

UNIVERSITY NAME

DOCTORAL THESIS

Conversations, Groupes et Communautés dans les Flots de Liens

Auteur :
Noé GAUMONT

Directeurs :
Clémence MAGNIEN Matthieu
LATAPY

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy*

in the

Research Group Name
Department or School Name

22 juin 2016

Table des matières

Remerciements	ii
Introduction	1
1 État de l'art sur la détection de communautés et les réseaux dynamiques	3
1.1 Communauté dans les graphes	3
1.1.1 Parititons de nœuds	4
1.1.2 Couverture de nœuds	4
1.1.3 Comapraison	5
1.2 Extension temporelle des graphes	5
1.2.1 snapshot	5
1.2.2 TVG/Evolving	5
1.2.3 Flots de liens	5
2 Flots de liens : extensions temporelle des graphes	7
2.1 Définition	7
2.2 Sous-flot induit	7
2.3 Degré et densité	7
Liste des notations	9
3 Étude d'une archive de courriels	11
3.1 Prétraitemet sur le jeu de données	11
3.2 Caractéristiques élémentaires des discussions	12
3.3 Étude des discussions en tant que sous-flots	14
3.3.1 Application de la Δ -densité	14
3.3.2 Répartition temporelle et structurelle des discussions	17
3.3.3 Flot quotient	19
3.3.4 Conclusion	20
3.4 Détection de structures denses	21
3.4.1 Méthode de détection	21
3.4.2 Comparaison des partitions	22
3.4.3 Conclusion	24
4 Détection de groupes denses (SNAM)	27
4.1 Calcul des groupes candidats	27
4.2 Calcul évaluation	27
4.3 Jeux de données	27
4.4 Application	27
4.5 Conclusion	27

5 Expected Nodes : communautés de liens dans les graphes statiques	29
5.1 Travaux existants	30
5.2 Définition d'Expected Nodes	33
5.3 Comparaison	36
5.3.1 Cas du graphe complet	36
5.3.2 Graphe LFR	37
5.4 Calcul et optimisation	40
5.5 Conclusion	41
5.5.1 Perspective	42
Amélioration de l'optimisation d' <i>Expected Nodes</i>	42
Cas d'utilisation d' <i>Expected Nodes</i>	43
6 Fonction de qualité	45
6.1 Définition	45
6.2 Générateur de flots de liens avec structure communautaire	45
7 Conclusion	47

Introduction

Chapitre 1

État de l'art sur la détection de communautés et les réseaux dynamiques

Sommaire

1.1	Communauté dans les graphes	3
1.1.1	Parititons de nœuds	4
1.1.2	Couverture de nœuds	4
1.1.3	Comaprison	5
1.2	Extension temporelle des graphes	5
1.2.1	snapshot	5
1.2.2	TVG/Evolving	5
1.2.3	Flots de liens	5

Au cours de cette thèse, nous étudions deux axes de recherches lié aux graphes qui sont orthogonaux. D'une part, il s'agit de la détection de structures dans les graphes et plus particulièrement de communautés. Un communauté est un sous-ensemble de nœuds de manière à ce qu'ils soient fortement connectés. Il n'existe cependant aucune définition exact et la notion de communauté fortement connectée dépend du contexte et de la méthode. Malgré cette définition floue, une structure communautaire a été trouvée dans de nombreux graphes dans plusieurs domaines tel que les réseaux de [...] REF. Ces notions de communautés sont définies dans la section 1.1.

D'autre part, il a été observé que la modélisation d'un réseau sous la forme d'un graphe peut poser problème notamment quand le réseau change au cours du temps REF. Face à ce problème plusieurs extensions de la théorie des graphes ont été proposées et elles sont résumées dans la section 1.2.

1.1 Communauté dans les graphes

Ce champs de recherche est très vaste et il est illusoire de vouloir énumérer les méthodes existantes dans ce domaine car les caractéristiques voulues d'une communauté peuvent varier [4]. Il y a tout de même deux grandes catégories qui permettent de séparer les méthodes existantes. Dans ces deux catégories, les méthodes existantes cherchent à capturer des communautés fortement connectées mais elles diffèrent sur ce qu'elles capturent comme structures communautaires. Les communautés d'une structure communautaire peuvent être disjointes, c'est-à-dire que deux communautés n'ont aucun nœuds en commun, ou alors recouvrantes, c'est-à-dire que deux communautés peuvent avoir plusieurs nœuds en commun. Dans le premier cas, on parle de partition des nœuds et dans le second on parle de couverture ou partitions

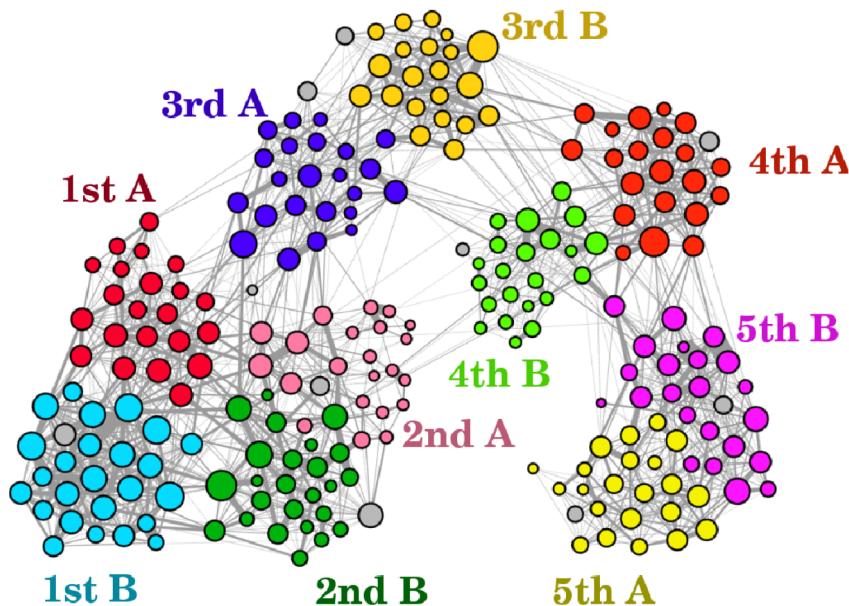


FIGURE 1.1 – Graphe de contact des enfants d'une école primaire. L'épaisseur du lien représente la durée de communication entre deux élèves. La couleur représente la classe de chaque élève. Les professeurs sont en gris.¹

chevauchantes de nœuds. Dans ces deux cas, il y a également une contrainte sur le fait que tout les nœuds doivent appartenir à au moins une communauté. C'est deux structures sont correspondant à deux visions possibles de l'organisation d'un graphe et du réseau sous-jacent. Nous présentons ces deux catégories dans les sous-sections suivantes.

1.1.1 Parititons de nœuds

Afin de mieux comprendre ce que peuvent capturer une partition de nœuds, il est plus facile de partir d'un exemple. Dans l'étude de Stehlé *et al.* [31], des enfants d'une école primaire ont eu pendant 2 jours des capteurs enregistrant lorsque deux enfants sont à une distance de moins de 1m50 l'un de l'autre. Ce dispositif permet de mesurer les interactions entre élèves et de construire le graphe des relations entre élève à l'école. Une illustration du graphe obtenu est visible dans la figure 1.1. La classe de chaque élève est également connue. Comme chaque élève appartient à une et une seule classe, les classes forment une partition des élèves. Cette partition est une bonne structure communautaire car on remarque que les élèves d'une même classe parlent beaucoup entre eux mais ils parlent peu entre élèves de classes différentes. Cela se remarque particulièrement bien pour la classe 3A. Il existe beaucoup de liens entre les élèves de la classe 3A et aucun entre eux et les élèves de la classe 5A par exemple.

Afin de capturer des partitions de nœuds, beaucoup de méthodes existent. Il y a d'ailleurs régulièrement des état de l'art qui sont publiés [11, 26, 23, 13]

limite modu : [12] extension [27, 7] modularité [24], infomap SBM [14], LPA, Surprise
lien entre SBM et modularité[25]

1.1.2 Couverture de nœuds

infomap, SBM, LPA, fonction de qualité locale (conductance, cohésion), fonction de proximité

¹. Image provenant de <http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0023176>.

[5, 16, 34]

1.1.3 Comaprison

ARI, NMI, F1-score, Omega index.

1.2 Extension temporelle des graphes

1.2.1 snapshot

snap + fenetre glissante

1.2.2 TVG/Evolving

1.2.3 Flots de liens

Le formalisme de flot de liens est défini plus en profondeur dans le chapitre 2.

Chapitre 2

Flots de liens : extensions temporelle des graphes

Sommaire

2.1 Définition	7
2.2 Sous-flot induit	7
2.3 Degré et densité	7

2.1 Définition

Un flot de liens est défini comme un triplet : $\mathcal{L} = (T, V, E)$, où $T = [\alpha, \omega]$ est un intervalle de temps, V un ensemble de n nœuds et $E \subseteq T \times T \times V \times V$ un ensemble de m liens. Les liens de E sont des quadruplets (b, e, u, v) , signifiant que la paire de nœuds (u, v) est connectée sur l'intervalle $[b, e] \subseteq [\alpha, \omega]$. Nous considérons un flot non orienté, i.e. $(b, e, u, v) = (b, e, v, u)$, et sans boucle, i.e. $u \neq v$.

Nous dénotons la durée du flot par $\bar{L} = \omega - \alpha$. $\beta_E = \min_{(b, e, u, v) \in E}(b)$ et $\psi_E = \max_{(b, e, u, v) \in E}(e)$ sont respectivement l'apparition du premier lien et la disparition du dernier lien.

Un flot de liens est simple si pour tout $(b, e, u, v) \in E$ et $(b', e', u, v) \in E$, $[b, e] \cap [b', e'] = \emptyset$. La simplification d'un flot de liens $\sigma = L$.

$V(E') = u_{(b, e, u, v) \text{ in } E'}$ sommets induits par un ensemble de liens.

$\xi(L, \Delta)$ ajout de Δ à chaque lien.

Que des flots avec durées ou bien delta densité?

2.2 Sous-flot induit

Sous flot induit par un ensemble de lien $E' : L(E') = ([\beta_{E'}, \psi_{E'}, V(E'), E'])$.

Sous flot induit par un ensemble de paire nœuds $S \in V^2 : L(S) : L(S) = ([\beta_{E'}, \psi_{E'}], V', E')$ avec $E' = \{(b, e, u, v) \in E, (u, v) \in S\}$. Par convention, on note $L(v) = L(\{v\}) \times V$.

Sous flot induit par un intervalle de temps $T' = [\alpha', \omega']$, $T' \subseteq T : L_{\alpha'.. \omega'} : L_{\alpha'.. \omega'} = ([\alpha', \omega'], V(E'), E')$ avec $E' = \{(b', e', u, v), \exists (b, e, u, v) \in E, b' = \max(b, \alpha'), e' = \min(e, \omega')\}$. Par convention, on note $L_{t..t} = L_t$

Il est aussi possible de combiner ces notions. Par exemple avec $V' \subset V$, $L_{\alpha'.. \omega'}(V'^2)$ est le sous flot correspondant au lien entre les nœuds de V' sur l'intervalle $[\alpha', \omega']$.

2.3 Degré et densité

Beaucoup de notions sont développées autour de cet objet. Degré temporelle d'un nœud u :

$$d_t(u) = |L_t(v)| = |\{(b, e, u, v) \in E, b \leq t \leq e\}| \quad (2.1)$$

Par extension pour un ensemble de sommets V' :

$$d_t(V') = |L_t(V'^2)| = |\{(b, e, u, v) \in E, u, v \in V', b \leq t \leq e\}| \quad (2.2)$$

Degré interne maintenant ?

$$d_t(E') = |\{(b, e, u, v) \in E', b \leq t \leq e\}| \quad (2.3)$$

Il est possible d'intégrer sur l'ensemble du temps

$$D_{\alpha..w}(v) = \int_{\alpha}^{\omega} d(v, t) dt = D(v) \quad (2.4)$$

Par convention, on notera le degré moyen de v : $d(v) = \langle d(v, t) \rangle = \frac{D(v)}{\omega - \alpha}$. Il en va de même pour les autres degrés.

C'est le cas de la densité :

$$\delta(L) = \frac{2 \sum_{l \in E}}{n(n-1)(\bar{L})} \quad (2.5)$$

Liste des notations

Symbol	description
L	Flot de liens
T	intervalle de temps
V	ensemble de nœuds
E	ensemble de liens : (b, e, u, v)
n	nombre de nœuds
$ L , E $	nombre de liens dans le flot
β_E	temps d'apparition du premier lien
ψ_E	temps de disparition du dernier lien
$\xi(L, \Delta)$	Flot de liens où chaque lien dure Δ
$L(V'^2)$	sous flot induits par les nœuds de V'
$L_{t..t'}$	sous flot induits par l'intervalle $[t, t']$
$d_t(v)$	degré de v à l'instant t
$d(v)$	degré moyen v sur T
$\delta(L)$	densité du flot
$\delta_\Delta(L)$	densité du flot ou chaque lien dure Δ

Chapitre 3

Étude d'une archive de courriels

Sommaire

3.1	Prétraitement sur le jeu de données	11
3.2	Caractéristiques élémentaires des discussions	12
3.3	Étude des discussions en tant que sous-flots	14
3.3.1	Application de la Δ -densité	14
3.3.2	Répartition temporelle et structurelle des discussions	17
3.3.3	Flot quotient	19
3.3.4	Conclusion	20
3.4	Détection de structures denses	21
3.4.1	Méthode de détection	21
3.4.2	Comparaison des partitions	22
3.4.3	Conclusion	24

Nous nous intéressons ici à une archive de courriels publiquement disponibles¹. Cette archive contient l'ensemble des courriels échangés par différent utilisateurs pour résoudre un problème survenu lors de l'utilisation de Debian. Typiquement, une personne ayant un problème lors de l'installation envoie un courriel à la liste afin de demander de l'aide. Toute personne inscrite sur la liste reçoit ce courriel et peut y répondre ce qui donne lieu à une discussion visible par tous. Ces discussions ont déjà été étudiées dans le passé [8, 30, 32] mais cela a été fait en utilisant des méthodes statiques uniquement.

Or, ces données se représentent naturellement sous forme de flot de liens en associant chaque personne à un nœud et chaque courriel entre deux personnes à un lien dans le flot de liens à l'instant où le courriel a été envoyé. L'avantage de ces données de communications est que nous connaissons la discussion (*thread*) dans laquelle a lieu chaque message. Une discussion est un ensemble de courriels dont tout les messages répondent à un message précédent de la discussion excepté pour le premier qui a initié la discussion et que nous appelons *racine*. Ainsi, nous étudions la structure des discussions dans le flot liens représentant les courriels envoyés sur la liste.

Utiliser le formalisme de flot de liens est particulièrement intéressant car cette lite de diffusion existe depuis 1994. L'aspect temporel des discussions est donc important.

3.1 Prétraitement sur le jeu de données

Bien qu'accessible sur internet, ce jeu de données nécessite un ensemble de traitements avant de pouvoir exploiter les 724985 courriels que contenait l'archive en janvier 2015. Tout d'abords, les données ne sont pas sous la forme d'un flot de liens avec la structure des conversations. Les données sont accessibles via le site internet et ne sont pas structurées. Pour avoir ces informations sous la forme d'un flots de liens, un script d'extraction a été développé [URL](#).

1. <https://lists.debian.org/debian-user/>

Lors de l'extraction, 2269 courriels n'ont pas pu être pris en compte car certaines informations étaient manquantes ou mal formées, typiquement à cause de la date ou d'un fuseau horaire non reconnu.

Une fois les informations brutes récupérées, il faut les transformer en un flot de liens cohérent. Pour chaque message m , nous extrayons son auteur $a(m)$, l'instant $t(m)$ auquel le message a été posté², le message auquel il répond $p(m)$ trouvé via le champ IN-REPLY-TO, son destinataire $a(p(m))$ et la discussion $D(m)$ dans laquelle il apparaît. Comme les messages *racines* ne répondent à aucun autre, nous imposons $p(m) = m$. L'ensemble de liens du flot est donc $\{(t(m), a(m), a(p(m)))\}_m$. Nous ne prenons pas en compte la direction des liens.

Une fois le flot créé, il est encore nécessaire de vérifier sa cohérence. Un message peut être filtré pour différentes raisons : le courriel apparaît avant le message auquel il est censé répondre, le message auquel il répond n'est pas présent dans l'archive, l'auteur et le destinataire sont la même personne. Cette dernière condition permet notamment d'éviter la présence de boucles dans le flot. Cela concerne principalement les *racines*. Il s'agit de vérifications simples auquel il faut ajouter les vérifications sur la cohérence de la structure des discussions. Ainsi, une discussion est entièrement retirée du jeu de données s'il manque la *racine*, si un de ses messages a été retiré à l'étape précédente. Après ces vérifications, environ 7% des discussions sont retirées. À cela, il faut également tenir compte de notre temps d'observation qui est partiel. En effet, une discussion dont le dernier message a lieu 1 semaine avant la fin de la capture peut ne pas être terminée. De même, une discussion durant très longtemps n'a qu'une faible probabilité d'être capturée en entier. Pour corriger ces biais, nous filtrons également les discussions ayant débutées trop récemment ou durant trop longtemps. La limite pour considérer une discussion trop récente ou trop longue a été fixé à 4 ans (1.26×10^8 s) car nous avons constaté qu'uniquement quelques discussions dépassées ce seuil sur la distribution de durées des discussions dans la figure 3.1. Toutes les discussions qui ont débutées moins de 4 ans avant la capture du jeu de données ne sont donc pas prises en compte.

Une fois tout ces messages filtrés, nous obtenons un flot de liens avec 316 569 liens entre 34 648 personnes pendant presque 19 ans et 116 999 discussions. Mis à part les 237 664 messages de début de discussion, ce sont 168 482 courriels qui ont été filtrés soit environ 23%. La majorité des courriels filtrés l'ont été car ils appartiennent à une discussions trop récente.

3.2 Caractéristiques élémentaires des discussions

Les caractéristiques les plus élémentaires des discussions sont le nombre de courriels, le nombres de personnes, le nombre de paires de personnes distinctes en interaction directes et leur durée. Dans la figure 3.1, sont présentées les distributions cumulatives inverses de ces quantités et on remarque qu'elles sont toutes hétérogènes. On remarque que les données filtrées ne diffèrent pas qualitativement des données brutes.

La distribution des durées des discussions montre que la majorité des discussions dure environ une journée ou moins (10^5 secondes équivaut à moins de 28 heures). Par ailleurs, on remarque qu'il n'existe que quelques discussions durant plus d'un an.

Ces premières observations sont nécessaires mais pas suffisantes pour comprendre les caractéristiques d'une discussion. Nous avons également étudié la corrélation entre ces différentes notions et une partie d'entre elles sont présentées dans la figure 3.2.

La corrélation entre la durée et le nombre de courriels, dans la figure 3.2 partie gauche, met en évidence que plus une discussion est grande en nombre de courriels plus elle dure longtemps, ce qui est attendu. Par contre, on observe que les petites discussions ont des durées très variables. Dans la partie droite de la figure 3.2 présentant la corrélation entre le nombre de courriels et d'auteurs, on observe un autre fait attendu [8] qui est qu'une discussion est

2. Cet instant est convertit en *timestamp* en tenant compte des fuseaux horaires.

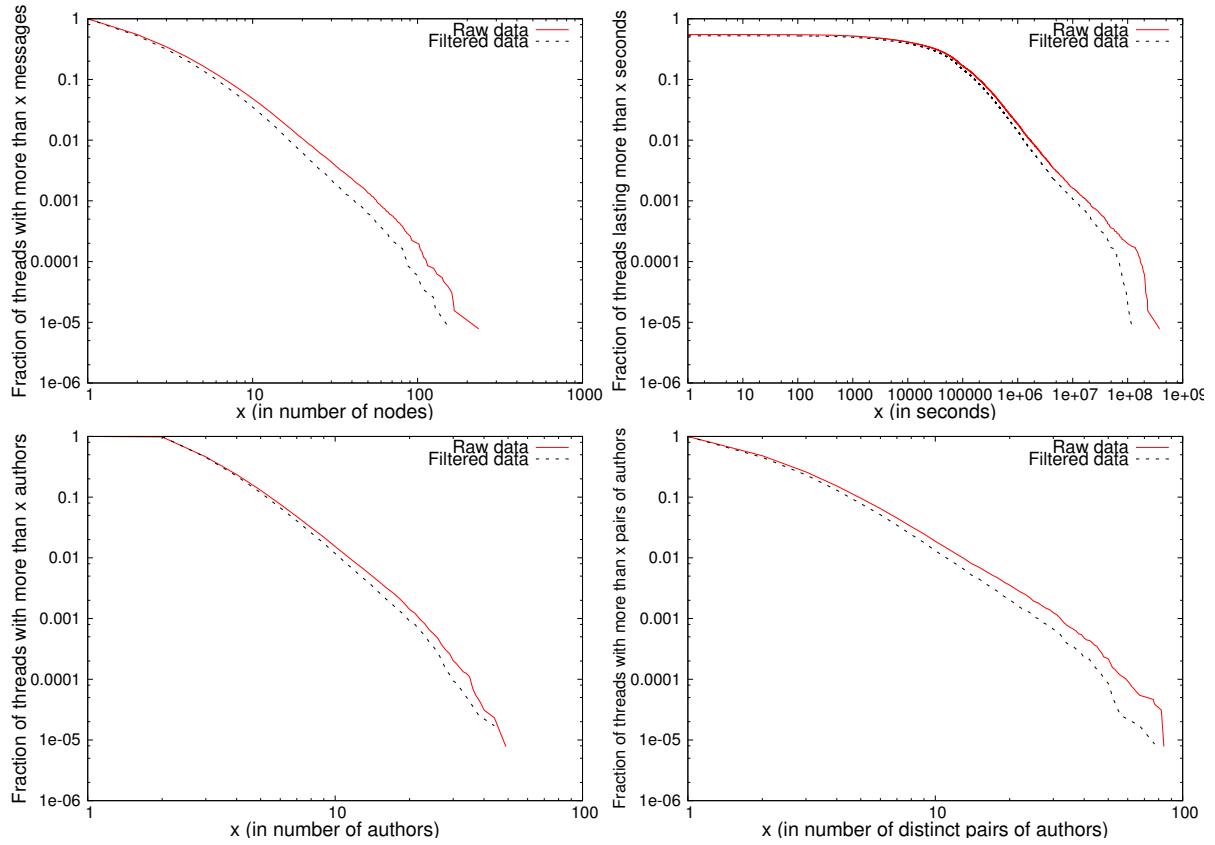


FIGURE 3.1 – Distribution cumulative inverse de différentes caractéristiques pour les données brutes (ligne pleine) et filtrées (ligne pointillé). En haut à gauche : nombre de courriels dans une discussion ; en haut à droite : durée d'une discussion ; en bas à gauche : nombre de personnes dans une discussion ; en bas à droite : nombre de paires d'auteurs distinct dans une .

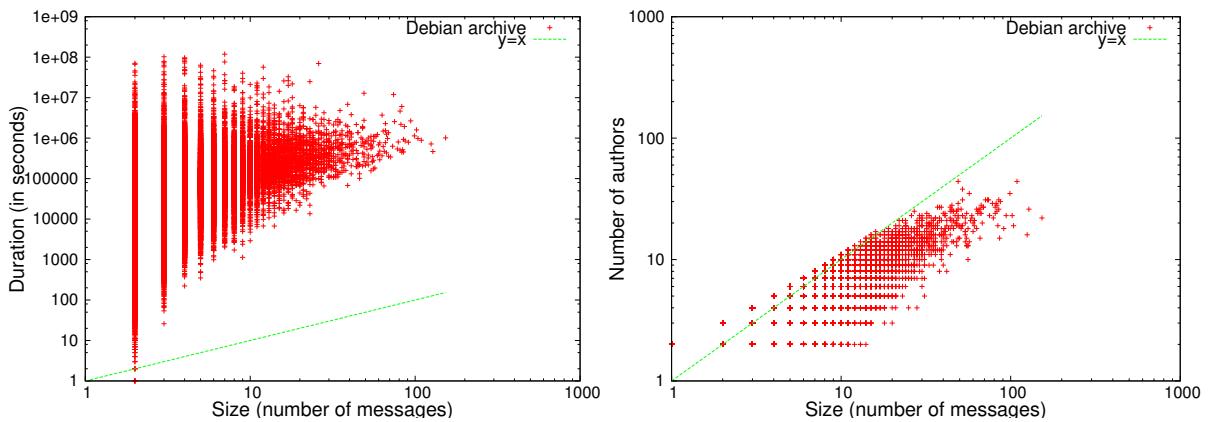


FIGURE 3.2 – Gauche : Corrélation entre le nombre de courriels et la durée d'une discussion. Droite : Corrélation entre le nombre de courriels et le nombre d'auteurs dans une discussion.

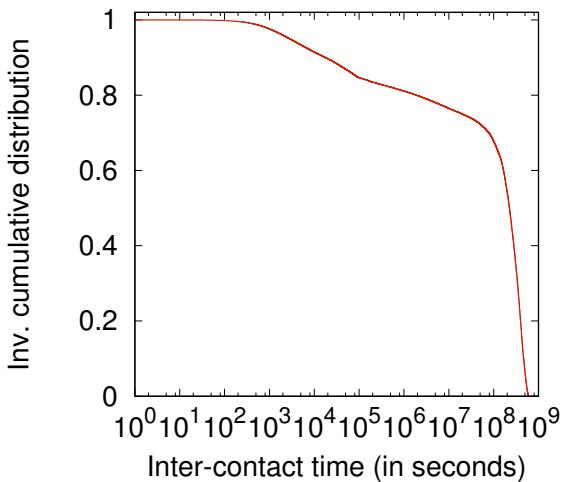


FIGURE 3.3 – Distribution des temps inter-contacts dans le fil de discussions.

constituée, en général, de plus de messages que de participants. Ainsi lors d'une discussion, c'est un petit nombre de personnes qui échangent potentiellement beaucoup de messages.

Enfin, il est intéressant d'observer la dynamique des échanges entre deux personnes. Soit $\tau(u, v) = (t_{i+1} - t_i)_{i=0..k+1}$ la séquence des temps inter-contacts des k liens entre les noeuds u et v , où t_0 est le temps entre α et le premier lien et t_{k+1} est le temps entre le dernier lien et Ω . Il s'agit du temps écoulé avant que deux personnes se contactent à nouveau, indépendamment peu importe la conversation. Dans la figure 3.3 est représentée la distribution cumulative inverse du temps inter-contacts. 21% des temps inter-contacts sont inférieurs à 30 jours (2.6×10^6 s). Ce chiffre bien que relativement faible est tout de même important car il s'agit de discussions ouvertes où tout le monde peut participer. En particulier, une personne peut envoyer une demande d'aide à un moment donné et ne plus jamais échanger avec les mêmes personnes. Or, on observe que 21% des contacts sont renouvelés en moins de 30 jours. La participation est donc relativement élevée.

3.3 Étude des discussions en tant que sous-flots

3.3.1 Application de la Δ -densité

Jusqu'à maintenant aucune notion intrinsèquement liée aux flots de liens n'a été utilisée pour caractériser les discussions. Le but est d'évaluer si cette structure de flot peut se rapprocher d'une structure communautaire. Comme dit précédemment, les communautés sont souvent définies comme étant des structures devant être densément connectées. C'est pourquoi nous nous attachons à étudier la densité des discussions.

Comme ces données se modélisent par un flot de liens où les liens n'ont pas de durée, nous étudions la Δ -densité pour différentes valeurs de Δ entre 1 seconde et 20 ans. Tout d'abord dans la figure 3.4 est représentée la Δ -densité globale du le flot. En couvrant un spectre aussi large de Δ , on observe que la Δ -densité est croissante avec Δ mais surtout on observe bien la convergence de Δ -densité vers 3.139×10^{-4} , la densité du graphe agrégé, lorsque Δ est proche de $\omega - \alpha$.

Cependant, la Δ -densité du flot n'apporte que peu d'informations en elle-même. Elle est surtout utile pour comparer les valeurs de Δ -densité des sous-flots que sont les discussions. Ainsi dans la figure 3.5, est présentée la distribution cumulative inverse de la Δ -densité des discussions pour différentes valeurs de Δ . On remarque que les différentes valeurs de Δ ne

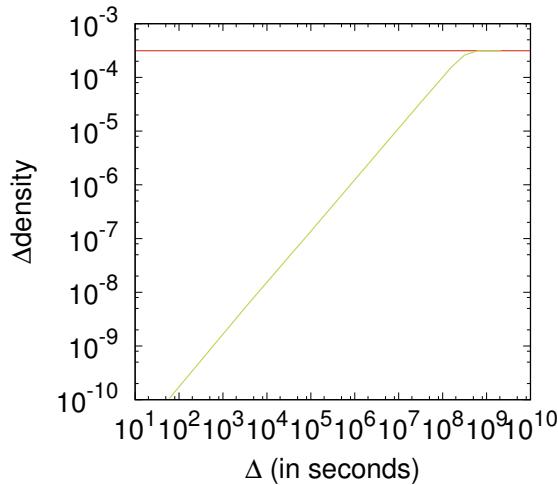


FIGURE 3.4 – Évolution de la Δ -densité (en vert) du flot de liens pour Δ de 60 seconde à 20 ans. En rouge, la densité dans le graphe agrégé.

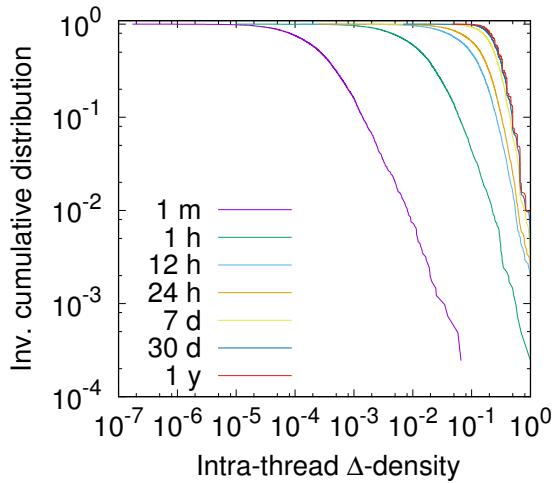


FIGURE 3.5 – Distribution cumulative inverse de la Δ -densité des discussions pour différentes valeurs de Δ s.

semblent pas influencer qualitativement la distribution de Δ -densité. Cette courbe met surtout en évidence que les discussions sont des structures beaucoup plus denses que le flot. En effet, la densité médiane des discussions est, selon la valeur de Δ , entre 2.69×10^{-4} et 0.28 alors que le flot a une Δ -densité variant entre 1.05×10^{-10} et 3.42×10^{-5} . La Δ -densité des discussions est donc en moyenne 10^5 fois plus élevée que celle du flot. Bien que notable, ce fait est attendu notamment car le flot dure beaucoup plus longtemps et concerne beaucoup plus de nœuds que les discussions.

Afin d'aller plus loin dans l'étude de cette structure, il faut revenir à une définition plus précise de ce qu'est une bonne communauté. En soi, une valeur de densité n'est pas suffisante pour définir une structure communautaire. En effet, une discussion ayant une densité de 0.8 peut ne pas être une communauté tandis qu'une autre ayant une densité proche de zéro peut être une communauté. Il faut définir un point de comparaison pour effectivement affirmer qu'une structure est particulièrement dense. La prise en compte de la densité globale est un début mais n'est pas suffisante.

Une autre définition d'une communauté est qu'elle devrait être plus densément connectée à l'intérieur qu'avec les autres communautés adjacentes. Pour un graphe $G = (V, E)$ et une

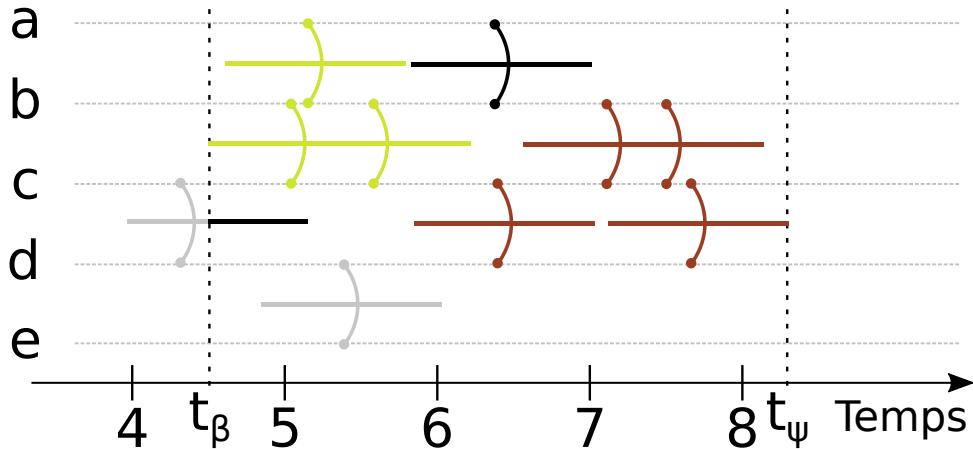


FIGURE 3.6 – Le flot entre les discussions verte et rouge est constitué des liens ou partie de liens en noir. Les liens en gris ne sont pas pris en compte.

communauté C_i de la partition $C = \{C_j\}_{j_1..k}$ de V en k communautés, cela se traduit par le calcul de la densité entre les communautés, $\delta^{inter}(C_i)$:

$$\delta^{inter}(C_i) = \frac{1}{|C|-1} \sum_{j,i \neq j} \frac{|\{(u,v) \in E \text{ t.q. } u \in C_i \text{ and } v \in C_j\}|}{|C_i| \cdot |C_j|}. \quad (3.1)$$

Il s'agit tout simplement de la probabilité qu'un lien existe entre un nœud de C_i et un nœud d'une autre communauté. Encore une fois, cette notion n'a pas de sens direct dans le formalisme de flot de liens et il est nécessaire de l'adapter. Pour ce faire, nous définissons la Δ -densité inter discussions entre deux discussions D_i et D_j : $\delta_\Delta^{inter}(D_i, D_j)$. Soit $E_\Delta = \xi(E, \Delta)$ et $L_{inter}(D_i, D_j) = (T', V', E')$ avec $V' = V(D_i \cup D_j)$, $T' = [t_\beta(D_i \cup D_j), t_\psi(D_i \cup D_j)]$, et $E' = E_\Delta \setminus (D_i \cup D_j)$. Avec ces définitions, $\delta_\Delta^{inter}(D_i, D_j)$ est égale à $\delta_\Delta^{inter}(D_i, D_j) = \delta(L_{inter}(D_i, D_j))$. Il s'agit donc de la densité du flot inter discussions qui est constitué des liens entre les nœuds induits par D_i et D_j qui n'appartiennent ni à D_i ni à D_j . Dans la figure 3.6, un exemple de flot inter discussion est représenté.

Afin d'obtenir la Δ -densité inter discussions entre D_i et tout les autres discussions, nous utilisons la moyenne des densité inter discussion entre D_i et les autres discussions, soit :

$$\delta_\Delta^{inter}(D_i) = \frac{1}{|C|-1} \sum_{j,i \neq j} \delta_\Delta^{inter}(D_i, D_j). \quad (3.2)$$

La distribution cumulative inverse de la Δ -densité inter discussions est présentée dans la figure 3.7 pour différentes valeurs de Δ . Bien que similaire, le comportement de la Δ -densité inter discussions diffère qualitativement de celui de la Δ -densité. La Δ -densité inter discussions croît également en fonction de Δ mais il y a toujours une différence notable entre $\Delta = 1 \text{ mois}$ et $\Delta = 1 \text{ an}$ ce qui n'est pas le cas pour la Δ -densité. Cette différence est normale car lors du calcul de Δ -densité le nombre de liens considérés est fixe peut importe Δ alors qu'il croît avec Δ lors du calcul de Δ -densité inter discussions. Cet effet est visible dans la figure 3.6. Le lien (c, d) qui apparaît peut avant t_β n'est pas pris en compte si Δ est proche de 0 alors qu'il est en parti pris en compte lorsqu'un Δ plus grand est considéré, ce qui est le cas dans la figure.

Un autre facteur est aussi la duré considérée qui est plus longue que la durée des discussions.

Afin de comparer plus aisément Δ -densité et Δ -densité inter discussions, la corrélation entre ces deux mesures est présentée dans la figure 3.8 pour différentes valeurs de Δ . On remarque que les discussions sont effectivement plus denses intérieurement qu'avec les autres

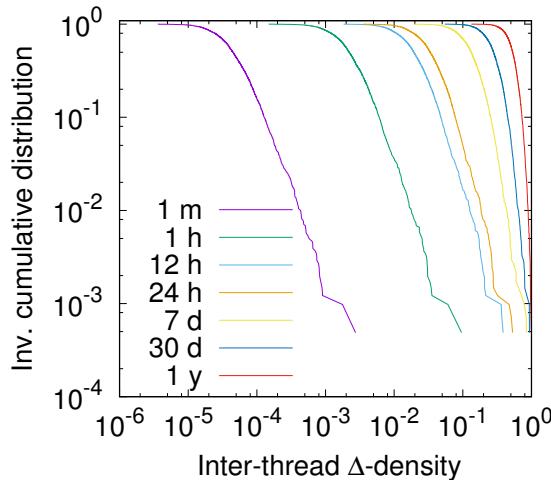


FIGURE 3.7 – Distribution cumulative inverse de la Δ -densité inter discussions pour différentes valeurs de Δ s.

discussions. La différence est de plusieurs ordres de grandeur lorsque Δ est petit et elle diminue lorsque Δ croît. Pour $\Delta = 20 \text{ ans}$ dans la figure ??, la différence n'est plus visible car à cette échelle de temps, l'ancrage temporel des discussions n'est plus décisif. On remarque tout de même que pour $\Delta = 1 \text{ an}$, la différence reste notable.

3.3.2 Répartition temporelle et structurelle des discussions

Nous avons étudié la densité des discussions et entre les discussions mais il est également intéressant d'observer comment ces discussions sont réparties topologiquement et temporellement. Pour étudier la répartition des discussions dans le temps, nous construisons un graphe d'intervalle [20] $X = (V_X, E_X)$ représentant le chevauchement temporel. Chaque discussion du flot devient un nœud de V_X et le lien (i, j) existe dans E_X si les discussions D_i et D_j correspondantes ont eu lieu au même instant, *i.e.* $[\alpha_i, \omega_i] \cap [\alpha_j, \omega_j] \neq \emptyset$. De manière similaire, nous définissons le graphe de chevauchement topologique $Y = (V_Y, E_Y)$. Les nœuds de ce graphe représentent encore une fois les discussions du flot et un lien existe entre deux discussions si au moins une personne a participé aux deux, *i.e.* $V(D_i) \cap V(D_j) \neq \emptyset$.

Ces deux graphes sont constitués de 116 999 nœuds et d'environ 2 millions de liens pour le graphe de chevauchement temporel et d'environ 63 millions de liens pour le graphe de chevauchement topologique. Par construction, ces graphes contiennent beaucoup d'informations sur les relations entre les discussions.

Dans la figure 3.9(gauche), est représentée la corrélation entre le degré d'une discussion dans le graphe de chevauchement temporel X et sa durée. Il y a une corrélation évidente entre ces deux notions lorsque les discussions ont une durée supérieur à 10^5 secondes. Plus une discussion dure longtemps, plus elle a de chance d'avoir lieu en même temps que beaucoup d'autres discussions. On observe également que, même pour les discussions durant moins d'un jour (8.6×10^4 s), il peut y avoir jusqu'à une centaine d'autres discussions actives sur la même période.

La figure 3.9(droite) présente la corrélation entre le degré d'une discussion dans le graphe de chevauchement topologique Y et son nombre de participants. La corrélation est moins nette mais il y a tout de même une tendance. Par contre, on remarque de manière frappante que même une petite discussions peut partager des nœuds avec les énormément d'autres discussions.

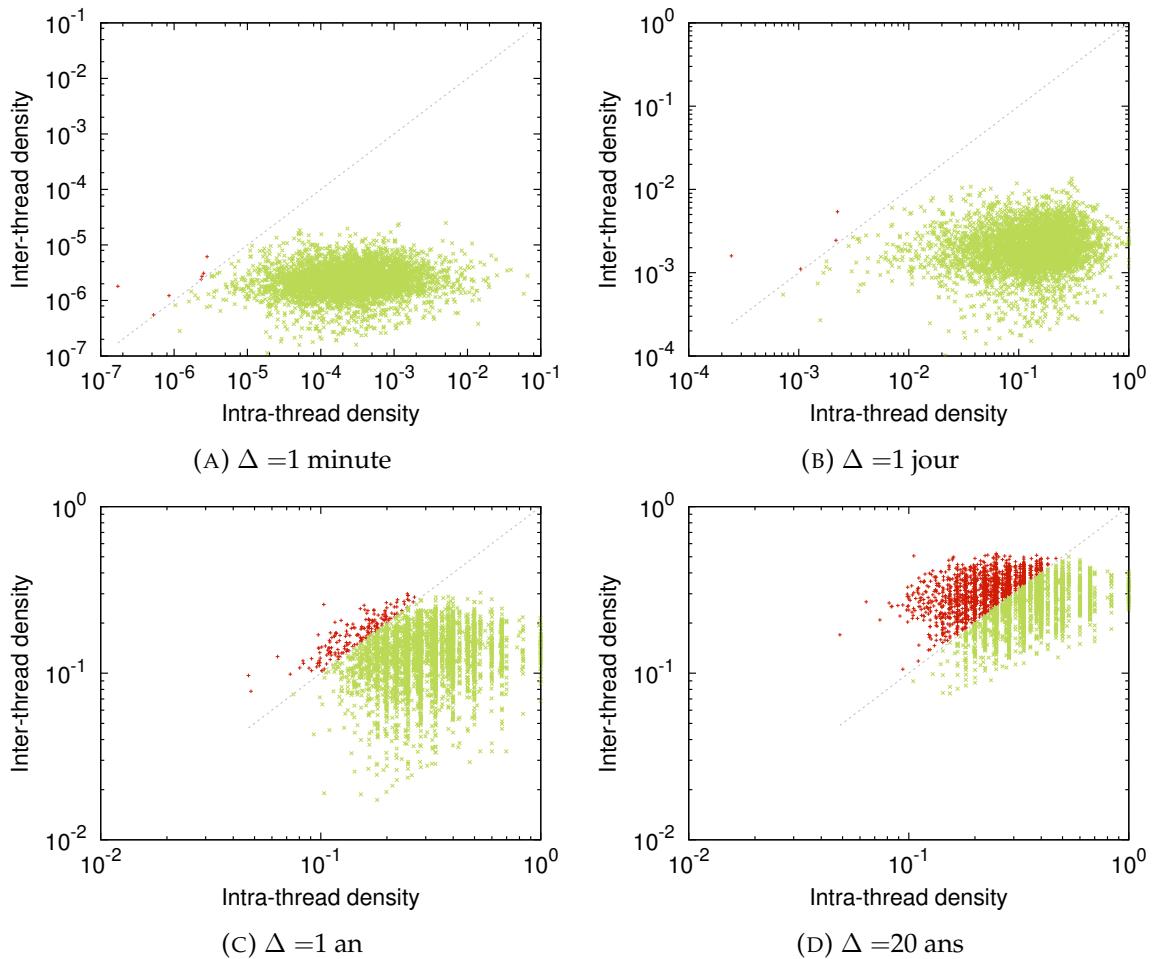


FIGURE 3.8 – Corrélations entre Δ -densité et Δ -densité inter discussions pour différentes valeurs de Δ . Une discussion est en vert (resp. rouge) si elle a une Δ -densité plus (resp. moins) élevée que sa Δ -densité inter discussions.

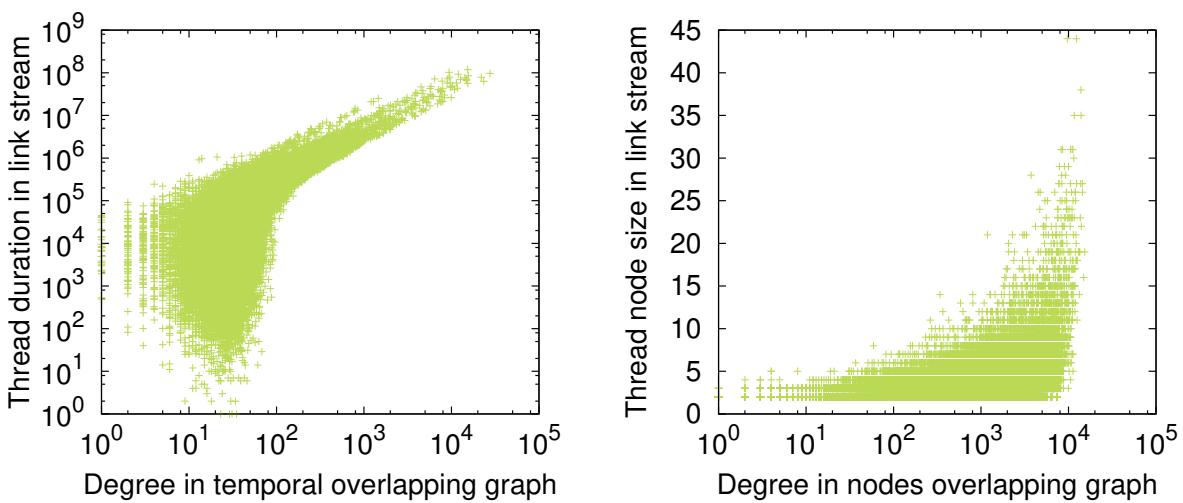


FIGURE 3.9 – Gauche : Corrélation entre le degré des discussions dans le graphe de chevauchement temporel et leur durée. Droite : Corrélation entre le degré des discussions dans le graphe de chevauchement topologique et leur nombre de participants.

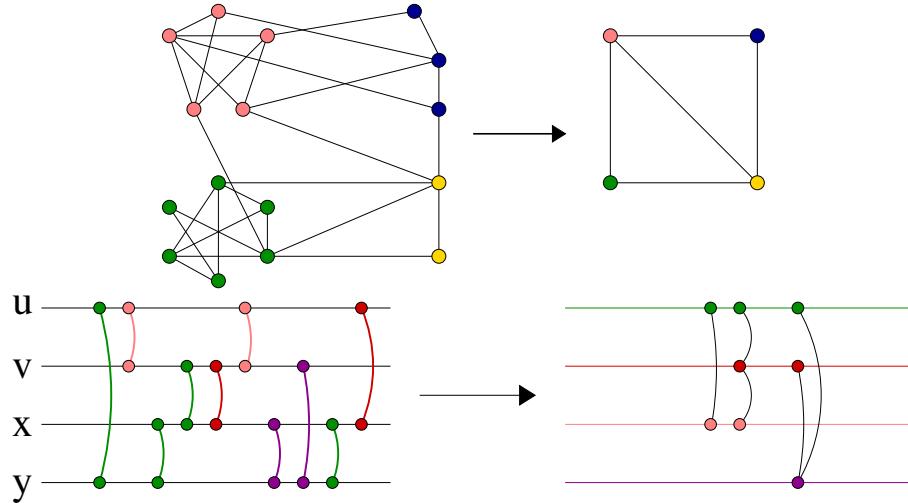


FIGURE 3.10 – Haut : Exemple de graphe ayant une structure communautaire et son graphe quotient associé. Bas : Exemple d'un flot de lien avec une structure ainsi que son flot quotient associé.

3.3.3 Flot quotient

Le graphe quotient est une autre notion clef pour étudier les relations entre les communautés d'un graphe $G = (V, E)$. Soit une partition $C = \{C_i\}_{1..k}$ des nœuds de G en k communautés, chaque communauté est représentée dans le graphe quotient \bar{G} par un nœuds dans V . Il y a un lien entre deux communauté C_i et C_j dans E si il existe au moins un lien entre un nœuds de C_i et un nœuds de C_j . Voir une illustration sur la figure 3.10. Il est possible d'ajouter un poids sur les liens de \bar{G} égale au nombre de liens reliant les communautés. Le graphe quotient permet de facilement étudier, dans un graphe, les relations entre les communautés.

Nous étendons ici cette notion de graphe quotient aux flots de liens. Nous définissons le flot quotient, $Q = (T_Q, V_Q, E_Q)$, induit par une partition $P = \{P_i\}_{1..k}$ en k sous-flots de la manière suivante. Chaque sous-flot P_i est représenté par un nœud dans V_Q . Il existe un lien (t, P_i, P_j) dans E_Q si il existe $(t_1, u, v) \in P_i$, $(t_2, u, v') \in P_j$ et $(t_3, u, v'') \in P_i$ avec $t_1 \leq t_2 \leq t_3$. En d'autre termes, il y a un lien dans E_Q si un nœud u a un lien dans P_j qui apparaît entre deux autres de ses liens du groupe P_i .

Le flot quotient induit par les discussions dans le jeu de données contient 12 281 269 liens impliquant 68 524 discussions différentes. Comme le jeu de données contient 116 999 discussions, il y a donc 48 475 discussions sans lien et qui ne seront pas prises en compte par la suite. Ce nombre de discussions non-reliées est élevé comparé à ce qui est obtenu dans un graphe. En effet dans un graphe, un nœud de degré 0 correspond à une communauté qui est une composante connexe (ou un union de composantes connexes). En ajoutant l'information temporelle, les discussions sont séparées par le temps dans flot. C'est pourquoi un grand nombre de discussions n'ont pas de liens dans le flot quotient. Ce phénomène est d'autant plus vrai pour les petites discussions.

Il faut aussi noter qu'il y a environ 20 fois plus de liens dans le flot quotient que dans le flot initial. Cela est normal car un lien dans le flot peut donner lieu à plusieurs liens dans le flot quotient. Ce cas est visible dans la figure 3.10. Le lien (x, y) du groupe violet du flot à gauche donne lieu au lien (*violet, rouge*) et au lien (*violet, vert*) dans le flot quotient à droite.

La figure 3.11 présente la Δ -densité du flot de liens initial et du flot quotient pour différentes valeurs de Δ . Le flot initial et le flot quotient ont le même comportant de densité mais le flot quotient est moins Δ -dense que le flot initial. Ce résultat diffère par rapport à ce qui est obtenu dans un graphe. Cela est dû au nombre de nœuds qui augmente dans le flot quotient. Mais le

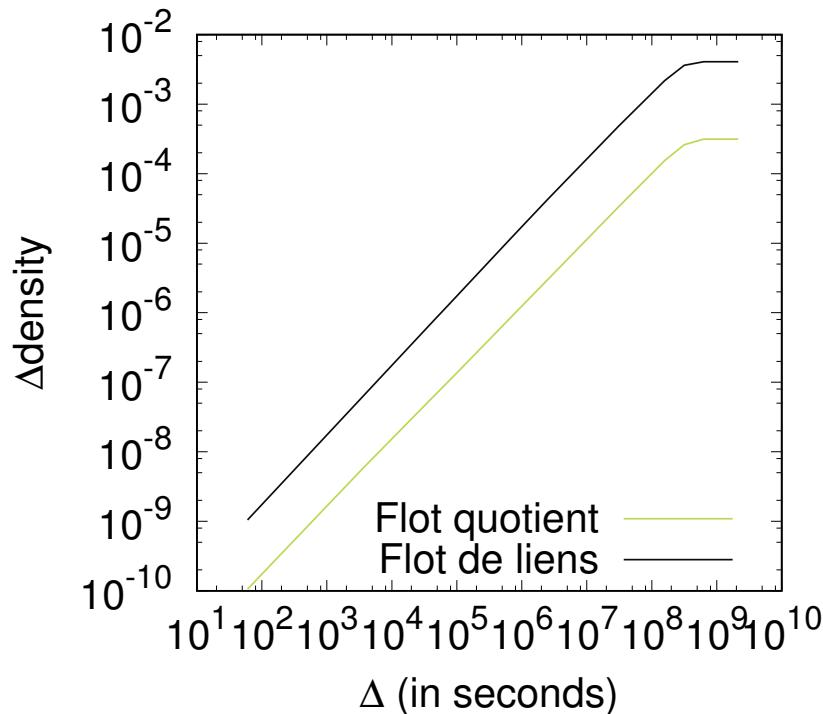


FIGURE 3.11 – Δ -densité du flot de liens et du flot de liens quotient en fonction de Δ pour $\Delta = 1mn, 1h, 12h, 1j, 7j, 30j, 1\text{ an}$ et 20 ans .

flot quotient contient tout de même beaucoup de liens. En effet, le degré moyen dans le flot quotient est moyenne 25 fois plus élevé que dans le flot.

3.3.4 Conclusion

Nous avons utilisé le modèle de flot de liens pour étudier une archive de courriels provenant du projet Debian. Grâce au modèle de flot de liens, nous avons étudié des notions clefs pour mieux comprendre la répartition temporelle et topologique des discussions. Nous avons étudié la notion de Δ -densité sur les discussions en elles mêmes. Puis, nous avons étudié les relations entre les discussions avec la Δ -densité inter discussions, les projections en graphe de chevauchement temporel ou topologique et le flot quotient.

Cette étude repose en grande partie sur la notion de Δ -densité qui nécessite un paramètre fixé arbitrairement. Nous avons à chaque fois testé un ensemble de valeurs de Δ variant d'une seconde jusque parfois 20 ans et, lors de ces tests, aucune valeur Δ caractéristique n'a pu être identifiée. Il semble donc que la Δ -densité soit relativement robuste vis-à-vis de Δ dans ce contexte.

Nous avons tout d'abord observé que les discussions forment une structure plus dense que le flot de liens. De manière encore plus forte, nous avons constaté, grâce à la Δ -densité inter discussion, que les discussions sont plus denses en interne qu'en externe. C'est une caractéristique importante des communautés que l'on trouve dans les graphes mais qui n'avait pas été observée dans un contexte temporel. À partir de ces observations, nous avons également observé les relations entre les discussions. Via le graphe de chevauchement temporel, nous avons validé le fait que différentes discussions ont lieu en même temps et que par conséquent une agrégation temporelle entraînerait une perte d'information. De même via le graphe de chevauchement topologique, on remarque que la structure est très recouvrante sur les noeuds, rendant ainsi l'utilisation de partitions statiques de noeuds difficilement envisageable pour décrire les discussions.

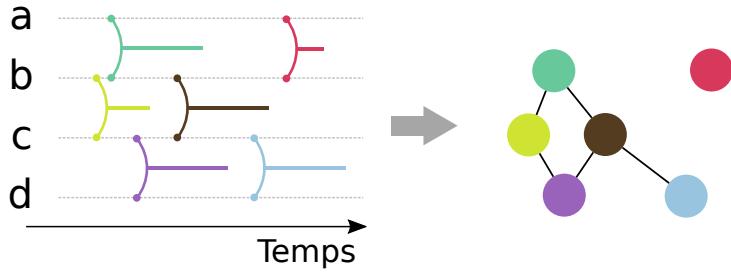


FIGURE 3.12 – Transformation d'un flot de liens avec 4 nœuds (a-d) et 6 liens à gauche en un graphe à droite à 6 nœuds. La couleur d'un nœuds dans le graphe indique le lien du flot qu'il représente.

3.4 Détection de structures denses

À partir du constat que les discussions forment une structure particulière, il est naturel d'essayer de les retrouver automatiquement. Pour y parvenir, il faut un moyen capable de trouver des sous-flots-denses dans le flots. C'est à dire une méthode capturant des groupes de liens qui soient proche temporellement et topologiquement. Il serait tentant d'optimiser directement la densité dans le flot mais ce n'est pas envisageable car un groupe constitué d'un unique lien a une densité de 1. Il faut donc trouver une autre méthode. C'est pourquoi nous avons construit une autre projection du flot en un graphe statique afin d'y appliquer une méthode de détection de communautés. Le problème est alors de réussir à créer une transformation de telle sorte que les informations temporelles et topologiques ne soient pas complètement détruites.

3.4.1 Méthode de détection

Afin de créer une transformation du flot vers un graphe, nous définissons un autre flot de liens, \mathcal{L} , dont les liens ont une durée. À partir de \mathcal{L} , nous créons un graphe non-orienté et non-pondéré $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

Chaque lien du flot est représenté par un nœud. Deux liens (b, e, u, v) et $(b', e', u', v') \in \mathcal{L}$ sont connectés dans le graphe s'ils partagent un nœud et si les intervalles s'intersectent, i.e. $\{u, v\} \cap \{u', v'\} \neq \emptyset$ et $[b, e] \cap [b', e'] \neq \emptyset$, voir figure 3.12. Ainsi, un lien dans le graphe représente une connexion structurelle et temporelle entre deux liens du flot de liens. Les groupes denses dans le graphe représentent donc des groupes de liens connectés temporellement et topologiquement dans le flot.

Il faut donc trouver une manière d'ajouter une durée à chaque lien pour créer \mathcal{L} . Lors du calcul de la densité dans la section 3.3.1, nous avions ajouté une durée arbitraire Δ . Ici, il n'est pas très pertinent d'appliquer la même logique. En effet, si on utilise un Δ faible, alors il n'y aura que très peu de liens dans \mathcal{E} et les nœuds représentant les liens d'une discussions ne seront pas forcément connexes. Il paraît illusoire d'espérer retrouver les discussions dans \mathcal{G} si elles ne sont même pas connexes. Si Δ est très grand alors toute information temporelle est perdue et cela revient à calculer le line graphe du graphe agrégé. C'est pourquoi nous adoptons une autre manière d'ajouter une durée sur les liens.

Pour chaque message m , nous connaissons $p(m)$, le message auquel il répond dans la discussion. Nous définissons alors les liens de \mathcal{L} qui ont une durée de la manière suivante : $(t(p(m)), t(m), a(m), a(p(m))_m)$. Ainsi, deux messages, m_1 et m_2 , se succédant dans une discussion sont par définition reliés topologiquement car $a(m_1) = a(p(m_2))$. Ces deux messages

sont aussi reliés temporellement car nous avons la relation suivante :

$$\begin{aligned}[t(p(m_1)), t(m_1)] \cap [t(\mathbf{p}(\mathbf{m}_2)), t(m_2)] &= \\ [t(p(m_1)), t(m_1)] \cap [t(\mathbf{m}_1), t(m_2)] &= [t(m_1)] \neq \emptyset.\end{aligned}$$

Par construction, une discussion est donc représentée dans \mathcal{G} par un ensemble connexe de nœuds. Un fois \mathcal{G} construit, on peut appliquer un algorithme de détection de communautés.

Avec cette construction, \mathcal{G} contient plus d'1 millions de liens entre 316 569 nœuds pour les 116 999 discussions présentes. Sur ce graphe, nous avons appliqué l'algorithme de Louvain [3] qui optimise la modularité. D'autres algorithmes peuvent également être appliqués s'ils capturent des groupes de nœuds disjoints et qu'ils passent à l'échelle. Les groupes trouvés par Louvain sont des communautés dans \mathcal{G} . Par conséquent, ils sont censé être densément connectés dans \mathcal{G} . Comme un lien de \mathcal{G} correspond à une connexion temporelle et topologique dans le flot, on peut espérer qu'ils correspondent à des groupes denses dans le flot.

3.4.2 Comparaison des partitions

Avant de comparer la structure des discussions, D , et la partition, \mathfrak{D} , trouvée par la méthode de Louvain sur \mathcal{G} , il est nécessaire de décrire cette dernière. Dans la figure 3.13, les distributions cumulatives inverses du nombre de liens, du nombre nœuds et de leur durée sont présentées pour les groupes de \mathfrak{D} . Pour rappel, les même données sont représentées pour les discussions. On remarque tout de suite que \mathfrak{D} contient des groupes beaucoup plus gros en nombre de nœuds et de liens alors qu'ils ont des durées similaires.

Ces deux structures sont donc très différentes mais cela pourrait être dû à l'algorithme de Louvain qui n'est pas adapté pour trouver des groupes denses. C'est pourquoi, nous avons également observé la densité des groupes de D et \mathfrak{D} dans le flot \mathfrak{L} . Le résultat est visible dans la figure 3.13d. Comme les liens de \mathfrak{L} ont une durée, il est possible d'utiliser directement la densité au lieu de la Δ -densité utilisée précédemment. On remarque que les groupes de \mathfrak{D} , bien que plus gros, sont plus denses que les groupes de D . Cependant la distribution cumulative inverse cache les effets de la taille sur la densité. Or à nombre de liens égale (entre 2 et 160), on remarque que les groupes trouvés par Louvain sont plus denses en moyenne ,0.46 contre 0.38. La médiane est également plus élevée : 0.34 contre 0.33. En revanche, les plus gros groupes ($|\mathfrak{D}_j| > 160$) trouvés par Louvain ont une densité plus faible ce qui peut être dû à leur taille.

Si les groupes de \mathfrak{D} sont plus denses, c'est peut être car ils regroupent plusieurs discussions de D dans un groupe. Pour comparer deux partitions, l'indice de Jaccard est classiquement utilisé pour calculer la *précision* et le *rappel* qui sont définis de la manière suivante :

$$\text{précision}(\mathfrak{D}_j) = \max_i \frac{|\mathfrak{D}_j \cap D_i|}{|\mathfrak{D}_j|}, \quad \text{rappel}(D_i) = \max_j \frac{|\mathfrak{D}_j \cap D_i|}{|D_i|}.$$

Dans la figure 3.14, est présenté la *précision* des groupes et le *rappel* des discussions en fonction de leur taille. Chaque point représente la moyenne du *précision* (resp. *rappel*) pour les groupes (resp. discussions) d'une taille donnée. On voit qu'il y a un important rappel et ce même pour les grandes discussions, ce qui veut dire qu'en générale une discussion D_i est totalement incluse dans un groupe \mathfrak{D}_j . En revanche, la précision est très faible car un groupe \mathfrak{D}_j contient plusieurs discussions, ce qui est cohérent avec la taille très importante des groupes de \mathfrak{D}_j .

Il semble donc que la partition \mathfrak{D} soit proche de D mais que ses groupes soient plus gros. Pour circonvenir à ce problème, nous appliquons de manière récursive l'algorithme de Louvain sur chaque graphe induit par un groupe \mathfrak{D}_j . Ce processus permet de subdiviser de manière récursive chaque groupe \mathfrak{D}_j . Soit $\mathfrak{D}_{j'}(h)$ un groupe trouvé au niveau h , par construction, il est

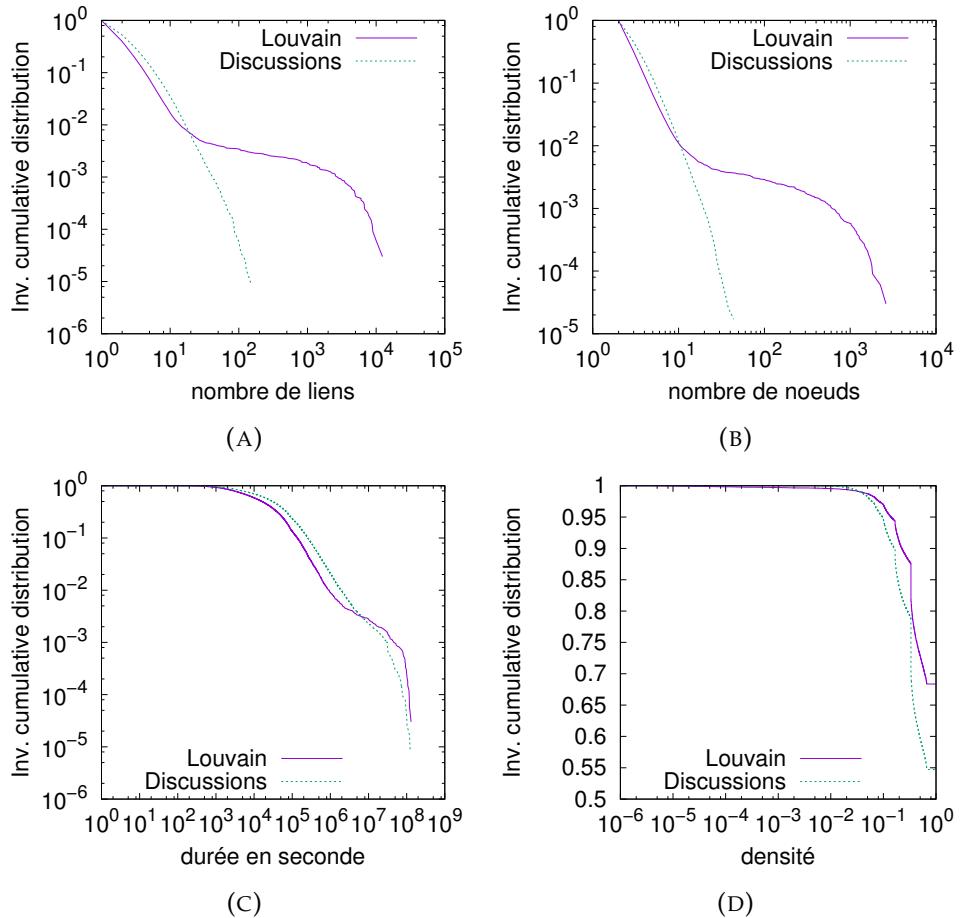


FIGURE 3.13 – Distribution cumulative inverses du nombre de liens (a), du nombre de nœuds (b), de la durée (c) et de la densité (d) pour les groupes trouvés par Louvain et les discussions.

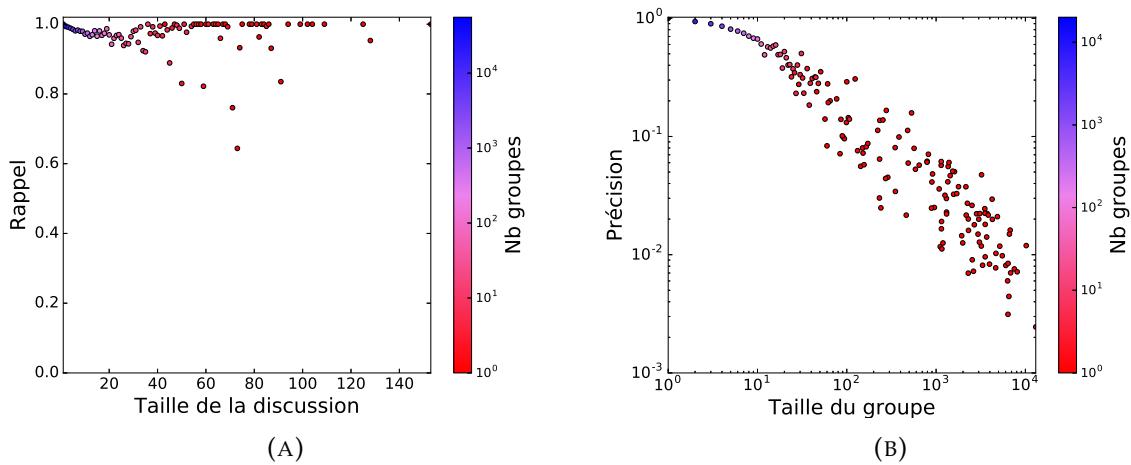


FIGURE 3.14 – (A) Rappel des discussions vis-à-vis des groupes trouvés par Louvain. (B) Précision des groupes trouvés vis-à-vis des discussions. Chaque point représente la moyenne du rappel (resp. précision) pour les discussions (resp. groupes) ayant la même taille. La couleur du point indique le nombre de groupes ayant la même taille.

inclus dans un groupe trouvé au niveau $h - 1$, c'est à dire $\mathfrak{D}_{j'}(h) \subseteq \mathfrak{D}_j(h - 1)$. Le niveau 0 est la première partition trouvée par l'algorithme de Louvain dans \mathfrak{G} .

Soit $D_i \in D$, notons $\mathfrak{D}_{\tilde{j}}(h)$ avec $h \in \mathbb{N}$ le groupe trouvé par la méthode de Louvain au niveau h qui soit le plus proche de D_i au niveau h , c'est-à-dire $|\mathfrak{D}_{\tilde{j}}(h) \cap D_i| = \max_j |\mathfrak{D}_j(h) \cap D_i|$. Avec ces définitions, on observe la relation suivante : $\mathfrak{D}_{\tilde{j}}(h) \cap D_i \subseteq \mathfrak{D}_{\tilde{j}}(h - 1) \cap D_i$. La définition de *rappel* de l'équation 3.4.2 n'est donc pas adaptée pour les niveaux inférieurs et nous l'adaptons de la manière suivante :

$$\text{rappel}(D_i, h) = \max_j \frac{|\mathfrak{D}_j(h) \cap D_i|}{|D_i \cap \mathfrak{D}_{\tilde{j}}(h - 1)|}. \quad (3.3)$$

Ainsi, le *rappel* au niveau h prends en compte le maximum d'élément qu'il est possible de trouver à ce niveau. La définition de *précision* ne pose quant à elle pas de problème. La figure 3.15 représente le *rappel* adapté et la *précision* pour le premier et deuxième niveau de récursion de la même manière que pour la figure 3.14. On remarque que, dès le premier niveau, le rappel baisse et que ce phénomène s'amplifie fortement au niveau suivant. Cela implique que les discussions ne sont plus incluses dans un groupe mais au contraire réparties dans plusieurs. La précision quant à elle augmente légèrement mais cela est dû à la baisse de la taille des groupes trouvés. Il semble donc qu'il ne soit pas possible avec cette approche de retrouver automatiquement les discussions.

3.4.3 Conclusion

Nous avons avec cette méthode mis en évidence des groupes denses. Les groupes trouvés sont plus gros et plus denses que la structure des discussions. Cependant, ces observations ne remettent pas en cause les conclusions faites dans la section 3.3 pour plusieurs raisons. Tout d'abord, les flots de lien étudiés ne sont pas exactement les-mêmes ($L \neq \mathfrak{L}$). Ce changement de flot est nécessaire pour le fonctionnement de la méthode de détection. Ensuite, les deux structures ne sont pas complètement différentes car les groupes trouvés semblent en fait agréger plusieurs discussions. Malheureusement, nous n'avons pas réussi avec notre méthode à isoler chaque discussion malgré notre approche récursive. Pourtant, il semble que la structure trouvée ai du sens au vu des valeurs de densité des groupes.

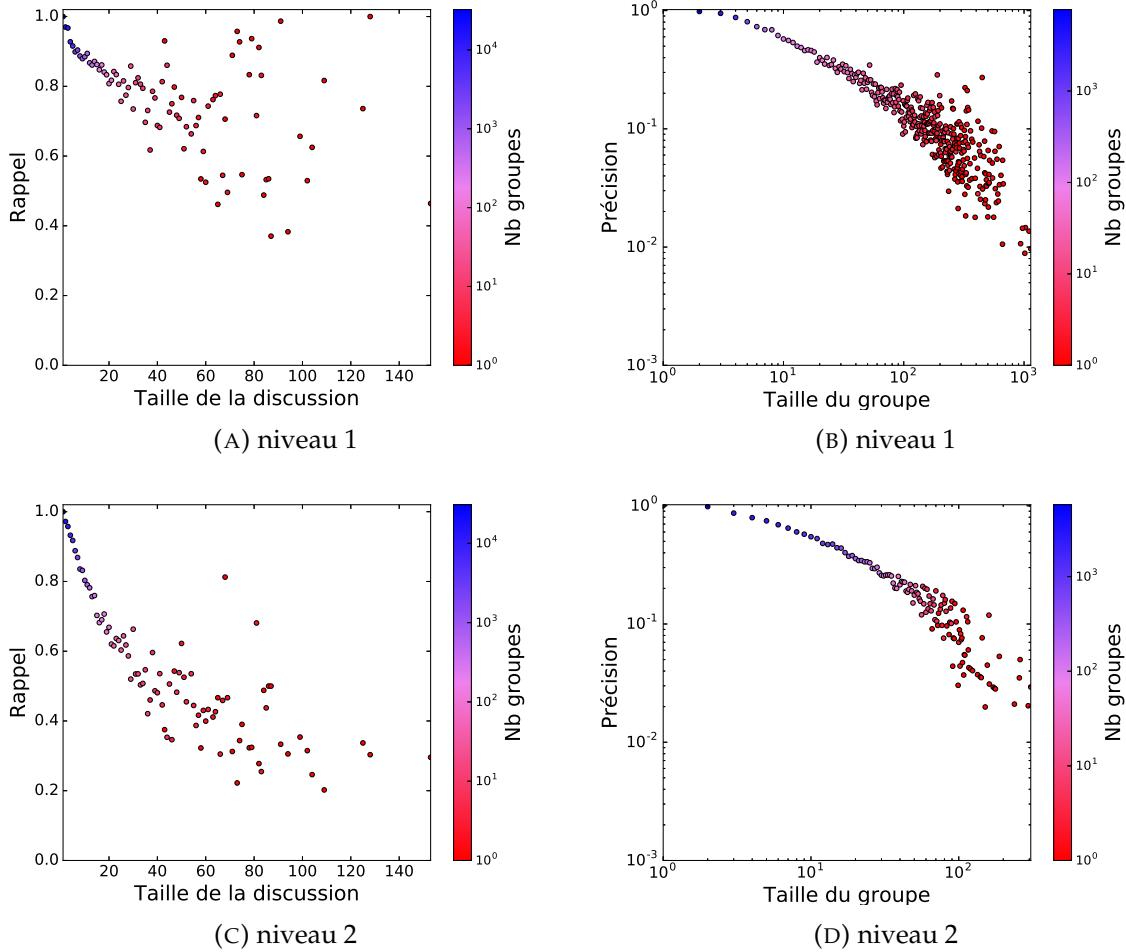


FIGURE 3.15 – (A,B,C) Rappel des discussions vis-à-vis des groupes trouvés par Louvain à différent niveaux récursif. (B,D,F) Précision des groupes trouvés vis-à-vis des discussions. Chaque point représente la moyenne du rappel (resp. précision) pour les discussions (resp. groupes) ayant la même taille. La couleur du point indique le nombre de groupes ayant la même taille.

Chapitre 4

Détection de groupes denses (SNAM)

Sommaire

4.1	Calcul des groupes candidats	27
4.2	Calcul évaluation	27
4.3	Jeux de données	27
4.4	Application	27
4.5	Conclusion	27

4.1 Calcul des groupes candidats

4.2 Calcul évaluation

4.3 Jeux de données

4.4 Application

4.5 Conclusion

Chapitre 5

Expected Nodes : communautés de liens dans les graphes statiques

Sommaire

5.1	Travaux existants	30
5.2	Définition d'Expected Nodes	33
5.3	Comparaison	36
5.3.1	Cas du graphe complet	36
5.3.2	Graphe LFR	37
5.4	Calcul et optimisation	40
5.5	Conclusion	41
5.5.1	Perspective	42

Les structures communautaires dans les graphes ont été beaucoup étudiées lorsqu'elles concernent les nœuds mais également, dans une moindre mesure, pour les liens. Par exemple dans un réseau social, chaque personne a plusieurs centres d'intérêts : famille, sport, politique... Lorsque deux personnes interagissent, la communication a lieu dans un contexte bien particulier. Bien que les personnes aient plusieurs centres d'intérêts, la raison de leur communication est souvent unique. Il semble donc qu'une information importante soit intrinsèquement liée au lien. Via la recherche de partitions de liens d'un graphe, c'est cette information que nous cherchons à capturer.

Afin d'illustrer ce que capture une partition, prenons l'exemple d'un réseau de personnes fictif où le contexte de l'interaction est connue. Certaines personnes communiquent ensemble car elles sont de la même **famille**, pratiquent le même **sport**, jouent au **go** ensemble ou bien encore car elles **travaillent** ensemble. Nous illustrons cet exemple dans la figure 5.1 où les

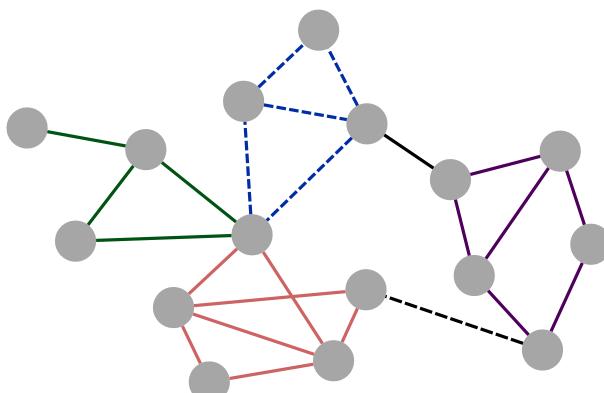


FIGURE 5.1 – Exemple de réseau de personnes avec une structure communautaire sur les liens qui est représentée par la couleur et le style de chaque lien.

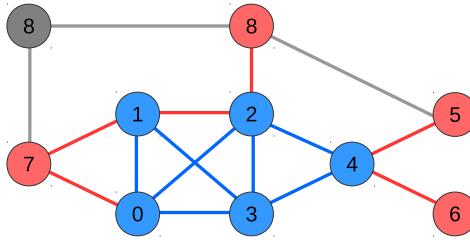


FIGURE 5.2 – Exemple d'un groupe de liens L (en bleu). Les liens rouges sont les liens adjacents L_{out} connectant les nœuds internes V_{in} (en bleu) aux nœuds adjacents V_{out} (en rouge).

interactions sont colorées en fonction de leur contexte. Il est intéressant de noter que les interactions d'un même type se regroupent ensemble. Un nœud peut avoir des interactions de plusieurs types, c'est le cas du nœud central qui a des interactions de type **famille**, **sport** et **go**. De même, certaines interactions font le lien entre différent groupes. C'est le cas du lien pointillé **noir** qui relie le **sport** et le **travail** ce qui pourrait se traduire par le financement de l'équipe par l'entreprise. Ainsi, les partitions de liens peuvent capturer des situations assez variées.

Il est également possible de manipuler des partitions chevauchantes ou couvertures. Dans ce cas, chaque nœud peut appartenir à plusieurs communautés. Pour répondre à ce problème de nombreux algorithmes ont été proposés pour la détection et l'évaluation de couvertures de nœuds, voir la section 1.1.2. Les couvertures sont une généralisation des partitions et aucune méthode ne fait encore consensus pour les évaluer car leur structure est complexe, voir la sous-section 1.1.2. Les partitions de liens, quant à elles, restent des objets plus simples à manipuler. De plus, elles permettent de mettre en avant une autre structure ayant également du sens.

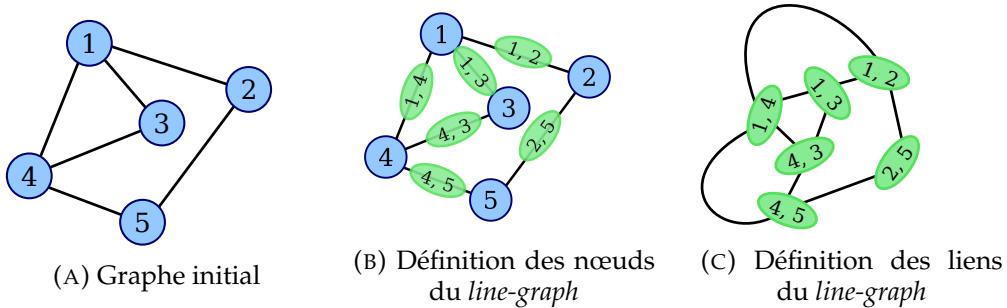
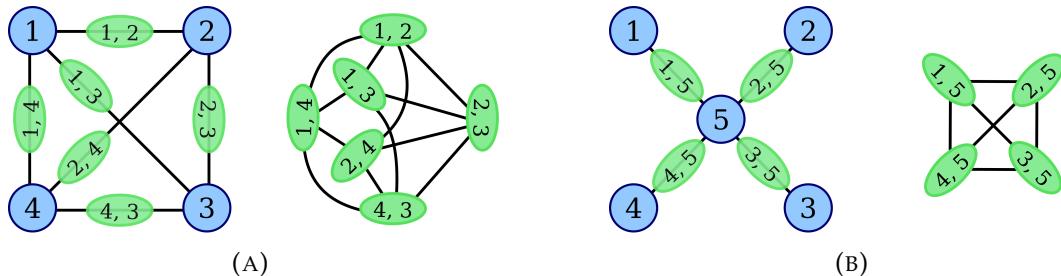
Il apparaît donc que les partitions de liens sont des objets à part entières pertinent à étudier. Pour ce faire, il est nécessaire d'adapter les outils d'analyses pour évaluer directement les partitions de liens. Nous développons ici une approche similaire à ce qui est fait pour les partitions de nœuds et la modularité [24]. Le but est de créer une fonction de qualité permettant d'évaluer une partition de liens d'un graphe.

5.1 Travaux existants

Les notations utilisées sont les suivantes. Soit $G = (V, E)$ un graphe non-orienté avec V l'ensemble des nœuds de taille n et $E \subseteq V \times V$ l'ensemble des liens de taille m . Le degré d'un sommet u de G est noté $d_G(u)$. Une partition des liens en k groupes est notée $\mathcal{L} = (L_1, L_2, \dots, L_k)$ avec $L_i \subseteq E \forall i, L_i \cap L_j = \emptyset \forall i \neq j$ et $\bigcup_i L_i = E$. Pour un groupe de liens $L \in \mathcal{L}$, on pose $V_{in} = \{u \in V, \exists (u, v) \in L\}$ l'ensemble des nœuds internes au groupe L , $V_{out} = \{u \in V \setminus V_{in}, (u, v) \in E \wedge v \in V_{in}\}$ représente les nœuds adjacents au groupe L et enfin $L_{out} = \{(u, v) \in E \setminus L, u \in V_{in} \vee v \in V_{in}\}$ l'ensemble des liens adjacents au groupe L (voir Figure 5.2).

Une approche naïve serait de transformer le graphe initial en un *line-graph*. Un *line-graph* est un graphe où chaque lien du graphe initial est transformé en un nœud dans le *line-graph*. Deux nœuds du *line-graph* sont reliés si les liens correspondants ont au moins un nœud en commun, voir la figure 5.3. Comme un *line-graph* est un graphe classique, on peut y appliquer toutes les méthodes déjà existantes, e.g. les algorithmes de détection de communautés. Ainsi, une partition des nœuds du *line-graph* représente une partition des liens du graphe initiale. Sur l'exemple 5.3, les liens $(1, 4)$, $(1, 3)$ et $(3, 4)$ du graphe forment un triangle dans le *line-graph* et pourraient être capturés comme étant une communauté. Ces liens dans le graphe initial forme également un triangle et peuvent être considérés comme une communauté valide.

1. Image provenant de https://en.wikipedia.org/wiki/Line_graph.

FIGURE 5.3 – Exemple de construction du *line-graph*¹.FIGURE 5.4 – Exemple de construction du *line-graph* d'une clique en (A) et d'une étoile en (B).

Cependant pour que ce type de méthode fonctionne, il est nécessaire que le *line-graph* résultant puisse être analysé comme un graphe. En particulier, il est nécessaire que la notion de communauté dans le *line-graph* est un sens dans le graphe initial. Or, un *line-graph* a une structure très différente du graphe initial.

Prenons pour l'exemple : la clique qui est la meilleure communauté possible et l'étoile qui est une des pires communautés possible. Le but est d'observer comment ses structures sont transformées dans le *line-graph*. Ces situations sont représentées dans la figure 5.4. L'étoile dans la figure 5.4b est transformée en une clique de 4 noeuds. Une des pires structures communautaires d'un graphe est transformé dans le *line-graph* en la meilleure structure communautaire. En effet, chaque noeuds de degré k du graphe initial donne lieu à une clique de taille k dans le *line-graph*. Les cliques du *line-graph* ne sont donc pas forcément des communautés dans le graphe. Dans le cas de la clique 4 dans la figure 5.4a, on remarque que le *line-graph* est composé de 6 noeuds et de uniquement 12 liens. Plus généralement une clique de taille n dans le graphe initial donne lieu dans le *line-graph* à $\frac{n(n-1)}{2}$ noeuds et $(n-1)n$ liens. Ainsi, plus la clique est grande dans le graphe, moins la structure résultante dans le *line-graph* est dense. Les cliques du graphe sont donc moins denses que ses étoiles, lorsqu'on les observe dans le *line-graph*.

Il n'est donc pas innocent d'utiliser le *line-graph* pour trouver des communautés de liens. Il est nécessaire d'adapter les outils à ce type de graphe.

Il existe différents types de méthodes pour la détection et l'évaluation de partitions de liens. Il y a des méthodes évaluant une partition de liens via la transformation de la partition en une couverture de noeuds [15, 22, 33]. Une transformation classiquement utilisée est qu'un noeud dans la couverture prend comme communautés l'ensemble des communautés de ses liens, voir la figure 5.5. Il serait tentant de considérer que les partitions de liens et les couvertures de noeuds sont équivalentes. Ainsi pour évaluer une partition de liens, il suffirait de transformer la partition en couverture. Or, ce changement n'est pas anodin. D'une part, les couvertures de noeuds permettent de modéliser beaucoup plus de situations car il n'y a aucune contrainte sur les couvertures. D'autre part, il n'est pas trivial de transformer une partition de liens en

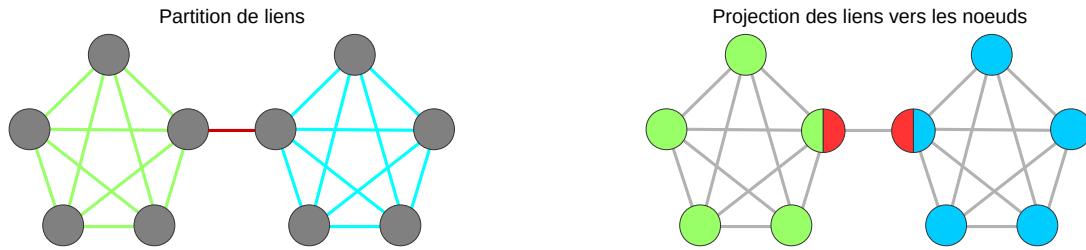


FIGURE 5.5 – Transformation d'une partition de liens à gauche en couverture de nœuds à droite. La couleur représente un groupe.

couverture de nœuds, et *vice versa*.

Dans l'exemple de la figure 5.5, il n'est évident pas que la communauté **rouge** constituée des deux nœuds centraux soit une communauté légitime. Selon le contexte, elle peut être considérée comme un artefact de la transformation. La transformation d'une partition n'est donc pas un acte neutre. Cet aspect a d'ailleurs été mis en avant par Esquivel *et al* [9]. Face à ce problème, nos travaux ainsi que quelques méthodes existantes proposent des méthodes évaluant directement les partitions de liens.

Ahn *et al.* [1] sont parmi les premiers à avoir proposé une méthode détectant les communautés de liens. Leur méthode *link clustering* est une méthode hiérarchique d'agglomération. Elle construit un dendrogramme en agglomérant de manière itérative les groupes de liens en fonction de leur similarité calculée par l'indice de Jaccard. Afin de décider de la coupe du dendrogramme et de la partition résultante, la fonction *Partition Density* est utilisée. Pour une partition de liens donnée \mathcal{L} , la *Partition Density* est définie de la manière suivante :

$$D(\mathcal{L}) = \frac{\sum_{L \in \mathcal{L}} |L| D(L)}{m} \quad D(L) = \frac{|L| - \min_D(|V_{in}|)}{\max_D(|V_{in}|) - \min_D(|V_{in}|)}, \quad (5.1)$$

où $\min_D(N) = N - 1$ est le nombre minimum de liens pour relier N nœuds et $\max_D(N) = \frac{N(N - 1)}{2}$ est le nombre maximum de liens qu'il puisse exister entre N nœuds. Malgré son nom la *Partition Density* n'est pas une densité mais le nombre de liens du groupe normalisé par le nombre de liens minimum et maximum pour un groupe de $|V_{in}|$ nœuds.

Après simplification, on obtient la formule suivante :

$$D(L) = 2 \frac{|L| - (|V_{in}| - 1)}{(|V_{in}| - 1)(|V_{in}| - 2)}. \quad (5.2)$$

Par convention, un groupe constitué d'un unique lien, et qui n'a donc que deux nœuds internes, a une qualité nulle.

D'autres chercheurs [21, 29] ont par la suite utilisé la *Partition Density* comme fonction à optimiser dans des algorithmes génétique. Leurs solutions semblent pour l'instant difficilement utilisable car leurs algorithmes reposent sur de nombreux critères et sont limités à de petit graphes.

Par ailleurs, la *Partition Density* ne peut pas être directement appliquée aux graphes pondérés. Une première proposition de Kim [17] a été faite dans ce sens.

Evans *et al.* [10] proposent trois fonctions de qualité pour évaluer les partitions de liens. Leurs fonctions de qualité sont basées sur trois marches aléatoires qui se déroulent sur les liens du graphe. L'approche est similaire à la modularité car la modularité peut également être définie à l'aide de marche aléatoire sur les nœuds du graphe [7]. Leurs trois fonctions de qualité peuvent être calculées et optimisées sur le graphe mais les auteurs ont montré que l'on pouvait, de manière complètement équivalente, utiliser la modularité sur des *line-graph* pondérés (LG_1 ,

LG_2 , LG_3). Ainsi, il suffit de construire le *line-graph* approprié puis d'utiliser un algorithme existant d'optimisation de la modularité tel que l'algorithme de *Louvain* [3].

Pour construire les *line-graphs* LG_1 , LG_2 et LG_3 , nous définissons $B \in \mathcal{M}_{n,m}$ la matrice d'incidence du graphe G : un élément $B_{i\alpha}$ de cette matrice $|V| \times |E|$ est égale à 1 si le lien α est relié au nœud i et 0 sinon. Les matrices LG_1 , LG_2 et LG_3 sont alors définies de la manières suivante :

	$x = 1$	$x = 2$	$x = 3$
$LG_x(\alpha, \beta)$	$B_{i\alpha}B_{i\beta}(1 - \delta_{\alpha\beta})$	$\sum_{i \in V, d_G(i) > 1} \frac{B_{i\alpha}B_{i\beta}}{d(i) - 1}$	$\sum_{i,j \in V, d(i)d_G(j) > 0} \frac{B_{i\alpha}A_{ij}B_{j\beta}}{d(i)d(j)}$

Soit $k_x(\alpha) = \sum_{\beta} LG_x(\alpha, \beta)$ le degré pondéré dans le line graphe LG_x du nœud représentant le lien α et $W_x = \sum_{\alpha, \beta \in |E|} LG_x(\alpha, \beta)$ la somme des poids des liens. Pour $x \in \{1, 2, 3\}$, la fonction de qualité $Evans_x$ est définie de la manière suivante :

$$Evans_x(\mathcal{L}) = \frac{1}{W_x} \sum_{L_i \in \mathcal{L}} \sum_{e_1, e_2 \in L_i^2} LG_x(e_1, e_2) - \frac{k_x(e_1)k_x(e_2)}{W}. \quad (5.3)$$

Kim *et al.* [18] ont exploré une extension du concept de *Minimum Length Description* (MDL) introduit par Rosvall *et al.* [28] qui est une méthode provenant de la théorie de l'information. Cette extension de la MDL évalue directement une partition de liens, contrairement à l'extension proposée par Esquivel *et al.* [9]. Un avantage de leur méthode est de pouvoir comparer la qualité d'une partition de liens et d'une partition de nœuds avec leur MDL respective. Cependant, leur méthode ne semble favoriser les communautés de liens que dans des cas très limités.

Pour résumer, il existe des méthodes pour capturer et évaluer des partitions de liens dans un graphe. Deux d'entre elles semblent faire consensus pour l'instant. D'une part, la *Partition density* est une fonction de qualité comparant le nombre de liens observé avec les nombres minimum et maximum de liens possibles entre les même nœuds induits. D'autre part, les fonctions de qualité $Evans_x$ se basent quand à elles sur des marches aléatoires sur les liens et d'une certaine manière sur un processus similaire à la modularité. Il n'existe cependant aucune méthode utilisant une comparaison d'une métrique à ce qui est attendu dans un modèle nul. Or, ce processus a été à l'origine de la modularité.

5.2 Définition d'Expected Nodes

Une idée souvent utilisée lors de la détection de communautés de nœuds est qu'une communauté devrait avoir beaucoup de connexion en interne. Pour évaluer ce genre de définition intuitive, il est nécessaire de définir à quoi comparer le nombre de connexions interne. Le choix qui est fait avec la modularité et d'autres méthodes est de définir un modèle nul aléatoire où il n'existe pas de structure communautaire. Le but est de construire un graphe aléatoire qui partage un certain nombre de caractéristiques avec le graphe initial mais dont la structure communautaire a été détruite lors du mélange.

Il existe de nombreux modèles et celui utilisé dans la modularité est le modèle de configuration [2]. Dans ce modèle, le nombre de nœuds et leurs degrés sont fixes mais la répartition des liens est aléatoire. Ainsi pour chaque nœud, ses voisins sont tirés de manière aléatoire avec une probabilité proportionnelle à leur degré. Comme les liens sont mélangés dans ce modèle, on suppose qu'il n'existe plus de structure communautaire dans le graphe.

Pour notre fonction de qualité *Expected Nodes*, nous utilisons également le modèle de configuration. Avant d'aller plus loin et de définir formellement *Expected Nodes*, il est utile d'avoir une définition informelle de la fonction de qualité.

Le but est d'évaluer un groupe de liens. Afin qu'un groupe de liens soit évalué comme une bonne communauté, les liens devraient induire un nombre relativement faible de nœuds internes. En effet, plus le nombre de nœuds internes est faible, plus le groupe de liens ressemble à une clique. De manière similaire à la modularité, nous utilisons le configuration modèle pour calculer le nombre de nœuds interne espéré dans le modèle de configuration. Si le groupe de liens a moins de nœuds internes qu'espérés alors ça indique que le groupe de liens est plus dense et qu'il devrait donc avoir une évaluation élevée.

Il est donc nécessaire de calculer l'espérance du nombre de nœuds interne, μ_G , d'un groupe de liens, L , dans le modèle nul. Un nœud de G est interne si au moins un de ses liens appartient à L . Ainsi pour calculer μ_G , il faut tirer aléatoirement et sans remise $2|L|$ demi-liens parmi les $2|E|$ demi-liens du graphe aléatoire avec la même distribution de degrés. Soit B_u la variable aléatoire correspondant au nombre de fois où le nœud u est tiré. Cette variable suit une loi hypergéométrique $B_u \sim \mathcal{G}(2|E|, d_G(u), 2m)$. Avec cette notation, on définit μ_G de la manière suivante :

$$\mu_G(|L|) = \sum_{u \in V} \mathbb{P}(B_u \geq 1) = \sum_{u \in V} 1 - \frac{\binom{2|E|-d(u)}{2|L|}}{\binom{2|E|}{2|L|}}. \quad (5.4)$$

Voici quelques propriétés de la fonction $\mu_G(|L|)$:

- La fonction μ_G dépend uniquement de la séquence de degrés $\{d_G(v)\}_{v \in V}$ et du nombre de liens.
- Pour une distribution de degrés donnée, la fonction $\mu_G(|L|)$ est une fonction croissante de $|E|$.
- Si $L = E$, alors le nombre de nœuds attendus est bien égal à $|V|$.
- On a $\mu_G(1) \leq 2$, en effet le modèle nul n'interdit pas la présence de boucle.

Avec μ_G , nous pouvons définir la qualité *interne*, Q_{in} d'un groupe de liens L :

$$Q_{in}(L) = \frac{\mu_G(|L|) - |V_{in}(L)|}{\mu_G(|L|)}. \quad (5.5)$$

Avec cette formulation, pour un groupe de taille $|L|$, plus le nombre de nœuds internes est faible, plus Q_{in} sera élevée.

Q_{in} permet d'évaluer la qualité interne d'un groupe mais il faut aussi tenir compte du voisinage. En effet, observer une clique à l'intérieur d'une autre clique n'est absolument pas surprenant. C'est pourquoi, nous définissons également une qualité externe. Le but est d'évaluer comment sont répartis les liens et nœuds adjacents. Pour ce faire, nous allons également comparer le nombre de nœuds adjacents observé au nombre espéré dans le modèle de configuration. Cependant à l'inverse de la qualité interne, la qualité externe est **mauvaise** si jamais le nombre de nœuds adjacent est plus faible qu'espéré. En effet, si il y a beaucoup de liens adjacents pour peu de nœuds, alors cela indique que le voisinage du groupe est dense et devrait être inclus dans le groupe. Le cas idéal est que chaque lien adjacent soit relié à un nœud différent.

Soit $\bar{d}(L, u) = \sum_{v \in V} \mathbf{1}_{(u,v) \in E \setminus L}$ le degré de u limité aux liens adjacents et $\bar{d}(L) = \sum_{u \in V_{in}(L)} \bar{d}(L, u)$. L'espérance du nombre de nœuds adjacents est calculé comme le nombre de nœuds tirés lorsque $\bar{d}(L)$ demi-liens sont choisis aléatoirement et sans remise dans le modèle de configuration où les liens de L ont été préalablement retirés. Ce graphe aléatoire a la distribution de degrés suivante : $\{d_{G \setminus L}(u)\}_{u \in V}$ où $G \setminus L = (V, E \setminus L)$. Dans ce cas, on ne tire pas aléatoirement un lien mais uniquement un demi-lien car l'autre demi-lien est un des demi-liens reliés aux nœuds internes. L'espérance du nombre de nœuds adjacents se définit de la manière suivante :

$$\mathbb{E}[\bar{d}(L)] = \mu_{G \setminus L}(\bar{d}(L)/2). \quad (5.6)$$

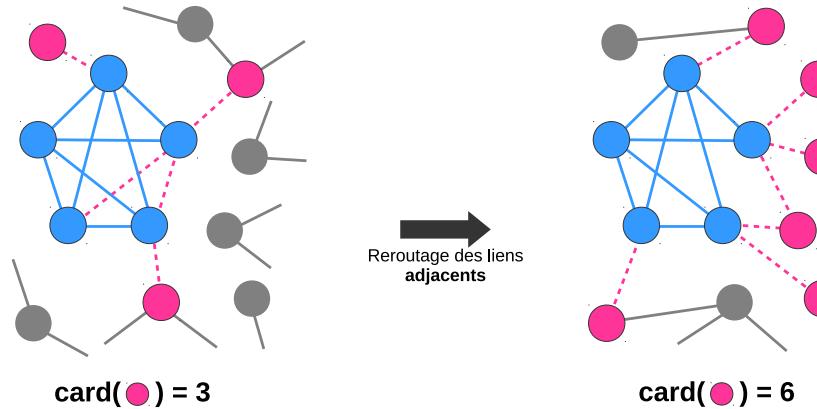


FIGURE 5.6 – Groupe de liens L en bleu et ces liens adjacents en rose pointillés dans le graphe initial à gauche. À droite, une réalisation du modèle de configuration où L a été figé.

Une illustration de ce processus est présentée dans la figure 5.6. Sur cette illustration, le groupe L a un très mauvais voisinage et cela se reflète par un nombre de nœuds adjacents observés plus faible qu'espéré.

Comme il est intéressant de pénaliser les groupes ayant de mauvais voisinage mais qu'un bon voisinage n'est pas suffisant pour définir une bonne communauté, nous bornons à 0 la qualité externe :

$$Q_{ext}(L) = \min \left(0, \frac{|V_{out}(L)| - \mu_{G \setminus L}(\bar{d}(L)/2)}{\mu_{G \setminus L}(\bar{d}(L)/2)} \right). \quad (5.7)$$

Enfin, nous définissons *Expected Nodes* pour un groupe L :

$$Q(L) = 2 \frac{|L|Q_{in}(L) + |L_{out}|Q_{ext}(L)}{|L| + |L_{out}|}. \quad (5.8)$$

La qualité interne est due aux liens de L et la qualité externe aux liens adjacents. C'est pourquoi nous pondérons Q_{in} par $|L|$ et Q_{ext} par $|L_{out}|$.

Nous détaillons maintenant certaines propriétés des formules 5.7 et 5.8 découlant des propriétés de μ_G en nous appuyant sur des exemples. En s'intéressant aux nœuds adjacents V_{out} , on pénalise la présence de nœuds adjacents fortement connectés avec les nœuds incidents à L . Prenons le cas extrême où L est une clique qui est incluse dans une plus grande clique alors que le reste du graphe est quelconque. Dans cette situation, $Q_{in}(L)$ est maximum car le groupe est une clique. En revanche, $Q_{ext}(L)$ va fortement pénaliser la qualité du groupe car il y a dans ce cas beaucoup de liens adjacents pour très peu de nœuds adjacents. Ainsi, la Q_{ext} permet de pénaliser les nœuds adjacents ayant beaucoup de liens avec L . L'idée est que ces liens adjacents devraient être intégrés à L .

Bien évidemment, la solution n'est pas de systématiquement d'intégrer les liens adjacents car intégré un lien adjacent peut également faire baisser Q_{in} . Par exemple, un lien reliant deux groupes denses disjoints peut avoir une qualité positive. Cette situation est visible dans la figure 5.5 partie gauche. Ce lien tout seul a d'une part une qualité interne positive car $\mu_G(1) \leq 2$ et d'autre part il a une qualité externe nulle car chaque nœud adjacent n'est relié qu'à un seul lien adjacent ce qui est le cas idéal. Dans une situation plus générale, la qualité d'un lien isolé dépend du nombre de triangles dans lequel il se trouve. Moins le nombre de triangle est élevé, meilleure sera la qualité externe.

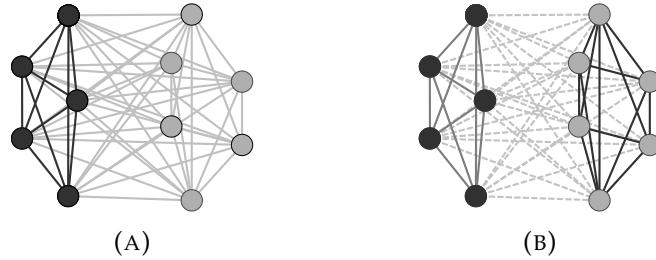


FIGURE 5.7 – Deux partitions de liens pour un graphe complet à 10 nœuds avec $p = 5$: (a) partition en deux groupes et (b) partition en trois groupes. Les nœuds noirs sont les nœuds appartenant à V' et la couleur d'un lien correspond à son groupe.

Enfin il est intéressant de noter que la qualité du groupe contenant tout les liens est nulle. En effet comme dit précédemment, $Q_{in}(E)$ égal à zéro, voir l'équation 5.4. De plus, si le groupe contient tout les liens alors il n'y a plus de liens adjacents et donc seule $Q_{in}(E)$ influe sur $Q(E)$.

Nous définissons *Expected Nodes* pour une partition de liens \mathcal{L} comme la moyenne pondérée de la qualité de chaque groupe :

$$Q_G(\mathcal{L}) = \frac{\sum_{L \in \mathcal{L}} |L| Q(L)}{|E|}. \quad (5.9)$$

D'autres choix de pondération pour Q_{in} , Q_{ext} et Q_G ont été testés en utilisant le nombre de nœuds au lieu du nombre de liens mais elles ont été abandonnées lors des tests qui sont présentés dans la section 5.3.

5.3 Comparaison

Nous évaluons maintenant *Expected Nodes* en utilisant deux jeux de test. Sur ces jeux de tests, nous appliquons également des fonctions de qualités reconnues : *Partition Density* [1] et les fonctions de qualité proposées par Evans *et al.* [10] que nous nommons *Evans1*, *Evans2* et *Evans3*. Pour chaque graphe de test, nous créons empiriquement plusieurs partitions de liens et nous évaluons chaque partition avec toutes les fonctions de qualité.

5.3.1 Cas du graphe complet

Le premier jeu de test est assez simple puisqu'il s'agit d'un graphe complet. Le but est de vérifier que *Expected Nodes* n'a pas un comportement dégénéré. Nous étudions un graphe complet de 100 nœuds². Sur ce graphe, nous définissons plusieurs partitions. La première est la partition triviale où tout les liens sont dans un unique groupe. Nous définissons également deux familles de partitions : une séparant les liens en deux groupes et une séparant les liens en 3 groupes. Soit V' un ensemble de p nœuds où p est un paramètre $p < |V|$. Les deux familles de partitions placent les liens de $V' \times V'$ dans un groupe. Pour la partition en 2 groupes, tout les autres liens sont mis dans un second groupe. Pour la partition en 3 groupes, les liens de $V \times V \setminus V'$ sont dans un second groupe et le reste dans un troisième. Ces répartitions sont illustrées dans la figure 5.7.

Comme le graphe est un graphe complet, la meilleure solution est d'avoir un seul groupe contenant l'ensemble des liens, *i.e.* la partition triviale devrait avoir une meilleure évaluation que les autres partitions. La figure 5.8 présente les résultats. Pour chaque valeur de p et chaque fonction de qualité, nous calculons les évaluations des partitions en deux et en trois groupes

2. Nous avons obtenus des résultats similaires pour un graphe de 500 nœuds.

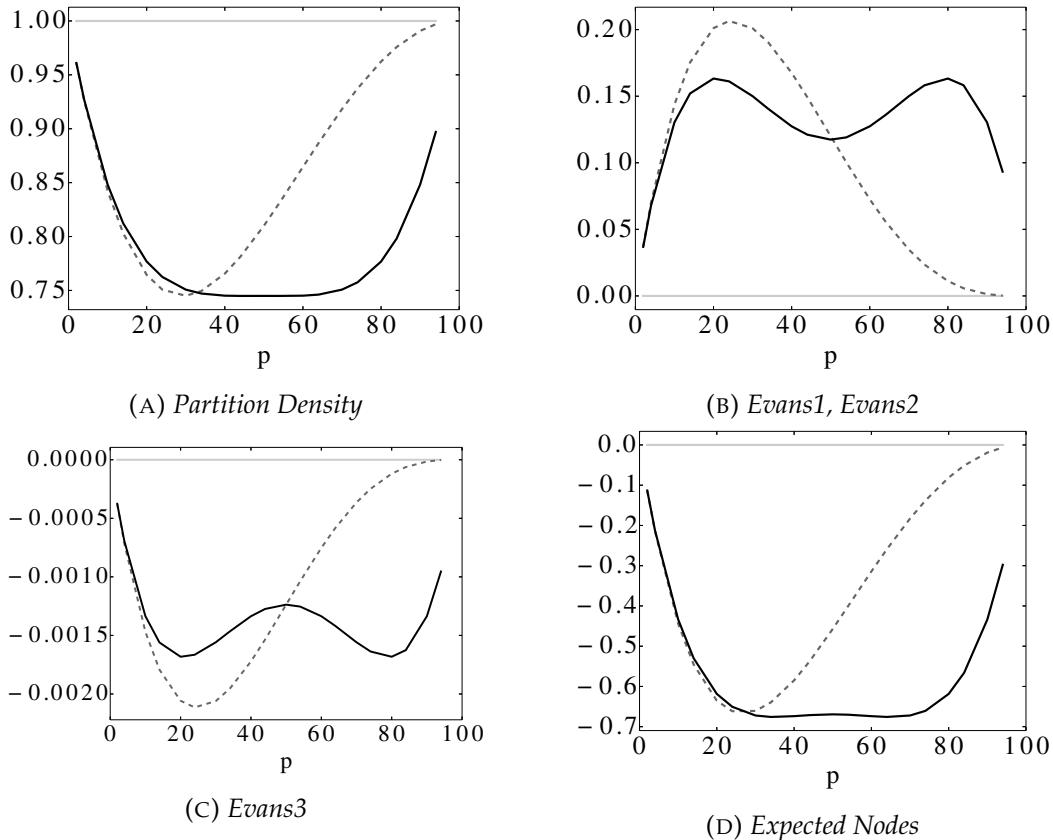


FIGURE 5.8 – Évaluation des 5 fonctions de qualité sur un graphe complet de 100 nœuds pour trois type de partitions. Les partitions testées sont présentées dans la section 5.3.1. Par définition, les résultats pour *Evans1* et *Evans2* sont identiques. Les lignes en gris, noir et pointillées représentent respectivement la partition triviale, les partitions en deux groupes et les partitions en trois groupes.

ainsi que l'évaluation de la partition triviale. L'évaluation de la partition triviale n'est pas dépendante de p et n'est calculée qu'une seule fois. Tout d'abord, la construction de cet exemple est symétrique pour la partition en deux groupes. En effet, la partition en deux groupes lorsque p égal 24 est complètement équivalente à la partition en deux groupes lorsque p égal 76. Par ailleurs, il est possible de définir formellement la qualité de chacune de ses partitions selon toutes les fonctions de qualité. Cependant, la complexité des équations, surtout pour *Expected Nodes* et *Evans_x*, rend le résultat difficilement interprétable.

De manière assez surprenante, les fonctions *Evans1* et *Evans2* ne passent pas ce test car elles évaluent la partition en deux ou trois groupes comme meilleure que la partition triviale. Selon la *Partition Density*, *Expected Nodes* et *Evans3*, la partition triviale est la meilleure des partitions. La fonction *Evans3* diffère légèrement des deux autres car elle a une amplitude plus faible ($\approx 10^{-3}$).

5.3.2 Graphe LFR

Nous utilisons maintenant un jeu de test plus évolué. Il n'existe pas à notre connaissance de générateur de graphe aléatoire avec une structure communautaire sur les liens. C'est pourquoi, nous utilisons le générateur proposé par Lancichinetti *et al.* [19]. Ce générateur aléatoire permet de générer des graphes ayant une structure communautaire chevauchante sur les nœuds. Comme nous voulons évaluer une partition de liens, il est nécessaire de transformer cette vérité

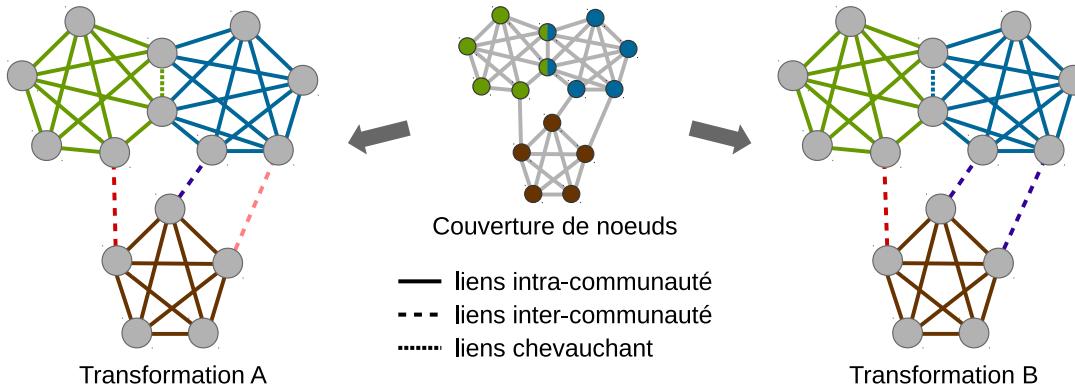


FIGURE 5.9 – Construction de TA et TB depuis une couverture de nœuds. La couleur des liens indique leur groupe.

de terrain. Nous introduisons deux transformations de la couverture des nœuds en deux partitions de liens, TA et TB , voir figure 5.9.

Reprendons l'exemple d'un réseau d'interactions de personnes où chaque personne appartient à différents groupes. Afin de simplifier l'exemple, nous considérons qu'il n'existe que deux groupes : *travail* et *amis*. Le but est de déterminer le type d'une interaction à partir des groupes des personnes.

Si deux personnes partagent un seul groupe, par exemple *travail*, alors il est clair que l'interaction entre ces deux personnes devraient également être de type *travail*.

Si deux personnes ne partagent aucune communauté, l'une est dans *travail* et l'autre dans *amis*, alors plusieurs choix sont possible pour le type de l'interaction. Soit l'interaction a un type *travail-amis* car on considère que toutes les interactions reliant deux personnes des groupes *travail* et *amis* sont de même types. Soit l'interaction a son propre type car on considère qu'elle est unique.

Enfin, deux personnes peuvent partager plus qu'une communauté si les deux sont dans les communautés *travail* et *amis*. Dans ce cas, l'interaction peut être due à l'un de ces deux groupes indépendamment.

On définit maintenant de manière plus formel cette transformation qui également illustrée dans la figure 5.9. Soit $u, v \in V$, $C_{u,v}$ désigne l'intersection des communautés de u et v dans la couverture et $U_{u,v}$ désigne leur union. Nous définissons le groupe d'un lien $(u, v) \in E$ dans TA et TB de la manière suivante :

intra-communauté si $|C_{u,v}| = 1$ alors (u, v) est dans la communauté $C_{u,v}$;

inter-communauté si $|C_{U,V}| = 0$ alors dans TA , (u,v) appartient à sa propre communauté.

Dans TB , les liens appartiennent à la communauté $U_{u,v}$, qui contient l'ensemble des liens (u', v') tel que $U_{u',v'} = U_{u,v}$;

chevauchant si $|C_{u,v}| > 1$ alors le lien (u, v) appartient aléatoirement à une des communautés appartenant à $C_{u,v}$.

Pour générer le graphe, nous avons appliqué un jeu de paramètre classiquement utilisé dans la littérature [11]. Ainsi, nous avons générés des graphes de 500 nœuds ayant un degré moyen de 25, un degré max de 50 et 10 nœuds appartenant à deux communautés et des communautés ayant une taille comprise entre 20 et 100. Le degré est tiré selon une loi exponentielle de paramètre -2 et la taille des communautés a -1 comme paramètre. Enfin, 90% des liens se sont à l'intérieur d'une communauté et les 10% restants sont répartis de manière aléatoire. Avec ces paramètres, il y a en moyenne 5620 liens intra-communautés, 625 liens inter-communauté et seulement 5 liens chevauchant.

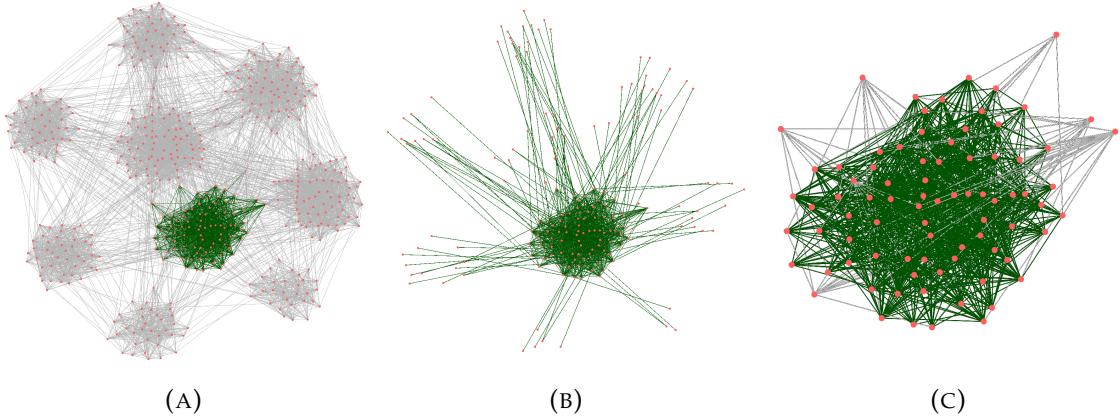


FIGURE 5.10 – Exemple de graphe généré par le LFR en (A) avec une communauté de la vérité de terrain mise en avant en vert. En (B) zoom sur une groupe détecté par $E2$ dont les liens sont en verts. En (C) zoom sur une groupe détecté par LC dont les liens sont en verts.

Pour chaque graphe généré, nous testons les partitions TA et TB mais aussi la partition LC trouvée par *link clustering* [1] et la partition $E2$ trouvée par la seconde méthode de Evans *et al.* [10]³. Ces deux algorithmes optimisent respectivement *Partition Density* et *Evans2*. Ces partitions sont ensuite évaluées par les fonctions de qualité *Partition Density*, *Evans2* et *Expected Node*. Une illustration d'un graphe généré et des exemples de groupes capturés par LC et $E2$ sont présentés dans la figure 5.10.

Dans LC , TA , TB et $E2$, il y a 720, 650, 70 et 11 groupes en moyenne. Afin d'observer la ressemblance de ces partitions, nous avons utilisé la NMI [6]. Il apparaît que les partitions TA et TB sont les plus proches. Ensuite, nous remarquons que la partition $E2$ diffère de TA et de TB uniquement sur les liens inter-communautés. En effet si ils ne sont pas pris en compte lors de la comparaison, alors $E2$, TA et TB sont équivalentes. Il semble en effet que les liens inter-communautés soient arbitrairement distribués entre les plus grosses communautés adjacentes, ce qui est visible dans la figure ???. Enfin, la partition LC , bien que proche de TA et TB , est un peu plus différente. Ces 720 groupes semblent plus petit mais aussi plus denses que ceux de TA ou TB . En particulier, les liens intra-communautés peuvent être séparés en plusieurs groupes, comme dans la figure ???. Les quatre partitions sont donc différentes et mettent en avant différentes caractéristiques.

Nous procédons maintenant à l'évaluation de ces partitions par les différentes fonctions de qualités. Comme le processus de génération de graphe est aléatoire, les évaluations présentées dans la figure 5.11 représentent 30 générations. On remarque tout d'abord que ni TA ni TB n'a la meilleure évaluation selon *Evans2* (figure 5.11b) ou *Partition Density* (figure 5.11a) même si TA et TB représentent nos vérités de terrain. Dans le cas de *Evans2*, c'est la partition $E2$ qui obtient la meilleure évaluation. Cela prouve l'efficacité de l'algorithme $E2$ pour optimiser *Evans2* mais remet en cause la pertinence du critère $E2$ pour mettre en avant la vérité de terrain. Notre fonction *Expected Nodes* se comporte différemment des 2 autres. Tout d'abord, c'est la vérité de terrain TA qui obtient la meilleure évaluation puis il s'agit de TB ou LC selon les générations. Notre mesure semble donc bien mettre en avant la vérité de terrain générée. De plus, *Expected Nodes* évalue différemment les partitions TA et TB contrairement à *Partition Density* et *Evans2*. C'est un point important car, dans la partition TB , les liens inter-communautés peuvent donner lieu à des groupes non-connexes dans TB , voir figure 5.9. Ce phénomène est très pénalisé par *Expected Nodes*. Enfin selon *Expected Nodes*, la partition $E2$ est une mauvaise partition et cela est également dû aux liens inter-communautés. En effet en les fusionnant à

3. Les résultats sont similaires en utilisant $E1$ et $E3$.

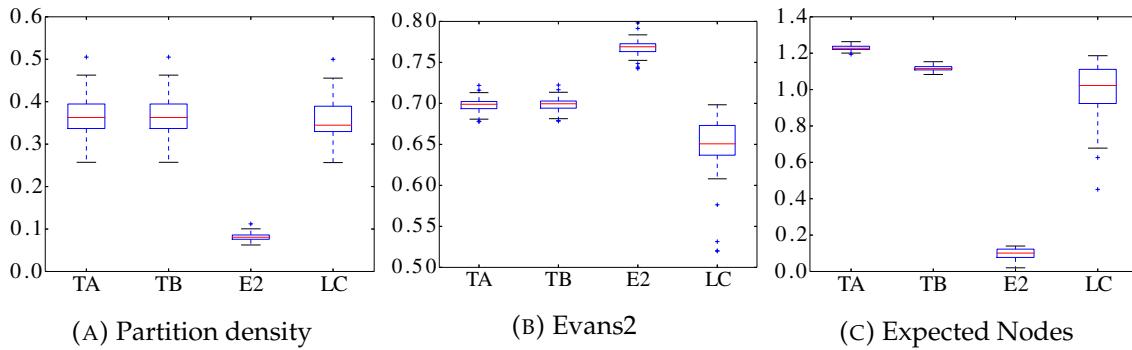


FIGURE 5.11 – Boîte à moustache des évaluations des trois fonctions de qualité pour les différentes partitions de liens. La boîte représente le premier et troisième quartile ainsi que la médiane. Les moustaches s'étendent sur 1.5 fois l'écart interquartile. Les croix sont les points au delà des moustaches.

une communauté adjacente, cela augmente fortement le nombre de noeuds interne ce qui fait baisser Q_{in} .

Avec ce test, nous mettons en évidence les limites des fonctions de qualités existantes. Les fonctions de qualité $Evans_x$ ne peuvent pas permettre de capturer les vérités de terrain générée par LFR. En effet, l'algorithme de Louvain est capable de trouver une partition ayant une évaluation plus élevée que la vérité de terrain. La *Partition Density*, quant à elle, souffre surtout de son incapacité à différencier clairement des partitions différentes. Cela se matérialise par des évaluations très similaires pour les vérités de terrain et la partition *LC*. Il en résulte que la *Partition Density* est sensible aux faibles perturbations. Lors de ces tests, *Expected Nodes* ne semble pas souffrir de ces défauts. Pour ces raisons, nous pensons que *Expected Nodes* est une mesure qui permet de bien évaluer les partitions de liens.

5.4 Calcul et optimisation

Jusqu'à maintenant, nous avons évalué *Expected Nodes* sans nous attacher ni à son calcul ni à son optimisation. Nous discutons maintenant de la complexité de calcul d'*Expected Nodes* pour une partition donnée. Le calcul de la qualité interne 5.5 nécessite d'évaluer la probabilité qu'un nœud soit tiré. Ce calcul de probabilité nécessite d'évaluer pour un nœud u : $1 - \frac{\binom{2|E|-d(u)}{2|L|}}{\binom{2|E|}{2|L|}}$,

ce qui peut être assez couteux. Ce calcul correspond à une loi hypergéométrique. Or sous certaines conditions, une loi hypergéométrique peut être approché par une loi binomiale ce qui simplifie le calcul à : $1 - (1 - \frac{|L|}{|E|})^{|L|}$ ce qui est plus rapide. Lors de nos tests, le biais induit par cette approximation reste assez faible, de l'ordre de 0.01% en moyenne. Ce changement de calcul est équivalent à considérer un tirage avec remise au lieu de tirage sans remise. De plus cette probabilité ne dépend que du degré du nœud et du nombre de liens dans le groupe. Donc, tous les nœuds ayant le même degré donnent lieu à la même probabilité. Si l'on considère que l'évaluation de la probabilité pour un nœud peut se faire en $O(1)$ alors, pour un groupe donné, il est possible de calculer sa qualité en $O(|\{d_G(v)\}_{v \in V}|)$. Cette formulation est très efficace lorsque beaucoup de nœuds ont le même degré. Enfin comme la qualité d'un groupe ne dépend que de sa taille, on peut calculer la qualité \mathcal{L} en $O(|\{L_i\}_{L_i \in \mathcal{L}}|)$.

Il est donc assez rapide de calculer Q_{in} . En revanche, la situation est complètement différente pour Q_{ext} . Le processus de calcul est similaire. On peut également appliquer l'approximation de la loi hypergéométrique par une loi binomiale mais deux nœuds de même degré ne vont

plus forcément avoir la même probabilité d'être tirés. En effet, Q_{ext} n'utilise pas le graphe initial mais le graphe $G \setminus L_i$. Pour chaque noeud, il faut évaluer son degré dans ce nouveau graphe. Le calcul de Q_{ext} pour un groupe se fait donc en $O(n)$. Pour l'évaluation d'une partition, il n'est pas non plus possible de considérer comme équivalents des groupes de même taille. Le coût pour évaluer une partition est donc en $O(|\mathcal{L}|n)$. Le code pour évaluer une partition de liens d'un graphe avec *Expected Nodes* mais aussi avec *Partition Density*, *Evans1*, *Evans2* et *Evans3* est disponible en ligne : <https://github.com/ksadof/ExpectedNodes>.

Malgré ce coût élevé, nous avons développé un premier algorithme d'optimisation glouton de *Expected Node*. Le principe de fonctionnement est le suivant. Chaque lien est initialement dans son propre groupe. Puis à chaque itération, on considère deux types de modification de la solution courante. Soit la meilleure fusion de deux communautés soit le meilleur changement de communauté d'un seul lien. On fusionne ou change de communauté un lien si cela améliore la qualité de la partition. Les fusions considérées sont les fusions entre des communautés adjacentes. Les changement de liens se font également que avec une communauté adjacente. On continue de modifier la solution courante tant qu'elle est améliorable par un de ces mouvements. Malheureusement cette approche souffre de deux problème majeurs. Tout d'abord le calcul du gain est couteux et rend impossible l'étude de grands jeux de données. Ensuite dans nos tests dans les graphes générés par LFR, il semble que cette méthode reste bloquée sur des optimum locaux bien plus faibles que la vérité de terrain. Il faudrait donc tester d'autres heuristiques d'optimisation mais aussi travailler sur une méthode de calcul du gain plus optimisée.

Malgré ces limitations, nous avons tout de même tiré parti de cet algorithme naïf afin de tester si une partition donnée peut être améliorée. En effet même si l'algorithme n'est pas adapté pour trouver une partition ayant une qualité élevée en partant de zéro, il peut modifier une partition donnée pour l'améliorer. Nous avons donc utilisé les partitions *TA* et *TB* comme partition de départ de l'algorithme et nous avons observé si l'algorithme est capable d'améliorer les vérités de terrain. Les changements qu'apportent notre algorithme se portent principalement sur les liens inter-communautés. Dans le cas de *TA*, chaque lien inter-communauté constitue une communauté. Or, il se peut que plusieurs liens inter-communautés soient connectés aux mêmes noeuds. Dans ce genre de situation, notre algorithme fusionne ces liens dans une communauté augmentant légèrement la qualité globale. Après optimisation, nous constatons que les partitions *TA* et *TB* peuvent être légèrement améliorées mais qu'elles sont très proches du maximum local lorsque l'on considère notre algorithme d'optimisation.

5.5 Conclusion

Nous considérons de nouveaux critères pour l'évaluation des partitions de liens tenant compte de la répartition des noeuds internes et externes d'un groupe. À partir de ces critères, nous définissons une mesure de qualité, *Expected Nodes*, basée sur la différence entre le nombre de noeuds induits par un groupe de lien et le nombre de noeuds attendu dans un modèle nul. Ce processus est similaire à celui qui a donné lieu à la modularité. En plus de s'inspirer de la modularité, *Expected Nodes* ne prends pas uniquement en compte la répartition de ses noeuds induits mais également celle de noeuds et liens adjacents. Ce changement permet d'avoir une évaluation plus fine d'une communauté. Pour montrer la pertinence de cette nouvelle fonction de qualité, nous évaluons quatre fonctions de qualité de la littérature. Nous montrons que, sur nos jeux de tests, *Expected Nodes* semble être la plus à même de capturer la vérité de terrain et de différencier des partitions différentes. Malgré le coût de calcul élevé d'*Expected Nodes*, nous avons implémenté un premier algorithme d'optimisation agglomératif. Cet algorithme n'est, pour l'instant, pas capable d'obtenir des partitions avec des scores élevés. Cependant il nous a permis de vérifier que les vérités de terrain générées par LFR sont des quasi-optimum locaux

selon *Expected Nodes*. Bien qu'il soit possible d'améliorer légèrement les vérités de terrains, la structure de l'optimum local est très proche de la partition initiale.

5.5.1 Perspective

Les perspectives autour d'*Expected Nodes* sont nombreuses et portent sur deux points : d'une part le calcul et l'optimisation d'*Expected Nodes* et d'autre part les cas d'applications possibles.

Amélioration de l'optimisation d'*Expected Nodes*

Considérer plus de modifications Il semble que les mouvements envisagés pour modifier une solution courante ne soient pas suffisant pour échapper à des maximums locaux qui sont bien inférieurs à la qualité des vérités de terrain. Il faudrait donc en envisager de nouveaux. Inclure la possibilité de diviser une communauté en deux ne semble pas envisageable car il existe trop de nombreuses coupes possibles et le coût de calcul associé est trop coûteux. Il semble, par contre, prometteur de considérer des mouvements associés à un nœud. En effet au lieu de changer un à un les liens d'une communauté, il devrait être possible d'intégrer en un seul mouvement tous les liens adjacents qui sont reliés à un nœud adjacent donné. Pour l'illustrer ce mouvement, reprenons l'exemple d'une clique incluse dans une autre clique plus grande. La solution finale devrait être la clique maximal. Si l'on ne considère que l'ajout de lien un à un⁴, alors il n'est pas possible d'atteindre la clique maximal. L'algorithme est bloqué car l'ajout d'un unique lien détériore la solution courante. Si on considère l'ajout de tous les liens adjacents à un nœud adjacent donné, alors, après l'ajout, le groupe considéré est toujours une clique et la qualité interne ne diminue pas mais en plus on peut espérer faire diminuer la pénalisation de la qualité externe. De cette manière, il devrait être possible d'éviter des maximums locaux. Plus globalement, les mouvements faisant intervenir plusieurs liens d'un seul coup semblent intéressants.

Diminuer le coût de calcul du gain Ici, il ne s'agit pas d'améliorer la performance de l'algorithme en terme de qualité obtenue par la partition trouvée par l'algorithme mais de le rendre plus rapide. Dans sa version courante, l'algorithme est trop coûteux pour traiter des graphes ayant plus de quelques dizaines de milliers de liens. La force de l'algorithme de Louvain réside dans l'existence d'une expression analytique du gain de fusion de deux communautés qui est rapide à évaluer. Dans notre algorithme, la lenteur est justement due au calcul du gain qui est très coûteux. En effet, nous ne calculons pas directement le gain d'un mouvement mais la différence entre la qualité courante et la qualité résultante du mouvement. C'est une différence importante car, dans notre cas, il faut calculer de zéro la qualité des communautés impactées. Une manière de résoudre ce problème serait de ne pas partir de zéro mais de calculer localement les changements ayant un impact. Prenons le cas de la fusion de communautés A et B . Cela est simple pour la qualité interne car elle ne dépend que du nombre de nœuds et liens internes. Le nouveau nombre de liens, après fusion, est simplement la somme des liens des communautés initiales. Le nouveau nombre de nœuds internes est la taille de l'union des nœuds internes. La situation est plus complexe pour la qualité externe. Le nouveau nombre de liens adjacents n'est pas simplement une somme. De même, l'identité et le degré des nouveaux nœuds adjacents dans $G \setminus (A \cup B)$ ne sont pas triviales à calculer. Il faudrait des parcours locaux sur $(A \cup B)$ et son voisinage pour arriver à mettre à jour ces données nécessaires au calcul de Q_{ext} . Si ce parcours est fait suffisamment soigneusement, il est sûrement possible de gagner en terme de vitesse d'exécution.

4. Dans le cas où la fusion de communauté n'est pas une option viable.

Changer la stratégie d'optimisation Une autre source potentielle de gain en vitesse de l'algorithme est la stratégie d'optimisation. Lors d'une itération, nous appliquons la meilleure fusion de communauté ou le meilleur changement de communauté d'un lien. Appliquer le meilleur mouvement peut améliorer la convergence de l'algorithme vers l'optimum local mais cela est coûteux car lors d'une itération tous les gains doivent être calculés. C'est problématique car c'est justement le calcul du gain qui est très coûteux. Une solution face à ce problème est d'utiliser une variante stochastique de l'algorithme. Dans cette variante, ce n'est pas le meilleur changement qui est appliqué mais le premier qui mène à une amélioration de la qualité. Pour ce faire, il est nécessaire de considérer un ordre aléatoire des mouvements possibles et d'appliquer le premier dont le gain est positif puis de recommencer. Avec ce changement, on teste beaucoup moins de mouvements avant d'en appliquer un. Cependant, il est possible que plus de mouvements soient nécessaires avant d'arriver à un optimum local. Par ailleurs avec ce changement, l'algorithme n'est plus déterministe car il dépend de l'ordre d'évaluation des mouvements. Il faudrait donc être très prudent vis-à-vis de ses aspects, lors du test de cette technique.

Cas d'utilisation d'*Expected Nodes*

Nous avons parlé de moyens possible pour trouver une méthode d'optimisation efficace d'*Expected Nodes*. Il est également nécessaire de revenir sur l'utilisation d'*Expected Nodes*. Nous avons testé *Expected Nodes* sur un algorithme largement utilisé dans la littérature et les résultats sont encourageants. Il y a cependant une différence majeur dans la manière dont nous avons testé *Expected Nodes* vis-à-vis des autres fonctions de qualités. Il n'existe pas, pour l'instant, d'algorithme efficace pour trouver une partition maximisant *Expected Nodes*. C'est pourquoi nous testons des partitions trouvées empiriquement par d'autres moyen. Cependant, il existe peut-être une partition que nous n'avons pas testé qui aurait une bien meilleure qualité que les vérité de terrains. L'amélioration de l'algorithme d'optimisation et le test d'autres partition accroiraient la pertinence des partitions testées et ainsi le bien fondé d'*Expected Nodes*. De manière analogue, il serait intéressant d'évaluer *Expected Nodes* sur des jeux de données ayant une réelle vérité de terrain sur les liens, que ce soit des données réelles ou simulées. Plus globalement, il serait intéressant de comprendre ce qu'il est possible de capturer et représenter comme structure avec une partition de liens.

Nous avons parlé jusque maintenant de l'application d'*Expected Nodes* sur des graphes non pondérés et non dirigés. La question de la pondération est délicate à prendre en compte pour *Expected Nodes*. En effet, notre méthode repose sur des tirages aléatoires de demi-liens. La notion de tirage est intrinsèquement lié à des valeurs entières. En effet, il n'existe pas à ma connaissance de pondant continue à la loi hypergéométrique ; on ne tire pas aléatoirement 0.6 demi-liens parmi 14.3 demi liens possible. L'application d'*Expected Nodes* à un graphe pondéré semble donc compromise. Cependant, il devrait être trivialement possible de considérer des multigraphes. Dans les multigraphes, deux nœuds peuvent être relié par plus qu'un lien. Les notions de tirages aléatoires peuvent s'adapter à ce genre graphe assez facilement. Il y a toute fois une différence majeur lorsque l'on considère les multigraphes par rapport aux graphes classiques : les contraintes sur la partition résultat. Dans le cas des multigraphes, il serait sûrement intéressant d'ajouter la contrainte que chaque communauté ne doit contenir qu'une seule occurrence d'un lien donné. Ainsi s'il existe deux liens entre u et v , alors ces deux liens devront être dans deux communautés différentes.

Chapitre 6

Fonction de qualité

Sommaire

6.1	Définition	45
6.2	Générateur de flots de liens avec structure communautaire	45

6.1 Définition

6.2 Générateur de flots de liens avec structure communautaire

Chapitre 7

Conclusion

Bibliographie

- [1] Yong-Yeol Ahn, James P. Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307) :761–764, aug 2010.
- [2] Edward A Bender and E.Rodney Canfield. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory, Series A*, 24(3) :296–307, may 1978.
- [3] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics : Theory and Experiment*, 2008(10) :P10008, oct 2008.
- [4] Michele Coscia, Fosca Giannotti, and Dino Pedreschi. A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining*, 4(5) :512–546, oct 2011.
- [5] Maximilien Danisch, Jean-Loup Guillaume, and Bénédicte Le Grand. Towards multi-ego-centred communities : a node similarity approach. *International Journal of Web Based Communities*, mar 2012.
- [6] Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics : Theory and Experiment*, 2005(09) :P09008–P09008, sep 2005.
- [7] J-C Delvenne, S N Yaliraki, and M Barahona. Stability of graph communities across time scales. *Proceedings of the National Academy of Sciences of the United States of America*, 107(29) :12755–60, jul 2010.
- [8] Remi Dorat, Matthieu Latapy, Bernard Conein, and Nicolas Auray. Multi-level analysis of an interaction network between individuals in a mailing-list. *Ann Telecommun*, 62 :325–349, 2007.
- [9] Alcides Viamontes Esquivel and Martin Rosvall. Compression of Flow Can Reveal Overlapping-Module Organization in Networks. *Physical Review X*, 1 :1–11, 2011.
- [10] T. S. Evans and Renaud Lambiotte. Line graphs, link partitions, and overlapping communities. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 80(1) :016105, jul 2009.
- [11] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3–5) :75–174, feb 2010.
- [12] Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the United States of America*, 104(1) :36–41, 2007.
- [13] S Harenberg, G Bello, and L Gjeltema. Community detection in large-scale networks : a survey and empirical evaluation. *Wiley*, 2014.
- [14] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels : First steps. *Social Networks*, 5(2) :109–137, 1983.
- [15] Lan Huang, Guishen Wang, Yan Wang, Enrico Blanzieri, and Chao Su. Link Clustering with Extended Link Similarity and EQ Evaluation Division. *PLoS ONE*, 8(6) :e66005, jun 2013.
- [16] Rushed Kanawati. Seed-centric approaches for community detection in complex networks. In Gabriele Meiselwitz, editor, *6th international conference on Social Computing and Social Media*, volume LNCS 8531, pages 197–208. Springer International Publishing, 2014.

- [17] Sungmin Kim. Community Detection in Directed Networks and its Application to Analysis of Social Networks. 2014.
- [18] Youngdo Kim and Hawoong Jeong. Map equation for link communities. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 84(2) :026110, aug 2011.
- [19] Andrea Lancichinetti and Santo Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 80 :016118, 2009.
- [20] C. Lekkeikerker and J. Boland. Representation of a finite graph by a set of intervals on the real line. *Fundamenta Mathematicae*, 51(1) :45–64, 1962.
- [21] Zhenping Li, Xiang-Sun Zhang, Rui-Sheng Wang, Hongwei Liu, and Shihua Zhang. Discovering link communities in complex networks by an integer programming model and a genetic algorithm. *PloS one*, 8 :e83739, 2013.
- [22] Sungsu Lim, Seungwoo Ryu, Sejeong Kwon, Kyomin Jung, and Jae-Gil Lee. LinkSCAN*: Overlapping community detection using the link-space transformation. In *2014 IEEE 30th International Conference on Data Engineering*, pages 292–303. IEEE, mar 2014.
- [23] FD Malliaros and M Vazirgiannis. Clustering and community detection in directed networks : A survey. *Physics Reports*, 2013.
- [24] M. E J Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 69, 2004.
- [25] MEJ Newman. Community detection in networks : Modularity optimization and maximum likelihood are equivalent. *arXiv preprint arXiv:1606.02319*, 2016.
- [26] M Planté and M Crampes. Survey on social community detection. *Social Media Retrieval*, 2013.
- [27] Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 74(1), 2006.
- [28] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences of the United States of America*, 105(4) :1118–23, jan 2008.
- [29] Chuan Shi, Yanan Cai, Di Fu, Yuxiao Dong, and Bin Wu. A link clustering based overlapping community detection algorithm. In *Data and Knowledge Engineering*, volume 87, pages 394–404, 2013.
- [30] Sulayman Sowe, Ioannis Stamelos, and Lefteris Angelis. Identifying knowledge brokers that yield software engineering knowledge in OSS projects. *Information and Software Technology*, 48(11) :1025–1033, 2006.
- [31] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean François Pineton, Marco Quaggiotto, Wouter van den Broeck, Corinne Régis, Bruno Lina, and Philippe Vanhems. High-resolution measurements of face-to-face contact patterns in a primary school. *PLoS ONE*, 6(8), 2011.
- [32] Yi Bo Wang, Wen Jun Wang, Dong Liu, Xiao Liu, and Peng Fei Jiao. Using Prior Knowledge for Community Detection by Label Propagation Algorithm. In *Advanced Materials Research*, volume 1049-1050, pages 1566–1571, nov 2014.
- [33] Zhihao Wu, Youfang Lin, Huaiyu Wan, and Shengfeng Tian. A fast and reasonable method for community detection with adjustable extent of overlapping. In *Proceedings of 2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering, ISKE 2010*, pages 376–379, 2010.
- [34] Jierui Xie, Stephen Kelley, and Boleslaw K. Szymanski. Overlapping community detection in networks. *ACM Computing Surveys*, 45(4) :1–35, aug 2013.