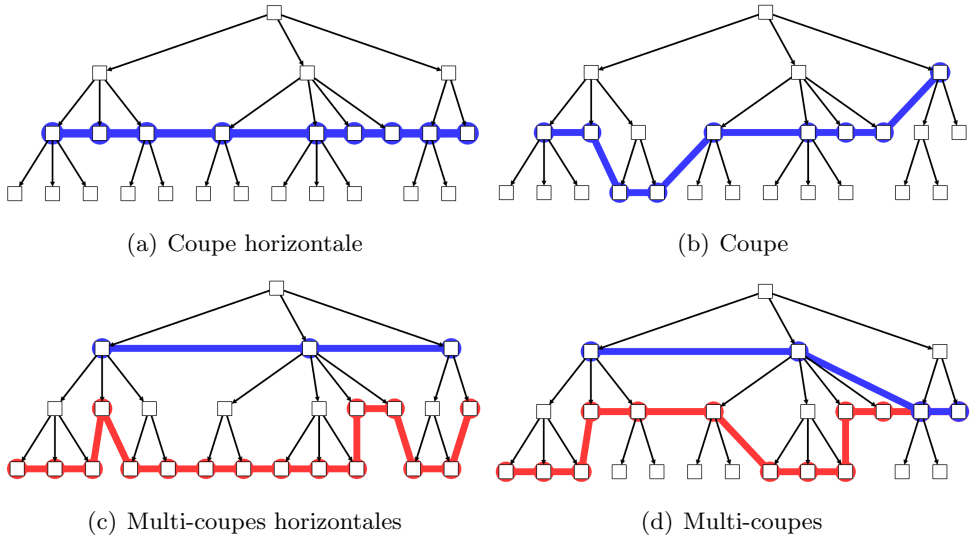
Dans cette section nous présentons un algorithme polynomial pour optimiser une partition hiérarchique d'un graphe. L'évaluation de la qualité est faite en utilisant une mesure multi-niveaux telle que celle définie dans la section 4. Le problème auquel nous tentons de répondre est présenté dans la sous-section 5.2.1. L'algorithme proposé est détaillé dans la sous-section 5.2.2. Enfin, nous discutons la pertinence de cette méthode dans la sous-section 5.2.3.

5.2.1 Motivations

Comme décrit dans la section 5.1, beaucoup de méthodes de partitionnement produisent directement ou indirectement des partitions hiérarchiques. Les hiérarchies ne sont généralement pas exploitables car elle contiennent un nombre de niveaux trop grand. C'est notamment le cas pour les dendrogrammes produits par les méthodes de *clustering* hiérarchique par exemple.

Ces structures sont le plus souvent utilisées pour rechercher un partitionnement moins complexe issu de ces structures. En effet, si on dispose d'un arbre de partition T, une partition plate peut être obtenue un sélectionnant un niveau donné de l'arbre, on fait alors une *coupe horizontale* de l'arbre (voir Figure 5.2(a)). La partition peut être choisie en fonction de différents critères : un nombre de groupes prédéterminé ou une mesure de qualité à maximiser. Dans ce cadre, il existe $h(T)$ candidats : $(N_1(T), N_2(T), \ldots, N_{h(T)}(T))$. Cependant, il est possible de trouver une *coupe non-horizontale* de qualité supérieure (voir figure 5.2(b)). C'est notamment ce type de coupe qui est recherché dans [110] pour maximiser une mesure de qualité fortement additive $\Phi(G, \mathcal{C})$. Leur algorithme nécessite au plus $\mathcal{O}(n)$ évaluations des gains $\phi(G, C)$.

La coupe (horizontale ou non) permet d'extraire une partition plate d'une partition hiérarchique. On peut étendre cette idée pour extraire une sous-partition hiérarchique à partir d'un arbre de partition T. Une première solution est d'extraire un sous-ensemble de *coupes horizontales* (voir Figure 5.2(c)). Le nombre de sous-partitions hiérarchiques



(a) Coupe horizontale (b) Coupe

(c) Multi-coupes horizontales (d) Multi-coupes

FIGURE 5.2: Illustration de différents types de coupes possibles dans un arbre de partition.

de T pouvant être générées est alors de $2^{h(T)}$ (un niveau étant présent ou non dans la sous-partition hiérarchique).

Là encore, si on souhaite trouver des sous-partitions hiérarchiques de bonne qualité, on peut utiliser plusieurs *coupes non-horizontales* (voir Figure 5.2(d)). C'est ce dernier problème que nous traitons ici : pour une partition hiérarchique T donnée, trouver une sous-partition hiérarchique T^* de T qui maximise $\overline{\Phi}(G, T^*)$, une mesure de qualité multi-niveaux telle que celle proposée dans la section 4.

Notons que ce problème revient à chercher l'ensemble de sommets à supprimer de T de façon à obtenir T^* maximisant $\overline{\Phi}(G, T^*)$. En effet, il suffit de retirer les sommets non-couverts par des enveloppes dans la Figure 5.2(d) (hormis la racine de l'arbre) pour obtenir l'arbre de partition ne contenant que les sommets issus des coupes. Cette notion de *suppression* est définie ci-dessous.

Définition 5.1 ***(Suppression)*** *Soit T un arbre de partition des sommets d'un graphe G et un nœud interne $t \in (T \setminus \mathcal{F}(T))$ différent de la racine $r(T)$. La* suppression *de t dans l'arbre T est notée $T' = T \setminus \{t\}$. L'opération consiste à retirer le nœud T de l'arbre et à assigner $p(t)$ comme nouvel ancêtre direct des nœuds de $\sigma_T(t)$.*

Pour toute suppression sur T, l'arbre de partition T' correspond à une sous-partition hiérarchique de T. Remarquons que nous nous concentrons ici sur les suppressions des nœuds internes et non sur l'*élagage* des feuilles de l'arbre. La question de la suppression des feuilles de T est discutée dans la section 5.2.3.

Espace de solutions : Pour chaque nœud interne de T, on peut considérer que celui-ci va être conservé ou non dans l'arbre T^*. Le nombre de sous-partitions hiérarchiques

de T avec le même ensemble de feuilles $\mathcal{F}(T)$ est donc $2^{|T|-|\mathcal{F}(T)|-1}$ où $|T|$ est le nombre de nœuds dans T. Un parcours exhaustif de l'ensemble des possibilités conduit donc à un algorithme exponentiel. C'est pourquoi nous détaillons dans la section suivante une approximation pour ce problème basée sur un algorithme glouton.

5.2.2 Algorithme par suppression de nœuds internes

La mesure $\overline{\Phi}$ (équation 8) permet de comparer différentes partitions hiérarchiques et de déterminer laquelle est la mieux adaptée pour le réseau étudié. Comparer différents arbres de partitions implique que l'on peut évaluer le gain obtenu après une modification opérée sur un arbre donné. Ainsi on peut déterminer si il est opportun de retirer un nœud interne de l'arbre en calculant le gain (ou la perte) de qualité correspondante (voir définition 5.2).

Définition 5.2 ***(Gain par suppression)*** *Soit T un arbre de partition des sommets d'un graphe G et un nœud interne $t \in T - \mathcal{F}(T)$ différent de la racine $r(T)$. Le* gain par suppression *de t, noté $\Delta_t(T)$, correspond à la différence entre la qualité de l'arbre T après et avant la suppression de t.*

$$\Delta_t\overline{\Phi}(T) = \overline{\Phi}(G, T \setminus \{t\}) - \overline{\Phi}(G, T) \tag{1}$$

Partant d'un arbre de partition T initial, notre procédure d'optimisation de T consiste à supprimer le nœud interne t (si il existe) ayant le gain $\Delta_t(T)$ positif maximum (voir algorithme 3). Cette procédure est répétée itérativement.

Algorithme 3: Optimisation gloutonne de T par suppression de nœuds internes

Entrées : $G = (V, E)$ un graphe, T une partition hiérarchique de G
Sorties : T' une sous-partition hiérarchique de G
$T' = T$;
$t_{max} = \arg\max_{t \in T'} \Delta_t\overline{\Phi}(G, T')$;
tant que $\Delta_{t_{max}}\overline{\Phi}(T') > 0$ **faire**
  $T' = T' \setminus \{t_{max}\}$;
  $t_{max} = \arg\max_{t \in T'} \Delta_t\overline{\Phi}(G, T')$;
fin
retourner T';

La suppression d'un nœud t de l'arbre T conduit à plusieurs changements dans le calcul de la mesure multi-niveaux $\overline{\Phi}(G, T)$.
Premièrement, le poids des descendants directs de t devient plus important puisqu'ils interviennent désormais à un niveau hiérarchique plus bas.
Deuxièmement, ces descendants $\sigma_T(t)$ ne forment plus une partition simple du graphe induit $G[V_t]$ mais un sous-ensemble de groupes dans la partition de $G[V_{p(t)}]$. Cette dernière

observation implique que les gains ϕ pour chaque nœud $t' \in \sigma_T(t) \cup \sigma_T(p(t))$ vont être modifiés. Toutefois, si la mesure Φ est fortement additive, alors seuls les gains des nœuds appartenant à $\sigma_T(t)$ doivent être recalculés.
Prenons l'exemple de la figure 5.3, la suppression du nœud t_1 fait que les nœuds t_3 et t_4 deviennent descendants directs de la racine de T. Dans ce cadre, le nombre d'arêtes sortant des sous-graphes G_{t_3} et G_{t_4} augmente puisqu'il faut y inclure les arêtes allant vers le groupe correspondant au nœud t_2. On doit donc, dans tous les cas, calculer les gains $\phi(G[V], V_{t_3})$ et $\phi(G[V], V_{t_4})$. Si ϕ tient compte de la façon dont tous les sommets sont partitionnés (*i.e.* si Φ n'est pas fortement additive), alors on doit calculer les gains $\phi(G[V], (V_{t_2}, V_{t_3}, V_{t_4}), V_{t'})$ pour $t' \in (t_2, t_3, t_4)$.

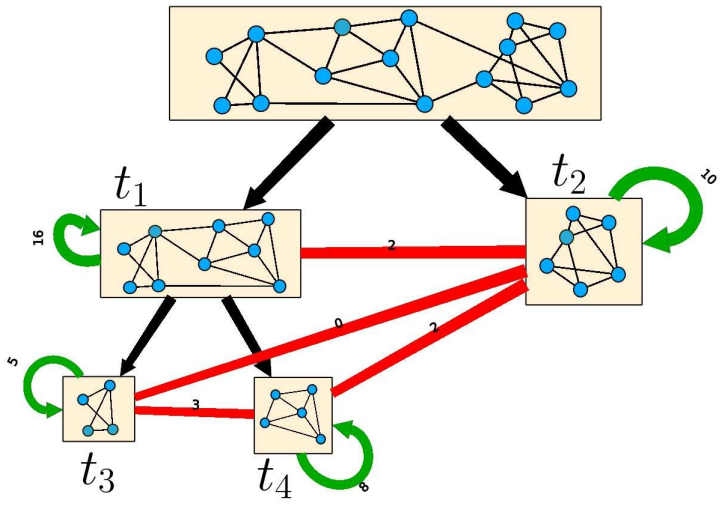


FIGURE 5.3: Illustration des informations conservées en mémoire dans le cadre de l'algorithme 3. Seules les boucles vertes et les arêtes rouges horizontales interviennent dans le calcul de $\overline{\Phi}(G, T)$. Si le nœud t_1 est supprimé, les arêtes (t_2, t_3) et (t_2, t_4) deviennent horizontales.

Complexité : L'algorithme 3 est un algorithme de recherche glouton. La complexité de notre procédure est $\mathcal{O}(h(T)^2|E| + |T|^3)$ en considérant que le calcul des gains ϕ est réalisé en temps constant et dépend du nombre d'arêtes internes et/ou externes à chaque groupe.. Comme indiqué dans la section 4.2, si ces informations sont pré-calculées et gardées en mémoire, le calcul de la fonction se fait en $\mathcal{O}(1)$. Le calcul du nombres d'arêtes internes/externes pour chaque nœud de T est réalisé en $\mathcal{O}(h(T)|E|)$ où $h(T)$ est la hauteur de l'arbre de partition T.

Toutefois, le nombre d'arêtes externes à un groupe peut changer lorsque l'on modifie l'arbre T. Pour garantir un calcul en temps constant des gains, il est possible de conserver en mémoire le nombre de connections de t avec tous les autres sommets de T qui ne sont pas ses descendants ou ses ancêtres. Cette opération est réalisée en $\mathcal{O}(h(T)^2|E|)$ pour un

coût en mémoire de $O(|T|^2)$. Dans ce cadre, le calcul de $\Phi(G,T)$ est de $O(|T|)$ puisqu'il s'agit d'un simple parcours de l'arbre T. Il en est de même pour le calcul de $\Delta_t(T)$ pour tout $t \in T$.

5.2.3 Discussion

L'idée de la procédure décrite par l'algorithme 3 est de se rapprocher petit à petit d'une solution optimale (au moins localement). Toutefois, pour déterminer si cette procédure permet de converger efficacement vers une solution, il faudrait disposer d'une distance entre différentes partitions hiérarchiques. Cette distance permettrait de déterminer si une recherche basée essentiellement sur la suppression de nœuds internes est appropriée.

L'algorithme proposé ici ne teste pas si il est opportun de supprimer ou non les feuilles de l'arbre. Rappelons que les feuilles ne correspondent pas forcément aux éléments de V isolé. Si un sommet t n'a que des feuilles dans ses descendants directs $\sigma_T(t)$ alors la contribution des $\sigma_T(t)$ à $\overline{\Phi}(G,T)$ est toujours positive du moment que la qualité de la partition $(V_{t'})_{t' \in \sigma_T(t)}$ a une qualité positive. Ce qui est normal : tant que l'on gagne à partitionner le graphe plus en profondeur (même si on gagne peu) pourquoi s'arrêter ?
Ce problème peut être résolu en comparant la qualité liée à l'absence de partition de G_t et la partition $(V_{t'})_{t' \in \sigma_T(t)}$. Pour certaines mesures, telle que la modularité Q, l'absence de partition a mécaniquement une qualité nulle et un partitionnement, même de mauvaise qualité, sera souvent préféré. Ce n'est pas le cas pour les mesures MQ et ΔL (voir section 3).

5.3 Applications à l'Algorithme *Louvain*

La procédure présentée dans la section précédente permet d'optimiser n'importe quelle partition hiérarchique d'un graphe. Nous appliquons ici notre méthode à la hiérarchie produite indirectement par l'algorithme *Louvain* [23] présenté dans la section 5.1.

Nous justifions ce choix d'application dans la sous-section 5.3.1. Nous verrons en particulier que la partition hiérarchique peut comporter des "biais de construction" que notre méthode permet de corriger. Des expérimentations sur un *benchmark* aléatoire connu sont données dans la sous-section 5.3.2. Enfin, nous discutons l'utilisation de cette version "optimisée" de l'algorithme *Louvain* dans la sous-section 5.3.3.

5.3.1 Motivations

L'algorithme de [23] produit donc une hiérarchie T en appliquant un algorithme de partitionnement plat à un graphe quotient construit à partir de la partition précédente. Chaque niveau de la hiérarchie correspond à un maximum local de la modularité. Les auteurs estiment que le niveau le plus significatif est le dernier identifié $N_1(T)$ puisqu'il correspond au maximum de la modularité. Ils soulignent cependant l'intérêt que peuvent avoir les autres niveaux ainsi que la hiérarchie complète.
En effet, la mesure de modularité Q souffre d'une *limite de résolution* (voir section 3.1) et tend à favoriser la présence de grands groupes même si ils sont faiblement connectés. L'utilisation de la partition T pourrait être une solution puisqu'elle va permettre de garder trace de partitions de bonne qualité (en tant que maximum local) contenant des groupes plus petits.

La question est de savoir si cette hiérarchie est pertinente pour étudier la structure du réseau étudié. À chaque itération de l'algorithme, une partition des sommets est trouvée en déplaçant des éléments entre différents groupes. Ces changements étant réalisés sommet après sommet pris dans un ordre aléatoire, il est probable que l'algorithme n'arrive pas directement à fusionner deux groupes de grande taille durant cette phase. C'est là que la phase d'agrégation est intéressante.
Il donc possible que certains niveaux de la hiérarchie produite ne soient que des étapes intermédiaires vers la fusion de plusieurs groupes. Ces groupes en devenir ne sont donc pas pertinents et on fait l'hypothèse que leur surpression va améliorer la qualité globale de la hiérarchie.

On va donc chercher à filtrer la hiérarchie produite lors du déroulement de l'algorithme *Louvain* en utilisant la procédure décrite dans la section 5.2 et en utilisant comme gain pour le groupe représenté par le nœud interne $t \in T$:

$$\phi(G_{p(t)}, V_t) = \frac{e_t}{|e_{p(t)}|} - \left(\frac{d_t}{2|e_{p(t)}|}\right)^2 \tag{2}$$

Notons que si t est un descendant direct de la racine, on a $|e_{p(t)}| = |E(G)|$ et on retrouve bien la formule classique de la modularité.

5.3.2 Évaluation sur le *benchmark* LFR

Pour valider notre méthode, nous utilisons le *benchmark* LFR [88] étendu au cas multi-niveaux [89] (voir une description complète de ce modèle dans la section 2.6.5). Plusieurs travaux sur le partitionnement hiérarchique de graphes utilisent cette méthode pour évaluer la qualité des algorithmes (voir par exemple [124]).

En effet, ce *benchmark* permet la génération de graphes possédant une distribution des degrés en loi de puissance et une structure de communautés à deux niveaux. Ces deux niveaux correspondent à des *macro-communautés* découpées en *micro-communautés*. Le nombre d'arêtes et de communautés étant fixés, le modèle consiste à faire varier la cohésion de ces communautés à chacun des deux niveaux de façon à rendre la hiérarchie plus ou moins identifiable. On utilise ainsi deux paramètres μ_1 et μ_2 qui correspondent à la proportion d'arêtes entre macro-communautés (respectivement micro-communautés issues de la même macro-communauté).

La performance des algorithmes à bien détecter les deux niveaux de partition peut être évaluée en utilisant la variation d'information (voir définition 2.9). Cette mesure est une distance entre deux partitions d'un même ensemble aboutissant à un score compris entre 0 (les deux partitions sont complètements différentes) et 1 (les partitions sont identiques). Nous fixons le nombre de sommets à 10000 avec une distribution de degré d'exposant 2 avec une moyenne de 20 et un maximum de 100. La taille des macro-communautés est comprise entre 400 et 4000 sommets et celle des micro-communautés entre 10 et 100. Ces paramètres sont ceux utilisés dans des travaux existants utilisant ce modèle [89, 124].

La figure 5.4 détaille les résultats obtenus. Pour chaque graphique, l'axe des abscisses correspond à la valeur de $\mu_1 + \mu_2$ c'est-à-dire la proportion d'arêtes externes aux micro-communautés. Pour chaque valeur de μ_1, on fait varier μ_2 dans $[\mu_1, 1]$. L'axe des ordonnées représente la distance entre les partitions comparées. On compare ainsi les micro-communautés réelles aux fragmentations $N_2(T)$ et $\mathcal{F}(T)$ (courbes orange et rouges respectivement) et les macro-communautés à la fragmentation $N_1(T)$ (courbes bleues). Les résultats correspondent à une moyenne sur cent simulations.

Nous analysons tout d'abord la partition hiérarchique obtenue avec l'algorithme *Louvain* sans optimisation (colonne de gauche dans la figure 5.4). Les micro-communautés sont assez bien identifiées à partir du moment où la proportion d'arêtes entre ces sous-graphes est faible $\mu_2 < 0.5$. Il en est de même pour les macro-communautés à condition que la mixité entre micro-communautés soit bien supérieure à la mixité entre macro-communautés. On constate en revanche que les arbres de partition produits par l'algorithme contiennent souvent des niveaux supplémentaires. Dans le cas idéal, $N_2(T)$ devrait correspondre à $\mathcal{F}(T)$ et aux micro-communautés or cela n'est pas le cas ici.
Cette dernière observation confirme les risques évoqués en section 5.3.1. Un décalage peut se créer entre les différentes exécutions de l'algorithme, ce qui entraîne l'ajout dans la hiérarchie d'étapes intermédiaires non pertinentes. Un exemple illustrant le problème est donné dans la Figure 5.5.

L'analyse des résultats obtenus en considérant les partitions hiérarchiques optimisées par notre méthode (colonne de droite dans la figure 5.4) montre que ces groupements

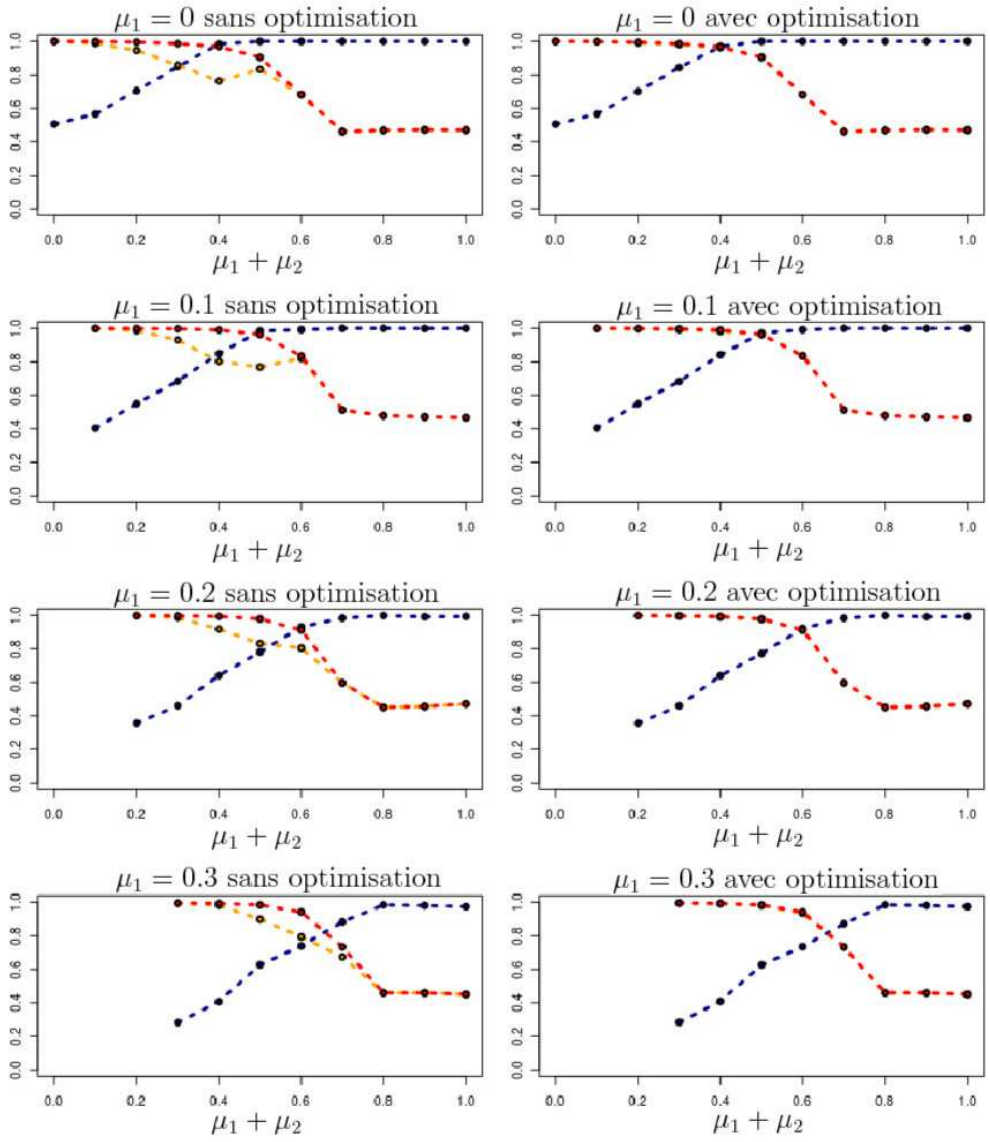


FIGURE 5.4: Résultats sur le *benchmark* LFR hiérarchique pour différentes valeurs de μ_1 et μ_2. La courbe bleue correspond à la distance entre $N_1(T)$ et les vraies macro-communautés (en terme de variation d'information). La courbe rouge entre $\mathcal{F}(T)$ et les vraies micro-communautés. Enfin, la courbe orange entre $N_2(T)$ et les vraies micro-communautés.

intermédiaires sont supprimés et que $N_2(T)$ correspond bien aux micro-communautés. De plus, la similarité entre $N_1(T)$ et les macro-communautés ne change pas, de même pour la similarité entre $\mathcal{F}(T)$ et les micro-communautés. Cela signifie que l'on ne supprime pas à tort certains regroupements.

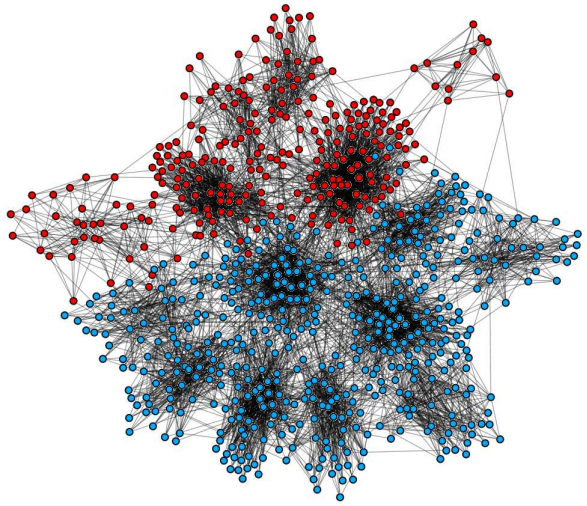


FIGURE 5.5: Un exemple de macro-communauté obtenue avec le modèle LFR hiérarchique et extraite par l'algorithme *Louvain*. Les micro-communautés sont visuellement distinguables en utilisant un algorithme de dessin par modèle de force. Dans cet exemple, l'algorithme *Louvain* passe par un partitionnement en deux groupes (sommets bleus et rouges) avant d'identifier correctement la macro-communauté.

5.3.3 Discussion

Notre méthode permet effectivement de filtrer efficacement la hiérarchie produite par l'algorithme *Louvain*. Des expérimentations réalisées sur des graphes aléatoires dont la structure de communauté multi-niveaux est connue confirme la pertinence de notre méthode. Cette application a pour objectif d'illustrer l'efficacité de notre procédure de suppression gloutonne. Cependant, plusieurs critiques peuvent être faites sur l'utilisation des hiérarchies ainsi produites dans des cas concrets.

Tout d'abord, le problème des niveaux intermédiaires peut être réduit en utilisant une version récursive de l'algorithme *Louvain* en suivant la structure donnée dans l'algorithme 2. Il faudrait alors que la fonction $partition(G)$ renvoie le premier niveau de la hiérarchie créée par l'heuristique. Le résultat sur le *benchmark* LFR montre que le deuxième niveau de la hiérarchie correspond bien plus aux micro-communautés. Cependant, n'ayant aucune information *a priori* sur la profondeur de la hiérarchie, l'algorithme va la plupart du temps découper plus en profondeur les micro-communautés. En effet, comme on l'a déjà indiqué, la modularité va souvent préférer une partition du graphe même de très mauvaise qualité à l'absence de partition.

De plus, nous ne pouvons pas garantir que la partition hiérarchique produite au final soit proche de la partition hiérarchique optimale. En effet, la formulation multi-niveaux de la qualité est utile dans le cas où les premiers niveaux de la hiérarchie ne sont pas

optimaux lorsqu'évalués indépendamment. Or, pour l'algorithme Louvain, le premier niveau est justement celui avec la plus grande modularité (car l'algorithme s'arrête si une phase supplémentaire d'agrégation donne une modularité inférieure). Une solution pour explorer un plus grand nombre de possibilités serait de continuer l'agrégation, par exemple en effectuant des fusions menant à la plus petite perte de modularité.

5.4 Partitionnement hiérarchique d'un réseau de *commuters* par filtrage d'arêtes

Nous présentons ici une approche permettant d'extraire une partition hiérarchique pertinente basée sur du filtrage d'arête. Elle s'inspire notamment de l'algorithme de Auber *et al.* [14]. Nous illustrons cette approche sur un exemple concret : un graphe modélisant les flux domicile-travail dans une région française. Le travail réalisé suggère d'autres applications possibles de l'algorithme présenté dans la section 5.2, celles-ci sont discutées dans la section 5.4.3.

5.4.1 Motivations

Nous nous intéressons ici à un réseau géographique représentant des flux domicile-travail. On parle également de réseau de navetteurs (*commuters* en anglais). Les données dont nous disposons sont issues du recensement national de l'INSEE (`www.insee.fr`) en 1999. Ces flux représentent des déplacements réguliers (la plupart du temps quotidiens) entre le lieu de résidence et le lieu de travail [127]. Les géographes utilisent ces données pour étudier la structure polycentrique des systèmes urbains en utilisant des méthodes issues de l'analyse de réseaux (voir entre autres [60, 106, 108]).

Ces données sont également utilisées pour proposer un découpage du territoire français. L'INSEE définit une classification des communes en tant que *centres urbains*, *villes unipolaires*, *villes multi-polaires* et *villes rurales*. Ces différentes composantes correspondent à la classification ZAUER (Zonage en aires urbaines et espaces ruraux) définie suite au recensement de 1999. Ce type de classification a un rôle important dans la conduite des politiques territoriales (la construction de routes par exemple) et les analyses démographiques (évaluation de l'exode rural ou de la *rurbanisation*).

Les centres urbains et les villes unipolaires sont regroupés dans des *aires urbaines*, ces groupes sont connectés par les *villes multi-polaires* en *aires métropolitaines*. On peut interpréter cette classification comme une partition à deux niveaux du réseau de flux, la densité de flux étant plus importante au sein des aires urbaines et métropolitaines. Ce

partitionnement hiérarchique ne couvre pas les communes rurales qui peuvent être laissées en singletons.

Notre objectif est ici de proposer des classifications alternatives basées sur des méthodes de partitionnement hiérarchique de graphes. Dans ce contexte, De Montis *et al.* [41] utilisent l'algorithme *Louvain* [23] pour analyser un réseau de *commuters* de l'île de Sardaigne. Lancichinetti *et al.* [89] utilisent l'algorithme *OSLOM* pour proposer une fragmentation multi-niveaux d'un réseau de *commuters* anglais.

Nous illustrons notre propos sur une seule région française : les Pays-de-la-Loire. La partition hiérarchique issue de la classification ZAUER est disponible dans la Figure 5.7(a). La position des sommets correspond aux coordonnées géographiques des communes correspondantes. Notons au passage que la décomposition officielle peut regrouper des villes appartenant à différentes régions.
Le graphe modélisant ce réseau est non-orienté et sans boucle (on ne considère pas les personnes vivant et travaillant dans la même ville). Dans ce cadre, une arête est pondérée par le nombre de personnes vivant dans une ville A et travaillant dans une ville B (ou inversement). Nous considérons en effet que pour extraire des sous-groupes denses, l'orientation des arêtes n'est pas utile. Ce graphe contient ainsi environ 1500 sommets (correspondant aux communes) et environ 24000 arêtes. Le poids total des arêtes est de 162000 personnes, ce qui représente 12% de la population active de la région à cette époque.

5.4.2 Détermination des coupes horizontales pertinentes

Pour partitionner hiérarchiquement le réseau décrit ci-dessus, nous utilisons une méthode proche de [14] basée sur le filtrage d'arêtes, un type d'algorithme divisif détaillé dans la section 5.1.2. Nous définissons d'abord une métrique δ permettant d'évaluer si une arête (u, v) appartient à un groupe dense (en terme de flux de travailleurs). Nous formons ensuite une partition hiérarchique en filtrant les arêtes selon différents niveaux de seuils. Le choix de ces seuils s'effectue en utilisant la mesure de qualité multi-niveaux $\overline{MQ}(G, T)$.

Nous utilisons comme métrique la proportion de travailleurs issus des villes u ou v se déplaçant dans un voisinage proche de u et v. Soit $t(u, v)$ le nombre de travailleurs voyageant entre u et v, la métrique $\delta(u, v)$ est donnée dans la formule 3.

$$\delta(u, v) = \frac{I(u, v)}{\sum_{w \in N_u \cup N_v} (t(u, w) + t(v, w)) - I(u, v)} \tag{3}$$

où N_u correspond au voisinage direct du sommet u. On a

$$I(u, v) = t(u, v) + \sum_{w \in N_u \cap N_v} (t(u, w) + t(v, w))$$

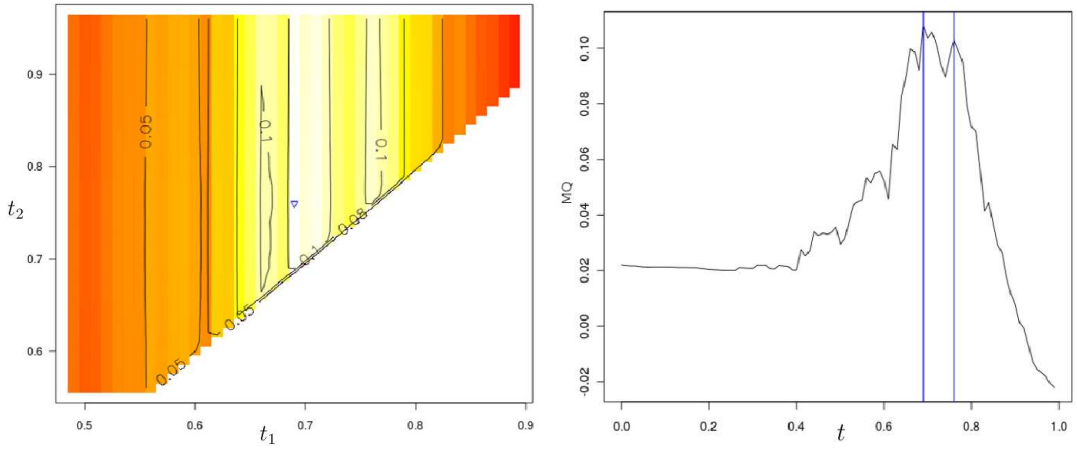
Cette mesure est proche de l'indice de Jaccard [70] entre le voisinage du sommet u et le voisinage du sommet v. Toutefois, nous prenons ici le poids des arêtes en compte. La mesure $\delta(u, v)$ est comprise entre 0 et 1. Une valeur proche de 1 indique que la relation entre les deux communes apparaît au sein d'un groupe dense. Une valeur proche de 0 indique que l'arête forme un pont entre deux communautés.

Rappelons qu'une manière simple pour obtenir un partitionnement plat des sommets du graphe est de choisir une valeur seuil t et de supprimer l'ensemble des arêtes vérifiant $\delta(u, v) \leq t$. La partition est alors donnée par les composantes connexes du sous-graphe obtenu. C'est notamment la méthode utilisée dans [14]. Cette opération correspond à choisir une coupe horizontale dans le dendrogramme obtenu en supprimant les arêtes triées par ordre croissant de δ.
Choisir différentes valeurs de seuils permet d'obtenir un partitionnement hiérarchique du graphe. Ceci correspond au fait de choisir un ensemble de coupes horizontales dans le dendrogramme. Le nombre maximum de coupes est $|V|$.

Dans notre cas, nous explorons les partitions à deux niveaux obtenues en choisissant deux seuils $\{(t_1, t_2) \in [0, 1] \times [0, 1] \,|\, t_1 < t_2\}$. Ce choix vient du fait que l'on cherche un partitionnement de même nature que celui inféré en utilisant la classification ZAUER.
Pour déterminer la paire de seuils, nous utilisons la mesure multi-niveaux $\overline{MQ}(G, T)$ (voir équation 8). La connectivité interne des groupes est comparée au graphe complet et la connectivité externe au graphe biparti complet. De plus, on utilise comme pondération le nombre de communes contenues dans un groupe. Notons ici que la fonction de qualité ne va pas tenir compte du poids des arêtes, cependant cette dimension est déjà capturée par la métrique δ.

La Figure 5.6(a) montre les différentes valeurs obtenues pour un ensemble de paires de seuil (t_1, t_2). Les couleurs les plus claires correspondent aux zones de plus grande qualité. Nous sélectionnons au final la paire donnant la plus grande qualité, ce qui correspond donc à la meilleure paire de coupes horizontales du dendrogramme. La partition hiérarchique correspondante est donnée en Figure 5.7(b). La qualité du premier niveau est de 0.107, les groupes sont redécoupés en utilisant le second seuil et on aboutit à une qualité globale de 0.11. Cette augmentation est faible en raison de la faible densité du graphe. Notons que la matrice illustrée dans la Figure 5.6(a) contient plusieurs zones où la qualité est proche et qui sont des candidates potentielles pour des partitions hiérarchiques différentes. Un avantage de cette méthode de parcours est que l'on peut considérer des situations où la première coupe t_1 n'est pas forcément un maximum global (en terme de qualité).

La partition obtenue est différente de la "partition officielle" de l'INSEE. Ce constat se fait simplement en comparant les représentations des deux partitions (Figure 5.7(b) et 5.7(a)). On peut toutefois remarquer des similarités. En effet, les groupes de plus haut



(a) Représentation matricielle des intégrales de MQ multi-niveaux.

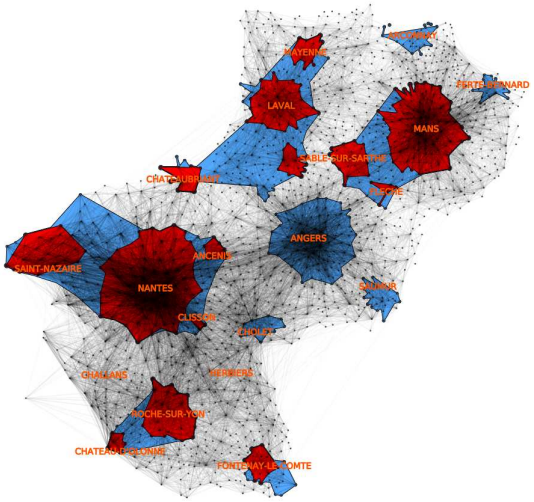
(b) Évolution de la mesure MQ à un niveau en fonction d'un unique seuil t.

FIGURE 5.6: Évolution de la qualité MQ pour des coupes horizontales du réseau en fonction de valeurs de seuils. À gauche : La qualité des partitions hiérarchiques obtenues utilisant deux seuils $t_1 < t_2$ est projetée sur une échelle de couleur ; du rouge (qualité faible) au blanc (qualité forte). Le meilleur couple de seuils pour une partition à deux niveaux est représenté par un triangle bleu. À droite : Les lignes bleues verticales correspondent à la meilleur paire de seuils trouvée.

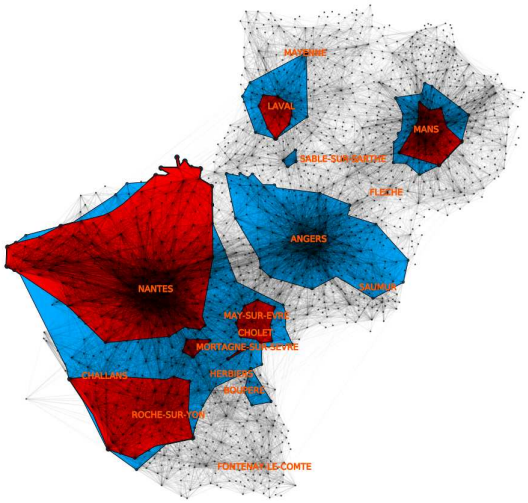
niveau ne sont pas répartis de façon uniforme sur toute la région et organisés autour des plus grandes villes. De plus, les groupes de second niveau correspondent surtout aux banlieues des plus grandes villes.

Nous comparons également les résultats obtenus en utilisant l'algorithme de *Louvain* optimisé selon la méthode détaillée dans la section 5.3 (voir Figure 5.7(c)). L'optimisation supprime plusieurs grands regroupements au second niveau pour se rapprocher d'une hiérarchie de profondeur 2. Les résultats de l'algorithme *Infomap* sont également disponibles (voir Figure 5.7(d)).
On constate que ces deux partitions contiennent beaucoup moins de sommets isolés au premier niveau. En particulier, le découpage au premier niveau est dans certains cas proche de la division en départements, ce qui est particulièrement vrai si on étudie les résultats donnés par *Infomap*. Dans les deux cas, les derniers niveaux des hiérarchies couvrent une partie importante des communes.

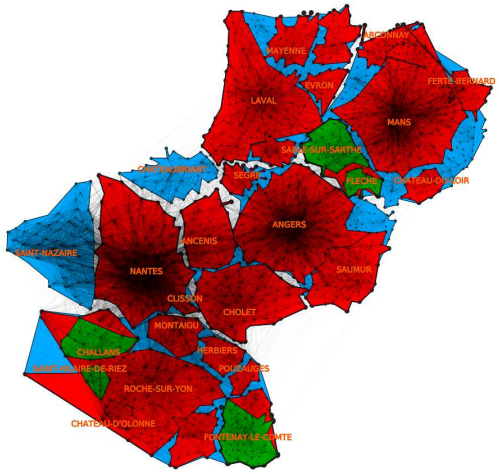
Cet exemple confirme que les méthodes à base de densité telles que MQ sont adaptées à ce type de réseau. En comparant la connectivité des groupes à des situations idéales, beaucoup de communes rurales restent isolées dans le partitionnement. Il est en effet beaucoup plus dur de déterminer les régions denses en terme de *commuters* en utilisant les résultats obtenus avec les algorithmes *Louvain* et *Infomap*.



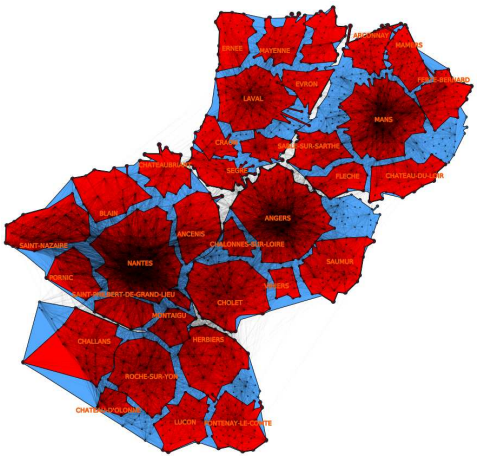
(a) Hiérarchie issue de la classification ZAUER



(b) Coupes horizontales utilisant les valeurs de seuils optimaux (Figure 5.6(a))



(c) Hiérarchie optimisée issue de *Louvain* [23]



(d) Hiérarchie issue de *Infomap* [124]

FIGURE 5.7: Partitionnements hiérarchiques du réseau des flux domicile-travail dans la région des Pays-de-la-Loire (*source : INSEE*). Les groupes sont dessinés en utilisant des enveloppes concaves (Bleues : $N_1(T)$, Rouges : $N_2(T)$, Vertes : $N_3(T)$). Seuls les groupes de villes regroupant plus de cinq mille travailleurs sont affichés. Les étiquettes indiquent le nom de la plus grande ville (en terme de population active) dans les groupes de plus bas niveau.

5.4.3 Perspectives

Le travail effectué ici suggère plusieurs perspectives intéressantes. Premièrement, l'approche présentée ici consiste à choisir plusieurs coupes horizontales au sein d'un dendrogramme. Cependant, on peut supposer extraire des partitions de meilleure qualité en cherchant des coupes non-horizontales. Dans ce cadre, l'algorithme de suppression itératif

des nœuds internes introduit dans la section 5.2 pourrait être utilisé directement sur le dendrogramme. Cette approche serait toutefois coûteuse d'un point de vue complexité, cet arbre contient un très grand nombre de nœuds internes.
Deuxièmement, notons que les seuils optimaux correspondent à des maxima locaux dans l'évolution de la mesure MQ à un niveau (voir Figure 5.7(b)). On retrouve ce phénomène sur d'autres exemples. Il est raisonnable de penser que les maxima locaux pour les partitions plates sont des candidats potentiels pour une sous-partition hiérarchique optimale. Si c'est bien le cas, alors un arbre de partition pourrait être créé en n'utilisant que ce sous-ensemble de coupes. On pourrait alors appliquer l'algorithme glouton d'optimisation de hiérarchie.

5.5 Conclusion et Perspectives

Nous avons proposé dans ce chapitre différentes utilisations des mesures multi-niveaux présentées dans le chapitre 4. Nous avons introduit une procédure de post-traitement d'un arbre de partition permettant d'améliorer sa qualité en supprimant certains nœuds internes. La méthode a été appliquée à l'algorithme de [23] en utilisant une généralisation de la modularité au cadre multi-niveaux. Des tests réalisés sur des graphes aléatoires montrent que la hiérarchie optimisée est bien plus proche de la vraie configuration du réseau.

Pour déterminer si cet algorithme glouton est adapté, il faudrait disposer d'une mesure de distance entre partitions hiérarchiques. Il serait ainsi possible de voir si l'algorithme se rapproche d'une situation optimale suppression après suppression ou si il tombe facilement dans un maximum local.
Cette mesure pourrait également répondre à un problème important : comparer des partitions hiérarchiques extraites automatiquement à une vérité terrain. En effet, l'approche généralement utilisée dans le domaine est la comparaison des niveaux des hiérarchies deux à deux. C'est également cette approche que nous avons utilisée dans la section 5.3.2. L'interprétation des résultats est plus compliquée dans ce contexte.

Nous avons également proposé une méthode permettant de déterminer un sous-ensemble de coupes horizontales dans un dendrogramme obtenu utilisant un algorithme divisif. Cette approche semble adéquate pour l'étude des réseaux de *commuters* lorsque couplée avec la mesure multi-niveaux $\overline{MQ}$ pondérée. Nous suggérons une perspective intéressante qui permettrait d'extraire des coupes non-horizontales pertinentes. La hiérarchie obtenue pourrait être par la suite filtrée en utilisant l'algorithme glouton présenté en section 5.2.

De manière générale, disposer d'une mesure de qualité hiérarchique ouvre la voie à de très nombreuses applications. Nous avons vu en début de ce chapitre que beaucoup

d'algorithmes permettent la génération d'une hiérarchie, que ces derniers soit divisifs ou agglomératifs. Dans la plupart des cas, les méthodes reposent sur une stratégie d'agrégation (ou de division) créant un partitionnement plat du graphe courant. Notons que ces stratégies pourraient être mélangées et/ou combinées au cours des appels récursifs/itératifs afin de combler les défauts de certains algorithmes.

Chapitre 6

Évaluation de la visibilité des communautés dans les diagrammes nœuds-liens

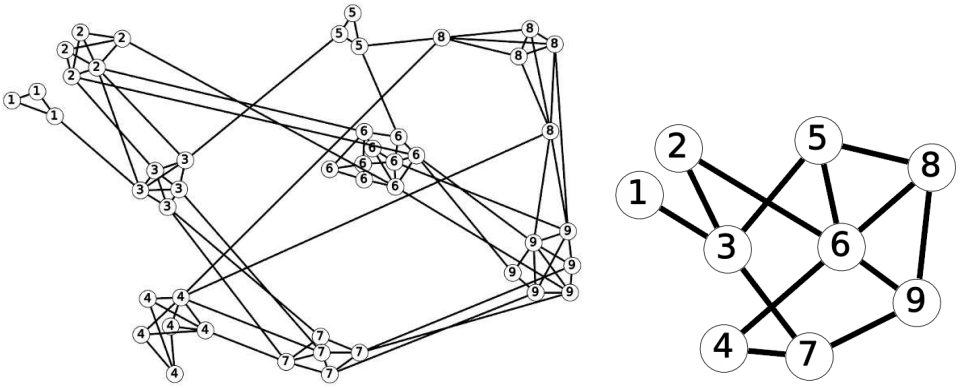
Dans ce chapitre, nous proposons des mesures esthétiques permettant d'évaluer si un dessin de graphe reproduit fidèlement la structure de communauté existant dans un réseau. Ces mesures prennent explicitement en compte une partition des sommets et un dessin du graphe. Elles généralisent des concepts ou des mesures connues utilisées dans le domaine du dessin de graphe. Nous proposons également une évaluation de différents algorithmes de dessins se basant sur ces mesures.

6.1 Problématiques et Motivations

Un problème important de la visualisation de graphes est la définition de ce qui constitue un bon dessin. On peut préciser le terme "bon" : il s'agit de dessiner un graphe de façon à permettre à un utilisateur de réaliser des tâches d'analyse le plus simplement possible. Ces tâches consistent à extraire de l'information à partir de la topologie du graphe (en terme de connectivité, plus court chemin entre deux sommets etc.).
Certaines tâches d'analyse correspondent à l'étude d'une méta-structure cachée au sein de la topologie du graphe. La présence de communautés (sous la forme d'une partition des sommets d'un graphe) en est un exemple. Un utilisateur peut chercher à identifier clairement les différentes communautés, déterminer les relations entre elles ou détecter les sommets interagissant à l'extérieur de leur communauté.

Des mesures quantitatives, appelées *mesures esthétiques* [147], ont été proposées pour déterminer l'aide apportée par le dessin pour la réalisation des tâches d'analyse simples. Historiquement, la première mesure esthétique est le nombre de croisements d'arêtes, cette quantité est au cœur du problème de l'usine de briques de Turán [139]. La minimisation ou la maximisation de certains de ces critères correspondent la plupart du temps à des

problèmes difficiles.
En ce qui concerne la résolution du second type de tâches décrit plus haut, des méthodes de visualisation ont été proposées permettant de souligner explicitement une partition des sommets. On peut citer l'usage de graphes quotients [46] ou d'enveloppes englobantes (voir section 7).



(a) Un graphe avec une structure de communauté

(b) Graphe quotient représentant la partition de l'exemple a)

FIGURE 6.1: Un exemple de graphe avec une structure de communauté et le graphe quotient associé. Dans le graphe quotient, la position d'un sommet correspond au barycentre de la communauté correspondante.

Dans ce chapitre, nous nous plaçons dans l'optique de la visualisation de graphes sous forme de diagrammes nœuds-liens sans autres métaphores visuelles. Nous cherchons à évaluer si la seule position des sommets permet de répondre aux questions qu'un utilisateur se pose. Notons que la plupart des tâches de recherche associées aux communautés sont proches des tâches d'analyse "classiques" mais à un niveau supérieur d'abstraction. Par exemple, "déterminer le nombre de communautés atteignables à partir d'une communauté" est similaire à "déterminer le degré d'un sommet".

Nous allons généraliser des mesures esthétiques au cas où le réseau contient une structure de communauté connue sous la forme d'une partition des sommets. Le terme "généralisation" est employé ici car dans le cas (appelé *cas nul*) où les communautés correspondent à des sommets isolés, les mesures proposées ici sont alors équivalentes à des mesures connues du domaine. Nous étudions le dessin de base du graphe, sans transformation. Une difficulté est que les éléments étudiés correspondent à des groupes de sommets ce qui peut compliquer la "quantification" de certains critères. Par exemple, nous mesurons à quel point deux groupes d'arêtes externes adjacentes à une même communauté sont visuellement séparables. Ce critère correspond, dans le cas nul, à la résolution angulaire entre deux arêtes partageant une extrémité, toutefois, dans notre cas, les deux groupes d'arêtes n'ont pas

forcément la même origine dans le dessin.

Notons que travailler avec le graphe quotient induit par la partition (voir l'exemple de la Figure 6.1) représente une perte d'information importante. Les mesures esthétiques classiques appliquées au graphe quotient ne peuvent pas être utilisées pour juger la qualité du dessin de base. En effet, un chevauchement visuel dans le dessin du graphe quotient n'implique pas forcément un chevauchement dans le dessin du graphe. Par exemple, la méta-arête $(4, 8)$ chevauche le méta-sommet 6 dans la Figure 6.1(b) sans que ce chevauchement se retrouve dans la Figure 6.1(a). De même, les méta-arêtes $(4, 8)$ et $(6, 9)$ ne se croisent pas alors que des croisements existent pourtant entre les arêtes reliant ces communautés.

6.2 État de l'art

La définition et l'étude de mesures esthétiques sont des problèmes très étudiés dans le domaine du dessin de graphe [147, 114]. Purchase [113] propose notamment une évaluation utilisateur permettant de déterminer les mesures les plus pertinentes. "Pertinentes" signifie ici celles qui impactent le plus l'efficacité des utilisateurs à effectuer des tâches de recherche. Ces tâches correspondent à des questions précises posées à un utilisateur : quelle est la longueur du plus court chemin entre deux sommets ? quelle est la taille de la coupe minimum entre deux sommets ; en nombre de sommets ? en nombre d'arêtes ? Plusieurs algorithmes de dessins tiennent directement compte de ces mesures, en tentant d'optimiser une fonction de score. Des méthodes basées sur le recuit simulé [39] ou les algorithmes génétiques [79] furent utilisées dans ce cadre.

McGrath *et al.* [95] proposent de leur côté une évaluation utilisateur où des tâches d'analyse de réseaux sociaux sont demandées. Ils montrent que la perception de la structure en communautés et des liens entre elles est rendue plus difficile lorsque le dessin contient des chevauchements entre les groupes. Plus récemment, Van Ham et Rogowitz [141] montrent que des dessins réalisés sans contraintes par des utilisateurs diffèrent des critères esthétiques classiques lorsque le graphe contient une structure de communauté évidente. Par exemple, Les longueurs d'arêtes sont plus hétérogènes que dans les dessins créés automatiquement par des algorithmes. Les auteurs proposent également de nouveaux critères basés sur la forme des communautés dans un dessin de graphe. En effet, ces groupes sont généralement représentés par des nuages de points denses dans les dessins faits par des utilisateurs. Ils observent également que les points sont fermés par un ensemble d'arêtes.

Les critères utilisés par Van Ham et Rogowitz [141] se focalisent surtout sur la position des communautés (par exemple les distances deux-à-deux) et leurs formes. Les autres critères utilisés tels que le nombre de croisements ou la longueur des arêtes ne tiennent

pas spécifiquement compte de la structure communautaire du graphe.
La section suivante tente de combler ces lacunes en proposant une liste plus complète de critères et des mesures formellement définies pour les évaluer. Nous redéfinissons également certaines mesures proposées dans [141] en les améliorant si besoin.

6.3 Mesures esthétiques pour des dessins de graphes partitionnés

Le Tableau 6.3 fournit la liste des critères proposés. La plupart sont une généralisation de mesures existantes *i.e.* dans le "cas nul" (où le partitionnement correspond à des groupes formés d'un unique sommet) on retrouve les mesures classiques. D'autres mesures tenant compte de la forme des groupes sont des reformulations de mesures proposées dans [141]. Dans cette section, nous définissons formellement chaque critère.

Famille	**Sans partition**	**Nom (Formule)**	**Domaine**	**Interprétation**
Encombrement	Chevauchement sommets	Chevauchement communautés CO (1)	$[0,1]$	-
	Croisement sommets-arêtes	Croisement communautés-arêtes ECC (2)	$[0,1]$	-
	Croisement arêtes	Croisement arêtes internes Cr_{in} (3)	$[0,1]$	-
		Croisement arêtes externes Cr_{out} (4)	$[0,1]$	-
Longueur	Homogénéité longueurs	Différence moyenne T (5)	$\mathbb{R}$	+
		Homogénéité interne $V(E_{in})$ (6)	$\mathbb{R}^+$	-
		Homogénéité externe $V(E_{out})$ (7)	$\mathbb{R}^+$	-
Résolution	Résolution angulaire	Résolution arêtes externes EER (9)	$[0,1]$	+
		Séparation arêtes externes EES (11)	$[0,1]$	+
Forme		Dispersion-communautés CD (12)	$[0,1]$	+
		Délimitation-communautés CH (13)	$[0,1]$	+

TABLE 6.1: Mesures esthétiques proposés pour un graphe muni d'une structure de communauté (troisième colonne). Les critères usuels correspondant se trouvent dans la seconde colonne. Le champ "Interprétation" indique si il est préférable que la mesure soit faible (-) ou forte (+).

Pour un graphe $G = (V, E)$, on considère que les sommets sont plongés dans le plan $[0, 1]^2$ (sans perte de généralité), on désigne par $p(u)$ les coordonnées du sommet u. Ces dernières sont dessinées par des carrés de côté $\frac{1}{|V|}$. Ainsi, le graphe peut être dessiné sans chevauchement de sommets si on place tous les sommets sur une ligne. Les arêtes du graphes sont dessinées par des segments entre les points $p(u)$ et $p(v)$.

On suppose qu'une structure de communauté existe, celle-ci correspond à une partition des sommets $\mathcal{C} = (C_1, \ldots, C_k)$ connue. On désigne par $C(u)$ la communauté contenant le sommet u. Le "cas nul" correspond au partitionnement en $|V|$ singletons : $\mathcal{C}_{nul} = (v_1, \ldots, v_{|V|})$. L'ensemble des arêtes internes est noté E_{in} et l'ensemble des arêtes externes E_{out}, notons que dans le cas nul on a $E_{out} = E$. On utilise également $H^{vex}(C_i) = [p_1, \ldots, p_h]$ (respectivement $H^{cave}(C_i)$) qui est l'ensemble de coordonnées formant l'enveloppe convexe (resp. concave) minimum des points de C_i et $H^{vex}(\mathcal{C})$ (resp. $H^{cave}(\mathcal{C})$) l'ensemble de ces polygones. Les enveloppes concaves sont ici calculées en utilisant une méthode de *clipping* de polygones [142] : les sommets et les arêtes d'une communauté sont associés à des carrés et des rectangles que l'on fusionne itérativement en supprimant les "trous" (pour plus de détails voir Section 7). Les sommets étant représentés par des carrés de taille $\frac{1}{|V|}$, on définit également par $H(u)$ les quatre coordonnées du carré correspondant.

Nous allons maintenant détailler chaque mesure proposée en les séparant selon leur famille, dans l'ordre donné par la seconde colonne du tableau 6.3.

6.3.1 Mesures d'encombrement

Pour évaluer l'encombrement d'un dessin de graphe, on compte généralement le nombre d'éléments se chevauchant à tort dans le dessin. Différents exemples sont donnés en Figure 6.2.

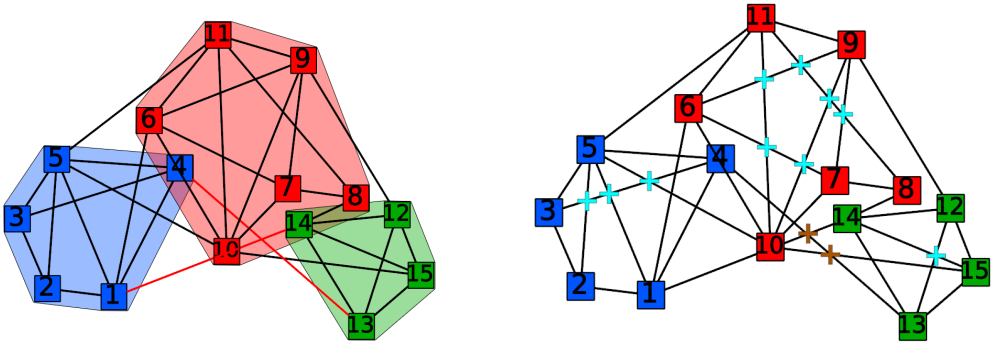
Définition 6.1 ***(Chevauchement communautés)*** *On définit le Chevauchement communautés d'un dessin comme la part moyenne de polygones convexes que chaque sommet chevauche. On note cette mesure CO.*

$$CO = \frac{1}{2|V|} \sum_{u \in V} \frac{1}{k-1} \sum_{C \neq C(u)} \delta_C(u) \tag{1}$$

où $\delta_C(u)$ est égal à 1 si le carré $H(u)$ chevauche le polygone $H^{vex}(C)$ et 0 sinon.

Cette mesure est appelée la "séparation des communautés" dans [141]. Notons que dans le cas nul, la mesure CO_{nul} permet d'évaluer le chevauchement entre sommets. Dans l'exemple donné en Figure 6.2(a), on a $CO = 0.05$ (trois sommets chevauchent à tort une autre communauté) en revanche on n'a pas de chevauchement entre sommets $CO_{nul} = 0$.

On veut maintenant évaluer si une arête externe peut être affectée sans erreur aux communautés qu'elle relie. On suppose que cette tâche est plus simple dans le cas où



(a) Les sommets 4, 8 et 14 chevauchent la zone d'une autre communauté. Les arêtes rouges traversent une communauté sans que leurs extrémités y appartiennent.

(b) Les croix bleus-clairs (respectivement marron) indiquent les croisements internes (resp. externes).

FIGURE 6.2: Illustration des mesures d'encombrement avec un graphe formé de 3 communautés (sommets bleus, rouges et verts).

l'arête ne croise aucune autre communauté. Ce critère correspond au fait qu'un segment dans un dessin ne croise pas d'autres sommets que les deux extrémités de l'arête.

Définition 6.2 ***(Chevauchement communautés-arêtes)*** *On définit le Chevauchement communautés-arêtes d'un dessin comme la part moyenne de polygones convexes que chaque arête externe croise. On note cette mesure ECC.*

$$ECC = \frac{1}{|E_{out}|} \sum_{(u,v)\in E_{out}} \frac{1}{k-2} \sum_{C\neq C(u),C(v)} \delta_C(u,v) \tag{2}$$

où $\delta_C(u,v)$ est égal à 1 si le segment $(p(u),p(v))$ croise le polygone $H^{vex}(C)$ et 0 sinon.

Dans le cas nul, chaque communauté n'est formée que d'un singleton. La mesure ECC_{nul} permet ainsi d'évaluer les croisements sommets-arêtes. Dans l'exemple donné en Figure 6.2(a), on a $ECC = 0.25$ (2 arêtes externes sur les 8 chevauchent à tort une troisième communauté) et $ECC_{nul} = 0,06$ (2 arêtes sur 33 chevauchent un sommet).

Le dernier type d'encombrement que nous examinons est le croisement entre les segments représentant les arêtes. Nous différencions pour cela les arêtes internes des arêtes externes. Pour un sous-ensemble $S \subseteq E$, on note $Cr(S)$ le nombre de croisements de segments dans S. Une première borne supérieure pour cette quantité est $\binom{|S|}{2}$. Toutefois une limite plus réaliste s'obtient en observant que deux arêtes partageant une extrémité commune ne peuvent introduire plus de bruit dans le dessin [114]. Il faut donc retirer, pour chaque sommet u de G, le nombre de paires formées par les arêtes adjacentes à u. On aboutit à l'inégalité suivante

$$Cr(S) \leq \binom{|S|}{2} - \sum_{u\in V(S)} \binom{d_{G(S)}(u)}{2} \tag{2}$$

où $G(S)$ est le sous-graphe de G formé par S et $V(S)$ et l'ensemble des sommets de $G(S)$.

Définition 6.3 ***(Croisement interne)*** *On définit la mesure de croisements internes d'un dessin, notée Cr_{in}, comme la moyenne de la part de croisements dans chaque communauté.*

$$Cr_{in} = \frac{1}{k}\sum_{i=1}^{k} \frac{Cr(E_i)}{\binom{|E_i|}{2} - \frac{1}{2}\sum_{u\in C_i} d_{in}(u)(d_{in}(u)-1)} \tag{3}$$

Dans l'exemple donné en Figure 6.2(b), on a $Cr_{in} = \frac{1}{3}\left(\frac{3}{10} + \frac{6}{24} + \frac{1}{3}\right) = 0.29$. Puisque nous travaillons ici avec la proportion de croisements observés nous ne tenons pas compte de la taille des communautés dans la mesure.

Définition 6.4 ***(Croisement externe)*** *On définit la mesure de croisements externes d'un dessin, notée Cr_{out}, comme la part de croisements dans les arêtes externes aux communautés.*

$$Cr_{out} = \frac{Cr(E_{out})}{\binom{|E_{out}|}{2} - \frac{1}{2}\sum_{u\in V} d_{out}(u)(d_{out}(u)-1)} \tag{4}$$

Notons que, dans le cas nul, on a $E_{out} = E$, ainsi la mesure Cr_0 correspond à la part de croisements dans le dessin. Dans l'exemple, on a $Cr_{out} = \frac{3}{22} = 0.13$.

6.3.2 Mesures sur les longueurs

L'évaluation d'un dessin de graphe repose en partie sur la distribution des longueurs des arêtes. On veut en général que celles-ci soient homogènes (de variance faible). Des métriques classiques basées sur les longueurs des arêtes sont la moyenne, la variance (ou écart-type) et l'étendue. D'autres statistiques plus "robustes" à la présence de valeurs extrêmes sont, par exemple, la médiane ou l'écart inter-quartile.

On peut supposer que les longueurs sont globalement très dispersées mais faiblement dispersées localement *i.e.* la variance provient en grande partie de la variance entre les valeurs moyennes de différentes classes. On voudrait, dans notre cas, tester cette situation en supposant l'existence de deux classes E_{in} et E_{out}. Pour déterminer si ce modèle est présent dans un dessin, on va d'abord tester l'égalité des longueurs moyennes dans les deux groupes.

Définition 6.5 ***(Différence longueurs moyennes)*** *La différence entre les longueurs moyennes des arêtes internes et externes est donnée par la statistique T.*

$$T = \frac{L(E_{in}) - L(E_{out})}{\sqrt{\frac{Var(E_{in})}{|E_{in}|} + \frac{Var(E_{out})}{|E_{out}|}}} \tag{5}$$

où $L(E')$ (respectivement $Var(E')$) désigne la longueur moyenne (resp. la variance) des arêtes de $E' \subseteq E$.

La statistique T peut être approchée par une loi de Student dont le degré de liberté est donné par l'équation de Welch-Satterthwaite [129, 150]. On peut ainsi calculer le risque statistique associé à l'hypothèse nulle : "les longueurs moyennes dans E_{in} et E_{out} sont égales".
Si l'écart entre les longueurs est statistiquement significatif, il peut être intéressant de dissocier les deux ensembles afin d'évaluer l'homogénéité des longueurs. On peut utiliser les écarts-types des longueurs internes et externes définis ci-dessous :

$$\sigma_{in} = \sqrt{Var(E_{in})} \quad (6)$$

$$\sigma_{out} = \sqrt{Var(E_{out})} \quad (7)$$

Notons que dans le cas nul, le test d'égalité des longueurs moyennes n'a pas lieu d'être puisqu'il n'existe qu'une population (les arêtes externes). La mesure 7 permet alors d'évaluer l'homogénéité des longueurs : si σ_{out} est faible alors les arêtes ont globalement une longueur proche. Dans le cas contraire, les arêtes sont de taille très variable. La même interprétation peut être faite sur les deux groupes E_{in} et E_{out}.

6.3.3 Mesures de résolution

Un aspect important dans l'analyse d'un réseau est d'identifier la nature des relations. Les connections entre communautés doivent être facilement distinguables les unes des autres. On formule ici un critère de *résolution* pour évaluer ce critère.
Dans le cas nul, ce dernier repose sur la séparation entre les arêtes adjacentes à un même sommet. Pour évaluer cela, on calcule l'angle minimum entre deux arêtes ayant un sommet en commun. La mesure correspondante est appelée *résolution angulaire* [53]. Elle est maximisée dans les dessins appelés *orthogonaux*.

Dans notre cas, on veut évaluer la séparation entre deux groupes d'arêtes E_{ij} et E_{ig} où toutes possèdent exactement une extrémité dans la communauté C_i (voir Figure 6.3(a)). On note V_{ij} les sommets de C_i ayant au moins un voisin dans C_j. Intuitivement, ces deux groupes d'arêtes seront bien séparés si les arêtes ont, en moyenne, des origines et des directions éloignées. Dans la Figure 6.3(a), le groupe d'arêtes bleues est bien séparé des groupes vert et rouge. En revanche, les arêtes vertes et rouges semblent avoir des origines et des directions proches.

On compare ici deux ensembles et non simplement une paire d'arêtes. On utilise pour cela la notion d'*arête moyenne* définie ci-dessous (un exemple est donné en Figure 6.3(b)).

Définition 6.6 ***(Arête moyenne)*** *Soit un ensemble d'arêtes externes E_{ij} reliant les communautés C_i et C_j. On note $\tilde{b}(C)$ le barycentre de $C \subseteq V$ où chaque sommet $u \in C$ est pondéré par la quantité $d_{G[C]}(u)$. L'*arête moyenne *de E_{ij} est la demi-droite $[\tilde{b}(V_{ij}), \tilde{b}(V_{ji}))$.*

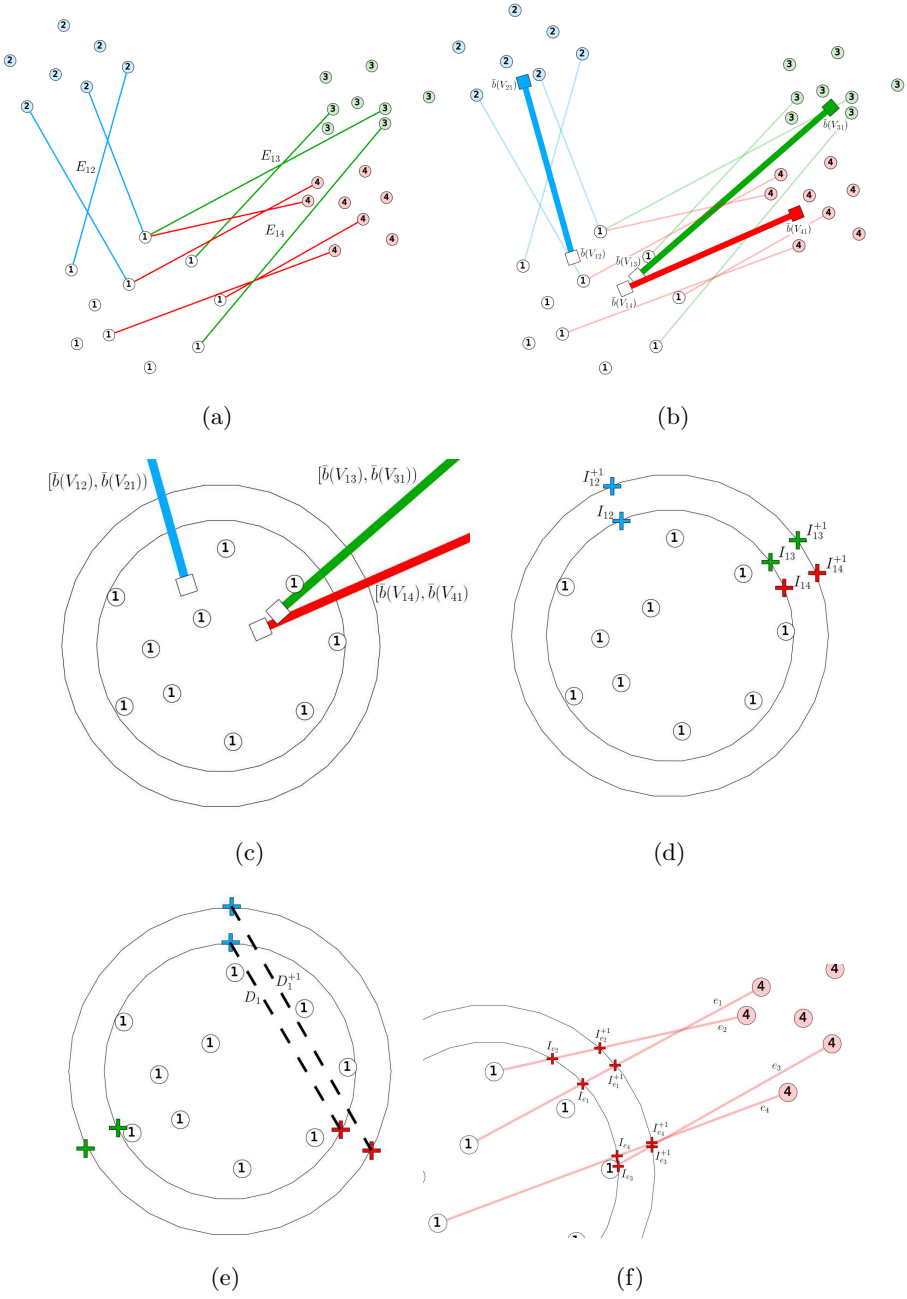


FIGURE 6.3: Illustration des mesures de résolution et de séparation externes dans un dessin. Les sommets sont étiquetés selon leur communauté. a) On étudie les 3 groupes d'arêtes sortant de C_1 (sommets blancs). b) et c) Les 3 arêtes moyennes en tant que segments et demi-droites. d) Intersection des arêtes moyennes avec les deux cercles englobants. e) Positionnement des intersections maximisant la résolution. f) Évaluation de la séparation des arêtes dans E_{14}.

Notons que si $C_i = \{u\}$ alors chaque arête moyenne issue de C_i est une arête adjacente à u et le barycentre $\tilde{b}(C_i) = p(u)$.

Les arêtes moyennes issues d'une communauté C_i n'ont pas toutes la même origine

dans le dessin, on ne peut donc pas directement utiliser une approche basée sur les angles. Pour résoudre ce problème, on va s'intéresser aux projections des arêtes moyennes sur deux cercles englobants les points de C_i. Le premier est le cercle englobant minimum [137] de rayon r calculé en utilisant l'algorithme de Fischer *et al.* [50]. Le second cercle a la même origine et son rayon est $r + 1$ (voir Figure 6.3(c)). Le premier cercle permet d'évaluer la séparation entre deux groupes d'arêtes lorsque celles-ci "sortent" de la communauté. On utilise le deuxième cercle pour évaluer l'écart entre les directions prises par les deux groupes. Ainsi le rayon du second cercle importe peu, du moment qu'il est supérieur à r.

Dans le cas nul, les arêtes (u, v) et (u, w) ayant la même origine, la distance entre les deux intersections de deux arêtes avec n'importe quel cercle centré en $p(u)$ est directement liée à l'angle formé par (u, v) et (u, w). C'est toujours le cas pour une combinaison linéaire des distances entre les intersections avec deux cercles centrés en $p(u)$. On va donc utiliser pour deux groupes E_{ij} et E_{ig} la somme des écarts entre les intersections des arêtes moyennes $[\tilde{b}(V_{ij}), \tilde{b}(V_{ji}))$ et $[\tilde{b}(V_{ig}), \tilde{b}(V_{gi}))$ avec les deux cercles englobants (voir Figure 6.3(d)).

Définition 6.7 ***(Résolution Externe)*** *Soit un triplet de communautés C_i, C_j, C_g, on note I_{ij} (respectivement I_{ij}^{+1}) l'intersection de l'arête moyenne $[\tilde{b}(V_{ij}), \tilde{b}(V_{ji}))$ avec le cercle englobant de rayon minimum r (resp.$r+1$). La* résolution *entre les deux groupes est donnée par*

$$R(E_{ij}, E_{ig}) = \frac{||I_{ij} - I_{ig}||^2 + ||I_{ij}^{+1} - I_{ig}^{+1}||^2}{D_i + D_i^{+1}} \tag{8}$$

où D_i (respectivement D_i^{+1}) est la distance entre les deux points du cercle englobant de rayon minimum r (resp. $r + 1$) si ils étaient placés à équi-distance sur le cercle.

La résolution externe*, notée EER, est donnée par le triplet i, j, g qui minimise la résolution entre E_{ij} et E_{ig}.*

$$ERR = \min_{i,j,g} R(E_{ij}, E_{ig}) \tag{9}$$

Dans l'équation 8, l'écart est maximisé par $D_i + D_i^{+1}$, qui est l'écart dans le cas où les arêtes moyennes sont les plus facilement identifiables (voir Figure 6.3(e)). De la même manière, l'angle entre deux arêtes (u, v) et (u, w) est maximisé par $\frac{360}{d_G(u)}$. Nous considérons ici la résolution minimum pour l'ensemble du dessin. D'autres statistiques pourraient être employées comme le premier quantile ou la borne extrême inférieure. Nous prenons le minimum car c'est celle couramment utilisée pour la résolution angulaire. Dans le cas nul, la mesure EER est directement liée à l'angle minimum entre deux arêtes adjacentes à un même sommet.

La mesure ERR permet d'évaluer la séparation entre les arêtes reliant une communauté aux autres. Il est également important d'étudier les arêtes entre deux communautés pour

pouvoir analyser la façon dont ces deux communautés sont connectées entre elles. Pour cela, nous proposons de mesurer la résolution minimum entre deux arêtes appartenant au même ensemble E_{ij} (voir Figure 6.3(f)). Deux arêtes de E_{ij} peuvent ne pas partager d'extrémité dans C_i, on utilise ici aussi la comparaison entre les intersections avec deux cercles englobants.

Définition 6.8 ***(Séparation Externe)*** *Pour $u \in C_i$ et $v \in C_j$, on note $I_{(u,v)}$ (respectivement $I^{+1}_{(u,v)}$) l'intersection de la demi-droite $[p(u), p(v))$ avec le cercle englobant de C_i de rayon minimum r (resp. $r+1$). La* résolution *entre deux arêtes $e, e' \in E_{ij}$ est donnée par*

$$R(e, e') = \frac{||I_e - I_{e'}||^2 + ||I_e^{+1} - I_{e'}^{+1}||^2}{D_i + D_i^{+1}} \tag{10}$$

La séparation externe*, notée EES, correspond à la résolution minimum pour tout ensemble E_{ij} et toute paire d'arêtes $e, e' \in E_{ij}$.*

$$EES = \min_{e,e'} R(e, e') \tag{11}$$

On considère, dans le cas nul, que la séparation externe EES vaut 1 étant donné que chaque ensemble E_{ij} ne contient qu'une seule arête.

6.3.4 Mesures sur les formes

La forme des sommets est un aspect relativement peu étudié dans le dessin de graphe puisque, en général, celle-ci est fixée à l'avance (des carrés dans notre cas). Les formes couramment employées sont convexes et "compactes" (carrés, cercles *etc.*).On veut ici évaluer à quel point ces deux critères (convexité et faible dispersion) sont respectés.

Pour déterminer la dispersion des communautés, on peut se baser sur les distances entre sommets appartenant à la même communauté. Dans [141], les auteurs utilisent la moyenne de ces distances et la mesure correspondante est appelée *Cluster Extraction*. On propose ici une mesure légèrement différente. En effet, si les nuages de points correspondant aux communautés forment des ensembles compacts et bien séparés alors les barycentres des communautés sont une bonne approximation de la position de leurs membres. De la même façon, si on devait classer les points en se basant uniquement sur la distance qui les sépare, la meilleure classification serait donnée par les communautés. On va donc pouvoir évaluer la dispersion au sein des communautés en calculant le coefficient de détermination (aussi appelé R^2) associé à cette classification.

Définition 6.9 ***(Dispersion)*** *La mesure de dispersion des communautés, notée CD, correspond au rapport entre la dispersion des barycentres de communautés et la dispersion globale des sommets.*

$$CD = \frac{\sum_{i=1}^{k} |C_i| \|b(C_i) - b(V)\|^2}{\sum_{u \in V} \|p(u) - b(V)\|^2} \tag{12}$$

où $b(V')$ correspond au barycentre des sommets de $V' \subseteq V$.

La mesure $CD \in [0, 1]$. Une valeur de 1 indique que tous les sommets d'une même communauté sont à la même position. Une valeur de 0 indique que les barycentres des communautés se confondent avec le barycentre du graphe. Dans le cas nul, on a $CD_{nul} = 1$. L'avantage de cette méthode comparée à la mesure proposée dans [141] est que l'on tient ici compte de la dispersion globale des points dans le plan.

Un autre critère d'esthétisme pour le dessin de communautés consiste à dire qu'une communauté sera plus facilement identifiable si elle est représentée de façon convexe. Dans [141], les auteurs proposent comme mesure le nombre de communautés entourées complètement par un ensemble d'arêtes. Cette condition est satisfaite lorsque l'enveloppe convexe du sous-graphe induit par la communauté est confondue avec son enveloppe concave. Nous allons ici utiliser une évaluation moins stricte en calculant la proportion entre les aires des enveloppes concaves et les aires des enveloppes convexes.

Définition 6.10 ***(Délimitation)*** *La mesure de délimitation des communautés, notée CH, correspond au rapport entre la somme des aires concaves et la somme des aires convexes.*

$$CH = \frac{\sum_{i=1}^{k} A(H^{cave}(C_i))}{\sum_{i=1}^{k} A(H^{vex}(C_i))} \tag{13}$$

où $A(H)$ est l'aire du polygone H.

Dans le meilleur des cas, l'enveloppe concave d'une communauté est confondue avec son enveloppe convexe. On retombe alors sur le cas évalué dans [141]. Si cette situation se retrouve pour chaque communauté, on aura $CH = 1$. Notons cependant qu'une valeur proche de 0 est difficile à obtenir (bien que ce ne soit pas ce que l'on recherche *a priori*). En effet, si les communautés sont fortement inter-connectées alors il y a de grandes chances de calculer des enveloppes concaves qui s'avèrent être convexes (c'est toujours le cas pour une clique). Dans le cas nul, on a $CH_{nul} = 1$ car tous les sommets sont représentés par des carrés.

6.4 Comparaison de différents algorithmes de dessin

Nous allons maintenant utiliser les différentes mesures esthétiques que nous avons définies pour comparer plusieurs algorithmes de dessin de graphes.

6.4.1 Jeu de données utilisé

Nous utilisons des graphes générés aléatoirement et dont la structure de communauté est connue. Le modèle LFR [88] est bien adapté dans ce cadre (pour une présentation complète de ce modèle voir la Section 2.6.5). En effet, celui-ci permet de générer des

graphes ayant une structure proche des réseaux réels : la distribution des degrés et la taille des communautés suivent des lois de puissance (bornées). Pour cette raison, ce modèle est souvent utilisé pour tester la validité d'algorithmes de partitionnement.

Nous travaillons avec des graphes de 150, 300 et 600 sommets. Les résultats étant sensiblement similaires dans tous les cas, nous reportons dans cette section ceux obtenus pour 300 sommets (nous préciserons, si nécessaire, les différences notables). Les paramètres suivants correspondent à ceux choisis dans [88] pour comparer plusieurs algorithmes de partitionnement. Ainsi, la distribution des degrés est une loi de puissance ayant pour exposant 2, le degré moyen est de 12 et le degré maximum de 30. La taille des communautés suit une loi de puissance d'exposant 1. Les sommets ont 15% de leur arêtes adjacentes allant dans une autre communauté. Ce dernier paramètre garantit que la partition qui permet de générer le graphe représente bien une structure de communauté du graphe produit. Nous avons vérifié cela *a posteriori* en calculant les valeurs de plusieurs mesures de qualité. Notons également qu'il n'existe *a priori* pas d'attachements préférentiels entre les communautés. En effet, les arêtes externes adjacentes à un sommet se répartissent uniformément dans les autres communautés. Une communauté n'est donc pas "plus proche", topologiquement parlant, d'une communauté en particulier.

6.4.2 Algorithmes de dessin étudiés

Les algorithmes que nous allons utiliser se basent sur des modèles de forces. Ce choix se justifie par le fait que ce type d'algorithmes fonctionne sur n'importe quel type de graphe. De plus, le dessin produit par ces algorithmes est censé révéler la présence de communautés. Des approches différentes ont cependant été proposées dans ce contexte (voir par exemple l'heuristique proposée par [83]).

D'après [102], le "partitionnement spatial" induit par le dessin obtenu avec ce type de modèle est similaire à un partitionnement maximisant la modularité (voir description dans la Section 3.1). Un partitionnement spatial est, dans ce cadre, une partition des sommets obtenue en retirant les arêtes les plus longues dans le dessin, ces arêtes devraient en effet correspondre aux arêtes externes dans la partition. Les algorithmes utilisés sont détaillés dans le Tableau 6.4.2. Des exemples de sorties des cinq algorithmes sont disponibles dans la Figure 6.4. Nous avons généré en tout 200 graphes pour chaque algorithme étudié.

Pour cette évaluation, nous choisissons de comparer trois algorithmes ne tenant pas directement compte de la structure de communauté du graphe. Nous donnons également des hypothèses *a priori* que nous allons tester en utilisant les mesures esthétiques.

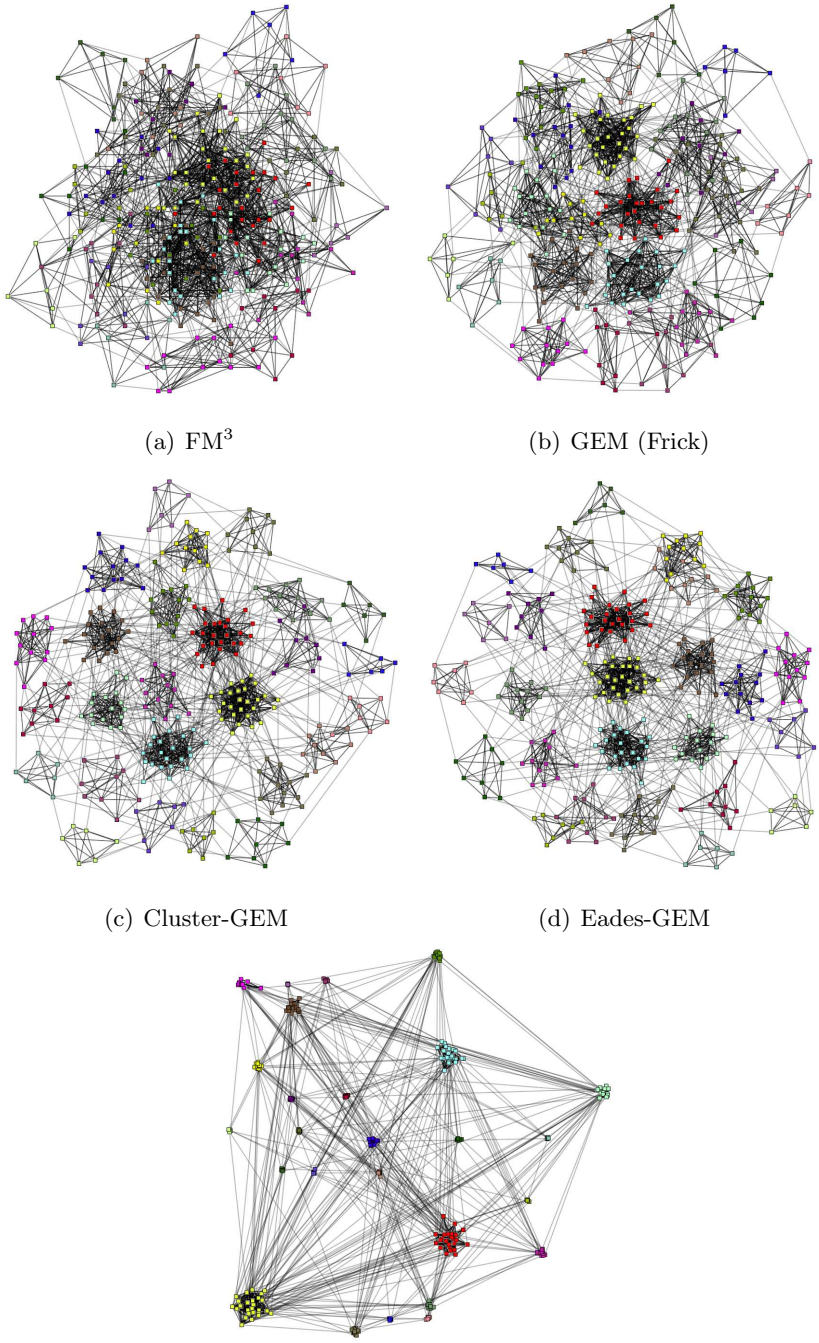
Le premier est l'algorithme **GEM (Frick)** [58] qui utilise un modèle de forces et est

Type	Nom	Ref.	Impl.	Paramètres
Sans partition	GEM (Frick)	[58]	Tulip [15]	Itérations : $\|V\|^2$
	LinLog	[103]	Tulip	Répulsion 0, attraction 1, gravitation 0.05
	FM^3	[69]	OGDF [30]	Défaut
Avec partition	Cluster-GEM	-	Tulip	Attraction interne 1, externe 2
	Eades-GEM	-	Tulip	Attraction interne 1, externe 2, virtuelle 2

TABLE 6.2: Détails sur les algorithmes de dessins utilisés et leurs paramètres (seul les plus importants sont décrits ici).

souvent présent dans les logiciels de visualisation de réseaux. Chaque itération de l'algorithme est réalisée en $\mathcal{O}(|V|^2)$. Le second est l'algorithme **FM**^{3} [69]. C'est un algorithme multi-niveaux également basé sur un modèle de forces. Il a l'avantage d'être bien plus performant que les autres algorithmes utilisés ici. Son temps d'exécution étant de l'ordre de $\mathcal{O}(|V|\log|V| + |E|)$, utiliser cet algorithme pourrait donc être intéressant même si les résultats du point de vue des mesures esthétiques sont légèrement moins bons que ceux de **GEM (Frick)**. L'algorithme **Linlog** est spécialement conçu pour dessiner un graphe contenant une structure de communauté. On peut s'attendre à ce que cet algorithme donne des meilleurs résultats que **GEM (Frick)** et **FM**^{3}.

Les deux autres algorithmes utilisés prennent en entrée une partition des sommets. Le premier est un algorithme qui nous nommerons **Eades-GEM**. Il est inspiré par la méthode proposée dans [47]. Dans un premier temps, un sommet virtuel (un représentant) est ajouté dans chaque communauté en connectant les sommets à ce représentant. Dans un second temps, un algorithme de force (ici **GEM (Frick)**) est appliqué en considérant trois types de forces différentes : la force externe qui s'applique sur les arêtes externes, la force interne qui s'applique sur les arêtes internes et la force virtuelle qui s'applique sur les arêtes connectant les sommets à leur représentant. Le second algorithme considéré est l'algorithme que nous nommerons **Cluster-GEM**. Il est semblable à l'algorithme **Eades-GEM** mais sans ajouter de sommet virtuel au sein de chaque communauté. En comparant **Eades-GEM** et **Cluster-GEM**, on va pouvoir juger l'intérêt d'utiliser des sommets virtuels. Nous allons fournir en entrée de ces deux algorithmes la partition utilisée pour construire les graphes aléatoires LFR, on peut donc s'attendre à ce que ces deux algorithmes fournissent globalement des meilleurs résultats que les trois autres.



(a) FM^3 (b) GEM (Frick)

(c) Cluster-GEM (d) Eades-GEM

(e) LinLog

FIGURE 6.4: Résultats de cinq algorithmes de dessin sur un graphe issu du *benchmark* LFR. Les communautés sont représentées avec un code couleur sur les sommets. La taille des sommets est ici plus grande que dans le protocole utilisé.

6.4.3 Analyse statistique des résultats

Nous allons comparer les algorithmes en utilisant les mesures proposées dans ce chapitre. Nous utiliserons également les mesures dans le cas nul. En effet, nous souhaitons

déterminer si le dessin peut à la fois bien refléter la structure de communauté mais également respecter les critères esthétiques "classiques".

Pour chaque mesure, on calcule la moyenne et l'écart-type sur 200 réalisations. Un test ANOVA [10] permet de déterminer si il existe une différence significative selon l'algorithme utilisé pour dessiner le graphe. Notons que ce test est possible car les observations correspondent à des couples "graphe/dessin" et sont indépendantes puisque on n'utilise pas deux fois le même graphe. L'hypothèse testée dans ce cadre est l'égalité des cinq moyennes. Pour chaque famille de mesures nous allons présenter les résultats sous la forme de diagrammes en bâton. Nous indiquons également sur chaque barre l'intervalle de confiance au seuil 0.001 sous l'hypothèse que les valeurs observées soient normalement distribuées. Le F-score et la p-valeur du test d'égalité des moyennes sont donnés pour chaque mesure.

Une première observation est que les variances des mesures sont généralement faibles. Cette remarque peut être faite pour différentes tailles de graphes. Ceci indique que les algorithmes étudiés sont consistants, c'est-à-dire qu'il fournissent une sortie similaire pour des entrées proches. Dans ce cadre, l'hypothèse d'égalité des moyennes est presque toujours rejetée. On peut alors comparer deux-à-deux les moyennes en utilisant la méthode de *Tukey-Kramer* [10]. Comme nous allons le voir, certains algorithmes fournissent des résultats proches. Toutefois, ce test n'étant pas transitif, on ne peut pas fournir un classement partiel des algorithmes basé sur son résultat. Notons également que deux algorithmes peuvent avoir une moyenne significativement différente sans que l'écart absolu entre les moyennes soit important par rapport à l'étendue de la mesure. Nous tiendrons compte de cela en comparant les algorithmes.

Encombrement

Si on s'intéresse aux chevauchements pouvant compliquer l'identification des communautés, on constate que les algorithmes de dessins tenant compte de l'existence d'un partitionnement produisent de meilleur résultats (voir Figure 6.5). Les algorithmes **Eades-GEM** et **Cluster-GEM** produisent des résultats semblables. **LinLog** produit moins de chevauchements arêtes ou sommets avec les communautés. On constate cependant que les chevauchements entre sommets ou avec les arêtes sont beaucoup plus importants avec cet algorithme. Ce constat doit être relativisé. En effet, les valeurs sont très faibles dans ces deux cas : les proportions sont proches de 0.

Un regroupement similaire des algorithmes peut être fait avec les mesures de croisements d'arêtes (voir Figure 6.6). Du point de vue des croisements internes, les algorithmes basés sur **GEM** se comportent de façon similaire. **FM**^{3} et **Linlog** produisent légèrement plus et moins de croisements respectivement. Le résultat pour **Linlog** peut sembler contre-intuitif car les communautés sont dessinées dans des zones réduites (voir Figure 6.4(e)),

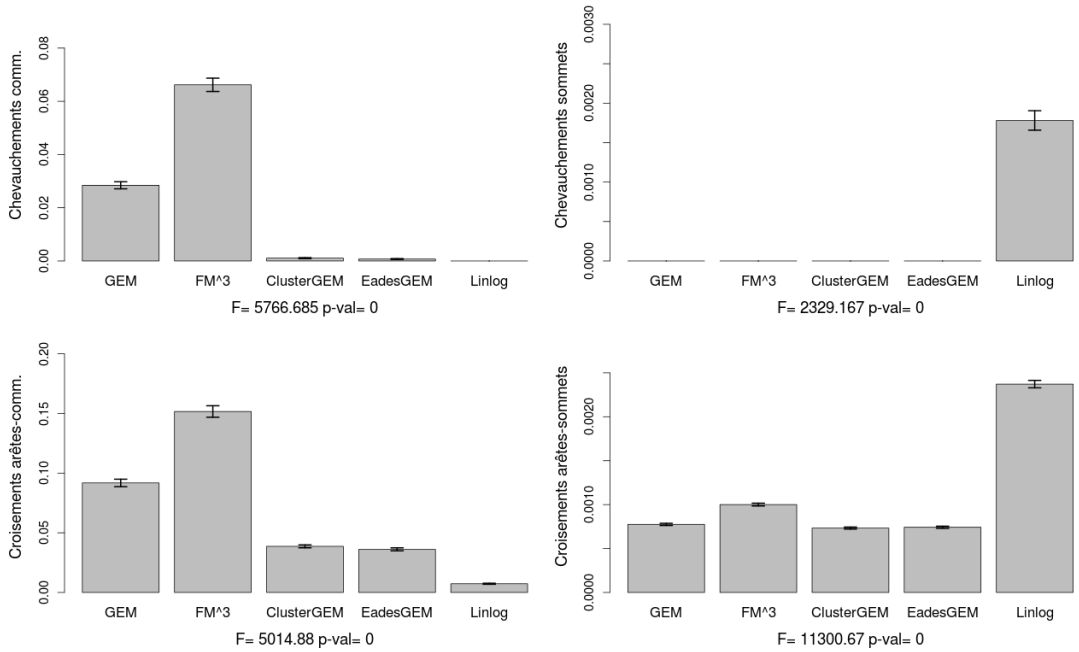


FIGURE 6.5: Moyennes des cinq algorithmes sur les mesures d'encombrement.

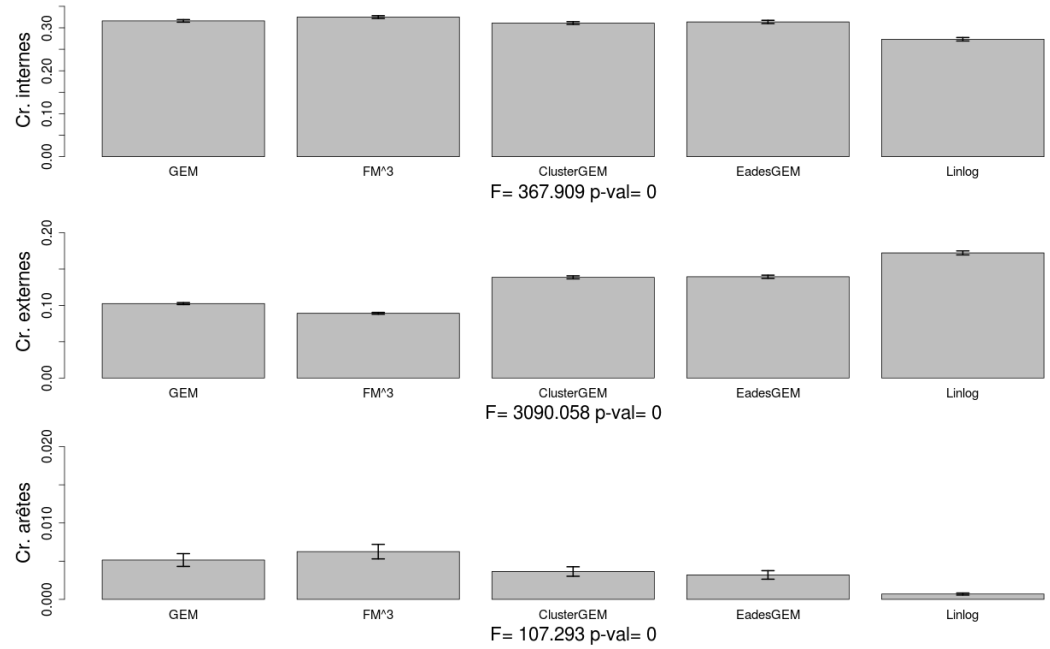


FIGURE 6.6: Moyennes des cinq algorithmes sur les mesures de croisements d'arêtes.

cela offre toutefois moins de contraintes quant au positionnement des sommets que dans le cas où les communautés sont proches entre elles.

Cette tendance s'inverse avec les croisements externes, où les algorithmes **GEM** et **FM**^{3} produisent de meilleurs résultats. Vouloir bien percevoir les communautés semble donc

mener à un plus grand nombre de croisements entre les arêtes les connectant. Cependant, si on regarde les croisements d'arêtes dans le cas nul, on constate que ces derniers sont globalement moins nombreux pour les algorithmes tenant compte de la structure de communauté. Notons que les écarts absolus entre les proportions sont faibles (entre 0 et 0.01). Les croisements internes ou externes sont ici séparés car il est possible de considérer les croisements internes comme "moins graves" que les croisements externes. Dans notre cas, les communautés ont une densité de connections importante. Il est naturel dans ce cas de les dessiner sans tenir compte des croisements car ces derniers seraient très nombreux peu importe le dessin utilisé.

Homogénéité des longueurs

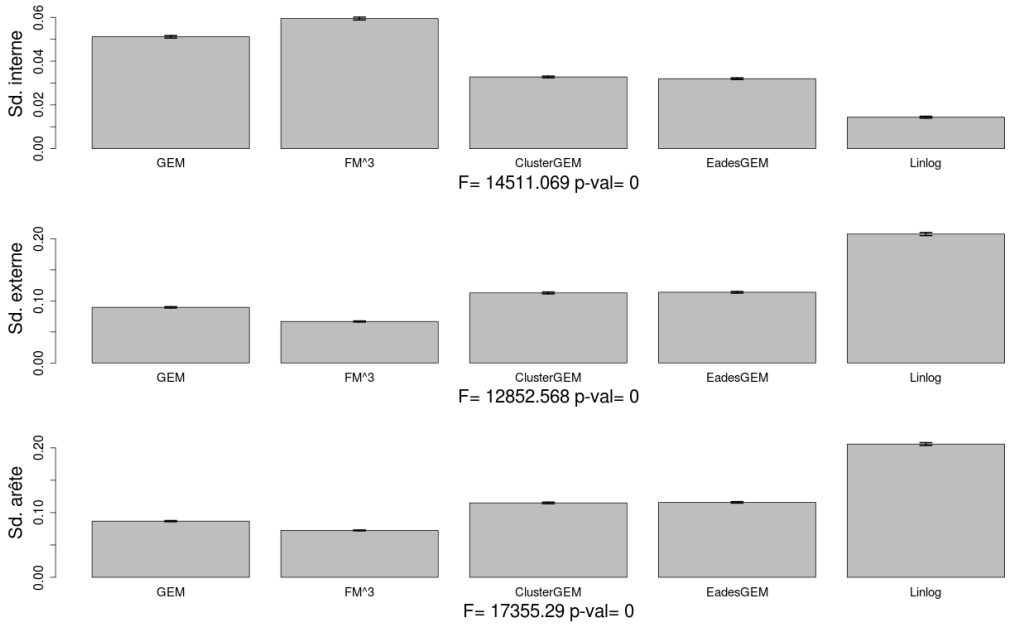


FIGURE 6.7: Moyennes des cinq algorithmes sur les mesures d'homogénéité des longueurs.

Pour tous les observations graphe/dessin réalisées ici, l'hypothèse d'égalité des longueurs moyennes entre arêtes internes et externes a toujours été rejetée. On peut donc s'intéresser à la variation des longueurs dans les deux cas. On constate que les arêtes internes sont de longueurs plus homogènes en utilisant l'algorithme **LinLog**. Cependant, la variance des longueurs d'arêtes externes est bien plus forte. Comme dit au début de cette section, les communautés générées par le modèle LFR n'ont pas d'attachements préférentiels, on pourrait donc s'attendre à ce que les arêtes externes aient des longueurs proches puisque aucune communauté n'est censée être plus proche topologiquement d'une autre. Ce constat devrait idéalement se retrouver dans le dessin. On peut donc dire que **LinLog** se comporte moins bien dans ce cas.

Résolution

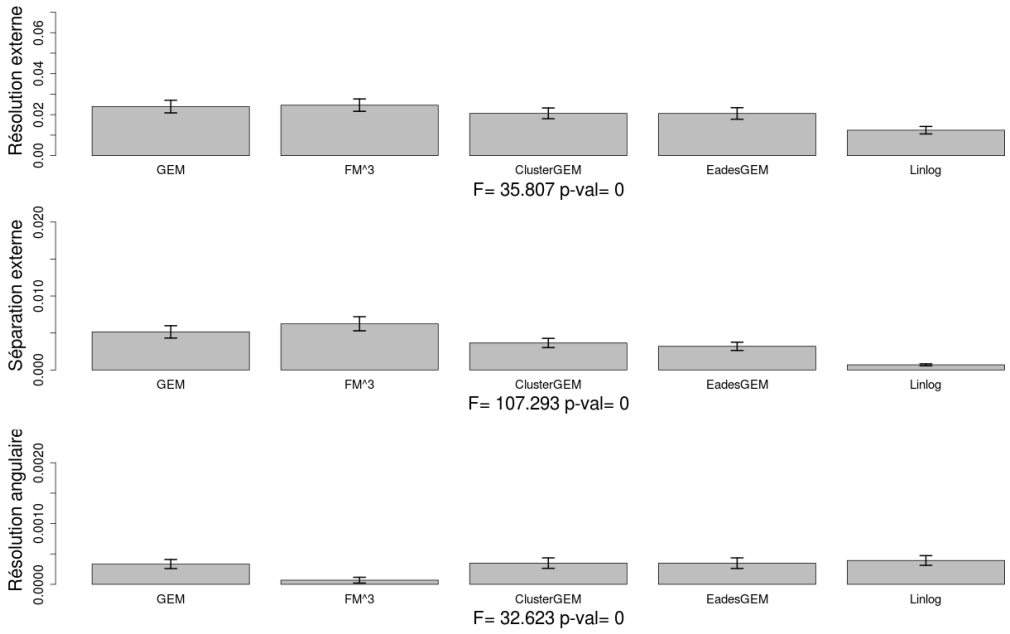


FIGURE 6.8: Moyennes des cinq algorithmes sur les mesures de résolution.

On constate ici aussi que **LinLog** se détache significativement des autres algorithmes (voir Figure 6.8). En effet, le test de Tukey-Kramer montre que, pour la résolution externe, les quatre autres algorithmes fournissent des résultats similaires. Ce n'est pas le cas si on s'intéresse à la séparation externe. Dans ce cas, les algorithmes **GEM** et **FM**^{3} semblent fournir des résultats similaires. Il en est de même pour **Cluster-GEM** et **Eades-GEM**. Dans les deux cas, **LinLog** produit de moins bons résultats. Pour ce qui est de la résolution angulaire "classique", **FM**^{3} fournit de moins bons résultats que les quatre autres algorithmes. Les moyennes pour ces derniers ne sont pas significativement différentes quand prises deux à deux.

Forme

On peut constater que **LinLog** crée des dessins où les communautés sont très faiblement dispersées (voir Figure 6.9), ce qui les rend facilement identifiables. De ce point de vue, les algorithmes **Cluster-GEM** et **Eades GEM** se comportent, encore une fois, de la même manière. Les algorithmes **FM**^{3} et **GEM** donnent des résultats différents. Les "convexités" sont très proches quel que soit l'algorithme utilisé. Notons donc au passage que les deux mesures employées ici capturent bien des phénomènes différents. **LinLog** fournit toutefois des communautés significativement plus convexes que les autres tandis qu'elles sont dessinées de manière plus concave avec **GEM**.

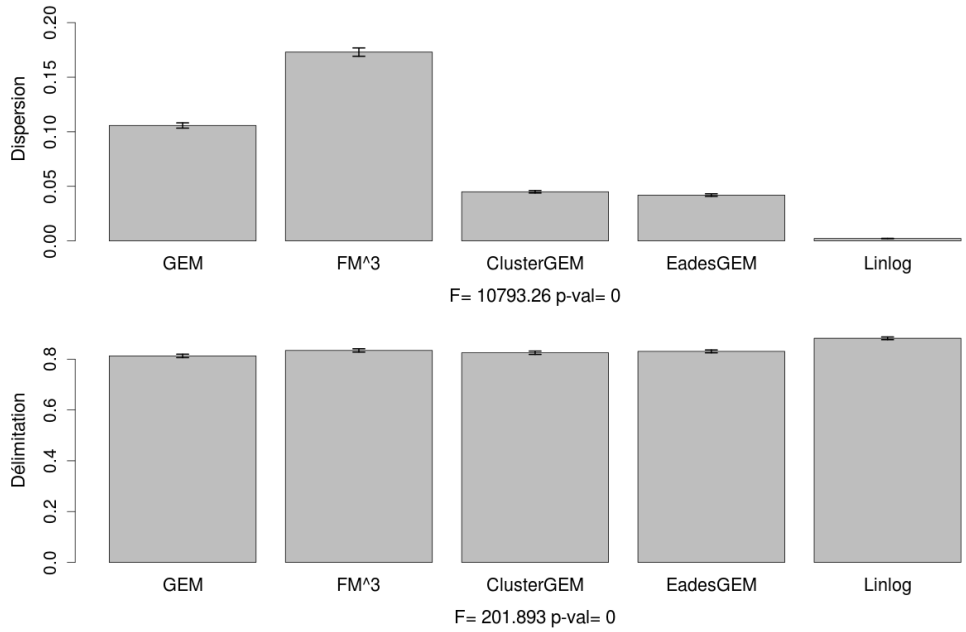


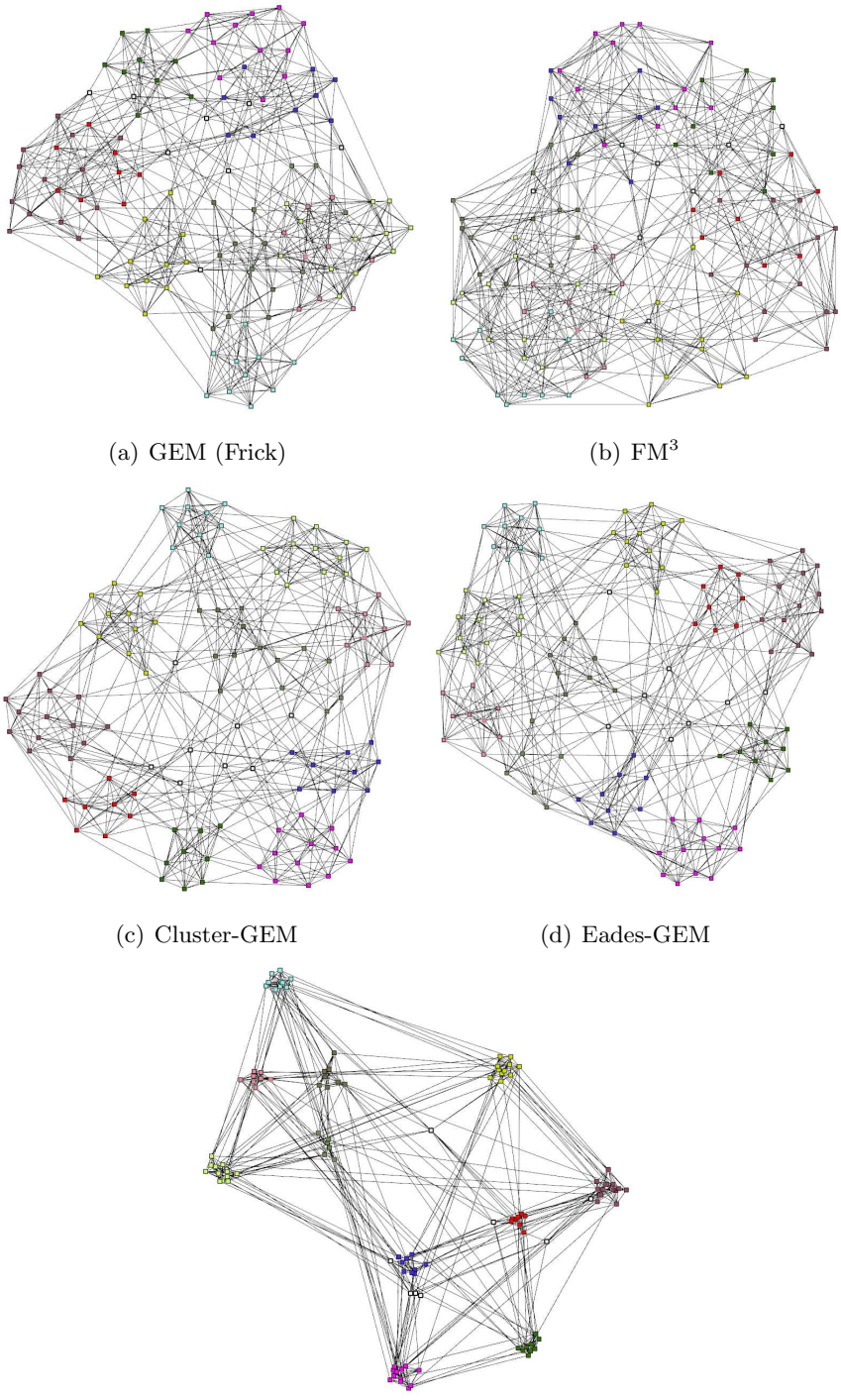
FIGURE 6.9: Moyennes des cinq algorithmes sur les mesures de forme des communautés.

6.4.4 Conclusions de l'étude

Les différents résultats obtenus ici correspondent globalement aux impressions obtenues en comparant visuellement les dessins (voir Figure 6.4). Les algorithmes **GEM** et **FM**^{3} sont globalement moins bons mais on observe cependant moins de croisements et des longueurs plus homogènes pour les arêtes externes. Ces deux algorithmes ne sont pas similaires : **FM**^{3} semble être moins performant, rappelons toutefois que cet algorithme est très efficace au niveau du temps de calcul.
On peut constater que les résultats de **Cluster-GEM** et **Eades-GEM** sont proches pour de nombreux critères. L'ajout du sommet virtuel dans chaque communauté est donc discutable car cela correspond à un coût algorithmique supplémentaire.

L'algorithme **LinLog** arrive à bien séparer les communautés même si la partition n'est pas une entrée de l'algorithme. C'est un avantage dont il faut tenir compte si on cherche à comparer ses résultats avec ceux de **Cluster-GEM** et **Eades-GEM**. Notons toutefois que les communautés peuvent ici être facilement identifiées par un algorithme de partitionnement. Il est assez surprenant de constater que l'algorithme obtient des dessins où les communautés sont à la fois très faiblement dispersées et où le nombre de croisements internes n'est pas plus important que pour les autres algorithmes. Ceci peut s'expliquer par le fait que les communautés sont très séparées et que les sommets ont plus de "liberté de mouvement". Cet algorithme souffre néanmoins de certains défauts : par exemple, le manque d'homogénéité dans les arêtes externes peut suggérer, à tort, que des communautés

sont plus proches entre elles que d'autres.



(a) GEM (Frick) (b) FM^3

(c) Cluster-GEM (d) Eades-GEM

(e) LinLog

FIGURE 6.10: Résultats des cinq algorithmes de dessin sur le réseau Football Universitaire. Les conférences sont représentées avec un code couleur sur les sommets. Les équipes indépendantes correspondent aux sommets blancs.

Les conclusions faites ici doivent tenir compte du fait que les graphes générés le sont

selon un modèle bien défini avec des paramètres fixes. Dans le cadre d'un système de visualisation, l'utilisateur pourra appliquer une procédure similaire pour déterminer l'algorithme de dessin le mieux adapté à ses données. Notons que d'autres facteurs comme la complexité en temps et en mémoire des différents algorithmes sont également à prendre en compte.

A titre illustratif, les sorties de cinq algorithmes utilisés ici sur le réseau de Football Universitaire, présenté dans la section 4.3, sont données en Figure 6.10. Les points forts et faibles des différents algorithmes peuvent également se retrouver en analysant les différents dessins. Les algorithmes **GEM (Frick)** et **FM**^{3} produisent en effet beaucoup de chevauchements entre communautés, ce qui n'est pas le cas pour les algorithmes **Cluster-GEM**, **Eades-GEM** et **LinLog**. Ce dernier rassemble avec succès les conférences dans des zones réduites du dessin. Ici, la proximité entre les groupes semble toutefois correspondre à la proximité géographique des conférences.

6.5 Discussion et Perspectives

Les mesures introduites ici sont cohérentes pour juger la qualité de représentation de la structure de communauté. En effet, nous généralisons ici des critères esthétiques bien connus à des dessins de graphes partitionnés. En se plaçant dans un cas nul, où chaque communauté correspond à un seul sommet, on retrouve les mesures esthétiques usuelles du domaine.
Une évaluation-utilisateurs serait toutefois importante pour déterminer quels sont les critères qui impactent le plus la réalisation de certaines tâches de recherche (comme le nombre de communautés, l'identification des arêtes internes/externes etc.). En particulier, on pourrait comparer ces résultats à l'étude réalisée par Van Ham et Rogowitz [141].

Nous pensons que notre travail peut aider à la conception et à la validation d'algorithmes de dessins. On pourrait, par exemple, concevoir des algorithmes cherchant à maximiser une fonction de score [39, 79]. Ce score intégrerait les mesures que nous avons proposées ici.

Le modèle LFR que nous utilisons pour notre étude suppose l'absence de connections préférentielles entre les communautés. Cette hypothèse est discutable si on s'intéresse à des réseaux réels. Un problème différent serait alors de considérer que les communautés peuvent se situer à plusieurs niveaux. La structure utilisée pour modéliser ce phénomène ne serait plus une simple partition mais une partition hiérarchique. Il serait intéressant de repenser les critères et les mesures esthétiques dans ce cadre, par exemple en considérant que les arêtes ne sont plus seulement internes ou externes mais internes jusqu'à (respectivement externes à partir d') un certain niveau de granularité. Certaines mesures esthétiques

proposées ici pourraient ainsi être généralisées au cas multi-niveaux en s'inspirant de la méthode décrite en Section 4. Il faut pour cela que les mesures respectent la contrainte d'additivité (voir Définition 2.51) (ce n'est pas le cas des mesures de résolution).

Notre hypothèse de travail est que la structure de communauté correspond à une partition des sommets. Toutefois, de nombreux travaux suggèrent que l'utilisation de fragmentations serait plus adaptée. Un sommet pourrait alors appartenir à différentes communautés. D'autres critères esthétiques, pour le dessin, sont à envisager. Par exemple, on peut considérer qu'un sommet ayant le même nombre de connections au sein de deux communautés doit être positionné entre les deux dans le dessin.

Chapitre 7

Visualisation de décompositions de graphes

La structure de communauté dans un réseau correspond à l'identification de *motifs* topologiques. Dans certains cas, la structure de communauté est modélisée par une fragmentation des sommets du réseau où des éléments peuvent être partagés entre différentes communautés. De nombreux algorithmes retournant ce type de décomposition ont été proposés dans ce contexte (voir par exemple [104], [89] ou [3]). Certaines communautés peuvent également être imbriquées entre elles.

L'analyse de ces résultats correspond au problème de visualisation d'ensembles chevauchants. Il est important pour un expert d'étudier les groupes un par un mais l'étude des relations entre les différents motifs et leur chevauchement est également importante. Des méthodes de visualisation dédiées sont nécessaires pour aider l'utilisateur dans ce processus d'exploration.
Pour ce problème, nous avons les contraintes suivantes : à partir d'un graphe $G = (V, E)$ dont les éléments (sommets et arêtes) sont plongés sur une surface à deux dimensions, nous cherchons à représenter explicitement une collection $(G_1, G_2, \ldots, G_l)$ de sous-graphes de G sans modifier ni le graphe, ni la position de ses éléments dans le diagramme. On peut en effet supposer que la position des éléments contient une sémantique particulière : c'est par exemple le cas dans un réseau de transports où les sommets sont associés à des coordonnées géographiques. Notons également qu'on ne suppose pas que les sous-graphes $(G_1, G_2, \ldots, G_l)$ sont des sous-graphes induits de G. En revanche, nous posons comme hypothèse que chacun de ces sous-graphes est connexe. Comme nous allons le voir dans la prochaine section, ce problème a déjà été largement étudié avec des contraintes relâchées ou en utilisant des algorithmes de complexité en temps exponentiel.

Nous proposons ici une méthode permettant de visualiser une collection de sous-graphes dans un diagramme de type nœuds-liens. Notre technique utilise la topologie du graphe pour générer des enveloppes concaves représentant chaque motif. Cette méthode ne

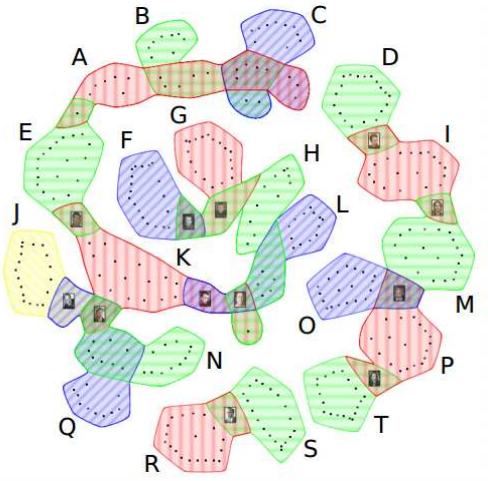
modifie pas les positions des éléments dans le diagramme. Un problème d'ordonnancement optimal des enveloppes est proposé dans ce contexte. Une heuristique est utilisée pour répondre à ce problème, cette dernière est particulièrement adaptée au cas où les inclusions entre les sous-graphes sont nombreuses. La méthode décrite dans cette section fournit une vue globale des ensembles chevauchants ainsi que la possibilité pour l'utilisateur d'effectuer une analyse locale.

7.1 État de l'art

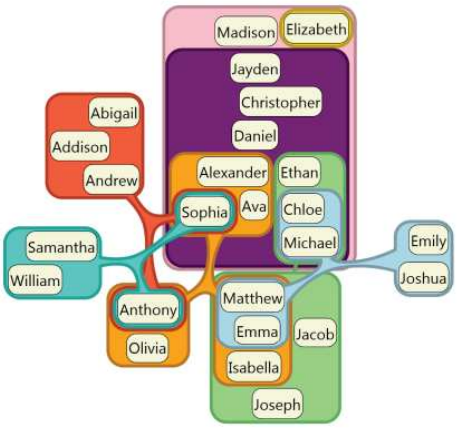
Le problème de la visualisation d'ensembles chevauchants a été largement étudié ces dernières années. Nous décrivons ici plusieurs solutions, dans certains cas les contraintes posées précédemment sont respectées.

Une première méthode consiste à créer un plongement particulier des éléments sur une surface respectant la proximité des groupes (en terme d'intersection). Simonetto *et al.* [136] proposent ainsi une technique permettant la génération de diagrammes d'Euler (voir Figure 7.1(a)). Les auteurs utilisent des enveloppes concaves pour représenter les ensembles. Différentes couleurs ou textures sont alors utilisées pour les distinguer. Toutefois, cette méthode ne permet de représenter qu'un nombre faible de chevauchements. Une autre technique proposée par Riche et Dwyer [120] utilise un algorithme de dessin de graphe basé sur la résolution de contraintes pour positionner les éléments. Les ensembles sont ici représentés par des enveloppes rectangulaires (voir Figure 7.1(b)). Ces deux approches supposent une modification du positionnement des éléments et ne sont donc pas adaptées à notre problème.

Certains travaux tiennent compte d'un positionnement préexistant en évitant tout ajustement. L'utilisation d'enveloppes convexes a été étudiée (voir [72, 45]), toutefois l'utilisation de ces formes augmente substantiellement le risque d'ambiguïtés visuelles, c'est-à-dire détecter à tort l'appartenance d'un élément à un ensemble. Collins *et al.* [34] utilisent des enveloppes continues et concaves pour délimiter les ensembles (voir Figure 7.1(c)). Une autre métaphore visuelle est proposée par Alper *et al.* [8] et consiste à représenter un chemin reliant les éléments appartenant à un même groupe (voir Figure 7.1(d)). Cette méthode est plus adaptée pour représenter des ensemble de points. Son utilité pour la visualisation de sous-graphes non induits n'est pas évidente, en effet la visualisation ne permet *a priori* que de représenter des sous-ensembles de points dans le dessin. Enfin ces deux méthodes ont un coût en temps de calcul important. Par exemple, le calcul des lignes reliant les éléments de chaque ensemble dans [8] peut être réduit au problème du *voyageur de commerce* [52] qui est NP-difficile. De plus, il n'existe pas, à ce jour, d'implémentation de leurs algorithmes.

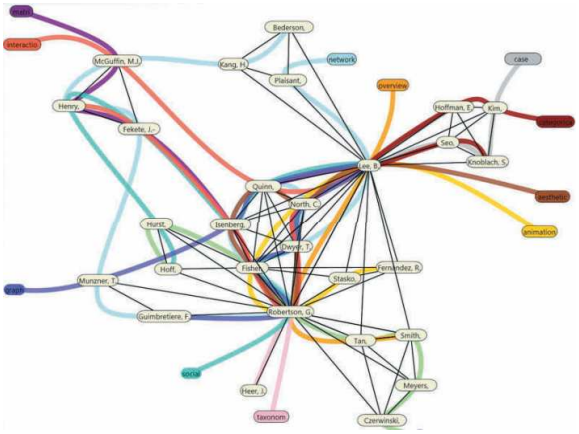


(a) Simonetto *et al.* [136]



(b) Riche et Dwyer [120]

(c) Collins *et al.* [34]



(d) Alper *et al.* [8]

FIGURE 7.1: Quatre méthodes pour la visualisation d'ensembles chevauchants. Ces images sont extraites des publications des différents auteurs.

7.2 Visualisation par enveloppes concaves utilisant la topologie du graphe

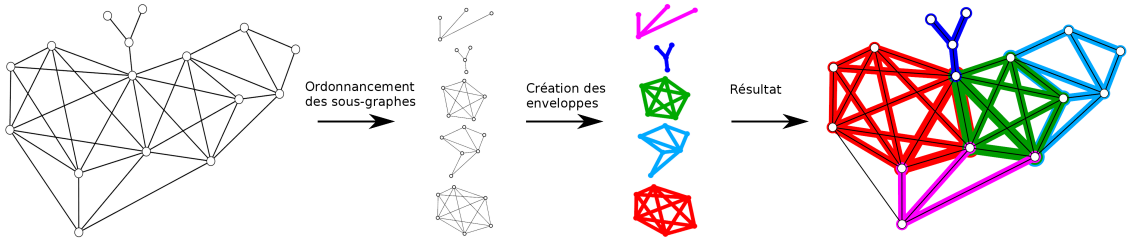


FIGURE 7.2: Les trois étapes pour la construction d'enveloppes concaves permettant la visualisation d'ensembles chevauchants dans un diagramme nœuds-liens.

Les différentes étapes que nous utilisons pour visualiser les sous-graphes sont illustrées dans la Figure 7.2. Nous allons utiliser des enveloppes concaves pour représenter les groupes. Ces enveloppes peuvent éventuellement contenir des trous. Elles sont ajoutées au dessin en tant que couches graphiques. Ceci permet en particulier que le graphe de base soit toujours visible (les sommets et les arêtes sont dessinées "par dessus" les enveloppes).

Nous cherchons à fournir une vue simultanée de l'ensemble des sous-graphes donnés en entrée. Nous voulons que cette vue réduise au maximum l'*occlusion topologique*. On entend par *occlusion topologique* le chevauchement des enveloppes représentant des sous-graphes qui partagent un ensemble d'éléments du graphe. Ce chevauchement va en effet générer de l'*occlusion visuelle* : une enveloppe recouvrira en partie une ou plusieurs autres enveloppes dans ce cas. Notons cependant que du chevauchement visuel peut exister sans qu'il corresponde à un chevauchement topologique. Un exemple montrant cette différence est donné en Figure 7.3. L'occlusion visuelle ne résultant pas d'une occlusion topologique pourrait être réduite en modifiant la position des éléments (on peut en effet dessiner le graphe en Figure 7.3 de façon planaire). Toutefois, nous avons ici comme contrainte de ne pas modifier la position des éléments du graphe.

Pour éviter à un utilisateur de détecter, à tort, l'appartenance d'un élément à une enveloppe sur laquelle il se situe, nous proposons des interactions simples. La première consiste à cliquer sur un élément du graphe pour ne conserver dans l'image que les enveloppes auxquelles l'élément en question appartient. La seconde consiste à sélectionner une seule enveloppe et occulter les autres. Ces interactions sont illustrées dans la Figure 7.4.

Notre procédure comprend deux étapes majeures. Dans un premier temps un ordonnancement des sous-graphes est réalisé de manière à assurer que deux enveloppes ne partageront pas de frontières communes. Ceci a pour but de réduire l'occlusion topologique. Cet ordonnancement permet également de déterminer dans quel ordre les enveloppes sont

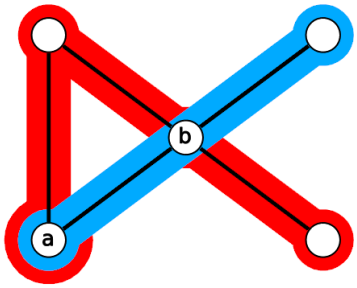
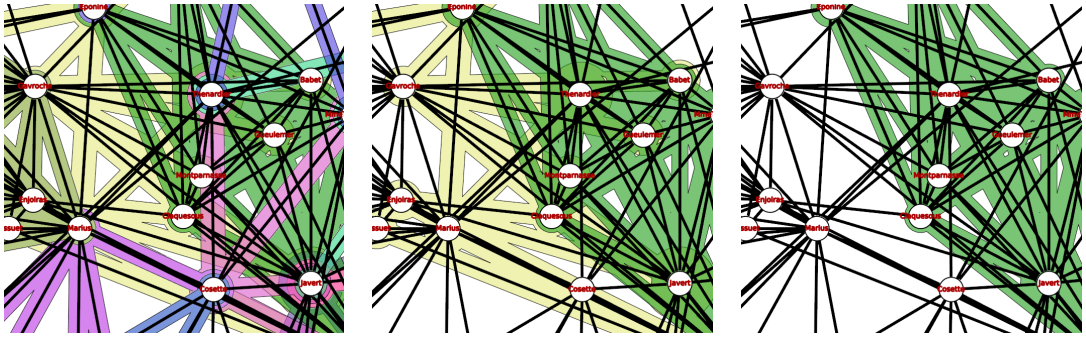


FIGURE 7.3: Illustration des différents problèmes d'occlusions avec deux ensembles (rouge et bleu). Le sommet nommé **a** implique un chevauchement topologique (il appartient aux deux sous-graphes). Le sommet nommé **b** crée de l'occlusion visuelle car il chevauche l'enveloppe bleue sans appartenir au sous-graphe correspondant.



(a) (b) (c)

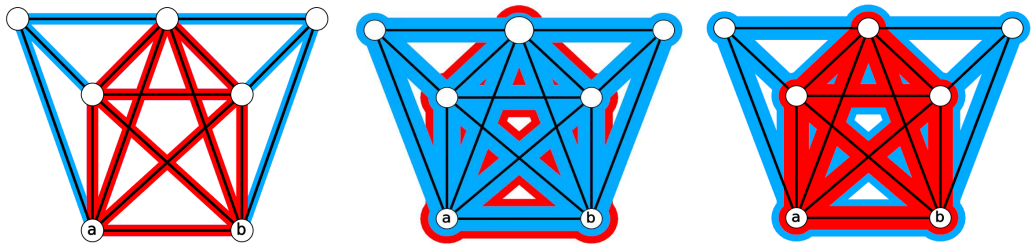
FIGURE 7.4: Illustration de techniques d'interactions simples pour résoudre les ambiguïtés visuelles. a) Image de base contenant toutes les enveloppes. Sur cet exemple, il est dur de dire si "Montparnasse" appartient au groupe vert. b) Image obtenue en cliquant sur "Montparnasse", seules les enveloppes auxquelles il appartient demeurent. Le sommet appartient bien à l'ensemble vert. c) Image obtenue en cliquant sur l'enveloppe verte. Ce sous-graphe est dense.

affichées. Dans un second temps, les enveloppes sont calculées en utilisant un algorithme de *clipping* de polygones. Cette étape peut être réalisée indépendamment pour chaque sous-graphe. Ces deux étapes sont plus amplement détaillées ci-dessous.

7.2.1 Ordonnancement des sous-graphes

On suppose ici que l'on peut générer des enveloppes avec un écart minimum noté $\Delta_i(u)$ entre la bordure de l'enveloppe correspondant au sous-graphe G_i et chaque élément (sommet ou arête) de G_i. Nous utilisons un ordre sur les sous-graphes pour calculer ces écarts. La démarche est expliquer dans cette section.

Pour chaque sommet ou arête du graphe noté u, nous associons un écart à chaque



(a) Deux enveloppes concaves ayant une bordure commune.

(b) Deux enveloppes concaves mal ordonnancées.

(c) Deux enveloppes concaves bien ordonnancées.

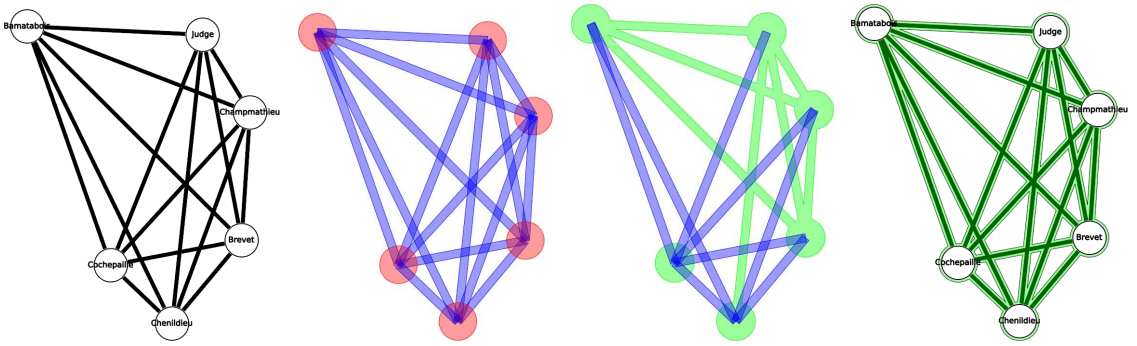
FIGURE 7.5: Illustration du problème d'ordonnancement lié à l'utilisation d'enveloppes concaves.

sous-graphe auquel u appartient. Cet ensemble sera noté $\mathcal{G}(u)$ et la liste des écarts correspondant est notée $(\Delta_1(u), \Delta_2(u), \ldots, \Delta_{|\mathcal{G}(u)|}(u))$. Les écarts doivent être différents pour chaque élément. Cette contrainte permet d'éviter le cas où une enveloppe occulte une autre. Ce problème est illustré dans la Figure 7.5(a), dans cet exemple il est en effet difficile de déterminer à première vue si l'arête (a, b) appartient à l'ensemble bleu.

On considère comme donné en paramètre un écart de base donné par l'utilisateur noté d. Cette quantité sera l'écart minimum entre deux enveloppes contenant au moins un élément en commun. Cet écart va permettre de bien distinguer, pour chaque élément, le nombre voire les différents sous-graphes auxquels l'élément appartient. Pour associer un écart à chaque enveloppe autour d'un élément u, nous utilisons un ordre sur les sous-graphes $\mathcal{G}(u) = [G_1, G_2, \ldots]$ et poser $\Delta_i(u) = id$. Remarquons que, dans ce cas, les enveloppes associées à $[G_1, G_2, \ldots]$ doivent être affichées dans l'ordre inverse (les plus grands avant les plus petits sous-graphes). En effet, si pour $i < j$, le sous-graphe G_i est un sous-graphe de G_j, alors l'enveloppe de G_i sera entièrement recouverte par l'enveloppe de G_j.

Les choix faits ici ne concernent que les enveloppes associées à un élément u. On peut toutefois facilement remarquer que la relation d'ordre entre deux sous-graphes G_i et G_j doit être la même pour chaque élément de $(G_i \cap G_j)$. Notre problème revient donc à choisir un ordre des sous-graphes $[G_1, G_2, \ldots, G_l]$. Pour chaque élément u, il suffit alors d'utiliser comme ordre la sous-séquence des sous-graphes auxquels l'élément appartient.

Tout ordre des sous-graphes est possible. Nous voulons toutefois refléter au maximum l'inclusion entre les sous-graphes *i.e.* si G_i est un sous-graphe de G_j alors G_i devrait se situer avant G_j dans $[G_1, G_2, \ldots, G_l]$. Cela va permettre d'éviter des situations telles que celle illustrée dans la Figure 7.5(b). Nous allons donc choisir d'ordonner les sous-graphes par taille (en terme de nombre de sommets) de façon croissante. Si ce choix permet de bien répondre à l'inclusion entre sous-graphes (voir Figure 7.5(c)), il n'est pas certain que



(a) Le sous-graphe. (b) Les polygones représentant chaque élément. (c) Union itérative des polygones. (d) Ajout de l'enveloppe en tant que couche graphique.

FIGURE 7.6: Illustration de la procédure de *clipping* de polygones.

d'autres solutions ne soit pas plus adaptées dans certains cas. Nous discutons le problème d'optimisation correspondant dans la Section 7.4.

7.2.2 Construction des enveloppes

Comme indiqué précédemment, l'écartement des enveloppes par rapport aux éléments doit être modulé en fonction de la position du sous-graphe dans l'ordonnancement. La procédure utilisée pour générer des enveloppes concaves doit donc prendre en compte un paramètre d'espacement entre les "squelettes" du sous-graphe (qui correspond aux sommets et aux arêtes) et les bordures externes. Certaines méthodes permettent la génération d'enveloppes concaves ayant recours à l'extraction d'isocontours dans l'espace image [34]. Il n'est cependant pas directement possible de moduler la taille des enveloppes créées dans ce cas.

La procédure nous permettant de créer des enveloppes concaves (avec ou sans trous) à partir d'un sous-graphe et d'un placement de ses éléments se base au contraire sur l'espace topologique. Une illustration du déroulement pour un sous-graphe donné se trouve en Figure 7.6. Notre solution se base sur le *clipping* de polygones. L'idée est la suivante : chaque sommet ou arête d'un sous-graphe (voir Figure 7.6(a)) est visuellement représenté par un cercle ou un rectangle. On crée pour chaque sommet un deuxième cercle centré à la position du sommet et de même rayon plus la largeur désirée pour ce sous-graphe. On fait de même pour chaque arête. Les polygones créés englobent les éléments du sous-graphe qui leur sont associés (voir Figure 7.6(b)). L'union de ces polygones forme ainsi une enveloppe concave (avec trous) entourant le motif (voir Figure 7.6(d)).

L'étape importante dans cette procédure est le calcul de l'union des différents polygones. Nous utilisons pour cela la librairie `Clipper` implémentant l'algorithme de Vatti [142]

répondant au problème de *clipping* de polygones. L'union des différents formes se fait de manière itérative comme illustré dans la Figure 7.6(c). La fonction permettant l'union d'un ensemble de polygones retourne à son tour plusieurs polygones : un premier qui correspond à la bordure extérieure et plusieurs autres qui correspondent aux trous. L'utilisateur pourra donc choisir d'afficher ou non les trous dans les enveloppes. Retirer les trous améliore la vue d'ensemble des différentes enveloppes et accélère les temps de calcul, il faut toutefois s'attendre à ce que l'occlusion soit bien plus importante.

Notons, pour finir, que le coût algorithmique de notre méthode est dominé par le calcul des enveloppes (qui peut être parallélisé dans notre cas). En effet, l'ordonnancement des sous-graphes revient à effectuer un tri par rapport au nombre de sommets.

7.3 Applications

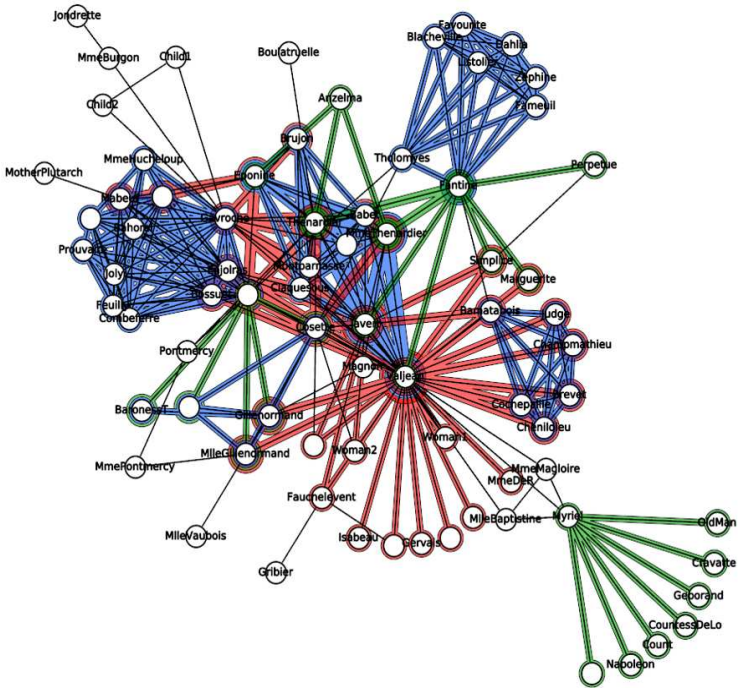
Nous illustrons ici notre procédure sur différents exemples. Remarquons premièrement que l'ordonnancement des sous-graphes en fonction de leur taille est particulièrement adapté si la décomposition du graphe se réduit à une partition hiérarchique des sommets. Dans ce cadre, un sous-graphe sera toujours placé au dessus de ses ancêtres dans l'image. La plupart des visualisations de partitions hiérarchiques de ce manuscrit ont été générées en utilisant cette procédure.

Réseau de co-occurrence *Les Misérables* [78]. Nous montrons ici l'intérêt de notre méthode pour visualiser les informations issues d'une décomposition du réseau des *Misérables*. Les 77 sommets du graphe correspondent aux personnages du roman de Victor Hugo et une arête relie deux personnages si ils apparaissent ensemble dans au moins un chapitre. Le graphe contient 254 arêtes (co-occurrences entre personnages).

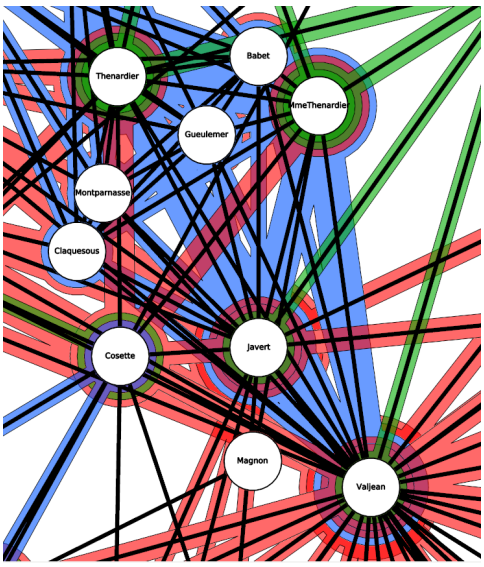
Nous allons étudier la décomposition obtenue en utilisant l'algorithme de Ahn *et al.* [3]. Cet algorithme fournit une partition des arêtes du réseau. Les arêtes sont groupées en fonction de la similarité entre le voisinage direct de leurs extrémités. Ainsi les sous-graphes détectés par l'algorithme ne sont pas uniquement des groupes de sommets fortement interconnectés. Nous allons différencier ici trois types de sous-graphes :

- **Sous-graphe dense :** Un sous-ensemble de sommets avec une forte proportion d'arêtes entre eux.
- **Sous-graphe biparti :** Ensemble d'arêtes connectant deux sous-graphes denses.
- **Sous-graphe arbre :** Sous-graphe étant un arbre qui peut également connecter des sous-graphes denses.

Nous pouvons utiliser notre procédure pour visualiser ces différents motifs dans une seule image. Pour cela, nous colorions les enveloppes en fonction du type de sous-graphe que l'enveloppe représente.



(a) Vue globale



(b) Zoom sur une partie dense de l'image

FIGURE 7.7: Visualisation d'une décomposition chevauchante du réseau *Les Misérables* [78] obtenue en utilisant l'algorithme de Ahn *et al.* [3]. Le positionnement des sommets est obtenu en utilisant l'algorithme GEM (Frick) [58]. Un sommet est étiqueté avec le nom du personnage correspondant. Les sous-graphes sont classés en trois catégories : les denses (en bleu), les bipartis (en rouge) et les arbres (en rouge).

Les résultats sont données en Figure 7.7. La vue globale (voir Figure 7.7(a)) permet de voir les différents groupes. Les enveloppes bleues ne chevauchent pas d'enveloppes du même type. Ainsi un personnage appartient donc au plus à un groupe dense.

Le centre du dessin est plus difficile à analyser. Celui-ci rassemble des sous-graphes bipartis connectant des groupes denses. La Figure 7.7(b) est un zoom sur cette zone. Rappelons que des sous-graphes ayant des sommets communs sont dessinés avec des largeurs différentes. On peut donc déterminer pour chaque sommet les différents types de sous-graphes auxquels le sommet appartient. Les personnages Valjean et Javert appartiennent tous deux à quatre sous-graphes bipartis. Ceci semble logique étant donné qu'ils jouent un rôle de médiateurs dans le roman. Notons également que Madame Thénardier appartient aux mêmes groupes que Monsieur Thénardier (son mari dans le roman).

7.4 Discussion et Perspectives

Nous avons présenté une nouvelle méthode pour visualiser une décomposition d'un graphe représenté par un diagramme nœuds-liens. La procédure génère des enveloppes concaves pour chaque motif sans modifier le positionnement des éléments dans le diagramme. Les enveloppes partageant des éléments communs peuvent être distinguées, en effet elles vont disposer de largeurs différentes.

Pour déterminer la largeur des enveloppes ainsi que l'ordre de rendu de celles-ci, les sous-graphes sont classés par ordre de taille (en terme de nombre de sommets) croissant. Ce choix n'est peut être pas le plus adapté dans tous les cas. Nous pourrions ainsi trier les sous-graphes selon d'autres critères, par exemple la moyenne des distances entre les sommets. De manière plus générale, il serait intéressant de déterminer le meilleur ordre en fonction d'une mesure objective. Cette mesure pourrait par exemple se baser sur l'aire visible des enveloppes. Dans ce cadre, l'*aire visible* de l'enveloppe d'un sous-graphe G_i, notée $A^*(G_i)$, est la surface qui n'est pas recouverte par les enveloppes des sous-graphes $[G_1, G_2, \ldots, G_{i-1}]$.

Nous pensons qu'il serait opportun de maximiser l'entropie dans la distribution des aires visibles (voir définition 7.1). Cette mesure peut être interpréter comme l'incertitude quant à l'enveloppe se trouvant en un point de l'image choisi aléatoirement (voir définition 2.53). Cette mesure va donc fortement pénaliser des ordres pour lesquels des enveloppes sont totalement recouvertes.

Définition 7.1 ***(Entropie de la distribution des aires visibles)*** *Soit un ordre total $\mathcal{G} = [G_1, G_2, \ldots, G_l]$ sur les sous-graphes de G. Soit $A(\mathcal{G})$ l'aire de l'union des enveloppes (invariant par rapport à l'ordre choisi). L'entropie dans la distribution des aires visibles,*

notée $H(\mathcal{G})$, est donnée par la formule suivante.

$$H(\mathcal{G}) = -\sum_{i=1}^{l} \frac{A^*(G_i)}{A(\mathcal{G})} \log_2 \left(\frac{A^*(G_i)}{A(\mathcal{G})} \right) \tag{1}$$

La recherche d'un algorithme répondant au problème de la maximisation de ce type de mesure est (à notre connaissance) un problème ouvert. Notons qu'un algorithme trivial revient à parcourir l'ensemble des ordonnancements possibles, c'est-à-dire $l!$ permutations, ce qui aboutit à un algorithme exponentiel.
La résolution de ce type de problème devrait en outre tenir compte non seulement du chevauchement topologique entre les différents sous-graphes mais également du chevauchement visuel. Cet aspect n'est pas directement intégré à notre méthode.

Cependant, la question du meilleur ordonnancement dépend des critères choisis. Pour déterminer si la mesure proposée ici est pertinente il faudrait conduire plusieurs études-utilisateur. Cela permettrait également de comparer notre méthode avec les autres visualisations disponibles dans la littérature.

Dans la méthode proposée, l'écartement entre les éléments du graphe et les enveloppes dépend d'un paramètre fourni par l'utilisateur. Il serait toutefois possible de déterminer l'espace maximum disponible autour de chaque élément. Cet espace pourrait ensuite être partagé entre les différentes enveloppes. Cette approche a plusieurs avantages : diminuer le nombre de paramètres, optimiser l'utilisation de l'espace vide et éviter de créer du chevauchement entre les enveloppes entourant des éléments proches.

Chapitre 8
Conclusion et perspectives

Nous avons abordé dans cette thèse trois problématiques liées à la décomposition de réseaux complexes modélisés par des graphes : les mesures de partitionnement, les algorithmes pour le partitionnement et la visualisation de graphes partitionnés. La décomposition de graphes étant un domaine de recherche relativement récent, l'évaluation de ces décompositions est essentielle car elle permet de concevoir et de tester des algorithmes. Cette évaluation peut se faire par le biais de *mesures de qualité* mais également en fournissant des *métaphores visuelles* permettant à un utilisateur d'analyser les résultats.

Le chapitre 3 a été consacré aux mesures permettant d'évaluer la qualité d'une partition plate des sommets d'un graphe. Nous avons présenté différentes approches telles que la *modularité*, la *qualité de compression* et la mesure MQ se basant sur les densités locales des groupes. Nous avons proposé une extension de cette dernière mesure en introduisant une pondération sur chaque groupe et la notion de "graphe de références" qui permet de définir un cas idéal auquel les groupes devraient correspondre : typiquement des cliques déconnectées du reste du graphe. Nous avons analysé cette mesure en la comparant à la *modularité* et montré qu'elle possédait de bonnes propriétés.
Nous pensons toutefois qu'il n'existe pas de meilleure mesure de qualité, c'est-à-dire une mesure adaptée à tous les cas d'études. Une mesure de qualité permet de formaliser des critères informels sur le type de partition ou sur la forme des groupes qui sont recherchés. La mesure MQ pondérée semble adaptée au partitionnement des réseaux de *commuters* (voir section 5.4) car la présence de sommets isolés ne pénalise pas fortement la qualité de la partition.

Dans le chapitre 4, nous avons présenté une généralisation des mesures de qualité additives aux partitions hiérarchiques. Nous associons à un arbre de partition un polynôme à une variable qui permet de favoriser les partitions plus ou moins profondes. Il est cependant possible d'utiliser l'intégrale de ce polynôme si il n'y a pas de préférence sur la profondeur de la hiérarchie recherchée. Nous avons validé analytiquement et empiriquement cette nouvelle approche en étudiant le comportement de la mesure MQ multi-niveaux sur différents

exemples. Des exemples simples ont permis de constater que, pour obtenir la meilleure partition hiérarchique, il faut dans certains cas se baser sur des partitions plates *a priori* sous-optimales.

Disposer d'une mesure de qualité pour les partitions hiérarchiques permet de comparer différentes hiérarchies mais également de concevoir des procédures pour extraire ces structures. Ce fut l'objet du chapitre 5. Nous avons introduit une procédure de post-traitement permettant d'améliorer la qualité d'une partition hiérarchique en supprimant itérativement des regroupements internes. Cette heuristique a une complexité polynomiale dépendant de la taille de l'arbre de partition (en termes de nombres de nœuds et de profondeur). En utilisant le *benchmark* LFR hiérarchique, nous avons constaté que cette procédure permettait effectivement de rapprocher la hiérarchie produite indirectement par l'algorithme *Louvain* [23] de la vraie hiérarchie. L'utilisation de cette mesure est une bonne alternative à l'application récursive de cet algorithme. Nous avons également discuté la construction de partitions hiérarchiques sur un cas d'étude concret à savoir un réseau de *commuters* français.

Dans le chapitre 6, nous nous sommes intéressés au problème de dessin de graphes modélisant des réseaux possédant une structure de communauté. Dans ce cadre, nous avons proposé une généralisation de critères esthétiques connus. Ces mesures ont permis de comparer la qualité des dessins produits par différents algorithmes basés sur des modèles de forces. On a constaté que la prise en compte de la structure de communauté dans les algorithmes améliorait globalement la qualité des dessins au détriment d'autres critères qui peuvent être considéré comme moins importants d'après des évaluations utilisateurs (comme le nombre de croisements au sein des groupes).

Dans le chapitre 7, nous avons présenté une méthode pour générer des enveloppes concaves afin de représenter explicitement une décomposition d'un graphe dans un diagramme nœuds-liens. Cet algorithme tire parti de la topologie du graphe pour créer des enveloppes en utilisant un algorithme de *clipping* de polygones. Une heuristique d'ordonnancement de ces enveloppes permet de réduire les problèmes d'occlusion liés à l'appartenance de certains éléments à différents fragments. Elle est particulièrement adaptée dans le cas où la décomposition est une partition hiérarchique.

8.1 Perspectives

Nous détaillons ici plusieurs perspectives ou travaux en cours se basant sur le contenu de cette thèse. Ces perspectives ne couvrent pas toutes celles évoquées en conclusion de chapitre.

8.1.1 Distances entre partitions hiérarchiques

Une fonction de distance entre partitions (voir définition 2.6) est un outil important : elle permet de comparer les partitions extraites automatiquement par des algorithmes à une vérité terrain, de déterminer un consensus entre différentes partitions [65] ou de comprendre le comportement d'une mesure de qualité (comme nous l'avons fait dans la section 3.3.3). On peut formuler les mêmes problèmes dans le cadre de partitions hiérarchiques, il serait donc intéressant de disposer d'une distance pour ces objets. Cette perspective a été brièvement évoquée dans la section 5.5. Notons cependant que ce problème dépasse le partitionnement de graphe. La littérature est, à notre connaissance, pauvre à ce sujet et, dans beaucoup de cas, la comparaison de partitions hiérarchiques se fait en comparant les niveaux deux à deux.

Une partition hiérarchique peut être représentée comme un arbre de partition ; c'est d'ailleurs cette modélisation qui est utilisée dans cette thèse. Dans ce cadre, les méthodes de réécriture d'arbres [21], issues de la combinatoire, pourraient être employées. Toutefois, dans le cas de partitions hiérarchiques, les arbres sont non-ordonnés et les étiquettes des nœuds dépendent des étiquettes de leur descendants. Nous pourrions également relier ce problème à l'évaluation de distances entre des *arbres phylogénétiques* [123]. Les arbres phylogénétiques permettent de modéliser la proximité entre différentes espèces du vivant. Cependant, ces arbres sont généralement binaires et ne sont pas toujours enracinés.

Les partitions plates sont un cas particulier de partitions hiérarchiques, nous pensons qu'il est naturel qu'une distance sur les partitions hiérarchiques généralise une distance sur les partitions plates. Nous travaillons actuellement sur l'adaptation de la *distance par suppression* (voir définition 2.7). L'avantage de cette mesure est qu'elle a une interprétation claire, c'est le nombre minimum d'échanges devant être réalisés entre les groupes pour obtenir deux partitions identiques. Ce concept pourrait, par exemple, être étendu à l'échange d'éléments entre groupes partageant le même ancêtre commun dans une hiérarchie.

8.1.2 Applications à la fragmentation de graphes

Les travaux présentés ici, en particulier la généralisation des mesures de qualité aux partitions hiérarchiques, peuvent servir de base pour développer des méthodes de décompositions différentes du partitionnement. Nous avons évoqué la possibilité d'utiliser des fragmentations pour capturer la structure de communautés de réseaux complexes puisqu'elles permettent de modéliser les chevauchements entre communautés. Ce sujet a fait l'objet de nombreux travaux évoqués en introduction de ce document. Bien que les fragmentations incluent les partitions hiérarchiques en cas particulier, des inclusions entre les

fragments peuvent être réalisées en agrégeant certains fragments (cette méthode est utilisée dans l'algorithme *OSLOM* [89]) ou en appliquant récursivement un algorithme de fragmentation. De telles structures peuvent *a priori* être évaluées en utilisant des mesures multi-niveaux telles que celles proposées dans le chapitre 4. Il faudrait pour cela utiliser des mesures de qualité de fragmentations, des généralisations de certaines mesures usuelles ont par exemple été proposées dans [24].

8.1.3 Dessins multi-niveaux de graphes

Un domaine d'application important des partitions hiérarchiques est le dessin multi-niveaux de graphes. En effet, les algorithmes de dessins, tels que ceux par modèle de forces [58, 103], ont des temps d'exécution relativement longs. Or un partitionnement hiérarchique permet de mettre en place des stratégies du type *"diviser pour régner"* : dessiner indépendamment les groupes de la hiérarchie en se basant sur les dessins faits de leurs descendants. Ce type d'approche est utilisée dans [69] ou [145] pour améliorer les performances et la qualité des algorithmes par modèle de forces. Dans [12], les auteurs proposent d'utiliser des algorithmes de dessins différents selon le type du sous-graphe induit par chaque groupe.

Comme nous l'avons montré vu dans le chapitre 6, l'algorithme FM^3 se base sur une décomposition hiérarchique mais ne permet pas de représenter efficacement la structure de communauté du réseau, cela est peut-être dû à la façon dont cette décomposition est construite. Ce sont toutefois certaines propriétés de cette hiérarchie (comme par exemple le nombre maximum de sommets dans un groupe) qui permettent la réduction de complexité obtenue avec cet algorithme. Une perspective intéressante est donc de concevoir des méthodes pour calculer des partitions hiérarchiques qui respectent ce type de propriétés tout en capturant au mieux la structure de communauté du graphe.

Chapitre 8
Bibliographie

[1] E. H. L. Aarts and v. Laarhoven. Statistical cooling : A general approach to combinatorial optimization problems. *Philips J. Res.*, 40(4) :193–226., 1985.

[2] James Abello, Frank van Ham, and Neeraj Krishnan. Ask-graphview : A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, 12(5) :669–676, 2006.

[3] Y.Y. Ahn, J.P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307) :761–764, 2010.

[4] H. Akima. A method of bivariate interpolation and smooth surface fitting for irregularly distributed data points. *ACM Transactions on Mathematical Software (TOMS)*, 4(2) :148–159, 1978.

[5] R. Albert and A.L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1) :47–97, 2002.

[6] Rodrigo Aldecoa and Ignacio Marín. Deciphering network community structure by surprise. *PloS one*, 6(9) :e24195, 2011.

[7] Anthony Almudevar and Chris Field. Estimation of single-generation sibling relationships based on dna markers. *Journal of Agricultural, Biological, and Environmental Statistics*, pages 136–165, 1999.

[8] B. Alper, N. Riche, G. Ramos, and M. Czerwinski. Design study of linesets, a novel set visualization technique. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12) :2259–2267, 2011.

[9] J. Ignacio Alvarez-Hamelin, Luca Dall'Asta, Alain Barrat, and Alessandro Vespignani. K-core decomposition of internet graphs : hierarchies, self-similarity and measurement biases. *Networks and Heterogeneous Media*, 3(2) :371–293, 2008.

[10] SJ Amster. Beyond anova, basics of applied statistics. *Technometrics*, 29(3) :387–387, 1987.

[11] Konstantin Andreev and Harald Racke. Balanced graph partitioning. *Theory of Computing Systems*, 39(6) :929–939, 2006.

[12] Daniel Archambault, Tamara Munzner, and David Auber. Topolayout : Multilevel graph layout by topological features. *Visualization and Computer Graphics, IEEE Transactions on*, 13(2) :305–317, 2007.

[13] Alex Arenas, Alberto Fernandez, and Sergio Gomez. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10(5) :053039, 2008.

[14] D. Auber, Y. Chiricota, G. Melancon, and F. Jourdan. Multiscale navigation of small world networks. In *IEEE Symposium on Information Visualisation*, pages 75–81, Seattle, GA, USA, 2003. IEEE Computer Science Press.

[15] David Auber, Daniel Archambault, Romain Bourqui, Antoine Lambert, Morgan Mathiaut, Patrick Mary, Maylis Delest, Jonathan Dubois, and Guy Melançon. The Tulip 3 Framework : A Scalable Software Library for Information Visualization Applications Based on Relational Data. Rapport de recherche RR-7860, INRIA, January 2012.

[16] A.L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439) :509, 1999.

[17] Marc Barthélemy. Spatial networks. *Physics Reports*, 499(1) :1–101, 2011.

[18] Michael Batty. Hierarchy in cities and city systems. In D. Pumain, editor, *Hierarchy in natural and social sciences*, volume 3 of *Methodos series*, pages 143–168. Springer, 2006.

[19] R. Bauer, M. Krug, and D. Wagner. Enumerating and generating labeled k-degenerate graphs. In *7th Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 90–98, 2010.

[20] C. Biemann. Chinese whispers : an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, pages 73–80. Association for Computational Linguistics, 2006.

[21] P. Bille. A survey on tree edit distance and related problems. *Theoretical computer science*, 337(1-3) :217–239, 2005.

[22] J. Blitzstein and P. Diaconis. A sequential importance sampling algorithm for generating random graphs with perscribed degrees. *Journal of Internet Mathematics*, 6(4) :489–522, 2010.

[23] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics : Theory and Experiment*, 2008(10) :P10008, 2008.

[24] Romain Bourqui. *Décomposition et Visualisation de graphes : Applications aux Données Biologiques*. These, Université Sciences et Technologies - Bordeaux I, October 2008.

[25] F. Boutin and M. Hascoët. Cluster Validity Indices for Graph Partitioning. In *IV'04 : 8th IEEE International Conference on Information Visualization*, pages 376–381, London (UK), 2004. IEEE.

[26] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2) :163–177, 2001.

[27] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 20(2) :172–188, 2008.

[28] Rainer Burkard, Mauro Dell'Amico, and Silvano Martello. *Assignment problems*. Cambridge University Press, 2012.

[29] Chandra S Chekuri, Andrew V Goldberg, David R Karger, Matthew S Levine, and Cliff Stein. Experimental study of minimum cut algorithms. In *Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms*, pages 324–333. Society for Industrial and Applied Mathematics, 1997.

[30] Markus Chimani, Carsten Gutwenger, Michael Jünger, Karsten Klein, Petra Mutzel, and Michael Schulz. The open graph drawing framework. In *15th International Symposium on Graph Drawing*, pages 23–26, 2007.

[31] A. Clauset, C. Moore, and M.E.J. Newman. Structural inference of hierarchies in networks. In *Proceedings of the 2006 conference on Statistical network analysis*, pages 1–13. Springer-Verlag, 2006.

[32] A. Clauset, C. Moore, and M.E.J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191) :98–101, 2008.

[33] A. Clauset, M.E.J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(6) :66111, 2004.

[34] C. Collins, G. Penn, and S. Carpendale. Bubble sets : Revealing set relations with isocontours over existing visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6) :1009–1016, 2009.

[35] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

[36] Michele Coscia, Giulio Rossetti, Fosca Giannotti, and Dino Pedreschi. Demon : a local-first discovery method for overlapping communities. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 615–623. ACM, 2012.

[37] Michael AA Cox and Trevor F Cox. Multidimensional scaling. In *Handbook of data visualization*, pages 315–347. Springer, 2008.

[38] J.J. Daudin, F. Picard, and S. Robin. A mixture model for random graphs. *Statistics and computing*, 18(2) :173–183, 2008.

[39] R. Davidson and D. Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics (TOG)*, 15(4) :301–331, 1996.

[40] Javier De Las Rivas and Celia Fontanillo. Protein–protein interactions essentials : key concepts to building and analyzing interactome networks. *PLoS computational biology*, 6(6) :e1000807, 2010.

[41] A. De Montis, S. Caschili, and A. Chessa. Commuter networks and community detection : a method for planning sub regional areas. *Arxiv preprint arXiv :1103.2467*, 2011.

[42] Maylis Delest and Jean-Marc Fédou. Attribute grammars are useful for combinatorics. *Theoretical Computer Science*, 98(1) :65–76, 1992.

[43] Luca Donetti and Miguel A Munoz. Detecting network communities : a new systematic and efficient algorithm. *Journal of Statistical Mechanics : Theory and Experiment*, 2004(10) :P10012, 2004.

[44] Selan Dos Santos and Ken Brodlie. Gaining understanding of multivariate and multidimensional data through visualization. *Computers & Graphics*, 28(3) :311–325, 2004.

[45] Tim Dwyer, Kim Marriott, Falk Schreiber, Peter Stuckey, Michael Woodward, and Michael Wybrow. Exploration of networks using overview+detail with constraint-based cooperative layout. *IEEE Transactions on Visualization and Computer Graphics*, 14 :1293–1300, November 2008.

[46] Peter Eades and Qing-Wen Feng. Multilevel visualization of clustered graphs. In *Graph Drawing*, pages 101–112, 1996.

[47] Peter Eades and Mao Lin Huang. Navigating clustered graphs using force-directed methods. *J. Graph Algorithms Appl.*, 4(3) :157–181, 2000.

[48] P. Erdős and A. Rényi. On random graphs i. *Publ. Math. Debrecen*, 6 :290–297, 1959.

[49] Leonhard Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, 8 :128–140, 1741.

[50] K. Fischer, B. Gärtner, and M. Kutz. Fast smallest-enclosing-ball computation in high dimensions. *Algorithms-ESA 2003*, pages 630–641, 2003.

[51] P. Flajolet and R. Sedgewick. *Analytic combinatorics*. Cambridge Univ Pr, 2009.

[52] Merrill M Flood. The traveling-salesman problem. *Operations Research*, 4(1) :61–75, 1956.

[53] M. Formann, T. Hagerup, J. Haralambides, M. Kaufmann, FT Leighton, A. Symvonis, E. Welzl, and G. Woeginger. Drawing graphs in the plane with high resolution. *SIAM J COMPUT.*, 22(5) :1035–1052, 1993.

[54] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5) :75–174, 2010.

[55] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1) :36–41, 2007.

[56] Santo Fortunato, Vito Latora, and Massimo Marchiori. Method to find community structures based on information centrality. *Physical review E*, 70(5) :056104, 2004.

[57] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[58] A. Frick, A. Ludwig, and H. Mehldau. A fast adaptive layout algorithm for undirected graphs (extended abstract and system demonstration). In *Graph Drawing*, pages 388–403. Springer, 1995.

[59] John Gantz and David Reinsel. Extracting value from chaos. *IDC iView*, pages 1–12, 2011.

[60] F. Gargiulo, M. Lenormand, S. Huet, and O.B. Espinosa. A commuting network model : going to the bulk. *Arxiv preprint arXiv :1102.5647*, 2011.

[61] B. Gaume, F. Venant, and B. Victorri. Hierarchy in lexical organization of natural language,. In D. Pumain, editor, *Hierarchy in natural and social sciences*, volume 3 of *Methodos series*, pages 121–142. Springer, 2006.

[62] Mohammad Ghoniem, Jean-Daniel Fekete, and Philippe Castagliola. On the readability of graphs using node-link and matrix-based representations : a controlled experiment and statistical analysis. *Information Visualization*, 4(2) :114–135, 2005.

[63] M. Girvan and M.E.J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12) :7821, 2002.

[64] Michelle Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy Science USA*, 99 :7821–7826, 2002.

[65] Andrey Goder and Vladimir Filkov. Consensus clustering algorithms : Comparison and refinement. In *ALENEX*, volume 8, pages 109–117, 2008.

[66] B.H. Good, Y.A. De Montjoye, and A. Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4) :46106, 2010.

[67] Roger Guimera, Marta Sales-Pardo, and Luís A Nunes Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2) :025101, 2004.

[68] D. Gusfield. Partition-distance : A problem and class of perfect graphs arising in clustering. *Information Processing Letters*, 82(3) :159–164, 2002.

[69] Stefan Hachul and Michael Jünger. Drawing large graphs with a potential-field-based multilevel algorithm. In *Graph Drawing*, pages 285–295. Springer, 2005.

[70] L. Hamers, Y. Hemeryck, G. Herweyers, M. Janssen, H. Keters, R. Rousseau, and A. Vanhoutte. Similarity measures in scientometric research : the jaccard index versus salton's cosine formula. *Information Processing & Management*, 25(3) :315–318, 1989.

[71] Jing-Dong J Han, Nicolas Bertin, Tong Hao, Debra S Goldberg, Gabriel F Berriz, Lan V Zhang, Denis Dupuy, Albertha JM Walhout, Michael E Cusick, Frederick P Roth, et al. Evidence for dynamically organized modularity in the yeast protein–protein interaction network. *Nature*, 430(6995) :88–93, 2004.

[72] J. Heer and D. Boyd. Vizster : visualizing online social networks. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 32 –39, oct. 2005.

[73] P Jaccard. Bulletin de la société vaudoise des sciences naturelles. *Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines*, 37 :241–272, 1901.

[74] David R Karger. Global min-cuts in rnc, and other ramifications of a simple min-out algorithm. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, pages 21–30. Society for Industrial and Applied Mathematics, 1993.

[75] B. Karrer, E. Levina, and M.E.J. Newman. Robustness of community structure in networks. *Physical Review E*, 77(4) :46119, 2008.

[76] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1) :359–392, 1998.

[77] Jacob Katriel. On a generalized recurrence for bell numbers. *Journal of Integer Sequences*, 11(2) :3, 2008.

[78] Donald E. Knuth. *The Stanford GraphBase : a platform for combinatorial computing*. ACM, New York, NY, USA, 1993.

[79] Corey Kosak, Joe Marks, and Stuart M. Shieber. A parallel genetic algorithm for network-diagram layout. In Richard K. Belew and Lashon B. Booker, editors, *ICGA*, pages 458–465. Morgan Kaufmann, 1991.

[80] Alexander Kraskov, Harald Stögbauer, Ralph G Andrzejak, and Peter Grassberger. Hierarchical clustering using mutual information. *EPL (Europhysics Letters)*, 70(2) :278–284, 2005.

[81] Marek Kubale. *Graph colorings*, volume 352. Amer Mathematical Society, 2004.

[82] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2) :83–97, 1955.

[83] Pascale Kuntz, Dominique Snyers, and Paul J. Layzell. A stochastic heuristic for visualising graph clusters in a bi-dimensional space prior to partitioning. *J. Heuristics*, 5(3) :327–351, 1999.

[84] Antoine Lambert, Romain Bourqui, and David Auber. Winding Roads : Routing edges into bundles. *Computer Graphics Forum*, 29(3) :853–862, 06 2010.

[85] Antoine Lambert, François Queyroi, and Romain Bourqui. Visualizing patterns in Node-link Diagrams. In *Proceedings of the 16th International Conference on Information Visualisation (IV'12)*, pages 48–53, Montpellier, France, 2012.

[86] A. Lancichinetti. Community detection algorithms : a comparative analysis. *Physical Review E*, 80(5) :056117, 2009.

[87] A. Lancichinetti and S. Fortunato. Limits of modularity maximization in community detection. *Arxiv preprint arXiv :1107.1155*, 2011.

[88] A. Lancichinetti and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4) :046110, 2008.

[89] A. Lancichinetti, F. Radicchi, J.J. Ramasco, and S. Fortunato. Finding statistically significant communities in networks. *PloS one*, 6(4) :e18961, 2011.

[90] Thibault Laurent, Nathalie Villa-Vialaneix, et al. Using spatial indexes for labeled network analysis. *Information-Interaction-Intelligence*, 11(1), 2011.

[91] John Aldo Lee, Amaury Lendasse, and Michel Verleysen. Curvilinear distance analysis versus isomap. In *Proceedings of ESANN*, pages 185–192, 2002.

[92] Ian XY Leung, Pan Hui, Pietro Liò, and Jon Crowcroft. Towards real-time community detection in large networks. *Physical Review E*, 79(6) :066107, 2009.

[93] Spiros Mancoridis, Brian S Mitchell, Chris Rorres, Y Chen, and Emden R Gansner. Using automatic clustering to produce high-level system organizations of source code. In *Program Comprehension, 1998. IWPC'98. Proceedings., 6th International Workshop on*, pages 45–52. IEEE, 1998.

[94] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, 2008.

[95] C. McGrath, J. Blythe, and D. Krackhardt. Seeing groups in graph layouts. *Connections*, 19(2) :22–29, 1996.

[96] Ruth M Mickey, Olive Jean Dunn, and Virginia Clark. *Applied statistics : analysis of variance and regression*. Wiley-Interscience, 2004.

[97] Marni Mishna. Attribute grammars and automatic complexity analysis. *Advances in Applied Mathematics*, 30(1-2) :189–207, 2003.

[98] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random Structures & Algorithms*, 6(2-3) :161–180, 1995.

[99] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physics Reviews E*, 69 :066133, 2004.

[100] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences, USA*, 103 :8577–8582, 2006.

[101] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physics Reviews E*, 69(026113), 2004.

[102] A. Noack. Modularity clustering is force-directed layout. *Physical Review E*, 79(2) :026102, 2009.

[103] Andreas Noack. An energy model for visual graph clustering. In *Proceedings of the 11th International Symposium on Graph Drawing*, pages 425–436. Springer, 2004.

[104] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043) :814–818, 2005.

[105] Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. Community detection in social media. *Data Mining and Knowledge Discovery*, 24(3) :515–554, 2012.

[106] R. Patuelli, A. Reggiani, S.P. Gorman, P. Nijkamp, and F.J. Bade. Network analysis of commuting flows : A comparative static approach to german data. *Networks and Spatial Economics*, 7(4) :315–331, 2007.

[107] Richard Perline. Strong, weak and false inverse power laws. *Statistical Science*, pages 68–88, 2005.

[108] G. Pflieger and C. Rozenblat. Discovery and evaluation of graph-based hierarchical conceptual clusters. *Urban Studies (Special Issue : Urban Networks and Network Theory)*, 47(13) :2723–2735, 2010.

[109] Pascal Pons. *Détection de communautés dans les grands graphes de terrain*. PhD thesis, PhD thesis, Univ. Paris 7, 2007.

[110] Pascal Pons and Matthieu Latapy. Post-processing hierarchical community structures : Quality improvements and multi-scale view. *Theoretical Computer Science*, 412(8-10) :892 – 900, 2011.

[111] Daniel Cosmin Porumbel, Jin Kao Hao, and Pascale Kuntz. An efficient algorithm for computing the distance between close partitions. *Discrete Applied Mathematics*, 159(1) :53–59, 2011.

[112] Denise Pumain. *Hierarchy in Natural and Social Sciences*, volume 3 of *Methodos Series*. Springer, 2006.

[113] H Purchase. Which aesthetic has the greatest effect on human understanding ? *Lecture notes in computer science*, pages 248–261, 1997.

[114] H.C. Purchase. Metrics for graph drawing aesthetics. *Journal of Visual Languages & Computing*, 13(5) :501–516, 2002.

[115] François Queyroi. Optimizing a hierarchical community structure of a complex network. *Advances In Knowledge Discovery and Management*, 4, 2012. à paraître.

[116] François Queyroi, Maylis Delest, Jean-Marc Fédou, and Guy Melançon. Assessing the quality of multilevel graph clustering. *Data Mining and Knowledge Discovery*, 2011. à paraître.

[117] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9) :2658–2663, 2004.

[118] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3) :036106, 2007.

[119] Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74(1) :016110, 2006.

[120] Nathalie Henry Riche and Tim Dwyer. Untangling euler diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 16(6) :1090–1099, November 2010.

[121] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[122] G.O. Roberts and J.S. Rosenthal. Markov-chain monte carlo : some practical implications of theoretical results. *Canadian Journal of Statistics*, 26(1) :5–20, 1998.

[123] DF Robinson and Leslie R Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1) :131–147, 1981.

[124] M. Rosvall and C.T. Bergstrom. Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PloS one*, 6(4) :e18209, 2011.

[125] Martin Rosvall, Daniel Axelsson, and Carl T Bergstrom. The map equation. *The European Physical Journal-Special Topics*, 178(1) :13–23, 2009.

[126] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4) :1118–1123, 2008.

[127] J. Rouwendal and P. Nijkamp. Living in two worlds : A review of home-to-work decisions. *Growth and Change*, 35(3) :287–303, 2004.

[128] Loic Royer, Matthias Reimann, Bill Andreopoulos, and Michael Schroeder. Unraveling protein networks with power graph analysis. *PLoS computational biology*, 4(7) :e1000108, 2008.

[129] Franklin E Satterthwaite. An approximate distribution of estimates of variance components. *Biometrics bulletin*, 2(6) :110–114, 1946.

[130] Venu Satuluri and Srinivasan Parthasarathy. Scalable graph clustering using stochastic flows : applications to community discovery. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 737–746. ACM, 2009.

[131] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1 :27–64, 2007.

[132] John Scott. *Social network analysis*. SAGE Publications Limited, 2012.

[133] Claude Elwood Shannon, Warren Weaver, Richard E Blahut, and Bruce Hajek. *The mathematical theory of communication*, volume 117. University of Illinois press Urbana, 1949.

[134] Robin Sibson. Slink : an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1) :30–34, 1973.

[135] H.A. Simon. The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6) :467–482, 1962.

[136] P. Simonetto, D. Auber, and D. Archambault. Fully automatic visualisation of overlapping sets. In *Computer Graphics Forum*, volume 28, pages 967–974. Wiley Online Library, 2009.

[137] J.J. Sylvester. A question in the geometry of situation. *Quarterly Journal of Pure and Applied Mathematics*, 1, 1857.

[138] J.J. Thomas and K.A. Cook. Illuminating the path : The research and development agenda for visual analytics. *IEEE Computer Society*, 2005.

[139] Paul Turán. A note of welcome. *J. Graph Theory*, 1(1) :7–9, 1977.

[140] Stijn van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.

[141] F. van Ham and B. Rogowitz. Perceptual organization in user-generated graph layouts. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6) :1333–1339, 2008.

[142] Bala R. Vatti. A generic solution to polygon clipping. *Communications of the ACM*, 35 :56–63, July 1992.

[143] Allesandro Vespignani. Evolution thinks modular. *Nature*, 35(2) :118–119, 2003.

[144] Ken Wakita and Toshiyuki Tsurumi. Finding community structure in mega-scale social networks :[extended abstract]. In *Proceedings of the 16th international conference on World Wide Web*, pages 1275–1276. ACM, 2007.

[145] Chris Walshaw. A multilevel algorithm for force-directed graph drawing. In *Graph Drawing*, pages 171–182. Springer, 2001.

[146] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301) :236–244, 1963.

[147] C. Ware, H. Purchase, L. Colpoys, and M. McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2) :103–110, 2002.

[148] Stanley Wasserman and Joseph Galaskiewicz. *Advances in social network analysis : Research in the social and behavioral sciences*. Sage, 1994.

[149] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684) :440–442, 1998.

[150] Bernard L Welch. The generalization of student's problem when several different population variances are involved. *Biometrika*, 34(1/2) :28–35, 1947.

[151] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Englewood Cliffs, 2001.

[152] Douglas R White and Frank Harary. The cohesiveness of blocks in social networks : Node connectivity and conditional density. *Sociological Methodology*, 31(1) :305–359, 2001.

[153] Faraz Zaidi, Daniel Archambault, and Guy Melançon. Evaluating the quality of clustering algorithms using cluster path lengths. In *ICDM*, pages 42–56, 2010.