HEC MONTRÉAL

Comparaison d'algorithmes de détection de communautés dynamiques dans les réseaux évolutifs

Par

Anaïs Correc

Sciences de la gestion
(Analytique d'affaires)

Mémoire présenté en vue de l'obtention du grade de maîtrise ès sciences (M.Sc.)

Novembre 2015

Résumé

Un réseau est constitué d'un ensemble d'éléments et de leurs interactions. Il peut être modélisé par un graphe dans lequel une arête entre deux sommets traduit l'existence d'une relation entre deux entités. Lorsqu'il représente un réseau réel, un graphe exhibe certaines propriétés comme un arrangement non aléatoire de ses sommets en groupes aux caractéristiques similaires, dits communautés ou modules : il s'agit de sous-ensembles d'éléments plus densément connectés les uns les autres.

La détection de communautés disjointes est un problème combinatoire difficile. Sa complexité grandit lorsque le recouvrement, soit l'appartenance à de multiples communautés, est considéré, plus encore si chaque relation est assortie d'une information temporelle. Le réseau, dit alors évolutif, n'est plus figé dans le temps, mais s'étend, se contracte, évolue au fil du temps. Les communautés observables sur de plus ou moins longues périodes subissent ainsi des modifications dont la chronologie peut être racontée. C'est autant au problème de la détection des communautés avec et sans recouvrement dans les réseaux évolutifs qu'au suivi de leur évolution que s'intéresse ce document.

Il existe une pléthore d'algorithmes aux fondements théoriques divers abordant ce problème. Une sélection d'entre eux sera confrontée à une base de données anonymes jamais auparavant étudiée et l'évaluation des résultats portera sur des considérations quantitatives et sur la composition interne des communautés. Il s'avère que peu d'algorithmes sont vraisemblablement envisageables ou pratiques ; qu'aucun ne semble supérieur aux autres ; que l'évaluation de la qualité d'une solution reste un enjeu incontournable. Plusieurs failles sont aussi mises à jour, principalement l'influence du choix de la modélisation et de celui d'un algorithme sur la structure communautaire découverte.

Mots clés : *communautés dynamiques, réseau évolutif, réseau social, algorithme de détection, analyse de réseaux sociaux, théorie des réseaux, modularité, comparaison quantitative*

Remerciements

Je souhaite remercier mes directeurs de recherche Sylvain Perron et Gilles Caporossi dont la science, la disponibilité, les critiques et les conseils auront été d'une aide précieuse à la réalisation de ce mémoire.

Mes remerciements à Franck Barès, professeur agrégé à HEC Montréal, pour sa contribution.

Finalement, à Simon et Éloan : je ne saurais assez vous remercier pour votre infinie patience.

Table des matières

Liste des figures

Liste des tableaux

Liste des algorithmes

Liste des acronymes

A^{3}CS	*AdAptive Algorithm for Community Structure in dynamic networks*
AFFECT	*Adaptive Forgetting Factor for Evolutionary Clustering and Tracking*
AFOCS	*Adaptive Finding Overlapping Community Structure*
CNM	algorithme de Clauset, Newman, Moore
COPRA	*Community Overlap PRopagation Algorithm*
CPM	*Clique Percolation Method*
CPMDyn	*Clique Percolation Method Dynamic*
CSS	*ConSenSus clustering*
DBLP	*Digital Bibliography*
DBSCAN	*Density-Based Spatial Clustering of Applications with Noise*
DiDiC	*Distributed Diffusive Clustering*
DYN-LSNNIA	*DYNamic Local Search Nondominated Neighbor Immune Algorithm*
DYN-MOGA	*DYNamic MultiObjective Genetic Algorithms*
DSBM	*Dynamic Stochastic Block Model*
GN	jeu de données synthétique de Girvan et Newman
iLCD	*intrinsic Longitudinal Community Detection*
IM	Information Mutuelle
IMN	Information Mutuelle Normalisée
LFR	jeu de données synthétique de Lancichinetti, Fortunato, Radicchi, (et Ramasco)
LWEP	*Local Weighted-Edge-based Pattern*
MIEN	*Modules Identification in Evolving Networks*
NEO-CDD	*Nash Extremal Optimization for the Dynamic Community Detection problem*
NTF	*Non-negative Tensor Factorization*
OSBM	*Overlapping Stochastic Block Models*

OSLOM	*Order Statistics Local Optimization Method*
PDEC	*Particle-and-Density based Evolutionary Clustering*
RAK	algorithme de Raghavan, Albert et Kumar
SIM	*SIMulated annealing*

Chapitre 1

Introduction

1.1 Contexte

La modélisation sous forme de graphe est couramment utilisée pour comprendre les relations entre des individus dans des réseaux complexes naturels et artificiels tels les réseaux sociaux [1], neuraux [2], biologiques [3], de citation [4], de collaboration [5], d'information tel le *World Wide Web* [6] et d'autres. Ici, le terme individu désigne une *entité* unique du réseau - non pas nécessairement une *personne* puisqu'il peut s'agir d'un neurone du cerveau ou d'une page *Web* - laquelle est représentée par un nœud. Les arêtes liant les individus entre eux expriment une relation. Elle peut être binaire (la relation existe ou non), dirigée (la relation n'est pas réciproque), ou encore pondérée (la relation est plus ou moins forte). L'étude de tels graphes a mené à la découverte de propriétés intéressantes comme les réseaux petits-mondes [1] [7] ; la distribution des degrés des nœuds, c'est-à-dire le nombre de liens incidents, selon une loi de puissance [8] ; ou encore l'existence de groupes d'individus aux caractéristiques similaires et plus densément connectés entre eux qu'avec le reste du graphe, dits communautés ou modules [9]. La connaissance de la structure communautaire d'un réseau a de nombreuses applications comme de contenir une épidémie sur un réseau de téléphonie mobile [10], de suivre la tendance des discussions sur *Twitter* [11] ou de découvrir les thèmes de recherche dans la littérature scientifique [12].

1. *small world*

Ainsi, la détection de communautés dans des graphes captive de nombreux scientifiques depuis les vingt dernières années et à ce sujet Porter et al. [13] et surtout Fortunato et al. [14] ont produit d'excellentes et très complètes revues. S'il semble y avoir un consensus sur les algorithmes les plus efficaces de partition de larges graphes statiques, de nombreuses questions demeurent quant à la découverte de communautés avec recouvrement et plus encore lorsque les graphes sont assortis d'une information temporelle. D'une part, la division d'un réseau en modules disjoints n'est pas toujours représentative de la réalité : un individu peut faire partie de plusieurs groupes distincts (travail, famille, amis, etc.) simultanément. D'autre part, les nouvelles technologies ont permis la compilation de très larges jeux de données horodatées tels les appels entre abonnés d'un service de téléphonie mobile, par exemple. Des propriétés découlant de l'information temporelle sont ignorées si l'on se contente d'agréger les données pour en déduire un graphe statique. Alors, le fait de connaitre l'historique de l'affiliation d'un individu à une certaine communauté pourrait aider à mieux classer le nœud lorsque de nouveaux liens rendent son affiliation ambigüe. Par ailleurs, la classification des sommets dans un contexte dynamique peut révéler l'évolution des communautés (naissance, croissance, contraction, mort, etc.) et permettre une compréhension approfondie du réseau.

De ce fait, la détection de communautés dynamiques avec ou sans recouvrement dans des graphes évolutifs, c'est-à-dire assortis d'une information temporelle, est une question fortement débattue et les publications à son sujet se sont multipliées dans les dernières années. Ce mémoire propose de faire l'exercice de confronter la littérature scientifique à la réalité d'une large base de données horodatées et anonymes n'ayant encore jamais été étudiée.

1.2 Problématique

L'étude des communautés dans un graphe évoluant dans le temps examine deux questions : d'une part, celle d'identifier une bonne répartition des nœuds en groupes partageant certaines propriétés à un instant donné et, d'autre part, celle de savoir comment chacun de ces groupes évolue au fil du temps. La composante temporelle ajoute une couche de complexité à la problématique populaire de la détection de communautés dans les graphes statiques puisqu'il ne s'agit plus simplement de

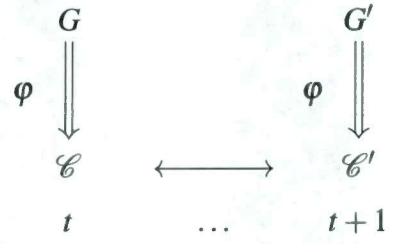


FIGURE 1.1 – Stratégie indépendante de détection de communautés dynamiques dans un graphe temporel. Un certain algorithme φ est appliqué indépendamment aux graphes G et G' afin de trouver leur structure modulaire respective $\mathscr{C}$ et $\mathscr{C}'$. Un problème d'appariement est résolu entre les communautés $\mathscr{C}$ et $\mathscr{C}'$.

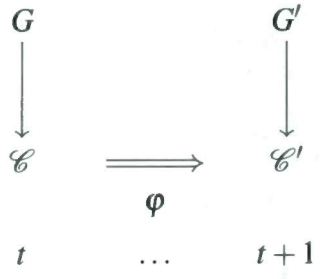


FIGURE 1.2 – Stratégie incrémentale de détection de communautés dynamiques dans un graphe temporel. Un certain algorithme φ est appliqué indépendamment à la structure modulaire $\mathscr{C}$ du graphe G pour trouver la nouvelle structure modulaire $\mathscr{C}'$ en incluant les modifications du graphe G'.

regrouper les nœuds, mais de suivre un groupe entre deux instants.

Le problème peut être abordé selon deux stratégies bien distinctes : la détection indépendante et la détection informée. La première stratégie, dite indépendante, cherche les communautés avec ou sans recouvrement dans le graphe à chaque instant puis résout un problème d'appariement entre les communautés trouvées à des temps successifs. Il pourrait être nécessaire de comparer aussi avec tous les temps antérieurs, dans le cas de communautés à apparitions périodiques. La seconde stratégie, dite incrémentale ou informée, exploite la structure modulaire de l'étape précédente pour en déduire les communautés actuelles. Les deux stratégies sont respectivement illustrées aux figures 1.1 et 1.2.

Il est à noter que l'intérêt de ce mémoire porte principalement sur l'étude des réseaux sociaux comme ceux de collaboration, de téléphonie, etc., lesquels présentent des structures et des propriétés bien différentes des réseaux biologiques ou artifi-

ciels. La modélisation sous forme de graphe dépend alors de la nature des données et de la stratégie privilégiée, soit indépendante ou incrémentale.

1.3 Plan du mémoire

Le second chapitre de ce mémoire propose un cadre formel à la problématique. Des notations mathématiques et définitions sont exposées afin de faciliter la compréhension des concepts et la cohésion du document.

Le troisième chapitre de ce mémoire présente une revue de la littérature sur la détection de communautés dynamiques dans les graphes évolutifs. Notons que les adjectifs *évolutif* et *temporel* sont utilisés pour désigner de façon univoque la qualité des données sur les individus et leurs liens dans le réseau d'être horodatées. Y sont discutés dans cet ordre : les différentes définitions du concept de communauté ; le formalisme de l'évolution des communautés dynamiques ; les mesures de qualité ou de comparaison des modules ; les algorithmes de détection de communautés avec ou sans recouvrement dans des graphes statiques et évolutifs ; les jeux de données synthétiques et réels les plus rencontrés dans la littérature.

Le quatrième chapitre s'attarde à la méthodologie de ce mémoire. Dans un premier temps, nous discutons des propriétés du réseau à l'étude à considérer dans le choix des algorithmes de détection de communautés statiques ou dynamiques. Dans un second temps, les différentes modélisations de ce réseau social sont détaillées en fonction de la stratégie d'analyse privilégiée. Puis, la sélection des méthodes tirées de la revue de la littérature est expliquée.

Le cinquième chapitre présente les résultats des algorithmes de détection de communautés. Il s'ouvre sur une analyse descriptive du jeu de données horodatées ; puis élabore sur chacune des méthodes testées tant en ce qui a trait aux résultats qu'aux obstacles rencontrés ; et conclut avec une analyse comparative.

Enfin, le sixième chapitre clôt ce mémoire avec des suggestions pour de futurs travaux.

Chapitre 2

Notations, définitions et acronymes

Ce chapitre est consacré aux différentes définitions et notations utilisées tout au long de ce document. Il est divisé en deux sections.

La première section présente les notations mathématiques employées pour désigner formellement les graphes statiques ou temporels et les communautés.

La seconde section définit certains concepts permettant d'offrir un cadre rigoureux à la problématique. La littérature sur la détection de communautés dans des graphes manie bon nombre d'idées parfois de façon inconsistante d'un auteur à l'autre : de là, la nécessité de préciser le vocabulaire.

2.1 Notations

Un réseau complexe peut être modélisé à l'aide d'un graphe tel que les *nœuds* représentent des individus et les *arêtes* un certain type d'interaction entre ceux-ci. Les termes $\{nœud, sommet\}$ sont utilisés de façon interchangeable pour désigner strictement le même concept ; de même pour $\{lien, arête\}$ et $\{communauté, cluster, module\}$. Le terme $\{arc\}$ est réservé aux arêtes orientées, c'est-à-dire aux relations non réciproques.

2.1.1 Notations des graphes statiques

Soit

- (u,v) une arête reliant les sommets u et v. Si l'arête est non orientée, alors $(u,v) = (v,u)$;
- $G = G(V,E,W)$ un graphe simple (sans boucles ni liens multiples), pondéré et non orienté (sauf indication contraire) représentant un réseau où V est l'ensemble des nœuds et E celui des arêtes, et où W est une matrice $\mathbb{R}^{|V|\times|V|}$ telle que $W(u,v) = w_{uv}$ est le poids de l'arête (u,v) de E. Dans le cas des graphes non-pondérés, $w_{uv} = 1$, pour chaque lien (u,v) de E. Dans ce cas particulier, le graphe sera noté simplement $G = G(V,E)$;
- $N = |V|$ le cardinal de l'ensemble V, c'est-à-dire le nombre de sommets du graphe ;
- $M = |E|$ le cardinal de l'ensemble E, c'est-à-dire le nombre de liens du graphe ;
- $d(u) = d_u$ le degré du sommet u de V, c'est-à-dire la somme des poids des arêtes non orientées dont une extrémité pointe le sommet u ;
- $d^{in}(u) = d_u^{in}$ le degré entrant du sommet u de V, c'est-à-dire la somme des poids des arcs vers le sommet u ;
- $d^{out}(u) = d_u^{out}$ le degré sortant du sommet u de V, c'est-à-dire la somme des poids des arcs depuis le sommet u ;
- $N(u)$ l'ensemble des voisins du sommet u de V, c'est-à-dire l'ensemble des nœuds $v \in V$ tels que $(u,v) \in E$;
- $A \in \mathbb{R}^{N\times N}$ la matrice d'adjacence telle que

$$A(u,v) = a_{uv} = \begin{cases} 1, & \text{si } (u,v) \in E, \\ 0, & \text{sinon.} \end{cases} \tag{2.1}$$

  Dans le cas d'un graphe non pondéré, $W \equiv A$.

2.1.2 Notations des communautés statiques

Soit

— $\mathscr{C} = \{C_1, C_2, \ldots, C_k\}$ une collection de sous-graphes de V telle que $C_i = G(V_{C_i}, E_{C_i}, W_{C_i}) \in \mathscr{C}$ est une communauté de G. Dans le cas de communautés disjointes, $C_i \cap C_j = \emptyset$ pour tout $C_i \neq C_j$, alors que dans le cas de communautés avec recouvrement l'ensemble $C_i \cap C_j$ peut être non vide pour $C_i \neq C_j$. Dans les deux cas, un sommet peut être non-affilié, c'est-à-dire $u \in G$ mais $u \notin C_i, \forall i$;

— $N_{\mathscr{C}} = |\mathscr{C}|$ le cardinal de l'ensemble $\mathscr{C}$, c'est-à-dire le nombre de communautés du graphe ;

— $n_{C_i} = |V_{C_i}|$ le cardinal de l'ensemble V_{C_i}, c'est-à-dire le nombre de sommets du sous-graphe C_i ;

— $N_{C_i}(u)$ l'ensemble des voisins du sommet u dans C_i, c'est-à-dire l'ensemble des nœuds $v \in V_{C_i}$ tels que $(u,v) \in E_{C_i}$;

— $Com(u)$ un vecteur d'étiquettes du sommet u, c'est-à-dire l'affiliation de u à une ou plusieurs communautés ;

— Y l'ensemble des sommets appartenant à plus d'une communauté

$$Y = \{u \in V : |Com(u)| > 1\};$$

— X l'ensemble des sommets appartenant à une seule communauté

$$X = \{u \in V : |Com(u)| = 1\};$$

— Z l'ensemble des sommets n'appartenant à aucune communauté

$$Z = \{u \in V : |Com(u)| = 0\}, V = X \cup Y \cup Z;$$

— $d_{C_i}^{in}(u)$ le degré interne du nœud u dans la communauté C_i, c'est-à-dire la somme des poids des arêtes (u,v) de E telles que $u, v \in C_i$,

$$d_{C_i}^{in}(u) = \sum_{v \in V_{C_i}} w_{uv};$$

— $d_{C_i}^{out}(u)$ le degré externe du nœud u dans la communauté C_i, c'est-à-dire la

somme des poids des arêtes (u,v) de E telles que $u \in V_{C_i}, v \notin V_{C_i}$,

$$d_{C_i}^{out}(u) = \sum_{v \notin V_{C_i}} w_{uv};$$

— C_i^{in} le degré interne de $C_i \in \mathscr{C}$, c'est-à-dire la somme des poids des arêtes ayant deux extrémités dans la communauté C_i,

$$C_i^{in} = \frac{1}{2} \sum_{u \in V_{C_i}} d_{C_i}^{in}(u); \tag{2.2}$$

— C_i^{out} le degré externe de $C_i \in \mathscr{C}$, c'est-à-dire la somme des poids des arêtes ayant exactement une extrémité dans C_i,

$$C_i^{out} = \sum_{u \in V_{C_i}} d_{C_i}^{out}(u). \tag{2.3}$$

2.1.3 Notations des graphes évolutifs

Soit

— $\mathscr{G}$ un graphe évoluant dans le temps. $\mathscr{G} = (G^{(0)}, G^{(1)}, \ldots, G^{(t_{max})})$ est une séquence de graphes où $G^{(t)} = G(V^{(t)}, E^{(t)}, W^{(t)})$ est l'état du réseau au temps $t \in \{0, \ldots, t_{max}\}$;

— $\Delta V^{(t)}$ les changements apportés à l'ensemble des nœuds du réseau entre les états t et $t+1$, c'est-à-dire les ajouts ou retraits de nœuds ;

— $\Delta E^{(t)}$ les changements apportés à l'ensemble des liens du réseau entre les états t et $t+1$, c'est-à-dire les ajouts ou retraits de liens ;

— $\Delta G^{(t)} = (\Delta V^{(t)}, \Delta E^{(t)}, \Delta W^{(t)})$ les changements apportés au réseau entre les états t et $t+1$ et tels que $G^{(t+1)} = G^{(t)} + \Delta G^{(t)}$;

— $C_i^{(t)}$ la i-ème communauté de l'ensemble des communautés $\mathscr{C}^{(t)}$ détectées dans le graphe $G^{(t)}$.

De façon générale, l'exposant «$^{(t)}$» est utilisé lorsqu'une notation définie aux sous-sections 2.1.1 et 2.1.2 est associée à un temps t.

2.2 Définitions

2.2.1 Méta-communauté

Communauté elle-même formée de plus petites communautés.

2.2.2 Graphe biparti

Un graphe est dit biparti si l'ensemble de ses sommets peut être divisé en deux sous-ensembles disjoints, dits classes, tels qu'il n'existe aucun lien entre les sommets d'une même classe.

2.2.3 Partition, répartition

Soit $\mathscr{C} = \{C_1, \ldots, C_{N_\mathscr{C}}\}$, l'ensemble des communautés d'un graphe G. $\mathscr{C}$ est une partition de G si $C_i \cap C_j = \emptyset$ pour $C_i \neq C_j$. $\mathscr{C}$ est une répartition de G si $C_i \cap C_j \neq \emptyset$ pour $C_i \neq C_j$. Il peut exister un ou des sommets qui ne sont affiliés à aucune communauté.

2.2.4 Instantané

On désigne par *instantané* l'état du réseau au temps t en tant que l'agrégat d'événements survenus entre les temps $t-1$ et t. Ainsi, $G^{(t)} = G(V^{(t)}, E^{(t)}, W^{(t)})$ est l'instantané de $\mathscr{G}$ à t.

2.2.5 Réseau évolutif

Un réseau évolutif [1] est un réseau dont la structure évolue avec le temps. Il peut être modélisé de deux façons distinctes. La première consiste à considérer des instantanés successifs $(G^{(0)}, G^{(1)}, \ldots)$ tels que $G^{(t+1)} = G^{(t)} + \Delta G^{(t)}$, où $\Delta G^{(t)}$ est l'ensemble des modifications du réseau entre les temps discrets t et $t+1$. La seconde consiste à considérer les états successifs $(G^{(0)}, G^{(1)}, \ldots)$ tels que $G^{(i+1)} =$

1. *evolving network*

$G^{(i)} + \Delta G^{(i)}$, où $\Delta G^{(i)}$ est un élément de $\{+u, -u, +uv, -uv\}$ pour la i-ème étape.

2.2.6 Lien *intra*-communauté, lien *inter*-communautés

Une arête connectant deux sommets $u, v \in V$ tels que $u \in V_{C_i}$ et $v \in V_{C_i}$ pour un certain i est dit *intra*-communauté. Une arête connectant deux sommets $u, v \in V$ tels que $u \in V_{C_i}$ et $v \in V_{C_j}$ avec $C_i \neq C_j$ est dit *inter*-communautés.

2.2.7 k-clique, k-clique maximale

Une k-clique U dans un graphe non orienté $G(V, E, W)$ est un sous-ensemble de V de k nœuds tel qu'il existe un lien entre chaque paire de sommets de U. Une k-clique maximale désigne la plus grande k-clique de $G(V, E, W)$ telle qu'il n'existe aucune clique plus grande l'incluant.

2.2.8 Composante connexe

Une composante connexe U dans un graphe $G(V, E, W)$ est un sous-ensemble de V tel qu'il existe au moins un chemin entre chaque paire de sommets de U.

2.2.9 Algorithme des k-moyennes

L'algorithme des k-moyennes[2] dans un graphe partitionne N sommets en k communautés de sorte qu'un individu appartienne à la communauté dont la moyenne est la plus proche par rapport à la distance d'un vecteur normalisé dans l'espace euclidien.

2. *k-means clustering*

Chapitre 3

État de l'art

Ce chapitre offre une revue de la vaste littérature scientifique sur le thème de la détection de communautés dynamiques dans les réseaux évolutifs. Depuis le début du millénaire, le nombre de parutions abordant le thème de la détection quantitative de communautés a explosé ; les approches proposées, aussi diverses que nombreuses, résultent d'efforts interdisciplinaires de physiciens statistiques, mathématiciens, sociologues, etc. Si l'attention a été portée principalement au cadre statique, la détection de communautés dynamiques dans des réseaux évolutifs prend de l'ampleur depuis quelques années seulement. La disponibilité des larges jeux de données horodatées sollicite l'attention de la communauté scientifique et appelle au développement de méthodes rapides et efficaces permettant non seulement de regrouper les données en ensembles avec ou sans recouvrement, mais aussi de suivre l'évolution de tels groupes. Il n'existe à ce jour aucune revue complète telle que celles produites par Fortunato et al. [14] et Porter et al. [13] spécifiquement sur la détection de communautés dynamiques dans les graphes temporels, bien que ces deux articles abordent partiellement le sujet. Plusieurs auteurs ont tenté l'expérience, mais les résultats sont à notre avis peu convaincants. Bilgin et Yener [15] ont publié en 2006 une première revue examinant l'évolution des réseaux en général et le problème de la détection des communautés dynamiques dans un tel cadre en particulier, mais leurs conclusions sont somme toute dépassées depuis. Les revues de Fortunato et al. de 2010, de Aynaud et al. [16] de 2013 ainsi que la section concernant l'état de l'art jusqu'en 2012 de la thèse de Cazabet [17] couvrent certaines méthodes encore pertinentes, mais beaucoup d'autres ont été ajoutées depuis. Plus récemment, Aggarwal et Sub-

bian [18] ont offert une revue de l'évolution des réseaux en général et Hartmann et al. [19] l'ont fait plus précisément sur la détection de communautés dynamiques par les méthodes en ligne [1], soit des méthodes qui n'utilisent que l'information sur le réseau à des temps antérieurs, à l'opposé des méthodes hors-ligne [2] qui peuvent prendre en compte l'information à des temps postérieurs. Enfin, il n'existe aucune étude quantitative comparative outre les tests produits par chaque auteur dans son propre article. Ceux-ci se limitent à quelques méthodes, quelques jeux de données et leurs résultats sont difficilement comparables d'un article à l'autre. Ainsi, nous tenterons de mettre un peu d'ordre dans une littérature abondante où il y a peu de consensus, mais beaucoup de bruit.

La première section étudie les différentes définitions du concept de communauté.

La seconde section aborde des notions générales discutées par les nombreux auteurs dans la littérature que sont l'évolution des communautés et les mesures d'évaluation de la structure modulaire trouvée.

La troisième section décrit l'état de l'art des méthodes de détection de communautés dans les graphes statiques pertinentes et efficaces lorsque la composante dynamique est ajoutée.

La quatrième section expose l'état de l'art en matière de détection de communautés dynamiques dans les graphes évolutifs.

La dernière section détaille la méthodologie d'évaluation des méthodes, en particulier l'utilisation de jeux de données synthétiques et réels.

3.1 Notion de *communauté*

Le *Petit Robert* définit le terme *communauté* comme un *groupe social dont les membres vivent ensemble, ou ont des biens, des intérêts communs.* L'idée importante derrière le concept à l'étude dans cette partie est que certains individus d'un réseau partagent *quelque chose*, qu'ils ont *quelque chose* en commun ; et c'est précisément à déterminer ce *quelque chose* que la détection de communautés s'intéresse. Si nous connaissions déjà, de par leur nature, ce qui pousse certains individus à

1. *online*
2. *offline*

se regrouper, nul besoin y aurait-il de s'attarder à la topologie du réseau. Dans les sous-sections qui suivent, nous formaliserons le concept général de communauté, puis ceux particuliers de *communauté intrinsèque* et de *communauté relative*.

3.1.1 Concept de base

Il n'existe pas de définition rigoureuse de la notion de communauté, cependant l'idée selon laquelle il s'agit d'un ensemble de nœuds plus densément connectés entre eux qu'avec le reste du graphe fait largement consensus. Considérons la densité *intra*-communauté $\delta_{int}(C_i)$ d'un sous-graphe C_i comme le ratio du degré interne de C_i sur le nombre possible d'arêtes *intra*-communauté

$$\delta_{int}(C_i) = \frac{C_i^{in}}{n_{C_i}(n_{C_i}-1)/2},$$

et la densité *inter*-communautés $\delta_{ext}(C_i)$ d'un sous-graphe C_i comme le ratio du degré externe de C_i sur le nombre maximal d'arêtes *inter*-communautés

$$\delta_{ext}(C_i) = \frac{C_i^{ext}}{n_{C_i}(n_{C_i}-1)}.$$

Les densités *intra*-communauté et *inter*-communautés ainsi décrite n'ont de sens que si le graphe est non-orienté.

Une communauté bien définie doit satisfaire les critères parfois contradictoires qui sont d'être à la fois dense et séparée du réseau dans son ensemble. Il s'agit alors d'arriver à un compromis entre une valeur de $\delta_{int}(C_i)$ significativement plus grande que $\delta(G)$ et une valeur de $\delta_{ext}(C_i)$ significativement plus petite que $\delta(G)$, où $\delta(G)$ est le ratio du degré interne de tout le graphe G sur le nombre possible d'arêtes du graphe.

3.1.2 Communautés intrinsèques

Les communautés intrinsèques [17] ou définies localement [14] sont évaluées indépendamment du réseau dans son ensemble : seuls les sous-graphes en question et leurs voisins immédiats sont considérés. Une définition locale simple mais restrictive est celle de clique maximale. Cependant, si les triangles sont fréquemment

observés, les cliques larges sont plutôt rares dans les larges réseaux, ce qui rend une telle définition inapplicable. Une variante moins limitative est de considérer les k-cliques maximales. Il reste que, dans un graphe peu dense, tout algorithme basé sur ces définitions locales risque de produire des solutions inconséquentes. En ce qui concerne la séparation d'un sous-graphe d'avec l'ensemble du réseau, il faut évaluer sa connectivité externe. Ainsi que précisé par Raddichi et al. [20], une communauté C_i est dite *forte* si

$$d_{C_i}^{in}(u) > d_{C_i}^{out}(u), \forall u \in V_{C_i};$$

ou *faible* si

$$\sum_{u \in V_{C_i}} d_{C_i}^{in}(u) > \sum_{u \in V_{C_i}} d_{C_i}^{out}(u).$$

Enfin, il est possible de définir localement une communauté par une mesure de qualité (par exemple, la valeur de $\delta_{int}(C_i)$). Notons que tout sous-ensemble qualifié de communauté intrinsèque le serait peu importe la nature ou la structure du graphe dans lequel il est inclus.

3.1.3 Communautés relatives

Les communautés relatives [17] ou définies globalement [14] sont évaluées en fonction du graphe dans son intégralité. En général, il est supposé qu'un graphe possède une structure communautaire si cette dernière diffère d'un graphe nul [3], c'est-à-dire un graphe présentant une certaine similarité avec le graphe original, mais lequel est produit aléatoirement. Les algorithmes basés sur la définition globale de communautés utilisent ainsi des probabilités pour un sommet d'appartenir à un groupe ou à un autre. La sous-section 3.2.2 discutera de la notion de *modularité* qui a l'avantage d'être dérivée simultanément des définitions locale et globale du concept de communauté.

Le nœud du problème de la partition, ou répartition, d'un graphe est en quelque sorte la définition même du concept de *communauté*. Il semble raisonnable de croire que la définition peut être déduite *a posteriori* selon les caractéristiques propres au réseau, et ce, par la capacité pour un expert d'interpréter la structure modulaire produite par un certain algorithme. Ainsi, il ne saurait y avoir de définition universelle.

3. *null model*

3.2 Généralité sur les communautés dynamiques

Sont formalisées dans cette partie quelques notions récurrentes de la littérature scientifique sur la détection de communautés. La première sous-section décrit les opérations possibles pour les nœuds et arêtes du réseau, de même que les phases du cycle de vie d'une communauté. La seconde sous-section présente les mesures de qualité utilisées dans la comparaison des algorithmes de détection de communautés avec ou sans recouvrement.

3.2.1 Évolution des communautés

Palla et al. [21] sont reconnus comme les premiers à avoir défini rigoureusement les étapes du cycle de vie d'une communauté sur les instantanés d'un graphe évolutif, bien qu'Asur et al. [22] en aient fait de même parallèlement. Ces derniers offrent de plus un cadre rigoureux aux événements affectant les individus.

3.2.1.1 Événements concernant les individus

Apparition : un individu se joint au réseau,

$$u \notin V^{(t-1)} \wedge u \in V^{(t)}.$$

Disparition : un individu quitte le réseau,

$$u \in V^{(t-1)} \wedge u \notin V^{(t)}.$$

Affiliation : un individu se joint à une communauté,

$$u \notin V_{C_i}^{(t-1)} \wedge u \in V_{C_i}^{(t)}.$$

Désaffiliation : un individu quitte sa communauté,

$$u \in V_{C_i}^{(t-1)} \wedge u \notin V_{C_i}^{(t)}.$$

3.2.1.2 Événements concernant les communautés

Continuité : une communauté $C_i^{(t-1)}$ continue au temps t lorsque ses sommets sont inchangés,

$$V_{C_i}^{(t-1)} = V_{C_i}^{(t)},$$

comme illustré à la figure 3.1.

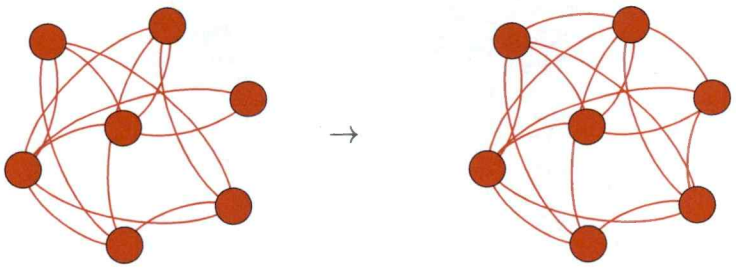


FIGURE 3.1 – Continuité d'une communauté. Les nœuds d'une même couleur appartiennent à une même communauté.

Naissance : une communauté $C_i^{(t)}$ nait au temps t lorsqu'aucun de ses sommets n'étaient préalablement regroupés,

$$\emptyset \rightsquigarrow C_i^{(t)},$$

comme illustré à la figure 3.2.

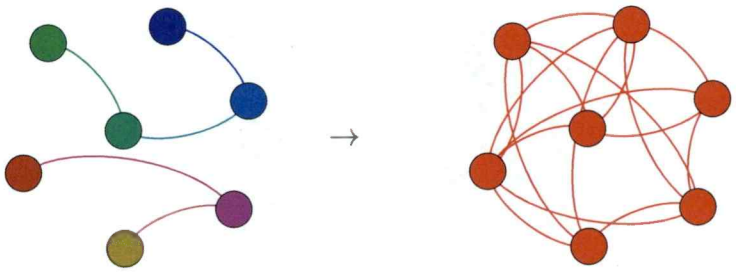


FIGURE 3.2 – Naissance d'une communauté. Les nœuds d'une même couleur appartiennent à une même communauté.

Mort : une communauté $C_i^{(t-1)}$ meurt au temps t lorsqu'aucun des ses sommets

n'est désormais affilié à la même communauté,

$$C_i^{(t)} \rightsquigarrow \emptyset,$$

comme illustré à la figure 3.3.

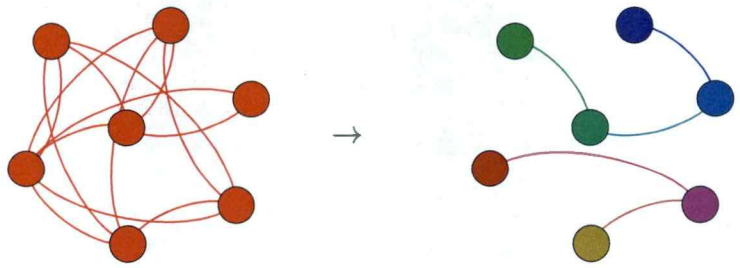


FIGURE 3.3 – Mort d'une communauté. Les nœuds d'une même couleur appartiennent à une même communauté.

Fusion : k communautés fusionnent au temps t lorsque leurs sommets (ou une certaine proportion d'entre eux) forment une seule communauté $C_i^{(t)}$,

$$\bigcup_{j \neq i} C_j^{(t-1)} \rightsquigarrow C_i^{(t)},$$

comme illustré à la figure 3.4.

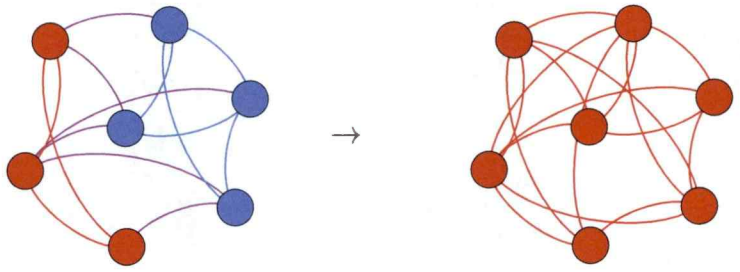


FIGURE 3.4 – Fusion de deux communautés. Les nœuds d'une même couleur appartiennent à une même communauté.

Division : une communauté $C_i^{(t-1)}$ se divise au temps t lorsque ses sommets (ou une

certaine proportion d'entre eux) appartiennent à k communautés,

$$C_i^{(t-1)} \rightsquigarrow \bigcup_{j \neq i} C_j^{(t)},$$

comme illustré à la figure 3.5.

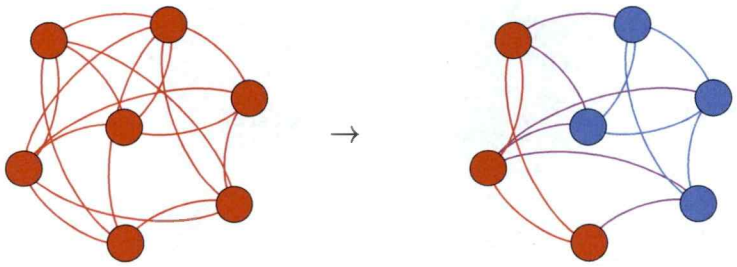


FIGURE 3.5 – Division d'une communauté. Les nœuds d'une même couleur appartiennent à une même communauté.

Croissance : une communauté croît lorsque sa taille augmente,

$$n_{C_i^{(t-1)}} < n_{C_i^{(t)}},$$

comme illustré à la figure 3.6.

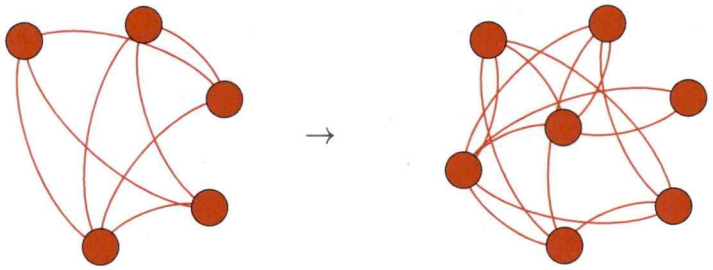


FIGURE 3.6 – Croissance d'une communauté. Les nœuds d'une même couleur appartiennent à une même communauté.

Contraction : une communauté se contracte lorsque sa taille diminue,

$$n_{C_i^{(t-1)}} > n_{C_i^{(t)}},$$

comme illustré à la figure 3.7.

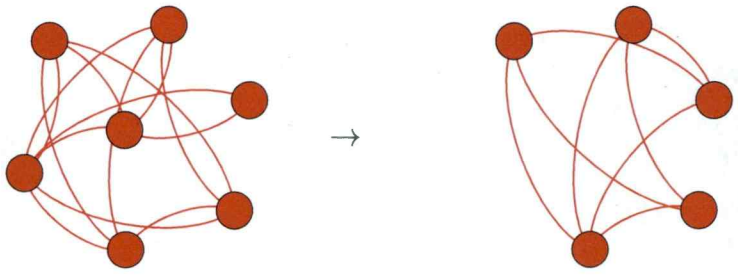


FIGURE 3.7 – Contraction d'une communauté. Les nœuds d'une même couleur appartiennent à une même communauté.

Cazabet [17] propose l'ajout d'une opération pour rendre compte du fait que certaines communautés peuvent apparaitre de façon saisonnière :
Résurgence : une communauté resurgit lorsqu'elle nait et meurt périodiquement,

$$C_i^{(t)} \rightsquigarrow \emptyset \rightsquigarrow C_i^{(t+l)}, l > 1.$$

3.2.2 Mesures d'évaluation des communautés

Il existe de nombreuses mesures de la qualité de la structure communautaire d'un graphe statique ou dynamique. Nous ne présentons que les plus pertinentes pour la suite du document.

3.2.2.1 Mesures de la qualité de la partition statique

La **modularité** de Girvan et Newman [23] est la plus répandue et la plus populaire des mesures de qualité de la partition statique d'un graphe en cluster. Basée sur l'idée qu'un graphe aléatoire ne devrait démontrer de structure modulaire, elle compare la densité de liens d'un sous-graphe à celle obtenue si les sommets du graphe avaient été connectés de façon aléatoire. Elle se mesure de la façon suivante

$$mod(\mathscr{C}) = \frac{1}{2m} \sum_{u,v \in V} \Big[A_{uv} - P_{uv} \Big] \delta(Com(u), Com(v)), \tag{3.1}$$

où A_{uv} est la matrice d'adjacence définie en (2.1), P_{uv} est le nombre attendu d'arêtes entre u et v dans le graphe aléatoire et le delta de Kronecker $\delta(Com(u), Com(v))$ vaut 1 lorsque les sommets u et v appartiennent à la même communauté, 0 sinon,

$|Com(u)| = 1, \forall u \in V$. La distribution naturelle des degrés des sommets dans les graphes réels suggère l'utilisation d'un graphe aléatoire présentant la même probabilité de distribution $P_{uv} = \frac{d_u d_v}{2m}$ et alors

$$mod(\mathscr{C}) = \frac{1}{2m} \sum_{u,v \in V} \left[A_{uv} - \frac{d_u d_v}{2m} \right] \delta(Com(u), Com(v)). \tag{3.2}$$

La valeur de la modularité appartient à l'intervalle $[-1/2, 1]$: plus sa valeur optimale s'approche de 1, plus grande peut être notre croyance en l'existence d'une structure modulaire claire. La valeur de la modularité peut dépendre de la taille du graphe, de sa densité et du nombre de communautés plus ou moins bien définies ; elle ne peut donc servir à comparer des réseaux entre eux [14]. Aussi, un graphe formé d'une seule communauté aura une modularité nulle.

La notion de modularité globale telle que définie ci-dessus peut être généralisée au cas où les arêtes sont pondérées. À ce titre, Newman [24] propose une transformation du graphe pondéré en multi-graphe non pondéré sur lequel (3.2) peut évaluer la justesse d'une certaine structure modulaire. Enfin, il existe des versions de la modularité s'appliquant aux graphes bipartis, soit celles de Barber [25] et de Guimerà et al. [26]. Nous ne retiendrons que la première basée sur la formule (3.1), mais où A et P sont des matrices bloc-diagonales

$$A = \begin{pmatrix} 0_{p\times p} & \tilde{A}_{p\times q} \\ (\tilde{A})^T_{q\times p} & 0_{q\times q} \end{pmatrix}, P = \begin{pmatrix} 0_{p\times p} & \tilde{P}_{p\times q} \\ (\tilde{P})^T_{q\times p} & 0_{q\times q} \end{pmatrix}, \text{ avec } \tilde{P}(u,v) = \tilde{p}_{uv} = \frac{d_u d_v}{m}, \tag{3.3}$$

où p est le nombre de sommets d'une classe et q le nombre de sommets de l'autre classe. Les blocs $0_{p\times p}$ et $0_{q\times q}$ expriment le fait qu'il ne devrait exister d'arêtes entre les sommets d'une même classe dans le graphe aléatoire correspondant.

Notons cependant que l'optimisation de la modularité n'est pas sans failles. Fortunato et Barthelemy [27] ont démontré l'existence d'une limite de résolution. En effet, la définition sous-jacente à la mesure de modularité peut entrainer la détection de communautés regroupées en un ensemble plus grand même si chacune d'entre elles est bien définie en soi. À ce titre, Li et al.[28] proposent une version appelée

densité modulaire

$$D = \sum_{C_i} \frac{\sum_{u,v \in V_i} A_{uv} - \sum_{u \in V_i, v \in V \setminus V_i} A_{uv}}{n_{C_i}}, \tag{3.4}$$

où A est la matrice d'adjacence du graphe G, mais elle reste peu utilisée.

D'autres métriques servent à l'occasion selon le contexte et les algorithmes de détection de communautés telles la **conductance** [29]

$$\phi(C_i) = \frac{C_i^{out}}{\min\{2C_i^{in} + C_i^{out}, 2(m - C_i^{in}) - C_i^{out}\}} \tag{3.5}$$

qui compare la densité interne des liens avec la connectivité externe ; l'**expansion**

$$exp(C_i) = \frac{C_i^{out}}{n_{C_i}} \tag{3.6}$$

qui mesure la moyenne des liens externes ; ou la **coupe normalisée** [4][30]

$$\psi_N(C_i) = \frac{C_i^{out}}{2C_i^{in} + C_i^{out}} + \frac{C_i^{out}}{2(m - C_i^{in}) - C_i^{out}}$$

qui mesure la dissemblance entre les régions C_i et $G \setminus C_i$.

3.2.2.2 Mesures de la qualité de la répartition statique

Il existe plusieurs généralisations de la modularité aux cas de communautés se chevauchant. Supposons un graphe orienté G et un ensemble $\mathscr{C}$ de communautés possiblement non disjointes. Nicosia et al. [31] définissent un vecteur de facteurs d'appartenance $[\alpha_{u,C_1}, \ldots, \alpha_{C_{u,C_{|\mathscr{C}|}}}]$, où $\alpha_{u,C_i} \in [0,1], \forall u \in V, \forall i \in \{1, \ldots, |\mathscr{C}|\}$, indique dans quelle mesure le sommet u appartient à la communauté $C_i \in \mathscr{C}$ avec $\sum_i \alpha_{u,C_i} = 1$. Soit le coefficient d'appartenance $\beta_{(u,v),C_i}$ de l'arc entre les sommets u et v par rapport à la communauté C_i fonction des facteurs d'appartenance $\alpha_{u,C_i}, \alpha_{v,C_i}$, c'est-à-dire $\beta_{(u,v),C_i} = f(\alpha_{u,C_i}, \alpha_{v,C_i})$ (par exemple $f(a,b) = ab$ ou $f(a,b) = \max\{a,b\}$). La

4. *normalized cut*

modularité est alors telle que

$$mod_{d,ov}(\mathscr{C}) = \frac{1}{m}\sum_i \sum_{u,v\in V}\left[\beta_{(u,v),C_i}A_{uv} - \frac{\beta^{out}_{(u,v),C_i}d^{out}_u \beta^{in}_{(u,v),C_i}d^{in}_v}{m}\right], \tag{3.7}$$

où

$$\beta^{out}_{(u,v),C_i} = \frac{\sum_{v\in V}\beta_{(u,v),C_i}}{n},$$
$$\beta^{in}_{(u,v),C_i} = \frac{\sum_{u\in V}\beta_{(u,v),C_i}}{n},$$

et d^{in}_v le degré entrant de v et d^{out}_u le degré sortant de u.

À partir de ces travaux, Chen et al. [32] élaborent une formule pour la modularité dans un graphe pondéré non orienté. En particulier, ils offrent une expression simple pour les facteurs d'appartenance α_{u,C_i},

$$\alpha_{u,C_i} = \frac{d^{in}_{C_i}(u)}{\sum_j d^{in}_{C_j}(u)}.$$

Alors

$$mod_{ov}(\mathscr{C}) = \frac{1}{2m}\sum_i \sum_{u,v\in V}\alpha_{u,C_i}\alpha_{v,C_i}\left[A_{uv} - \frac{d_u d_v}{2m}\right]. \tag{3.8}$$

Plus simplement, Shen et al. [33] proposent

$$mod_o(\mathscr{C}) = \frac{1}{2m}\sum_i \sum_{u,v\in C_i}\frac{1}{O_u O_v}\left[A_{uv} - \frac{d_u d_v}{2m}\right], \tag{3.9}$$

où O_u est le nombre de communautés d'un graphe non orienté auxquelles u appartient. Lorsque les modules sont disjoints, la formule et équivalente à (3.2).

3.2.2.3 Mesures de la qualité de la partition dynamique

Chakrabarti et al. [34] ont défini le premier *cadre évolutif*[5] prenant en compte à la fois la qualité statique, notée Q_{stat}, et la qualité séquentielle, notée Q_{seq}, de la

5. *evolutionary framework*

structure modulaire. L'idée est que la répartition des sommets en communautés dans un graphe évolutif à un instant donné doit adéquatement représenter la structure modulaire inhérente au réseau selon l'information immédiate et, au même moment, ne pas être trop éloignée de celle de la période précédente. Ainsi, Q_{seq} indique à quel point $\mathscr{C}^{(t)}$ diffère de $\mathscr{C}^{(t-1)}$. Le modèle mesure

$$Q = \alpha Q_{stat} + (1-\alpha)Q_{seq}, \tag{3.10}$$

où α est un paramètre dans $[0,1]$ déterminant le poids attribué à l'une ou l'autre des quantités Q_{stat} ou Q_{seq}. Nous reviendrons à la sous-section 3.4.2.1 sur les méthodes proposées par Chakrabarti et al. ainsi que par d'autres à leur suite pour calculer ces mesures de qualités.

3.2.2.4 Mesures comparatives de partitions

Il peut être nécessaire de comparer entre eux des ensembles de communautés afin de déterminer à quel point ils sont similaires ou au contraire différents. Des mesures de similitudes fréquemment utilisées sont l'**Information Mutuelle** (IM) [35, 36] et l'**Information Mutuelle Normalisée** (IMN) [37]. Soit deux partitions $\mathscr{C} = \{C_1, C_2, \ldots\}, \mathscr{D} = \{D_1, D_2, \ldots\}$ d'un graphe G. L'IM se calcule comme suit

$$IM(\mathscr{C},\mathscr{D}) = \sum_{C\in\mathscr{C}}\sum_{D\in\mathscr{D}} P(C,D)\log_2\frac{P(C,D)}{P(C)P(D)}, \tag{3.11}$$

où $P(C) = \frac{|C|}{N}$ et $P(C,D) = \frac{|C\cap D|}{N}$; et l'IMN comme suit

$$IMN(\mathscr{C},\mathscr{D}) = \frac{2IM(\mathscr{C},\mathscr{D})}{\mathscr{H}(\mathscr{C}) + \mathscr{H}(\mathscr{D})}, \tag{3.12}$$

où $\mathscr{H}(\mathscr{C}) = -\sum_{C\in\mathscr{C}} P(C)\log_2 P(C)$ est l'entropie de Shannon [38] de $\mathscr{C}$. Plus grande (respectivement plus proche de 1) est la valeur de l'IM (respectivement de l'IMN) et plus similaires sont les ensembles.

Enfin, d'autres mesures sont, à l'occasion, rencontrées : l'**indice de Rand** [39] déterminant le ratio du nombre de paires de sommets correctement classés dans cha-

cune des partitions $\mathscr{C}$ et $\mathscr{D}$ sur le nombre total de paires

$$R(\mathscr{C},\mathscr{D}) = \frac{M_{11}+M_{00}}{M_{01}+M_{10}+M_{11}+M_{00}}; \tag{3.13}$$

l'**indice de Jaccard** [40] déterminant le ratio du nombre de paires de sommets classés dans la même communauté dans les deux partitions sur le nombre de paires de sommets classés dans la même communauté dans au moins l'une des partitions

$$J(\mathscr{C},\mathscr{D}) = \frac{M_{11}}{M_{01}+M_{10}+M_{11}}; \tag{3.14}$$

où, dans les deux cas,

— M_{00} est le nombre de paires de sommets classés dans différentes communautés dans chacune des deux partitions ;

— M_{01} est le nombre de paires de sommets classés dans différentes communautés dans $\mathscr{C}$ et dans la même communauté dans $\mathscr{D}$;

— M_{10} est le nombre de paires de sommets classés dans la même communauté dans $\mathscr{C}$ et dans différentes communautés dans $\mathscr{D}$;

— M_{11} est le nombre de paires de sommets classés dans la même communauté dans chacune des deux partitions.

3.3 Méthodes de détection de communautés statiques

L'idée de regrouper en communautés des individus d'un réseau est dérivée du problème classique de partitionnement d'un graphe en k ensembles de tailles similaires. Dans l'espoir d'étudier des graphes réels, la limite imposée par l'ignorance du paramètre k a mené Girvan et Newman [9] à poser le problème en termes de détection automatique de communautés sans connaissance préalable du réseau. Il existe un nombre imposant de méthodes de détection de communautés allant de l'optimisation de la modularité par programmation linéaire [41, 42], par recherche à voisinage variable [43], par recuit simulé [44], par optimisation spectrale [45, 46], par optimisation extrémale [47], par algorithme hiérarchique [12], agglomératif [48] ou glouton [49] ; aux marches aléatoires [50, 51] ; à la percolation de cliques (*Clique*

Classe de méthode	Nom	Référence	R	H
Optimisation gloutonne	**CNM**	Clauset et al. [48]		✓
	Louvain	Blondel et al. [49]		✓
Propagation d'étiquettes	**RAK**	Raghavan et al. [53]		
	COPRA	Gregory [54]	✓	
Marche aléatoire	**Infomap**	Rosvall et Bergstrom [51]	✓	
	WalkTrap	Pons et Latapy [50]		✓
Approche probabiliste	**OSLOM**	Lancichinetti et al. [55]	✓	✓
CPM	**CFinder**	Palla et al. [52]	✓	
Modèle génératif	**MOSES**	McDaid et Hurley [56]	✓	
Optimisation par recuit simulé	-	Guimerà et al. [44]		

TABLEAU 3.1 – Méthodes de détection de communautés statiques. La liste n'est en aucun cas exhaustive, seules les méthodes pertinentes à la suite de ce document ont été répertoriées. R : la structure modulaire solution peut être avec recouvrement. H : la structure modulaire solution est en niveaux hierarchiques, i.e. sommets/communautés/méta-communautés/etc.

Percolation Method, noté CPM) [52] ; ou encore à la propagation d'étiquettes [6] [53]. Pour une revue approfondie, quoique non exhaustive, nous référons à Fortunato et al. [14].

Bon nombre de méthodes statiques de détection de communautés sont utilisées sur les instantanés d'un graphe évolutif afin d'en comprendre la structure, de là l'importance de présenter celles pertinentes à ce document. Elles sont répertoriées dans le tableau 3.1. Les cinq sous-sections qui suivent correspondent aux classes de méthodes figurant dans cette table.

3.3.1 Optimisation gloutonne de la modularité

La modularité définie à la sous-section 3.2.2.1 compare la proportion de liens *intra*-communauté d'un graphe avec celle d'un graphe aléatoire. Cette mesure de la qualité d'une partition, voire répartition, est souvent utilisée comme définition même d'une communauté. Ainsi, meilleure est la classification des nœuds d'un graphe en communautés, plus élevée est la valeur de la modularité, d'où l'idée de maximiser la fonction de qualité énoncée en (3.2).

6. *label propagation*

Clauset et al. [48] proposent **CNM** (pour Clauset, Newman et Moore), un algorithme hiérarchique glouton capable de traiter de larges graphes plus rapidement que ses prédécesseurs, en particulier Newman [12] sur lequel il est basé ; et qui va comme suit : initialement, chaque sommet est considéré comme une communauté de singleton, ensuite les modules sont fusionnés deux à deux tels que le gain de modularité est maximal, et ce, jusqu'à ce que le graphe ne forme qu'une seule communauté. La composition hiérarchique du réseau est représentée par un dendrogramme. La sélection de la meilleure coupe du dendrogramme est donnée par la valeur maximale de la modularité *mod*. L'apport de Clauset et al. est de réduire les opérations inutiles et d'améliorer l'efficacité de la méthode par l'utilisation habile de structures de données appropriées.

L'algorithme **Louvain** par Blondel et al. [49] considère le même état initial que **CNM**. À la première itération, chacun des sommets est associé à celui de ses voisins occasionnant le plus grand gain de modularité Δmod, à la condition qu'il existe un Δmod positif. Ces groupes forment dès lors des *super*-nœuds sur lesquels le même procédé est appliqué, à la seule différence que le gain de modularité est calculé localement sur le graphe initial. L'algorithme s'arrête lorsqu'il devient impossible d'augmenter la valeur de *mod*. Tout comme **CNM**, la solution obtenue est en niveaux hiérarchiques sans recouvrement de communautés. La méthode **Louvain** est l'une des plus utilisées surtout pour sa rapidité et sa justesse.

3.3.2 Propagation d'étiquettes

Raghavan et al. [53] proposent une approche très différente. Ici, le réseau est initialisé de telle sorte que chaque nœud se voit assigner sa propre étiquette correspondant à son indice. À chacune des itérations, chaque sommet prend l'étiquette partagée par la majorité de ses voisins immédiats. En cas d'égalité, l'étiquette est choisie au hasard parmi celles des sommets adjacents. L'algorithme **RAK** (pour Raghavan, Albert et Kumara) converge vers une solution telle que tous les sommets partageant la même étiquette appartiennent à une seule communauté. Cette méthode a le désavantage de tendre vers la formation de très larges clusters [57].

Gregory [54] suggère une modification à **RAK** telle que pour chacun des sommets sont conservées, dans un vecteur, les étiquettes de tous ses voisins ainsi qu'un coefficient exprimant la force du lien. Cette modification permet l'appartenance à plus

d'une communauté. Pour contenir l'expansion du vecteur, seules les étiquettes dont le coefficient est supérieur à un seuil sont conservées. **COPRA** (*Community Overlap PRopagation Algorithm*) s'arrête après un nombre arbitraire d'itérations, mais il donne toutefois de bons résultats en pratique selon les auteurs.

3.3.3 Marche aléatoire

Lorsqu'un graphe présente une structure modulaire évidente, une marche aléatoire, soit une série de pas le long des arêtes du graphe de sorte qu'à chaque sommet l'arête empruntée est choisie au hasard parmi les arêtes adjacentes à ce même sommet, passera plus de temps à l'intérieur d'une communauté à cause des nombreuses connexions entre ses nœuds. De ce principe sont dérivées de nombreuses méthodes de détection de modules. Pons et Latapy [50] utilisent dans **WalkTrap** une mesure de distance entre les nœuds basée sur les marches aléatoires telle qu'une courte marche restera à l'intérieur de la communauté dont elle est issue puisqu'il existe peu de liens vers l'extérieur. Cette mesure sert alors à grouper les sommets à l'aide d'un algorithme agglomératif hiérarchique.

Rosvall et Bergstrom [51, 58] utilisent la théorie de l'information pour déduire la structure d'un réseau. Si une communauté est un ensemble d'individus plus fortement connectés les uns aux autres, un flot d'information aléatoire sur un graphe avec une structure modulaire apparente aura tendance à rester coincé à l'intérieur d'une même communauté, alors qu'à l'opposé, il glissera librement dans un graphe aléatoire. Un identifiant est attribué à chacune des structures importantes du graphe (tel qu'il l'est fait pour une ville sur une carte routière, par exemple) ainsi qu'aux sommets qu'elles contiennent. Les noms des sommets sont réutilisés à l'intérieur des structures différentes permettant une compression de l'information plus grande que si chacun des nœuds avait eu un identifiant unique (comme le nom d'une rue qui peut être identique dans deux villes différentes). La détection de communautés dans un graphe se réduit donc au problème d'encodage minimal : soit celui de déduire une partition ou répartition (les deux sont possibles) qui minimise la description d'une marche aléatoire sur un graphe. De toutes les méthodes statiques, **Infomap** est considérée par certains [14, 59] comme la meilleure avec **Louvain** : elle est à la fois rapide et donne des résultats supérieurs sur des graphes synthétiques dont la véritable structure modulaire est connue. Elle possède aussi l'avantage d'être appli-

cable à des graphes pondérés ou orientés.

3.3.4 Approche probabiliste

Lancichinetti et al. [55] proposent **OSLOM** (*Order Statistics Local Optimization Method*), une méthode à multiples paramètres et valeurs seuils applicable sur des graphes pondérés, orientés et aux communautés se chevauchant. L'idée de base est l'optimisation locale d'une fonction d'ajustement exprimant à quel point les clusters sont statistiquement significatifs par rapport aux fluctuations aléatoires.

Dans sa version la plus efficace, **OSLOM** considère une partition obtenue avec **Infomap**, **Louvain** ou **COPRA** comme structure de départ *a priori*. Dans un premier temps, une communauté C_i est prise au hasard et élargie progressivement par ajout ou retrait de nœuds selon la probabilité qu'ils ont d'être affiliés au même groupe par rapport au graphe aléatoire discuté à la sous-section 3.2.2.1. **OSLOM** mesure la probabilité que le nombre de liens *intra*-communauté pour un nœud u soit plus grand que celui entre u et les nœuds de C_i dans le modèle aléatoire correspondant. Dans ce cas, le sommet est considéré statistiquement significatif et ajouté à C_i. De plus, la structure interne de la communauté est validée de telle sorte que les nœuds non significatifs en soient retirés. Dans un second temps, **OSLOM** détermine s'il y a lieu d'unir des communautés entre elles. Ce processus est répété aléatoirement sur chacun des modules et aboutit à une répartition statistiquement significative des sommets. Finalement, un graphe est formé de sorte que les communautés deviennent des *super*-sommets sur lesquels le procédé décrit ci-haut est répété, permettant ainsi de révéler la structure hiérarchique du réseau.

OSLOM possède l'avantage de prendre en compte les nœuds du graphe n'appartenant à aucune communauté, comme il en existe dans les réseaux réels. Elle est considérée comme l'une des méthodes les plus efficaces [17, 59, 60] et ses résultats ont été nombre de fois confirmés.

3.3.5 CPM

La percolation de cliques est basée sur l'idée que les liens plus denses *intra*-communautés sont plus susceptibles de former une clique que les liens *inter*-communautés.

La méthode de Palla et al. [52] implantée dans **CFinder** cherche dans un graphe toutes les k-cliques pour des valeurs de k fixées, puis fusionne toutes les cliques ne différant que d'un nœud. Une communauté de k-cliques (dite k-communauté) est définie comme la composante connexe maximale d'un graphe obtenue par l'union d'une k-clique et de toutes les k-cliques qui lui sont adjacentes. Ainsi, des k-communautés peuvent se chevaucher. L'algorithme de percolation procède à une recherche locale des communautés. La structure modulaire d'un réseau est alors obtenue par itérations sur toutes les k-cliques. Au final, certains sommets appartiennent à plus d'une communauté et d'autres restent non-affiliés.

La définition de communauté inhérente à la percolation de cliques est assez stricte et pourrait mal représenter la réalité de certains réseaux. De plus, un graphe trop dense ou au contraire trop clairsemé serait mal divisé, puisque dans un cas le résultat pourrait être trivialement une seule communauté et dans l'autre des singletons. Enfin, il n'est pas évident quelle valeur de k produit la meilleure répartition ; quoique de façon empirique, Palla et al. situent la valeur adéquate entre 3 et 6.

3.3.6 Autre

MOSES de McDaid et Hurley [56] combine un modèle génératif **OSBM** (*Overlapping Stochastic Block Models*) de Latouche et al. [61] à l'optimisation gloutonne locale d'une fonction d'ajustement pour détecter la structure modulaire avec recouvrement d'un graphe.

Une dernière approche, de Guimerà et al. [44], propose de maximiser la modularité par recuit simulé, une méthode traditionnelle d'optimisation combinatoire de Kirkpatrick et al. [62] inspirée de la science des matériaux. Le recuit simulé permet à un algorithme d'optimisation d'éviter de rester coincé sur un optimum local.

3.4 Méthodes de détection de communautés dynamiques

La détection de communautés dynamiques dans les graphes évolutifs s'intéresse à deux problèmes distincts :

P1 : Quelle est la structure modulaire d'un graphe à l'instant t ? (3.15)

P2 : Comment évolue une communauté dynamique du réseau évolutif sur tout l'horizon temporel ? (3.16)

La littérature scientifique analysée dans ce document s'intéresse parfois à l'une des questions, parfois à l'autre et parfois aux deux. Les méthodes qui y sont proposées sont aussi diverses que nombreuses ; et si certaines modifient des algorithmes de détection statiques pour les rendre applicables au cas de graphes évolutifs, d'autres innovent complètement. Nous faisons part ici de l'état de l'art le plus récent en la matière sans toutefois prétendre à l'exhaustivité, couvrant tout de même plus de 60 articles et quatre revues ([16, 19], partie deux de [17], partie treize de [14]). Mentionnons que ces revues n'ont que très peu de redondances, ce qui laisse croire qu'il faudra encore plusieurs années avant d'arriver à un consensus ignorant le bruit et se ralliant à l'essentiel.

La première sous-section présente les approches indépendantes, lesquelles détectent les communautés avec ou sans recouvrement dans le graphe à chaque instant puis résolvent un problème d'appariement entre les communautés trouvées à des temps successifs.

La seconde sous-section décrit les approches informées assurant une continuité entre les modules détectés sur des instantanés successifs.

La troisième et la quatrième sous-sections concernent les approches incrémentales, soit celles qui considèrent un réseau temporel comme une suite de mises à jour locales de la structure sur des échelles de temps d'une part continue et d'autre part discrète.

La dernière sous-section est dédiée aux méthodes particulières qui modélisent la globalité du réseau sur tout l'horizon temporel en un seul et même graphe pour en déduire une structure communautaire longitudinale.

Un résumé de l'état de l'art est offert dans les tableaux B.1, B.2 et B.3 en annexe.

3.4.1 Approches indépendantes sur des instantanés

3.4.1.1 Approches indépendantes

La première étude de l'évolution des communautés dans un graphe est, de façon consensuelle, attribuée à Hopcroft et al. [63]. Les auteurs considèrent deux instantanés successifs sur lesquels un algorithme agglomératif hiérarchique fusionne les clusters les plus similaires selon la mesure cosinus [7]

$$cos(r_{C_1}, r_{C_2}) = \frac{r_{C_1} \cdot r_{C_2}}{\|r_{C_1}\| \, \|r_{C_2}\|},$$

où r_{C_i} est le vecteur du voisinage associé à C_i, $\|r_{C_i}\|$ sa norme et $\cdot$ le produit scalaire, et retourne leur arbre (dendrogramme) respectif. Un tel algorithme produisant des résultats instables, Hopcroft et al. déterminent les communautés naturelles, c'est-à-dire des communautés consistantes avec la perturbation du réseau entrainée par le retrait aléatoire de 5% des sommets, en mesurant l'appariement de $C_i \in \mathscr{C} = \{C_1, C_2, \ldots\}$ et $D_j \in \mathscr{D} = \{D_1, D_2, \ldots\}$, sur plusieurs itérations pour chaque instantané de la façon suivante

$$match(C_i, D_j) = \min\left(\frac{|C_i \cap D_j|}{|C_i|}, \frac{|C_i \cap D_j|}{|D_j|}\right), \tag{3.17}$$

telles que C_i et D_j vont de pair lorsque la taille de leur intersection est grande. Ils associent les communautés naturelles de chaque instantané de la même façon.

Palla et al. [21] construisent l'union des instantanés t et $t+1$, $G_{t \cup t+1}$, de laquelle l'algorithme **CPM** [52] extrait les k-communautés. Toute communauté de soit $G^{(t)}$, soit $G^{(t+1)}$ apparait une seule fois sur l'union $G_{t \cup t+1}$. Cette propriété des k-communautés permet d'associer les clusters des instantanés successifs. Lorsqu'une communauté de $G^{(t)}$ et une de $G^{(t+1)}$ correspondent à une seule communauté de $G_{t \cup t+1}$ selon une fonction d'auto-corrélation, il s'agit de la même ; lorsque l'union contient plus d'une communauté de chacun des pas de temps, elles sont appariées en ordre décroissant de leur chevauchement relatif.

7. *cosine similarity*

Événement	Notation	Indicateur
Survie	$C_i^{(t)} \rightsquigarrow C_i^{(t+1)}$	$C_i^{(t+1)} = match(C_i^{(t)}) \wedge \forall C_j^{(t)} \neq C_i^{(t)}, C_i^{(t+1)} \neq match(C_j^{(t)})$
Division	$C_i^{(t)} \rightsquigarrow \cup_{l=1}^{k} C_{i_l}^{(t+1)}$	$\forall l, C_{i_l}^{(t+1)} \cap C_i^{(t)}$ est assez large et $\cup_{l=1}^{k} C_{i_l}^{(t+1)} \cap C_i^{(t)}$ est assez large
Fusion	$C_i^{(t)} \subseteq C_i^{(t+1)}$	$C_i^{(t+1)} = match(C_i^{(t)}) \wedge \exists C_j^{(t)} \neq C_i^{(t)}, C_i^{(t+1)} = match(C_j^{(t)})$
Naissance	$\emptyset \rightsquigarrow C_i^{(t+1)}$	$\forall j, C_i^{(t+1)} \neq match(C_j^{(t)})$
Mort	$C_i^{(t)} \rightsquigarrow \emptyset$	dans tous les autres cas

TABLEAU 3.2 – Formalisme des événements externes des clusters de données de Spiliopoulou et al. [65]. $match()$ est une fonction d'appariement entre les communautés, par exemple : $match(C_1) = C_2$ si $V_1 = V_2$.

Contrairement aux auteurs précédents qui se préoccupent de la stabilité des communautés d'un instantané à l'autre, Chen et al. [64] se concentrent sur l'aspect dynamique et proposent une méthode, sans paramètres, de détection et de suivi des communautés, définies strictement comme des cliques maximales. Les auteurs expriment les représentants du graphe $G^{(t)}$ comme les sommets aussi présents dans $G^{(t-1)}$ ou $G^{(t+1)}$; et les représentants de communautés à t comme les nœuds apparaissant dans un minimum de cliques maximales. Il devient alors très simple de suivre ces représentants d'un instantané à l'autre et de détecter les événements de l'évolution des communautés.

3.4.1.2 Cadres méthodologiques de suivi des communautés

Certains auteurs s'intéressent uniquement à décrire les événements du cycle de vie d'une communauté (naissance, croissance, fusion, division, etc.) et proposent des cadres méthodologiques indépendants de la méthode de détection de communautés pour répondre à (3.16).

Dans un cadre méthodologique nommé **MONIC**, Spiliopoulou et al. [65] généralisent une mesure d'appariement $match()$ pouvant être interprétée en termes d'événements de l'évolution des clusters de données ; événements qui sont résumés dans le tableau 3.2.

Wang et al. [66] identifient l'ensemble des communautés en établissant de manière non paramétrée les nœuds centraux [8] et en suivant leur évolution sur chaque instantané. Leur cadre de suivi, **CommTracker**, repose sur l'idée que ces individus sont plus stables que ceux situés aux frontières, lesquels pourraient appartenir à d'autres groupes simultanément. À cet effet, deux nœuds affectés à des communautés distinctes à $t-1$ qui se trouvent regroupés à t traduisent une opération de fusion. Cependant, l'information de la structure modulaire d'un réseau peut difficilement se réduire au comportement de quelques individus.

Greene et al. [67] suggèrent de procéder initialement à la partition d'un graphe avec **MOSES** ou **Louvain** sur chacun des instantanés. Les communautés détectées sur l'instantané le plus récent sont appelées les *front* $\{F_1,\ldots,F_k\}$. **MOSES** résout le problème d'appariement en établissant la similitude d'une communauté à chaque *front* grâce à l'indice de Jaccard,

$$J(C_i,F_j)=\frac{|C_i\cap F_j|}{|C_i\cup F_j|}. \tag{3.18}$$

Si $J(C_i,F_j)>\theta$ pour $\theta\in[0,1]$ fixé, C_i et F_j sont couplés, sinon, un nouveau *front* est ajouté (il représente alors une communauté qui aura disparu quelque part sur l'horizon temporel du réseau).

Goldberg et al. [68] introduisent des axiomes sur lesquels se base leur algorithme de suivi de l'évolution des communautés par construction d'un graphe évolutif multipartite.

Enfin, Takaffoli et al. [69] proposent un cadre résolvant le problème d'appariement des modules avec recouvrement entre les instantanés de telle sorte que deux communautés sont couplées lorsque leur part de membres en commun est au moins k pour cent de la plus grande des deux

$$sim(C_i,C_j)=\begin{cases}\frac{|C_i\cap C_j|}{\max(|C_i,C_j|)}, & \text{si } \frac{|C_i\cap C_j|}{\max(|C_i,C_j|)}\geq k,\\ 0, & \text{sinon.}\end{cases} \tag{3.19}$$

8. *core nodes*

3.4.2 Approches informées sur des instantanés

Ne pourrait-on pas apprendre de la structure modulaire passée pour comprendre l'évolution du réseau ? C'est à cela que s'attaque la stratégie informée de détection de communautés dynamiques dans les graphes évolutifs. En effet, Aynaud et Guillaume [70] ont démontré l'instabilité des algorithmes statiques en comparant les solutions de **WalkTrap, CNM** et **Louvain** lorsque le réseau est légèrement altéré, lesquelles se sont avérées très différentes. La distinction incertaine entre les différences dues à l'évolution réelle et celles dues à l'instabilité rend difficile l'appariement des communautés entre deux instantanés. Ainsi, connaitre l'historique de partition (répartition) d'un réseau évolutif permet une meilleure détection des communautés actuelles lorsque l'affiliation de certains membres est ambigüe.

Tang et al. [71] étudient les réseaux multimodaux, c'est-à-dire que les acteurs y entretiennent des relations de natures différentes traduites par des poids sur les liens, et ajoutent un terme régulateur, témoin de l'évolution du réseau, dans le partitionnement du graphe par k-moyennes en groupes homogènes.

Sun et al. [72] proposent un modèle génératif **Evo-NetClus** de détection des *net-clusters* définis comme des communautés hétérogènes, ou composées d'objets de types variés, dans les graphes multimodaux, capable de décider du nombre naturel de clusters et qui incorpore l'information séquentielle des *net-clusters* des instantanés passés.

Comme la composante aléatoire des algorithmes de détection de communautés rend les résultats instables, Lancichinetti et Fortunato [73] appliquent la partition consensuelle statique, laquelle consiste à comparer les structures modulaires de plusieurs itérations d'un même algorithme dans une matrice de consensus Dc, telle que Dc_{uv} indique le nombre de fois que u et v sont dans la même communauté divisé par le nombre total de communautés. Les valeurs inférieures à un seuil sont mises à zéro. Le même algorithme de partition appliqué à Dc construit Dc', une nouvelle matrice de consensus et ainsi de suite jusqu'à ce que Dc^{final} soit diagonale. Pour un graphe évolutif, Dc est obtenue par décalage d'une fenêtre Δt couvrant plusieurs pas de temps de telle sorte que la partition consensuelle d'un instantané est obtenue sur un sous-ensemble d'instantanés consécutifs. Enfin, l'appariement entre les communautés $\mathscr{C}^{(t)}$ et $\mathscr{C}^{(t+1)}$ est résolu avec l'indice de Jaccard (3.14). Ainsi, la solution retournée par l'algorithme est une partition médiane unique sur le sous-ensemble

d'instantanés, de sorte que l'historique de l'évolution des communautés reste entièrement inconnu.

3.4.2.1 *Evolutionary clustering*

Le cadre évolutif de Chakrabarti et al. [34] énoncé dans la sous-section 3.2.2.3 assure un compromis entre les critères conflictuels que sont la justesse actuelle de la partition (ou répartition) du graphe avec celle historique. La fonction (3.10) rappelée ci-dessous,

$$Q = \alpha Q_{stat} + (1-\alpha)Q_{seq},$$

assure un lissage temporel entre les instantanés. Chakrabarti et al. proposent deux implémentations du cadre évolutif avec d'une part un algorithme agglomératif hiérarchique et de l'autre un algorithme des k-moyennes.

Chi et al. [74] élaborent **EvolSpec**, le premier algorithme de regroupement spectral [9] dans le même cadre évolutif ; c'est-à-dire basé sur la réduction du spectre des valeurs propres de la matrice de similitude définie sur l'ensemble des nœuds. Plus précisément, le regroupement spectral résout le problème de partitionnement par optimisation d'une mesure (coupe normalisée, coupe ratio, coupe Min-Max, etc.). Dans sa version dynamique, la mesure devient le coût statique dans (3.10). Les auteurs décrivent le coût séquentiel selon deux approches : dans la première Q_{seq} mesure à quel point la partition trouvée à t décrit la structure modulaire à $t-1$; dans la seconde, Q_{seq} mesure à quel point la partition actuelle diffère de la précédente.

FacetNet de Lin et al. [75, 76] analyse simultanément la répartition des sommets en communautés avec recouvrement et leur évolution à partir d'un modèle génératif probabiliste, c'est-à-dire qu'il résout simultanément (3.15) et (3.16). Leur cadre évolutif est équivalent à celui de Chakrabarti et al., seulement le problème est exprimé comme celui de la maximisation de l'estimation *a posteriori* de la structure modulaire intégrant les données actuelles du graphe et la distribution *a priori* étant donné la structure modulaire historique. **FacetNet** converge vers une solution locale optimale, mais lentement. Par conséquent, **FacetNet** est inapproprié pour les larges graphes. Par ailleurs, le nombre de communautés doit être connu et, à cet effet, les auteurs suggèrent un nombre de communautés maximisant une version flexible de

9. *spectral clustering*

la modularité de Newman et Girvan adaptée aux communautés se chevauchant.

Kim et Han [77] proposent une méthode de densité-et-particule de détection (notée **PDEC** pour *Particle-and-Density based Evolutionary Clustering*) et de suivi de communautés dans les graphes évolutifs capable de supporter la variation du nombre de modules à chaque pas de temps et de pallier les limites de **FacetNet** que sont la lenteur et l'impossibilité de rendre compte de tous les événements du cycle de vie d'une communauté. Leur méthode de regroupement basée sur la densité [10], dérivée de **DBSCAN** (*Density-Based Spatial Clustering of Applications with Noise*) de Ester et al. [78], modélise le réseau temporel comme un ensemble de particules noté *nano*-communauté et où un cluster est défini comme un sous-ensemble dense de telles particules. Kim et Han incorporent les coûts statiques et séquentiels, selon la distance euclidienne unidimensionnelle entre les sommets, et la mesure de modularité pour partitionner le graphe ; puis maximisent l'IM pour coupler des communautés d'un instantané à l'autre.

Le cadre évolutif de Xu et al [79], légèrement différent de celui de Chakrabarti et al., définit la matrice d'adjacence lissée à l'instant t par la récurrence

$$\begin{cases} \bar{A}^{(0)} = \alpha^{(t)} A^{(0)}, \\ \bar{A}^{(t)} = \alpha^{(t)} \bar{A}^{(t-1)} + (1-\alpha^{(t)}) A^{(t)}, \quad t \geq 1, \end{cases}$$

où $\alpha^{(t)}$ contrôle le poids attribué aux données passées. Au lieu d'optimiser une combinaison linéaire de qualités statiques et séquentielle, ce sont les données mêmes qui sont modifiées pour refléter l'historique du réseau. Ainsi, la décomposition modulaire en k communautés est obtenue avec l'algorithme de regroupement spectral de Yu et Shi [80] avec coupe normalisée sur $\bar{A}^{(t)}$.

Xu et al. [81] élargissent ce dernier cadre, maintenant noté **AFFECT** (*Adaptive Forgetting Factor for Evolutionary Clustering and Tracking*), aux algorithmes respectivement agglomératif hiérarchique et des k-moyennes procédant au regroupement de sommets de graphe évolutif ou à celui de données [11]. Notons qu'aucun de ces deux articles, [79] et [81], ne règle clairement le problème d'appariement entre les communautés identifiées sur des instantanés successifs.

Avec **DYN-MOGA** (*DYNamic MultiObjective Genetic Algorithms*), Folino et Piz-

10. *density-based clustering*
11. *data clustering*

zuti [82] suggèrent la mesure de qualité de la partition en communautés

$$Q = \alpha Q_{stat} + (1-\alpha)Q_{seq} = \alpha mod(\mathscr{C}^{(t)}) + (1-\alpha)IMN(\mathscr{C}^{(t)}, \mathscr{C}^{(t-1)})$$

où α n'est pas prédéterminé. Le problème d'optimisation multiobjectif est résolu par un algorithme génétique. Une version plus rapide, **DYN-LSNNIA** (*DYNamic Local Search Nondominated Neighbor Immune Algorithm*), de Gong et al. [83], le résout cette fois-ci avec un algorithme immunitaire associé à une recherche locale.

Enfin, Chen et al. [84] remplacent la modularité de Newman et Girvan par la densité modulaire D (voir (3.4)) et alors

$$Q = \alpha D(\mathscr{C}^{(t)}) + (1-\alpha)IMN(\mathscr{C}^{(t)}, \mathscr{C}^{(t-1)}).$$

Contrairement aux auteurs précédents, Chen et al. [85] s'intéressent aux réseaux avec chevauchement évoluant rapidement tels les réseaux de communication. Ils proposent une relaxation convexe au problème combinatoire de la détection de communautés par maximisation d'une fonction de qualité instantanée (mesurée par la norme-ℓ) sujette à une contrainte de lissage temporel (mesurée par la distance entre les répartitions à t et $t+1$)

$$\begin{aligned}
&\max_{\{R^{(t)}\}} \sum_{t=1}^{t_{max}} f_{A^{(t)}}(R^{(t)}) \\
&\text{s.l.c.} \sum_{t=1}^{t_{max}-1} \varphi_{A^{(t+1)},A^{(t)}}(R^{(t+1)}, R^{(t)}) \leq \delta \\
&||R(t)||_* \leq B, t = 1, \ldots, t_{max},
\end{aligned}$$

où R est la matrice de la répartition $\mathscr{C}$ telle que $R_{uv} = |\{C \in \mathscr{C} : u \in C, v \in C\}|$ et $||R||_*$ la trace de R ; f la fonction de qualité

$$f_A(R) = -\sum_{u,v} |\omega_{uv}(R_{uv} - A_{uv})|$$

avec $\omega_{uv} = \left|A_{uv} - \frac{d_u d_v}{2M}\right|$; et la distance φ

$$\varphi_{A^{(t+1)},A^{(t)}}(R^{(t+1)}, R^{(t)}) = \sum_{u,v} A_{uv}^{(t+1)} A_{uv}^{(t)} |R_{uv}^{(t+1)} - R_{uv}^{(t)}|.$$

Cette méthode hors-ligne nécessite la fixation de paramètres B et δ, mais les auteurs n'offrent que peu d'information pour ce faire.

3.4.3 Approches incrémentales

Les approches *incrémentales* ou *dynamiques* entrent dans la catégorie des approches informées. Cependant, à la différence de celles décrites à la sous-section 3.4.2, il ne s'agit plus d'appliquer un algorithme statique sur l'ensemble du réseau, mais de mettre à jour localement la structure modulaire.

3.4.3.1 Approches incrémentales sur des instantanés par initialisation de l'algorithme

Une façon de transmettre l'information de la structure communautaire d'un graphe est de fournir à l'algorithme à la période t la structure calculée à la période $t-1$.

L'approche statique de Kim et al. [86] cherche la solution Pareto optimale de deux objectifs conflictuels tels que le premier est maximisé lorsque tous les nœuds sont regroupés, alors que le second est maximisé lorsque tous les nœuds sont séparés. La qualité de la partition du graphe est évaluée avec la *Silhouette* de Rousseeuw [87], une mesure de similitude des éléments d'un même module. De façon triviale, la version dynamique correspond au fait de fournir à l'algorithme la solution de l'instantané précédent comme point de départ.

Aynaud et Guillaume [70] proposent une version dynamique de l'algorithme **Louvain** où l'état initial d'un instantané t n'est plus une partition en singleton, mais celle de l'instantané $t-1$ avec un certain pourcentage de nœuds aléatoirement désaffiliés.

Inspirés par les travaux de Greene et al. [67] et Wang et al. [66], Wang et al.[88] initialisent l'algorithme **Louvain** avec les nœuds centraux de l'étape précédente et établissent la similitude entre les communautés et l'ensemble des *front* (voir sous-section 3.4.1.2).

Dans le but d'étudier les réseaux pair à pair, Gehweiler et Meyerhenke [89] développent **DiDiC** (*Distributed Diffusive Clustering*), un algorithme de distribution diffuse de κ charges dans un réseau optimisant implicitement une mesure de qualité (par exemple la modularité). **DiDiC** est appliqué à une partition initiale aléatoire en

k clusters, puis initialisé avec les modules de l'étape précédente.

Takaffoli et al. [90] suggèrent d'optimiser une modularité locale appelée *L-metric*. L'idée derrière cette mesure est qu'une communauté devrait avoir peu de connections entre ses nœuds frontières et la partie inconnue du graphe, tout en ayant plus de liens avec ses communautés avoisinantes. Ainsi, tout ce qui est à l'extérieur d'une certaine frontière peut être ignoré et la méthode en est d'autant plus rapide. L'algorithme de détection procède comme suit : de la structure modulaire de l'instantané $t-1$ sont extraites les composantes connexes, chacune sert alors de point de départ à la recherche de communautés avec la métrique en question. Les auteurs décrivent aussi **MODEC**, un cadre de suivi des opérations sur les communautés, introduit préalablement dans [91], pour résoudre le problème d'appariement entre les instantanés.

OSLOM, dont il a été question dans la sous-section 3.3.4, est adaptable aux communautés dynamiques, bien que ces auteurs soient restés avares de détails quant à l'application de la méthode aux graphes évolutifs. L'idée est de fournir à **OSLOM** la partition calculée sur l'instantané précédent. Étant donné l'évolution lente des réseaux étudiés, la méthode de Lancichinetti et al. dilate ou contracte les communautés de $t-1$ pour représenter la structure modulaire du réseau à t.

Inspiré par la théorie des jeux, **NEO-CDD** (*Nash Extremal Optimization for the Dynamic Community Detection problem*) de Lung et al. [92] traite la détection de communautés dynamiques comme un jeu où chaque nœud, un joueur, rejoint la communauté qui maximise sa fonction de profit, un «score» communautaire calculé comme la différence entre la qualité de la communauté avec et sans lui. L'équilibre de Nash s'obtient lorsqu'aucun joueur ne peut améliorer son sort grâce à un algorithme d'optimisation extrémale maintenant deux structures, l'une pour la recherche et l'autre avec la meilleure solution. Dans sa version dynamique, la méthode identifie les modifications au réseau par les variations dans les «scores» puis réinitialise l'une des structures, alors qu'elle conserve celle contenant l'information sur la structure modulaire précédente intacte.

3.4.3.2 Approches incrémentales par opération successives

Il ne s'agit plus ici de considérer le réseau temporel comme une suite d'instantanés, mais comme une suite d'ajouts et de retraits de nœuds et d'arêtes du graphe.

DENGRAPH de Falkowski et al. [93], basé sur la densité comme **PDEC** [77], détecte les sous-groupe denses c'est-à-dire à l'intérieur d'un rayon ε selon une mesure de distance spécifique. Les sommets, dits centraux, de ce rayon et leurs voisins se voient attribuer l'étiquette de leur communauté. À l'étape suivante, les distances sont mises à jour localement faisant apparaitre ou disparaitre des nœuds centraux. Un ensemble de règles décrit alors les événements qui surviennent dans la structure communautaire, par exemple : si un nœud est à l'extérieur du rayon ε de ses voisins de même étiquette après la mise à jour des distances, il doit quitter sa communauté et en joindre ou en former une autre. Il existe une version avec recouvrement (Falkowski et al. [94]) et hiérarchique (Schlitter [95]).

Görke et al. [96] proposent de maintenir la structure modulaire au fil du temps en incorporant progressivement les modifications locales tout en assurant une continuité entre les communautés à des instants successifs. Cette extension de l'algorithme statique par coupe minimale d'un arbre de Flake et al. [97] avec garantie d'optimalité modifie partiellement un arbre Gomory-Hu [98] incomplet.

Görke et al. [99] sont les premiers à modifier les algorithmes **Louvain** et **CNM** de telle sorte que les heuristiques gloutonnes mettent à jour localement la solution maximisant la modularité de Newman et Girvan avec **DGlobal** et **TDLocal**. Leurs travaux subséquents [100] incorporent le lissage temporel de Chakrabarti et al. [34], c'est-à-dire la qualité séquentielle, avec l'indice de Rand 3.13 dans la fonction de qualité (3.10) à optimiser localement. Une stratégie de préparation libère un certain nombre de sommets de leur affiliation communautaire $V_{libre}^{(t)}$: ceux-ci peuvent être l'ensemble des nœuds affectés et leurs voisins, ou les parents des nœuds affectés dans le dendrogramme. **Louvain** et **CNM** sont utilisés dans le regroupement avec gain de modularité maximal sur $(\mathscr{C}^{(t)} \backslash V_{libre}^{(t)}) \cup V_{libre}^{(t)}$.

Une autre version dynamique de **Louvain** est l'algorithme adaptatif **QCA** de Nguyen et al. [10]. Pour une certaine partition du graphe à t, **QCA** évalue la modification locale à apporter aux communautés maximisant la modularité selon les quatre cas suivants : ajout de nœud, retrait de nœud, ajout de lien ou retrait de lien. Les auteurs élargissent la méthode aux communautés avec chevauchement en considérant la fonction de densité

$$\tau(C) = \frac{\sigma(C)}{\binom{|C|}{2}}, \text{ où } \sigma(C) = \binom{|C|}{2}^{1-\frac{1}{\binom{|C|}{2}}}$$

comme mesure de la qualité d'une communauté locale C à maximiser. **AFOCS** (*Adaptive Finding Overlapping Community Structure*, Nguyen et al. [101]) découvre en premier lieu toutes les composantes denses du réseau temporel non pondéré et détermine lesquelles définissent une communauté unique. Par la suite, les changements au réseau sont apportés de façon incrémentale selon les quatre cas définis dans **QCA**. Notons que la méthode prend en compte un paramètre déterminant le seuil de chevauchement des communautés.

Ning et al. [102] démontrent que le problème de regroupement spectral peut être accéléré grandement par la mise à jour incrémentale des valeurs propres et de leurs vecteurs propres associés pour incorporer les changements du réseau un à la fois.

L'algorithme de Bansal et al. [103] garde en mémoire le dendrogramme produit par l'algorithme agglomératif hiérarchique **CNM** et le gain de modularité à chaque embranchement. Pour chaque sommet u impliqué dans une modification d'arête ponctuelle du réseau, les étapes de fusion du dendrogramme sont répétées jusqu'à atteindre u, puis **CNM** complète l'arbre par maximisation du gain de modularité.

Duan et al. [104] proposent une version incrémentale de la percolation de cliques de Palla et al. [21] dans laquelle le réseau évolutif est représenté par un flux infini d'ajouts et de retraits de liens. Selon le type de modification, une structure auxiliaire H, notée graphe de clique, est mise à jour localement. H est un graphe où les nœuds sont les cliques maximales de taille au moins k du graphe et tel qu'une arête existe lorsque deux cliques partagent au moins $k-1$ nœuds. Un parcours en profondeur extrait les composantes connexes de H et le résultat est exprimé par une forêt dans laquelle un arbre décrit une communauté.

Shang et al. [105] proposent une approche en deux temps : les communautés sont détectées par **Louvain** sur le graphe à t_0, puis les modifications sont incorporées liens par liens selon leur type (lien *inter*-communautés, lien *intra*-communauté, lien *mi-nouveau* si l'une de ses extrémités est nouvelle, lien *nouveau* si ses deux extrémités sont nouvelles) et la structure est modifiée localement de manière à accroître la modularité ou du moins à la diminuer le moins possible.

Avec **iLCD** (*intrinsic Longitudinal Community Detection*), Cazabet et al. [106, 107, 17] s'attaquent à la détection des communautés avec chevauchement. En résumé, l'algorithme procède comme suit

— Pour chaque ajout d'arête (u,v) *inter*-communautés : v change-t-il d'affilia-

tion pour rejoindre la communauté de u et quelles sont les conséquences de son intégration ?

— Pour chaque retrait d'arête (u,v) *intra*-communauté : des nœuds doivent-ils quitter la communauté ? Ou cette dernière doit-elle se diviser ?

— Pour chaque modification : y a-t-il fusion, division ou mort de communautés affectées par les changements locaux ?

Plusieurs implémentations de l'algorithme sont proposées ([106, 107, 17]), mais nous ne retiendrons que la plus récente et la meilleure selon ses auteurs. Cazabet définit les mesures de **représentativité**, laquelle détermine à quel point un sommet représente sa communauté d'appartenance C

$$Rp(u,C) = \frac{d_C^{int}(u)}{d(u)};$$

de **cohésion intrinsèque**

$$CI(C) = \sum_{u \in C} Rp(u,C);$$

et de **force d'appartenance** de u à C

$$FA(u,C) = \sum_{v \in N_C(u)} Rp(v,C).$$

Ces métriques déterminent le résultat de chaque modification du réseau, à savoir si un nœud doit être intégré à une communauté ou en sortir, ou s'il doit y avoir des fusions, divisions, créations de modules, etc.

3.4.3.3 Approches incrémentales sur des instantanés

Contrairement à la section précédente, concernée par l'arrivée en continue d'information sur les événements singuliers, il s'agit ici plutôt de l'arrivée de données en «lot». Les méthodes décrites plus bas opèrent des modifications locales à la structure modulaire du réseau temporel étant donné les changements apportés entre deux instantanés.

À partir d'une partition du réseau évolutif calculée à $t = 0$ par **CNM**, **MIEN** (*Modules Identification in Evolving Networks*) de Dinh et al. [108] réduit la représentation topologique en un graphe compact où les nœuds sont les communautés et

les arêtes pondérées les sommes des liens *inter*-communautés. À l'instantané suivant, leur méthode isole en singleton les nœuds affectés par les modifications au réseau et **CNM** est appliqué localement avant de réduire à nouveau le graphe à sa représentation compacte.

Exploitant la distribution des degrés en loi de puissance, le cadre méthodologique adaptatif **A^{3}CS** (*AdAptive Algorithm for Community Structure in dynamic networks*) de Dinh et al. [109] identifie rapidement la structure modulaire d'un réseau social. Des étiquettes de *meneur, suiveur* ou *indépendant* sont attribuées aux nœuds de telle sorte que les *suiveurs* soient assignés à la même communauté que leur *meneur* selon la maximisation du gain de modularité et que les *indépendants* soient non-affiliés. À chaque itération, seuls les sommets apparaissant dans $\Delta G^{(t)}$ sont mis à jour.

Avec **LabelRankT**, Xie et al. [110] généralisent aux graphes évolutifs pondérés et orientés le cadre statique de propagation d'étiquettes **LabelRank**, proposé dans [111]. De façon similaire à **COPRA**, **LabelRank** maintient un vecteur d'étiquettes

Algorithme 1 LabelRankT

Entrée : $G^{(0)}, \ldots, G^{(t_{max})}$
pour $t = 1, \ldots, t_{max}$ **faire**
Déterminer l'ensemble $U = \{u_k\}$ des nœuds ayant subi des modifications depuis $t-1$.
si $u_k \notin U$ **alors**
$P_k^{(t)} = P_k^{(t-1)}$
sinon
Initialiser la distribution des étiquettes selon 3.20
Mettre à jour les étiquettes des nœuds de U par *propagation, inflation, mise à jour conditionnelle*.

traduisant la probabilité d'appartenance à une certaine communauté (voir les détails dans 3.3.2). La *propagation*, où un nœud se voit attribuer une probabilité de distribution

$$P_{uv} = \frac{w_{uv}}{\sum\limits_{y \in N(u)} w_{uy}}, \quad \forall v \in C \text{ t.q. } w_{uv} > 0; \tag{3.20}$$

est suivie de l'*inflation*, où un opérateur contracte la propagation ; puis de la *coupure*, où les étiquettes inférieures à un seuil sont ôtées ; et enfin de la *mise à jour conditionnelle* seulement si son étiquette diffère beaucoup de celles de ses voisins.

LabelRankT, la version dynamique de l'algorithme précédent, procède comme décrit dans l'algorithme 1. Mentionnons que Xie et al. font une faute méthodologique importante en comparant le temps de calcul de leur approche à ceux des implantations de **FacetNet** et **iLCD** respectivement en MATLAB® et Java™ : uniformiser les langages de programmation ou offrir une analyse de complexité serait plus juste.

Riedy et Bader [112] abordent le problème des larges jeux de données avec un algorithme parallélisé maintenant la structure communautaire entre les instantanés plutôt que le graphe entier. Cette simplification contribue à réduire grandement le temps de calcul, d'autant plus que la détection de communautés est limitée aux seules modifications et résolue par agglomération incrémentale, c'est-à-dire appliquée aux parties affectées par les modifications. L'algorithme désassemble la structure modulaire de $t-1$ dans le but d'extraire les nœuds dont les liens ont changé à t et agglomère à nouveau par attribution d'un score local, couplage et contraction jusqu'à ce qu'il ne soit plus possible d'améliorer la modularité. Cependant, les auteurs considèrent que les ajouts et retraits d'arêtes ne causent ni fusion, ni division de communautés, ce qui nous semble une hypothèse trop forte pour les cas réels.

La plupart des approches décrites considèrent la topographie d'un graphe comme généralement homogène, c'est-à-dire traduisant des comportements similaires à travers le réseau. Opposés à cette idée, Wang et al. [113] développent une nouvelle représentation appelée *Local Weighted-Edge-based Pattern* (LWEP) définie comme un ensemble de nœuds décrivant une région d'un graphe localement homogène et calculée par des mesures statistiques. Des structures de données auxiliaires, les listes *top-k neighbor* et *top-k candidate*, sont maintenues de façon incrémentale entre les états temporels du réseau, accélérant le calcul en ligne des statistiques servant à la création hors-ligne de LWEPs sur chacun des instantanés. Ces listes conservent respectivement des poids attribués aux comportements cumulés et instantanés des voisins d'un individu selon l'idée qu'une relation qui perdure dans le temps est plus forte qu'une autre ponctuelle. Enfin, l'indice de Jaccard (3.14) est utilisé dans la phase de couplage des communautés trouvées entre les instantanés.

3.4.3.4 Autre

Graphscope de Sun et al. [114] s'attaque aux réseaux bipartis dont l'information arrive en temps réel ; mais, contrairement aux approches par opérations successives

ou sur des instantanés, la structure modulaire est maintenue entre deux *segments de graphe*, c'est-à-dire une succession d'instantanés pour laquelle la structure est inchangée. D'une part, les communautés de nœuds *sources* et *destinations* similaires sont déterminées de sorte à minimiser le coût de l'encodage. Ensuite, si le prochain instantané est semblable, il est ajouté au segment et les partitions sont mises à jour en itérant à partir de leur état précédent. Si le prochain instantané est différent, un nouveau segment est créé.

3.4.4 Modélisation longitudinale du réseau temporel

La catégorie de méthodes abordée dans cette section ne s'intéresse pas nécessairement à la question de la partition à chaque instant, mais plutôt aux communautés cohérentes sur un intervalle. Le problème peut être abordé en considérant chacun des instantanés du réseau comme une couche du graphe où les nœuds récurrents d'un pas de temps à l'autre apparaissent à de multiples occasions. Des *arêtes temporelles* entre les apparitions d'un sommet, différents des liens relationnels entre des individus distincts, sont ajoutés entre les couches successives.

Ben Jdidia et al. [115] sont les premiers à construire ainsi un graphe longitudinal où les liens représentent des relations actives dans un réseau de collaboration. Outre les *arêtes temporelles*, des arêtes lient les co-auteurs, mais aussi les auteurs ayant un co-auteur en commun à un an d'intervalle. Ils déterminent les communautés sur le graphe global avec **WalkTrap**.

Mucha et al. [116] ajoutent des *arêtes temporelles* entre les représentations d'un même individu sur des couches différentes, mais pas nécessairement successives et détectent les communautés sur le graphe longitudinal avec l'algorithme **Louvain** dans lequel la formule de modularité est modifiée pour inclure les strates temporelles.

Tantipathananandh et al. [117] offrent un cadre méthodologique assez restrictif où il est présumé qu'un individu ne devrait pas changer trop souvent d'affiliation et devrait interagir avec sa communauté d'origine la plupart du temps. Le réseau temporel est modélisé globalement par l'ajout d'arêtes entre les apparitions temporelles des sommets et de nœuds colorés représentant les groupes donnés. Les auteurs formulent un problème d'optimisation combinatoire lequel est résolu par une heuris-

tique. La fonction de coût social paramétré comprend trois pénalités : une première pour les changements d'affiliation, une seconde lorsque des sommets du même groupe n'interagissent jamais et une dernière lorsque deux sommets interagissent, mais sont dans des groupes différents. L'algorithme approximatif minimise la fonction sous forme d'utilisation minimale de couleurs sur le graphe longitudinal. Tantipathananandh et Berger-Wolf [118] se débarrassent de la supposition que la partition à chaque instant est connue. Ils résolvent le problème de détection des communautés par une formulation en programmation semi-définie positive suivie d'un processus d'arrondi pour une solution entière.

Aynaud et al. [119] utilisent aussi **Louvain** pour détecter des communautés cohérentes sur tout ou une partie de l'horizon temporel. Leur idée est de calculer une seule décomposition du graphe évolutif en communautés par optimisation de la modularité moyenne sur tout ou une partie de l'horizon temporel ; et bien que la partition ne soit jamais la meilleure sur un instantané précis, elle est, en moyenne, bonne pour décrire la structure modulaire. Aynaud et Guillaume [120] extraient automatiquement un intervalle sur lequel le partitionnement moyen est pertinent.

Avec **DSBM** (*Dynamic Stochastic Block Model*), Yang et al. [121] proposent la modélisation à blocs stochastiques des communautés dans un cadre méthodologique Bayésien. Le nombre de modules est passé en paramètre, ce qui restreint l'utilisation d'une telle approche. Dans sa version en ligne, seules les observations passées sont considérées dans l'estimation des paramètres du modèle génératif, alors que celles futures le sont aussi dans la version hors-ligne.

Mitra et al. [122] arguent que les méthodes de détection de communautés dans les graphes évolutifs de leur prédécesseurs sont longitudinales plutôt que dynamiques. À cet effet, aucune méthode avant eux ne sait gérer la situation où une communauté se renouvelle alors que chacun de ses membres a été remplacé au fil du temps. La modélisation que Mitra et al. proposent ne s'applique cependant qu'à un type de réseau particulier où les interactions entre les individus sont des réponses à un événement antérieur. Le graphe temporel comprend un nœud pour chaque individu pour chacune de ses apparitions. Par exemple : soit A et B tels que à $t = 10$ B réponde à un courriel précédemment envoyé par A à $t = 5$. Un arc lie alors $B^{(10)}$ à $A^{(5)}$. Alors, le graphe temporel correspond à l'historique du réseau et c'est sur cette représentation que l'algorithme **Louvain** détecte les communautés.

Inspirée des techniques de décomposition canonique utilisées sur les graphes statiques du fait de leur capacité à détecter le recouvrement, **NTF** (*Non-negative Tensor Factorization*) de Gauvin et al. [123] procède simultanément à la détection et au suivi de l'évolution de communautés se chevauchant. Une séquence ordonnée de matrices d'adjacences (à $t = 0, 1, 2, \ldots$) représentant le réseau temporel est combinée dans un tenseur cubique non-négativement factorisé. L'algorithme retourne trois matrices dont les premières expriment la structure modulaire et la dernière l'activité temporelle du réseau.

3.5 Évaluation des méthodes de détection de communautés

S'il existe un grand nombre d'algorithmes et de cadres méthodologiques abordant le problème de la détection des communautés dynamiques dans les graphes temporels, comment assurer la qualité de la structure modulaire trouvée ? Nous détaillons ici plus amplement comment certaines des mesures présentées à la sous-section 3.2.2 sont employées à cet effet.

La disponibilité en ligne toujours plus grande de jeux de données dynamiques permet aux chercheurs d'établir la qualité de leurs algorithmes et à en comparer les résultats sur deux fronts : l'efficacité et la précision. Nous en présentons un certain nombre dans la première sous-section.

D'une part, l'indicateur de performance le plus simple est évidemment le temps de calcul, un point intéressant lorsque l'on tente de traiter des larges graphes. Ainsi, certains visent avant tout l'efficacité quitte à faire un compromis sur la qualité ([93, 94, 95, 96, 99, 103, 104, 108, 110, 111, 112] et d'autres) et la plus simple façon d'évaluer un algorithme est de comparer une mesure du temps entre plusieurs méthodes appliquées au même ensemble de données.

D'autre part, valider la précision d'un résultat est beaucoup plus complexe, mais c'est à cela que nous nous intéressons dans la seconde sous-section. Rappelons qu'il n'existe pas de définition rigoureuse d'une communauté et les résultats des divers algorithmes sont acceptés dans la mesure où il est possible d'arriver à un consensus qui semble correspondre à la vérité, mais c'est précisément cette vérité qui pose

Graphe synthétique	Citations
GN	Lin et al. [75, 76], Kim et Han [77], Gong et al. [83], Folino et Pizzuti [82], Lung et al. [92], Yang et al. [121]
LFR	Greene et al. [67], Wang et al.[88], Nguyen et al. [101], Cazabet et al. [106, 107, 17], Dinh et al. [109], Lancichinetti et Fortunato [73]

TABLEAU 3.3 – Occurrences des jeux de données synthétiques dans la littérature scientifique revue.

problème.

3.5.1 Jeux de données synthétiques

S'il existe des graphes de référence synthétiques et statiques entérinés, il en est tout autrement des graphes évolutifs. Et l'absence de tels réseaux est encore, à ce jour, un enjeu majeur de la détection de communautés dynamiques. En particulier, l'étude empirique de réseaux temporels (Leskovec et al. [124], Goldberg et al. [68]) a mis au jour certaines propriétés comme la densification de la distribution en loi de puissance des degrés des nœuds ; l'invariance d'échelle ; la croissance linéaire ou non linéaire ; la durée de vie des communautés comme fonction de leur taille initiale ; etc. Beaucoup de travail reste à faire pour produire un réseau évolutif exhibant toutes les propriétés des graphes réels, tout en présentant une structure modulaire avérée, incorporant fusion, division, mort ou naissance de communautés.

Deux graphes synthétiques sont cependant fréquemment utilisés : **GN** de Girvan et Newman [9], un graphe de 128 nœuds séparés en quatre groupes sans recouvrement et tel que les arêtes *inter*-communautés et *intra*-communauté suivent des distributions données en paramètre, et **LFR** de Lancichinetti et al. [125], un graphe similaire à **GN** de taille $N \in [100, 5000]$ mais tel que la taille des communautés et les degrés des sommets suivent une loi de puissance. Nous référons à [14] pour plus de détails sur la modélisation statique de **GN** et de **LFR**.

La version dynamique de **GN** ou de **LFR** considère dix pas de temps consécutifs, $t = \{0, \ldots, 9\}$. Pour les instantanés $t = 1$ à $t = 9$, seule une quantité déterminée de sommets quittent leur communauté d'origine et sont reconnectés à l'une des autres communautés aléatoirement. Les dix instantanés ont cependant autant de nœuds

et ne peuvent imiter que des réseaux au comportement très régulier, de sorte que d'évaluer la performance d'un algorithme sur un graphe synthétique est plutôt théorique.

Les occurrences dans la littérature scientifique revue des jeux de données **GN** et **LFR** sont compilées dans le tableau 3.3.

3.5.2 Jeux de données réelles

Il existe un grand nombre de bases de données réelles rencontrées lors de cette revue de la littérature et nous ne décrivons, ci-dessous, que les six les plus fréquemment utilisées. À ce titre, l'Université Stanford donne généreusement accès à une collection de bases de données de réseaux sociaux ou de réseaux d'information (facebook, Google+, Amazon, gnutella, reddit, flickr, etc., voir [126]) pour quiconque voudrait tester des algorithmes d'analyse de réseaux.

Les occurrences dans la littérature scientifique revue des six jeux de données sont compilées dans le tableau 3.4.

3.5.2.1 ArXiv

L'archive de publications électroniques **ArXiv** [12] contient les données horodatées de publications scientifiques depuis 1992 et est utilisée comme réseau social de collaboration (ou de citation), où un sommet est un auteur et une arête une relation de collaboration (ou de citation). Des périodes de quelques années constituent les instantanés du graphe.

3.5.2.2 DBLP

Le réseau social de collaboration **DBLP** [13] (*Digital Bibliography & Library Project*), composé d'articles de conférence dans le domaine des sciences informatiques, est modélisé de sorte que les sommets sont les auteurs liés par une arête lorsqu'ils ont participé au même document. Seul le sous-graphe le plus dense de la plus large

12. `http://arxiv.org/`
13. `http://www.informatik.uni-trier.de/ley/db/`

Jeux de données réelles	Citations
ArXiv	Wang et al. [66], Görke et al. [99], Görke et al. [100], Nguyen et al. [10], Chen et al. [85], Dinh et al. [108], Dinh et al. [109]
DBLP	Lin et al. [75, 76], Kim et Han [77], Yang et al. [121], Tang et al. [71], Bansal et al. [103], Takaffoli et al. [69], Sun et al. [72], Goldberg et al. [68], Duan et al. [104], Wang et al. [113]
ENRON	Wang et al. [66], Chen et al. [64], Tang et al. [71], Sun et al. [114], Falkowski et al. [93, 94, 95], Shang et al. [105], Takaffoli et al. [69], Nguyen et al. [10], Duan et al. [104], Takaffoli et al.[90, 91], Dinh et al. [108]
NCAA Football	Kim et Han [77], Gong et al. [83], Folino et Pizzuti [82], Lung et al. [92]
NEC-blog	Chi et al. [74], Lin et al. [75, 76], Yang et al. [121], Ning et al. [102]
VAST	Gong et al. [83], Lung et al. [92]

TABLEAU 3.4 – Occurrences des jeux de données réelles les plus rencontrés dans la littérature scientifique revue.

composante connexe est retenu. Des périodes de quelques années constituent les instantanés du graphe.

3.5.2.3 ENRON

La base de données **ENRON** [14] contient les courriels reçus ou envoyés par l'équipe de direction de la compagnie du même nom. Un nœud représente un individu ou son adresse de courrier électronique et un arc, orienté ou non, entre deux sommets représente un message envoyé entre deux individus. Les instantanés du graphe sont constitués des communications sur une période d'une semaine [104], d'un mois [69] ou encore de trois mois [71], selon les auteurs.

14. `http://snap.stanford.edu/data/`

3.5.2.4 NCAA Football

Le jeu de données **NCAA Football** [15] est le calendrier de parties de football de 2009 de la division collégiale américaine 1-A de la *National Collegiate Athletic Association* (NCAA), où quelques 116 collèges sont répartis en 11 conférences, soit la «véritable» structure communautaire. Le réseau est modélisé statiquement de sorte que chaque équipe soit un sommet, chaque partie soit une arête.

3.5.2.5 NEC-blog

NEC-blog est une collection de billets et de liens colligés par un robot d'indexation sur une période de plus d'un an. Le jeu de données contient 148 681 liens entre 407 blogues et est agrégé mensuellement.

3.5.2.6 VAST

VAST [16], un jeu proposé dans le cadre du *IEEE VAST Challenge* de 2008, comprend les informations horodatées de 9834 appels entre 400 usagers de téléphonie mobile sur une période de 10 jours. Le graphe évolutif équivalent est donc un ensemble de 10 instantanés où les nœuds sont les usagers et les arêtes les appels.

3.5.3 Méthodologies d'évaluation

La valeur d'une approche dynamique est mesurée en comparaison avec 1) une méthode statique, 2) une autre méthode dynamique, 3) la topologie véritable [17] ou empirique.

3.5.3.1 Comparaison avec une méthode statique

Une «bonne» partition (ou répartition) d'un graphe évolutif sur l'horizon temporel $[0, t_{max}]$ doit l'être à chaque instant $t \in [0, t_{max}]$. Ainsi, toute méthode statique de détection de communautés peut être appliquée à $G^{(t)}$ dans l'intention de comparer

15. `http://www.jhowell.net/cf/scores/scoresindex.htm`
16. `http://www.cs.umd.edu/hcil/VASTchallenge08/`
17. *ground truth*

Nom de la méthode	Citations
CNM*	Görke et al. [99, 96], Lancichinetti et al. [55], Tantipathananandh et Berger-Wolf [118], Shang et al. [105], Cazabet et al. [106, 107, 17], Lancichinetti et Fortunato [73] , Dinh et al. [109]
COPRA*	Nguyen et al. [101]
DGlobal, TDLocal	Görke et al. [100]
DYN-LSNNIA	Lung et al. [92]
DYN-MOGA	Gong et al. [83], Chen et al. [84], Lung et al. [92]
EvolSpec	Lin et al. [75, 76], Yang et al. [121], Xu et al. [81]
FacetNet	Kim et Han [77], Folino et Pizzuti [82], Nguyen et al. [101], Yang et al. [121], Dinh et al. [109], Takaffoli et al.[90, 91], Wang et al. [113], Xie et al. [111, 110]
iLCD	Nguyen et al. [101], Xie et al. [111, 110]
Infomap*	Lancichinetti et al. [55], Cazabet et al. [106, 107, 17], Lancichinetti et Fortunato [73] , Xie et al. [111, 110]
Louvain*	Aynaud et Guillaume [70], Dinh et al. [108], Görke et al. [99, 96], Lancichinetti et al. [55], Shang et al. [105], Lancichinetti et Fortunato [73] , Cazabet et al. [106, 107, 17], Dinh et al. [109]
MIEN	Nguyen et al., [10], Nguyen et al. [101], Dinh et al. [109]
OSLOM	Dinh et al., [109], Lancichinetti et Fortunato [73]
PDEC	Folino et Pizzuti [82], Wang et al. [113]
CFinder*	Nguyen et al. [101], Cazabet et al. [106, 107, 17]
QCA	Nguyen et al. [101], Dinh et al. [109]
WalkTrap*	Aynaud et Guillaume [70]

TABLEAU 3.5 – Comparaisons entre les méthodes de détection de communautés dans les graphes évolutifs offertes dans la littérature revue à la section 3.4.
* : Algorithmes statiques.

ses résultats à ceux obtenus par une méthode dynamique. Les algorithmes statiques ayant servi de référence dans la littérature revue pour ce document sont présentés dans le tableau 3.5.

3.5.3.2 Comparaison avec une autre méthode dynamique

Comme il l'a été décrit dans les sections précédentes, les méthodes statiques ont le désavantage d'aboutir à des solutions instables et ne tirent pas avantage de l'historique de l'affiliation des individus à un groupe ou à un autre. Ainsi, on pourrait

confronter les résultats de plusieurs approches dynamiques sur des instantanés, mais encore faut-il qu'elles définissent de façon similaire la notion de communauté. Par ailleurs, la détection de communautés dynamiques dans les réseaux temporels s'intéresse à l'évolution de la structure modulaire. La comparaison de méthodes dynamiques permet alors de confirmer les événements importants des modules comme la naissance, la fusion, la division, etc. Le tableau 3.5 résume les comparaisons observées à la lecture de l'état de l'art.

Notons que certaines méthodes sont difficilement comparables, en particulier celles de la section 3.4.4 où les réseaux étudiés sont modélisés de manière longitudinale.

3.5.3.3 Validation topologique

Il est possible d'inférer une structure modulaire véritable dans un graphe généré : cette idée est d'ailleurs derrière les jeux de données synthétiques. De plus, les communautés de certaines bases de données réelles sont bien connues et validées empiriquement (par exemple le club de karaté de Zachary [127] dans lequel une dispute du président et de l'instructeur a causé un schisme, scindant le club en deux entités ; ou encore l'association des équipes américaines de Football collégial [9] dont le regroupement en conférence est une partition connue). Alors, mesurer la qualité d'une solution se fait aisément avec l'IM ou IMN entre la partition calculée avec un algorithme et celle inhérente au réseau que l'on pourrait qualifier de «vérité». De la même façon, certains jeux de données réelles ont été si abondamment étudiés que l'on semble être parvenu à un consensus sur la structure modulaire sous-jacente. C'est à cette dernière que sont comparés les résultats d'une quelconque approche avec l'indice IM ou IMN. Mentionnons que Gauvin et al. [123] procèdent à l'observation des interactions sociales d'étudiants du secondaire et comparent avec la solution fournie par un algorithme qu'ils ont développé. Il s'agit du seul article, parmi ceux revus, à fournir à la fois une analyse empirique et quantitative.

3.5.3.4 Commentaire

Lee et Cunnningham [128] mettent en garde contre la validation topologique basée sur l'évaluation de résultats à partir des jeux de données sociales comme ceux mentionnés plus tôt. Ils arguent que le fait qu'un algorithme produise de bons ré-

sultats sur ces petites bases de données colligées à la main et étudiées par des experts, comme le club de karaté de Zachary, ne justifie pas son utilisation sur de plus larges bases collectées automatiquement. Lee et Cunnningham montrent que les deux types de bases de données diffèrent de manière si importante que l'efficacité d'une méthode - ils testent entre autres **Infomap** et **Louvain** - sur l'une n'est pas garante de son efficacité sur l'autre. Nous prendrons bien note de leurs réserves lorsque viendra le temps d'analyser de grands jeux de données.

Chapitre 4

Méthodologie

Le problème de la détection de communautés dynamiques dans les réseaux évolutifs est très large et, comme le montre la revue de la littérature du chapitre précédent, les solutions proposées sont tout aussi diverses qu'il existe de variantes au problème ; ainsi, nous n'aspirons aucunement à généraliser l'étude des communautés dynamiques, mais bien à confronter un jeu de données réelles à l'état de l'art en la matière. Ce chapitre circonscrit le travail accompli dans ce mémoire.

La première section décrit et commente les caractéristiques d'un réseau traité à la suite.

La seconde section propose des modélisations d'un jeu de données réelles satisfaisant les contraintes et les caractéristiques imposées selon les multiples stratégies de détection de communautés : détection soit statique, dynamique indépendante ou dynamique informée.

Les algorithmes revus aux sections 3.3 et 3.4 adaptés au problème en question et jugés les plus prompts à fournir de bonnes solutions dans le but d'être comparés dans ce mémoire sont présentés dans la troisième section.

Enfin, la quatrième section détaille la méthodologie proposée pour évaluer les résultats des comparaisons entre les algorithmes retenus.

4.1 Propriétés du réseau évolutif

Dans cette section sont développées les propriétés du jeu de données qui doivent être considérées dans le choix des algorithmes de détection parmi ceux décrits à la section 3.4. Il s'agit là, en quelque sorte, de définir la nature du problème que ce mémoire tente d'aborder.

4.1.1 Réseau social

Chaque type de réseau, qu'il soit biologique, neural, social, d'information, etc., présente une structure propre, que ce soit en rapport à la distribution des degrés des nœuds de son graphe, à l'homogénéité de sa densité à travers toutes les régions ou à l'inverse à son hétérogénéité, etc. Dans l'idée de jeter les bases d'un champ d'intérêt qui pourrait être exploité à notre suite, nous jugeons plus pertinent l'étude d'individus et des relations qu'ils entretiennent. En effet, les applications au marketing ou à la gestion de la classification en groupes aux caractéristiques similaires sont évidentes. On peut penser à la suggestion de produit en fonction des achats des membres d'un groupe, à la gestion organisationnelle, au ciblage de la clientèle ou au développement de campagnes virales. Pour plus de détails sur la théorie des graphes appliquée à l'analyse de réseaux sociaux, voir Newman et al. [129].

4.1.2 Réseau temporel

Il semble évident, de par l'attention que nous avons consacrée aux réseaux évolutifs, que les données de la base étudiée doivent être horodatées. En effet, notre intérêt se porte non seulement aux communautés à tout instant, mais à leur évolution, de leur création à leur destruction, s'il y a lieu.

4.1.3 Réseau hors-ligne

Un jeu de données hors-ligne suppose que toute l'information est disponible avant de modéliser le réseau, c'est-à-dire qu'il s'agit de valeurs historiques connues. Ainsi, il devient facile de déterminer la taille de l'intervalle entre deux instantanés, de calculer des composantes connexes ou d'éliminer de l'information peu pertinente.

4.1.4 Interactions pondérées

Les rapports qu'entretiennent des individus ne sont pas toujours de nature unique. Ainsi, modéliser les liens de façon binaire entre les sommets d'un graphe pourrait masquer une part de l'information sur la force de la relation entre ces individus. Il est alors raisonnable de penser qu'un poids pour chacun des liens peut exprimer la nature de la liaison. Par exemple, ce poids peut représenter la durée ou la répétition de l'interaction. À cet effet, Newman [24] donne l'exemple d'un groupe de macaques rhésus dont la durée des périodes de toilettage entre individus avait été enregistrée. L'auteur démontre que de modéliser ces données dans un graphe non pondéré ne permet pas d'extraire une structure modulaire significative, mais qu'à l'inverse, dans un graphe pondéré, les communautés émergent clairement. Ici, l'alternative - sans pondération - est tout de même considérée par soucis de complétude.

4.1.5 Communautés avec recouvrement

La multiplicité des interactions sociales implique qu'un individu fait rarement exclusivement partie d'un seul groupe. On peut penser à un réseau de collaboration dans lequel un chercheur pourrait tout autant contribuer à des travaux dans des disciplines éloignées et ainsi faire partie de plus d'une communauté scientifique de recherche. Ainsi, il est plus naturel de penser que, s'il existe des communautés dans le réseau temporel, il est possible - mais pas nécessaire - qu'elles se chevauchent.

4.2 Modélisation d'un réseau social temporel

La base de données à laquelle nous avons accès et sur laquelle seront testées les méthodes de détection de communautés statiques ou dynamiques décrit le processus de mise à jour d'un logiciel pour une période déterminée par les employés d'une compagnie dans le domaine des technologies de l'information dont le nom restera confidentiel.

Le jeu contient 715 555 données anonymes de la forme

$$\{\textit{individu, bureau, mise à jour, rôle, date}\}$$

où

- *individu* est un code anonyme attribué à l'un des 1045 employés participant occasionnellement au développement du logiciel ;
- *bureau* est un code anonyme attribué au lieu géographique où se trouve l'employé parmi 46 emplacements plus ou moins éloignés les uns des autres ;
- *mises à jour* est un code numérique attribué à l'une des 280 350 mises à jour du logiciel ;
- *rôle* est une description du rôle de l'individu. Le rôle principal est celui d'*initiateur* de la mise à jour, puis chaque donnée peut décrire l'un des six autres rôles notés $\{rôle\ i\}, i = 1, \ldots 6$ pour des raisons de confidentialité, traduisant le quelconque apport d'un autre employé. Chacune des mises à jour n'implique pas nécessairement la participation d'autres individus et un même individu peut intervenir plus d'une fois dans des rôles différents sur une même mise à jour ;
- *date* est la date à laquelle l'entrée est enregistrée dans la base de données à l'intérieur d'un intervalle de 1974 jours.

Ainsi, il s'agit en quelque sorte d'un échantillon temporel d'un réseau social en personne et en ligne de collaboration où les interactions répétées, temporaires et horodatés entre les individus se font par l'entremise de la participation à une même mise à jour.

Il est à noter qu'en aucun cas les bureaux ne constituent la vérité topologique de la structure modulaire, c'est-à-dire un bureau n'équivaut pas à une communauté. Dans les faits, les employés n'ont aucune obligation, ni même encouragement, à participer aux mises à jour d'employés du même bureau ou autrement géographiquement rapprochés. Il n'existe *a priori* aucune communauté connue et c'est d'ailleurs l'intérêt de cette base de données : soit de déterminer des sous-groupes d'individus qui collaborent plus souvent ensemble qu'avec d'autres sans *a priori* sur l'organisation du réseau ainsi que d'observer leur évolution dans le temps. Contrairement aux travaux revus dans la section sur l'état de l'art, dans lesquels on atteste de la performance des algorithmes développés sur des graphes aléatoires avec structure modulaire implantée ou encore sur des graphes tant étudiés qu'il existe un consensus sur ladite structure, nous ne cherchons pas à reproduire des résultats connus.

Le premier enjeu rencontré et discuté dans cette section est celui de la modélisation

du jeu de données sous forme de graphe. Notons que, faute de connaissance plus profonde quant aux relations exprimées par les données, nous faisons le choix de considérer les rapports simples comme réciproques et de force égale peu importe le rôle des personnes impliquées.

Les sous-sections suivantes présentent les paradigmes de modélisation.

4.2.1 Modélisation en graphe biparti

Dans un graphe biparti, les sommets peuvent appartenir à deux classes distinctes ; dans le cas qui nous intéresse, il s'agit des mises à jour et des individus qui forment chacun une classe de nœuds telle qu'il n'existe de liens entre deux individus ou entre deux mises à jour. Le poids d'une arête est d'au plus un, de sorte que le graphe biparti est non pondéré, soit $W = A$. La modélisation bipartie a l'avantage de conserver l'information sur la nature du lien entre les individus une fois les communautés détectées, c'est-à-dire la possibilité de retrouver au travers de quelles mises à jour certains employés ont collaboré et ainsi faciliter l'analyse et la catégorisation *a posteriori* des communautés détectées.

Exemple 4.2.1. À titre d'illustration, considérons les informations fictives suivantes de la forme $\{$*individu, bureau, mise à jour, rôle, date*$\}$:

- *1, 1, A, initiateur, 12/02/14*
- *2, 2, A, autre, 12/02/14*
- *3, 1, A, autre, 13/02/14*
- *4, 3, A, autre, 14/02/14*
- *3, 1, B, initiateur, 15/02/14*
- *4, 3, B, autre, 15/02/14*

de sorte que l'ensemble des mises à jour est $\mathscr{P} = \{A,B\}, |\mathscr{P}| = n_p = 2$, et celui des individus est $\mathscr{I} = \{1,2,3,4\}, |\mathscr{I}| = n_i = 4$.

Alors, $G = (V,E,W)$, illustré à la figure 4.1, est tel que

$$V = \mathscr{P} \cup \mathscr{I} = \{A,B,1,2,3,4\},$$
$$E = \{(1,A),(2,A),(3,A),(4,A),(B,3),(B,4)\},$$

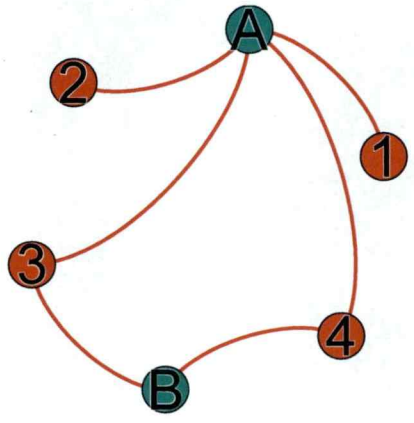


FIGURE 4.1 – Modélisation en graphe biparti. Les sommets A et B représentent des mises à jour et $1,2,3$ et 4, les employés. Une arête indique la participation d'un individu à une mise à jour.

et

$$W \in \mathbb{R}^{(n_i+n_p)\times(n_i+n_p)}, \text{ où } w_{uv} = \begin{cases} 1, & \forall (u,v) \in V, \\ 0, & \text{sinon.} \end{cases}$$

4.2.2 Projection de la modélisation en graphe biparti

Certains algorithmes de détection de communautés sont limités par la taille du graphe ; ainsi, il peut être intéressant de considérer la projection d'un graphe biparti dans laquelle les sommets de l'une des deux classes sont ôtés du graphe. Soit u un sommet d'une classe à retirer - les mises à jour - et v, w des sommets de l'autre classe - les individus - tels qu'il existe un lien entre u et v et entre u et w dans la modélisation bipartie du graphe ; la projection est telle qu'un lien est créé entre v et w, puis u est éliminé. Dans la version pondérée du graphe, si deux individus ont collaboré à plus d'une reprise à la même mise à jour, plusieurs arêtes unitaires existent entre les sommets les représentant. Elles sont alors agrégées en un seul lien dont le poids est la somme des poids des arêtes unitaires.

Exemple 4.2.2. Reprenons les données de l'exemple 4.2.1. Dans le cas de la projection pondérée de la modélisation en graphe biparti, $G = (V,E,W)$, illustré à la

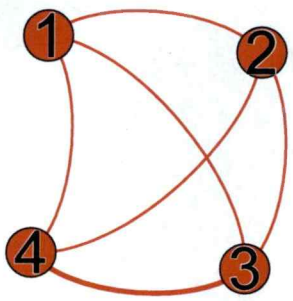


FIGURE 4.2 – Projection de la modélisation en graphe biparti. Les sommets $1,2,3,4$ représentent les employés. Les individus ayant collaboré à une même mise à jour forment autant de cliques qu'il existe de mises à jour : ici, il s'agit des cliques $\{1,2,3,4\}$ et $\{3,4\}$.

figure 4.2, est tel que

$$V = \mathscr{I} = \{1,2,3,4\},$$
$$E = \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\},$$

et

$$W \in \mathbb{R}^{n_i \times n_i}, W = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 2 \\ 1 & 1 & 2 & 0 \end{pmatrix}.$$

4.2.3 Modélisation en graphe étoilé

La modélisation de la sous-section précédente ne fait aucune différence dans les rôles adoptés par les employés dans leur participation à une mise à jour. En particulier, elle accorde la même importance dans la relation entre l'initiateur de la mise à jour et tout autre individu qu'entre individus aux rôles i et j, $i,j \in \{1,\ldots,6\}$. Cependant, rien ne confirme l'existence d'un rapport quelconque entre des rôles accessoires, alors qu'un rapport semble bien réel entre l'auteur de la mise à jour et tout autre individu y ayant participé. Dans la modélisation en graphe étoilé, pour chaque mise à jour, des arêtes unitaires sont affectées uniquement entre le sommet représentant l'initiateur et tout autre sommet représentant un participant. Dans la version

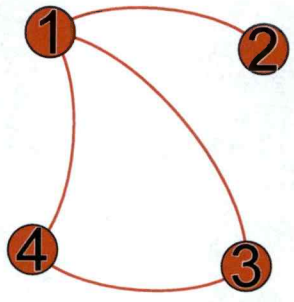


FIGURE 4.3 – Modélisation en graphe étoilé. Les sommets $1,2,3,4$ représentent les employés. Les sommets 1 et 3 sont respectivement les auteurs des mises à jour A et B ; ainsi il existe des arêtes entre 1 et 2, 3 et 4 ; et entre 3 et 4.

pondérée du graphe, les liens incidents aux deux mêmes sommets sont agrégés en une seule arête dont le poids est la somme des poids des arêtes unitaires.

Exemple 4.2.3. Reprenons les données de l'exemple 4.2.1. Dans le cas de la modélisation en graphe étoilé pondéré, $G = (V,E,W)$, illustré à la figure 4.3, est tel que

$$V = \mathscr{I} = \{1,2,3,4\},$$
$$E = \{(1,2),(1,3),(1,4),(3,4)\},$$

et

$$W \in \mathbb{R}^{n_i \times n_i}, W = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

4.2.4 Modélisation temporelle

4.2.4.1 Instantanés de graphe

La création d'un graphe instantané $G^{(t)}$ modélisé selon l'un des trois cas - soit biparti, étoilé ou projeté - ne consiste qu'à partitionner la base de données selon la date de chacune des entrées à l'intérieur d'un intervalle $(t-1,t]$. Ici, le jeu de données horodatées sur 1974 jours est découpé en onze intervalles couvrant des périodes de

182 jours ($\sim$ six mois) chacun. Le choix de la période repose sur la nécessité d'un nombre d'intervalles d'une part suffisamment grand pour comprendre la chronologie des événements sur les communautés et d'autre part suffisamment petit pour ne pas représenter une contrainte calculatoire. Notons que les arêtes et sommets ne sont pas cumulés d'un intervalle de temps à l'autre et qu'il s'agit bien d'instantanés disjoints. Dans le cas contraire, des nœuds pourraient survivre sur tout l'horizon temporel alors qu'ils n'apparaissent que sporadiquement, faussant ainsi l'interprétation des communautés détectées. De plus, la partition finale serait dans les faits équivalente à la partition sur $\mathscr{G} = \{G^{(0)}, \ldots, G^{(10)}\}$ et beaucoup d'information sur l'évolution des communautés serait alors diluée dans une solution intertemporelle.

4.2.4.2 Évolution en continu

Nombre d'algorithmes de détection de communautés dynamiques dans les graphes évolutifs sont construits à partir des modifications incrémentales dans le réseau que sont l'addition ou le retrait de nœuds ou d'arêtes. Cette perspective pose la question de la permanence de la relation une fois l'acte de collaboration accompli. Seule une connaissance plus approfondie de la réalité du fonctionnement du réseau permettrait de déterminer cette durée avec certitude. Ainsi, il revient aux experts ou aux acteurs capables d'interpréter les données de définir d le nombre de périodes (jours, mois, années, etc.) qu'une relation entre des individus persiste si elle n'est jamais ravivée ; c'est-à-dire que si une arête (u,v) entre deux sommets est ajoutée au temps t_1 pour représenter la collaboration entre deux individus, l'arête est retirée à $t_1 + d$ à moins d'une nouvelle collaboration entre u et v au temps $t_2 < t_1 + d$ auquel cas l'arête sera retirée à $t_2 + d$. Cette approche suppose la création d'un sommet - s'il n'est pas déjà dans le graphe - dès qu'il est rencontré dans la base de données et sa destruction lorsque son degré devient nul. Le réseau ainsi constitué peut être considéré soit en continu, soit en instantanés $G^{(t)}$, c'est-à-dire partitionné de sorte que ne sont retenues que les opérations d'ajout et de retraits d'arêtes et de nœud sur l'intervalle $(t-1,t]$ pour $t \in (0,t_{max}]$ et $G^{(0)} = \emptyset$.

Exemple 4.2.4. Reprenons les données de l'exemple 4.2.1. Supposons le graphe biparti $\mathscr{G}$ évoluant de manière continue et $d = 2$ le nombre de jours pendant lesquels

la relation perdure. Ainsi,

$$
\begin{array}{rll}
12/02/14, t=0: & +A & \\
 & +1, & +(A,1) \\
 & +2, & +(A,2) \\
13/02/14, t=1: & +3, & +(A,3)
\end{array}
$$

$$
\begin{array}{rll}
14/02/14, t=2: & +4, & +(A,4) \\
 & -(A,1), & -1 \\
 & -(A,2), & -2 \\
15/02/14, t=3: & +B, & +(B,3) \\
 & & +(B,4) \\
 & & -(A,3)
\end{array}
$$

La modélisation temporelle des données peut se faire de façon similaire dans le cas du graphe projeté ou étoilé.

4.3 Sélection d'algorithmes de détection de communautés

Comprendre, tester, comparer et analyser tous les algorithmes de détection de communautés statiques ou dynamiques dans les réseaux évolutifs rapportés dans l'état de l'art constituerait une tâche impossible à réaliser dans le cadre de ce mémoire : nous offrons dans cette section une sélection de méthodes populaires s'appuyant sur des fondements théoriques divers et choisis parce qu'en mesure de s'appliquer à la base de données disponible. Bien que nous soyons intéressée aux méthodes s'attaquant aux graphes temporels hors-ligne pondérés et non pondérés, non orientés où les communautés présentent une part de recouvrement ; ces critères s'avèrent plutôt restrictifs, en particulier le dernier, de sorte qu'il ne resterait, au final, que très peu d'algorithmes envisageables. Nous retenons donc des algorithmes de partition stricte et d'autres plus complets, tous décrits ci-dessous.

— **A^{3}CS**, Dinh et al. [109], voir 3.4.3.3 : approche incrémentale de détection de communautés avec recouvrement dans les réseaux sociaux par maximisation

de la modularité sur des instantanés dans lequel seuls les sommets affectés par les modifications de $\Delta G^{(t)}$ sont réévalués et réassignés s'il-y-a-lieu ;

— **AFOCS**, Nguyen et al. [101], voir 3.4.3.2 : approche incrémentale par opérations successives, de sorte qu'une fois la structure modulaire initiale avec recouvrement calculée, seules les parties du graphe affectées par des modification locales (ajout/retrait de sommets ou d'arêtes) sont réévaluées. **AFOCS** est disponible publiquement à l'adresse `http://pages.towson.edu/npnguyen/publications.html` ;

— **CFinder**, Palla et al. [52], voir 3.3.5 : approche statique avec recouvrement basée sur une définition stricte de communautés comme l'union des k-cliques à leurs k-cliques adjacentes. **CFinder** est disponible publiquement dans un progiciel à l'adresse `http://www.cfinder.org/` ;

— **COPRA**, Gregory [54], voir 3.3.2 : approche statique de détection de communautés avec recouvrement dans la classe des cadres d'apprentissages semi-supervisés par propagation d'étiquettes maintenues et ordonnées dans un vecteur ;

— **CPMDyn** [1], Palla et al. [21], voir 3.4.1.1 : approche indépendante sur des instantanés dans laquelle les communautés sont détectées avec **CFinder** sur l'union des instantanés t et $t+1$ et comparées à celles détectées indépendamment ;

— **CSS**[1], Lancichinetti et Fortunato [73], voir 3.4.2 : approche d'une part statique, d'autre part informée sur des instantanés permettant d'appliquer n'importe lequel des algorithmes statiques de partition. **CSS** recherche un consensus dans la multiplication des itérations ; en particulier, dans sa version dynamique, il s'agit du consensus entre les partitions $t-1$ et t, puis t et $t+1$, etc. Une implémentation avec **RAK**, **SIM** ou **Louvain** est disponible publiquement à l'adresse `https://sites.google.com/site/andrealancichinetti/software` ;

— **FacetNet**, Lin et al. [75, 76], voir 3.4.2.1 : cadre évolutif avec modèle généré plutôt lent très souvent utilisé comme base de comparaison aux autres méthodes (voir le tableau 3.5). Une implantation pour MATLAB® est disponible à l'adresse `http://www.yurulin.com/download/code/facetnet.html` ;

1. Nom donné par nous pour simplifier la lecture.

— **iLCD**, Cazabet et al. [106, 107, 17], voir 3.4.3.2 : approche incrémentale de détection de communautés avec recouvrement par opérations successives et règles de décision - concernant le statut du sommet affecté par une modification - basé sur une définition locale de la notion de communautés. **iLCD** est disponible publiquement à l'adresse `http://cazabetremy.fr/iLCD.html` ;

— **Infomap**, Rosvall et Bergstrom [51], voir 3.3.3 : approche statique de détection avec ou sans recouvrement dans la classe des méthodes de marches aléatoires avec encodage minimal de l'information. Une implantation statique et une application dynamique (sans recouvrement) avec appariement entre les instantanés et représentation en diagramme alluvial sont toutes deux disponibles publiquement dans un progiciel de Edler et Rosvall, *The MapEquation*, disponible publiquement à l'adresse `http://www.mapequation.org` ;

— **LabelRankT**, Xie et al. [111, 110], voir 3.4.3.3 : algorithme incrémental sur des instantanés basé sur la propagation d'étiquettes ;

— **Louvain**, Blondel et al. [49], voir 3.3.1 ; approche statique rapide et efficace par optimisation gloutonne de la modularité de Newman et Girvan. Une version est disponible publiquement à l'adresse `https://sites.google.com/site/findcommunities/` ;

— **LouvainDyn** [2], Aynaud et Guillaume [70], voir 3.4.3.1 : approche incrémentale par initialisation de **Louvain** avec la partition de l'instantané précédent ;

— **OSLOM**, Lancichinetti et al. [55], voir 3.3.4 et 3.4.3.1 : cadre méthodologique avec réinitialisation qui semble prometteur en particulier par le fait qu'il permette l'utilisation de **Louvain**, **Infomap** ou **COPRA** - indépendamment ou en combinaisons - à l'intérieur de son processus de détection des communautés. **OSLOM** est disponible publiquement à l'adresse `http://www.oslom.org/software.htm` ;

— **PDEC**, Kim et Han [77], voir 3.4.2.1 : cadre évolutif où le regroupement est basé sur la densité et le lissage temporel se fait par maximisation de l'IM entre les partitions successives ;

— **RAK**, Raghavan et al. [53], voir 3.3.2 : approche statique de partition de graphe par propagation d'étiquettes uniques. **RAK** est disponible publique-

2. Nom donné par nous pour simplifier la lecture.

ment dans un prologiciel à l'adresse
`https://sites.google.com/site/andrealancichinetti/software` ;

— **SIM**^{2}, Guimerà et al. [44], voir 3.3.6 : approche statique de détection de communautés qui maximise la modularité par recuit simulé. **SIM** est disponible publiquement à l'adresse
`https://sites.google.com/site/andrealancichinetti/software`.

Le tableau 4.1 résume les méthodes sélectionnées ci-dessus. Outre le calcul de la structure modulaire, faut-il encore appairer les communautés entre chacun des intervalles de temps lorsque la méthode ne répond pas à cette question, comme c'est le cas des algorithmes statiques appliqués indépendamment à chacun des instantanés. Dans cette perspective, nous tentons d'exploiter les cadres méthodologiques suivants :

— **CommTracker**, Wang et al. [66], voir 3.4.1.1 : cadre de suivi de l'évolution de nœuds centraux prenant en compte l'évolution drastique des réseaux sociaux où seule une petite proportion des sommets existe de manière stable sur tout l'horizon temporel ;

— Takaffoli et al. [69], voir 3.4.1.2 : cadre de suivi de l'évolution établissant la similitude entre les communautés aux pas de temps successifs ou non lorsqu'elles ont au moins k pour cent des sommets de la plus grande des deux en commun (3.19) (page 33).

La mesure populaire de couplage qu'est l'indice de Jaccard (3.14) (page 24) fait aussi partie des cadres d'appariement considérés. Les détails de l'algorithme d'appariement des modules sont reportés en annexe, en A.

4.4 Évaluation des résultats

Se rencontre ici une énorme difficulté dans l'évaluation des solutions des différents algorithmes de détection de communautés statiques - sur tout l'horizon temporel - ou dynamiques - sur des instantanés ou en continu : selon quels critères devrons-nous baser notre comparaison ? En effet, la base de données dont nous disposons n'ayant jamais encore été étudiée, nous ignorons la structure topologique

Classe	Nom	Référence	Pondéré	Recouvrement
Approche probabiliste	**OSLOM**	Lancichinetti et al. [55]	✓	✓
Consensus	**CSS**◇	Lancichinetti et Fortunato [73]	✓	
CPM	**CFinder***	Palla et al. [52]	✓	✓
	CPMDyn◇	Palla et al. [21]	✓	✓
Marche aléatoire	**Infomap***	Rosvall et Bergstrom [51]	✓	✓
Modèle généré	**FacetNet**	Lin et al. [75, 76]	✓	✓
Optimisation de la modularité	**A^{3}CS**	Dinh et al. [109]		✓
	Louvain*	Blondel et al. [49]	✓	
	LouvainDyn◇	Aynaud et Guillaume [70]	✓	
	SIM*◇	Guimerà et al. [44]	✓	
Particule-et-densité	**PDEC**	Kim et Han [77]		✓
Programmation dynamique	**iLCD**	Cazabet et al. [106, 107, 17]		✓
Propagation d'étiquettes	**LabelRankT**	Xie et al. [111, 110]	✓	✓
	RAK*	Raghavan et al. [53]	✓	
	COPRA*	Gregory [54]	✓	✓
Autre	**AFOCS**	Nguyen et al. [101]		✓

TABLEAU 4.1 – Résumé des méthodes de détection de communautés dans un graphe évolutif évaluées. * Algorithme uniquement statique. ◇ Nom donné par nous pour simplifier la lecture.

sous-jacente voire même s'il existe une telle structure. Toute métrique entre la «véritable» partition (répartition) et la partition (répartition) calculée, telle que l'IM ou IMN, devient alors inutile ; à moins de considérer l'une des solutions comme «vérité» topologique. Cette idée n'apparait cependant pas rigoureuse.

4.4.1 Modularité

L'une des mesures de qualité dont il a souvent été fait mention dans les sections précédentes est la modularité de Girvan et Newman, équation (3.2), et ses variantes traitant les graphes bipartis de Barber (3.3) ou le chevauchement des communautés (3.9). Une valeur positive de la modularité atteste de l'existence de communautés - la partition n'est pas aléatoire - et plus sa valeur s'approche de un, meilleure est la solution *selon la définition de communauté implicite à cette mesure*. Ainsi, s'il est possible d'affirmer, en calculant sa modularité, qu'une solution produite par un quelconque algorithme atteste une structure modulaire, il en est tout autre de discuter de sa qualité outre lorsque l'algorithme est lui-même basé sur l'optimisation de cette mesure comme le sont **Louvain**, **SIM**, **OSLOM**[3], **CSS**[3] et d'autres. Par exemple, une moindre valeur de la modularité de la solution fournie par **Infomap** par rapport à celle de **Louvain** ne signifie pas *nécessairement* une moindre qualité. Néanmoins, nous examinerons ce que le calcul de cette mesure peut nous apprendre des solutions des différents algorithmes.

4.4.2 Conductance

Autre mesure, la conductance (3.5) indique à quel point une communauté C est bien détachée du reste du réseau : $\phi(C) = 0$ signifie qu'elle n'a aucun lien externe ; $\phi(C) = 1$, une infinité. C'est d'ailleurs la mesure de qualité qu'ont privilégiée Leskovec et al.[130] dans l'étude systématique d'une centaine de réseaux statiques. Comme l'ont fait Šubelj et Bajec [131], Gargi et al. [132] et d'autres, nous évaluons plutôt la conductance moyenne

$$\bar{\phi}(\mathscr{C}) = \frac{1}{N_{\mathscr{C}}} \sum_i \phi(C_i) \tag{4.1}$$

3. Selon les algorithmes passés en paramètres.

des solutions comme mesure de comparaison entre les partitions calculées. Notons qu'aucun des algorithmes évalués ne repose sur l'optimisation de la conductance.

4.4.3 Autre

Remarquons aussi que nous ne nous intéressons pas uniquement aux solutions, mais aussi au processus y ayant mené. Ainsi, l'évaluation et la comparaison des algorithmes portent non seulement sur le résultat, mais sur la manipulation et les difficultés de ce fait rencontrées.

Finalement, une partition ou répartition du graphe est supposée adéquate dans la mesure où elle est interprétable ; et bien que nous ne soyons pas en position de comprendre en détail la base de données, nous évaluerons de manière qualitative et au mieux de notre connaissance les communautés statiques et dynamiques détectées. Il faut souligner que cet intérêt pour l'analyse de la composition et de l'agencement des communautés d'un réseau est rare dans la littérature scientifique revue et donc d'autant plus pertinent.

Chapitre 5

Expérimentations sur une base de données réelles

Ce chapitre présente les résultats des expérimentations effectuées sur la base de données compilant les interactions entre employés par l'entremise des mises à jour d'un logiciel informatique. Nous disposons de très peu d'information sur le jeu de données et le réseau qu'il représente outre le fait qu'un individu contribue de façon volontaire à mettre le logiciel à jour soit en étant l'auteur d'un quelconque ajustement, soit en collaborant au travail d'un autre, et ce, peu importe son emplacement géographique. Ainsi, rien ne garantit que nous soyons en mesure même de découvrir des communautés significatives ou interprétables. L'étude des algorithmes de détection et de leurs éventuelles solutions permettra d'éprouver les hypothèses selon lesquelles la structure communautaire sous-jacente aux données est d'une part différente d'un arrangement aléatoire et d'autre part liée à l'information sur la localisation d'un individu.

La première section propose une analyse descriptive dans le but de comprendre la nature du réseau en ce qui a trait au comportement des individus et aux relations par l'examen de la représentation graphique en sommets et liens incidents.

La seconde section évalue la praticité, la pertinence et éventuellement les résultats de chacun des algorithmes de détection de communautés dans un contexte d'une part statique et d'autre part dynamique.

Enfin, la troisième section confronte les partitions ou répartitions des diverses mo-

délisations tant du point de vue quantitatif que descriptif.

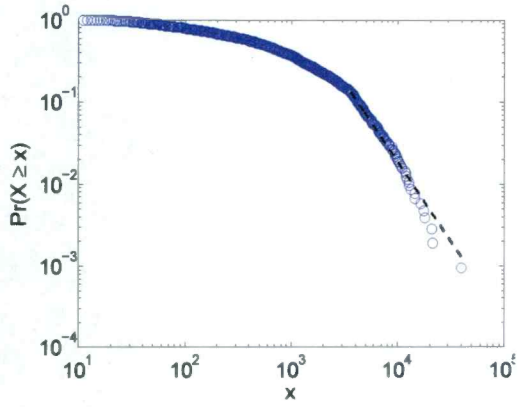
5.1 Analyse descriptive du jeu de données

Avant de modéliser les données, notons que l'activité de chaque individu - le nombre d'entrées dans la base de données associées à un des 1045 employés - pourrait plausiblement suivre une loi de puissance, distribution souvent observée dans la nature (Albert et Barabási [8], Clauset et al. [133], Newman [134]), où une quantité x obéit à une loi de puissance si sa probabilité de distribution est de la forme

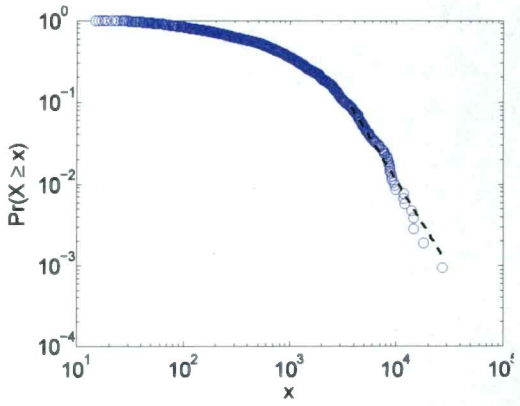
$$P(x) \propto x^{-\alpha},$$

où α est la constante de proportionnalité généralement dans l'intervalle $(2,3)$. Cette proposition est soutenue par la figure 5.1a représentant la distribution cumulative $P(x)$ et son ajustement par maximisation de la vraisemblance à une loi de puissance avec $\alpha = 2,92$, selon une méthodologie tirée de Clauset et al. [133]. Les tests d'ajustement de Kolmogorov-Smirnov avec seuil $0,1$ du modèle avec les paramètres estimés sur les données originales par une approche d'échantillonnage de type *bootstrap*, lesquels mesurent la distance entre la fonction cumulative de répartition des observations et celle qui ajuste au mieux les données, ne permettent pas d'écarter l'hypothèse nulle que la loi de puissance avec les paramètres calculés est plausible pour représenter les données, voir 5.1a. En conséquence, les degrés des sommets dans les graphes modélisés devraient adopter une distribution similaire.

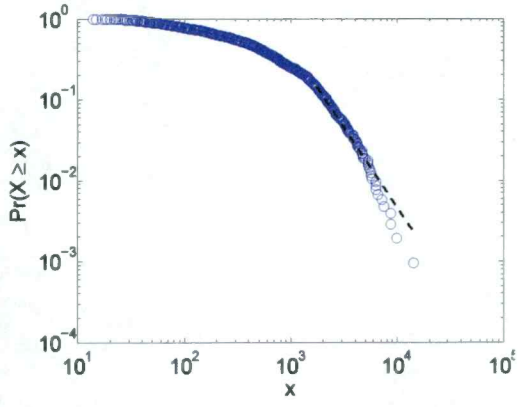
De fait, en analysant de nombreux réseaux sociaux, Albert et Barabási [135] ont observé que 1) les sommets d'un réseau de collaboration, comme le réseau que nous étudions, suivent une loi de puissance avec une traine épaisse ; 2) il existe des sommets avec un très grand degré. En fait, Barabási et al. [136, 137] arguent que les interactions sociales sont caractérisées par des crues d'occurrences suivies de longues périodes d'inactivité de sorte qu'elles peuvent être modélisées plus fidèlement par une distribution en loi de puissance avec une traine épaisse. Stouffer et al. [138] suggèrent qu'une loi log-normale décrirait statistiquement mieux la situation ; elle n'aurait cependant aucun fondement théorique en tant qu'interprétation du comportement humain (Barabási et al. [139]). Les histogrammes à la figure 5.2 présentent les répartitions des degrés pour chacune des modélisations - graphe projeté,



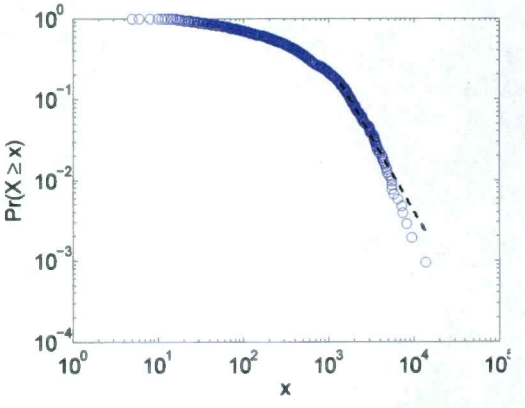
(a) $x =$ activité des individus enregistrée dans la base de données. $\alpha = 2,92$. Valeur-$p = 0,546$.



(b) $x =$ degré des sommets dans le graphe projeté. $\alpha = 3,11$. Valeur-$p = 0,148$



(c) $x =$ degré des sommets dans le graphe étoilé. $\alpha = 2,96$. Valeur-$p = 0,182$.



(d) $x =$ degré des sommets dans le graphe biparti. $\alpha = 2,83$. Valeur-$p = 0,102$.

FIGURE 5.1 – Distributions cumulatives $P(x)$ et leur ajustement par maximisation de la vraisemblance à une loi de puissance. Les ajustements et visualisations sont réalisés avec les fonctions **plift.m** et **plplot.m** pour MATLAB® par Clauset disponibles à l'adresse `http://tuvalu.santafe.edu/~aaronc/powerlaws/`. Les tests d'ajustement (voir les valeurs-p) ne permettent pas d'écarter l'hypothèse nulle que la loi de puissance avec les paramètres calculés est plausible pour représenter les données dans tous les cas.

biparti, étoilé. Les variables pourraient plausiblement suivre une loi de puissance, voir la distribution cumulative $P(x)$ et son ajustement par maximisation de la vraisemblance à une loi de puissance dans les figures 5.1c, 5.1b et 5.1d, mais aussi une loi log-normale (les observations pour des grandes valeurs de x, soit la queue de la série, semblent bien moins s'ajuster à la distribution théorique de la loi de puis-

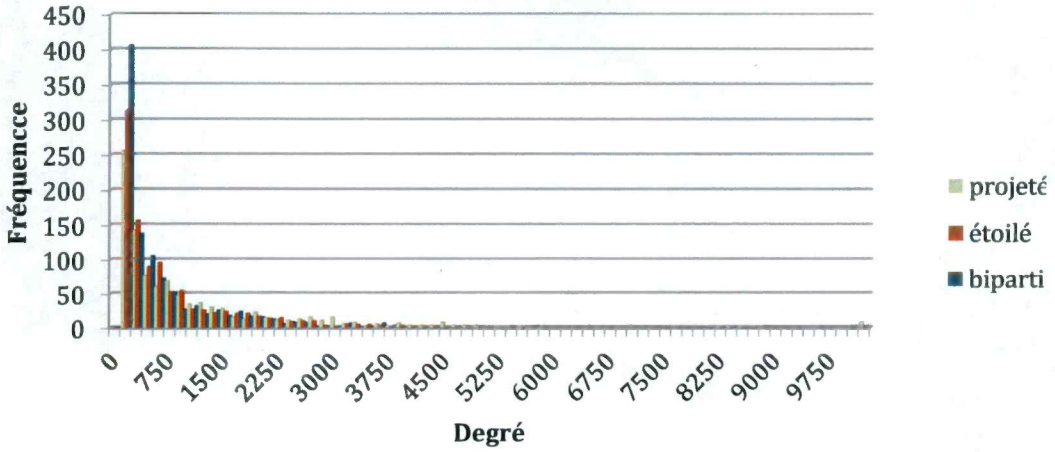


FIGURE 5.2 – Histogramme représentant la répartition des degrés des sommets pour chacun des graphes étoilé, projeté et biparti.

sance) : une étude plus approfondie du comportement des individus impliqués dans ce réseau de collaboration dépasse le cadre de ce mémoire et nous nous contentons d'observer que la majorité des sommets ont un faible degré, c'est-à-dire que la majorité des individus sont peu connectés ou actifs et que les traines épaisses des histogrammes 5.2 démontrent l'existence de sommets au degré très élevé, c'est-à-dire d'individus très connectés ou actifs, en accord avec les observations de Barabási et al. Dans les faits, les données semblent dépeindre une situation quelque part entre la règle générale [1] du 1 % (1 % des participants contribuent activement, 9 % sporadiquement, 90 % très rarement) et le principe de Pareto [140] (20 % des participants sont responsables de 80 % des contributions). Remarquons que les interactions du réseau considéré dans ce document sont beaucoup plus fréquentes que celles observées dans les réseaux de collaborations scientifiques tels que ceux étudiés par Albert et Barabási ; en particulier, certains employés posent plus de 20 000 actions donc beaucoup plus qu'un scientifique ne pourrait publier sur la même période. Le fait saillant de ce long préambule est qu'il est attendu que les communautés détectées exhibent un profil de distribution approchant (sans être identique) celui de la loi de puissance conformément avec la littérature scientifique [52, 125, 141].

De plus, peu importe le paradigme de modélisation, le graphe dessiné est connexe et clairsemé [2] ($M < N^2$). Le tableau 5.1 résume le nombres de liens, de sommets et le degré moyen pour chacun des paradigmes de modélisation. Il est à noter que les

1. *Rule of thumb*
2. *Sparse*

Modélisation	$\|V\|$	$\|E\|$	$\sum_u \frac{d_u}{N}$
Graphe étoilé	1045	42 146	832,93
Graphe projeté	1045	54 766	1300,30
Graphe biparti	281 395	715 555	5,09

TABLEAU 5.1 – Nombre de sommets, d'arêtes et degré moyen d'un sommet pour chacun des trois paradigmes de modélisation du graphe statique.

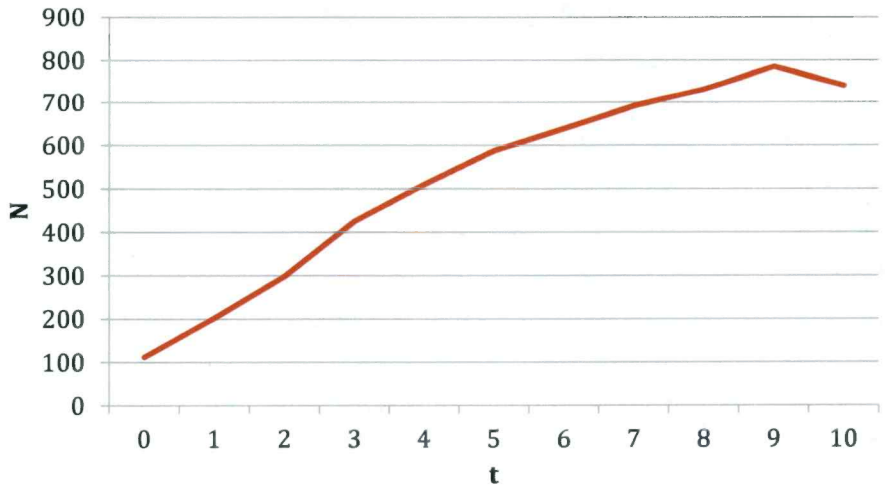


FIGURE 5.3 – Évolution du nombre de sommets dans les instantanés $G^{(t)}, t = \{0,\ldots,10\}$ des graphes projetés et étoilés. Notons la croissance plus ou moins linéaire du nombre de sommets *individus* sur les dix premiers intervalles.

graphes respectivement étoilé et projeté sont de taille relativement modeste en comparaison aux réseaux disponibles pour évaluation (voir [126]), alors que le graphe biparti peut quant à lui être considéré comme un graphe de grande dimension. Dans tous les cas, une solution exacte semble inaccessible, quelle que soit la méthode employée.

En ce qui a trait au réseau temporel, le nombre de sommets dans chacun des onze instantanés croît linéairement sur les dix premiers intervalles comme l'illustre la figure 5.3, ce qui est encore une fois compatible avec les remarques de Albert et Barabási [135] sur l'évolution des réseaux sociaux. Comme chacun des instantanés est disjoint, c'est-à-dire que les liens n'y sont pas cumulés d'un intervalle de temps à l'autre, la croissance linéaire du degré moyen des sommets, telle que présentée à la figure 5.4, implique que le nombre d'arêtes augmente beaucoup plus rapide-

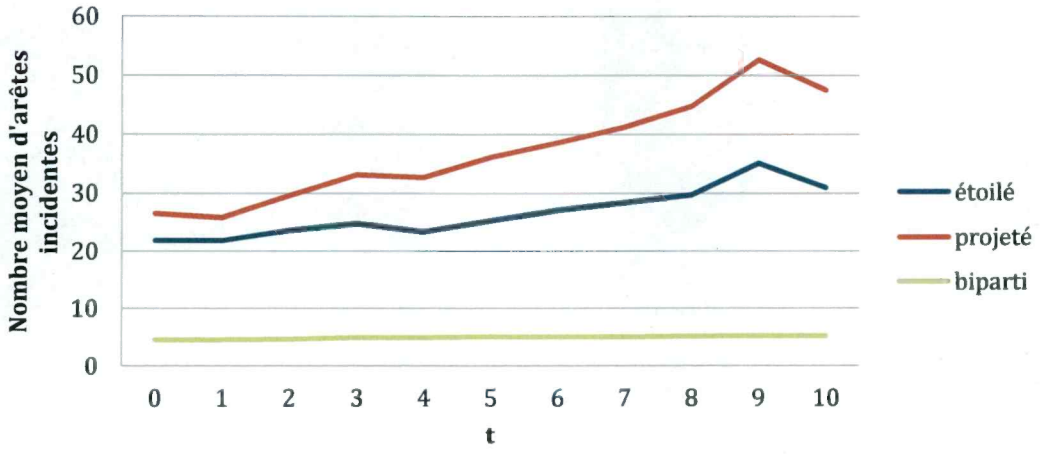


FIGURE 5.4 – Évolution du nombre moyen d'arêtes incidentes à un sommet sur les instantanés $G^{(t)}, t = \{0, \ldots, 10\}$ pour chacun des graphes projeté, étoilé et biparti. Dans le graphe biparti les sommets comprennent les individus et les mises à jour. Notons la croissance linéaire de cette mesure sur les 10 premiers intervalles.

ment que le nombre de sommets, phénomène que Dorogovtsev et Mendes [142] décrivent comme la «croissance accélérée» d'un réseau temporel, c'est-à-dire que les sommets sont de plus en plus densément connectés et le graphe de plus en plus gros. Ceci reste vrai, même sans cumuler de façon incrémentale les nœuds et les arêtes, soit dans le cas d'instantanés disjoints. Le tableau résumant les nombres de liens, de sommets et les degrés moyens sur chacun des intervalles pour chacun des paradigmes de modélisation est reporté en annexe B.4. La faible décroissance sur le dernier intervalle pourrait indiquer l'arrivée à maturité du logiciel et l'amorce de l'effritement des communautés et du réseau en général ; cependant, sans plus de données, nous ne ferons pas grand cas de ce détail. Quant à la structure modulaire, il peut être attendu qu'elle suive une évolution similaire, soit que les communautés sont plus nombreuses et plus denses avec le temps.

Si le graphe $G^{(t+1)}$ est plus large et plus dense que $G^{(t)}$, dans quelle mesure les sommets de $G^{(t)}$ sont-ils affectés par les modifications de $\Delta G^{(t)}$? Il s'avère que seul un très petit nombre d'individus (moins de 1 %) ont exactement le même voisinage d'un instantané à l'autre comme montré dans le tableau 5.2 pour les modélisations en graphe étoilé et projeté (tout sommet - individu ou mise à jour - d'un graphe biparti a, du fait que les mises à jour sont constamment renouvelées, un voisinage toujours différent).

	$\Delta G^{(0)}$	$\Delta G^{(1)}$	$\Delta G^{(2)}$	$\Delta G^{(3)}$	$\Delta G^{(4)}$	$\Delta G^{(5)}$	$\Delta G^{(6)}$	$\Delta G^{(7)}$	$\Delta G^{(8)}$	$\Delta G^{(9)}$
Graphe étoilé	0	0	0	3	1	0	2	1	0	1
Graphe projeté	0	0	0	1	0	0	0	1	1	0

TABLEAU 5.2 – Nombre de sommets au voisinage inchangé de $G^{(t)}$ à $G^{(t+1)}$.

5.2 Évaluation des algorithmes de détection de communautés

Nous avions initialement deux approches (statique et dynamique) ; trois paradigmes de représentation graphique (en graphe étoilé, biparti et sa projection) ; deux possibilités de type de graphe (pondéré et non pondéré) ; seize méthodes de détection dont cinq strictement statiques et onze avec recouvrement des communautés ; trois cadres d'appariement des communautés intertemporelles ; de nombreux paramètres à fixer ; etc. ; c'est-à-dire une quantité exponentielle de possibilités dont plusieurs seront écartées dans les sections qui suivent. Nous en expliquerons les raisons à mesure.

5.2.1 Cadre statique

Les résultats des algorithmes de détection de communautés statiques sont rapportés en annexe dans le tableau B.5 et sont discutés à la section 5.3. Les sous-sections ci-dessous détaillent les expérimentations sur le jeu de données complet - sans information temporelle - pour chacun des algorithmes choisis.

5.2.1.1 Louvain

L'algorithme de Blondel et al. nous permet de disposer de l'option des graphes non pondérés respectivement étoilé et projeté. En effet, la modularité de la partition calculée dans un graphe dont les arêtes sont toutes de poids unitaire dépasse à peine le seuil minimal exprimant l'existence d'une structure modulaire ; c'est-à-dire que la solution est à peine meilleure qu'une solution aléatoire. Nous nous permettons donc de nous concentrer pour la suite sur les graphes pondérés sauf dans le cas du

graphe biparti. **Louvain** a une composante aléatoire, nous retenons la meilleure de 20 itérations. Voir le détail des solutions dans le tableau B.5 en annexe.

5.2.1.2 CFinder

La complexité calculatoire - supposée polynomiale - de l'algorithme **CPM** de Palla et al. est difficile à évaluer précisément [60]. Chose certaine, l'implantation à l'intérieur de **CFinder** est incapable de détecter les communautés de notre base de données, et ce, même dans le cas des plus petits graphes que sont ceux étoilé et projeté. Nous sommes alors forcée de mettre de côté non seulement **CFinder**, mais aussi la version dynamique de la percolation de cliques **CPMDyn**.

5.2.1.3 COPRA

Il n'a pas été possible de trouver d'implantation de **COPRA** outre celle à l'intérieur de **OSLOM**, laquelle ne permet pas, par contre, l'utilisation de l'algorithme seul. Il ne semble pas non plus imaginable de programmer tous les algorithmes, nous nous contentons de tester **COPRA** uniquement en tant que paramètre d'**OSLOM**.

5.2.1.4 CSS

Les solutions de **CSS/Louvain**, **CSS/RAK** et **CSS/SIM** ont été obtenues par consensus sur dix itérations de chacun des algorithmes **Louvain**, **RAK** ou **SIM** respectivement - l'implantation disponible ne permet pas de mélanger les algorithmes, bien qu'il soit théoriquement possible de le faire. Nous avons fait l'exercice de comparer les solutions de chacune des itérations avec celle obtenue par consensus en termes de la valeur de la modularité, comme présenté dans le tableau 5.3 dans le cas où l'algorithme en paramètre est **Louvain**. Les valeurs reproduites dans ce tableau impliquent que la modularité de la solution consensuelle n'est pas meilleure que la moyenne des dix itérations (la remarque reste la même pour les algorithmes **RAK** et **SIM**) : il est légitime alors de s'interroger sur sa plus-value. Lancichinetti et Fortunato [73] assurent que l'objectif de leur méthode n'est pas d'améliorer l'optimum, mais plutôt d'offrir une solution médiane plus stable et représentative de la «véritable» topologie du réseau dont ils mesurent la qualité avec l'IMN. Force est

	Itération	$mod(\mathscr{C})$
Louvain	1	0,485652
	2	0,480788
	3	0,477922
	4	0,488573
	5	0,488315
	6	0,498273
	7	0,484363
	8	0,483085
	9	0,485170
	10	0,481359
	Moyenne	0,485350
CSS/Louvain		0,482767

TABLEAU 5.3 – Comparaison des modularités des solutions de dix itérations de **Louvain** et leur moyenne à la modularité de la solution de **CSS/Louvain** dans le graphe étoilé. $mod(\mathscr{C})$ (3.2) : modularité de Newman et Girvan [23].

de constater que l'amélioration de la solution n'est pas évidente à évaluer puisque nous ignorons la structure sous-jacente aux données. Cependant, il peut être supposé, si l'on en croit ses auteurs, que la structure modulaire fournie par **CSS** sera, au mieux, meilleure qu'une quelconque itération de l'algorithme en paramètre. Voir le détail des solutions dans le tableau B.5 en annexe.

5.2.1.5 iLCD

L'algorithme de Cazabet et al. construit par incrément le graphe avec l'ajout de sommets et d'arêtes. Dans la version statique, il s'agit de la répartition obtenue à la fin de l'horizon temporel sans retrait d'arêtes ou de sommets ; voir 4.2.4.2 avec $d = \infty$. **iLCD** est relativement lent et limité par l'allocation dynamique de mémoire : il n'a été en mesure de fournir de solution que dans le cas du graphe étoilé (le plus petit) et cette solution est difficilement interprétable avec 278 920 communautés de très petites tailles - deux ou trois sommets - se chevauchant énormément - un sommet est en moyenne dans 829,2 groupes. Voir le détail de la solution dans le tableau B.5 en annexe.

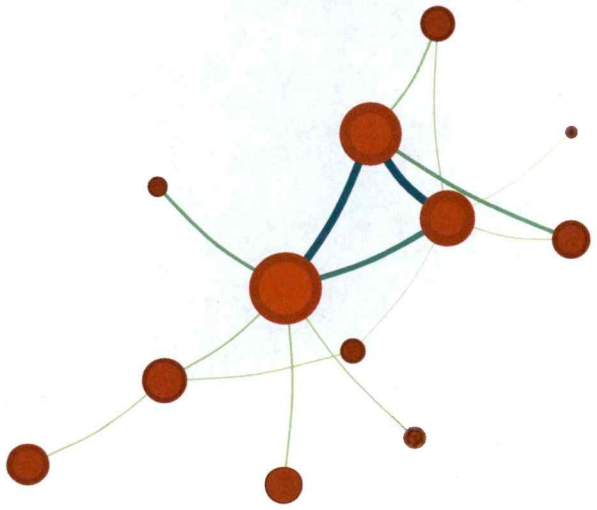


FIGURE 5.5 – Vue partielle de la répartition statique des sommets du graphe étoilé produite par l'application *Hierarchical Network Navigator* avec l'algorithme **Infomap**. Chacun des nœuds représente une communauté et sa taille est fonction du flot entrant et sortant. Les arêtes *inter*-communautés sont plus ou moins épaisses selon le flot.

5.2.1.6 Infomap

L'application *TheMapEquation* avec représentation graphique est de loin la plus simple et la plus pratique - cela ne garantit en rien de la qualité de la répartition. La figure 5.5 montre une vue partielle de la structure modulaire du réseau produite avec *Hierarchical Network Navigator*. La solution de l'algorithme **Infomap** est le pallier hiérarchique à encodage minimal de l'information, mais il est possible d'astreindre la solution à deux niveaux uniquement, c'est-à-dire un premier niveau où les sommets décrivent chacun une communauté et un second dans lequel ils sont regroupés. Les solutions multi-niveaux - soit optimales - des graphes étoilé et projeté s'avèrent à deux niveaux. Ce n'est cependant pas le cas du graphe biparti pour lequel la structure modulaire à encodage minimal est formée de méta-communautés de communautés de sommets, soit à trois niveaux.

Infomap a une composante aléatoire, l'application autonome retourne la meilleure de r itérations. Voir le détail des structures communautaires à deux niveaux avec et sans recouvrement des graphes étoilé ou projeté et celle à deux et trois niveaux du graphe biparti dans le tableau B.5 en annexe pour $r = 50$. Il n'a pas été possible de calculer la structure modulaire avec recouvrement du graphe biparti.

5.2.1.7 OSLOM

Lancichinetti et al. suggèrent que la multiplication des itérations des algorithmes **Infomap**, **Louvain** ou **COPRA** en paramètres permet une meilleure exploration des modules. Les solutions avec recouvrement dans le tableau B.5 en annexe sont produites par dix itérations de chacun des algorithmes, en plus des passages multiples de la procédure de nettoyage. **OSLOM** offre l'option de retirer les sommets non assignés - les singletons - du graphe, option que les auteurs comparent à filtrer le bruit du réseau. Sans celle-ci, les nœuds non-affiliés sont assignés de force à une communauté de sorte à maximiser un certain score. Notons que dans le cas d'un graphe biparti, les singletons sont tous des mises à jour : cette solution n'est pas particulièrement intéressante puisque nous voulons connaitre les regroupements d'individus plus précisément. L'option sera donc ignorée dans le calcul de la répartition dynamique du graphe biparti.

5.2.1.8 RAK et SIM

Il est impossible d'évaluer si une solution fournie par **RAK** - respectivement **SIM** - est significativement différente de celle obtenue par consensus sur plusieurs itérations comme expliqué plus haut (voir 5.2.1.4). Seule la solution consensuelle supposée meilleure **CSS/RAK** - respectivement **CSS/SIM** - est retenu dans le tableau B.5 en annexe.

5.2.2 Cadre dynamique

Les résultats des algorithmes de détection de communautés dynamiques sont rapportés dans les tableaux B.6 et B.7 en annexe. Les sous-sections ci-dessous détaillent les expérimentations sur le jeu de données pour chacun des algorithmes choisis.

5.2.2.1 A^3CS

L'algorithme avec maximisation de la modularité suppose la réévaluation des sommets de $\Delta G^{(t)}$. Or, plus de 99 % des individus dans les graphes respectivement étoilé

et projeté - 100 % pour le graphe biparti - sont affectés par des modifications : soit ils apparaissent pour la première fois à $t+1$, soit leur voisinage est modifié d'une quelconque façon. Ainsi, une telle approche incrémentale n'exploite nullement l'information temporelle et devient alors strictement équivalente à celles indépendantes sur des instantanés. Il est donc inutile de poursuivre l'étude de **A^{3}CS**.

5.2.2.2 AFOCS

L'algorithme ne concerne que les graphes non pondérés : il ne peut qu'être appliqué au graphe biparti. Le paramètre de recouvrement β est fixé à 0,7 comme suggéré par Nguyen et al. [101] malgré que ce choix demeure arbitraire et sujet à approfondissement. L'implantation disponible d'**AFOCS** n'a pas été en mesure de fournir de solution, outre sur les intervalles un et deux, à cause de la taille du graphe, des nombreux bogues ou fuites de mémoire, ce même en corrigeant le code.

5.2.2.3 CPMDyn

Voir **CFinder**, 5.2.1.2.

5.2.2.4 CSS

Dans l'objectif d'exploiter l'information de la structure modulaire à des temps antérieurs et de suivre l'évolution des communautés, **CSS** devrait calculer la solution consensuelle entre $\mathscr{C}^{(t)}$ et $\mathscr{C}^{(t+1)}$, puis celle entre la même partition $\mathscr{C}^{(t+1)}$ et $\mathscr{C}^{(t+2)}$, etc. Malheureusement, l'implantation de l'algorithme ne permet pas de reprendre la même partition en paramètre, de sorte que la solution est en fait une moyenne entre $\mathscr{C}^{(t)}$ et $\mathscr{C}^{(t+1)}$, puis une autre entre $\mathscr{D}^{(t+1)}$ et $\mathscr{C}^{(t+2)}$, $\mathscr{D}^{(t+1)} \neq \mathscr{C}^{(t+1)}$, sans relation entre les deux. **CSS** tel qu'il est disponible ne peut alors répondre au problème posé mieux que n'importe quel autre algorithme utilisé de façon indépendante sur chaque instantané, nous ne poursuivrons alors pas son évaluation.

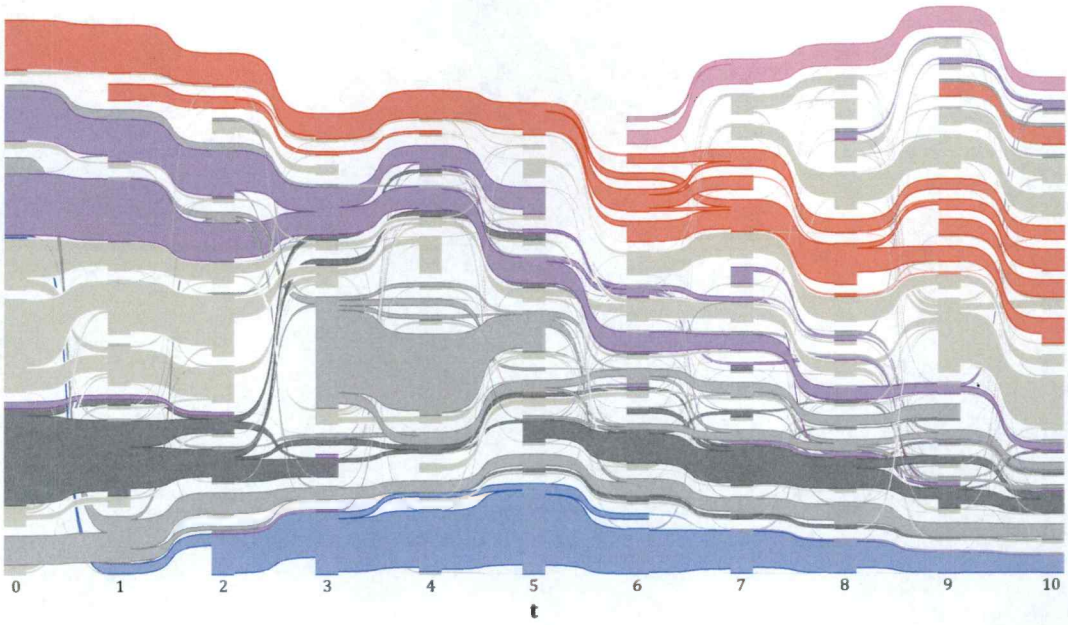


FIGURE 5.6 – Diagramme alluvial représentant l'évolution des communautés détectées par l'algorithme **Infomap** à deux niveaux hiérarchiques dans le graphe étoilé et produit avec l'application *Alluvial Generator* (`http://www.mapequation.org/`). Sur l'axe vertical, les sommets sont regroupés en communautés. Les apparitions temporelles d'un même sommet (d'une même couleur) d'un instantané à l'autre sont liées par une bande d'une certaine épaisseur. Il est possible de suivre l'évolution de certaines communautés par leur apparition, fusion, division, etc., en suivant un ruban coloré sur plusieurs périodes Mentionnons qu'il s'agit d'une vue partielle du réseau et que la largeur des bandes n'est pas proportionnelle à la taille des communautés mais au volume de flot.

5.2.2.5 Infomap

L'algorithme est utilisé de façon indépendante sur les instantanés ; le tableau en annexe B.6 montre le nombre de communautés dans chacun d'entre eux et B.7 leur taille moyenne. La figure 5.6 présente le diagramme alluvial crée avec *TheMapEquation*. L'évolution de certaines communautés sur tout ou une partie de l'horizon temporel peut en être déduite, par contre l'appariement entre les communautés d'un instantané à l'autre est implicite : il est possible de suivre le flot d'un groupe de sommets à travers le diagramme, mais il nous revient la tâche de décider «visuellement» dans quelle mesure il s'agit de la même communauté ou d'une nouvelle.

Lorsque l'algorithme n'est pas contraint à fournir une solution sur deux niveaux hiérarchiques, il tend à former en alternance des méta-communautés sur trois niveaux

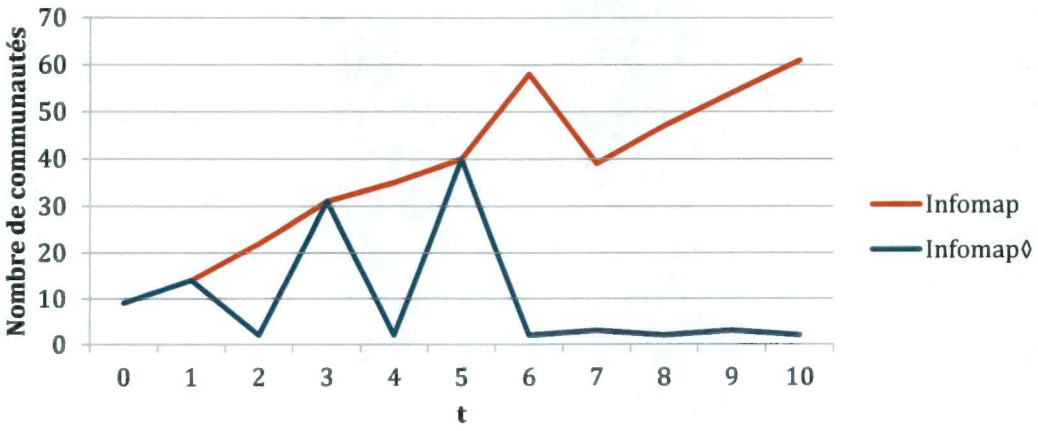


FIGURE 5.7 – Évolution du nombre de communautés dynamiques détectées avec l'algorithme **Infomap** dans chacun des instantanés $G^{(t)}, t = \{0, \ldots, 10\}$ du graphe étoilé. ◇Version d'**Infomap** multi-niveaux.

et des plus petites sur deux niveaux ainsi que présenté à la figure 5.7, c'est-à-dire qu'il «saute» d'un niveau hiérarchique à l'autre en quelque sorte. Selon cette figure, pour $t = 1,2,3$, la structure modulaire à deux niveaux passe de quatorze à 22 puis 31 communautés ; celle à encodage minimal passe de quatorze modules à deux méta-communautés lesquelles se disloquent en 31 communautés. Ces sauts d'un niveau hiérarchique à l'autre causent des discontinuités dans la chronologie des événements affectant les communautés. Pour cette raison, nous ignorons les solutions multi-niveaux dans l'évaluation des sections suivantes.

5.2.2.6 LabelRankT

Notre implantation de l'algorithme **LabelRankT** n'a pas été en mesure de reproduire les exemples présentés par Xie et al. dans [110, 111, 143], faute de précision sur la procédure de normalisation de la matrice de distribution des étiquettes. Notons que **LabelRankT** est déterministe et doit donc toujours arriver à la même solution : le fait d'en obtenir une autre ébranle la confiance que nous avions dans la méthode. Nous mettons de côté cette dernière, les auteurs ayant refusé d'éclairer nos lumières devant ces inconsistances.

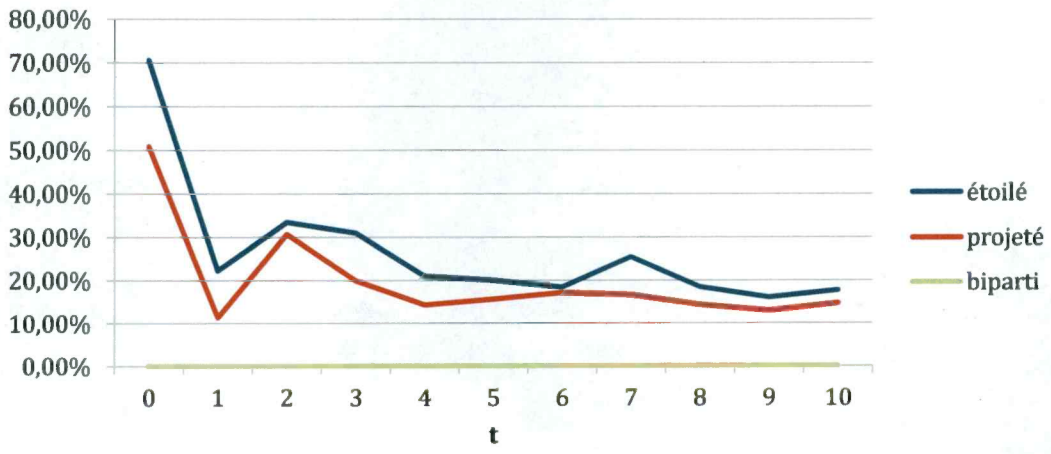


FIGURE 5.8 – Évolution du pourcentage de sommets - représentant les individus uniquement - non affiliés à une quelconque communauté par l'algorithme **OSLOM** sur les instantanés $G^{(t)}, t = \{0, \ldots, 10\}$. Les valeurs numériques et d'autres précisions sont disponibles dans le tableau B.8 en annexe.

5.2.2.7 Louvain

L'algorithme est utilisé de façon indépendante sur les instantanés. Voir le détail des solutions dans les tableaux B.6 et B.7 en annexe.

5.2.2.8 LouvainDyn

Cette version dynamique de **Louvain** suppose de redémarrer l'algorithme à t en isolant les nœuds affectés pas les modifications depuis $t-1$. Il est rejeté pour la même raison que **A^{3}CS** l'avait été, 5.2.2.1.

5.2.2.9 OSLOM

Il est possible de fournir à l'algorithme la répartition $t-1$ dans le calcul de la solution à t. Comme il l'avait été fait dans le cas statique, les solutions avec et sans sommets retirés, dont les détails apparaissent dans les tableaux B.8, B.6 et B.7 en annexe, sont produites par dix itérations de chacun des algorithmes **Infomap**, **Louvain** et **COPRA**, en plus des passages multiples de la procédure de nettoyage. La figure 5.8 présente l'évolution du nombre de sommets retirés par **OSLOM** lorsque l'option est considérée. Elle suggère une structure robuste sur une partie minoritaire du graphe (entre 30 % et 50 %) à $t = 0$, laquelle devient beaucoup plus grande (entre

	d	Communautés *vivantes*	Communautés *mortes*	Total
	1	14	14 045	14 059
	7	162	19 361	19 523
	30	941	39 058	39 999
Graphe	60	4852	117 653	122 505
étoilé	90	11 816	154 855	166 671
	180	35 305	197 044	232 349
	365	84 059	170 689	254 748
	∞	278 920	0	278 920

TABLEAU 5.4 – Nombre de communautés dans chacun des instantanés du graphe étoilé proposé par l'algorithme de détection dans le contexte dynamique **iLCD** pour plusieurs valeurs de d. Les communautés dites *vivantes* existent encore à t_{10} alors que les communautés dites *mortes* ont disparu quelque part sur l'intervalle $[t_0, t_{10}]$

70 % et 88 %) avec le temps ; c'est-à-dire que le bruit dans le réseau diminue à mesure que sa structure communautaire se précise et se renforce. Nous ignorons à cette étape s'il s'agit d'une conséquence de l'algorithme ou de la nature des données et la chute drastique du pourcentage de sommets retirés à $t = 1$ reste à expliquer.

5.2.2.10 PDEC

Il n'a pas été possible de trouver d'implantation de l'algorithme.

5.2.2.11 FacetNet

L'implantation de l'algorithme suppose le nombre de communautés constant sur tout l'horizon temporel et la relative lente évolution de ces dernières. Ces hypothèses semblent trop fortes en regard de la progression du nombre de sommets, observée à la section 5.1, et sa conséquence probable sur le nombre de modules, autant qu'à la lumière des résultats offerts par les autres méthodes (**OSLOM**, **Louvain**, **Infomap**) : l'algorithme est rejeté.

5.2.2.12 iLCD

La méthode de Cazabet et al. semblait *a priori* avoir un grand potentiel à résoudre le problème de la détection de communautés dans notre base de données, mais ses résultats sont pour le moins surprenants ou plutôt décevants. **iLCD** procède par incrément, donc les liens et les sommets peuvent apparaitre lorsqu'une relation entre deux personnes se crée et disparaitre après un certain laps de temps d si elle n'est pas ravivée entre temps. Le tableau 5.4 montre le nombre de communautés encore vivantes à la fin de l'horizon temporel ou mortes entre temps pour $d = 1, 7, 30, 60, 90, 180$ ou 365 jours et pour $d = \infty$; soit lorsque la relation entre des individus perdure un jour, une semaine, un mois, deux mois, trois mois, six mois, un an ou toujours. Ni la solution statique - avec $d = \infty$ - ni aucune des solutions dynamiques ne sont satisfaisantes parce que dépassant la dizaine de milliers de communautés. La structure modulaire est si morcelée et les communautés si imbriquées les unes à l'intérieur des autres qu'il devient impossible de représenter et d'interpréter le résultat.

5.2.3 Appariement des communautés

Une fois les communautés connues sur chacun des instantanés, il faut comprendre leur évolution à l'aide d'un cadre méthodologique d'appariement et d'une mesure de similitude ; ici, l'indice de Jaccard (3.14), la mesure de Takaffoli et al. (3.19) ou **CommTracker**. Et si l'historique des communautés offert par les deux premières est consistant, celui de **CommTracker** mène à certaines incongruités. Prenons deux exemples tirés de la partition indépendante sur des instantanés par l'algorithme **Louvain** sur le graphe étoilé afin de démontrer pourquoi la méthodologie de **CommTracker** sera écartée.

À $t = 5$, les communautés $C_3^{(5)}, C_5^{(5)}, C_7^{(5)}, C_{13}^{(5)}$ ont respectivement onze, trois, dix-huit et seize sommets en commun avec $C_8^{(6)}$ ainsi que deux, un, un et quatre nœuds centraux en commun. Alors, d'après **CommTracker**, la communauté $C_8^{(6)}$ nait par fusion des communautés $C_3^{(5)}, C_5^{(5)}, C_7^{(5)}, C_{13}^{(5)}$ tout simplement parce qu'elle a des nœuds centraux en commun avec plus d'une communauté à une période antérieure. Les mesures de Jaccard ou de similitude de Takaffoli et al. avec seuil $\theta \leq 0,25$ amènent une interprétation différente : soit que $C_7^{(5)}$ croît par agglomération de sous-

$C_i^{(t)}$	$\lvert C_i^{(t)}\rvert$	$C_j^{(t+1)}$	$\lvert C_j^{(t+1)}\rvert$	$\lvert C_i^{(t)} \cap C_j^{(t+1)}\rvert$	Nœuds centraux	J	*sim*
$C_3^{(5)}$	119	$C_8^{(6)}$	66	11	2	0,0632	0,0924
$C_5^{(5)}$	85	$C_8^{(6)}$	66	3	1	0,0203	0,0353
$C_7^{(5)}$	22	$C_8^{(6)}$	66	18	1	**0,2571**	**0,2727**
$C_{13}^{(5)}$	37	$C_8^{(6)}$	66	16	4	0,1839	0,2424
$C_{10}^{(7)}$	68	$C_1^{(8)}$	30	8	4	0,0889	0,1176
$C_{10}^{(7)}$	68	$C_8^{(8)}$	57	50	9	**0,6667**	**0,7353**

TABLEAU 5.5 – Exemples tirés de la partition indépendante sur des instantanés par l'algorithme **Louvain** sur le graphe étoilé. Dans le premier, les communautés $\{C_3^{(5)}, C_5^{(5)}, C_7^{(5)}, C_{13}^{(5)}\}$ à $t = 5$ contribuent à la communauté $C_8^{(6)}$ à $t = 6$, soit des sous-groupes issus de plusieurs communautés sont fusionnés. Dans le second, la communauté $C_{10}^{(7)}$ à $t = 7$ contribue aux communautés $\{C_1^{(8)}, C_8^{(8)}\}$ à $t = 8$.
Nœuds centraux : nombre de nœuds centraux en commun calculés par **CommTracker**. $J = J(C_i^{(t)}, C_j^{(t+1)})$: indice de Jaccard. $sim = sim(C_i^{(t)}, C_j^{(t+1)})$: similitude selon Takaffoli et al. Plus J et *sim* sont proches de un, plus la correspondance est grande entre les communautés.

groupes provenant d'autres communautés et devient $C_8^{(6)}$. Les mesures reprises dans le tableau 5.5 indiquent toutefois une relativement faible correspondance entre les deux.

À $t = 7$, la communauté $C_{10}^{(7)}$ a respectivement huit et 50 sommets en commun avec $C_1^{(8)}$ et $C_8^{(8)}$ et respectivement quatre et neuf nœuds centraux en commun. Alors, d'après **CommTracker**, la communauté $C_{10}^{(7)}$ se divise en deux nouvelles communautés ($C_1^{(8)}$ et $C_8^{(8)}$) parce qu'elle a plus de deux nœuds centraux en commun avec plus d'une communauté à une période postérieure. Les mesures de Jaccard ou de similitude de Takaffoli et al. avec seuil $\theta \leq 0,65$ amènent une interprétation différente, soit que $C_{10}^{(7)}$ décroît, mais reste foncièrement la même communauté car la correspondance avec $C_8^{(8)}$ est très forte comme inscrit dans le tableau 5.5, $C_{10}^{(7)} \rightsquigarrow C_8^{(8)}$.

En fait, **CommTracker** qui a l'avantage d'être sans paramètres, peine à faire la différence entre la fusion et l'absorption (de façon équivalente entre la division et l'attrition) : lorsque des parties de plusieurs communautés distinctes se rassemblent, il est légitime de croire que la plus grande a en fait absorbé les plus petites surtout lorsqu'elle représente la majorité de l'agglomération en même temps que de repré-

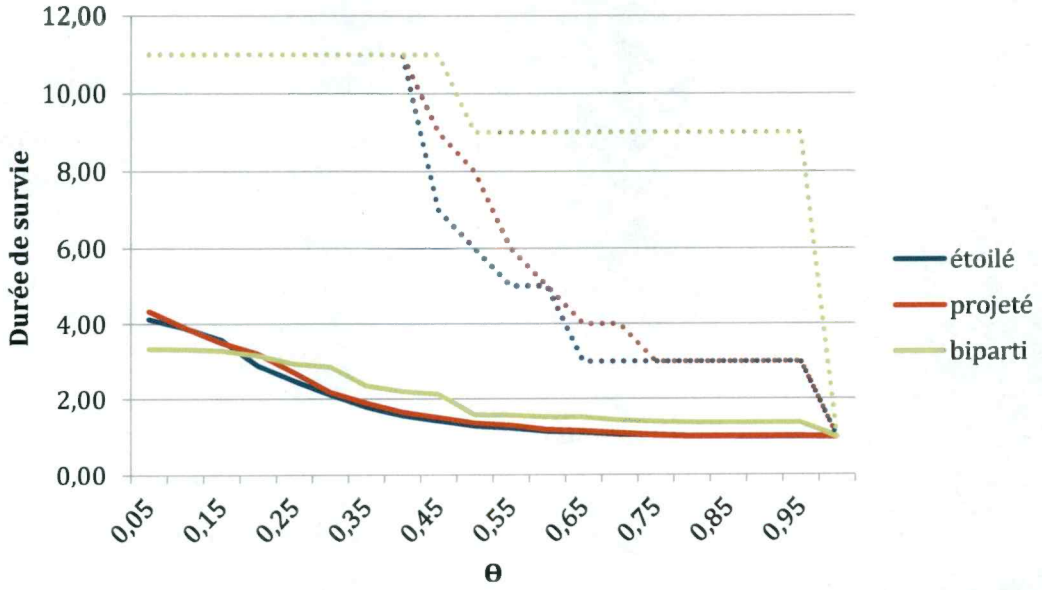


FIGURE 5.9 – Relation entre la durée de survie, en nombre d'intervalles, des communautés intertemporelles calculées avec l'algorithme **Infomap** et la valeur du paramètre θ comme seuil d'appariement avec l'indice de Jaccard. Durée moyenne en trait plein, durée maximale en trait pointillé.

senter un large groupe de sa communauté d'origine. Ces concepts sont implicites à l'indice de Jaccard et la similitude de Takaffoli et al. en particulier pour des petites valeurs du seuil θ. Il faut cependant faire attention aux trop petites valeurs de θ qui amènent à la «collision» de communautés, c'est-à-dire que k communautés $C_i^{(t)}, i \in \{1,2,\ldots,k\}$ sont appariées à une communauté à $C^{(t+1)}$. Dans ce cas, la correspondance entre les communautés se fait par maximisation de la mesure comme règle de décision supplémentaire. La procédure complète d'appariement des modules est reportée à l'annexe A.

La figure 5.9 exprime la relation entre la valeur de θ et la durée de survie moyenne des communautés des solutions de **Infomap** où la similitude est calculée par l'indice de Jaccard. Pour des petites valeurs de θ, les modules survivent sur de plus longues périodes ; pour des grandes valeurs de θ, ils sont de courte durée. Dans le cas d'un graphe fortement dynamique, tel que celui produit par le jeu de données étudié, la valeur du seuil doit être relativement faible autrement très peu de communautés survivraient sur plus d'une période puisqu'une grande part de leurs membres peut être renouvelée d'un intervalle à l'autre. Cependant, nous ne disposons d'aucune information supplémentaire permettant d'établir une (ou plusieurs) valeur optimale de θ contrairement à Takaffoli et al. [69] par exemple qui avaient

utilisé la correspondance dans les thèmes des courriels entre employés dans l'étude de la base de données **ENRON** ou la correspondance dans les sujets d'articles en collaboration entre scientifiques dans l'étude de la base de données **DBLP**. À défaut de cela, nous établirons θ selon des considérations qualitatives.

5.3 Comparaison des résultats

De toutes les méthodes considérées, il n'en reste au final que sept (dix en comprenant les options) dans le cadre statique et quatre (cinq en comprenant les options) dans le cadre dynamique ayant pourvu des solutions cohérentes. Les sous-sections suivantes discutent des résultats selon le contexte.

5.3.1 Cadre statique

L'analyse qui suit est basée sur les solutions offertes par les méthodes de détection de communautés statiques **iLCD**, **CSS/Louvain**, **CSS/RAK**, **CSS/SIM**, **Infomap** avec et sans recouvrement, **Infomap** à deux ou multiples niveaux, **Louvain**, **OSLOM**, **OSLOM** avec retrait des sommets non-affiliés, telles que décrites dans les tableaux B.5 et B.9 en annexe.

D'une part, le nombre de communautés détectées varie énormément d'une méthode à l'autre, allant de 8 à 409 (moyenne 92,5, écart-type 132,98) sans la solution d'**iLCD**, mais c'est seulement dans le graphe biparti que le nombre de communautés détectées dépasse la centaine. Alors, plusieurs de ces modules sont formés d'un très petit nombre d'individus, voire d'un seul individu, et de plusieurs mises à jour. Ils sont par conséquent peu intéressants. C'est par exemple le cas pour **CSS/Louvain**, **CSS/RAK** et **Infomap** à deux niveaux où respectivement 50, 80 et 257 individus forment des modules de un élément, quoiqu'il soit possible d'interpréter ces sommets comme inclassables, c'est-à-dire non clairement affiliés à un groupe comme le propose aussi une des versions d'**OSLOM**. Les algorithmes paraissent arriver à des résultats comparables en termes de nombre de communautés dans les graphes étoilé et projeté, avec légèrement plus de groupes dans le second, mais ils mettent en évidence les niveaux hiérarchiques. En effet, une structure modulaire comme celle offerte par **CSS/Louvain** avec 75 (respectivement 60) communautés dans le

graphe étoilé (respectivement projeté) présente un découpage plus précis de celle offerte par **Louvain** avec treize communautés (respectivement douze). Notons que la différence de $N_{\mathscr{C}}$ entre **CSS/Louvain** et **Louvain** n'est pas due à la procédure de consensus, mais à l'implantation puisque les solutions proviennent de deux versions différentes de l'algorithme. En effet, **CSS/Louvain** procède au calcul d'une solution médiane basée sur r itérations de **Louvain** dont la structure communautaire est du même ordre de grandeur (autour de 75 modules pour le graphe étoilé, respectivement 60 pour le graphe projeté) alors que la seconde version de **Louvain** offre des solutions dans la dizaine de modules et d'une bien meilleure modularité. Pourquoi **CSS/Louvain** s'arrête-t-il à un niveau hiérarchique inférieur malgré le fait qu'il n'ait pas atteint une valeur optimale de la modularité, telle qu'inscrite dans le tableau B.9 en annexe ? S'agit-il d'un optimum local sur lequel l'heuristique fige (et la raison pour laquelle le recuit simulé, **CSS/SIM**, produit une solution du même ordre de grandeur que **Louvain** avec une grande amélioration de la modularité) ? Nous doutons de la qualité des solutions de **CSS/Louvain** : pour constituer une amélioration de **Louvain** encore faudrait-il que le consensus se fasse sur ses meilleures partitions, ce qui n'est pas le cas ici.

Outre le nombre de modules, c'est surtout la taille de chacun d'entre eux qui renseigne sur le comportement des algorithmes. Comme l'ont observé Palla et al. [52], la distribution en loi de puissance ne se traduit pas parfaitement à l'échelle mésoscopique, voir les figures B.2 et B.3 représentant les distributions des tailles des communautés du graphe projeté et les ajustements à une loi de puissance en annexe. Les observations qui suivent s'appliquent également à toutes les modélisations. La variable $x =$ taille des communautés pourrait être plus fidèlement ajustée à d'autres distributions telles exponentielle ou log-normale selon l'algorithme ; sauf pour **OSLOM** dont la solution a une distribution en loi triangulaire. En fait, le processus de nettoyage des multiples itérations de **COPRA**, **Louvain** et **Infomap** à l'intérieur d'**OSLOM** amenant à la découverte de communautés statistiquement significatives semble lisser les solutions extrêmes. Mais ce n'est pas tant de déterminer quelles lois sont les mieux ajustées à x que de constater l'existence de différences énormes entre les distributions des tailles des communautés détectés qui est ici une conclusion majeure. En effet, s'il existe une structure modulaire «véritable», témoin de la topologie du réseau, ne devrait-on pas s'en rapprocher peu importe l'algorithme utilisé ? Le fait que les sommets soient distribués dans des modules dont la taille est

une conséquence de l'algorithme utilisé est un obstacle fondamental au problème de détection de communautés. En particulier, **RAK**, avec ou sans consensus, regroupe plus de 700 sommets à l'intérieur (> 70 % du graphe) d'un même module. La taille d'une telle communauté dépasse largement le nombre de Dunbar [144] ($\approx$ 150 individus) considéré comme la limite naturelle au bon fonctionnement du groupe (voir aussi Leskovec et al. [130]) et met en doute sa validité et son intelligibilité. La propagation d'étiquettes forme parfois, selon la densité du graphe [57], un super-module englobant la majorité du réseau : nous en avons un exemple ici. Dans une moindre mesure, tous les algorithmes étudiés engendrent la création d'une super-communauté et d'une majorité de petites communautés - fait illustré par les distributions exponentielle, log-normale ou puissance, en particulier leur traine plus ou moins longue et épaisse - puisqu'il s'agit là du portrait attendu de la répartition de la taille des communautés. Et si la distribution près de la loi de puissance est «naturelle», c'est-à-dire qu'elle peut être observée dans la nature, il n'en demeure pas moins que son portrait précis est le résultat du choix de l'algorithme. Il pourrait être intéressant de valider cette observation avec d'autres bases de données.

En ce qui concerne le chevauchement des communautés, **Infomap**, **OSLOM** et **iLCD** proposent des résultats difficilement interprétables. Oublions la structure modulaire fortement recouvrante de **iLCD** où 928 des 1045 individus sont dans plus d'une communauté (certains sont même dans plus de 1000 groupes, ce qui ne représente aucunement une situation réelle) ; celles de **Infomap** et **OSLOM** traduisent sans surprise une forte corrélation linéaire entre le nombre de communautés et le nombre d'individus associés à plus d'une communauté. Mais qu'en dire de plus ? Ces sommets font-ils véritablement partie de plusieurs modules, ou est-ce plutôt leur affiliation ambigüe qui les place de la sorte ? Il demeure plus de questions que de réponses tant en ce qui a trait à l'évaluation topologique du graphe qu'à l'interprétation qualitative du réseau d'employés. L'intersection des ensembles $Y_i, i = \{1, \ldots, 8\}$, soit les ensembles de sommets appartenant à plus d'une communauté, de chacune des huit solutions (**Infomap** avec recouvrement dans le graphe étoilé et le graphe projeté, **OSLOM** et **OSLOM** avec retrait des sommets non-affiliés dans les trois graphes) est vide, c'est-à-dire qu'aucun sommet n'est, pour toute solution, affilié à plus d'une communauté ; l'union contient le trois-quarts des sommets de V, c'est-à-dire que le quart des sommets sont dans X_i, l'ensemble des sommets dont l'affiliation est stricte et unique ; les autres sont parfois dans Y_i, par-

fois dans X_i (et Z_i, l'ensemble des sommets non affiliés, s'il y a lieu), $i = \{1, \ldots, 8\}$; en particulier moins d'un dixième des sommets ont plus souvent plus d'une appartenance qu'ils n'ont une appartenance stricte. Cependant, il semble problématique de comparer des structures modulaires exhibant des paliers hiérarchiques différents (**Infomap** et **OSLOM**), puisque des sommets pourraient appartenir à plus d'une communauté au niveau inférieur, lesquelles sont regroupées dans une méta-communauté au niveau supérieur. Le peu de connaissances dont nous disposons à propos de la base de données parait limiter l'interprétation que nous pouvons faire du chevauchement de modules outre le fait que le nombre d'individus vraisemblablement à l'intersection de plusieurs groupes est assez faible.

D'autre part, en termes de valeur de la modularité, les méthodes basées sur l'optimisation (**Louvain**, **CSS/Louvain** et **CSS/SIM**), en particulier dans le graphe biparti, font sans surprise meilleure figure voir pour cela le tableau B.9. D'aucuns pourraient mettre en doute de la pertinence de l'optimisation de la modularité «normale» dans le cas d'un graphe biparti. À cela nous répondons que la modularité bipartie converge vers la modularité régulière dans le graphe étudié, comme montré dans le tableau B.9, de sorte que maximiser l'une ou l'autre engendre sensiblement le même résultat. Une démonstration plus rigoureuse serait essentielle, mais elle dépasse le cadre de ce mémoire. La modularité, peu importe sa version - normale (mod), bipartie (mod_b) ou avec recouvrement (mod_o) - n'apporte que peu d'information. La conductance moyenne est aussi optimale (minimale) pour **Louvain**, **CSS/Louvain** et **CSS/SIM**. Les algorithmes détectant les modules avec recouvrement ont, quant à eux, les plus grandes valeurs, mais il s'agit là d'une conséquence de la structure de la solution : des nœuds appartiennent à plus d'une communauté, ils sont alors nécessairement bien connectés avec le complément de chacun des modules auxquels ils appartiennent. Dans ce cas, la conductance moyenne n'a pas de réelle interprétation quant à la qualité de la solution.

Si la modularité et la conductance moyenne du graphe renseignent à propos de la qualité de la partition du point de vue topologique, elles ne permettent pas d'apprécier précisément la nature des solutions. Il faudrait pour cela regarder la composition de chacune des communautés, ce qui n'est pas réaliste dans le contexte de l'analyse de larges jeux de données. Dans le cas qui nous intéresse, ni la modularité ni la conductance ne permettent de faire le lien entre la composition des communautés et l'information sur l'appartenance géographique des individus. *A priori*, les

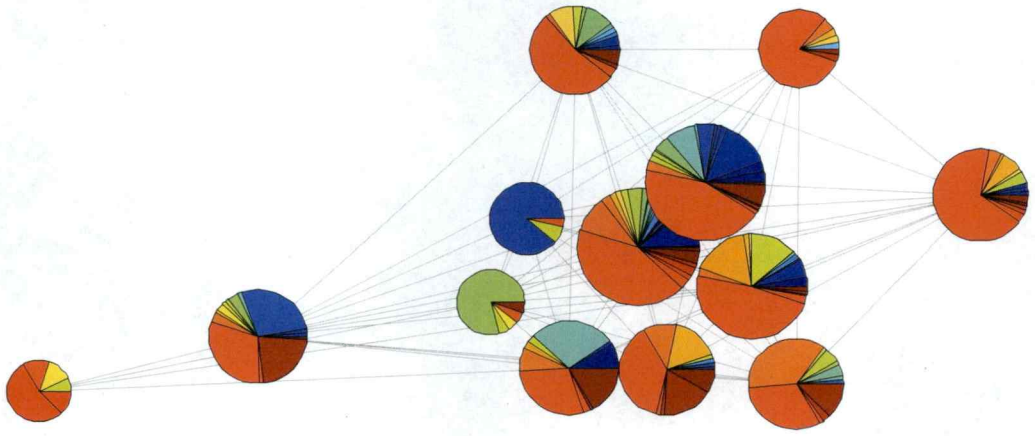


FIGURE 5.10 – Communautés détectées par l'algorithme **Louvain** dans le graphe étoilé statique, soit avec les données de l'ensemble de l'horizon temporel de 1974 jours. Les couleurs indiquent l'affiliation à l'un des 46 bureaux. Certaines communautés sont composées de manière dominante par des employés d'un seul bureau alors que d'autres sont composées majoritairement d'employés de deux ou trois bureaux en parts plus ou moins égales. La visualisation est produite avec les outils de Traud et al. [145] pour MATLAB® modifiant les algorithmes basées sur les forces (*Force-based* ou *Force-directed algorithms*) de Fruchtermann et Reingold [146] et Kamada et Kawai[147] et disponibles à l'adresse `http://netwiki.amath.unc.edu/VisComms/VisComms`.

employés ne sont pas encouragés à participer aux mises à jour de leurs collègues proches physiquement ; ils pourraient entretenir des relations plus éparpillées. Cependant, comme la figure 5.10 le démontre, certaines communautés - pas toutes - semblent présenter une forte homogénéité en regard de la dépendance à un bureau particulier. Nous proposons de faire un parallèle entre la diversité à l'intérieur d'une même communauté, selon le bureau d'origine d'un employé, et la fractionalisation ethnique [148] définie comme la segmentation de la population d'un pays, ou territoire délimité pas des frontières, en groupes distincts en matière de langue, culture, religion, ou autre facteurs socio-démographiques, et calculée par

$$\Upsilon_i = 1 - \sum_j s_{ij}^2,$$

où s_{ij} est la part du groupe ethnique j dans la population du pays i. L'indice exprime la probabilité que deux individus choisis au hasard appartiennent à des groupes ethniques distincts. Ainsi, nous substituons l'idée d'appartenance à un bureau à celle de l'ethnicité, et un pays à une communauté ; alors, la fractionalisation indique dans

Communauté	Composition	$\Upsilon(C_i)$
C_1	un bureau (100 %)	0
C_2	deux bureaux (95 %, 5 %)	0.095
C_3	deux bureaux (75 %, 25 %)	0.375
C_4	deux bureaux (50 %, 50 %)	0.5
C_5	trois bureaux (50 %, 25 %, 25 %)	0.625
C_5	n bureaux $\left(\frac{100}{n}\ \%, \frac{100}{n}\ \%, \ldots\right)$	$1-\frac{1}{n}$

TABLEAU 5.6 – Exemples de valeurs de l'indice de fractionalisation selon la composition de la communauté. Une communauté parfaitement homogène à un indice de 0. Plus la communauté est hétérogène, en particulier plus les portions qui la composent sont petites, plus l'indice se rapproche de 1.

quelle mesure une communauté est homogène en regard de l'affiliation géographique de ses membres. Le tableau 5.6 illustre la relation entre la valeur de l'indice et la composition d'une communauté. La moyenne sur toutes les communautés

$$\bar{\Upsilon}(\mathscr{C}) = \frac{1}{N_{\mathscr{C}}} \sum_{i=1}^{N_{\mathscr{C}}} \Upsilon(C_i) = \frac{1}{N_{\mathscr{C}}} \sum_{i=1}^{N_{\mathscr{C}}} \left(1 - \sum_j s_{ij}^2\right), \tag{5.1}$$

où s_{ij} est la part du bureau j dans $C_i \in \mathscr{C}$, indique à quel point les communautés de la partition $\mathscr{C}$ (ou répartition) sont homogènes.

Ainsi, il existe une forte corrélation linéaire négative entre le nombre de communautés et la moyenne de l'indice de fractionalisation (-0,7067), c'est-à-dire que plus les modules sont nombreux, plus ceux-ci sont homogènes. De cela, il peut être compris que le niveau hiérarchique inférieur est composé de petites communautés très homogènes, c'est-à-dire dont les individus proviennent du même bureau ; tandis que ces dernières sont regroupées à l'intérieur d'entités bien moins définies au niveau hiérarchique supérieur. Il s'agit là d'une conclusion surprenante étant donné ce que l'on sait de la base de données : il existe une interprétation géographique des communautés dans le réseau de collaboration. Notons que l'indice n'est pas utilisé comme mesure qualitative de la solution, mais plutôt comme outil d'interprétation. Par exemple, la partition produite par **CSS/RAK** et représentée à la figure 5.11 a la meilleure valeur de $\bar{\Upsilon}(\mathscr{C})$ pour les modélisations en graphe étoilé et en graphe projeté, pourtant nous avions émis des réserves plus haut quant à la rigueur de la propagation d'étiquettes à cause de sa propension à former une supercommunauté, ici fortement hétérogène, et beaucoup de petites communautés, ici

fortement homogènes. Autre remarque, les structures modulaires avec recouvrement solutions d'**Infomap** sont bien plus hétérogènes que celle sans recouvrement malgré qu'elles soient plus morcelées. Il s'agirait d'une conséquence de l'existence de sous-groupes d'un ou de plusieurs individus qui, dans une répartition, influencent la valeur de la fractionalisation de plus d'une communauté hétérogène auxquelles ils appartiennent.

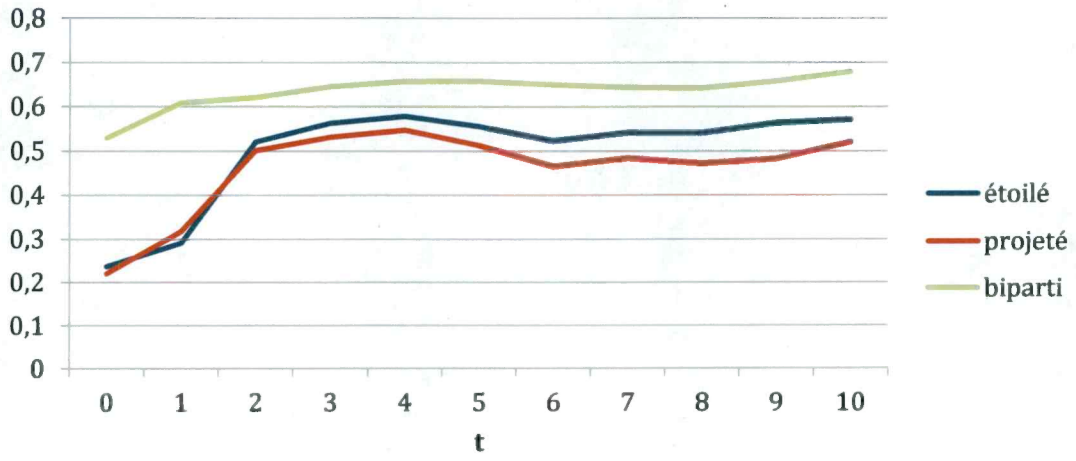
5.3.2 Cadre dynamique

L'analyse qui suit est basée sur les solutions produites par les méthodes de détection de communautés dynamiques dans les réseaux temporels **AFOCS**, **Louvain**, **Infomap** à deux niveaux, **OSLOM**, **OSLOM** avec retrait des sommets non-affiliés, telles que décrites dans les tableaux en annexe B.6, B.7, B.10, B.11 et B.12. Les remarques faites à la sous-section 5.3.1 peuvent s'appliquer à chacun des instantanés $t = \{0, \ldots, 10\}$ et ne sont donc pas répétées. Nous portons notre intérêt principalement à la dynamique des communautés.

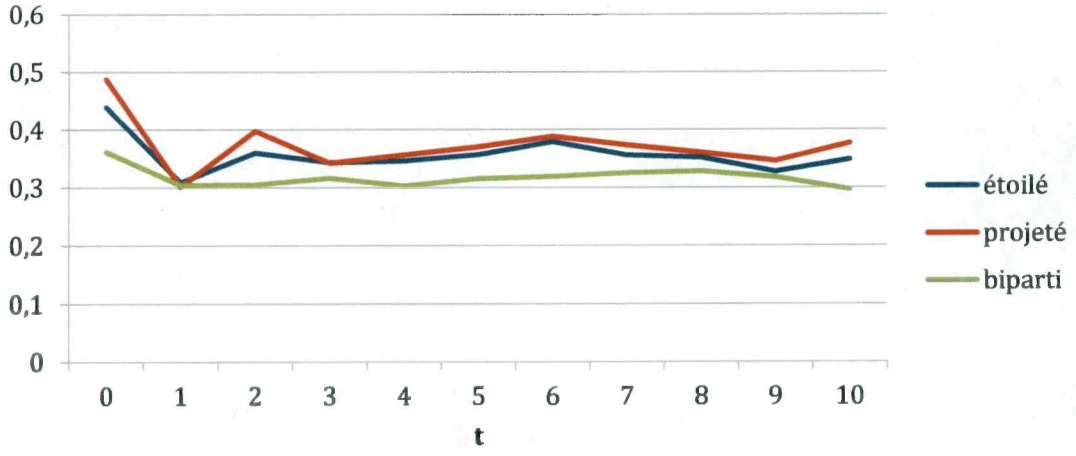
Malgré qu'il n'ait pas été possible de calculer une solution sur tout l'horizon temporel avec l'algorithme **AFOCS**, les structures modulaires des intervalles un et deux nous permettent de mettre en doute leur qualité, puisque leur modularité, près de zéro, et leur conductance, près de un, ne sont guère significatives. Selon quel critère pourrait-il s'agir de communautés valides si elles sont à la fois peu différentes de communautés aléatoires et peu séparées du reste du graphe ? L'algorithme consiste, dans un premier temps, à déterminer les modules de l'instantané initial, puis à apporter les changements de façon incrémentale. Ainsi est-il possible que la seconde partie de la procédure soit adéquate, cependant la phase initiale arrive à une solution peu convaincante selon la définition que nous faisons du concept de communauté. Faute de mieux que les mesures de qualité et en l'absence de vérité topologique, les solutions de **AFOCS** sont exclues de l'analyse.

D'une part, la structure modulaire du graphe temporel se précise fortement sur $[0,4]$: la valeur de la modularité augmente beaucoup, celle de la conductance moyenne - pour les algorithmes de partition uniquement - diminue faiblement. La structure semble plus stable sur $[6,10]$. Ainsi, les communautés sont, avec le temps, plus nombreuses, plus différentes d'un graphe aléatoire et légèrement mieux définies du reste du réseau, comme représenté aux figures 5.12a et 5.12b. Cette affir-

FIGURE 5.11 – Communautés détectées par l'algorithme **CSS/RAK** dans le graphe étoilé statique, soit avec les données de l'ensemble de l'horizon temporel de 1974 jours. Les couleurs indiquent l'affiliation à l'un des 46 bureaux. À l'exception de la grande communauté mal définie (au centre), les communautés présentent une forte homogénéité en termes d'appartenance à un bureau. La visualisation est produite avec les outils de Traud et al. [145] pour MATLAB® modifiant les algorithmes basées sur les forces (*Force-based* ou *Force-directed algorithms*) de Fruchtermann et Reingold [146] et Kamada et Kawai[147] et disponibles à l'adresse `http://netwiki.amath.unc.edu/VisComms/VisComms`.



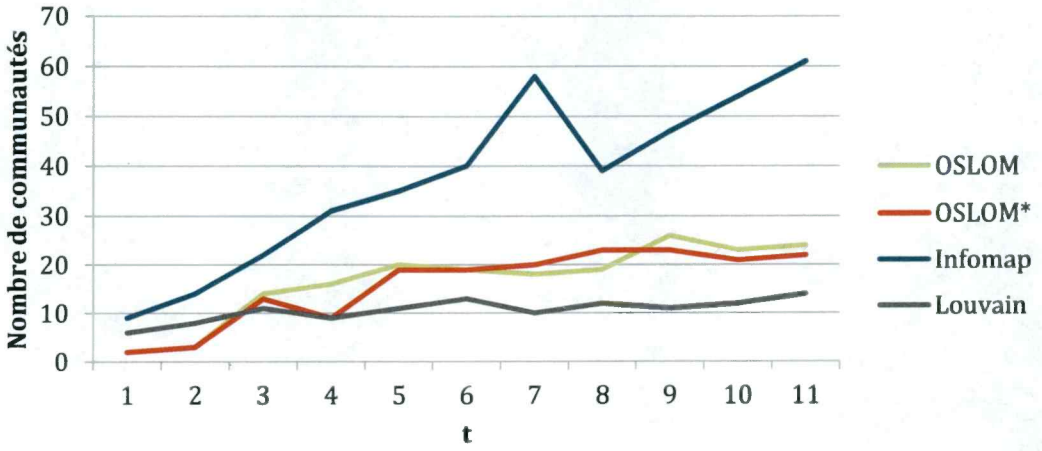
(a) Modularité.



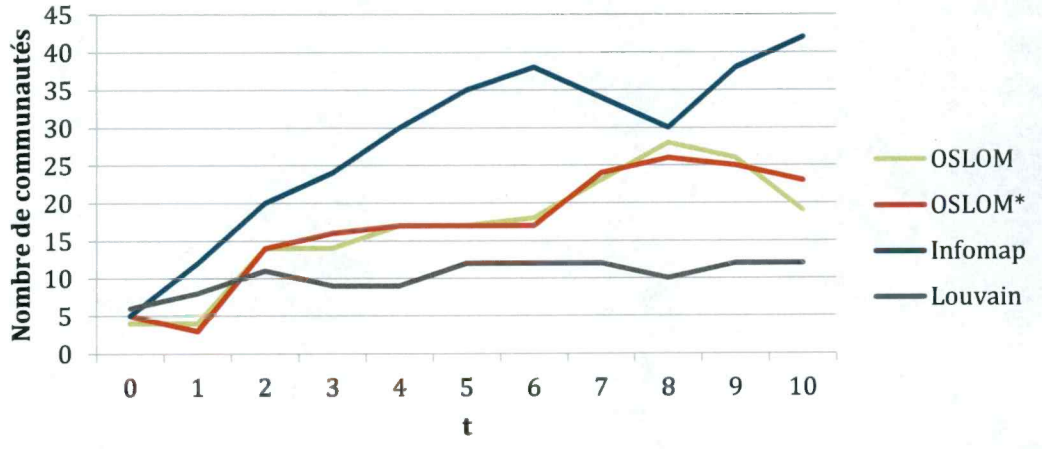
(b) Conductance moyenne.

FIGURE 5.12 – Évolution de la moyenne des mesures de qualité sur tous les algorithmes (algorithmes de partition seulement dans le cas de la conductance) des solutions dans chacun des instantanés $G^{(t)}, t = \{0, \ldots, 10\}$.

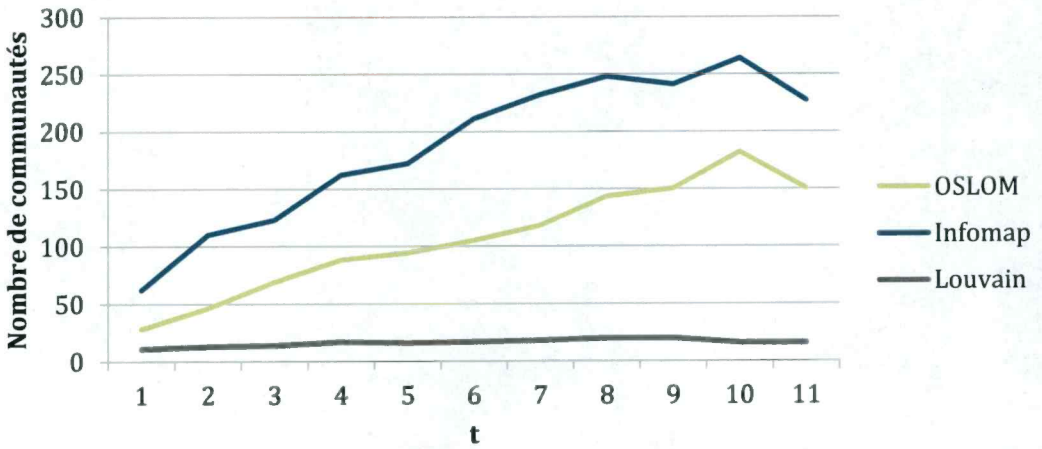
mation généralisée à tous les algorithmes et à chacune des modélisations indique qu'il s'agit bien d'une propriété du réseau évolutif étudié. Par ailleurs, si le nombre de communautés augmente, d'après tous les algorithmes étudiés comme illustré à la figure 5.13, il n'augmente pas à la même vitesse. Par conséquent, comme illustré à la figure 5.14, les communautés de **Louvain** sont en moyenne de plus en plus grandes, celles de **Infomap** et **OSLOM** de tailles constantes - autre conséquence de la distribution de la taille des communautés selon la méthode de détection, donc conséquence du choix de l'algorithme. **OSLOM** présente beaucoup plus de variabilité cependant. Plus étonnant encore, selon 5.13b, **Louvain** et **OSLOM** dépeignent dans le graphe projeté sur $[6,8]$ deux situations presque symétriquement opposées :



(a) Graphe étoilé

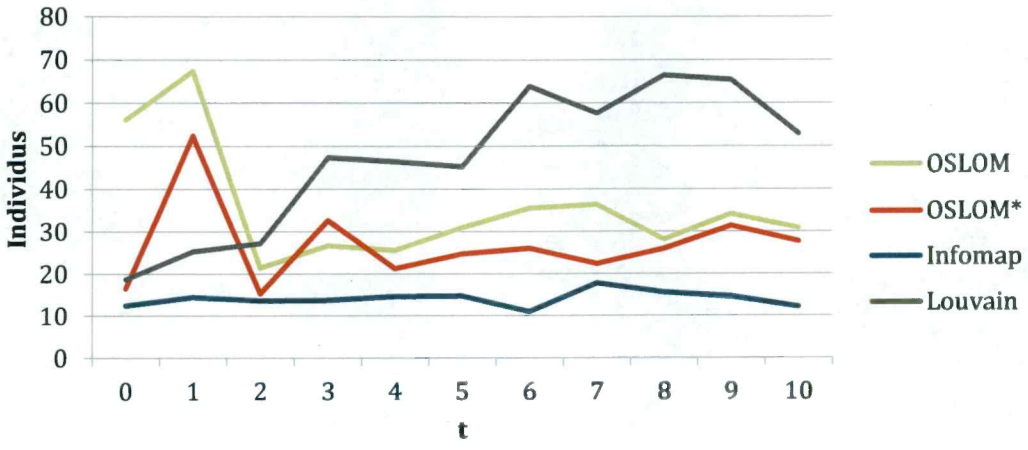


(b) Graphe projeté

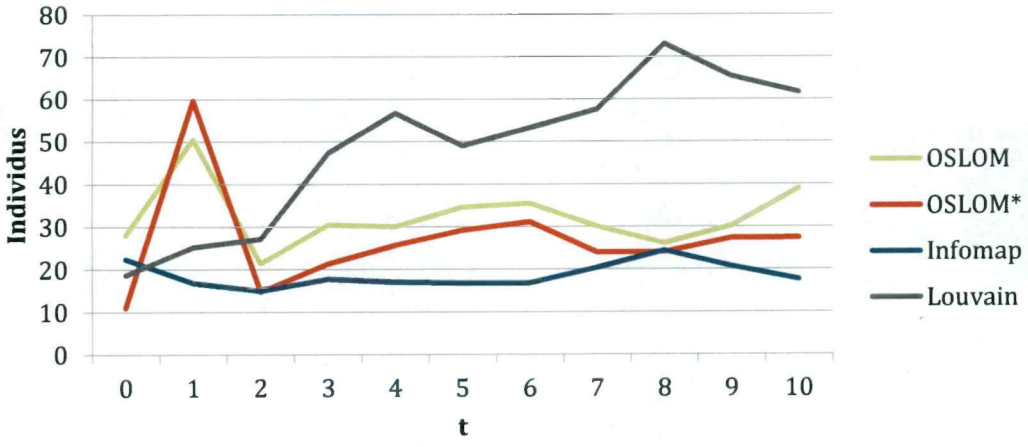


(c) Graphe biparti

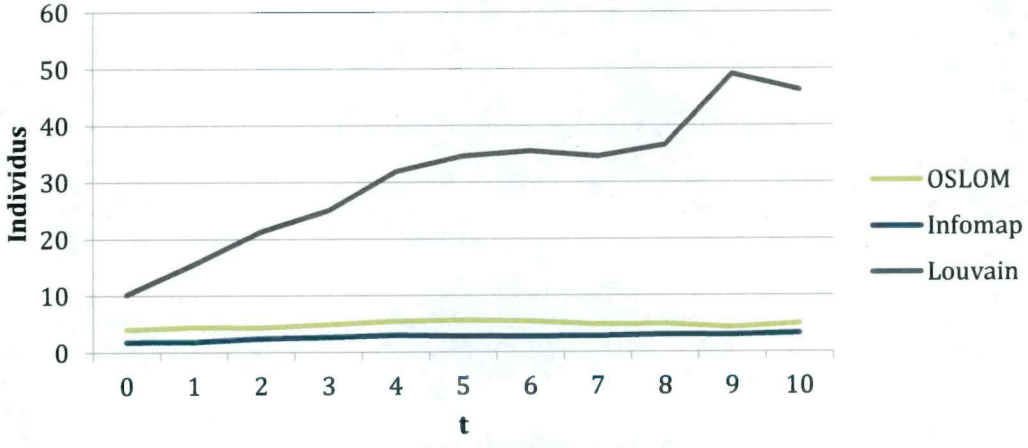
FIGURE 5.13 – Évolution du nombre de communautés dans les instantanés $G^{(t)}, t = \{0, \ldots, 10\}$ détectées dans les graphes. Notons la croissance plus ou moins rapide selon l'algorithme choisi. *Version d'**OSLOM** dans laquelle les sommets non assignés à une communauté sont retirés.



(a) Graphe étoilé



(b) Graphe projeté



(c) Graphe biparti

FIGURE 5.14 – Évolution de la taille moyenne des communautés (individus seulement) dans les instantanés $G^{(t)}, t = \{0, \ldots, 10\}$ détectées dans les graphes. *Version d'**OSLOM** dans laquelle les sommets non assignés à une communauté sont retirés.

d'après l'un, le nombre de modules chute ; d'après l'autre, il grimpe. Ceci implique deux interprétations contradictoires. Cette fois encore, l'algorithme et la modélisation, et non la structure sous-jacente du réseau, influencent les résultats obtenus puisque ces inflexions symétriques ne sont pas observées aussi clairement dans les autres graphes. Pour revenir au creux du nombre de sommets retirés par **OSLOM** en $t = 1$ (voir 5.8, page 85), nous observons une chute correspondante de la conductance et de la modularité (B.10 et B.11 en annexe) et une hausse soudaine de la taille des communautés. En fait, la structure communautaire détectée par l'algorithme n'est pas significative sur l'intervalle $[0,2)$ d'après la valeur de la modularité, cependant **OSLOM** n'est pas basé sur l'optimisation de cette mesure. Aussi, la conclusion tirée de 5.8 à $t = 1$ était plutôt qu'à l'inverse, la structure est robuste sur une grande partie du graphe. Ainsi, les différentes mesures amènent à des contradictions apparentes.

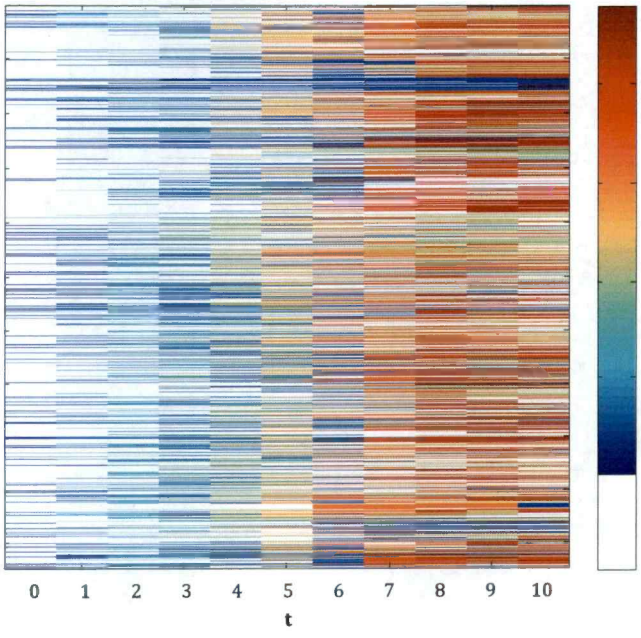
De la même façon qu'il a été possible de l'observer dans le contexte statique, la structure modulaire dynamique présente une certaine homogénéité en termes d'origine géographique : certains modules sont composés d'individus provenant majoritairement de un ou deux bureaux ; d'autres, beaucoup moins clairement interprétables, sont mal définis, regroupant ainsi des sous-ensembles d'employés d'un grand nombre d'emplacements. L'organisation s'homogénéise pour les solutions proposées par **Infomap** dans le graphe étoilé et le graphe projeté, conséquence directe de l'augmentation plus rapide du nombre de communautés. En effet, nous avons à la section précédente démontré la relation entre le nombre de communautés, indirectement leur cardinal, et la valeur de la fractionalisation. Dans tous les autres cas, la mesure est stable.

D'autre part, il est possible de voir l'apparition d'entités précises à un instant de l'horizon temporel ainsi que représenté à la figure 5.6 page 83 : la communauté en rose apparait à $t = 6$ et perdure presque intégralement jusqu'à la fin de l'horizon temporel ; la communauté en bleu clair apparait en $t = 2$, croît jusqu'en $t = 5$ puis décroît jusqu'en $t = 10$. Bien d'autres événements comme la mort, la résurgence, la croissance ou la décroissance de communautés peuvent être déduits de cette représentation visuelle, mais en faire la liste n'est pas le but de ce mémoire. Notons cependant que chaque algorithme raconte une histoire et bien que sa trame narrative soit la même pour tous - il s'agirait de la vérité inhérente à la base de données - ses détails peuvent varier grandement. En particulier, certaines communautés plutôt

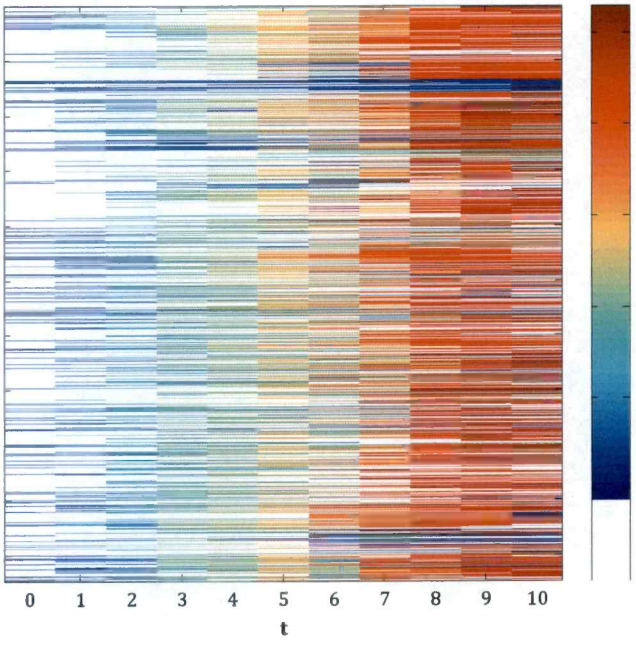
homogènes et avec une interprétation géographique évidente sont détectées systématiquement par tous les algorithmes dans toutes les modélisations et suivies sur une certaine période, cependant leur composition n'est pas strictement identique d'une solution à l'autre.

Nous sommes aussi confrontée à la difficulté de représenter statiquement des larges modèles animés, d'autant plus qu'il n'existe aucun outil capable de traiter graphiquement les réseaux évolutifs autrement qu'en juxtaposant des instantanés de graphes, comme à la figure 5.6. À cet effet, les figures 5.15 décrivent l'appartenance aux communautés dynamiques de chaque employé. L'appariement des modules d'un instantané à l'autre est calculé avec l'indice de similitude de Takaffoli ou celui de Jaccard avec des valeurs de θ optimisant la correspondance intertemporelle et minimisant le nombre total de communautés uniques (une communauté qui apparait sur plus d'une période n'est comptabilisée qu'une seule fois). Seule la représentation en couleurs des solutions de l'algorithme **Louvain** dans le graphe projeté est compréhensible puisqu'il s'agit de la méthode amenant au plus petit nombre total de communautés. Les autres algorithmes ont tant de modules distincts, donc autant de couleurs différentes, que leur représentation est illisible. De la figure 5.15, nous constatons que peu d'individus sont fidèles à une même communauté sur tout l'horizon temporel probablement en conséquence des fortes modifications du jeu de données d'un instantané à l'autre et du caractère aléatoire de **Louvain**. Observons aussi de grandes perturbations dans l'affiliation des individus sur l'intervalle $t = 5$ traduisant l'apparition et la disparition soudaine d'un plus grand nombre de modules, fait confirmé par les autres algorithmes en termes de proportion de naissances et de morts de communautés. Ainsi, au milieu de l'intervalle se sont produites de si fortes modifications dans le réseau que peu de communautés ont survécu telles quelles. Autre point intéressant, seules quelques communautés survivent sur une plus ou moins longue période de deux à onze intervalles (selon les valeurs établies pour θ) et une majorité d'entre elles n'existe que sur un seul instantané, de sorte que la distribution de la durée de vie des communautés suit encore une fois plausiblement une loi de puissance (ou encore une loi de Pareto, voir B.1 en annexe).

Il est difficile d'arriver à généraliser le portrait de l'évolution des modules dans un réseau évolutif comme l'ont fait Palla et al. [21], notant entre autres que 1) les plus larges sont en moyenne plus âgés, 2) les plus larges persistent lorsqu'ils sont en mesure de renouveler leurs membres, 3) les plus petits persistent lorsque leur com-

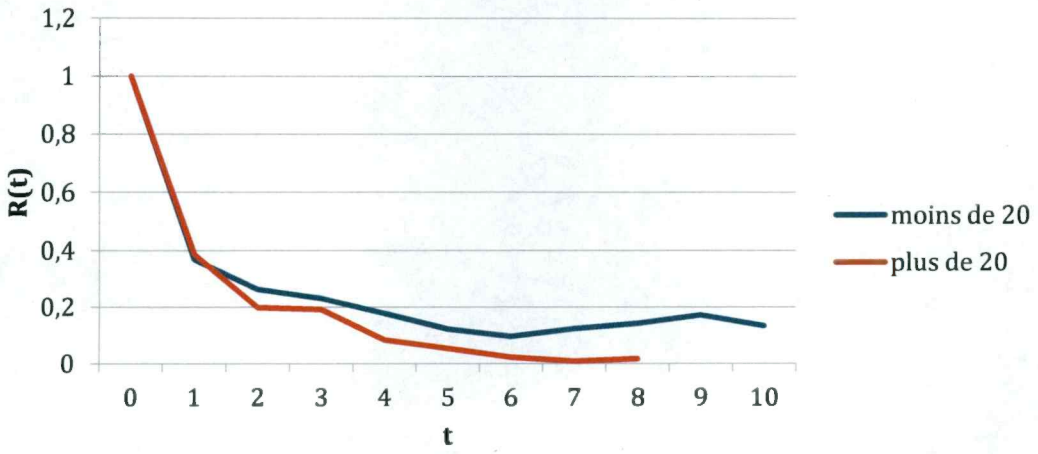


(a) Évolution des sommets dont la correspondance est établie avec l'indice de Jaccard pour $\theta = 0,25$.

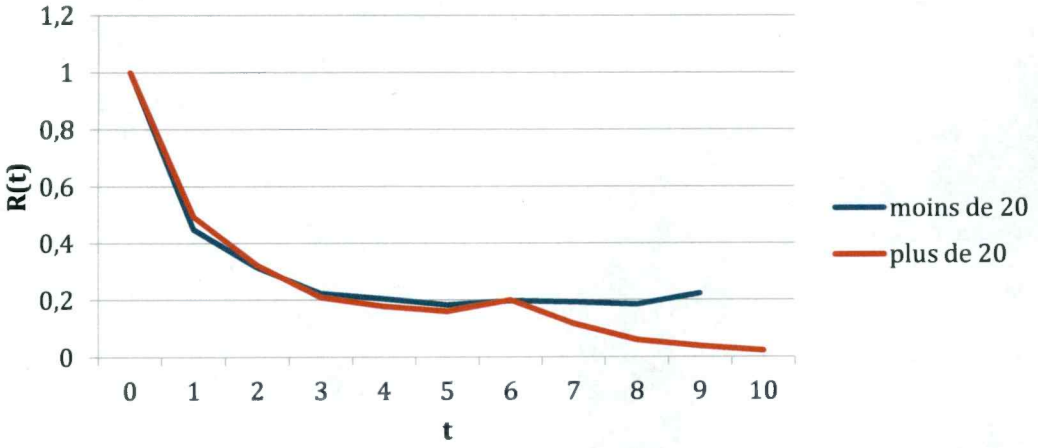


(b) Évolution des sommets dont la correspondance est établie avec la similitude de Takaffoli et al. pour $\theta = 0,35$.

FIGURE 5.15 – Évolution de l'appartenance des sommets de la solution de l'algorithme **Louvain** indépendamment sur les instantanés $G^{(t)}, t \in \{0,\ldots,10\}$ dans le graphe projeté. Les couleurs indiquent l'assignation à une certaine communauté et les sommets en abscisse sont dans le même ordre dans les deux figures a et b. Dans les deux cas, l'historique est assez similaire et peu de sommets sont fidèles à une même communauté sur tout l'horizon temporel.



(a) Solution du graphe étoilé avec l'algorithme **Louvain** et appariement par l'indice de Jaccard, $\theta = 0,25$.



(b) Solution du graphe projeté avec l'algorithme **OSLOM** et appariement par l'indice de Jaccard, $\theta = 0,20$.

FIGURE 5.16 – Fonction d'autocorrélation $R(t)$ moyenne des communautés inter-temporelles de plus ou moins 20 initialement. La fonction décroît légèrement moins rapidement dans le cas de plus petits modules.

position est stable. Aucune des modélisations, ni aucun des algorithmes ne suggère la conclusion 1). Les affirmations 2) et 3) sont démontrées par Palla et al. avec la fonction d'autocorrélation

$$R(t) = \frac{|C^{(t_0)} \cap C^{(t_0+t)}|}{|C^{(t_0)} \cup C^{(t_0+t)}|}, \quad t \in \{1, \ldots, 10\}$$

calculant la similitude entre l'état initial et l'état à t d'une même communauté. Le portrait de la base de données étudiée est beaucoup moins définitif ou concluant.

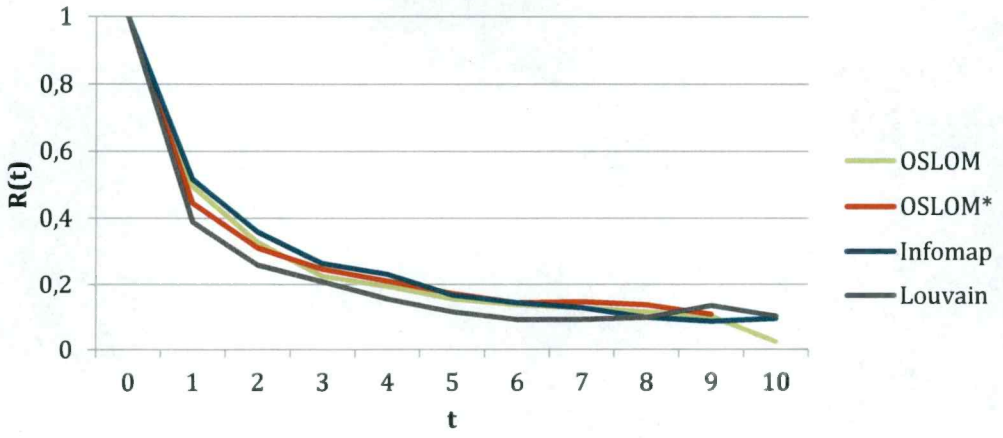


FIGURE 5.17 – Fonctions d'autocorrélation moyennes $R(t)$ des communautés dynamiques appariées d'un instantané à l'autre par l'indice de Jaccard avec $\theta = 0,20$. La fonction pour **Infomap** décroît moins rapidement, les modules sont plus stables sur l'horizon temporel.

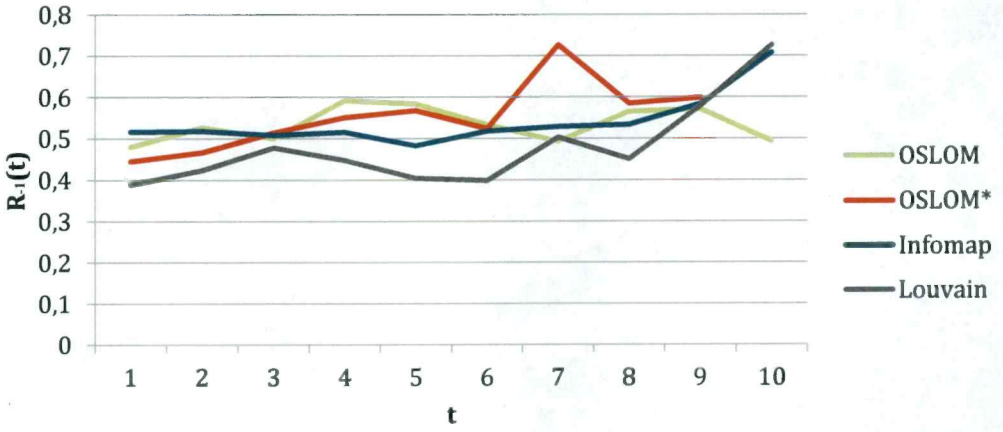


FIGURE 5.18 – Fonctions d'autocorrélation moyennes sur une période $R_{-1}(t)$ des communautés dynamiques appariées d'un instantané à l'autre par l'indice de Jaccard avec $\theta = 0,20$. La fonction pour **Louvain** est dominée par les autres sur $[1,9]$, i.e. les modules intertemporels calculés par l'algorithme sont en moyenne plus dissemblables d'un instantané à l'autre que ceux déterminés par les autres algorithmes.

La figure 5.16 montre la fonction d'autocorrélation moyenne de communautés nées avec plus ou moins de 20 sommets pour deux des solutions laquelle décroît légèrement plus rapidement lorsque les modules sont plus grands, donc démontre que les grandes communautés sont quelque peu moins stables. Il n'en demeure pas moins que les plus petites n'ont pas une composition stable pour autant. Outre cela, la fonction d'autocorrélation des communautés intertemporelles déterminées avec **OSLOM** ne décroît pas moins rapidement que celle de **Infomap** ou **Louvain** mal-

gré qu'elle soit au final la seule méthode mettant à profit l'information temporelle à $t-1$ dans le calcul de la solution à t. Il aurait été légitime de s'attendre à plus de stabilité de sa part, c'est cependant **Infomap** qui l'est le plus, **Louvain** le moins. À cet effet, les figures 5.17 et 5.18 présentent respectivement les fonctions d'autocorrélation moyennes et les fonctions d'autocorrélation moyennes sur une période

$$R_{-1}(t) = \frac{|C^{(t-1)} \cap C^{(t)}|}{|C^{(t-1)} \cup C^{(t)}|}, \quad t \in \{1, \ldots, 10\},$$

calculant la similitude entre les états successifs d'une même communauté. L'idée est de représenter à quel point la structure moyenne des communautés dynamiques est modifiée d'une part avec le temps et d'autre part d'un instantané à l'autre. De ces graphes, nous concluons que la composition des communautés après plusieurs périodes est très différente de celle originale à cause de l'ajout ou du renouvellement des membres ; en particulier, c'est au début que les changements sont les plus radicaux. Les fonctions d'autocorrélation moyennes sur une période à la figure 5.18 semblent indiquer de façon unanime une légère tendance à la hausse, c'est-à-dire que la correspondance entre les communautés entre deux instantanés est légèrement plus forte avec le temps, ou encore que la structure modulaire sous-jacente est légèrement plus stable avec le temps. Aussi, les communautés découvertes par **Louvain** étant en moyenne de plus grande taille et de taille croissante, leur composition est plus dissemblable d'un intervalle de temps à l'autre comme le démontre sa fonction d'autocorrélation moyenne sur une période. Alors l'algorithme choisi est encore un facteur influent dans l'interprétation des résultats.

Il n'a pas été possible de faire de lien entre la durée de vie d'un module ou sa prédisposition à se désintégrer et sa taille, pas plus qu'avec sa densité *intra* ou *inter*-communautés 3.1.1, son **expansion** (3.6), son degré interne (2.2) ou externe (2.3) ou quelque autre mesure de cohésion ; non pas que ces liens n'existeraient pas en réalité, ils ne sont cependant exprimés par les solutions des algorithmes testés.

Pour finir, nous avons dans toutes ces observations la confirmation qu'il est extrêmement risqué de procéder à l'analyse des communautés d'un réseau en faisant au préalable le choix arbitraire d'une modélisation et d'un algorithme de détection : il serait possible de tirer des conclusions erronées ou de découvrir des comportements dans les faits inexistants. Et même avec la multiplication des approches, seul un portrait vague se dégage des données.

Chapitre 6

Conclusion

6.1 Résumé

Ce travail ambitieux de recherche et d'analyse avait pour objectif de faire le tri dans la foisonnante et éclectique littérature scientifique au sujet de la détection des communautés dynamiques dans les réseaux évolutifs et d'en évaluer l'applicabilité à une situation concrète. Plusieurs variations de la modélisation des données en graphe ont été considérées par soucis de complétude, mais aussi parce qu'en l'absence de vérité topologique, seule la multiplication des tentatives sous différents paramètres peut espérer s'approcher de la structure sous-jacente au réseau. Nous avons attaqué la version presque la plus complète d'un problème déjà complexe ; soit non seulement de trouver la structure modulaire, mais celle avec recouvrement dans un graphe pondéré et évolutif. Nombre de méthodes, n'en déplaisent à leurs auteurs, se sont essoufflées devant la tâche à accomplir, ne laissant au final que quelques algorithmes possibles (**A^{3}CS**, **AFOCS**, **CFinder**, **COPRA**, **CSS**, **FacetNet**, **iLCD**, **Infomap**, **LabelRankT**, **Louvain**, **OSLOM**, **PDEC**, **RAK**, **SIM**) voire adéquats (**COPRA**, **CSS**, **Infomap**, **LabelRankT**, **Louvain**, **OSLOM**, **RAK**, **SIM**). Des analyses abordant les angles quantitatifs et descriptifs ont été proposées, à la différence de la majorité des auteurs qui se concentrent sur la validation numérique des solutions dans des graphes générés ou dont la structure est connue. Aucun algorithme n'est ressorti clairement supérieur aux autres, mais l'étude de leurs solutions a mené à bon nombre d'observations intéressantes.

Ainsi, il s'est avéré impossible d'ignorer le partitionnement d'un réseau en contexte statique, problématique bien plus achevée que celle qui nous intéressait *a priori*, d'autant plus que la détection indépendante des communautés sur les instantanés du graphe produit des solutions somme toute raisonnables. En fait, il n'est pas évident que l'apport d'information temporelle soit utile dans un réseau avec d'aussi amples modifications d'un intervalle à l'autre comme le témoigne la faible stabilité des solutions informées sur des instantanés avec **OSLOM**. Une association passée n'est peut-être pas garante de l'association présente lorsque les interactions sont très fréquentes et très variables, ou encore l'information actuelle suffit à classer les sommets convenablement.

En outre, évaluer objectivement les solutions de divers algorithmes demeure l'enjeu principal puisqu'aucun critère n'est universellement applicable. Une sorte de consensus ou de portrait général se dessine dans la chronologie des événements ou dans l'agencement des individus dans un groupe ou encore dans les niveaux hiérarchiques des modules. Cependant, il subsiste une grande incertitude en partie à cause de la composante aléatoire des algorithmes et des inconsistances voire des contradictions entre les solutions de différentes méthodes.

Par ailleurs, le profil de la structure modulaire - en particulier le nombre de communautés, la distribution de leur taille ou leur homogénéité - résulte de décisions prises en amont comme le seraient le choix de la modélisation et, surtout, de l'algorithme. Ainsi, on peine à différencier ce qui émane des données de ce qui procède de la méthode.

En définitive, il semble plutôt hasardeux d'entreprendre une quelconque interprétation basée sur les résultats d'une seule méthode et qu'il vaille mieux faire preuve de prudence quitte à en faire plus que moins.

6.2 Suggestion de travaux futurs

Plusieurs points ont été soulevés tout au long de ce mémoire et mériteraient plus d'attention : certains en lien avec la base de données, d'autres en lien avec les algorithmes. Avant tout, il importerait de valider certaines des observations sur d'autres bases de données par l'analyse qualitative de la composition des solutions, soit au-delà des mesures quantitatives (modularité, IMN, etc.) habituellement privilégiées

par les auteurs pour garantir la qualité d'un algorithme.

Une autre avenue intéressante serait l'amélioration des modèles aléatoires et des modèles générés en contexte dynamique. Les modèles aléatoires sont utilisés par plusieurs algorithmes, leur performance serait peut-être améliorée si la répartition des liens ou le degré des sommets représentaient plus fidèlement un réseau naturel (mais sans communautés). Les modèles générés dynamiques utilisés pour tester des algorithmes devraient, quant à eux, exhiber de réels comportements évolutifs comme la croissance de leur taille, la densification des liens, l'évolution des communautés, l'existence de module de tailles suivant une certaine distribution, etc.

De plus, il y a encore de la place pour de nouvelles méthodes de détection de communautés. Nous croyons qu'un algorithme complètement autonome et libre de paramètre n'est pas la voie à privilégier, que l'utilisateur doit nécessairement apporter sa connaissance du réseau à la machine (ajout de métadonnées dans la procédure de détection de communautés, par exemple), que tout paramètre doit être interprétable et compréhensible, que tout algorithme doit démontrer son efficacité sur des réseaux réels non pas uniquement sur des graphes générés.

Finalement, la question demeure de savoir que faire des communautés ainsi détectées. La réponse passe nécessairement par la compréhension empirique des rouages *inter*-communautés ou des rôles *intra*-communautés.

L'engouement pour le sujet ne se tarit pas, nous espérons cependant avoir apporté quelques nuances permettant d'éclairer de nouveaux chemins à prendre.

Annexe A

Appariement des communautés

Algorithme 2 Couplage

Entrée : $\mathscr{C} = \{C_1, \ldots, C_n\}, \mathscr{D} = \{D_1, \ldots, D_k\}, \theta$

$\mathscr{P} = \emptyset$, vecteur de paires

pour tout $C_i \in \mathscr{C}$ **faire**

$\mathscr{L}_i$ est une liste de préférence ordonnée selon la valeur de la similitude s entre C_i et tout élément de $\mathscr{D}$ avec $s(C_i, D_j) \geq \theta, \forall j$; les égalités sont brisées aléatoirement

pour tout $D_j \in \mathscr{D}$ **faire**

$\mathscr{M}_j$ est une liste de préférence ordonnée selon la valeur de la similitude entre D_j et tout élément de $\mathscr{C}$ avec $s(C_i, D_j) \geq \theta, \forall i$; les égalités sont brisées aléatoirement

tout $C_i \in \mathscr{C}$ et $D_j \in \mathscr{D}$ tel que $|\mathscr{L}_i| > 0$ ou $|\mathscr{M}_j| > 0$ est *libre*

tant que il existe $C_i \in \mathscr{C}$ t.q. C_i est *libre* et $|\mathscr{L}_i| > 0$ **faire**

D_j le premier élément de $\mathscr{L}_i$

$\mathscr{L}_i \leftarrow \mathscr{L}_i \backslash \{D_j\}$

si D_j est *libre* **alors**

$\mathscr{P} \leftarrow \mathscr{P} \cup \{(C_i, D_j)\}$

sinon

si D_j préfère C_i à C_k, $C_i, C_k \in \mathscr{M}_j$ **alors**

$\mathscr{P} \leftarrow \mathscr{P} \cup \{(C_i, D_j)\} \backslash \{(C_k, D_j)\}$

C_k est *libre*

Sortie : $\mathscr{P}$ contient les couples

Algorithme 3 Chronologie de la structure modulaire

Entrée : $\mathscr{C} = \{\mathscr{C}^{(0)}, \ldots, \mathscr{C}^{(t_{max})}\}, \theta$

$\mathscr{P}, \mathscr{Q} = \emptyset$, vecteurs de paires

$\mathscr{H} = \{\{C_1^{(0)}\}, \ldots, \{C_i^{(t)}\}, \ldots\}$, ensembles disjoints (*disjoint-set structure*) des communautés de chaque intervalle

pour tout $t \in \{0, \ldots, t_{max} - 1\}$ **faire**

$\mathscr{P} \leftarrow \mathscr{P} \cup \{\text{Couplage}(\mathscr{C}^{(t)}, \mathscr{C}^{(t+1)}, \theta)\}$

pour tout $s \in \{t+2, \ldots, t_{max}\}$ **faire**

$\mathscr{Q} \leftarrow \mathscr{Q} \cup \{\text{Couplage}(\mathscr{C}^{(t)}, \mathscr{C}^{(s)}, \theta)\}$

pour tout $t \in \{0, \ldots, t_{max}\}$ **faire**

pour tout $C_i^{(t)} \in \mathscr{C}^{(t)}$ **faire**

si $\exists C_j^{(t+1)}$ t.q. $(C_i^{(t)}, C_j^{(t+1)}) \in \mathscr{P}, t < t_{max}$ **alors**

si il existe une liste de $\mathscr{H}$ dont $C_i^{(t)}$ est l'extrémité fin **alors**

fusionner $\{C_j^{(t+1)}\} \in \mathscr{H}$ à cette liste

sinon

fusionner $\{C_i^{(t)}\}$ à $\{C_j^{(t+1)}\}$

si $|C_i^{(t)}| < |C_j^{(t+1)}|$ **alors**

$C_i^{(t)} \rightarrow$ croissance à $t+1$

sinon si $|C_i^{(t)}| > |C_j^{(t+1)}|$ **alors**

$C_i^{(t)} \rightarrow$ décroissance à $t+1$

sinon $C_i^{(t)} \rightarrow$ continuité à $t+1$

si $\exists k > 1$ t.q. $\bigcup_{j=1}^{k} C_j^{(t+1)} \cap C_i^{(t)} \neq \emptyset, t < t_{max}$ **alors**

$C_i^{(t)} \rightarrow$ division à $t+1$

si $\exists k > 1$ t.q. $\bigcup_{j=1}^{k} C_j^{(t-1)} \cap C_i^{(t)} \neq \emptyset, t > 1$ **alors**

$C_i^{(t)} \rightarrow$ fusion à t

si $C_j^{(t+1)} \cap C_i^{(t)} = \emptyset, \forall j, t < t_{max}$ **alors**

$C_i^{(t)} \rightarrow$ mort à $t+1$

si $C_j^{(t-1)} \cap C_i^{(t)} = \emptyset, \forall j, t > 1$ **alors**

$C_i^{(t)} \rightarrow$ naissance à t

ordonner $\mathscr{Q}$

pour tout $(C_i^{(t)}, C_j^{(s)}) \in \mathscr{Q}$ **faire**

si il existe une liste $h_i \in \mathscr{H}$ dont $C_i^{(t)}$ est l'extrémité fin et $h_j \in \mathscr{H}$ dont $C_j^{(s)}$ est l'extrémité de départ **alors**

fusionner h_i à h_j

Sortie : $\mathscr{H}$ contient les communautés intertemporelles

Les algorithmes 2 et 3 retournent la chronologie de la structure modulaire, déterminée par une quelconque méthode, d'un graphe évolutif $\mathcal{G} = (G^{(0)}, \ldots, G^{(t_{max})})$. Dans 2, le couplage des communautés détectées dans l'instantané t à celles de $t+i, 1 \leq i \leq t_{max}$ est équivalent au problème de «mariages stables avec égalités et liste partielles» [1] (Gale et Shapley [149], Irving et al. [150], Manlove et al. [151], etc.) où la préférence d'une communauté pour une autre est calculé avec la valeur d'un quelconque indice d'appariement noté $s(C_j^{(t)}, C_k^{(t+i)})$ (indice de Jaccard ou de similitude de Takaffoli et al. ou tout autre mesure). Alors, un module de t peut être appairé à un seul module de $t+i, 1 \leq i \leq t_{max}$, auquel cas, il s'agira d'incarnations temporelles de la même entité ; et il est possible qu'un module à t ait la même préférence pour plus d'un module à $t+i, 1 \leq i \leq t_{max}$ (valeurs égales de s).

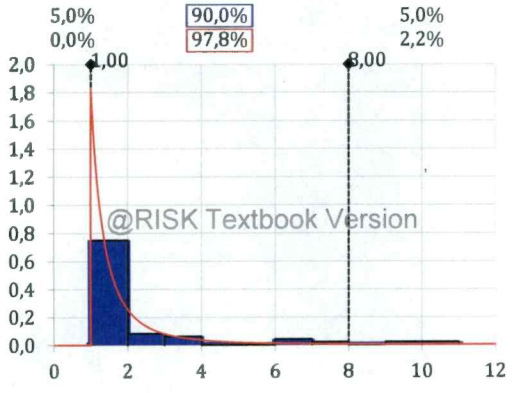
Ce problème mériterait plus d'attention, mais cela dépasse le cadre de ce mémoire. Notons que les ensembles à coupler sont ici assez petits et que nous ne cherchons pas nécessairement une solution *optimale*, mais une solution *possible*. Il demeure une part de subjectivité dans l'interprétation de l'évolution de communautés dynamiques, en particulier lorsque la correspondance est faible ou présentant des égalités ; et, pour cela, nous nous satisfaisons des solutions proposées par l'algorithme tel quel.

L'algorithme 3 retrace la chronologie de la structure modulaire par les différents événements concernant les communautés, soit *mort, naissance, fusion, division, croissance, décroissance, résurgence*, cette dernière par la correspondance entre des communautés séparées par plus d'un pas de temps et n'ayant pas d'incarnation entre les deux.

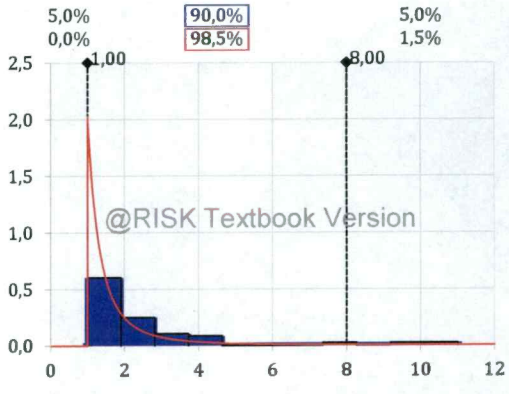
1. *Stable marriage with incomplete lists and ties*, SMTI

Annexe B

Figures et tableaux

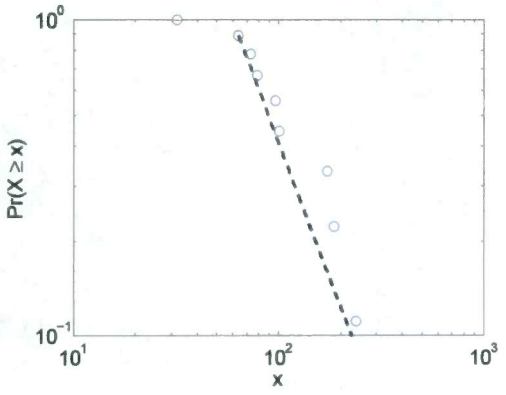


(a) Correspondance établie avec l'indice de Jaccard pour $\theta = 0,25$.

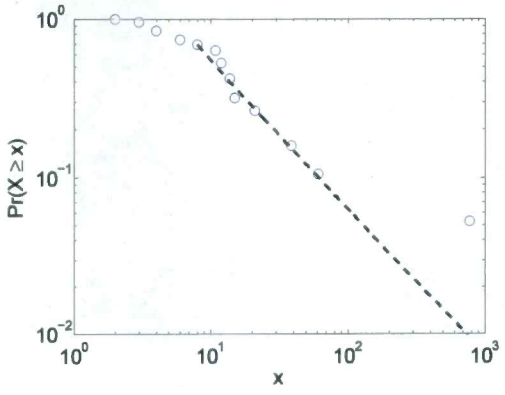


(b) Correspondance établie avec la similitude de Takaffoli et al. pour $\theta = 0,35$.

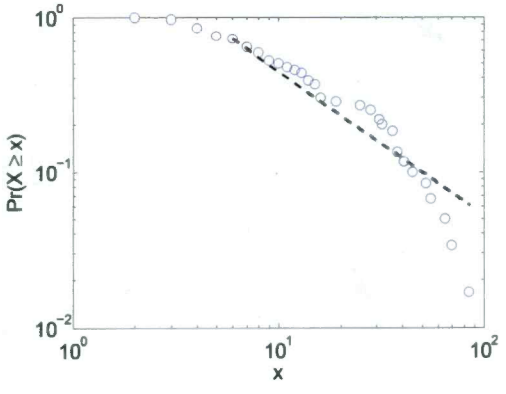
FIGURE B.1 – Distributions des durées de survie des communautés dynamiques détectées avec l'algorithme **Louvain** indépendamment sur les instantanés $G^{(t)}, t \in \{0,\ldots,10\}$ dans le graphe projeté. En rouge, une courbe de distribution possible : (a), (b) Pareto.



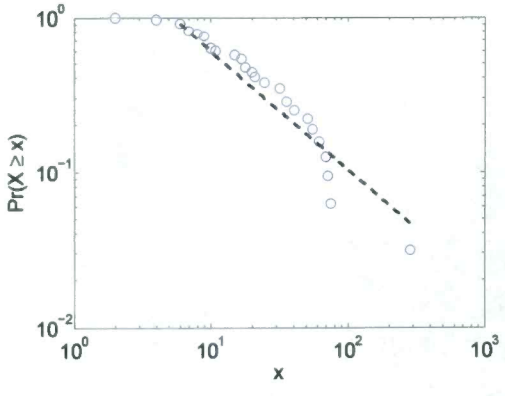
(a) **CSS/SIM**, $\alpha = 2,72$, valeur-$p = 0,665$



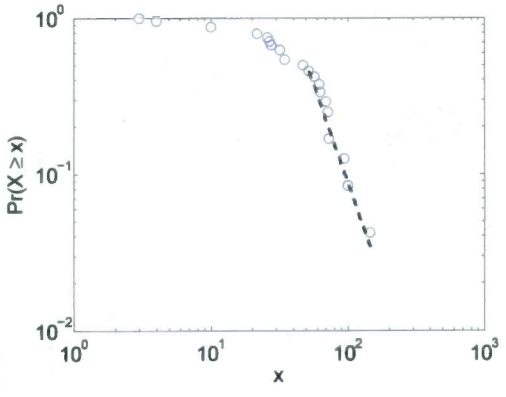
(b) **CSS/RAK**, $\alpha = 1,92$, valeur-$p = 0,083$



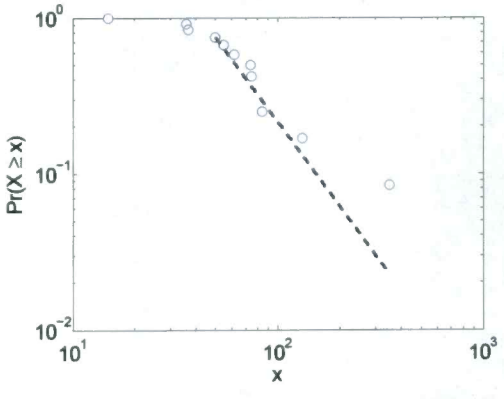
(c) **CSS/Louvain**, $\alpha = 1,9$, valeur-$p = 0,007$



(d) **Infomap**, $\alpha = 1,75$, valeur-$p = 0,033$



(e) **OSLOM**, $\alpha = 3,5$, valeur-$p = 0,478$



(f) **Louvain**, $\alpha = 2,79$, valeur-$p = 0,667$

FIGURE B.2 – Distributions cumulatives $P(x)$ et leur ajustement par maximisation de la vraisemblance à une loi de puissance de x la taille des communautés détectées par différents algorithmes dans le graphe projeté (méthodologie de [133]). Les ajustements et visualisations sont réalisés avec les fonctions **plift.m** et **plplot.m** pour MATLAB® par Clauset disponibles à l'adresse `http://tuvalu.santafe.edu/~aaronc/powerlaws/`. Le test d'ajustement de seuil 0,1 conclu que pour **CSS/RAK**, **Infomap** et **CSS/Louvain** la distribution de puissance n'est pas un modèle plausible (voir les valeurs-p ci-haut).

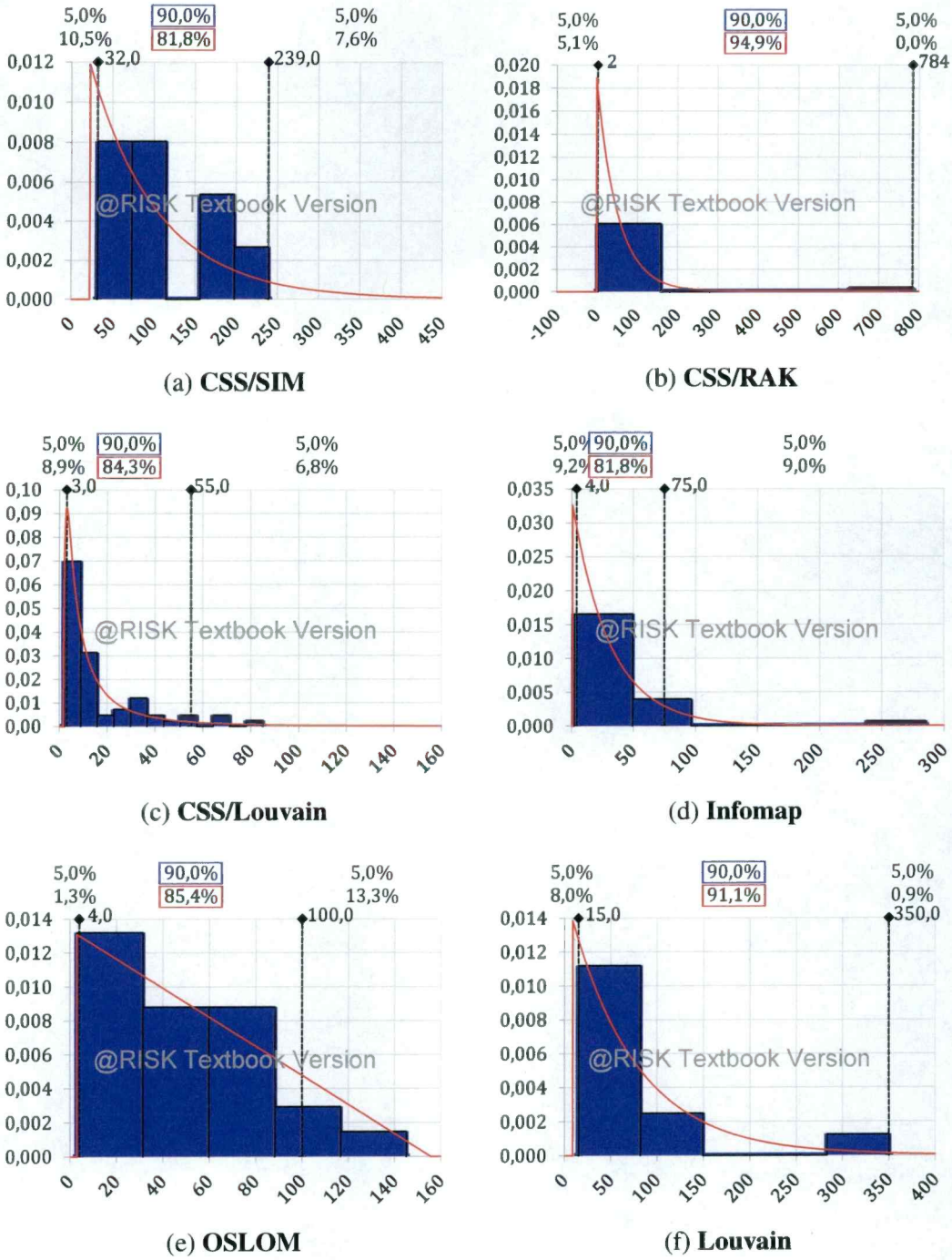


FIGURE B.3 – Distributions des taille des communautés détectées dans le graphe projeté par des algorithmes statiques. En rouge, une courbe de distribution possible : (a), (b), (c), (f) exponentielle ; (d) log-normale ; (e) triangulaire.

Classe	Nom	Référence	A S	R	É	F-D
Approche Probabiliste	**OSLOM**	Lancichinetti et al. [55]		✓		✓
	DSBM	Yang et al. [121]			✓	
Cadre de suivi	-	Goldberg et al. [68]		✓	✓	✓
	-	Greene et al. [67]	MOSES , Louvain	✓	✓	✓
	MONIC	Spiliopoulou et al. [65]			✓	✓
	-	Takaffoli et al. [69]		✓	✓	✓
	-	Tantipathananandh et al. [117]			✓	
	CommTracker	Wang et al. [66]	Louvain	✓	✓	✓
Consensus	-	Lancichinetti et Fortunato [73]				
CPM	-	Duan et al. [104]	CPM	✓		✓
	-	Palla et al. [21]	CPM	✓	✓	✓
Densité	**DENGRAPH**	Falkowski et al. [93, 94, 95]	DBSCAN	✓	✓	✓
Diffusion	**DiDiC**	Gehweiler et Meyerhenke [89]				
Factorisation	**NTF**	Gauvin et al. [123]		✓	✓	
Génétique	**DYN-MOGA**	Folino et Pizzuti [82]			✓	
	DYN-LSNNIA	Gong et al. [83]			✓	✓
	DNCD-MOEA	Chen et al. [84]			✓	
Marche aléatoire	-	Ben Jdidia et al. [115]	WalkTrap		✓	
Modèle généré	**FacetNet**	Lin et al. [75, 76]		✓		
	Evo-NetClus	Sun et al. [72]			✓	✓

TABLEAU B.1 – Résumé des méthodes de détection de communautés dynamiques dans un graphe évolutif. A S : Approche Statique. R : Recouvrement. É : Événements de base concernant les communautés (**naissance**, **mort**, **croissance**, **contraction**). F-D : **Fusion**, **Division**.

Classe	Nom	Référence	A S	R	É	F-D
Optimisation de la modularité	-	Aynaud et al. [119]	Louvain			
	-	Aynaud et Guillaume [120]	Louvain			
	-	Aynaud et Guillaume [70]	Louvain		✓	
	-	Bansal et al. [103]	CNM			
	A^{3}CS	Dinh et al. [109]				
	MIEN	Dinh et al. [108]	CNM		✓	✓
	-	Görke et al. [100]				
	QCA	Nguyen et al. [10]	Louvain			✓
	-	Mitra et al. [122]	Louvain		✓	
	-	Mucha et al. [116]	Louvain		✓	
	-	Shang et al. [105]	Louvain			✓
	.	Wang et al. [66]	Louvain	✓	✓	✓
	-	Wang et al.[88]	Louvain	✓	✓	
Particule-et-densité	**PDEC**	Kim et Han [77]		~	✓	✓
Programmation dynamique	**iLCD**	Cazabet et al. [106, 107, 17]		✓	✓	✓
Programmation linéaire	-	Kim et al. [86]				
Propagation d'étiquettes	**LabelRankT**	Xie et al. [111, 110]		✓		
Représentant	-	Chen et al. [64]		✓	✓	✓

TABLEAU B.2 – (Suite de B.1) Résumé des méthodes de détection de communautés dynamiques dans un graphe évolutif. A S : Approche Statique. R : Recouvrement. É : Événements de base concernant les communautés (**naissance**, **mort**, **croissance**, **contraction**). F-D : **Fusion**, **Division**.

Classe	Nom	Référence	A S	R	É	F-D
Spectral	-	Chakrabarti et al. [34]	*k*-moyenne, Spectral		✓	
	EvolSpec	Chi et al. [74]				
	-	Ning et al. [102]	Coupe normalisée			
	-	Tang et al. [71]	*k*-moyenne			
	-	Xu et al [79]	Spectral		✓	
	AFFECT	Xu et al. [81]	*k*-moyenne, Spectral			
Théorie de l'information	**GraphScope**	Sun et al. [114]				
Théorie des jeux	**NEO-CDD**	Lung et al. [92]				
Autre	-	Chen et al. [85]		✓	✓	✓
	DGlobal, TDLocal	Görke et al. [99, 96]	CNM, Louvian			✓
	-	Hopcroft et al. [63]			✓	✓
	AFOCS	Nguyen et al. [101]	Louvain	✓	✓	✓
	-	Riedy et Bader [112]				
	-	Takaffoli et al.[90, 91]			✓	✓
	-	Tantipathananandh et Berger-Wolf [118]			✓	
	-	Wang et al. [113]			✓	✓

TABLEAU B.3 – (Suite de B.2) Résumé des méthodes de détection de communautés dynamiques dans un graphe évolutif. A S : Approche Statique. R : Recouvrement. É : Événements de base concernant les communautés (**naissance**, **mort**, **croissance**, **contraction**). F-D : **Fusion**, **Division**.

	t	$\|V\|$	$\|E\|$	$\sum_u \frac{d_u}{N}$
Graphe étoilé	0	112	1223	180,304
	1	202	2207	157,762
	2	299	3530	142,234
	3	426	5278	143,559
	4	510	5967	122,059
	5	588	7443	125,558
	6	639	8666	132,156
	7	691	9802	145,988
	8	731	10 853	153,844
	9	785	13 765	204,227
	10	740	11 441	162,635
Graphe projeté	0	112	1485	239,893
	1	202	2609	207,554
	2	299	4424	194,040
	3	426	7062	212,545
	4	510	8342	184,655
	5	588	10 622	198,293
	6	640	12 365	206,200
	7	691	14 249	228,397
	8	731	16 352	250,353
	9	785	20 652	335,013
	10	740	17 558	266,573
Graphe biparti	0	7855	17 840	4,542
	1	12 341	28 73	4,550
	2	15 864	36 829	4,643
	3	20 903	51 55	4,885
	4	20 957	51 572	4,922
	5	23 937	60 263	5,035
	6	27 699	69 349	5,007
	7	33 224	82 972	4,995
	8	35 594	91 93	5,118
	9	50 41	129 415	5,172
	10	37 658	97 93	5,157

TABLEAU B.4 – Nombre de sommets, d'arêtes et degré moyen d'un sommet pour chacun des trois paradigmes de modélisation - graphes étoilé, projeté et biparti - des instantanés $G^{(t)}, t = \{0, \ldots, 10\}$.

	Nom	$N_{\mathscr{C}}$	$N/N_{\mathscr{C}}$	$\sum_{u \in V} \frac{\lvert Com(u) \rvert}{N}$	$\lvert Z \rvert$	$\lvert Y \rvert$
Graphe étoilé	**CFinder**	-	-	-	-	-
	CSS/Louvain	75	13,933	1	0	0
	CSS/RAK	24	43,542	1	0	0
	CSS/SIM	15	69,667	1	1	0
	iLCD	278 920	3,107	829,218	0	928
	Infomap	43	24,302	1	0	0
	Infomap$\diamond$	56	32,589	1,746	0	318
	Louvain	13	80,385	1	0	0
	OSLOM	22	50,182	1,056	0	55
	OSLOM*	23	42,174	1,061	131	45
Graphe projeté	**CFinder**	-	-	-	-	-
	CSS/Louvain	60	17,417	1	0	0
	CSS/RAK	19	55,000	1	0	0
	CSS/SIM	9	116,111	1	0	0
	iLCD	-	-	-	-	-
	Infomap	32	32,656	1	0	0
	Infomap$\diamond$	42	39,095	1,571	0	298
	Louvain	12	87,083	1	0	0
	OSLOM	24	64,417	1,091	0	59
	OSLOM*	33	31,121	1,075	90	52
Graphe biparti	**CFinder**	-	-	-	-	-
	CSS/Louvain	381	2,743	1	50	0
	CSS/RAK	371	2,817	1	80	0
	CSS/SIM	9	116,111	1	0	0
	iLCD	-	-	-	-	-
	Infomap	8	130,625	1	0	0
	Infomap$\diamond$	-	-	-	-	-
	Infomap$\diamond\diamond$	409	2,555	1	257	0
	Louvain	20	52,250	1	0	0
	OSLOM	266	3,929	1,627	0	419
	OSLOM*	254	4,114	1,512	0	374

TABLEAU B.5 – Résultats des algorithmes de détection dans le cadre statique. $\diamond$Version d'**Infomap** avec recouvrement. $\diamond\diamond$Version d'**Infomap** à deux niveaux. *Version d'**OSLOM** dans laquelle les sommets non assignés à une communauté sont retirés. $N_{\mathscr{C}}$: nombre de communautés détectées. $\frac{N}{N_{\mathscr{C}}}$ moyenne de sommets par communauté (individus seulement). $\frac{\sum_{u \in V} \lvert Com(u) \rvert}{N}$: nombre moyen de communautés auxquelles un sommet appartient (individus seulement). $\lvert Z \rvert$: nombre de sommets retirés ou non assignés (individus seulement) ou dans une communauté de un. $\lvert Y \rvert$: nombre de sommets assignés à plus d'une communauté (individus seulement).

	t	AFOCS	Louvain	Infomap	Infomap$\diamond$	OSLOM	OSLOM*
	0		6	9	9	2	2
	1		8	14	14	3	3
	2		11	22	2	14	13
	3		9	31	31	16	9
Graphe	4		11	35	2	20	19
étoilé	5		13	40	40	19	19
	6		10	58	2	18	20
	7		12	39	3	19	23
	8		11	47	2	26	23
	9		12	54	3	23	21
	10		14	61	2	24	22
	0		6	5	5	4	5
	1		8	12	12	4	3
	2		11	20	20	14	14
	3		9	24	24	14	16
Graphe	4		9	30	30	17	17
projeté	5		12	35	35	17	17
	6		12	38	2	18	17
	7		12	34	3	23	24
	8		10	30	3	28	26
	9		12	38	2	26	25
	10		12	42	2	19	23
	0	-	11	62	62	28	-
	1	29	13	110	110	46	-
	2	31	14	123	4	69	-
	3	-	17	162	5	88	-
Graphe	4	-	16	172	4	94	-
biparti	5	-	17	211	5	105	-
	6	-	18	232	4	118	-
	7	-	20	248	4	143	-
	8	-	20	241	7	150	-
	9	-	16	264	8	182	-
	10	-	16	227	3	150	-

TABLEAU B.6 – Résultats des algorithmes de détection dans le cadre dynamique : nombre de communautés dans chacun des instantanés $G^{(t)}, t = \{0, \ldots, 10\}$. $\diamond$Version d'**Infomap** multi-niveaux. *Version d'**OSLOM** dans laquelle les sommets non assignés à une communauté sont retirés.

	t	AFOCS	Louvain	Infomap	Infomap◇	OSLOM	OSLOM*
	0		18,67	12,44	12,44	56,00	56,00
	1		25,25	14,43	14,43	67,33	67,33
	2		27,18	13,59	149,50	21,36	23,00
	3		47,33	13,74	13,74	26,63	47,33
Graphe	4		46,36	14,57	255,00	25,50	26,84
étoilé	5		45,23	14,70	14,70	30,95	30,95
	6		63,90	11,02	319,50	35,50	31,95
	7		57,58	17,72	230,33	36,37	30,04
	8		66,45	15,55	365,50	28,12	31,78
	9		65,42	14,54	261,67	34,13	37,38
	10		52,86	12,13	370,00	30,83	33,64
	0		18,67	22,40	22,40	28,00	22,40
	1		25,25	16,83	16,83	50,50	67,33
	2		27,18	14,95	14,95	21,36	21,36
	3		47,33	17,75	17,75	30,43	26,63
Graphe	4		56,67	17,00	17,00	30,00	30,00
projeté	5		49,00	16,80	16,80	34,59	34,59
	6		53,25	16,82	319,50	35,50	37,59
	7		57,58	20,32	230,33	30,04	28,79
	8		73,10	24,37	243,67	26,11	28,12
	9		65,42	20,66	392,50	30,19	31,40
	10		61,67	17,62	370,00	38,95	32,17
	0	-	10,18	1,81	1,81	4,00	-
	1	6,97	15,54	1,84	1,84	4,39	-
	2	9,65	21,36	2,43	74,75	4,33	-
	3	-	25,06	2,63	85,20	4,84	-
Graphe	4	-	31,88	2,97	127,50	5,43	-
biparti	5	-	34,59	2,79	117,60	5,60	-
	6	-	35,50	2,75	159,75	5,42	-
	7	-	34,55	2,79	172,75	4,83	-
	8	-	36,55	3,03	104,43	4,87	-
	9	-	49,06	2,97	98,13	4,31	-
	10	-	46,25	3,26	246,67	4,93	-

TABLEAU B.7 – Résultats des algorithmes de détection dans le cadre dynamique : moyenne d'individus par communautés dans chacun des instantanés $G^{(t)}, t = \{0, \ldots, 10\}$. ◇Version d'**Infomap** multi-niveaux. *Version d'**OSLOM** dans laquelle les sommets non assignés à une communauté sont retirés.

		OSLOM			OSLOM*			
	t	$N_{\mathscr{C}}$	$\frac{N}{N_{\mathscr{C}}}$	$\sum_{u \in V} \frac{\lvert Com(u)\rvert}{N}$	$N_{\mathscr{C}}$	$\frac{N}{N_{\mathscr{C}}}$	$\sum_{u \in V} \frac{\lvert Com(u)\rvert}{N}$	$\lvert Z\rvert$
Graphe étoilé	0	2	58,500	1,045	2	18,000	1,091	79
	1	3	67,667	1,005	3	53,333	1,019	45
	2	14	22,000	1,030	13	15,769	1,030	100
	3	16	27,625	1,038	9	33,556	1,027	132
	4	20	27,000	1,059	19	23,316	1,099	107
	5	19	34,684	1,121	19	23,421	1,149	118
	6	18	40,167	1,130	20	30,350	1,165	118
	7	19	40,684	1,120	23	25,087	1,120	176
	8	26	31,615	1,125	23	30,435	1,175	135
	9	23	39,478	1,157	21	36,905	1,181	126
	10	24	38,625	1,253	22	31,046	1,122	131
Graphe projeté	0	4	28,500	1,018	5	11,800	1,073	57
	1	4	51,750	1,025	3	60,000	1,006	23
	2	14	21,357	1,000	14	14,786	1,000	92
	3	14	33,357	1,096	16	23,938	1,123	85
	4	17	34,941	1,165	17	30,294	1,179	73
	5	17	40,294	1,165	17	35,412	1,214	92
	6	18	41,278	1,161	17	37,765	1,211	110
	7	23	35,870	1,196	24	28,042	1,168	115
	8	28	33,964	1,301	26	31,654	1,315	105
	9	26	40,000	1,325	25	39,080	1,431	102
	10	19	45,895	1,178	23	32,391	1,181	109
Graphe biparti	0	28	281,786	1,005	-	-	-	-
	1	46	269,609	1,005	-	-	-	-
	2	69	231,609	1,007	-	-	-	-
	3	88	238,841	1,006	-	-	-	-
	4	94	226,968	1,018	-	-	-	-
	5	105	230,867	1,013	-	-	-	-
	6	118	238,441	1,016	-	-	-	-
	7	143	236,070	1,016	-	-	-	-
	8	150	239,820	1,011	-	-	-	-
	9	182	278,555	1,013	-	-	-	-
	10	150	253,807	1,011	-	-	-	-

TABLEAU B.8 – Résultats des versions de l'algorithme de détection dans le cadre dynamique **OSLOM** dans chacun des instantanés $G^{(t)}, t = \{0, \ldots, 10\}$. *Version d'**OSLOM** dans laquelle les sommets non assignés à une communauté sont retirés. $N_{\mathscr{C}}$: nombre de communautés détectées. $\frac{N}{N_{\mathscr{C}}}$ moyenne de sommets par communauté (individus seulement). $\frac{\sum_{u \in V} \lvert Com(u)\rvert}{N}$: nombre moyen de communautés auxquelles un sommet appartient (individus seulement). $\lvert Z\rvert$: nombre de sommets retirés ou non assignés (individus seulement).

	Nom	$N_{\mathscr{C}}$	$mod(\mathscr{C})$	$mod_b(\mathscr{C})$	$mod_o(\mathscr{C})$	$\bar{\phi}(\mathscr{C})$	$\bar{\Upsilon}(\mathscr{C})$
Graphe étoilé	**CFinder**	-			-	-	-
	CSS/Louvain	75	0,4828			0,5373	0,3854
	CSS/RAK	24	0,3941			**0,3661**	0,3404
	CSS/SIM	15	**0,5732**			0,3977	0,5799
	iLCD	-			0,0005	-	0,5213
	Infomap	43	0,5516			0,4759	0,4560
	Infomap◇	56			0,4550	0,7791	0,6478
	Louvain	13	**0,5718**			**0,2892**	0,6267
	OSLOM	22			0,4978	0,5199	0,5883
	OSLOM*	23			0,4605	0,5758	0,6110
Graphe projeté	**CFinder**	-			-	-	-
	CSS/Louvain	60	0,4558			0,5387	0,4062
	CSS/RAK	19	0,3690			0,3726	0,3513
	CSS/SIM	9	**0,5433**			**0,2891**	0,7033
	iLCD	-			-	-	-
	Infomap	32	0,5288			0,4593	0,4536
	Infomap◇	42			0,4540	0,7603	0,6697
	Louvain	12	**0,5403**			**0,3042**	0,6812
	OSLOM	24			0,4403	0,6296	0,5885
	OSLOM*	33			0,4500	0,6412	0,5622
Graphe biparti	**CFinder**	-		-		-	-
	CSS/Louvain	381	0,4403	0,4408		0,5433	0,3927
	CSS/RAK	371	0,4238	0,4238		0,5862	0,3531
	CSS/SIM	9	0,6477	0,6485		**0,1686**	0,6891
	iLCD	-			-	-	-
	Infomap	8	0,5710	0,5710		**0,1315**	0,6982
	Infomap◇	-			-	-	-
	Infomap◇◇	409	0,6311	0,6314		0,4496	0,1006
	Louvain	20	**0,7019**	**0,7023**		0,2332	0,6315
	OSLOM	266			0,4208	0,6761	0,3695
	OSLOM*	254			0,4731	0,6691	0,3669

TABLEAU B.9 – Résultats des algorithmes de détection dans le cadre statique : mesures d'évaluation des communautés. ◇Version d'**Infomap** avec recouvrement. ◇◇Version d'**Infomap** à deux niveaux.*Version d'**OSLOM** dans laquelle les sommets non assignés à une communauté sont retirés. $N_{\mathscr{C}}$: nombre de communautés. $mod(\mathscr{C})$ (3.2) : modularité de Newman et Girvan [23]. $mod_b(\mathscr{C})$: modularité bipartie de Barber [25]. $mod_o(\mathscr{C})$ (3.9) : modularité avec recouvrement. $\bar{\phi}(\mathscr{C})$ 4.1 : conductance moyenne du graphe. $\bar{\Upsilon}(\mathscr{C})$ (5.1) : fractionalisation moyenne du graphe.

Graphe étoilé

	Louvain			Infomap			OSLOM			OSLOM*		
t	$\bar{\Upsilon}$	$\bar{\phi}$	*mod*	$\bar{\Upsilon}$	$\bar{\phi}$	*mod*	$\bar{\Upsilon}$	$\bar{\phi}$	*mod*	$\bar{\Upsilon}$	$\bar{\phi}$	*mod*
0	0,6972	0,3993	0,4015	0,5352	0,4784	0,3971	0,7803	0,4063	0,0601	0,7472	0,3564	0,0850
1	0,6153	0,2445	0,5247	0,6124	0,3725	0,5187	0,6618	0,3355	0,0447	0,6098	0,3081	0,0746
2	0,6705	0,3130	0,5584	0,5761	0,4068	0,5417	0,6591	0,4613	0,4934	0,6552	0,4568	0,4872
3	0,6705	0,2253	0,6014	0,5416	0,4618	0,5833	0,6768	0,4990	0,5341	0,7368	0,3823	0,5309
4	0,6852	0,2675	0,6301	0,4889	0,4244	0,5966	0,6354	0,5158	0,5433	0,6533	0,5300	0,5408
5	0,6734	0,2673	0,6206	0,5011	0,4457	0,5981	0,6793	0,5448	0,4922	0,6768	0,5663	0,5090
6	0,7124	0,2739	0,5989	0,5341	0,4842	0,5576	0,7004	0,5554	0,4721	0,7050	0,5650	0,4619
7	0,6744	0,2896	0,6065	0,5280	0,4223	0,5909	0,6666	0,5776	0,4704	0,6436	0,5504	0,4950
8	0,7018	0,2483	0,6265	0,4857	0,4553	0,5894	0,6346	0,5739	0,4932	0,6606	0,5928	0,4520
9	0,6520	0,2192	0,6528	0,4860	0,4352	0,5943	0,6172	0,5080	0,5472	0,5957	0,5414	0,4589
10	0,6078	0,2225	0,6605	0,4540	0,4741	0,6129	0,6484	0,6018	0,4809	0,6506	0,5247	0,5301

TABLEAU B.10 – Résultats des algorithmes de détection dans le cadre dynamique : mesures d'évaluation des communautés dans chacun des instantanés $G^{(t)}, t = \{0, \ldots, 10\}$. ◇Version d'**Infomap** multi-niveaux. *Version d'**OSLOM** dans laquelle les sommets non assignés à une communauté sont retirés. $mod = mod(\mathscr{C}^{(t)})$: modularité avec ou sans recouvrement, bipartie ou non, selon le cas. $\bar{\phi} = \bar{\phi}(\mathscr{C}^{(t)})$ 4.1 : conductance moyenne de l'instantané. $\bar{\Upsilon} = \bar{\Upsilon}(\mathscr{C}^{(t)})$ (5.1) : fractionalisation moyenne de l'instantané.

Graphe projeté												
	Louvain			Infomap			OSLOM			OSLOM*		
t	$\bar{\Upsilon}$	$\bar{\phi}$	*mod*	$\bar{\Upsilon}$	$\bar{\phi}$	*mod*	$\bar{\Upsilon}$	$\bar{\phi}$	*mod*	$\bar{\Upsilon}$	$\bar{\phi}$	*mod*
0	0,6678	0,4468	0,3530	0,3570	0,5283	0,1383	0,6025	0,4890	0,1746	0,6094	0,5788	0,2139
1	0,6338	0,2545	0,5081	0,6263	0,3478	0,5008	0,6225	0,2430	0,1853	0,6105	0,2020	0,0774
2	0,7030	0,3653	0,5268	0,6051	0,4304	0,5184	0,5993	0,4667	0,4796	0,6043	0,4681	0,4788
3	0,6659	0,2484	0,5924	0,5523	0,4359	0,5793	0,6972	0,5700	0,4739	0,6352	0,5748	0,4798
4	0,6638	0,2459	0,5999	0,5178	0,4679	0,5772	0,7015	0,5736	0,5038	0,6723	0,5616	0,5077
5	0,6671	0,2828	0,5846	0,5311	0,4581	0,5633	0,7112	0,6114	0,4585	0,7207	0,6287	0,4435
6	0,6946	0,3101	0,5507	0,5750	0,4669	0,5203	0,7294	0,6633	0,3931	0,7319	0,6652	0,3968
7	0,6129	0,2898	0,5500	0,5402	0,4567	0,5348	0,6659	0,6791	0,4077	0,6389	0,6453	0,4423
8	0,6878	0,2582	0,5779	0,5307	0,4628	0,5629	0,6731	0,7365	0,3685	0,6688	0,7157	0,3796
9	0,5932	0,2333	0,6065	0,5011	0,4590	0,5677	0,6622	0,6781	0,3816	0,6480	0,6810	0,3743
10	0,6833	0,2780	0,6131	0,4890	0,4761	0,5815	0,6869	0,6070	0,4053	0,6697	0,5856	0,4829

TABLEAU B.11 – Résultats des algorithmes de détection dans le cadre dynamique (Suite de B.10) : mesures d'évaluation des communautés dans chacun des instantanés $G^{(t)}, t = \{0, \ldots, 10\}$. *Version d'**OSLOM** dans laquelle les sommets non assignés à une communauté sont retirés. $mod = mod(\mathscr{C}^{(t)})$: modularité avec ou sans recouvrement, bipartie ou non, selon le cas. $\bar{\phi} = \bar{\phi}(\mathscr{C}^{(t)})$ 4.1 : conductance moyenne de l'instantané. $\bar{\Upsilon} = \bar{\Upsilon}(\mathscr{C}^{(t)})$ (5.1) : fractionalisation moyenne de l'instantané.

Graphe biparti

	AFOCS			Louvain			Infomap			OSLOM		
t	$\bar{\Upsilon}$	$\bar{\phi}$	*mod*	$\bar{\Upsilon}$	$\bar{\phi}$	*mod*	$\bar{\Upsilon}$	$\bar{\phi}$	*mod*	$\bar{\Upsilon}$	$\bar{\phi}$	*mod*
0	-	-	-	0,5471	0,2978	0,5878	0,1174	0,4259	0,5685	0,4590	0,5874	0,4294
1	0,5087	0,8733	0,0086	0,6452	0,2032	0,6751	0,1046	0,4049	0,6415	0,4554	0,5821	0,5083
2	0,4866	0,8560	0,0103	0,6316	0,2113	0,6981	0,1590	0,3983	0,6607	0,4489	0,6025	0,5020
3	-	-	-	0,6586	0,2271	0,7108	0,1946	0,4049	0,6575	0,4461	0,6062	0,5674
4	-	-	-	0,6538	0,1996	0,7289	0,1848	0,4054	0,6847	0,4861	0,5879	0,5568
5	-	-	-	0,6770	0,2136	0,7313	0,1663	0,4171	0,6737	0,5140	0,5841	0,5672
6	-	-	-	0,6516	0,2026	0,7166	0,1965	0,4348	0,6840	0,5200	0,5949	0,5473
7	-	-	-	0,6616	0,2065	0,7229	0,1801	0,4429	0,6798	0,4636	0,6427	0,5272
8	-	-	-	0,6614	0,2213	0,7226	0,1869	0,4342	0,6736	0,4525	0,6164	0,5285
9	-	-	-	0,6556	0,1959	0,7393	0,1553	0,4390	0,6962	0,3723	0,6340	0,5362
10	-	-	-	0,6336	0,1753	0,7459	0,1859	0,4171	0,7063	0,4153	0,5921	0,5851

TABLEAU B.12 – Résultats des algorithmes de détection dans le cadre dynamique (Suite de B.11) : mesures d'évaluation des communautés dans chacun des instantanés $G^{(t)}, t = \{0, \ldots, 10\}$. *Version d'**OSLOM** dans laquelle les sommets non assignés à une communauté sont retirés. $mod = mod(\mathscr{C}^{(t)})$: modularité avec ou sans recouvrement, bipartie ou non, selon le cas. $\bar{\phi} = \bar{\phi}(\mathscr{C}^{(t)})$ 4.1 : conductance moyenne de l'instantané. $\bar{\Upsilon} = \bar{\Upsilon}(\mathscr{C}^{(t)})$ (5.1) : fractionalisation moyenne de l'instantané.

Bibliographie

[1] S. WASSERMAN, *Social network analysis : Methods and applications*, vol. 8. Cambridge university press, 1994.

[2] E. BULLMORE et O. SPORNS, « Complex brain networks : graph theoretical analysis of structural and functional systems », *Nature Reviews Neuroscience*, vol. 10, no. 3, p. 186–198, 2009.

[3] J. A. DUNNE, R. J. WILLIAMS et N. D. MARTINEZ, « Food-web structure and network theory : the role of connectance and size », *Proceedings of the National Academy of Sciences*, vol. 99, no. 20, p. 12917–12922, 2002.

[4] D. PRICE, « Networks of scientific papers. », *Science (New York, NY)*, vol. 149, no. 3683, p. 510, 1965.

[5] R. DE CASTRO et J. W. GROSSMAN, « Famous trails to paul erdős », *The Mathematical Intelligencer*, vol. 21, no. 3, p. 51–53, 1999.

[6] A. BRODER, R. KUMAR, F. MAGHOUL, P. RAGHAVAN, S. RAJAGOPALAN, R. STATA, A. TOMKINS et J. WIENER, « Graph structure in the web », *Computer networks*, vol. 33, no. 1, p. 309–320, 2000.

[7] D. J. WATTS et S. H. STROGATZ, « Collective dynamics of small-world networks », *nature*, vol. 393, no. 6684, p. 440–442, 1998.

[8] A.-L. BARABÁSI et R. ALBERT, « Emergence of scaling in random networks », *science*, vol. 286, no. 5439, p. 509–512, 1999.

[9] M. GIRVAN et M. E. NEWMAN, « Community structure in social and biological networks », *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, p. 7821–7826, 2002.

[10] N. P. NGUYEN, T. N. DINH, Y. XUAN et M. T. THAI, « Adaptive algorithms for detecting community structure in dynamic social networks », *in INFOCOM, 2011 Proceedings IEEE*, p. 2282–2290, IEEE, 2011.

[11] M. K. AGARWAL, K. RAMAMRITHAM et M. BHIDE, « Real time discovery of dense clusters in highly dynamic graphs : identifying real world events in highly dynamic environments », *Proceedings of the VLDB Endowment*, vol. 5, no. 10, p. 980–991, 2012.

[12] M. E. NEWMAN, « Fast algorithm for detecting community structure in networks », *Physical review E*, vol. 69, no. 6, p. 066133, 2004.

[13] M. A. PORTER, J.-P. ONNELA et P. J. MUCHA, « Communities in networks », *Notices of the AMS*, vol. 56, no. 9, p. 1082–1097, 2009.

[14] S. FORTUNATO, « Community detection in graphs », *Physics Reports*, vol. 486, no. 3, p. 75–174, 2010.

[15] C. C. BILGIN et B. YENER, « Dynamic network evolution : Models, clustering, anomaly detection », *IEEE Networks*, 2006.

[16] T. AYNAUD, E. FLEURY, J.-L. GUILLAUME et Q. WANG, « Communities in evolving networks : Definitions, detection, and analysis techniques », *in Dynamics On and Of Complex Networks, Volume 2*, p. 159–200, Springer, 2013.

[17] R. CAZABET, *Détection de communautés dynamiques dans des réseaux temporels*. Thèse doctorat, Université Paul Sabatier-Toulouse III, 2013.

[18] C. AGGARWAL et K. SUBBIAN, « Evolutionary network analysis : A survey », *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, p. 10, 2014.

[19] T. HARTMANN, A. KAPPES et D. WAGNER, « Clustering evolving networks », *arXiv preprint arXiv :1401.3516*, 2014.

[20] F. RADICCHI, C. CASTELLANO, F. CECCONI, V. LORETO et D. PARISI, « Defining and identifying communities in networks », *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 9, p. 2658–2663, 2004.

[21] G. PALLA, A.-L. BARABÁSI et T. VICSEK, « Quantifying social group evolution », *Nature*, vol. 446, no. 7136, p. 664–667, 2007.

[22] S. ASUR, S. PARTHASARATHY et D. UCAR, « An event-based framework for characterizing the evolutionary behavior of interaction graphs », *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 4, p. 16, 2009.

[23] M. E. NEWMAN et M. GIRVAN, « Finding and evaluating community structure in networks », *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[24] M. E. NEWMAN, « Analysis of weighted networks », *Physical Review E*, vol. 70, no. 5, p. 056131, 2004.

[25] M. J. BARBER, « Modularity and community detection in bipartite networks », *Physical Review E*, vol. 76, no. 6, p. 066102, 2007.

[26] R. GUIMERÀ, M. SALES-PARDO et L. A. N. AMARAL, « Module identification in bipartite and directed networks », *Physical Review E*, vol. 76, no. 3, p. 036102, 2007.

[27] S. FORTUNATO et M. BARTHELEMY, « Resolution limit in community detection », *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, p. 36–41, 2007.

[28] Z. LI, S. ZHANG, R.-S. WANG, X.-S. ZHANG et L. CHEN, « Quantitative function for community detection », *Physical review E*, vol. 77, no. 3, p. 036109, 2008.

[29] B. BOLLOBÁS, *Modern graph theory*, vol. 184. Springer Science & Business Media, 1998.

[30] J. SHI et J. MALIK, « Normalized cuts and image segmentation », *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, p. 888–905, 2000.

[31] V. NICOSIA, G. MANGIONI, V. CARCHIOLO et M. MALGERI, « Extending the definition of modularity to directed graphs with overlapping communities », *Journal of Statistical Mechanics : Theory and Experiment*, vol. 2009, no. 03, p. P03024, 2009.

[32] D. CHEN, M. SHANG, Z. LV et Y. FU, « Detecting overlapping communities of weighted networks via a local algorithm », *Physica A : Statistical Mechanics and its Applications*, vol. 389, no. 19, p. 4177–4187, 2010.

[33] H. SHEN, X. CHENG, K. CAI et M.-B. HU, « Detect overlapping and hierarchical community structure in networks », *Physica A : Statistical Mechanics and its Applications*, vol. 388, no. 8, p. 1706–1712, 2009.

[34] D. CHAKRABARTI, R. KUMAR et A. TOMKINS, « Evolutionary clustering », *in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 554–560, ACM, 2006.

[35] D. J. MacKay, *Information theory, inference, and learning algorithms*, vol. 7. Citeseer, 2003.

[36] R. M. Gray, *Entropy and information theory*. Springer Science & Business Media, 2011.

[37] L. Danon, A. Diaz-Guilera, J. Duch et A. Arenas, « Comparing community structure identification », *Journal of Statistical Mechanics : Theory and Experiment*, vol. 2005, no. 09, p. P09008, 2005.

[38] C. E. Shannon, « A mathematical theory of communication », *ACM SIGMOBILE Mobile Computing and Communications Review, Reprinted with corrections from The Bell System Technical Journal, Vol. 27, pp. 379–423, 623–656, July, October, 1948*, vol. 5, no. 1, p. 3–55, 2001.

[39] W. M. Rand, « Objective criteria for the evaluation of clustering methods », *Journal of the American Statistical association*, vol. 66, no. 336, p. 846–850, 1971.

[40] P. Jaccard, *Distribution de la Flore Alpine : dans le Bassin des dranses et dans quelques régions voisines*. Rouge, 1901.

[41] D. Aloise, S. Cafieri, G. Caporossi, P. Hansen, S. Perron et L. Liberti, « Column generation algorithms for exact modularity maximization in networks », *Physical Review E*, vol. 82, no. 4, p. 046112, 2010.

[42] G. Agarwal et D. Kempe, « Modularity-maximizing graph communities via mathematical programming », *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 66, no. 3, p. 409–418, 2008.

[43] S. Cafieri, P. Hansen et L. Liberti, « Edge ratio and community structure in networks », *Physical Review E*, vol. 81, no. 2, p. 026105, 2010.

[44] R. Guimerà, M. Sales-Pardo et L. A. N. Amaral, « Modularity from fluctuations in random graphs and complex networks », *Physical Review E*, vol. 70, no. 2, p. 025101, 2004.

[45] M. E. Newman, « Finding community structure in networks using the eigenvectors of matrices », *Physical review E*, vol. 74, no. 3, p. 036104, 2006.

[46] J. Ruan et W. Zhang, « Identifying network communities with a high resolution », *Physical Review E*, vol. 77, no. 1, p. 016104, 2008.

[47] J. Duch et A. Arenas, « Community detection in complex networks using extremal optimization », *Physical review E*, vol. 72, no. 2, p. 027104, 2005.

[48] A. CLAUSET, M. E. NEWMAN et C. MOORE, « Finding community structure in very large networks », *Physical review E*, vol. 70, no. 6, p. 066111, 2004.

[49] V. D. BLONDEL, J.-L. GUILLAUME, R. LAMBIOTTE et E. LEFEBVRE, « Fast unfolding of communities in large networks », *Journal of Statistical Mechanics : Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[50] P. PONS et M. LATAPY, « Computing communities in large networks using random walks », *in Computer and Information Sciences-ISCIS 2005*, p. 284–293, Springer, 2005.

[51] M. ROSVALL et C. T. BERGSTROM, « Maps of random walks on complex networks reveal community structure », *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, p. 1118–1123, 2008.

[52] G. PALLA, I. DERÉNYI, I. FARKAS et T. VICSEK, « Uncovering the overlapping community structure of complex networks in nature and society », *Nature*, vol. 435, no. 7043, p. 814–818, 2005.

[53] U. N. RAGHAVAN, R. ALBERT et S. KUMARA, « Near linear time algorithm to detect community structures in large-scale networks », *Physical Review E*, vol. 76, no. 3, p. 036106, 2007.

[54] S. GREGORY, « Finding overlapping communities in networks by label propagation », *New Journal of Physics*, vol. 12, no. 10, p. 103018, 2010.

[55] A. LANCICHINETTI, F. RADICCHI, J. J. RAMASCO et S. FORTUNATO, « Finding statistically significant communities in networks », *PloS one*, vol. 6, no. 4, p. e18961, 2011.

[56] A. MCDAID et N. HURLEY, « Detecting highly overlapping communities with model-based overlapping seed expansion », *in Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, p. 112–119, IEEE, 2010.

[57] I. X. LEUNG, P. HUI, P. LIO et J. CROWCROFT, « Towards real-time community detection in large networks », *Physical Review E*, vol. 79, no. 6, p. 066107, 2009.

[58] M. ROSVALL, D. AXELSSON et C. T. BERGSTROM, « The map equation », *The European Physical Journal-Special Topics*, vol. 178, no. 1, p. 13–23, 2009.

[59] A. LANCICHINETTI et S. FORTUNATO, « Community detection algorithms : a comparative analysis », *Physical review E*, vol. 80, no. 5, p. 056117, 2009.

[60] J. XIE, S. KELLEY et B. K. SZYMANSKI, « Overlapping community detection in networks : The state-of-the-art and comparative study », *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 43, 2013.

[61] P. LATOUCHE, E. BIRMELÉ, C. AMBROISE et AL., « Overlapping stochastic block models with application to the french political blogosphere », *The Annals of Applied Statistics*, vol. 5, no. 1, p. 309–336, 2011.

[62] S. KIRKPATRICK, M. P. VECCHI et AL., « Optimization by simulated annealing », 1983.

[63] J. HOPCROFT, O. KHAN, B. KULIS et B. SELMAN, « Tracking evolving communities in large linked networks », *Proceedings of the national academy of sciences of the United States of America*, vol. 101, no. Suppl 1, p. 5249–5253, 2004.

[64] Z. CHEN, K. A. WILSON, Y. JIN, W. HENDRIX et N. F. SAMATOVA, « Detecting and tracking community dynamics in evolutionary networks », *in Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, p. 318–327, IEEE, 2010.

[65] M. SPILIOPOULOU, I. NTOUTSI, Y. THEODORIDIS et R. SCHULT, « Monic : modeling and monitoring cluster transitions », *in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 706–711, ACM, 2006.

[66] Y. WANG, B. WU et N. DU, « Community evolution of social network : feature, algorithm and model », *arXiv preprint arXiv :0804.4356*, 2008.

[67] D. GREENE, D. DOYLE et P. CUNNINGHAM, « Tracking the evolution of communities in dynamic social networks », *in Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, p. 176–183, IEEE, 2010.

[68] M. GOLDBERG, M. MAGDON-ISMAIL, S. NAMBIRAJAN et J. THOMPSON, « Tracking and predicting evolution of social communities », *in Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, p. 780–783, IEEE, 2011.

[69] M. TAKAFFOLI, F. SANGI, J. FAGNAN et O. R. ZÄIANE, « Community evolution mining in dynamic social networks », *Procedia-Social and Behavioral Sciences*, vol. 22, p. 49–58, 2011.

[70] T. AYNAUD et J.-L. GUILLAUME, « Static community detection algorithms for evolving networks », *in Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on*, p. 513–519, IEEE, 2010.

[71] L. TANG, H. LIU, J. ZHANG et Z. NAZERI, « Community evolution in dynamic multi-mode networks », *in Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 677–685, ACM, 2008.

[72] Y. SUN, J. TANG, J. HAN, M. GUPTA et B. ZHAO, « Community evolution detection in dynamic heterogeneous information networks », *in Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, p. 137–146, ACM, 2010.

[73] A. LANCICHINETTI et S. FORTUNATO, « Consensus clustering in complex networks », *Scientific reports*, vol. 2, 2012.

[74] Y. CHI, X. SONG, D. ZHOU, K. HINO et B. L. TSENG, « Evolutionary spectral clustering by incorporating temporal smoothness », *in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 153–162, ACM, 2007.

[75] Y.-R. LIN, Y. CHI, S. ZHU, H. SUNDARAM et B. L. TSENG, « Analyzing communities and their evolutions in dynamic social networks », *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 2, p. 8, 2009.

[76] Y.-R. LIN, Y. CHI, S. ZHU, H. SUNDARAM et B. L. TSENG, « Facetnet : a framework for analyzing communities and their evolutions in dynamic networks », *in Proceedings of the 17th international conference on World Wide Web*, p. 685–694, ACM, 2008.

[77] M.-S. KIM et J. HAN, « A particle-and-density based evolutionary clustering method for dynamic networks », *Proceedings of the VLDB Endowment*, vol. 2, no. 1, p. 622–633, 2009.

[78] M. ESTER, H.-P. KRIEGEL, J. SANDER et X. XU, « A density-based algorithm for discovering clusters in large spatial databases with noise. », *in Kdd*, vol. 96, p. 226–231, 1996.

[79] K. S. XU, M. KLIGER et A. O. HERO III, « Tracking communities in dynamic social networks », *in Social Computing, Behavioral-Cultural Modeling and Prediction*, p. 219–226, Springer, 2011.

[80] S. X. YU et J. SHI, « Multiclass spectral clustering », *in Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, p. 313–319, IEEE, 2003.

[81] K. S. XU, M. KLIGER et A. O. HERO III, « Adaptive evolutionary clustering », *Data Mining and Knowledge Discovery*, vol. 28, no. 2, p. 304–336, 2014.

[82] F. FOLINO et C. PIZZUTI, « A multiobjective and evolutionary clustering method for dynamic networks », *in Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, p. 256–263, IEEE, 2010.

[83] M.-G. GONG, L.-J. ZHANG, J.-J. MA et L.-C. JIAO, « Community detection in dynamic social networks based on multiobjective immune algorithm », *Journal of Computer Science and Technology*, vol. 27, no. 3, p. 455–467, 2012.

[84] G. CHEN, Y. WANG et J. WEI, « A new multiobjective evolutionary algorithm for community detection in dynamic complex networks », *Mathematical problems in engineering*, vol. 2013, 2013.

[85] Y. CHEN, V. KAWADIA et R. URGAONKAR, « Detecting overlapping temporal community structure in time-evolving networks », *arXiv preprint arXiv :1303.7226*, 2013.

[86] K. KIM, R. I. MCKAY et B.-R. MOON, « Multiobjective evolutionary algorithms for dynamic social network clustering », *in Proceedings of the 12th annual conference on Genetic and evolutionary computation*, p. 1179–1186, ACM, 2010.

[87] P. J. ROUSSEEUW, « Silhouettes : a graphical aid to the interpretation and validation of cluster analysis », *Journal of computational and applied mathematics*, vol. 20, p. 53–65, 1987.

[88] Q. WANG, E. FLEURY et AL., « Mining time-dependent communities », *in LAWDN-Latin-American Workshop on Dynamic Networks*, 2010.

[89] J. GEHWEILER et H. MEYERHENKE, « A distributed diffusive heuristic for clustering a virtual p2p supercomputer », *in Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, p. 1–8, IEEE, 2010.

[90] M. TAKAFFOLI, R. RABBANY et O. R. ZAÏANE, « Incremental local community identification in dynamic social networks », *in Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, p. 90–94, ACM, 2013.

[91] M. TAKAFFOLI, F. SANGI, J. FAGNAN et O. R. ZAÏANE, « Modec-modeling and detecting evolutions of communities. », *in ICWSM*, 2011.

[92] R. I. LUNG, C. CHIRA et A. ANDREICA, « Game theory and extremal optimization for community detection in complex dynamic networks », *PloS one*, vol. 9, no. 2, p. e86891, 2014.

[93] T. FALKOWSKI, A. BARTH et M. SPILIOPOULOU, « Dengraph : A density-based community detection algorithm », *in Web Intelligence, IEEE/WIC/ACM International Conference on*, p. 112–115, IEEE, 2007.

[94] T. FALKOWSKI, *Community analysis in dynamic social networks*. Sierke, 2009.

[95] N. SCHLITTER, T. FALKOWSKI et J. LÄSSIG, « Dengraph-ho : a density-based hierarchical graph clustering algorithm », *Expert Systems*, 2013.

[96] R. GÖRKE, T. HARTMANN et D. WAGNER, *Dynamic graph clustering using minimum-cut trees*. Springer, 2009.

[97] G. W. FLAKE, R. E. TARJAN et K. TSIOUTSIOULIKLIS, « Graph clustering and minimum cut trees », *Internet Mathematics*, vol. 1, no. 4, p. 385–408, 2004.

[98] R. E. GOMORY et T. C. HU, « Multi-terminal network flows », *Journal of the Society for Industrial & Applied Mathematics*, vol. 9, no. 4, p. 551–570, 1961.

[99] R. GÖRKE, P. MAILLARD, C. STAUDT et D. WAGNER, *Modularity-driven clustering of dynamic graphs*. Springer, 2010.

[100] R. GÖRKE, P. MAILLARD, A. SCHUMM, C. STAUDT et D. WAGNER, « Dynamic graph clustering combining modularity and smoothness », *Journal of Experimental Algorithmics (JEA)*, vol. 18, p. 1–5, 2013.

[101] N. P. NGUYEN, T. N. DINH, S. TOKALA et M. T. THAI, « Overlapping communities in dynamic networks : their detection and mobile applications », *in Proceedings of the 17th annual international conference on Mobile computing and networking*, p. 85–96, ACM, 2011.

[102] H. NING, W. XU, Y. CHI, Y. GONG et T. S. HUANG, « Incremental spectral clustering by efficiently updating the eigen-system », *Pattern Recognition*, vol. 43, no. 1, p. 113–127, 2010.

[103] S. BANSAL, S. BHOWMICK et P. PAYMAL, « Fast community detection for dynamic complex networks », *in Complex Networks*, p. 196–207, Springer, 2011.

[104] D. DUAN, Y. LI, R. LI et Z. LU, « Incremental k-clique clustering in dynamic social networks », *Artificial Intelligence Review*, vol. 38, no. 2, p. 129–147, 2012.

[105] J. SHANG, L. LIU, F. XIE, Z. CHEN, J. MIAO, X. FANG et C. WU, « A real-time detecting algorithm for tracking community structure of dynamic networks », *in 6th SNA-KDD Workshop*, vol. 12, 2012.

[106] R. CAZABET, F. AMBLARD et C. HANACHI, « Detection of overlapping communities in dynamical social networks », *in Social Computing (SocialCom), 2010 IEEE Second International Conference on*, p. 309–314, IEEE, 2010.

[107] R. CAZABET et F. AMBLARD, « Simulate to detect : a multi-agent system for community detection », *in Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, vol. 2, p. 402–408, IEEE, 2011.

[108] T. N. DINH, I. SHIN, N. K. THAI, M. T. THAI et T. ZNATI, « A general approach for modules identification in evolving networks », *in Dynamics of Information Systems*, p. 83–100, Springer, 2010.

[109] T. N. DINH, N. P. NGUYEN et M. T. THAI, « An adaptive approximation algorithm for community detection in dynamic scale-free networks », *in INFOCOM, 2013 Proceedings IEEE*, p. 55–59, IEEE, 2013.

[110] J. XIE, M. CHEN et B. K. SZYMANSKI, « Labelrankt : Incremental community detection in dynamic networks via label propagation », *in Proceedings of the Workshop on Dynamic Networks Management and Mining*, p. 25–32, ACM, 2013.

[111] J. XIE et B. K. SZYMANSKI, « Labelrank : A stabilized label propagation algorithm for community detection in networks », *in Network Science Workshop (NSW), 2013 IEEE 2nd*, p. 138–143, IEEE, 2013.

[112] J. RIEDY et D. A. BADER, « Multithreaded community monitoring for massive streaming graph data », *in Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International*, p. 1646–1655, IEEE, 2013.

[113] C.-D. WANG, J.-H. LAI et S. Y. PHILIP, « Dynamic community detection in weighted graph streams », *in Proc. of SDM*, p. 151–161, SIAM, 2013.

[114] J. SUN, C. FALOUTSOS, S. PAPADIMITRIOU et P. S. YU, « Graphscope : parameter-free mining of large time-evolving graphs », *in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 687–696, ACM, 2007.

[115] M. B. JDIDIA, C. ROBARDET, E. FLEURY *et al.*, « Communities detection and analysis of their dynamics in collaborative networks. », *in ICDIM*, p. 744–749, 2007.

[116] P. J. MUCHA, T. RICHARDSON, K. MACON, M. A. PORTER et J.-P. ONNELA, « Community structure in time-dependent, multiscale, and multiplex networks », *Science*, vol. 328, no. 5980, p. 876–878, 2010.

[117] C. TANTIPATHANANANDH, T. BERGER-WOLF et D. KEMPE, « A framework for community identification in dynamic social networks », *in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 717–726, ACM, 2007.

[118] C. TANTIPATHANANANDH et T. Y. BERGER-WOLF, « Finding communities in dynamic social networks », *in Data Mining (ICDM), 2011 IEEE 11th International Conference on*, p. 1236–1241, IEEE, 2011.

[119] T. AYNAUD, J.-L. GUILLAUME et AL., « Long range community detection », *in LAWDN-Latin-American Workshop on Dynamic Networks*, 2010.

[120] T. AYNAUD et J.-L. GUILLAUME, « Multi-step community detection and hierarchical time segmentation in evolving networks », *in Fifth SNA-KDD Workshop Social Network Mining and Analysis, in conjunction with the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2011)*, 2011.

[121] T. YANG, Y. CHI, S. ZHU, Y. GONG et R. JIN, « Detecting communities and their evolutions in dynamic social networks ? a bayesian approach », *Machine learning*, vol. 82, no. 2, p. 157–189, 2011.

[122] B. MITRA, L. TABOURIER et C. ROTH, « Intrinsically dynamic network communities », *Computer Networks*, vol. 56, no. 3, p. 1041–1053, 2012.

[123] L. GAUVIN, A. PANISSON et C. CATTUTO, « Detecting the community structure and activity patterns of temporal networks : a non-negative tensor factorization approach », *PloS one*, vol. 9, no. 1, p. e86028, 2014.

[124] J. LESKOVEC, J. KLEINBERG et C. FALOUTSOS, « Graphs over time : densification laws, shrinking diameters and possible explanations », *in Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, p. 177–187, ACM, 2005.

[125] A. LANCICHINETTI, S. FORTUNATO et F. RADICCHI, « Benchmark graphs for testing community detection algorithms », *Physical Review E*, vol. 78, no. 4, p. 046110, 2008.

[126] J. LESKOVEC et A. KREVL, « SNAP Datasets : Stanford large network dataset collection ». `http://snap.stanford.edu/data`, juin 2014.

[127] W. W. ZACHARY, « An information flow model for conflict and fission in small groups », *Journal of anthropological research*, p. 452–473, 1977.

[128] C. LEE et P. CUNNINGHAM, « Benchmarking community detection methods on social media data », *arXiv preprint arXiv :1302.0739*, 2013.

[129] M. NEWMAN, A.-L. BARABÁSI et D. J. WATTS, *The structure and dynamics of networks*. Princeton University Press, 2006.

[130] J. LESKOVEC, K. J. LANG, A. DASGUPTA et M. W. MAHONEY, « Community structure in large networks : Natural cluster sizes and the absence of large well-defined clusters », *Internet Mathematics*, vol. 6, no. 1, p. 29–123, 2009.

[131] L. ŠUBELJ et M. BAJEC, « Robust network community detection using balanced propagation », *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 81, no. 3, p. 353–362, 2011.

[132] U. GARGI, W. LU, V. S. MIRROKNI et S. YOON, « Large-scale community detection on youtube for topic discovery and exploration. », *in ICWSM*, 2011.

[133] A. CLAUSET, C. R. SHALIZI et M. E. NEWMAN, « Power-law distributions in empirical data », *SIAM review*, vol. 51, no. 4, p. 661–703, 2009.

[134] M. E. NEWMAN, « Power laws, pareto distributions and zipf's law », *Contemporary physics*, vol. 46, no. 5, p. 323–351, 2005.

[135] R. ALBERT et A.-L. BARABÁSI, « Statistical mechanics of complex networks », *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.

[136] A.-L. BARABASI, « The origin of bursts and heavy tails in human dynamics », *Nature*, vol. 435, no. 7039, p. 207–211, 2005.

[137] A. VÁZQUEZ, J. G. OLIVEIRA, Z. DEZSÖ, K.-I. GOH, I. KONDOR et A.-L. BARABÁSI, « Modeling bursts and heavy tails in human dynamics », *Physical Review E*, vol. 73, no. 3, p. 036127, 2006.

[138] D. B. STOUFFER, R. D. MALMGREN et L. A. AMARAL, « Comment on barabasi, nature 435, 207 (2005) », *arXiv preprint physics/0510216*, 2005.

[139] A.-L. BARABÁSI, K.-I. GOH et A. VÁZQUEZ, « Reply to comment on" the origin of bursts and heavy tails in human dynamics" », *arXiv preprint physics/0511186*, 2005.

[140] V. PARETO, *Manuale di economia politica*, vol. 13. Societa Editrice, 1906.

[141] A. ARENAS, L. DANON, A. DIAZ-GUILERA, P. M. GLEISER et R. GUIMERA, « Community analysis in social networks », *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 38, no. 2, p. 373–380, 2004.

[142] S. N. DOROGOVTSEV et J. F. MENDES, « 14 accelerated growth of networks », *Handbook of Graphs and Networks : From the Genome to the Internet*, 2006.

[143] J. XIE, *Agent-based dynamics models for opinion spreading and community detection in large-scale social networks*. Thèse doctorat, Rensselaer Polytechnic Institute, 2012.

[144] R. I. DUNBAR, « Neocortex size as a constraint on group size in primates », *Journal of Human Evolution*, vol. 22, no. 6, p. 469–493, 1992.

[145] A. L. TRAUD, C. FROST, P. J. MUCHA et M. A. PORTER, « Visualization of communities in networks », *Chaos*, vol. 19, no. 4, p. 041104, 2009.

[146] T. FRUCHTERMANN et E. REINGOLD, « Graph drawing by force-directed placement », *Software, Practice and Experience*, no. 21, p. 1129–1164, 1991.

[147] T. KAMADA et S. KAWAI, « An algorithm for drawing general undirected graphs », *Information Processing Letters*, no. 31, p. 7–15, 1988.

[148] A. ALESINA, A. DEVLEESCHAUWER, W. EASTERLY, S. KURLAT et R. WACZIARG, « Fractionalization », *Journal of Economic growth*, vol. 8, no. 2, p. 155–194, 2003.

[149] D. GALE et L. S. SHAPLEY, « College admissions and the stability of marriage », *American mathematical monthly*, p. 9–15, 1962.

[150] R. W. IRVING, D. F. MANLOVE et G. O'MALLEY, « Stable marriage with ties and bounded length preference lists », *Journal of Discrete Algorithms*, vol. 7, no. 2, p. 213–219, 2009.

[151] D. F. MANLOVE, R. W. IRVING, K. IWAMA, S. MIYAZAKI et Y. MORITA, « Hard variants of stable marriage », *Theoretical Computer Science*, vol. 276, no. 1, p. 261–279, 2002.