
Détection de communautés statiques et dynamiques

Ahmed Ould Mohamed Moctar et Idrissa Sarr

*Université Cheikh Anta Diop, Département de Mathématiques et Informatique
Dakar - Fann BP 5005, Sénégal
{ahmed.ouldmoctar, idrissa.sarr}@ucad.edu.sn*

RÉSUMÉ. La découverte de groupes cohésifs, de cliques ou de communautés à l'intérieur d'un réseau fait partie des sujets les plus étudiés dans l'analyse des réseaux sociaux. Elle a attiré de nombreux chercheurs en sociologie, biologie, informatique, physique, criminologie, etc. Cependant, une bonne partie des études faites dans ce domaine s'est intéressée à la détection de communautés dans les réseaux statiques laissant souvent de côté les réseaux dynamiques et encore plus ceux temporels. Pourtant, dans ces derniers cas, des aspects tels que l'évolution et parfois même, l'origine de la formation de ces communautés sont des problématiques intéressantes à étudier. Compte tenu de la prolifération des réseaux sociaux issus des données des applications mobiles, force est de reconnaître que l'étude des communautés dynamiques devient un enjeu de taille puisque celles-ci sont parfois volatiles et/ou évolutives. L'objectif de cet article est de faire un survol actualisé de l'état de l'art sur la détection de communautés statiques et dynamiques. De plus, il sera question de souligner les limites et de relever quelques paramètres à intégrer pour mieux étudier les communautés dynamiques.

ABSTRACT. The discovery of cohesive groups, cliques and communities within a network is one of the most studied topics in social network analysis. It attracted many researchers in sociology, biology, computer science, physics, criminology, etc. However, most of the studies in this area have focused on static networks while leaving aside the dynamic networks and even more those temporal. However, in these cases, issues such as evolution, the structure change and sometimes even behind the formation of these communities are interesting issues to address. Given the proliferation of social networks from the spatio-temporal data, we have to point out that the study of dynamic communities becomes a paramount challenge the since they are sometimes volatile and/or mutable. The objective of this article is to reviewing studies on communities detection. Afterwards, we highlight the limits of most communities detection approaches in order to point out how we can deal with them.

MOTS-CLÉS : analyse des réseaux sociaux, détection de communautés, réseaux dynamiques.

KEYWORDS: social networks analysis, community detection, dynamic networks.

DOI:10.3166/RIA.30.469-496 © 2016 Lavoisier

1. Introduction

Avec la prolifération des applications sociales et mobiles, les utilisateurs sont en interaction de manière constante à travers des partages de documents, de photos/videos, de messages, etc. Ainsi, il convient de noter que les réseaux sociaux, qui peuvent être formés par les individus et/ou leur contenu, sont par nature dynamiques et ne pourraient être représentés statiquement sans perdre de l'information. Pour intégrer ce fait, il est possible de voir les réseaux comme étant dynamiques et de les modéliser par un graphe sur lequel des changements sont introduits au fil du temps.

La détection de communautés vise à trouver des sous-graphes au sein d'un réseau donné. C'est une approche, à la fois graphique et analytique, permettant de décomposer le réseau en plusieurs groupes de nœuds ayant des intérêts similaires ou partageant des propriétés communes. Dans cette perspective, une communauté peut désigner un groupe de nœuds densément connectés entre eux et légèrement liés avec le reste du réseau. L'une des premières études sur la détection de communautés date de 1955 et a été effectuée par (Weiss, Jacobson, 1955) dans le but de rechercher des groupes de travail dans une agence de gouvernement. L'idée de base est de trouver un découpage du réseau de sorte que le nombre d'arêtes à l'intérieur de chaque groupe soit considérablement supérieur au nombre d'arêtes sortantes. Ainsi, chaque communauté identifiée présente une certaine cohésion interne et une faible connectivité avec les autres communautés.

L'identification de communautés est avantageuse dans de nombreux domaines. Par exemple, elle est utilisée dans l'amélioration des stratégies de marketing (Weng *et al.*, 2013) car un individu aura tendance à s'intéresser à un produit/service si la quasi-totalité de ses amis s'y intéresse. De même, la structure d'une communauté révèle une gamme d'informations pouvant être utilisée pour comprendre et prédire les caractéristiques de la diffusion de certaines épidémies (Yulian, 2011).

Aujourd'hui, on décompte un grand nombre de travaux relatifs à la détection de communautés. De plus, il existe beaucoup d'études qui ont fait la revue des travaux les plus significatifs dans le domaine. Toutefois, il est à noter que la plupart des algorithmes de détection de communautés reposent entièrement sur les propriétés statiques du réseau et négligent souvent le fait que les réseaux évoluent au fil du temps. Ainsi, le nombre de travaux consacrés à la détection de communautés statiques est largement supérieur à celui visant les communautés dynamiques. Pourtant, les interactions entre les nœuds changent constamment dans le temps et se traduisent par l'insertion et/ou la suppression de nœuds ou de liens ayant un fort impact sur la structure des communautés.

Si le déséquilibre entre les études consacrées aux communautés statiques et les travaux relatifs aux communautés dynamiques pouvait être expliqué par l'inexistence d'outils efficaces permettant de prendre en charge les réseaux dynamiques, il est à noter que les technologies permettant de construire des solutions évolutives capables de gérer de grandes masses de données ne cessent d'être démocratisées. Par conséquent, on assiste de plus en plus à une multiplication d'études sur les communautés

dynamiques et les revues sur les solutions proposées devraient également suivre. En d'autres mots, la revue de la littérature sur les communautés statiques est très bien couverte à travers des travaux comme celui de (Fortunato, 2010) ; celle relative aux communautés dynamiques ne l'est pas encore. L'objectif ce papier est de proposer une revue sur les solutions de communautés dynamiques afin de réduire un peu le manque dans ce domaine tout en essayant d'intégrer au mieux les derniers travaux y afférents.

Pour procéder à la revue des solutions, un des premiers postulats est de savoir par où commencer compte tenu de l'abondance de l'existant. C'est pourquoi, nous catégorisons les solutions pour permettre au lecteur de mieux suivre les grandes tendances relatives aux communautés statiques et dynamiques. Cela étant, cette présente revue peut être considérée comme un complément par rapport aux revues du même domaine avec un aperçu couvrant au mieux les dernières solutions. Pour atteindre notre but, nous avons établi une grille de classification composée d'un ensemble de critères (voir section 3). L'application de cette grille sur les revues existantes montrent que tous les critères ne sont pas encore pris en compte et des solutions sont encore à envisager. Enfin, nous précisons d'ores et déjà que l'objet de notre travail n'est pas de décrire chaque solution individuelle du fait que les travaux concernés sont multiples. Ainsi, la démarche entreprise pour des soucis de présentation est de regrouper les solutions existantes par mode de fonctionnement et de faire des commentaires sur ces modes.

La suite de l'article est organisée comme suit. Nous commençons par un historique sur les communautés dans la section 3. Puis, nous présentons dans la section 4 les principales approches de détection de communautés statiques. Ensuite, nous présentons dans la section 5 les différentes méthodes relatives aux communautés dynamiques. Enfin, nous montrons les limites des solutions existantes et surtout des défis à relever dans la section 6 avant de conclure par la section 7.

2. Définitions

Dans cette section, nous définissons quelques concepts de base pour faciliter la compréhension de la suite de cet article.

Graphe. Un graphe est un ensemble d'objets nommés nœuds ou sommets reliés par des arcs qui modélisent les interactions entre objets. On parle de graphe orienté si les liens sont unidirectionnels ou asymétriques. De même, le graphe est non orienté si les liens sont symétriques. Un même couple de nœuds du graphe peut être relié par plusieurs arêtes. Un graphe est dit pondéré si chaque arête est affectée d'un nombre réel positif, appelé poids de cette arête et représente l'intensité des liens entre les nœuds. Enfin, si chaque nœud est lié à tous les autres nœuds, alors le graphe est dit complet.

Clique. Une clique d'un graphe est un sous-ensemble des nœuds de ce graphe dont le sous-graphe induit est complet, c'est-à-dire que deux sommets quelconques de la clique sont toujours adjacents. Une k -Clique est une clique de k nœuds. Deux

k -cliques sont considérées adjacentes si elles partagent $k - 1$ sommets. Une clique est appelée maximale si elle possède le plus grand nombre de nœuds dans un graphe donné.

Communauté dynamique. Une communauté est dite dynamique si sa structure ou sa composition change au fil du temps.

Communauté chevauchante. Une communauté est chevauchante si une partie de ses nœuds appartient simultanément à d'autres communautés.

Marche aléatoire. La marche aléatoire est un modèle mathématique d'un système possédant une dynamique discrète composée d'une succession de pas aléatoires, ou effectués « au hasard ».

3. Aperçu historique sur la détection de communautés

Les réseaux du monde réel sont structurés de sorte que les éléments partageant des propriétés similaires tendent à former des groupes denses qui sont peu connectés entre eux. Réellement, la détection de communautés cherche à créer des sous-graphes en regroupant des nœuds vérifiant certaines contraintes de connectivité. Une des hypothèses que l'on trouve très souvent avec ces algorithmes est que le nombre de clusters ou partitions est connu à l'avance. Toutefois, il est quasi-impossible de connaître à priori le nombre de communautés dans la plupart des réseaux réels. Par conséquent, il convient d'adapter ou d'améliorer ces algorithmes pour relâcher cette forte hypothèse. La définition de communauté révèle deux contraintes discordantes : la cohésion interne et la disjonction avec le reste du réseau. D'une part, les algorithmes utilisant les cliques maximales trouvent les communautés les plus denses mais en contrepartie, les communautés détectées sont très liées au reste du réseau. D'autre part, les algorithmes qui cherchent à détecter des groupes en ne s'appuyant que sur la connectivité des nœuds, obtiennent des communautés relativement denses mais plus isolées. Autrement, il est difficile d'avoir un partitionnement du réseau dont la cohésion et la séparation sont à la fois de bonne qualité. Ainsi, il faut trouver un compromis en fonction des objectifs et des types de communautés recherchées (par exemple chevauchantes ou non).

Le problème de la détection de communautés a connu un regain d'intérêt au début du troisième millénaire avec les travaux de (Girvan, Newman, 2002) qui ont conclu que la structuration en communautés fait partie des caractéristiques des réseaux petits-mondes (Jeffrey Travers, 1969). Au regard du nombre de travaux existants, il s'avère que la détection de communautés constitue une thématique intéressante. C'est pourquoi faire un état de l'art sur le domaine se révèle difficile. Le tableau 1 présente quelques revues sur les travaux relatifs à la détection de communautés. La grille de classification utilisée est composée de quatre critères. Le premier indique quels types de réseaux sont couverts par la revue. Trois sous-critères sont utilisés pour évaluer ce critère, à savoir, l'orientation des liens, leur pondération et le fait que la détection est basée sur un réseau égo-centré ou pas. Le deuxième critère donne un aperçu sur les types de communautés couvertes : statiques, dynamiques et chevauchantes. Le

troisième critère met l'accent sur quatre catégories d'approches que nous avons identifiées pour faire notre état de l'art. Les catégories sont basées sur le principe utilisé pour détecter les communautés et sont : Hiérarchique, Clique, Marche aléatoire et Connectivité. Le dernier critère donne une idée du nombre de travaux étudiés et du niveau de complétude de la littérature couverte par la revue considérée.

Les critères utilisés dans la grille relèvent d'une importance capitale car ils nous permettent de savoir les intrants des approches couvertes par les revues (types de réseaux étudiés), les extrants (natures des communautés détectées), les méthodologies couvertes (algorithmes de détection) et enfin, la quantité de littérature couverte. Ainsi, en fonction du niveau de couverture d'un critère, il devient plus aisé de voir les manquements de la revue et de pouvoir envisager les compléments de lecture nécessaire.

De plus, les revues choisies sont basées sur la chronologie pour montrer comment le sujet a évolué au fil des années. A la lecture du tableau 1, on peut observer qu'il n'y a que deux revues qui prennent en compte tous les critères (Papadopoulos *et al.*, 2012; Fortunato, 2010). Toutefois, en regardant de très près le contenu de ces revues, nous avons remarqué que les communautés dynamiques sont très peu développées et que les études des cinq dernières années ne sont pas prises en compte. Un des défis de ce travail est alors de prendre en compte les solutions les plus récentes pour gérer les communautés dynamiques et de discuter de l'importance des critères non couverts par les revues.

4. Détection de communautés statiques

Nous présentons dans cette section les principales approches de détection de communautés statiques. L'objectif est de reprendre la littérature et regrouper les travaux relatifs à la détection de communautés en quatre catégories.

Les algorithmes que nous avons sélectionnés sont classés dans les catégories d'approches en s'appuyant sur le principe utilisé pour détecter les communautés. Les catégories identifiées sont les approches hiérarchiques, approches basées sur les cliques, approches basées sur les marches aléatoires et les approches basées sur la propagation d'information (labels). La figure 1 montre une représentation des catégories et de dix algorithmes que nous avons retenus dans ce travail compte tenu de l'intérêt qu'ils ont reçus de la part de la communauté scientifique. Il convient de noter que le point commun de tous ces algorithmes est qu'ils sont destinés à des réseaux statiques non pondérés et non orientés et produisent des communautés soient disjointes soient chevauchantes. Dans ce présent travail, nous ne fixons pas d'hypothèses sur le type de réseaux mais nous observerons comment les solutions présentées les prennent en charge.

4.1. Approche hiérarchique

L'approche hiérarchique est basée sur la définition d'une mesure de similarité entre les nœuds du réseau. Elle se divise en deux catégories : divisive et agglomérative. La

Tableau 1. Classification des articles de revue sur la détection de communautés

Références	Types de réseaux			Nature des communautés			Approche			
	Orienté	Pondéré	Ego-centrique	Statique	Dynamique	Chevauchante	Hiéar-chique	Clique	Marche Aléatoire	Connectivité (Diffusion information, ...)
Porter <i>et al.</i> (2009)	X	✓	X	✓	X	✓	✓	✓	X	X
Fortunato (2010)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
B. Yang <i>et al.</i> (2010)	X	✓	X	✓	X	✓	✓	✓	X	X
Coscia <i>et al.</i> (2011)	✓	✓	X	✓	✓	✓	✓	✓	✓	✓
Papadopoulos <i>et al.</i> (2012)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Cazabet (2013)	X	X	X	✓	✓	✓	✓	✓	✓	✓
Harenberg <i>et al.</i> (2014)	✓	✓	X	✓	X	✓	✓	✓	✓	✓
Khattoon, Banu (2015)	✓	✓	X	✓	X	✓	✓	✓	X	X

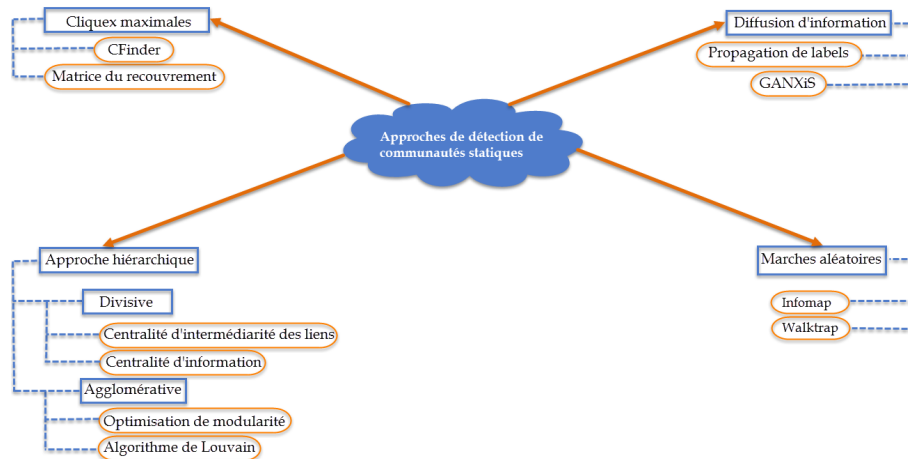


Figure 1. Taxonomie des méthodes de détection de communautés statiques exposées dans cet article

première catégorie est de nature descendante, elle place initialement tous les nœuds dans la même communauté puis retire les liens entre les nœuds de similarité plus faible. Ainsi, la communauté mère se dissocie en plusieurs partitions. Ce processus est répété jusqu'à ce que tout nœud soit dans une seule communauté. Chaque algorithme de cette catégorie définit son propre critère pour supprimer les liens à chaque itération. En revanche, la seconde catégorie est de type ascendant, elle consiste à mettre chaque nœud dans une communauté différente, à ce stade les communautés forment des singletons. Puis à chaque itération, les deux communautés les plus similaires sont regroupées, jusqu'au moment où tous les nœuds sont dans la même communauté. Pour plus de détails, nous renvoyons le lecteur à l'article de synthèse (Nagpal *et al.*, 2013).

4.1.1. Approche divisive

L'approche divisive place initialement tous les nœuds dans une seule communauté, située en haut du dendrogramme, puis cherche à composer le réseau en plusieurs communautés tout en retirant progressivement les arêtes qui relient les nœuds de basse similarité. Petit à petit, la communauté mère se dissocie en plusieurs partitions. Ce processus est répété jusqu'à avoir des communautés composées d'un seul nœud représentant les feuilles du dendrogramme. Dans chaque itération, toute partition connexe est considérée comme communauté. La figure 2, extraite de (Papadopoulos *et al.*, 2012), en donne une illustration.

L'algorithme hiérarchique divisif le plus connu, abrégé souvent par GN pour désigner les auteurs Girvan et Newman, est présenté dans (Girvan, Newman, 2002). Il est l'un des algorithmes les plus référencés dans le domaine de la détection de communautés. GN est basé sur une mesure de centralité d'intermédiarité des liens, définie par le nombre de plus courts chemins passant par un lien. Vu qu'il existe peu de liens entre les communautés, les auteurs considèrent ces liens comme des ponts de passage

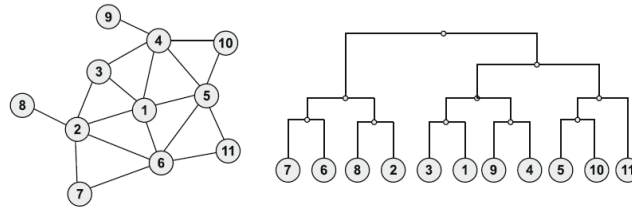


Figure 2. Exemple d'un dendrogramme hiérarchique

permettant de relier rapidement deux communautés différentes. Ces liens se trouvent donc sur un nombre important de plus courts chemins. En effet, l'algorithme GN mesure la centralité d'intermédierité pour un lien (u, v) en s'inspirant de la centralité d'intermédierité d'un nœud¹.

L'algorithme GN est de complexité $\mathcal{O}(n \times m^2)$ avec n le nombre de nœuds et m le nombre de liens du réseau. Cette complexité très élevée rend GN inexploitable pour les réseaux de grande taille. Mais malgré cela, GN a fait l'objet de plusieurs travaux de recherche visant à l'améliorer ou à l'adapter (Wilkinson, Huberman, 2004; Massen, Doye, 2005; Yoon *et al.*, 2006) grâce à ses concepts théoriques robustes et il est également implémenté dans plusieurs outils d'analyse des réseaux complexes (Csardi, Nepusz, 2006; Schult, Swart, 2008).

Une variante de l'algorithme GN est proposée par (Fortunato *et al.*, 2004) et est basée sur une mesure de similarité appelée « centralité d'information ». Les auteurs définissent l'efficacité de communication ε_{ij} entre deux nœuds i et j du graphe comme étant l'inverse de leur distance d_{ij} .

Cet algorithme est plus efficace que GN mais moins rapide car il a une complexité de $\mathcal{O}(n \times m^3)$.

4.1.2. Approche agglomérative

L'idée de base de cette approche est de s'appuyer sur un mécanisme similaire à celle du clustering hiérarchique dans lequel les sommets sont regroupés itérativement en communautés selon leur similarité. A cet effet, l'approche agglomérative part d'une composition atomique du réseau en supposant initialement que chaque nœud du réseau représente une communauté. Initialement, il y a n communautés (où n est le nombre de nœuds) et on commence par calculer les distances entre les communautés et fusionner les deux communautés les plus proches pour former une nouvelle communauté. À chaque étape, on recalcule toutes les distances entre les communautés et on fusionne deux communautés. Lorsqu'il n'y a qu'une seule communauté représentant le graphe entier, il n'existe plus de distance à calculer.

1. La centralité d'intermédierité d'un nœud mesure le nombre de fois où il est l'intermédiaire sur le plus court chemin entre deux autres. La communication des nœuds non adjacents dépend d'autres acteurs qui se trouvent sur le chemin. Tels acteurs ont la capacité d'interrompre le passage d'information. Ainsi, plus un acteur se trouve au « milieu » du chemin, plus il est central de ce point de vue.

Pour savoir si deux communautés sont proches, on s'appuie sur la modularité qui est proposé dans (Newman, 2004). La modularité représente la différence entre la valeur d'adjacence entre deux nœuds d'une même communauté et la probabilité pour que ceux-ci soient connectés. Avec la modularité, une communauté est vue comme un ensemble de nœuds qui ont plus de liens entre eux que de liens avec des nœuds à l'extérieur de la communauté. Par conséquent, deux communautés sont fusionnées que lorsque leur union satisfait la condition précédente. Concrètement, la modularité cherche à maximiser le nombre de liens à l'intérieur des communautés. L'optimisation exacte de modularité sur toutes les partitions du réseau représente un problème NP-complet (Brandes *et al.*, 2006). En général, les algorithmes d'optimisation de modularité cherchent à trouver un découpage de meilleure qualité possible dans un délai raisonnable.

En réalité, les méthodes d'optimisation de modularité sont nombreuses, toutefois, elles adoptent le même principe de base. Un exemple d'optimisation est la méthode de Louvain (Blondel *et al.*, 2008) et est très bien utilisée par la communauté scientifique.

La méthode de Louvain est à la base de plusieurs applications des réseaux complexes (Haynes, Perisic, 2009; Hui, Sastry, 2009; Roma, Herrera, 2010) ainsi que de plusieurs algorithmes de détection de communautés dynamiques (Greene *et al.*, 2010; Mucha *et al.*, 2010).

4.2. *Marches aléatoires*

Dans leur ensemble, les algorithmes de cette approche tournent autour de la même idée : la marche aléatoire sert à calculer la distance entre les nœuds puis une fonction de qualité est utilisée pour choisir le partitionnement optimal du réseau. Dans ce papier, nous présentons deux algorithmes utilisant les marches aléatoires : Infomap et Walktrap. Chaque algorithme se distingue par la façon dont il mesure cette distance et le choix de la fonction du découpage.

Infomap (Rosvall *et al.*, 2009) constitue l'un des algorithmes les plus connus dans cette approche et permet de trouver un partitionnement statique et sans recouvrement. Cet algorithme est basé sur la minimisation de Map Equation (Rosvall, Bergstrom, 2008) qui transforme le problème de détection de communautés à celui d'une compression d'information. En effet, InfoMap s'appuie sur le principe du marcheur aléatoire qui a tendance à rester piégé dans une communauté vu que les liens au sein de la communauté sont aussi nombreux que ceux situés à l'extérieur. L'avantage principal de cette approche est sa capacité de préciser si le réseau est divisible en communautés ou non. Autrement dit, si on prend un réseau aléatoire et que l'on n'arrive pas à localiser le marcheur pendant un bon moment dans une zone précise du réseau, alors, on peut conclure qu'il n'y a pas de communautés pertinentes.

Un autre algorithme qui s'appuie également sur les marches aléatoires est Walktrap (Pons, Latapy, 2005) représente également un autre algorithme, sauf que celui-ci fait recours à un algorithme agglomératif pour partitionner le réseau.

4.3. Propagation de labels

L'algorithme de propagation de labels (Raghavan *et al.*, 2007) s'inspire du mécanisme de diffusion d'information au sein des réseaux. L'idée générale est que l'information se propage rapidement dans les groupes de nœuds fortement interconnectés. Ainsi, chaque nœud rejoint la communauté à laquelle appartient la majorité de ses voisins. Le principe de cet algorithme se résume en trois étapes :

1. un label différent est attribué à chaque nœud ;
2. de façon itérative, chaque nœud du réseau remplace son label par celui utilisé par le plus grand nombre de ses voisins (le label est sélectionné aléatoirement en cas d'égalité) ;
3. petit à petit, l'ensemble de nœuds densément liés tend à utiliser le même label. Ainsi, les nœuds partageant le même label constituent une communauté.

L'algorithme de propagation de labels a fait l'objet de plusieurs propositions (Gregory, 2010; Leung *et al.*, 2009; Xie *et al.*, 2011). La proposition la plus récente est appelée GANXiS (Xie, Szymanski, 2012). Les auteurs se basent sur la version traditionnelle de (Raghavan *et al.*, 2007) pour concevoir un algorithme capable de détecter les communautés disjointes et chevauchantes. La principale nouveauté de GANXiS réside dans la mise en place d'une mémoire servant à stocker, pour chaque nœud, les labels qui lui ont été attribués. Par conséquent, le degré d'appartenance aux communautés est mesuré par la probabilité de trouver plusieurs labels dans la mémoire d'un nœud, et non pas seulement l'état des labels lors de la dernière itération.

L'algorithme GANXiS adopte le principe de « speaker/listener » éclairci comme suit : à chaque itération, un nœud est considéré comme « listener » et ses voisins sont vus comme « speakers ». Initialement, la mémoire de chaque nœud est configurée avec un label unique. Ainsi, chaque nœud appartient à une communauté différente. Ensuite, de manière itérative, le label en question se déplace de « speaker » au « listener ». Chaque label propagé est choisi au hasard avec une probabilité proportionnelle à sa fréquence dans la mémoire du « speaker » qui l'a envoyé. Le « listener » adopte le label qui figure le plus dans sa mémoire. Ce processus de propagation de labels est répété N fois, avec N est défini par l'utilisateur. Enfin, une distribution de probabilités des labels est obtenue pour la mémoire de chaque nœud. Un label est supprimé si sa probabilité d'apparence est inférieure à un seuil donné $\tau \in [0, 1]$. Les nœuds de même label sont considérés comme communauté. Le cas de chevauchement se présente si un nœud possède plusieurs labels. Il est à noter que plus la valeur de τ est élevée, plus le nombre de communautés chevauchantes augmente (lorsque $\tau \leq 0.5$, l'algorithme détecte uniquement des communautés disjointes).

4.4. Cliques maximales

Cette approche consiste à construire les communautés à partir de cliques. Par conséquent, une communauté est composée d'un ensemble de nœuds très fortement liés entre eux. Les cliques maximales sont essentiellement idéales dans le cas où les

interactions se passent de façon transitive, c'est-à-dire, si deux nœuds a et b sont liés au nœud c , alors a et b auront tendance à être liés entre eux. Ainsi, les communautés sont riches en cliques. L'un des avantages de cette approche est qu'elle permette de détecter des communautés tout en se passant de la forte contrainte d'une clique qui exige un sous-graphe complet. L'un des exemples les plus utilisés est la Clique Percolation Method (CPM) et est l'une des méthodes de recouvrement les plus connues. CFinder (Palla *et al.*, 2005) représente une implémentation de CPM, considérant une communauté comme une chacune de k -cliques adjacentes. L'avantage principal de cette approche est la détection de communautés chevauchantes, un sommet pouvant appartenir à plusieurs k -cliques non forcément adjacentes. Une limite de cette méthode est qu'elle nécessite un paramétrage: la valeur de k , qui représente la taille des cliques à considérer. La modification de cette valeur influe fortement sur les résultats trouvés. Par défaut, l'algorithme cherche toutes les valeurs possibles de k et donne à l'utilisateur la possibilité de choisir la valeur qui lui semble plus utile. L'augmentation de k implique la densité de communautés trouvées. Cependant, une fois k choisi, les communautés de $(k - 1)$ -cliques ne seront pas considérées. La figure 3 présente un exemple de communautés détectées par CFinder, où les communautés considérées représentent chacune une 4-clique. Donc, toute composition de sommets ne vérifiant pas cette condition ne sera pas affichée.

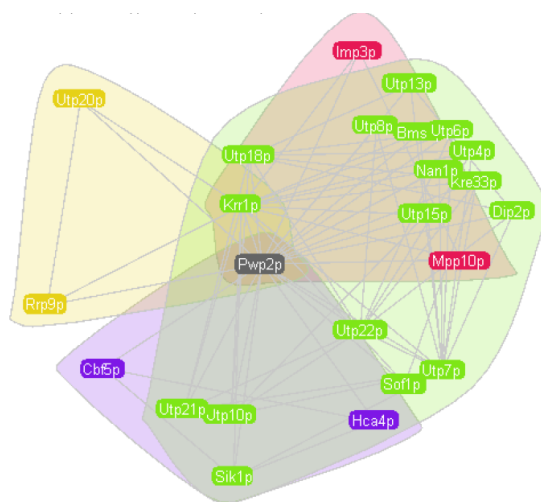


Figure 3. Ensemble de communautés détectées pour $k = 4$ dans un réseau d'interactions protéine-protéine

Dans (Li *et al.*, 2014), les auteurs proposent d'utiliser une technique, appelée matrice de chevauchement, pour extraire des communautés de k -cliques. Comme illustré à la figure 4, extraite de (Li *et al.*, 2014), l'image A montre un graphe de 6-cliques, marquées par des couleurs différentes dans la matrice de chevauchement de l'image B. Par exemple, pour extraire les communautés de 4-cliques, nous remplaçons par zéro les éléments au-dessus de la diagonale étant inférieurs à 3 ainsi que ceux en-dessous de diagonale étant inférieurs à 4. Le résultat de cette opération est représenté

sous forme de matrice de recouvrement dans l'image C. Les composantes connexes (communautés de k -cliques) correspondant à cette matrice sont affichées dans l'image D.

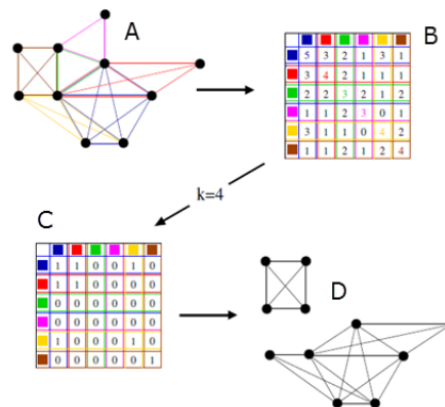


Figure 4. Illustration du fonctionnement de matrice de chevauchement

L'avantage principal de cette technique est que la matrice de chevauchement code toutes les informations nécessaires pour obtenir les communautés pour toute valeur de k . Ainsi, après avoir construit la matrice, les communautés de k -cliques peuvent être obtenues rapidement. Contrairement à la version traditionnelle de CPM où, pour toute valeur de k , l'algorithme cherche, de nouveau, à extraire les composantes de k -cliques.

Pour conclure, des dizaines d'autres algorithmes ont été proposés, certains ne rentrent pas dans les catégories précédemment citées. Sans prétendre à l'exhaustivité, nous pouvons citer les algorithmes suivants :

- **Partitionnement spectral** (Slanina, Zhang, 2005; Capocci *et al.*, 2005; Mitrović, Tadić, 2009) qui se base sur les propriétés des valeurs propres de la matrice laplacienne du graphe pour partitionner le réseau.

- **Mesures de similarité** (Gori *et al.*, 2007; Tong *et al.*, 2008) qui regroupe tous les algorithmes qui considèrent qu'une communauté est composée d'un ensemble de nœuds partageant des propriétés communes ou des caractéristiques similaires.

- **Modèles de spins** (Reichardt, Bornholdt, 2004; Son *et al.*, 2006) qui sont des modèles de physique statistique, utilisés dans la détection de communautés pour modéliser les interactions entre les nœuds du réseau (vus comme des particules, nommés spins).

Enfin, le tableau 2 résume les limites des solutions statiques vues dans cet article. Sur ce tableau, n est le nombre de nœuds du réseau. T est une valeur définie par l'utilisateur et représente le nombre de fois que le processus de propagation est répété. m est le nombre d'arêtes du réseau et k le nombre de communautés à détecter. A la lecture du tableau 2, nous observons qu'aucun algorithme ne couvre simultanément tous

les critères retenus. En effet, la plupart des algorithmes présentés détectent uniquement des communautés disjointes négligeant le fait que les communautés sont dans la réalité de nature chevauchante. Pour pallier cette insuffisance, des études ont été menées lors des dernières années pour détecter le chevauchement entre les communautés. Toutefois, il reste encore le fait que celles-ci ne prennent pas en compte les réseaux orientés (GANXiS) et/ou pondérés (Matrice de recouvrement).

Un autre problème plus troublant des algorithmes statiques est leur instabilité. De fait, un algorithme est dit instable s'il trouve des communautés fortement différentes sur deux graphes topologiquement proches. Certes, cette faiblesse est encore plus évidente pour les algorithmes non déterministes². Par conséquent, la réutilisation des algorithmes statiques pour détecter des communautés dynamiques introduit un biais car ils ne permettent pas de savoir s'il y a évolution d'une communauté entre deux instants ou si c'est leur instabilité qui laisse apparaître un changement.

5. Détection de communautés dynamiques

Comme mentionné dans l'introduction, la détection de communautés nécessite la prise en compte de deux aspects :

1. Un algorithme pour détecter et suivre l'évolution de communautés ;
2. Un algorithme pour visualiser les changements survenus au sein des communautés au fil du temps.

Les méthodes de représentation de communautés dynamiques sont d'une importance capitale dans le processus de détection de communautés car elle permettent de visualiser l'évolution au fil du temps. Des travaux ont été menés pour prendre en compte les deux aspects de la détection. Nous présentons dans la suite les méthodes de visualisation des réseaux dynamiques avant d'aborder les approches de détection/suivi des communautés dynamiques. Nous précisons que l'objectif de cette section n'est pas de faire l'état de l'art sur la représentation des communautés mais d'en donner un bref aperçu pour permettre au lecteur de percevoir l'apport de la visualisation dans le processus de détection qui est le cœur de cet article.

5.1. Représentation des réseaux dynamiques

La représentation des réseaux dynamiques soulève des défis majeurs compte tenu des contraintes hétérogènes entre la nature dynamique des réseaux, d'une part, et le comportement statique des graphes, d'autre part. En effet, les réseaux sont représentés habituellement sous forme de graphes. Dans cette approche, l'évolution du réseau est découpée en plusieurs instantanés dont chacun est représenté via un graphe statique. La figure 5 montre un exemple de représentation à travers un graphe. Deux nœuds sont

2. Un algorithme est dit non déterministe s'il détecte des communautés différentes en l'exécutant de manière répétitive sur le même jeu de données.

Tableau 2. Critique de quelques algorithmes de détection de communautés statiques

Algorithme	Orienté	Pondéré	Complexité	Exigence de paramètres	Déterminisme	Recouvrement	Référence
GN	✓	✓	$\mathcal{O}(nm^2)$	X	✓	X	(Girvan, Newman, 2002)
Modularité de Newman	✓	✓	$\mathcal{O}(mk \log n)$	✓	X	X	(Newman, 2004)
Centralité d'information	✓	X	$\mathcal{O}(n \times m^2)$	✓	X	X	(Fortunato <i>et al.</i> , 2004)
Walktrap	X	✓	$\mathcal{O}(mn^2)$	✓	✓	X	(Pons, Latapy, 2005)
CFinder	X	X	$\mathcal{O}(exp(n))$	✓	✓	✓	(Palla <i>et al.</i> , 2005)
Propagation de labels	X	✓	$\mathcal{O}(n+m)$	X	X	X	(Raghavan <i>et al.</i> , 2007)
Louvain	✓	X	$\mathcal{O}(m)$	X	X	X	(Blondel <i>et al.</i> , 2008)
Infomap	✓	✓	$\mathcal{O}(m)$	X	X	X	(Rosvall <i>et al.</i> , 2009)
GANXiS	X	✓	$\mathcal{O}(Tm)$	✓	X	✓	(Xie, Szymanski, 2012)
Matrice de chevauchement	✓	X	$\mathcal{O}(nm)$	X	✓	✓	(Li <i>et al.</i> , 2014)

reliés s'ils partagent une relation temporelle respectant des contraintes et l'ensemble du graphe constitue un graphe d'accessibilité dynamique ou « *Temporal Reachability Graphs (TRG)* ». Les liens peuvent être bidirectionnels ou unidirectionnels.

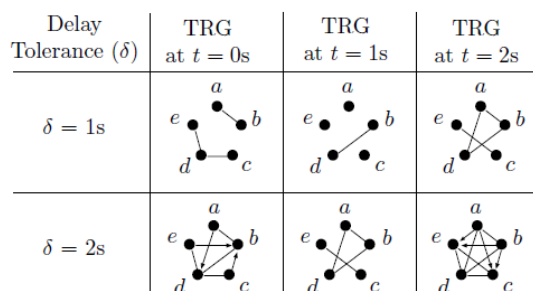


Figure 5. Représentation des différents instantanés d'un réseau dynamique en utilisant les graphes d'accessibilité dynamiques, d'après (Whitbeck et al., 2012)

Toutefois, les graphes sont par définition des objets statiques et supportent mal l'aspect dynamique. Pour faire face à ce problème, une représentation incrémentale par flots de liens a été proposée.

Dans cette approche, l'aspect évolutif du réseau est directement intégrée et de manière continue dans un espace à deux dimensions. Les nœuds sont placés sur l'axe des ordonnées et leurs liens temporels entre deux sont représentés par des arcs au fil du temps. La figure 6 en donne un aperçu. L'avantage de cette représentation est qu'elle permet de suivre comment évolue la relation entre deux nœuds et permet de visualiser tous les nœuds qui communiquent ensemble durant une période de temps. Cela étant, les communautés sont plus faciles à identifier comme l'illustre la figure 6 où chaque communauté est colorée différemment.

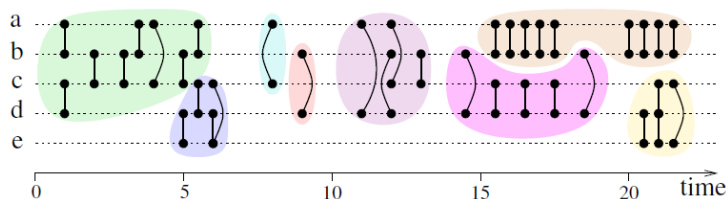


Figure 6. Schématisation d'un flot de liens à 5 sommets interagissant entre eux pendant 20 minutes. figure extraite de (Heymann, 2013)

En somme, les deux approches précédemment citées sont conceptuellement différentes. Les flots de liens schématisent les traces des activités d'un réseau au cours du temps de sorte qu'un lien apparaisse entre deux nœuds lorsqu'ils entrent en communication. Cette approche est donc essentiellement adéquate pour les interactions de courte durée et très souvent répétées, comme le cas des échanges par courrier électronique. En revanche, les méthodes par instantanés conviennent plutôt à l'analyse de « relations » qui perdurent dans le temps, telle que la propagation d'épidémies.

5.2. Opérations d'évolution de communautés

La démarche de détection de communautés dynamiques consiste à chercher les communautés significatives à chaque étape de l'évolution du réseau qui peut se manifester de plusieurs manières. L'un des premiers travaux ayant traité le suivi des communautés est celui décrit dans (Palla *et al.*, 2007). Dans ces travaux, les auteurs définissent six transformations possibles de communautés, à savoir, la naissance, la mort, la croissance, la contraction, la fusion et la division (voir figure 7). Des opérations similaires sont également identifiées dans Asur *et al.*, 2009; Greene *et al.*, 2010.

Dans (Bródka *et al.*, 2013), les auteurs identifient sept évolutions possibles des communautés dynamiques: le rétrécissement, l'expansion, le regroupement, la scission, l'apparition, la disparition et la monotonie. Notons qu'à part la dernière, les six autres sont équivalentes à celles définies par (Palla *et al.*, 2007). De plus, (Cazabet, 2013) ajoute une nouvelle opération, appelée « résurgence », où la communauté vit un état d'extinction temporaire entre deux instantanés non successifs.

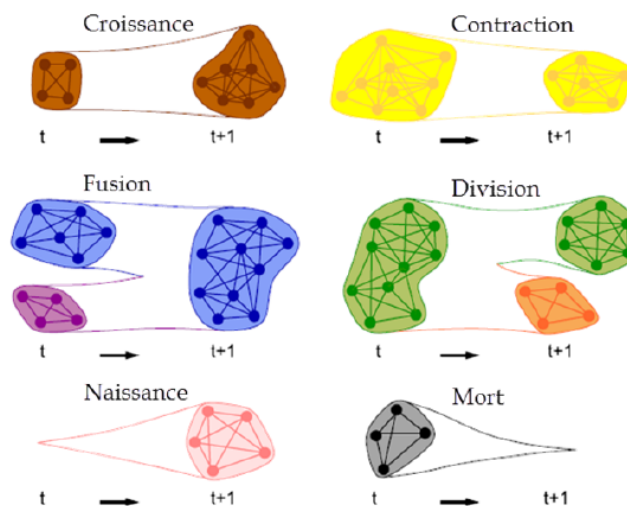


Figure 7. Opérations d'évolution de communautés dans les réseaux dynamiques.
Figure extraite de (Palla *et al.*, 2007)

Concrètement, les opérations d'évolution de communautés dynamiques peuvent être représentées comme suit.

- **Croissance** : la communauté grandit en intégrant des nouveaux nœuds/liens ;
- **Contraction** : la communauté rétrécit en perdant quelques nœuds/liens ;
- **Fusion**: plusieurs communautés se regroupent en une seule ;
- **Division** : la communauté se divise en plusieurs petites communautés ;
- **Naissance** : une composition de nœuds/liens forme une nouvelle communauté à l'instant actuel ;

– **Mort** : la communauté existant à l’instant précédent, disparaît à l’instant actuel. La disparition d’une communauté peut avoir lieu dans deux cas: soit tous les nœuds de la communauté disparaissent, comme le cas de communautés liées aux saisons, soit les liens entre les nœuds disparaissent à un moment donné et on parle alors de communautés événementielles ;

– **Monotonie** : la communauté garde la même composition, de nœuds/liens, d’un instantané à l’autre ;

– **Résurgence** : la communauté disparaît à un moment donné, puis réapparaît quelques temps plus tard sous une forme identique ou très proche. Tel est le cas des communautés périodiques. Concernant la non-équivalence entre la « résurgence » et la « mort+naissance », il convient de noter qu’au niveau de la « résurgence »:

- La mort et la re-naissance de communauté ne sont pas successives.
- Il peut y avoir des changements structurels dans la communauté reparue (surtout en ce qui concerne la distribution de liens).

En effet, les opérations d’évolution de communautés les plus basiques sont la naissance et la mort. Puis, nous avons la croissance et la contraction, qui sont plus difficiles à identifier. En revanche, les opérations de fusion et de division sont plus complexes car elles se comportent différemment selon le domaine d’application. Notons que rares sont les algorithmes qui détectent efficacement de telles transformations.

5.3. *Suivi de communautés*

Cette section a pour but d’expliquer les principes des approches existantes de suivi de communautés dynamiques au fil du temps. Rappelons que le suivi de communautés dynamiques revient à détecter et à interpréter leur évolution. Les méthodes qui visent à gérer l’aspect évolutif sont nombreuses. Nous discutons dans cette section les principales approches de détection de communautés dynamiques tout en mettant l’accent sur les avantages et les inconvénients de chacune.

5.3.1. *Approche par instantanés uniformes successifs*

Comme la détection de communautés statiques semble arrivée à maturité, la quasi-totalité des chercheurs ont eu l’idée de réutiliser les concepts des solutions statiques dans le cas dynamique. Ce qui a fait l’objet de plusieurs tentatives d’adaptation des algorithmes statiques aux réseaux dynamiques (Hopcroft *et al.*, 2004; Palla *et al.*, 2007; Wang *et al.*, 2008; Rosvall, Bergstrom, 2010; Chen *et al.*, 2010; Greene *et al.*, 2010). L’idée générale est de considérer un réseau dynamique comme un ensemble de succession de réseaux statiques, appelés « instantanés », représentant l’image du réseau dynamique à un instant donné. Le principe de cette approche s’articule donc autour de trois étapes. Tout d’abord, décomposer l’évolution du réseau en plusieurs instantanés, ensuite, appliquer un algorithme statique à chaque instantané. Enfin, faire correspondre les communautés trouvées dans un instantané avec celles de l’instantané précédent. La figure 8, extraite de (Aynaud *et al.*, 2013), montre trois instantanés d’un réseau dynamique avec une association entre les communautés des différentes étapes.

L'instantané t contient uniquement deux communautés colorées respectivement en bleu et en vert. Le communauté de couleur bleu à l'instantané $t + 1$ se divise en deux sous-communautés colorées en bleu et en rouge tandis que la communauté de couleur vert reste intacte. A l'instantané $t + 2$, la communauté de couleur bleu reste la même, celle en rouge et en vert se rétrécissent et il apparait une nouvelle communauté de couleur bleu clair.

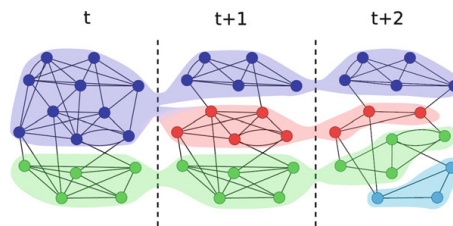


Figure 8. Illustration de l'approche par instantanés uniformes successifs

En effet, nous pouvons dire qu'en termes de discrétisation/continuité du temps, il n'existe que deux approches, celle d'instantanés et celle dynamique. Les autres méthodes reprennent l'idée de l'approche par instantanés en ajoutant ou modifiant quelques notions pour réduire l'instabilité des algorithmes statiques. Dans les paragraphes suivants, nous abordons deux approches qui se basent sur le même principe d'instantanés mais l'interprètent différemment.

5.3.2. Approche par instantanés consécutifs, deux à deux interdépendants

Cette approche s'inspire de la précédente (Chan *et al.*, 2009; Lin *et al.*, 2009; Lancichinetti *et al.*, 2009; Xu *et al.*, 2011). Toutefois, dans la perspective de résolution de l'instabilité des algorithmes statiques, les résultats obtenus à l'instantané T sont pris en compte à l'instantané $T+1$ en vérifiant si les communautés obtenus sont similaires ou en cohérence avec celles obtenues dans les fenêtres antérieures. Ce faisant, si un algorithme par son instabilité produit à un instant des résultats, alors, il devient possible de prendre les bonnes décisions au regard des observations antérieures. La différence entre l'approche précédente et celle-ci se manifeste dans la phase d'interprétation de communautés : l'approche par instantanés uniformes successifs effectue la détection de communautés sur chaque instantané indépendamment des autres fenêtres de temps. Alors que l'approche par instantanés consécutifs profite de communautés détectées à l'instantané T pour mieux identifier celles que l'on devrait avoir à l'instantané $T+1$.

5.3.3. Approche par instantanés multi-pas

Analyser chaque instantané séparément pose problème à cause de l'instabilité des algorithmes statiques de détection de communautés. Une extension de l'approche par instantanés est de ne pas se focaliser sur la stabilité temporaire du partitionnement mais plutôt sur la qualité globale du découpage à long terme (Tantipathananandh *et al.*, 2007; Jdidia *et al.*, 2007; Aynaud, Guillaume, 2010; Mucha *et al.*, 2010; T. Yang *et al.*, 2011). Ainsi, on n'impose pas à une communauté à l'instantané $T+1$ d'être proche

de celle précédente mais d'être pertinente à l'instantané $T+1$. Cela entraîne implicitement la stabilité en cherchant des communautés cohérentes sur plusieurs fenêtres de temps, consécutives ou pas, et non pas seulement sur deux pas de temps successifs. Sur la figure 9, l'image A présente trois instantanés d'un réseau dynamique, l'image B révèle les communautés sur chaque instantané et l'image C représente les communautés finales.

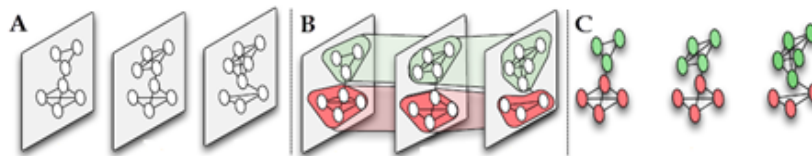


Figure 9. Illustration de l'approche par instantanés multi-pas. Extraite de (Cazabet, 2013) avec une légère modification

Cette approche est pertinente dans le cas où nous nous intéressons à la détection de communautés à long terme. Au lieu de chercher des communautés à chaque instant, il convient de chercher celles qui sont plutôt bonnes sur une longue durée. Un réseau de propagation des maladies, où la contagion se déroule dans la durée en est une illustration.

5.3.4. Approche incrémentale

Cette approche travaille directement sur le réseau dynamique, de ce fait, elle ne considère pas le réseau comme une succession d'instantanés, mais plutôt comme une suite de modifications sur le même réseau (Falkowski *et al.*, 2008; Cazabet *et al.*, 2010; Cazabet, Amblard, 2011; Li *et al.*, 2012; Shang *et al.*, 2014).

L'approche incrémentale vise donc à analyser un flux de données plutôt qu'un instantané. L'entrée est alors une séquence d'événements sur le réseau et l'algorithme essaie de maintenir une décomposition de qualité en mettant à jour sa décomposition courante plutôt qu'en recalculant une décomposition à partir de zéro. Ceci est particulièrement utile dans le cas d'une analyse en temps réel.

En effet, si les mises à jour sont gérées efficacement, l'approche incrémentale aura bien des performances meilleures qu'un calcul de communautés à chaque pas de temps indépendamment. La manière dont la mise à jour est faite, influe fortement sur les résultats obtenus. Nous discutons dans le paragraphe suivant les avantages et les inconvénients de l'approche incrémentale par rapport à l'approche par instantanés.

5.3.5. Avantages et inconvénients des approches précédentes

L'avantage certain de l'utilisation de l'approche par instantanés est de mettre à disposition énormément d'outils, sur lesquels beaucoup de chercheurs travaillent depuis longtemps. En revanche, cette approche souffre d'une faiblesse intrinsèque : sur deux graphes très proches topologiquement, le même algorithme peut trouver des communautés fortement différentes. Les modifications obtenues ne sont donc pas liées

aux changements structurels mais plutôt au caractère non déterministe de l'algorithme employé.

De plus, l'utilisation de l'approche par instantanés entraîne la manipulation de plusieurs instantanés. Donc, si le réseau est de grande taille, le stockage des instantanés peut présenter un problème, vu le volume considérable de l'ensemble des instantanés. Cependant, si nous parvenons à stocker l'intégralité des instantanés du réseau, nous aurons l'avantage d'avoir l'historique de détection de communautés aux différents moments, ce qui n'est pas le cas pour l'approche incrémentale.

Concernant la gestion des instantanés, la définition formelle de fenêtre de temps joue un rôle important dans la détection de communautés. Une fenêtre de temps peut être considérée comme l'image statique du réseau à un instant donné et uniquement à cet instant. Dans ce cas, il est alors possible de perdre beaucoup de données importantes comme les modifications topologiques qui se déclenchent et se terminent dans une période entre deux instants successifs. Une autre définition plus réaliste de fenêtre de temps consiste à considérer que l'instantané actuel reprend les modifications de là où l'instantané précédent s'est arrêté. Une telle définition a l'avantage de garder une vision contiguë de l'ensemble de modifications du réseau.

En somme, que ce soit l'approche par instantanés ou l'approche incrémentale, une question primordiale reste la même : à quel moment faut-il prélever les instantanés ou appliquer les mises à jour. Ce qui devrait pousser à étudier la fréquence optimale pour répertorier les changements. Hélas, cet aspect n'est pas encore bien étudié au moment où on écrit ce papier.

Le tableau 3 résume les avantages et les inconvénients des approches de détection de communautés dans les réseaux dynamiques.

6. Discussion sur les approches existantes

Dans cette section, nous présentons les limites des approches existantes de détection de communautés dynamiques. Nous avons catégorisé les limites en quatre défis. Pour chaque défi, nous commençons par citer les faiblesses actuelles pour ensuite proposer quelques pistes d'amélioration.

6.1. Défi 1 : taille des fenêtres temporelles

L'analyse de l'évolution des réseaux dynamiques nécessite le choix d'un intervalle de temps, appelé souvent « instantané » ou « fenêtre temporelle », déterminant le délai à partir duquel les mises à jour doivent être prises en compte. En réalité, le choix de la fenêtre de temps reste encore rarement détaillé dans les articles traitant les réseaux dynamiques. En général, les auteurs se limitent à préciser l'intervalle de temps qu'ils jugent adéquat sans en fournir aucune discussion ou argumentation. En effet, il existe deux méthodes de gestion de fenêtre temporelle :

Tableau 3. Critique des méthodes existantes de détection de communautés dynamiques

Catégorie	Avantages	Inconvénients
Approche par instantanés uniformes successifs	Avoir à disposition beaucoup d'outils et de techniques prêts à l'emploi. Possibilité de parallélisation du découpage du réseau pour améliorer davantage le temps d'exécution.	Instabilité. Volume croissant des instantanés si le réseau est de grande taille et les instantanés sont nombreux.
Approche par instantanés consécutifs, deux à deux interdépendants	Réduction de l'instabilité.	Lenteur en termes de temps d'exécution.
Approche par instantanés multi-pas	Réduire considérablement le niveau de stabilité en cherchant des communautés cohérentes sur plusieurs fenêtres de temps.	Complexité très élevée notamment si les instantanés sont nombreux. Aucun algorithme de cette approche ne permet de détecter efficacement les opérations de fusion et de division.
Approche incrémentale	Garantit, à court terme, la stabilité de communautés. La bonne complexité représente également un second avantage car si deux instantanés consécutifs sont topologiquement proches, la mise à jour de communautés s'effectue dans un délai très court.	Difficulté d'assurer une cohérence de communautés sur l'ensemble des étapes d'évolution, car le suivi de communautés se fait uniquement lors du passage d'un instantané à l'autre.

– **Méthode statique**, qui consiste à décomposer le réseau en plusieurs instantanés ayant tous la même taille.

– **Méthode dynamique**, qui subdivise le réseau en plusieurs instantanés où la taille de chacun peut éventuellement être différente de celle de l'autre.

Certainement, la méthode statique manque d'efficacité, à savoir, la période choisie n'est généralement pas pertinente car elle est basée uniquement sur l'observation empirique. Par conséquent, il est probable que les communautés détectées à l'instantané $T+1$ ne soient pas si différentes de celle de l'instantané T . Pour illustrer ce cas, prenons l'exemple suivant : soit un réseau de proximité géographique qui est composé de chercheurs scientifiques se rencontrant périodiquement. Supposons que nous ayons observé le réseau au cours d'une année, puis nous avons constaté que les chercheurs se sont rencontrés quatre fois : dans la première semaine de l'année, après un mois, après 3 mois et après 6 mois. Le choix d'une période fixe pour diviser le réseau implique l'obtention des instantanés contenant exactement la même composition de communautés. Ainsi, nous perdons du temps et des ressources en cherchant de nouvelles communautés alors que le réseau n'a pas changé. D'où l'importance des méthodes dynamiques qui donnent plus de garanties de prendre des captures que si et seulement si cela est nécessaire. La méthode dynamique, en revanche, adopte un

modèle dynamique permettant de choisir la taille optimale de la fenêtre temporelle en se basant sur la quantité des changements intervenus au fil du temps. Ainsi, la taille de la fenêtre peut toutefois varier d'un instantané à l'autre. Il convient de mentionner qu'à notre connaissance il n'existe aucun travail prenant en compte cette dimension pour décomposer convenablement l'évolution du réseau. Cela constitue un véritable défi à relever et auquel nos travaux futurs viseront à gérer.

6.2. Défi 2 : analyse multidimensionnelle

A la problématique du choix adéquat de fenêtre de temps, s'ajoutent les faiblesses inhérentes aux approches de détection de communautés dynamiques.

1. Dans l'approche par instantanés : les algorithmes statiques utilisés souffrent du problème d'instabilité. De plus, le cas particulier des réseaux égocentrés semble être ignoré dans les travaux existants surtout quand il s'agit de communautés égocentrées chevauchantes.

2. Dans l'approche incrémentale, les méthodes existantes ne gèrent pas efficacement le suivi des communautés, surtout en ce qui concerne les opérations de fusion et de division.

Par ailleurs, la quasi-totalité des méthodes actuelles traitent les communautés en se focalisant sur un nombre réduit de critères ou dimensions. Pourtant, la modélisation convenablement des réseaux issus des applications mobiles nécessite la combinaison de plusieurs dimensions pour détecter des communautés pertinentes. Par exemple, si nous voulons détecter les communautés dans un réseau de malfaiteurs (vendeurs de drogue), il sied de prendre en compte plusieurs dimensions à savoir :

- Dynamique (circulation du flux d'information) : nous devons nous intéresser plus au passage d'information au sein du réseau qu'à la structure topologique. Pour représenter l'intensité de communication, nous pouvons combiner la fréquence et la période de communication en une valeur sémantique considérée comme poids de liens.
- Temporelle : avoir une idée sur le transfert d'information ne suffit pas, il faut également sauvegarder l'historique de communication pour une utilisation éventuelle.
- Egocentrique : pour mieux repérer les éléments influents, il faudrait intégrer la représentation de communautés égocentrées, où chaque leader du réseau de malfaiteurs est entouré par les personnes auxquelles, il vend ou achète de la drogue.
- Spatiale : nous avons besoin de cet aspect pour comprendre la logique de propagation des toxicomanes au sein des groupes consommateurs. Tout comme, nous pouvons localiser les objets qui sont à l'origine d'une telle propagation.

Par ailleurs, les algorithmes de détection de communautés se focalisent généralement sur les propriétés structurelles du réseau, c'est-à-dire, les liens statiques reliant les nœuds. Or, en réalité, les liaisons entre les nœuds ne sont pas du même niveau, car les connexions entre les nœuds varient suivant la fréquence des échanges entre eux. D'où l'importance des algorithmes basés sur les flux de communication qui considèrent que plus deux nœuds communiquent davantage, plus la relation entre eux est

jugée forte. Pour modéliser les échanges entre les nœuds, les algorithmes font souvent recours aux poids des liens, où la valeur du poids augmente en fonction de l'élévation de communication entre les nœuds. De plus, s'il existe des éléments du réseau qui communiquent périodiquement (conférences internationales, journées mondiales, etc), il serait intéressant de trouver une solution qui ne se base pas totalement ni sur les liens statiques, ni sur l'abondance de communication.

6.3. Défi 3 : gestion du volume de données et de la confidentialité

En outre, la gestion de l'historique des réseaux dynamiques représente à nos jours un défi de taille. Comme, avec le temps, la sauvegarde et le traitement des états antérieurs du réseau deviennent difficiles à gérer. Compte tenu du volume croissant de données, il serait intéressant de développer une nouvelle approche capable d'optimiser la taille de données stockées. De même, il serait indispensable de développer une technique permettant à l'algorithme de détecter les communautés à partir des données optimisées. Notons qu'aucune proposition existante permet de gérer ces deux problématiques.

Parmi les questions non abordées dans les travaux de recherche, nous citons la détection de communautés dans les réseaux confidentiels, où l'algorithme doit être capable de détecter les communautés sans avoir besoin de lire le contenu des données cryptées. Tel est le principe du chiffrement homomorphique. Notons qu'à notre connaissance, il n'existe pas un algorithme qui permet de gérer de telles données.

6.4. Défi 4 : validation des algorithmes pour les communautés dynamiques

Enfin, la validation des algorithmes présente effectivement un défi. Dans le cas des communautés statiques, la technique habituelle est d'utiliser un réseau dont la structure est connue au préalable, puis comparer le résultat fourni par l'algorithme avec les données réelles dont nous disposons. Parmi les modèles les plus utilisés, nous citons les Misérables et le graphe de Karate. En revanche, la tâche devient plus complexe lorsqu'il s'agit des réseaux dynamiques. Il serait donc très intéressant de développer un outil permettant de générer des graphes dynamiques structurés en communautés en intégrant les opérations d'évolution de communautés, afin de mettre en exergue les forces et les faiblesses des approches de détection de communautés dynamiques.

Toutes ces problématiques révèlent de nombreuses questions fondamentales et ouvrent des pistes de recherche intéressantes.

7. Conclusion

Dans cet article, nous avons passé en revue des solutions de détection des communautés dans les réseaux sociaux. Dans cette perspective, nous avons abordé aussi bien les travaux sur les communautés statiques que ceux sur les communautés dynamiques. Nous avons élaboré une grille de classification composée d'un ensemble de

critères pour évaluer les revues existantes et relatives au domaine étudié. Ce processus a permis de jauger le niveau de couverture de la littérature par ces revues mais aussi d'identifier les limites de celles-ci.

Ce papier est, par conséquent, un complément aux revues existantes avec le but d'intégrer les solutions récentes mais aussi de regrouper les algorithmes par mode de fonctionnement. Cela donne l'opportunité au lecteur de voir dans un seul document les grandes tendances dans le domaine de la détection des communautés statiques et dynamiques.

Pour atteindre notre objectif, une classification des algorithmes de détection des communautés statiques a été faite et une synthèse des limites des solutions au regard de leur principe de fonctionnement a permis de souligner quelques questions non encore bien couvertes. Parmi celles-ci, l'aspect dynamique des réseaux et la prise en compte de leurs propriétés intrinsèques.

Ensuite, nous avons procédé à une classification des approches de détection de communautés dans les réseaux dynamiques pour identifier les faiblesses des algorithmes existants et les défis qui restent encore à relever. En effet, l'étude des principes, des avantages et des inconvénients des algorithmes ont permis de dégager quatre grands défis dont la prise en compte permettra de mieux faire face à des communautés dynamiques issues des applications sociales et mobiles dans lesquelles, les facteurs temps et espace occupent une bonne place.

Les résultats issus de cette étude nous ont permis de comprendre les défis qui restent encore à relever et d'identifier les pistes de recherche à explorer.

Bibliographie

- Asur S., Parthasarathy S., Ucar D. (2009). An event-based framework for characterizing the evolutionary behavior of interaction graphs. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, n° 4, p. 16.
- Aynaud T., Fleury E., Guillaume J.-L., Wang Q. (2013). Communities in evolving networks: definitions, detection, and analysis techniques. In *Dynamics on and of complex networks, volume 2*, p. 159–200. Springer.
- Aynaud T., Guillaume J.-L. (2010). Long range community detection. In *Latin-american workshop on dynamic networks*, p. 4–p.
- Blondel V. D., Guillaume J.-L., Lambiotte R., Lefebvre E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, vol. 2008, n° 10, p. P10008.
- Brandes U., Delling D., Gaertler M., Görke R., Hofer M., Nikoloski Z. *et al.* (2006). *On modularity- np -completeness and beyond*. Citeseer.
- Bródka P., Saganowski S., Kazienko P. (2013). Ged: the method for group evolution discovery in social networks. *Social Network Analysis and Mining*, vol. 3, n° 1, p. 1–14.

- Capocci A., Servedio V. D., Caldarelli G., Colaiori F. (2005). Detecting communities in large networks. *Physica A: Statistical Mechanics and its Applications*, vol. 352, n° 2, p. 669–676.
- Cazabet R. (2013). *Détection de communautés dynamiques dans des réseaux temporels*. Thèse de doctorat non publiée, Université Paul Sabatier-Toulouse III.
- Cazabet R., Amblard F. (2011). 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.
- Cazabet R., Amblard F., Hanachi C. (2010). Detection of overlapping communities in dynamical social networks. In *Social computing (socialcom), 2010 ieee second international conference on*, p. 309–314.
- Chan S.-Y., Hui P., Xu K. (2009). Community detection of time-varying mobile social networks. In *Complex sciences*, p. 1154–1159. Springer.
- Chen W., Liu Z., Sun X., Wang Y. (2010). A game-theoretic framework to identify overlapping communities in social networks. *Data Mining and Knowledge Discovery*, vol. 21, n° 2, p. 224–240.
- Coscia M., Giannotti F., Pedreschi D. (2011). A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining*, vol. 4, n° 5, p. 512–546.
- Csardi G., Nepusz T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, vol. 1695, n° 5, p. 1–9.
- Falkowski T., Barth A., Spiliopoulou M. (2008). Studying community dynamics with an incremental graph mining algorithm. *AMCIS 2008 Proceedings*, p. 29.
- Fortunato S. (2010). Community detection in graphs. *Physics reports*, vol. 486, n° 3, p. 75–174.
- Fortunato S., Latora V., Marchiori M. (2004). Method to find community structures based on information centrality. *Physical review E*, vol. 70, n° 5, p. 056104.
- Girvan M., Newman M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, vol. 99, n° 12, p. 7821–7826.
- Gori M., Pucci A., Roma V., Siena I. (2007). Itemrank: A random-walk based scoring algorithm for recommender engines. In *Ijcai*, vol. 7, p. 2766–2771.
- Greene D., Doyle D., Cunningham P. (2010). 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.
- Gregory S. (2010). Finding overlapping communities in networks by label propagation. *New Journal of Physics*, vol. 12, n° 10, p. 103018.
- Harenberg S., Bello G., Gjeltema L., Ranshous S., Harlalka J., Seay R. *et al.* (2014). Community detection in large-scale networks: a survey and empirical evaluation. *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 6, n° 6, p. 426–439.
- Haynes J., Perisic I. (2009). Mapping search relevance to social networks. In *Proceedings of the 3rd workshop on social network mining and analysis*, p. 2.

- Heymann S. (2013). *Exploratory link stream analysis for event detection*. Thèse de doctorat non publiée, Université Pierre et Marie Curie-Paris VI.
- Hopcroft J., Khan O., Kulis B., Selman B. (2004). Tracking evolving communities in large linked networks. *Proceedings of the National Academy of Sciences*, vol. 101, n° suppl 1, p. 5249–5253.
- Hui P., Sastry N. (2009). Real world routing using virtual world information. In *Computational science and engineering, 2009. cse'09. international conference on*, vol. 4, p. 1103–1108.
- Jdidia M. B., Robardet C., Fleury E. (2007). Communities detection and analysis of their dynamics in collaborative networks. In *Icdim*, p. 744–749.
- Jeffrey Travers S. M. (1969). An experimental study of the small world problem. *Sociometry*, vol. 32, n° 4, p. 425–443.
- Khatoon M., Banu W. A. (2015). A survey on community detection methods in social networks. *International Journal of Education and Management Engineering (IJEME)*, vol. 5, n° 1, p. 8.
- Lancichinetti A., Fortunato S., Kertész J. (2009). Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, vol. 11, n° 3, p. 033015.
- Leung I. X., Hui P., Lio P., Crowcroft J. (2009). Towards real-time community detection in large networks. *Physical Review E*, vol. 79, n° 6, p. 066107.
- Li J., Huang L., Bai T., Wang Z., Chen H. (2012). Cdbia: a dynamic community detection method based on incremental analysis. In *Systems and informatics (icsai), 2012 international conference on*, p. 2224–2228.
- Li J., Wang X., Cui Y. (2014). Uncovering the overlapping community structure of complex networks by maximal cliques. *Physica A: Statistical Mechanics and its Applications*, vol. 415, p. 398–406.
- Lin Y.-R., Chi Y., Zhu S., Sundaram H., Tseng B. L. (2009). Analyzing communities and their evolutions in dynamic social networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, n° 2, p. 8.
- Massen C. P., Doye J. P. (2005). Identifying communities within energy landscapes. *Physical Review E*, vol. 71, n° 4, p. 046101.
- Mitrović M., Tadić B. (2009). Spectral and dynamical properties in classes of sparse networks with mesoscopic inhomogeneities. *Physical Review E*, vol. 80, n° 2, p. 026123.
- Mucha P. J., Richardson T., Macon K., Porter M. A., Onnela J.-P. (2010). Community structure in time-dependent, multiscale, and multiplex networks. *science*, vol. 328, n° 5980, p. 876–878.
- Nagpal A., Jatain A., Gaur D. (2013). Review based on data clustering algorithms. In *Information & communication technologies (ict), 2013 ieee conference on*, p. 298–303.
- Newman M. E. (2004). Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 38, n° 2, p. 321–330.
- Palla G., Barabási A.-L., Vicsek T. (2007). Quantifying social group evolution. *Nature*, vol. 446, n° 7136, p. 664–667.

- Palla G., Derényi I., Farkas I., Vicsek T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, vol. 435, n° 7043, p. 814–818.
- Papadopoulos S., Kompatsiaris Y., Vakali A., Spyridonos P. (2012). Community detection in social media. *Data Mining and Knowledge Discovery*, vol. 24, n° 3, p. 515–554.
- Pons P., Latapy M. (2005). Computing communities in large networks using random walks. In *Computer and information sciences-iscis 2005*, p. 284–293. Springer.
- Porter M. A., Onnela J.-P., Mucha P. J. (2009). Communities in networks. *Notices of the AMS*, vol. 56, n° 9, p. 1082–1097.
- Raghavan U. N., Albert R., Kumara S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, vol. 76, n° 3, p. 036106.
- Reichardt J., Bornholdt S. (2004). Detecting fuzzy community structures in complex networks with a potts model. *Physical Review Letters*, vol. 93, n° 21, p. 218701.
- Roma G., Herrera P. (2010). Community structure in audio clip sharing. In *Intelligent networking and collaborative systems (incos), 2010 2nd international conference on*, p. 200–205.
- Rosvall M., Axelsson D., Bergstrom C. T. (2009). The map equation. *The European Physical Journal Special Topics*, vol. 178, n° 1, p. 13–23.
- Rosvall M., Bergstrom C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, vol. 105, n° 4, p. 1118–1123.
- Rosvall M., Bergstrom C. T. (2010). Mapping change in large networks. *PloS one*, vol. 5, n° 1, p. e8694.
- Schult D. A., Swart P. (2008). Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th python in science conferences (scipy 2008)*, vol. 2008, p. 11–16.
- Shang J., Liu L., Xie F., Chen Z., Miao J., Fang X. *et al.* (2014). A real-time detecting algorithm for tracking community structure of dynamic networks. *arXiv preprint arXiv:1407.2683*.
- Slanina F., Zhang Y.-C. (2005). Referee networks and their spectral properties. In *Acta physica polonica b*, p. 2797.
- Son S.-W., Jeong H., Noh J. D. (2006). Random field ising model and community structure in complex networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 50, n° 3, p. 431–437.
- Tantipathananandh C., Berger-Wolf T., Kempe D. (2007). 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.
- Tong H., Faloutsos C., Pan J.-Y. (2008). Random walk with restart: fast solutions and applications. *Knowledge and Information Systems*, vol. 14, n° 3, p. 327–346.
- Wang Y., Wu B., Du N. (2008). Community evolution of social network: feature, algorithm and model. *arXiv preprint arXiv:0804.4356*.

- Weiss R. S., Jacobson E. (1955). A method for the analysis of the structure of complex organizations. *American Sociological Review*, vol. 20, n° 6, p. 661–668.
- Weng L., Menczer F., Ahn Y.-Y. (2013). Virality prediction and community structure in social networks. *Scientific reports*, vol. 3.
- Whitbeck J., Amorim M. Dias de, Conan V., Guillaume J.-L. (2012). Temporal reachability graphs. In *Proceedings of the 18th annual international conference on mobile computing and networking*, p. 377–388.
- Wilkinson D. M., Huberman B. A. (2004). A method for finding communities of related genes. *proceedings of the national Academy of sciences*, vol. 101, n° suppl 1, p. 5241–5248.
- Xie J., Szymanski B. K. (2012). Towards linear time overlapping community detection in social networks. In *Advances in knowledge discovery and data mining*, p. 25–36. Springer.
- Xie J., Szymanski B. K., Liu X. (2011). Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *Data mining workshops (icdmw), 2011 ieee 11th international conference on*, p. 344–349.
- Xu K. S., Klinger M., Hero III A. O. (2011). Tracking communities in dynamic social networks. In *Social computing, behavioral-cultural modeling and prediction*, p. 219–226. Springer.
- Yang B., Liu D., Liu J. (2010). Discovering communities from social networks: methodologies and applications. In *Handbook of social network technologies and applications*, p. 331–346. Springer.
- Yang T., Chi Y., Zhu S., Gong Y., Jin R. (2011). Detecting communities and their evolutions in dynamic social networks—a bayesian approach. *Machine learning*, vol. 82, n° 2, p. 157–189.
- Yoon J., Blumer A., Lee K. (2006). An algorithm for modularity analysis of directed and weighted biological networks based on edge-betweenness centrality. *Bioinformatics*, vol. 22, n° 24, p. 3106–3108.
- Yulian N. (2011). *Community detection in relation to the spread of epidemics*. Mémoire de Master non publié, University of Oxford. Consulté sur http://people.maths.ox.ac.uk/porterm/research/yuli_Dissertation_final.pdf/