# Dynamique des graphes de terrain : analyse en temps intrinsèque

Alice Albano

# THÈSE

dirigée par Bénédicte Le Grand et Jean-Loup Guillaume

présentée pour obtenir le grade de

**DOCTEUR EN SCIENCES**
**DE L'UNIVERSITÉ PIERRE ET MARIE CURIE**

spécialité Informatique

---

# Dynamique des graphes de terrain : analyse en temps intrinsèque

---

Alice Albano

soutenue publiquement le 10 octobre 2014 devant le jury composé de

| | | |
|---|---|---|
| *Rapporteurs :* | Florence Sèdes | Professeur, Université de Toulouse |
| | Éric Gaussier | Professeur, Université Grenoble I |
| *Examinateurs :* | Martine Collard | Professeur, Université des Antilles et de la Guyane |
| | Anthony Perez | Maître de Conférences, Université d'Orléans |
| | Marcelo Dias De Amorim | Directeur de Recherche, Université Paris 6 |
| *Directeurs :* | Bénédicte Le Grand | Professeur, Université Paris I |
| | Jean-Loup Guillaume | Professeur, Université de La Rochelle |

Laboratoire d'Informatique de Paris 6

École doctorale EDITE

# Remerciements

Tout d'abord, je souhaite remercier Bénédicte et Jean-Loup pour ces trois années de thèse passées en leur compagnie. Je n'ai pas été une thésarde modèle, et sans vous, j'aurais sans doute arrêté en cours de route. Merci bien sûr pour votre encadrement, vos idées, votre façon de travailler et de me faire confiance. Un grand merci également pour tous les bons moments passés ensemble, les pouet canards, les journées couture, les lancers de peluche, et autres. J'espère bien que la fin de ma thèse ne marquera pas la fin de tout ça !

Je remercie Florence Sèdes et Éric Gaussier qui ont accepté d'être les rapporteurs de mon travail, ainsi que Martine Collard, Anthony Perez et Marcelo Dias de Amorim qui ont bien voulu faire partie de mon jury.

Merci à mes tous mes collègues du LIP6 avec qui j'ai travaillé pendant ces trois années. En particulier, merci à Max pour ta bonne humeur et tes théories pour le moins étranges. Merci à Daniel, Lionel et Serguey pour toutes les discussions en salle de pause. Thomas, je n'ai pas pu te croiser très longtemps au laboratoire, mais j'ai beaucoup apprécié ta compagnie. Un grand merci à Véronique aussi, qui gère de façon magistrale tout l'aspect administratif, aussi complexe soit-il. Merci également à tous mes collègues du CRI pour leur accueil chaleureux quand Bénédicte a rejoint leur laboratoire.

Je souhaite également remercier les membres du conseil de l'EDITE, auquel j'ai participé pendant deux ans. Cette expérience a été très instructive pour moi. En particulier, merci à Marilyn Galopin pour sa gentillesse à mon égard.

Un grand merci à mes cobureaux pendant ces trois ans, qui ont réussi avec plus ou moins de succès à me supporter au quotidien, et qui m'ont beaucoup aidé pendant ma thèse. Sébastien, notre partage de bureau était bien sympathique et je regrette qu'il se soit terminé prématurément. Mais il a été suivi du partage de notre directrice de thèse, ce qui m'a bien consolé. Damien, merci pour le petit koala mignon qui s'est accroché à mon écran pour surveiller la rédaction de ma thèse.

Je voudrais remercier Raphaël qui a été un collègue formidable, mais également un ami. Nous avons partagé longuement nos hésitations et notre cynisme sur nos carrières, et ça m'a beaucoup aidé. Je te souhaite bonne chance pour la suite !

Noé, je suis très contente d'avoir pu te croiser au labo pendant ma dernière année de thèse. Merci beaucoup pour toutes les pauses jeu, qui ont rendu mes journées de travail clairement plus agréables. Merci pour ton soutien en toutes circonstances, et pour ton sens de la contradiction sur lequel on peut toujours compter. Je proposerais bien de te remercier avec des cookies, mais je doute que tu puisses tous les manger !

Merci beaucoup à Julien et Cécile, vous avez toujours été adorables avec moi. Les week-end passés avec vous ont été très agréables.

J'adresse un grand merci à Mel pour sa compagnie pendant mes années de thèse. On a passé des très bons moments ensemble, et j'espère que nous pourrons en passer d'autres.

Je souhaite bien sûr remercier mes parents et mon frère pour leur soutien inconditionnel.

Enfin, Vico, tu partages ma vie, merci pour tout ton soutien et ton amour.

# Table des matières

# Table des figures

Chapitre 1

# Introduction

## Sommaire



## 1.1 Contexte

Nous sommes entourés par une multitude de réseaux d'interactions, issus de contextes très différents : réseau social, réseau d'interactions entre neurones, réseau de transport, réseau de contacts, réseau de cellules, réseau de téléphonie, etc. Ces réseaux d'interactions peuvent être modélisés par des graphes. Un réseau social, comme Twitter ou Facebook, peut ainsi être vu comme un graphe dans lequel les nœuds sont les utilisateurs, et les liens représentent les relations d'amitiés entre ces utilisateurs. Ces graphes modélisant des interactions du monde réel sont appelés **graphes de terrain** (ou *complex networks* en anglais).

Malgré ces origines variées, les graphes de terrain possèdent des **caractéristiques communes**. Notamment, ils ont une distribution de degrés hétérogène, c'est-à-dire que quelques nœuds sont très connectés, et la plupart des nœuds le sont très peu. D'autre part, le diamètre de ces réseaux, c'est-à-dire la distance entre les nœuds les plus éloignés, est en général assez petite. C'est ce qu'on appelle le phénomène de "petit monde".

Il a aussi été montré que les graphes de terrains possèdent une **structure en communautés**, c'est-à-dire en groupes de nœuds très liés entre eux, et peu liés avec les autres. L'identification des communautés permet de mieux comprendre la structure du graphe étudié, quel que soit son domaine. Elle permet par exemple de déterminer de manière automatique les groupes dans un réseau social : personnes travaillant au même endroit, promotion de classe, etc.

Un autre phénomène que l'on étudie sur les graphes dans de nombreux contextes est la **diffusion**. La propagation d'une maladie en est un exemple : une personne

atteinte d'un virus risque de contaminer les gens qu'elle côtoie. Ceux-ci peuvent alors contaminer à leur tour certaines de leurs relations, etc. Le virus se propage ainsi au sein de la population, en suivant les interactions entre les individus. L'étude de la diffusion des maladies fait partie du domaine de l'épidémiologie, mais une diffusion peut se produire dans d'autres contextes : diffusion d'information sur un réseau social ou un réseau de téléphonie, diffusion de fichier sur un réseau pair-à-pair, diffusion de monnaie dans une population, diffusion d'un virus informatique, etc.

Les questions de recherche présentées ci-dessus ont été beaucoup étudiées avec des graphes statiques. Cependant, les systèmes considérés sont en constante évolution, ce qui a donc naturellement donné lieu à l'utilisation de **graphes dynamiques** où les nœuds et les liens peuvent apparaître et disparaître au cours du temps. Par exemple, un graphe modélisant un réseau social en ligne comme Facebook est clairement dynamique : les relations entre les personnes sont en évolution permanente. Lorsqu'une personne s'inscrit, on observe un nouveau nœud dans le graphe. Quand elle ajoute ou enlève des contacts avec d'autres personnes, de nouveaux liens sont créés ou supprimés. L'étude des graphes dynamiques est un domaine de recherche assez récent, qui a émergé depuis une quinzaine d'années et le fait d'observer un graphe qui évolue au cours du temps soulève de nombreuses questions complexes : les mesures classiques sur les graphes statiques, comme la distance, le degré, le diamètre, la centralité ou les chemins ne peuvent pas être toujours être généralisées dans le cas des graphes dynamiques. Il faut définir de nouvelles métriques caractérisant la dynamique.

Les premières études sur les graphes de terrain dynamiques ont cherché à **se ramener à des graphes statiques**, afin de pouvoir réutiliser les connaissances existantes. Pour cela, on peut soit considérer un graphe dynamique comme une succession de graphes statiques, soit agréger toutes les informations d'un graphe dynamique en un unique graphe statique : ce seul graphe contient alors tous les nœuds et connexions qui ont existé au cours du temps dans le graphe dynamique. Cependant, essayer de ramener un phénomène en constante évolution à quelque chose d'immobile engendre nécessairement une perte d'information : même si l'on connaît toutes les connexions qui ont existé au cours du temps dans un graphe, on perd l'ordre dans lequel ces connexions se sont produites. Pour cette raison, les travaux plus récents sur les graphes dynamiques se focalisent sur l'étude de propriétés dynamiques et ne cherchent plus à se ramener à un graphe statique.

## 1.2 Contributions

Dans ce contexte, la façon d'étudier un phénomène dynamique dépend d'un paramètre important, mais souvent peu étudié : **l'échelle de temps** selon laquelle on observe ce phénomène. Selon l'échelle choisie, la dynamique du graphe peut

varier de manière très importante. Par exemple, sur un réseau social (au sein d'une zone géographique donnée), l'activité nocturne est en général plus limitée. Si l'on observe le graphe toutes les heures, cette baisse d'activité pourra clairement être constatée (en mesurant le nombre de nouveaux liens par heure au cours du temps par exemple). Cependant, si l'on observe le graphe toutes les 24 heures, la baisse d'activité chaque nuit ne pourra pas être détectée par la simple observation du nombre de nouveaux liens par jour. Le choix de l'échelle de temps est donc très important pour l'étude d'un graphe ou d'un processus dynamique.

Dans cette thèse, nous proposons d'étudier des graphes et des processus dynamiques en utilisant une échelle de temps adaptée à leur dynamique. Afin que celle-ci soit pertinente quelle que soit la dynamique du phénomène étudié, il faut qu'elle dépende directement du graphe, et pas d'une unité absolue comme la seconde ou la journée. Nous considérons ici une notion de temps relatif, que nous appelons le **temps *intrinsèque***, par opposition au temps "classique", que nous appelons temps *extrinsèque*.

Nous proposons de définir l'unité de temps intrinsèque par l'occurrence d'une modification sur le graphe. Un tel événement peut se définir de manière variée, comme l'apparition ou la disparition d'un lien par exemple. Dans ce cas, chaque variation de lien dans le graphe dynamique correspond à un pas de temps intrinsèque, quel que soit le temps écoulé en secondes depuis l'événement précédent. Il s'agit donc d'une **mesure naturellement liée au graphe**. Le fait de considérer un graphe selon cette échelle de temps intrinsèque change complètement les observations que l'on peut faire. En effet, dans un graphe avec une dynamique importante, le temps ne s'écoule plus de manière linéaire, mais suit les événements du graphe. Ainsi, de nombreuses variations de liens sur quelques secondes correspondent à un grand nombre d'unités de temps intrinsèque.

**La contribution majeure de cette thèse est de proposer et d'évaluer une méthodologie reposant sur la notion de temps intrinsèque pour l'étude des phénomènes de diffusion et du calcul de communautés dans des graphes dynamiques.** L'approche proposée dans cette thèse consiste à considérer le graphe à une échelle de temps qui reflète la dynamique de celui-ci, en la définissant directement par rapport au graphe observé.

## 1.3 Plan du mémoire

Ce mémoire est organisé en trois parties :

- **État de l'art et définition du temps intrinsèque** : Dans le chapitre 2, nous présentons un état de l'art sur l'étude de la dynamique des graphes, puis nous définissons la notion de temps intrinsèque sur un graphe évoluant au cours du temps. Nous présentons également les jeux de données que nous avons utilisés dans nos travaux.

- **Diffusion sur des graphes dynamiques en temps intrinsèque :** nous étudions dans le chapitre 3 des phénomènes de diffusion selon une échelle de temps intrinsèque, et nous comparons les résultats obtenus avec une échelle de temps extrinsèque. Ceci nous permet à la fois de montrer l'intérêt de l'échelle de temps intrinsèque pour mieux comprendre un phénomène de diffusion et la dynamique d'un graphe, et de mettre en évidence le fait qu'un même phénomène observé à deux échelles de temps différentes puisse présenter un comportement très différent. Nous avons appliqué cette méthodologie à trois jeux de données réels issus de contextes variés. Ces trois jeux de données possédant une dynamique variée, nous avons pu montrer l'intérêt de l'échelle intrinsèque dans trois contextes différents, ce qui montre le caractère générique de cette méthodologie. De plus, ces différences nous permettent d'évaluer la pertinence de notre définition du temps intrinsèque au regard de la dynamique du graphe. Ces travaux ont été publiés dans [Albano, 2011], [Albano et al., 2012], [Albano et al., 2013] et [Albano et al., 2014a].
- **Communautés stables au cours du temps :** nous analysons dans le chapitre 4 la pertinence de l'utilisation du temps intrinsèque pour la détection de communautés dynamiques. Pour cela, nous utilisons un algorithme connu, que nous appliquons aux échelles de temps intrinsèque et extrinsèque. La comparaison des communautés obtenues selon les deux échelles de temps nous montre qu'une échelle de temps intrinsèque permet la détection de communautés beaucoup plus significatives et détaillées que l'échelle de temps extrinsèque. De plus, l'utilisation de l'échelle de temps intrinsèque ne nécessite aucun paramétrage de l'algorithme pour fournir des résultats pertinents, contrairement à l'utilisation de l'échelle de temps extrinsèque. Ces travaux ont été publiés dans [Albano et al., 2014b].

Enfin, après une conclusion sur nos travaux et les pistes de recherches qu'ils ouvrent, nous présentons en annexe nos recherches sur un cas particulier de diffusion observée dans un réseau pair-à-pair dynamique, ainsi que sur l'étude de la dynamique et la structure en communautés d'un réseau représentant l'activité cérébrale.

Chapitre 2

# État de l'art et approche proposée

## Sommaire



Les graphes de terrains dynamiques sont issus de contextes très variés. Par exemple, les réseaux de contacts physiques consistent à représenter l'interaction de personnes : deux personnes assez proches en termes de distance possèdent un lien entre elles. Les réseaux de communication sont également des graphes de terrain dynamiques : qui téléphone ou écrit à qui ? Ces réseaux sont également très présents en biologie : réseaux d'interactions de neurones, réseaux de cellules, réseaux d'animaux, etc. Les travaux de [Holme and Saramäki, 2012] recensent des types très variés de réseaux dynamiques. Selon le phénomène que l'on observe, la dynamique du graphe peut varier de manière significative. Dans certains cas, tous les nœuds restent présents tout le temps, et seuls les liens varient au cours du temps. C'est le cas par exemple d'un réseau de neurones. Dans d'autres cas, les nœuds varient également, ce qui rend la dynamique plus complexe, comme dans les réseaux de contacts physiques.

La figure 2.1 montre un exemple de graphe dynamique. À $t = 1s$, nous observons un graphe avec 6 nœuds et 6 liens. À $t = 2s$, aucun changement ne se produit dans le graphe. À $t = 3s$, un lien se crée entre deux nœuds, on obtient donc un graphe à 6 nœuds et 7 liens. À $t = 4s$, un nœud et un lien se créent, et un autre lien disparaît. On a donc au final 7 nœuds et 7 liens. À $t = 5s$, un lien disparaît et un autre apparaît.

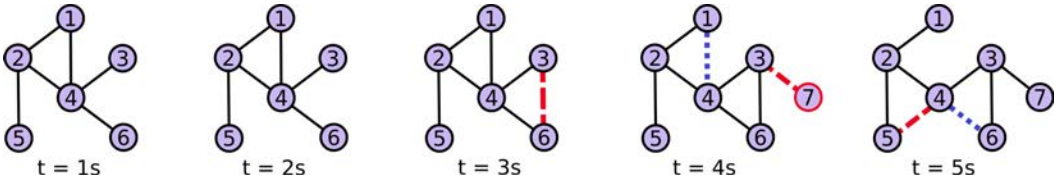



**Figure 2.1** – *Exemple de graphe dynamique.*

Nous voyons sur cet exemple que la dynamique peut être très variable. En effet, à certains moments, seuls les liens varient, et à d'autres, nœuds et liens changent en même temps. Nous pouvons également observer qu'à certains moments, seules des créations ou disparitions de nœuds et de liens se produisent, alors qu'à d'autres moments, les deux phénomènes se déroulent en même temps.

## 2.1 Formalisation d'un graphe dynamique

Dans les travaux existants sur le domaine, les graphes dynamiques sont désignés par différents noms : *temporal networks*, *evolving graphs*, *time-varying graphs*, *time-stamped graphs*, *dynamic graphs*, *link streams*. Tous ces noms désignent le même type d'objets. Deux principales familles de formalisations de graphes dynamiques peuvent être trouvées dans la littérature.

Nous présentons ici une notation pour chacune de ces familles, mais de légères variantes peuvent exister pour chaque cas. Ces variantes sont des modifications mineures et ne modifient pas l'intérêt et les inconvénients de la notation. Une façon de représenter un graphe dynamique consiste à le formaliser sous forme de flots de liens (link streams) [Kempe et al., 2000] : à chaque lien, on associe son temps d'apparition, et éventuellement sa durée si l'information est disponible. On obtient donc une suite de quadruplets au format suivant : $(u, v, t, \delta t)$, où $u$ et $v$ sont des nœuds, $t$ est le début de l'interaction entre $u$ et $v$, et $\delta t$ est la durée de l'interaction. On retrouve quelquefois la notation $(u, v, t_e, t_f)$, avec $t_e$ le temps de début du lien, et $t_f$ le temps de fin du lien. Si l'information de durée du lien n'existe pas, on utilise des triplets $(u, v, t)$ pour formaliser le graphe. Par exemple, si on considère un graphe d'envoi de mails, la seule information temporelle disponible a priori est celle de l'instant d'envoi des mails. Si l'on a besoin d'avoir une durée sur les liens, on peut éventuellement calculer artificiellement une durée de vie des liens,

mais les méthodes pour faire cette estimation peuvent être assez complexes, et le résultat obtenu difficile à évaluer en termes de qualité [Benamara and Magnien, 2010].

Si l'on représente le graphe dynamique de la figure 2.1 avec la formalisation en flot de liens, on obtient la notation suivante :

$G = \{$(1,2,1,5), (1,4,1,3), (2,4,1,5), (2,5,1,5), (3,4,1,5), (3,6,3,5), (3,7,5,6), (4,5,5,5), (4,6,1,4)$\}$

Une autre manière de formaliser les graphes dynamiques, que nous appelons ici notation matricielle, est présentée dans [Casteigts et al., 2011] : le graphe est représenté sous la forme $(V, E, T, \rho, \zeta)$, avec $V$ l'ensemble des nœuds, $E$ l'ensemble des liens, $T$ l'ensemble des instants de temps, $\rho : E \times T \to \{0, 1\}$ une fonction de présence sur les liens, et $\zeta : E \times T \to T$ une fonction de latence qui indique le temps nécessaire pour observer une arête à partir d'un instant donné. Cette notation est beaucoup plus lourde que la notation en flots de liens. Si l'on représente le graphe dynamique avec cette formalisation, on obtient la notation suivante :
$V = \{1, 2, 3, 4, 5, 6, 7\}$
$E = \{(1, 2), (1, 4), (2, 4), (2, 5), (3, 4), (3, 6), (3, 7), (4, 5), (4, 6)\}$
$T = \{1, 2, 3, 4, 5, 6\}$

| $\rho$ | $(1,2)$ | $(1,4)$ | $(2,4)$ | $(2,5)$ | $(3,4)$ | $(3,6)$ | $(3,7)$ | $(4,5)$ | $(4,6)$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 5 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| $\zeta$ | $(1,2)$ | $(1,4)$ | $(2,4)$ | $(2,5)$ | $(3,4)$ | $(3,6)$ | $(3,7)$ | $(4,5)$ | $(4,6)$ |
| 1 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 4 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 3 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 0 |
| 4 | 0 | x | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 5 | 0 | x | 0 | 0 | 0 | 0 | 0 | 0 | x |

Les travaux de [Blonder et al., 2012] fournissent un résumé des approches existantes pour représenter les graphes dynamiques. Selon la notation choisie, certaines opérations peuvent être simplifiées, au détriment d'autres aspects. Le stockage du graphe en notation matricielle est très lourd, car il nécessite deux matrices de taille potentiellement très importante (nombre de liens et ensemble des pas de temps). En revanche, la notation sous forme de flot de liens est très légère. Par contre, l'accès aux informations est en général plus facile dans la notation matricielle : par exemple, savoir un lien est présent à un instant donné est trivial

avec la matrice de présence. Dans la notation en flot de liens, il faut parcourir les liens pour extraire cette information. On peut toutefois remarquer dans la notation matricielle des redondances dans l'information donnée par la fonction de latence. Par exemple, si on sait à un instant $t$ qu'un lien apparaît dans $3$ secondes, on peut en déduire directement qu'à l'instant $t + 1$, le lien apparaît dans $2$ secondes.

En conclusion, la notation matricielle est plus intéressante que la notation flot uniquement dans le cas de graphes très petits, et avec un intervalle de temps d'observation assez court. Dans le cas contraire, la taille des deux matrices devient tellement importante que l'on perd l'avantage de la rapidité d'accès aux informations de cette notation. Dans nos travaux, nous utiliserons exclusivement la formalisation en flot de liens, car elle est très légère et permet de travailler sur des graphes de très grande taille.

## 2.2 État de l'art sur l'étude des graphes dynamiques

Un graphe dynamique étant un objet très complexe à étudier, les premiers travaux pour les représenter et pour les analyser ont cherché à se ramener à des graphes statiques. De plus, de nombreux jeux de données sont agrégés en temps durant la phase de mesure. Par exemple, si l'on mesure un phénomène chaque jour, tous les événements d'une journée se retrouvent naturellement groupés dans le même instant de temps dans le graphe mesuré. Nous présentons dans la suite les principales approches existantes dans la littérature pour étudier un graphe dynamique.

### 2.2.1 Agrégation sur des fenêtres et suivi de propriétés statiques

Certains travaux ont choisi de transformer les graphes dynamiques en une succession de graphes statiques : pour cela, on observe le graphe dynamique agrégé sur des petites fenêtres de temps. La figure 2.2 montre un exemple d'agrégation sur une fenêtre de temps de taille 2.

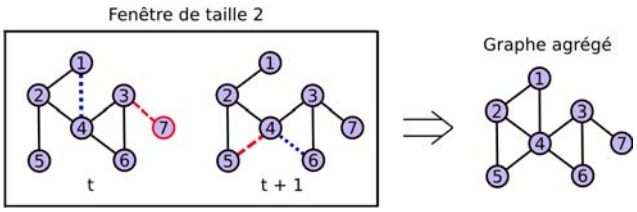



**Figure 2.2** – *Agrégation d'un graphe dynamique sur une fenêtre de temps.*

Le concept de l'agrégation d'un graphe dynamique consiste à construire un graphe statique contenant tous les nœuds et liens observés au moins une fois

pendant la fenêtre de temps. Sur la figure, les liens présents aux deux instants de temps, c'est-à-dire les liens (1,2), (2,4), (2,5), (3,4), (3,6) et (4,6), se retrouvent naturellement dans le graphe agrégé. De plus, les liens présents uniquement sur un des deux pas de temps se retrouvent également dans le graphe agrégé. C'est le cas des liens (1,4), (3,7), (4,5), (4,6).

Cette approche permet de se ramener à des graphes statiques, qui sont mieux connus, et pour lesquels de nombreuses techniques d'analyse existent. Toutefois, l'agrégation sur une fenêtre de temps, même de petite taille, engendre une perte d'information du point de vue de la dynamique. La figure 2.3 détaille le concept d'agrégation sur des fenêtres de temps consécutives ou glissantes, en reprenant le même graphe dynamique que celui de la figure 2.1.

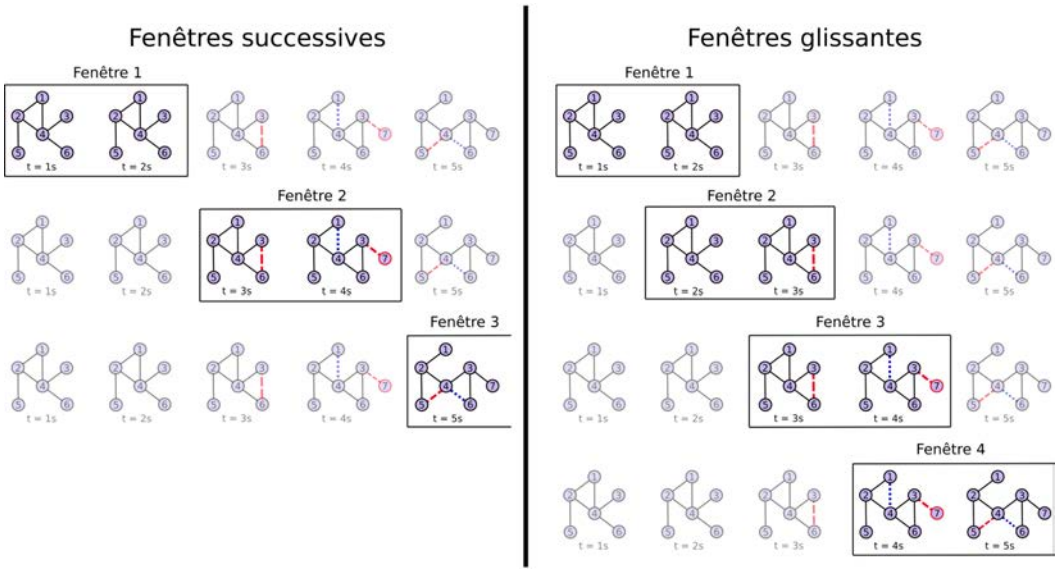



**Figure 2.3** – *Notions de fenêtres successives et glissantes.*

Sur la gauche de cette figure, nous pouvoir voir le graphe dynamique observé à l'aide de fenêtres successives de taille 2. La première fenêtre contient les deux premiers pas de temps, la deuxième contient les troisième et quatrième pas de temps, etc. Sur chacune de ces fenêtres, on considère le graphe agrégé sur les pas de temps contenus dans la fenêtre (comme détaillé dans la figure 2.2). Nous obtenons ici 3 graphes statiques. Sur la droite de la figure, le graphe dynamique est observé avec des fenêtres coulissantes de taille 2. La première fenêtre contient les deux premiers pas de temps, la deuxième fenêtre contient le deuxième et le troisième, etc. À chaque étape, on décale la fenêtre d'un pas de temps. On observe ensuite les graphes statiques obtenus par l'agrégation sur chaque fenêtre.

Les travaux de [Bajardi et al., 2011] étudient un réseau de bétail observé pendant un an et modélisé sous la forme d'un graphe dynamique. Les auteurs agrègent le graphe sur des fenêtres de temps de taille variable. Ils choisissent comme taille de

fenêtre 1 jour, 7 jours, 28 jours, et 365 jours (ce qui correspond donc au graphe agrégé sur l'ensemble de l'année). Ils observent l'impact de la taille de cette fenêtre de temps sur le graphe à l'aide de métriques telles que le nombre de nœuds et de liens sur chaque fenêtre, les poids des liens, ainsi que la distribution de degré. La comparaison de ces différentes propriétés sur l'ensemble des tailles de fenêtres considérées montre que les nœuds et liens centraux dans le graphe dépendent énormément de la fenêtre de temps utilisée. En revanche, la distribution de degrés ainsi que la distribution des poids conservent la même allure dans tous les cas.

Les travaux de [Vernon and Keeling, 2009] étudient également un réseau dynamique de bétail, avec une approche assez similaire : les auteurs comparent le graphe statique et le graphe dynamique agrégés sur des fenêtres de temps de taille variable. Ces travaux mettent en évidence la perte d'information si l'on étudie uniquement le graphe statique, ou si l'on prend une fenêtre de temps trop grande. Pour montrer cela, les auteurs simulent un processus de diffusion de virus sur le réseau. Ils constatent que s'ils considèrent le graphe dynamique avec une fenêtre de temps assez petite, ils observent une transmission plus lente que sur le graphe statique, ainsi qu'un ralentissement de la diffusion de la maladie pendant le week-end. En revanche, sur le graphe statique, ou s'ils utilisent une fenêtre de temps assez grande (une semaine par exemple) tous ces effets disparaissent, et la diffusion est très rapide.

Dans [Leskovec et al., 2005], les auteurs observent des instantanés de graphes dynamiques (donc des graphes statiques) issus de plusieurs domaines différents, et étudient l'évolution de plusieurs métriques au cours du temps. Les réseaux étudiés sont principalement croissants, c'est-à-dire que l'on constate en grande majorité des apparitions de nœuds et de liens, et très peu de disparitions. Les auteurs trouvent que la distribution des degrés moyens au cours du temps suit une loi de puissance et que le diamètre des graphes rétrécit quand le nombre de nœuds augmente. Ces observations pouvaient sembler contre-intuitives au premier abord, et montrent que les graphes de terrain dynamiques ont des propriétés communes, de la même façon que les graphes de terrains statiques, quel que soit le domaine dont le graphe est issu.

[Ribeiro et al., 2013] étudient également l'impact de la fenêtre d'observation sur les graphes dynamiques, en utilisant une marche aléatoire pour montrer les différences entre les aspects statique et dynamique. Pour cela, ils considèrent un marcheur, qui à chaque étape, choisit aléatoirement le prochain lien qu'il va suivre dans le graphe. Afin de comparer les différences obtenues selon la taille de la fenêtre de temps, les auteurs comparent le nombre de nœuds que le marcheur a pu atteindre à la fin du processus. Ils constatent que plus la fenêtre de temps considéré est petite, moins le marcheur couvre de nœuds. En effet, l'agrégation du graphe, même sur une très courte durée, ajoute de nombreux chemins disponibles dans le graphe, et permet donc au marcheur de parcourir plus de liens.

[Clauset and Eagle, 2007] étudient un réseau de contacts, et observent l'évolution du degré moyen et du coefficient de clustering au cours du temps, selon la taille de fenêtre choisie. La figure 2.4 montre l'impact de la taille de la fenêtre de temps sur le degré moyen (à gauche), et sur le coefficient de clustering (à droite). Chacune de ces courbes représente la propriété observée (degré ou coefficient de clustering) en fonction du temps, pour chaque taille de fenêtre.

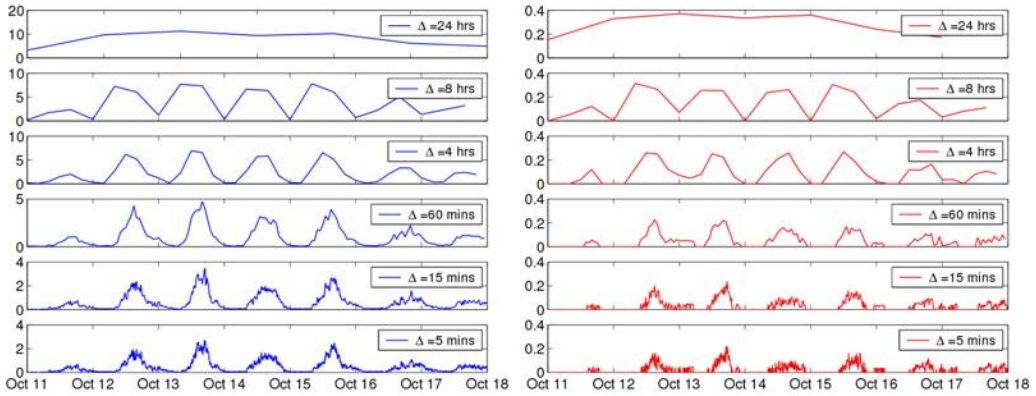



**Figure 2.4** – *Influence de la taille de la fenêtre de temps sur le degré moyen (à gauche), et sur le coefficient de clustering (à droite).*

On peut observer sur ces courbes que pour la plus grande taille de fenêtre (quand $\Delta$ = 1 jour), on perd énormément d'information par rapport à des fenêtres beaucoup plus petites. En effet, le degré moyen et le coefficient de clustering changent peu au cours du temps, et on n'observe aucun événement dans leur évolution. En revanche, avec la taille de fenêtre la plus petite ($\Delta$ = 5 minutes), ces mesures varient beaucoup plus. On observe très clairement un effet jour / nuit, c'est-à-dire un comportement du réseau similaire pendant les différentes journées, ainsi que pendant les nuits, introduisant ainsi des phénomènes périodiques. De nombreuses variations sont également présentes pendant chaque journée. Ce comportement jour / nuit est observé pour toutes les tailles de fenêtre exceptée la plus grande. Toutefois, plus la fenêtre d'observation grossit, plus les mesures sont lissées.

Ces observations sur l'impact de la taille de la fenêtre de temps sont valables sur de nombreux graphes dynamiques. En effet, c'est avec une fenêtre très petite que l'on peut observer le plus d'événements, mais cela engendre aussi beaucoup de bruit. En revanche, avec une taille de fenêtre très grande, on perd énormément d'information liée à la dynamique, et les métriques observées sur une fenêtre trop grande ne sont plus du tout représentatives du graphe.

[Sulo et al., 2010] étudient également les graphes dynamiques sur des fenêtres de temps, et proposent un algorithme pour trouver les fenêtres les plus pertinentes pour observer un graphe dynamique. Cet algorithme prend en entrée un flot de

liens et une métrique (par exemple le degré moyen), et génère ensuite des séries de graphes observés selon différentes tailles de fenêtres de temps. L'algorithme calcule ensuite la valeur de la métrique sur chacun des graphes agrégés et sa variance, ainsi qu'un "ratio de compression" qui mesure le bruit observé. En sortie de cet algorithme, on obtient la taille de fenêtre de temps qui minimise la différence entre la variance de la métrique et le ratio de compression. Avec cette méthode, on peut donc adapter de manière automatique la taille de fenêtre qui sera optimale pour observer une métrique précise. Les auteurs valident ensuite leur algorithme par des tests concluants sur plusieurs graphes dynamiques réels issus de contextes différents (échange d'e-mails, réseaux de contacts, réseaux d'animaux), ainsi que sur un graphe dynamique aléatoire. Cette méthode permet donc une agrégation optimale sur des fenêtres de temps, mais possède quand même les défauts généraux de l'agrégation d'un graphe et donc engendre une perte d'information sur la dynamique du graphe.

Au final, ces travaux permettent en premier lieu de montrer l'importance de considérer la dynamique d'un graphe pour l'étudier, car l'agrégation du graphe fait perdre énormément d'informations. Nous avons vu que selon ce que l'on cherche à observer dans le graphe, l'agrégation en fenêtres assez petites peut rester une solution acceptable pour ne pas perdre trop d'informations liées à la dynamique. En revanche, dans certains cas, la moindre agrégation du graphe, même très petite, change énormément le comportement de ce que l'on étudie, et doit donc être évitée. C'est le cas notamment pour l'étude de chemins temporels.

[Heymann and Le Grand, 2013] utilisent une fenêtre de temps glissante sur un graphe dynamique pour observer le nombre de nœuds du graphe au cours du temps. Cette étude utilise deux échelles de temps différentes pour étudier la dynamique du graphe : une fenêtre dont la taille est mesurée en temps (par exemple une fenêtre de 10 minutes), et une fenêtre dont la taille est mesurée en nombre de créations de liens dans le graphe (par exemple, 10 000 créations de liens). La figure 2.5 montre le nombre de nœuds du graphe observé selon ces deux échelles de temps. Sur la figure du haut, le nombre de nœuds est observé avec une fenêtre coulissante de 10 minutes. Sur la figure du bas, le nombre de nœuds est observé avec une fenêtre coulissante de taille 10 000 liens.

Cette figure met en évidence les différences que l'on observe selon l'échelle de temps que l'on choisit. En effet, les deux courbes, bien que représentant la même métrique sur le même graphe, ont une allure radicalement différente. Sur la courbe du haut observée en temps mesuré en minutes, on observe un effet périodique jour / nuit / semaine très important : on voit des pics d'activité tous les jours, sauf le samedi et dimanche. En revanche, sur la figure du bas, observée avec le temps mesuré en nombre de liens, les effets périodiques disparaissent complètement pour mettre en avant des événements plus isolés. Les observations sur la dynamique obtenues dans les deux cas montrent l'importance du choix de l'échelle de temps choisie sur l'étude d'un réseau dynamique.

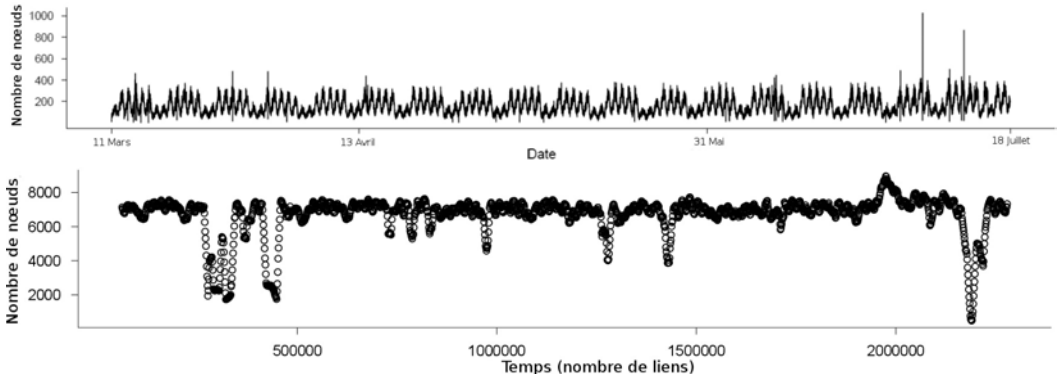



**Figure 2.5** – *En haut : nombre de nœuds en fonction du temps observé avec une fenêtre coulissante de taille 10 minutes. En bas : nombre de nœuds en fonction du temps (représenté en nombre de liens) observé avec une fenêtre coulissante de taille 10 000 liens.*

L'étude de graphes dynamiques au travers de fenêtres permet essentiellement de montrer la perte d'information engendrée par l'agrégation d'un graphe dynamique, ainsi que le suivi de quelques propriétés du graphe. L'étude de propriétés purement dynamiques nécessite une autre approche. Cependant, une nouvelle notion de l'échelle de temps utilisée peut permettre de "capturer" la dynamique du graphe, et rend donc son observation au travers de fenêtres très intéressante.

### 2.2.2 Modèles aléatoires

De nombreux modèles de graphes aléatoires existent pour modéliser des graphes statiques. En revanche, l'élaboration de modèles de graphes dynamiques aléatoires est une problématique beaucoup plus récente. [Scherrer et al., 2008] proposent un modèle qui cherche à reproduire le comportement de réseaux de contacts dynamiques. Le modèle proposé est assez complexe : à chaque pas de temps, chaque lien est actif ou inactif avec une probabilité qui suit une loi, cette loi cherchant à reproduire des propriétés observées sur le graphe réel, comme la distribution des temps inter-contacts par exemple.

Dans [Casteigts et al., 2011], les auteurs abordent rapidement la question de modèles de *time-varying graphs* aléatoires. Une idée consiste à observer une succession de graphes statiques suivant chacun un modèle d'Erdös-Rényi [Erdos and Rényi, 1961]. Ce modèle est toutefois très limité, puisque chaque graphe est indépendant des autres, et le graphe dynamique obtenu ne possède aucune des propriétés des graphes dynamiques réels. En effet, on ne peut observer aucune corrélation temporelle dans ce modèle, ce qui signifie que l'on n'observe aucun événement dans le graphe. De même, la présence d'un lien à l'instant $t$ est indépendante de sa présence à l'instant $t + 1$. En revanche, dans un graphe de terrain, la présence d'un lien n'est pas forcément indépendante d'un instant à l'autre. Les

auteurs citent ensuite [Clementi et al., 2010], qui propose un modèle de graphe dynamique où la probabilité de vie et de mort de chaque arête suit un processus markovien.

[Holme and Saramäki, 2012] proposent une liste des modèles de graphes dynamiques aléatoires existants. Les auteurs citent d'abord trois modèles spécifiques pour reproduire des propriétés de graphes dynamiques de réseaux sociaux et de réseaux de contacts. Ensuite, ils détaillent les modèles utilisés pour rendre aléatoire un graphe dynamique donné. Le plus simple de ces modèles consiste à re-câbler aléatoirement les liens d'un graphe, ce qui retire du graphe toutes les propriétés locales de groupes de nœuds ou d'arêtes. Ce modèle rend la topologie aléatoire. Il existe aussi son "équivalent" pour rendre le temps aléatoire. On fixe la structure du graphe dynamique et on permute aléatoirement les instants de temps. Cette fois-ci, ce sont les corrélations temporelles qui disparaissent. Les auteurs détaillent ensuite quelques autres modèles aléatoires qui sont dérivés de ces modèles-là.

Le choix du modèle aléatoire à utiliser dépend de ce que l'on veut étudier, et quel comportement de réseau on cherche à reproduire. Ainsi, si l'on veut un modèle avec une distribution de degrés homogène et uniquement une dynamique sur les liens, il est sans doute intéressant d'utiliser une généralisation du modèle d'Erdös-Rényi. En revanche, pour modéliser un réseau social, il vaut sans doute mieux utiliser un modèle différent et plus spécifique. La modélisation de graphes dynamiques aléatoires, sur lesquels on maîtrise tout ce qui se passe dans le graphe, est très importante pour pouvoir servir de base à d'autres études.

### 2.2.3 Propriétés dynamiques

De nombreuses métriques ont été définies pour étudier les propriétés des graphes statiques : degré, centralité, chemins, distance, diamètre, densité, etc. Sur les graphes dynamiques, ces métriques deviennent plus complexes ou bien perdent leur sens. Par exemple, le degré d'un nœud devient le degré d'un nœud au cours du temps, ce qui rend la distribution des degrés beaucoup plus compliquée à étudier et à représenter. De même la notion de densité change : on peut considérer la densité au cours du temps, mais elle ne reflète pas le nombre de connexions distinctes entre deux nœuds par exemple : elle ne prend en compte que l'aspect topologique du graphe, et non sa dimension temporelle.

[Holme and Saramäki, 2012], [Nicosia et al., 2013] et [Blonder et al., 2012] recensent les principales métriques existantes sur les graphes dynamiques, en distinguant métriques temporelles et topologiques. Ils détaillent les notions de chemin temporel, connectivité, distance, latence, diamètre, centralité et persistance. Un chemin temporel est défini comme une séquence de contacts pour aller d'un nœud A vers un nœud B, et pour laquelle les temps de présence des liens sont croissants. Intuitivement, cela traduit le fait d'emprunter uniquement les liens présents au

bon moment pour aller d'un point à un autre, plutôt que considérer tous les liens du graphe. La notion de distance, qui dans le cas d'un graphe statique correspond à la longueur d'un plus court chemin d'un point vers un autre, devient différente sur un graphe dynamique. Elle représente le plus court chemin temporel entre deux points. Le temps nécessaire pour parcourir ce plus court chemin temporel est quant à lui la définition de la latence. Ces notions de distance et de latence sont ensuite utilisées pour définir les concepts de centralité et de persistance.

[Kempe et al., 2000] sont parmi les premiers à avoir étudié des métriques spécifiques aux graphes dynamiques. Les auteurs définissent de façon formelle la notion de chemin dans un graphe dynamique. Par la suite, cette définition a été utilisée ou modifiée dans d'autres travaux. Dans [Pan and Saramäki, 2011], les auteurs étudient également la notion de chemin et redéfinissent la corrélation et la centralité pour des graphes dynamiques.

Plusieurs travaux sont dédiés à redéfinir une métrique unique sur un graphe dynamique. Par exemple, dans [Miorandi and De Pellegrini, 2010], les auteurs redéfinissent la notion de *k-shell*, définie initialement sur graphe statique par [Seidman, 1983]. Dans [Viard and Latapy, 2014], les auteurs définissent une nouvelle notion de densité pour les graphes dynamiques. Au lieu d'étudier la densité topologique d'un graphe statique, ils définissent une mesure prenant en compte à la fois des aspects structurels et temporels.

Le développement de nouvelles métriques pour graphes dynamiques permet de faciliter grandement l'étude de ces graphes. Cependant, certains travaux utilisent des approches différentes pour étudier la dynamique d'un graphe. Au lieu de chercher à se ramener à des graphes statique ou bien à définir des métriques dynamiques, ils observent directement le graphe dynamique, et utilisent pour cela des techniques variées. [Eckmann et al., 2004] étudient un graphe dynamique constitué d'échanges d'e-mails sur un serveur d'université. Afin d'étudier le graphe, les auteurs étudient la distribution des temps inter-contact, et observent clairement des effets jour / nuit / week-end dans le réseau. La discussion de l'article est particulièrement intéressante, car elle soulève la question de la signification d'une communauté de nœuds dynamiques, qui peut être très différente du cas statique. Dix ans plus tard, cette question reste ouverte dans ce domaine de recherche.

Dans [Karsai et al., 2014], les auteurs étudient un réseau dynamique de téléphonie mobile. Ils considèrent le réseau de façon égo-centrée, c'est-à-dire qu'ils observent un nœud et l'ensemble de ses contacts, puis ils comparent l'influence des liens forts, c'est-à-dire souvent présents (ce qui représente donc des contacts téléphoniques répétés entre deux personnes), à celle de liens faibles, en observant l'ordre dans lequel le réseau se construit. Pour cela, ils observent la probabilité $p(n)$ qu'un nœud ayant $n$ contacts crée un lien avec une nouvelle personne. $p(n)$ est donc une fonction dépendant du degré du nœud observé. Leurs résultats montrent que, quel que soit le nœud choisi pour centre du réseau, cette probabilité a toujours le même comportement. Les auteurs étudient par la suite la propagation d'une

| | $E(t)$ | $V(t)$ | $N_c(t)$ | $D(t)$ | $T(t)$ | $E_+(t)$ | $E_-(t)$ |
|---|---|---|---|---|---|---|---|
| $E(t)$ | 1 | 0.85 | -0.56 | 0.95 | 0.9 | 0.19 | 0.15 |
| $V(t)$ | 0.85 | 1 | -0.20 | 0.70 | 0.66 | 0.15 | 0.11 |
| $N_c(t)$ | -0.56 | -0.20 | 1 | -0.70 | -0.41 | -0.16 | -0.15 |
| $D(t)$ | 0.95 | 0.69 | -0.69 | 1 | 0.86 | 0.19 | 0.15 |
| $T(t)$ | 0.90 | 0.66 | -0.41 | 0.86 | 1 | 0.15 | 0.11 |
| $E_+(t)$ | 0.19 | 0.15 | -0.16 | 0.20 | 0.15 | 1 | 0.03 |
| $E_-(t)$ | 0.15 | 0.11 | -0.15 | 0.16 | 0.10 | 0.03 | 1 |

**Tableau 2.1** – *Corrélation entre plusieurs propriétés dynamiques d'un réseau de contacts.*

rumeur sur ces réseaux ego-centrés, et trouvent que les liens forts ont comme effet de restreindre la rumeur à un groupe de personne limité. Ce sont les liens faibles qui étendent principalement la rumeur.

[Neiger et al., 2012] étudient cinq réseaux de contacts dynamiques. Les auteurs observent la façon dont bougent les nœuds et les liens, avec une approche locale. Pour cela, ils utilisent le concept de graphe de différence. Ce graphe de différence entre deux graphes $G_1$ et $G_2$, nommé $\Delta G$, est constitué de l'ensemble des nœuds de $G_1$ et de $G_2$, et des liens présents dans uniquement l'un des deux graphes $G_1$ ou $G_2$. En étudiant le nombre de nœuds et de liens impliqués dans les changements entre deux pas de temps, ils trouvent que les nœuds et les liens qui bougent à chaque pas de temps sont très souvent dans une petite zone du graphe et constituent un phénomène local. Ils complètent leur étude en comparant ces résultats avec la façon dont les nœuds et liens évoluent sur un graphe dynamique aléatoire, et constatent que ce dernier ne reproduit absolument pas cette propriété.

Les auteurs de [Rocha and Blondel, 2013] observent un réseau de contacts dynamiques. Après avoir comparé les résultats d'une marche aléatoire sur le graphe dynamique et sur le graphe statique agrégé pour mettre en évidence les différences de comportement entre les deux graphes, ils s'intéressent à la détection et à la signification de motifs temporels, c'est-à-dire dans leur cas, des sous-graphes qui se répètent fréquemment dans le temps.

Les travaux de [Scherrer et al., 2008], dont nous avons déjà parlé dans la section précédente pour les modèles de graphes dynamiques aléatoires, portent également sur l'étude de deux réseaux de contacts dynamiques. Après une approche plus classique d'observation du graphe sous forme d'une série d'instantanés, les auteurs abordent une étude de la dynamique sans se ramener à un graphe statique. Ils regardent le nombre de composantes connexes au cours du temps $N_c(t)$, le nombre de triangles $T(t)$, le nombre de nœuds $V(t)$ et de liens $V(t)$, le degré moyen $D(t)$ et le nombre de liens et de nœuds nouveaux (respectivement $E_+(t)$ et $E_-(t)$) à chaque pas de temps.

Le tableau 2.1 montre la corrélation entre toutes ces mesures de la dynamique. On observe que la plupart des corrélations sont plutôt élevées, ce qui n'est pas très surprenant. En effet, le nombre de triangles et le nombre de liens présentent par exemple une corrélation évidente. Toutefois, on observe dans ce tableau que la création et la disparition de liens sont très peu corrélées avec le reste des propriétés observées, et sont également très peu corrélées entre elles. Les auteurs en concluent que le processus de création et disparition des liens peut être modélisé par un processus Markovien (c'est-à-dire sans mémoire) car il est indépendant de l'évolution des autres propriétés du graphe.

Les travaux de [Cardillo et al., 2013] étudient deux réseaux de contacts en utilisant le concept de modèle nul : ils comparent le graphe dynamique réel avec le même graphe dont les liens sont aléatoires (comme nous l'avons expliqué dans la section précédente sur les modèles de graphes dynamiques aléatoires), afin de différencier la dynamique réelle d'une dynamique aléatoire. Par la suite, ils utilisent des fenêtres de temps coulissantes pour observer des motifs temporels dans le graphe. Ils observent ainsi des effets jours / nuits très importants dans le réseau. Ils étudient enfin des aspects plus sociologiques, comme l'impact de la dynamique du graphe sur l'amélioration de la coopération entre les nœuds du réseau.

## 2.3 Notre approche : échelle de temps intrinsèque

Nous avons vu dans la description de l'état de l'art que l'étude des graphes dynamiques est un domaine assez récent, et que beaucoup de travaux se ramènent à l'étude de graphes statiques successifs. Les travaux proposant des approches pour étudier directement la dynamique d'un graphe sont relativement peu fréquents, comme nous l'avons détaillé dans les sections précédentes. Dans nos travaux, nous basons notre approche sur le changement de l'échelle de temps pour étudier la dynamique d'un graphe, comme les travaux de [Heymann and Le Grand, 2013] ont commencé à le faire.

En effet, l'observation d'un graphe selon une échelle de temps dépendant directement de la dynamique du graphe, que nous appellerons dans nos travaux *temps intrinsèque*, ne nous oblige plus à partitionner le temps en petits intervalles, et à observer le graphe toutes les $x$ secondes. Cela nous permet d'avoir une échelle de temps continue, que nous pouvons paramétrer directement selon des métriques du graphe dynamique. De plus, comme nous le montrons dans la suite, la comparaison de cette nouvelle échelle de temps avec la notion de temps classique peut donner des résultats intéressants, ainsi que des pistes prometteuses pour développer une méthodologie d'étude des graphes dynamiques.

Dans la suite, nous détaillons la définition de temps intrinsèque, ainsi que les travaux ayant déjà utilisé cette notion relative de temps.

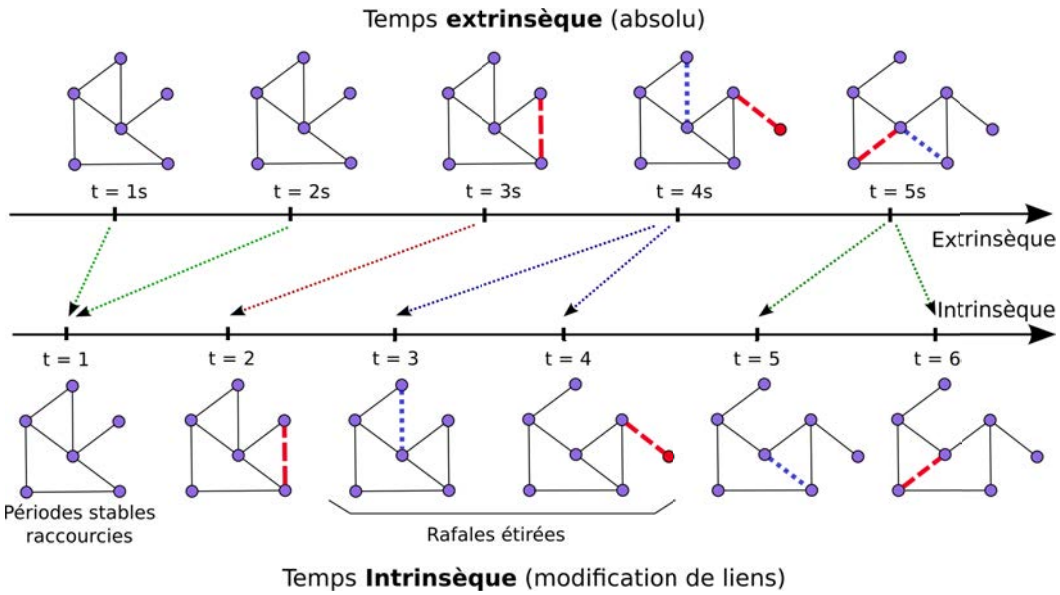



**Figure 2.6** – *Conversion du temps extrinsèque en temps intrinsèque (défini en modifications de liens).*

### 2.3.1 Définition du temps intrinsèque

Dans nos travaux, nous utilisons une définition du temps intrinsèque qui dépend directement de la dynamique du graphe : le temps intrinsèque s'écoule au fil des évolutions du graphe. Ces évolutions peuvent par exemple correspondre à l'apparition ou la disparition d'un nœud, d'un lien, d'un triangle, etc. La figure 2.6 montre un exemple de conversion du temps classique, que nous appelons extrinsèque, vers le temps intrinsèque, en utilisant comme définition d'une unité de temps intrinsèque l'apparition ou la disparition d'un lien dans le graphe.

À chaque création ou disparition de lien dans le graphe correspond donc un pas de temps intrinsèque. Nous voyons sur cette figure qu'aux instants $t = 1s$ et $t = 2s$, le graphe reste identique. Nous n'observons par conséquent pas de nouveau pas de temps intrinsèque. À $t = 3s$, un lien se crée dans le graphe, ce qui donne un nouveau pas de temps intrinsèque. À $t = 4s$, on observe l'apparition de deux nouveaux liens dans le graphe, et donc la création de deux nouveaux pas de temps intrinsèque. On observe le même scénario à $t = 5s$.

Cette figure nous montre également que le temps intrinsèque a tendance à rendre la dynamique du graphe plus uniforme : en effet, les périodes stables en temps extrinsèque sont raccourcies en temps intrinsèque, car il y a très peu de modifications dans le graphe. En revanche, les périodes très dynamiques sont étalées en temps intrinsèque, car le nombre de liens évolue très vite. Cette observation est confirmée par les travaux de [Eckmann et al., 2004], dont nous détaillons le contenu dans la section suivante.

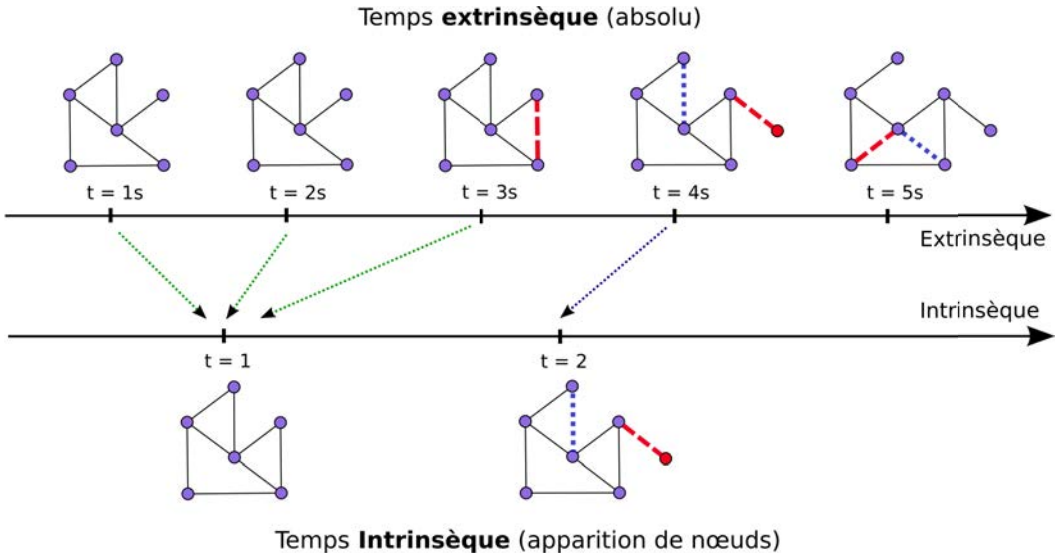



**Figure 2.7** – *Conversion du temps extrinsèque en temps intrinsèque (défini en modification de nœuds).*

Si l'on avait choisi une autre définition de temps intrinsèque, comme l'apparition de nœud par exemple, nous aurions une conversion très différente (voir figure 2.7) : en effet, sur l'exemple décrit, on observe une seule création de nœud à $t = 4s$, ce qui nous donnerait donc deux pas de temps intrinsèque (celui de départ, et celui au moment de la création du nœud). Le choix de la métrique à utiliser pour définir le temps intrinsèque est donc une question très importante. Il faut que cette métrique soit représentative de l'activité du graphe et du phénomène particulier que l'on veut étudier.

Dans cette thèse, nous avons choisi de tester deux définitions très simples pour le temps intrinsèque : l'apparition de lien (pour des graphes croissants uniquement), puis l'apparition et la disparition de liens. Ces deux définitions simples nous permettent de pouvoir étudier l'impact du temps intrinsèque sur la diffusion [1], ou encore la détection de communautés [2]. Nous avons étudié sur plusieurs graphes dynamiques réels l'impact de la définition de temps intrinsèque choisie, pour déterminer si elle est suffisamment représentative de la dynamique du réseau. Dans le chapitre 3, nous étudions uniquement des graphes croissants, et nous prenons comme définition l'apparition d'un lien. Dans le chapitre 4, les graphes que nous étudions varient en taille, et nous prenons comme mesure l'apparition et la disparition de liens.

1. Voir chapitre 3
2. Voir chapitre 4

### 2.3.2 Travaux existants sur la notion de temps intrinsèque

Des philosophes se sont intéressés de longue date à la définition du temps et à la perception que nous en avons [Newton, 1687] [Kant, 1781]. La notion de temps est complexe à définir ; elle est souvent présentée comme un concept absolu, c'est-à-dire que l'écoulement du temps ne dépend pas de la perception du monde et des événements. Mais le temps peut aussi être considéré comme un concept relatif. D'ailleurs, la seconde, qui est utilisée pour définir la notion de temps absolu, a elle-même une définition relative en tant qu'unité de base du système international [de la Convention du Mètre, 2006] : "la durée de 9 192 631 770 périodes de la radiation correspondant à la transition entre les niveaux hyperfins $F = 3$ et $F = 4$ de l'état fondamental $^6S_{1/2}$ de l'atome de césium 133".

Les chaînes de Markov à temps discret ([Kemeny and Snell, 1960]) sont un exemple de l'utilisation de la notion de temps relatif : quand un événement se produit (de façon aléatoire), la chaîne change d'état. Ainsi, on n'observe pas l'écoulement du temps en seconde, mais en nombre d'événements. Les circuits asynchrones ([Van Berkel et al., 1999]) qui sont des circuits logiques, utilisent exactement le même concept. Ils ne dépendent pas d'un signal d'horloge, mais d'un signal indiquant la fin d'une instruction.

Les travaux de [Eckmann et al., 2004] appliquent cette notion de temps relatif. Pour étudier leur réseau d'e-mail, ils utilisent le temps $\Delta_t$ qui correspond au délai entre un message de A vers B et sa réponse de B vers A. Ils constatent que les délais les plus fréquents sont $\Delta_t = 16$ heures et $\Delta_t = 24$ heures. Quand ils observent la distribution des $\Delta_t$ des messages en utilisant comme unité de temps l'envoi d'un message (donc un temps relatif aux changements dans le réseau), les pics observés précédemment disparaissent et la courbe est lissée. De la même façon, cette unité de temps atténue les événements directement liés à la mesure du temps absolu, comme la baisse d'activité la nuit et le week-end. Ces travaux sont donc un premier exemple de l'utilité d'un changement d'échelle pour observer la dynamique d'un graphe : on n'observe pas du tout les mêmes événements selon l'unité de temps considérée.

Plus récemment, les travaux de [Gauvin et al., 2013] se sont intéressés à la distribution des probabilités des temps d'arrivée et son impact sur un processus de diffusion (utilisant un modèle SI) sur un graphe dynamique. Au lieu de considérer des temps d'inter-contacts de manière classique, les auteurs proposent une définition de temps spécifique à chaque nœud : chaque nœud possède ainsi sa propre horloge, qui mesure le temps qu'il a passé en interaction avec d'autres nœuds. La diffusion devient donc indépendante de l'écoulement du temps classique, et cela change beaucoup le comportement du modèle SI [3] (nous reviendrons sur cette question en détail dans le chapitre 3).

3. Nous détaillons ce modèle dans le chapitre 3

La notion de temps intrinsèque reste très peu utilisée dans l'étude de graphes dynamiques. L'un des objectifs de cette thèse est d'approfondir l'utilisation du temps intrinsèque dans ce contexte. En effet, nous avons vu que l'échelle de temps considérée peut changer énormément les observations que l'on peut faire. Dans la suite, nous étudions deux phénomènes dynamiques (la diffusion, et la détection de communautés) observés au travers de cette échelle de temps intrinsèque.

Afin de mesurer l'importance et la singularité de cette notion de temps, nous comparons tous nos résultats avec l'observation de ces phénomènes selon une échelle de temps extrinsèque. Ainsi, nous mettons en évidence l'impact de l'utilisation du temps intrinsèque dans l'étude d'un graphe dynamique.

Dans la section suivante, nous présentons les jeux de données utilisés dans cette thèse pour appliquer notre méthodologie : nous décrivons leur contexte, et donnons quelques statistiques sur leur dynamique (nombre de nœuds, durée de la mesure, etc.).

## 2.4 Jeux de données utilisés

### 2.4.1 Graphes synthétiques Barabási-Albert (BA)

Afin de réaliser des tests, nous utilisons des graphes synthétiques construit suivant le modèle de Barabási-Albert (BA) [Barabási and Albert, 1999], en utilisant les notions de temps extrinsèque et intrinsèque. Dans le modèle BA, les nœuds sont ajoutés un par un au réseau. Chaque nouveau nœud est connecté à un nombre fixé $m$ de nœuds existants du graphe, selon le principe d'attachement préférentiel : plus le degré du nœud $x$ est élevé dans le graphe, plus la probabilité que les nouveaux nœuds se connectent à $x$ est élevée. La probabilité $p_i$ qu'un nouveau nœud crée un lien avec le nœud $i$ est définie de la façon suivante :

$$p_i = \frac{k_i}{\sum_j k_j}$$

avec $k_i$ le degré du nœud $i$, et $j$ un nœud dans l'ensemble des nœuds du graphe.

Les graphes BA sont par conséquent des réseaux dynamiques croissants, au travers du processus d'ajout de nœuds et de liens avec une dynamique simple et régulière. Cette croissance est une caractéristique importante pour le concept de temps intrinsèque. Le modèle BA est caractérisé par quatre paramètres : le nombre initial et final de nœuds dans le graphe, le pas de temps entre la création de deux nouveaux nœuds, et enfin le paramètre d'attachement préférentiel, noté $m$, qui correspond au nombre d'arêtes générées à chaque création d'un nouveau sommet.

Les graphes BA considérés dans cette thèse contiennent initialement 500 nœuds, et nous générons 500 noeuds additionnels en utilisant le modèle BA, ce qui nous donne un graphe de 1000 nœuds à la fin de l'évolution. Afin d'étudier les concepts d'échelles de temps intrinsèque et extrinsèque, nous avons simulé trois dynamiques différentes. La différence entre elles est le délai entre les arrivées des nœuds, qui est généré avec trois lois de probabilité différentes. Nous utilisons différentes lois de probabilité pour la génération des temps d'inter-arrivée car dans le modèle BA, il n'y a pas de notion de temps extrinsèque, seulement la création des liens (qui correspond pour nous à une notion de temps intrinsèque). Plus précisément, les trois lois que nous avons utilisées sont :

- une distribution uniforme où les temps d'inter-arrivées sont équiprobables ; la densité de cette loi est décrite par la fonction $f(x) = \frac{1}{b-a}$, avec $b = 11$ et $a = 1$
- une distribution exponentielle négative, c'est-à-dire que la probabilité d'observer de longs délais décroît exponentiellement ; la densité de cette loi est décrite par la fonction $f(x) = \lambda e^{-\lambda x}$, avec $\lambda = \frac{1}{5}$
- une distribution en loi de puissance, où la probabilité d'observer des délais longs décroît polynomialement ; cette loi est décrite par l'équation $f(x) = ax^k$, avec $k = -2.2$ et $a = 100$.

Nous choisissons les paramètres de ces lois de manière à ce que la moyenne des temps d'inter-arrivées soit égale pour ces trois lois. Ces graphes synthétiques sont utilisés dans le chapitre 3.

### 2.4.2 Réseau Github

#### 2.4.2.1 Description du jeu de données

Github (https://github.com) est une plateforme en ligne, créée en 2008, pour permettre aux développeurs de partager leurs codes open source et de collaborer. Bâtie sur le système de gestionnaire de versions décentralisé Git, cette plateforme facilite les contributions et les discussions en fournissant une interface Web. Github a atteint les trois millions d'utilisateurs le 16 Janvier 2013 ; ces utilisateurs participent à cinq millions de projets de développement.

Le jeu de données extrait de Github que nous utilisons ici décrit l'ensemble de l'activité entre les utilisateurs et les dépôts sur la plateforme Github, entre le 11 Mars 2011 et le 18 Juillet 2012. Nous avons extrait les données sur le site de Github Archive (http://www.githubarchive.org), qui met à disposition un enregistrement de toutes les activités publiques sur Github. À partir de ces données, nous avons construit le graphe de "qui contribue à quel dépôt" ; dans ce graphe, les nœuds représentent les utilisateurs ou les dépôts, et les liens représentent les différentes

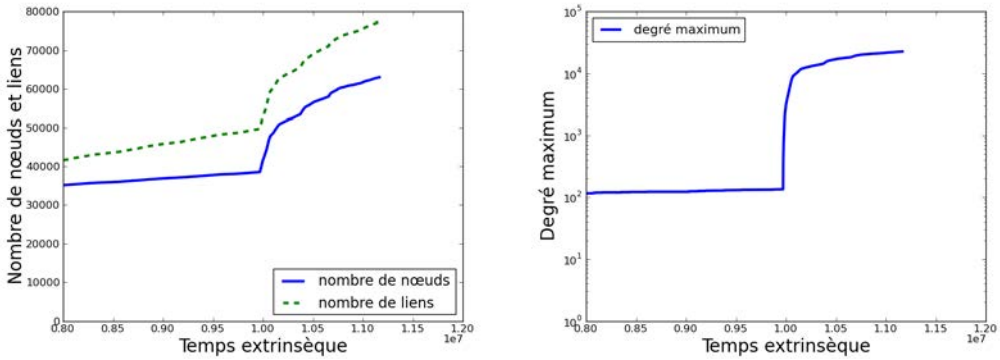



**Figure 2.8** – ***À gauche*** *: nombre de nœuds et de liens dans le graphe Github restreint à sa plus grande composante connexe.* ***À droite*** *: évolution du degré maximal dans le graphe Github, avec l'axe des ordonnées en échelle logarithmique.*

formes d'activité entre utilisateurs et dépôts (commit, push, ouverture et fermeture de bug report, commentaires sur des bugs, requêtes, création ou suppression de branches et de tags, édition du wiki du projet). Le graphe ainsi obtenu est un graphe biparti, qui contient un peu plus 336 000 nœuds et 2,2 millions de liens.

Chaque activité dans ce graphe est associée à une estampille temporelle. Afin d'étudier ce jeu de données, nous considérerons par la suite ce graphe comme croissant : une fois qu'un lien ou qu'un nœud apparait, on le considère présent jusqu'à la fin de la mesure. Cette hypothèse est raisonnable car une fois qu'un utilisateur a contribué à un projet, les modifications qu'il apporte restent dans la durée.

Afin de pouvoir simuler une diffusion sur ce jeu de données, nous avons extrait de ce graphe uniquement la plus grande composante connexe, qui contient un peu plus de 60 000 nœuds et 77 000 liens.

#### 2.4.2.2 Quelques statistiques sur ce jeu de données

La figure 2.8 (à gauche) montre le nombre total de nœuds et de liens dans le graphe au cours du temps (extrinsèque). Nous voyons sur cette courbe que le nombre total de nœuds et de liens croît plutôt lentement, puis d'un seul coup, subit une très brusque augmentation. Ce changement se produit à la date du 4 Juillet 2012, et est corrélé à une augmentation brusque du degré maximal dans la composante connexe. En cherchant ce qui est l'origine de ce changement, nous avons découvert que 506 utilisateurs interagissent avec le projet *Try-Git* à cette date. Ce projet est un tutoriel pour apprendre à utiliser Git, sur lequel Github est basé. La date du 4 Juillet correspond au moment oÙ le projet *Try-Git* a été rendu public (cette information est confirmée par un post sur blog de Github.com [4]).

4. https ://github.com/blog/1183-try-git-in-your-browser

Cet événement modifie radicalement la structure du réseau : en effet, si l'on observe à présent l'évolution du degré maximal dans le graphe (figure 2.8 à droite), on voit un changement très important à partir du 4 Juillet, date à laquelle le tutoriel est mis en ligne. Cela signifie donc qu'un nœud de très haut degré apparaît dans le réseau, ce qui est cohérent avec la popularité du tutoriel décrit dans le paragraphe ci-dessus.

Ce jeu de données présente un comportement assez régulier, excepté l'événement observé au 4 Juillet. Nous utilisons ce jeu de données pour l'étude de la diffusion. En effet, l'observation d'un événement très important à l'aide d'une échelle de temps intrinsèque permet de distinguer si l'événement est lié uniquement à un changement de la structure du graphe, ou bien s'il est aussi lié à un changement de la dynamique du graphe. La phase "stable" du début nous permet de comparer nos résultats avec ceux observés pendant l'événement.

### 2.4.3 Jeu de données Webfluence

#### 2.4.3.1 Description du jeu de données

Le jeu de données Webfluence a été collecté sur les blogs francophones entre le 1er Février 2010 et le 1er Juillet 2010 (5 mois) dans le contexte du projet ANR Webfluence. Ces blogs ont été sélectionnés par une entreprise spécialisée dans les blogs et l'analyse d'opinion (http://linkfluence.net), en fonction de leur popularité et de leur activité au sein de la blogosphère francophone. Le jeu de données est constitué de 10 309 blogs, 848 026 billets ou posts et 1 079 195 citations. Sur un blog, le ou les auteur(s) écrivent des billets, qui sont datés. Chaque blog est donc constitué d'une suite de billets. Les citations dans ce réseau sont des billets qui citent un billet d'un autre blog.

Nous avons extrait un graphe à partir de ces données, que nous appelons le graphe des blogs. Ce réseau est un graphe dirigé acyclique, dans lequel il y a un lien depuis le blog A vers le blog B si A cite B, c'est-à-dire s'il existe au moins un lien de citation depuis un billet du blog A vers un billet plus ancien du blog B. En termes de diffusion d'information, nous considérons que le contenu de B se diffuse vers A. Avec ces hypothèses, nous éliminons de notre graphe les blogs qui ne sont jamais impliqués dans des citations, c'est-à-dire les blogs dont aucun billet n'est cité, et dont les billets n'en citent aucun autre à l'extérieur du blog (dans le jeu de données).

#### 2.4.3.2 Quelques statistiques

Nous étudions tout d'abord le nombre de nœuds dans le réseau de blogs (figure 2.9). Nous observons sur cette courbe que le nombre de nœuds suit une croissance

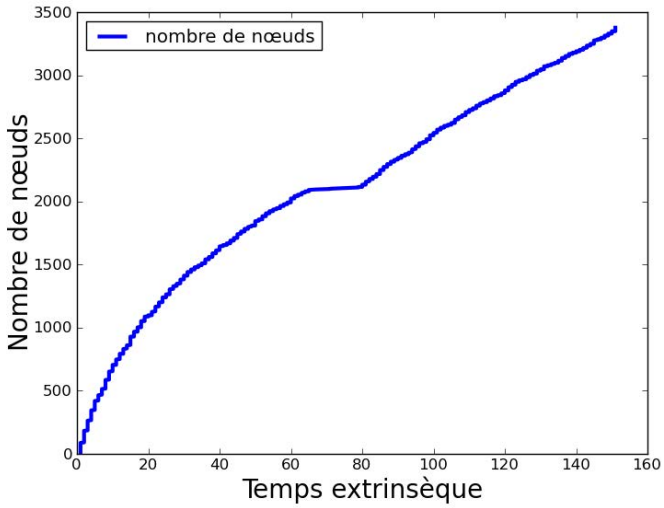



**Figure 2.9** – *Évolution du nombre de nœuds dans le réseau de blogs.*

plutôt régulière, sauf entre les jours 65 et 80 après le début de la mesure. Entre ces deux dates, le nombre de nœuds augmente extrêmement lentement, au point d'être presque constant. Ces dates correspondent à la période du 5 Avril au 20 Avril (2010), ce qui coïncide avec les vacances de Pâques de cette année-là. Nous pouvons observer une chute de l'activité de la blogosphère pendant ces vacances. À la fin des vacances, le nombre de nœuds reprend sa croissance au même rythme qu'avant.

Ce jeu de données est assez différent du jeu de données Github, car l'événement relié aux vacances de Pâques n'est pas du tout de la même ampleur que celui observé dans le graphe Github. Nous pourrons alors comparer les résultats obtenus entre ces deux jeux de données.

### 2.4.4 Jeu de données Infectious SocioPatterns

#### 2.4.4.1 Description du jeu de données

Ce jeu de données, étudié par [Isella et al., 2011], a été collecté pendant l'exposition Infectious SocioPatterns, qui s'est tenue en Irlande, entre Avril et Juillet 2011 (69 jours). Les visiteurs étaient équipés d'un capteur RFID, afin d'enregistrer les interactions entre les personnes. Le jeu de donnés ainsi obtenu forme donc un réseau de contacts : les nœuds sont les visiteurs et il y a un lien entre deux personnes s'ils sont restés face-à-face suffisamment longtemps (quelques secondes). Pour chaque lien, nous connaissons le moment de son apparition mais pas de sa disparition.

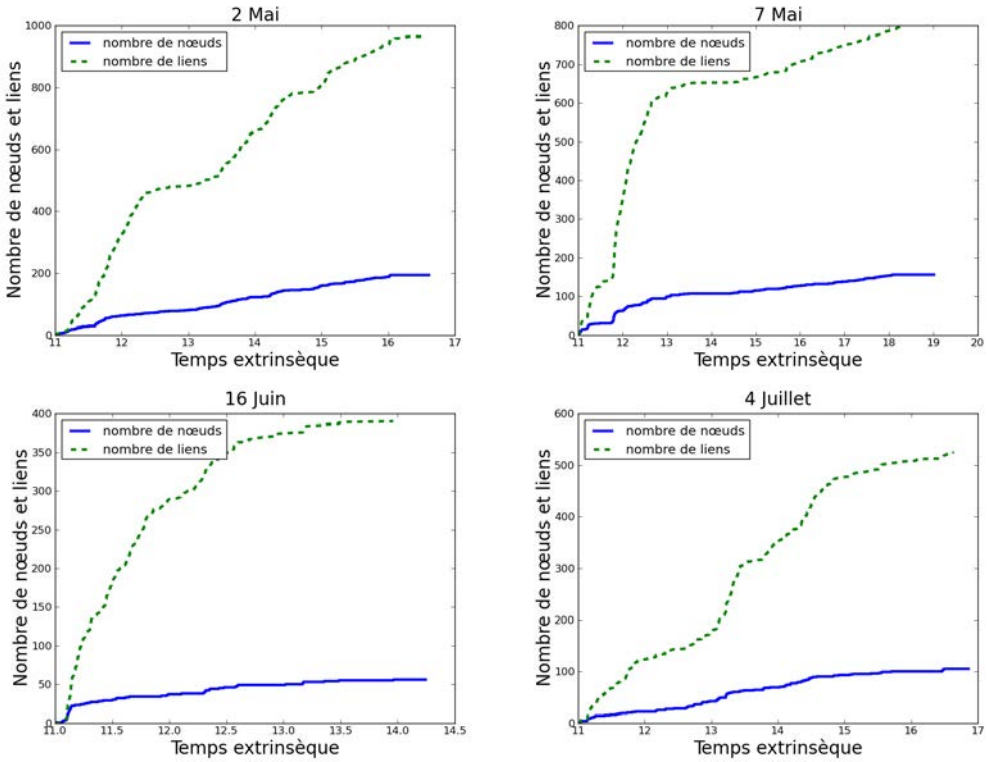



**Figure 2.10** – *Évolution du nombre de nœuds et de liens dans le réseau Infectious Socio Patterns pendant plusieurs journées.*

#### 2.4.4.2 Quelques statistiques

Sur ce réseau, nous avons choisi de montrer ici seulement 4 jours, qui mettent en évidence des comportements différents en terme de dynamique, et représentatifs de l'ensemble de la dynamique du réseau. La figure 2.10 montre le nombre cumulé de nœuds et de liens en fonction du temps pour ces quatre jours.

Nous observons sur cette figure que le nombre de nœuds et de liens augmente de façon assez irrégulière. Le 2 Mai, quand $t = 5000$, nous observons un ralentissement dans la croissance du nombre de nœuds. À l'instant $9000$, la croissance accélère de nouveau. Pour le 7 Mai, on observe le comportement inverse : au début, on a une croissance lente, et brusquement, le nombre de nœuds infectés augmente plus rapidement jusqu'à l'instant $6000$. Après cela, la croissance ralentit, jusqu'à la fin de la simulation. Le 4 Juillet, on observe quelques événements, aux instants $7000$ et $10000$ par exemple. Finalement, le 16 Juin, la croissance est plus régulière que les jours précédents, avec tout de même un ralentissement assez net vers la fin de la journée.

Nous voyons sur ces courbes que la dynamique et la diffusion sur ce réseau de contacts sont très différentes par rapport aux autres jeux de données, et même

d'un jour à l'autre. Certains jours, on observe plusieurs événements, alors que pour d'autres, on a une évolution régulière. Ce jeu de données va nous permettre de tester notre définition de temps intrinsèque sur un graphe avec une dynamique différente des deux précédents, et des événements beaucoup moins nets.

### 2.4.5 Jeu de données Infocom 2006

#### 2.4.5.1 Description du jeu de données

Ce jeu de données est un réseau de contacts mesuré pendant la conférence Infocom 2006 [5]. Certains participants se sont portés volontaires pour porter un équipement RFID. Quand deux personnes portant cet équipement ont été suffisamment proches, l'équipement RFID a enregistré ce contact comme un lien entre les deux personnes. Dans le jeu de données, les nœuds représentent les participants à l'expérience, et les liens représentent donc les contacts (proximité physique) entre les participants. Nous connaissons les instants (en seconde) d'apparition et de disparition de chaque lien.

L'expérience a durée pendant presque 4 jours (334 015 secondes), avec un total de 78 participants et 2953 liens différents sur le graphe agrégé, c'est-à-dire 2953 enregistrements de proximité entre participants. Si deux personnes ont été proches pendant une heure puis éloignées, et proches à nouveau plus tard, cela compte pour un seul lien dans le total. Étant donné le nombre de nœuds et de liens, on peut en déduire que pratiquement toutes les interactions potentielles entre participants ont été observées au moins une fois dans le graphe.

#### 2.4.5.2 Quelques statistiques

La figure 2.11 montre le nombre de liens en fonction du temps extrinsèque (agrégé) dans le réseau Infocom 2006. Nous voyons que ce réseau est très dynamique : on peut observer beaucoup de variations dans le nombre de liens. La mesure a démarré à la fin du premier jour de conférence, et on peut clairement observer juste après la première nuit (entre environ les instants 5000 et 10 000), puisqu'elle contient très peu de liens. Après la première nuit, nous observons une explosion du nombre de liens, qui correspond au deuxième jour (entre les instants 10 000 et 20 000) : pendant la deuxième journée, le nombre de liens varie considérablement, mais reste toujours très élevé, avec plus de trois contacts par individu en moyenne. L'augmentation soudaine pendant les phases de jour correspond aux pauses midi et aux pauses cafés. Ce schéma jour / nuit se répète sur tout le jeu de données : les phases de nuit sont très stables, avec peu de liens, tandis que les phases de jour sont beaucoup plus actives, avec beaucoup de variations dans le nombre de liens.

5. http://www.ieee-infocom.org/2006/

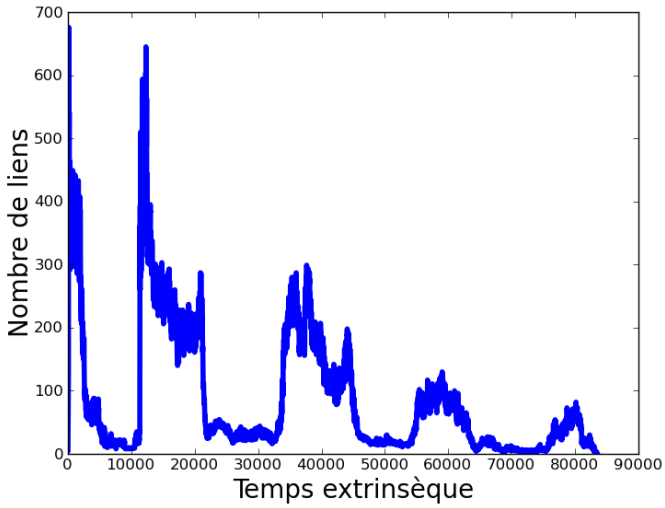



**Figure 2.11** – *Nombre de liens en fonction du temps extrinsèque (agrégé) dans le réseau Infocom 2006.*

De plus, le nombre de liens diminue progressivement pendant l'expérience : cette diminution est due au fait que certaines personnes sont parties avant la fin de la conférence. Nous utilisons ce jeu de données dans le chapitre 4.

## 2.5 Conclusion

Nous avons présenté dans ce chapitre les principaux travaux existants pour l'étude de graphes dynamiques. Nous avons vu que l'échelle de temps utilisée a une grande importance pour l'étude de la dynamique. Nous avons défini la notion de temps intrinsèque, et détaillé notre approche qui consiste à utiliser cette notion pour étudier des phénomènes de diffusion, et la structure en communautés d'un graphe dynamique. Nous avons également présenté l'ensemble des jeux de données que nous étudions dans cette thèse. Dans le chapitre suivant, nous appliquons la notion de temps intrinsèque à l'étude de phénomènes de diffusion.

Chapitre 3

# Diffusion et temps intrinsèque

## Sommaire



Les phénomènes de diffusion sur des graphes dynamiques sont complexes. En effet, la diffusion étant elle-même un processus dynamique, il est difficile de distinguer l'impact de la dynamique de la diffusion elle-même. Dans ce chapitre, nous définissons tout d'abord la notion de diffusion et présentons les travaux effectués dans ce domaine. Nous proposons par la suite notre méthodologie pour l'étude de la diffusion en temps intrinsèque, ainsi que nos résultats expérimentaux.

## 3.1 Diffusion : définition, modèles et état de l'art

### 3.1.1 Définition d'une diffusion

Un phénomène de diffusion consiste en la transmission d'un objet (virus, information, etc.) d'une entité à une autre : par exemple, la diffusion d'un virus consiste

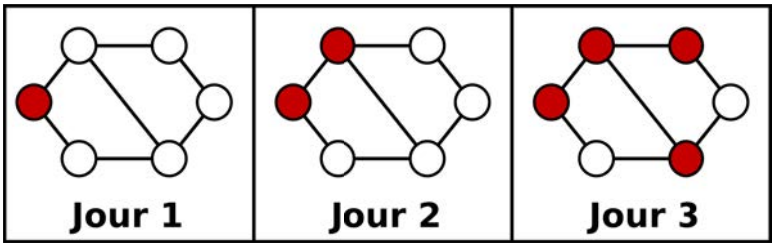



**Figure 3.1** – *Exemple de diffusion sur graphe statique.*

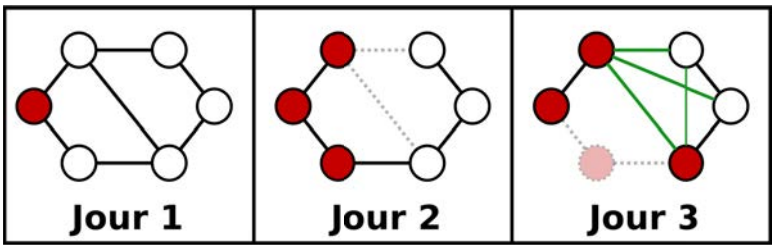



**Figure 3.2** – *Exemple de diffusion sur graphe dynamique.*

à transmettre d'une personne à une autre. Les exemples de diffusion sont nombreux et variés : diffusion de maladie, d'information (sur un réseau social par exemple) ou de rumeurs, diffusion de produits innovants, de virus informatique, de fichiers, etc. Dans tous les cas, l'objet qui se propage lors d'une diffusion circule d'un contact vers un autre, ce qui signifie que la diffusion se produit sur un graphe. Un lien entre deux nœuds signifie que ces deux nœuds sont en interaction, et que si l'un des deux est infecté, il peut contaminer l'autre. Dans les exemples cités ci-dessus, la diffusion se produit sur des types distincts de réseaux : réseau de contacts (entre personnes) pour une maladie, réseau social en ligne ou réseau de contacts physique pour une information, etc.

La figure 3.1 représente un exemple de diffusion sur un graphe statique. Le premier jour, un seul nœud possède l'objet à diffuser. Le deuxième jour, cet objet s'est diffusé vers un voisin du nœud de départ. Et au troisième jour, nous voyons que quatre nœuds possèdent l'objet diffusé.

Comme nous l'avons déjà expliqué dans le chapitre 2, les graphes de terrains sont le plus souvent dynamiques, ce qui signifie que la propagation d'un objet s'effectue sur un réseau en constante évolution. La figure 3.2 montre un exemple de diffusion sur un graphe dynamique. Au début, un seul nœud du graphe est contaminé. Le deuxième jour, deux nœuds supplémentaires (voisins du premier) sont contaminés. Nous observons également la disparition de deux liens par rapport au premier jour. Le troisième jour, un nouveau nœud est contaminé, et nous constatons également l'apparition de quatre nouveaux liens (en vert) et la disparition d'un nœud et deux liens.

Cette figure illustre l'impact que peut avoir l'évolution du graphe sur la diffusion. Le deuxième jour, le premier nœud a contaminé tous ses voisins. En revanche,

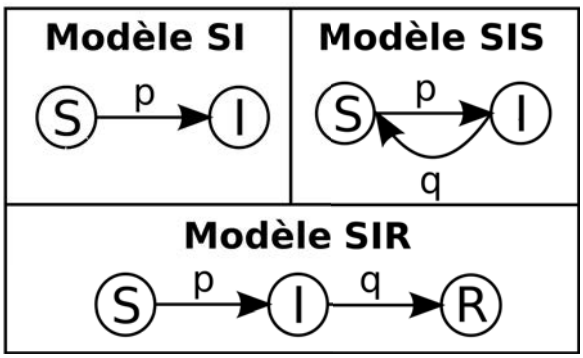



**Figure 3.3** – *Modèles de diffusion SI, SIS et SIR.*

avec la disparition des liens pendant le deuxième jour, deux des trois nœuds contaminés n'ont pas de voisin à contaminer à leur tour. La diffusion est donc ralentie, et seul le nœud du bas qui a un voisin sain risque de contaminer quelqu'un. Le fait que les nœuds et liens apparaissent et disparaissent au fil du temps a donc un impact très important sur les chemins que la diffusion emprunte, et donc sur la manière dont elle s'effectue.

### 3.1.2 Modèles de diffusion classiques

Afin d'étudier les phénomènes de diffusion et de mieux les comprendre, des travaux ont créé de longue date des modèles pour reproduire le comportement d'une diffusion et décrire de façon formelle comment s'effectue la propagation. Nous proposons ici de classer les modèles de diffusion de la littérature selon trois grandes catégories.

Le premier groupe, le plus ancien, est constitué des modèles épidémiologiques, c'est-à-dire des modèles qui décrivent la manière dont se comporte un virus ou une maladie, et comment il / elle se propage au sein d'une population. Les travaux de [Kermark and Mckendrick, 1927] et de [Bailey et al., 1975] sont les références dans ce domaine et détaillent les bases de la théorie de la diffusion épidémiologique. Le modèle épidémiologique le plus simple est le modèle SI (Susceptible - Infected). Dans ce modèle, les personnes d'une population sont dans l'un des états suivants : sain (état S) ou infecté (état I). Chaque personne a, à chaque instant, une probabilité $p$ d'être contaminée.

Ce modèle a rapidement été étendu : en effet, pour représenter une maladie, il n'est pas réaliste de considérer que les personnes infectées le restent indéfiniment. Il se peut en effet qu'elles meurent ou qu'elles guérissent. Dans le cas où elles guérissent, il y a deux possibilités : les personnes guéries peuvent être devenues immunisées à la maladie, ou bien avoir à nouveau le même risque que les autres d'être contaminées. On voit ainsi rapidement augmenter le nombre de modèles que l'on peut créer pour décrire toutes ces possibilités.

Hormis le modèle SI, les deux modèles épidémiologiques les plus simples sont les modèles SIR (Susceptible - Infected - Removed), et SIS (Susceptible - Infected - Susceptible). Le fonctionnement du modèle SIR est le même que celui du modèle SI, sauf que les personnes infectées ont une certaine probabilité à chaque instant de guérir et de passer à l'état R, c'est-à-dire qu'elles seront immunisées à la maladie, et ne pourront pas retourner à l'état I. Le fonctionnement du modèle SIS est également similaire à SI, sauf que les personnes infectées ont une probabilité à chaque instant de guérir et de revenir à l'état S (voir figure 3.3). Par la suite, ces modèles épidémiologiques ont été utilisés sur des graphes, où l'ensemble des nœuds représente la population, et les seules personnes qui interagissent sont les voisins, c'est-à-dire deux nœuds avec un lien entre les deux. Ainsi, dans le cas du modèle SI, chaque voisin dans l'état I d'un nœud dans l'état S a une probabilité $1/p$ de contaminer ce nœud.

La deuxième catégorie de modèles concerne l'adoption d'innovation. Les travaux qui font référence dans le domaine sont les travaux de [Jensen, 1982], de [Rogers, 1962] et de [Bass, 1969]. Les modèles décrits sont construits à partir d'observations empiriques pour décrire la propagation d'une innovation sur le marché. Dans ces modèles, la probabilité qu'un acheteur potentiel achète le produit à l'instant $t$ augmente de façon linéaire avec le nombre cumulé d'acheteurs. Ainsi, le nombre de personnes qui adoptent une innovation dépend de deux facteurs : le nombre de personnes possédant déjà l'innovation, ainsi que la taille du marché potentiel, qui introduit un phénomène de saturation. Ces modèles ont par la suite été étoffés et utilisés de nombreuses manières, en prenant par exemple en compte un découpage des tranches de personnes selon plusieurs critères : personnes avides de technologie (et donc rapides à adopter une innovation), personnes sceptiques, etc. [Valente, 1995].

Enfin, la troisième catégorie de modèles de diffusion cherche à caractériser la propagation d'influence au sein d'un réseau social. Les travaux de référence en la matière sont les travaux de [Kempe et al., 2003], qui introduisent les modèles LT (Linear Threshold) et IC (Independant Cascade). Dans le modèle LT, un nœud est influencé par ses voisins qui possèdent chacun un poids. Chaque nœud possède de plus un seuil (déterminé au hasard). Afin qu'un nœud devienne "actif", il faut que la somme des poids de ses voisins actifs soit supérieure à son seuil. Le modèle IC est une variante du modèle LT : en effet dans le modèle IC, les nœuds qui deviennent actifs ont une seule chance de rendre chacun de leurs voisins actifs. Ces modèles ont par la suite été généralisés dans [Saito et al., 2010], avec les modèles AsIC et AsLT, pour y introduire un délai asynchrone : quand un nœud devient "actif", il attend un délai (aléatoire) avant de tenter d'activer ses voisins.

Les modèles de ces trois catégories, bien que créés pour des objectifs et des domaines différents, sont utilisés dans d'autres contextes que ceux prévus au départ. Par exemple, les modèles épidémiologiques ont fréquemment été utilisés pour étudier la diffusion d'information [Karsai et al., 2011].

### 3.1.3 Travaux existants sur la diffusion

L'étude des processus de diffusion a commencé en même temps que l'apparition des premiers modèles pour les représenter [Kermark and Mckendrick, 1927]. Dans ces études, on a observé une population, sans plus de détail sur les interactions au sein de cette population. Plus tard, cette population a été décrite de façon plus détaillée, avec des interactions uniquement entre certaines personnes, constituant ainsi un graphe. Dans la suite, nous décrivons uniquement des travaux sur la diffusion se produisant sur un graphe.

#### 3.1.3.1 Diffusion et graphes statiques

L'ensemble des travaux étudiant la diffusion sur des graphes considère ce processus comme dynamique. En revanche, la plupart d'entre eux considèrent le graphe sur lequel la diffusion se produit de façon statique, alors qu'il est le plus souvent dynamique. Dans cette section, nous présentons les travaux analysant la diffusion sur un graphe statique, et dans la section suivante, ceux basés sur un graphe dynamique. Les objectifs des travaux sur la diffusion sont assez variés : nous en distinguons ici trois grands types.

Tout d'abord, une partie des travaux étudie le processus de diffusion afin de comprendre son fonctionnement, aussi bien sur des graphes théoriques aléatoires que sur des exemples réels. Les travaux de [Newman, 2002], [Volz, 2008], [Pastor-Satorras and Vespignani, 2001], [Keeling and Eames, 2005], [Barthelemy et al., 2005] et de [Girvan et al., 2002] étudient le comportement théorique d'une diffusion épidémiologique avec des modèles classiques de diffusion comme SI ou SIR sur des réseaux aléatoires. Selon la distribution de degré, homogène ou hétérogène, de ces réseaux, ils observent la présence d'un seuil épidémique, c'est-à-dire d'un nombre de nœuds infectés à partir duquel l'ensemble du réseau est contaminé à coup sûr.

Dans [Myers et al., 2012], les auteurs étudient les effets des influences extérieures au graphe sur la diffusion. Dans les modèles d'adoption classiques, l'information ne peut se propager que via des membres internes au réseau. Or, sur un réseau social par exemple, l'information peut provenir de l'extérieur, typiquement d'un journal ou de la télévision. Ces travaux proposent donc un paramètre représentant cette influence extérieure. Afin de fixer la valeur de ce paramètre, les auteurs utilisent des données issues de Twitter et regardent les "sauts" d'une information sur le réseau.

Il est également possible d'utiliser d'autres approches pour l'étude d'un phénomène de propagation : par exemple, [Goldenberg et al., 2001] utilisent des automates cellulaires pour générer un réseau complexe. Dans ce réseau, les nœuds ont deux types d'interactions possibles : un lien fort s'ils appartiennent au même groupe, et un lien faible sinon. De plus, chaque nœud est dans l'un des deux états suivants : informé ou non informé. Avec ce réseau, les auteurs montrent que les liens faibles

ont un effet au moins aussi important que celui des liens forts pour la transmission d'information. De plus, ils montrent que l'influence extérieure (comme la publicité) accélère la propagation d'information, surtout si elle se produit en début de diffusion.

Cette étude du comportement de la diffusion est aussi présente d'un point de vue expérimental sur des graphes "réels" considérés de façon statique : les travaux de [González-Bailón et al., 2011] étudient la façon dont la diffusion d'information sur le réseau social Twitter a influencé le nombre de participants au printemps arabe. Les auteurs observent ce réseau sur une période d'un mois, et cherchent à déterminer le rôle des premiers participants, ainsi que les utilisateurs qui ont agi en tant que source de diffusion. Ils constatent que les premiers participants (c'est-à-dire les premiers nœuds contaminés par les nœuds sources) n'ont pas de propriété topologique particulière dans le réseau, mais qu'en revanche, les nœuds sources sont toujours centraux.

Dans [Cha et al., 2010], les auteurs analysent l'influence des utilisateurs de Twitter. Ils définissent l'influence d'un utilisateur de trois façons différentes : son degré entrant, le nombre de "retweets" de cette personne (c'est-à-dire le nombre de fois où d'autres personnes citent ses tweets), et le nombre de mentions (c'est-à-dire le nombre de fois ou d'autres personnes mentionnent son nom). Les auteurs utilisent ces mesures sur plusieurs utilisateurs, et sur plusieurs sujets. Ils montrent que si le degré entrant d'un utilisateur est élevé, cela n'implique pas du tout que cet utilisateur aura de nombreux retweets et mentions. Les retweets nombreux sont associés en général à des nouvelles, alors que les mentions sont associées à des célébrités. Les auteurs montrent dans ces travaux que ce ne sont pas forcément les personnes avec le plus de "followers" qui ont le maximum d'influence, et que les utilisateurs qui ont le plus de retweets sont ceux qui contribuent à un seul sujet.

Les blogs sont également très étudiés dans la diffusion d'information : [Gruhl et al., 2004], [Salah Brahim et al., 2011] et [Adar and Adamic, 2005] étudient la propagation d'information dans la blogosphère. Ils cherchent à déterminer des thématiques pour les blogs, ainsi que des blogs et des billets [1] (ou posts en anglais) qui sont particulièrement influents (en mesurant le nombre de citations de ces posts). Ils observent la façon dont les blogs sont cités, à l'aide de l'étude de cascades de diffusion. Dans [Cheng et al., 2014], les auteurs analysent des cascades de diffusion sur le réseau social Facebook, afin de pouvoir prédire le comportement de ces cascades.

L'un des objectifs de l'étude de la diffusion consiste à modifier son comportement pour satisfaire un critère donné. L'exemple le plus classique est de vouloir, dans le cas épidémiologique, ralentir la diffusion (afin de ralentir et d'arrêter une épidémie). On peut notamment citer [Karkada et al., 2011] qui tentent de limiter la

1. Un blog est constitué d'une suite d'articles, qui sont datés et s'affichent à l'écran en commençant par le plus récent. Ces articles s'appellent des billets.

propagation d'une maladie en étudiant les transferts hospitaliers sur une durée de deux ans. Sur ce réseau, ils simulent la diffusion d'une maladie en choisissant un hôpital de départ au hasard, et observent comment elle évolue en utilisant diverses stratégies pour restreindre cette épidémie. Ils trouvent qu'il est beaucoup plus efficace de fournir plus de ressources à quelques hôpitaux centraux pour restreindre l'épidémie que de répartir ces ressources sur l'ensemble des hôpitaux. [Madar et al., 2004] s'intéressent à la même problématique, mais étudient un réseau théorique aléatoire. Ils testent plusieurs stratégies d'immunisation à la maladie, en utilisant le modèle SIR pour la diffusion. Ils proposent une stratégie pour limiter une épidémie, sans toutefois la tester sur un jeu de données réel.

L'objectif de l'étude de la diffusion peut être au contraire de vouloir maximiser la diffusion, par exemple pour diffuser une information ou un produit. [Mahajan et al., 1990] résument différentes approches étudiant l'impact de la publicité sur la diffusion d'un produit. Si les utilisateurs sont trop exposés à des publicités, ils finissent par ne plus en tenir compte du tout. Les méthodes d'optimisation de diffusion de produits cherchent donc toujours à cibler quelques personnes et envoyer peu de publicité afin de maximiser leur efficacité plutôt que d'inonder les utilisateurs d'information.

Enfin, d'autres travaux étudient le rôle de la topologie du graphe dans le processus de diffusion. En effet, une diffusion peut changer radicalement de comportement selon la structure du réseau sous-jacent. Cette approche est complémentaire à celle de l'étude des modèles de diffusion. Si l'on connait le comportement d'un modèle sur un type de réseau particulier, alors on peut complètement prédire le déroulement d'une diffusion. Dans le cas d'une épidémie, cela pourrait permettre de contenir la maladie en prenant les mesures nécessaires (en vaccinant ou en isolant les bonnes personnes par exemple). [Eames et al., 2012] étudient un jeu de données de personnes dans une école, dont ils savent s'ils elles ont eu la grippe H1N1v, et dont ils connaissent les contacts chaque jour. Leur étude montre que l'épidémie ralentit considérablement pendant les vacances scolaires, car le graphe des contacts devient beaucoup moins dense. Ils montrent de plus que le fait de modifier les contacts entre les personnes est un facteur d'accélération pour la diffusion de la maladie.

Dans d'autres domaines, l'effet de la topologie du réseau sur la diffusion est intéressant aussi, comme pour la diffusion d'innovation par exemple. [Delre et al., 2010] regardent, pour des réseaux à distribution de degrés hétérogène, le rôle des nœuds centraux dans la diffusion d'innovation. Pour cela, ils considèrent un modèle de diffusion dans lequel chaque nœud a des préférences, c'est-à-dire une plus ou moins grande tendance à adopter une innovation. Cette notion de préférence est liée à celle d'influence des nœuds. Les auteurs montrent que la diffusion est ralentie par la présence de nœuds influents, par rapport à un réseau avec une distribution de degrés homogènes.

[Bakshy et al., 2012] étudient également le rôle de la topologie du réseau sur la diffusion d'information, en s'intéressant au cas d'un réseau social (Facebook). Ils observent, sur plusieurs centaines de millions d'utilisateurs, le degré d'exposition à différentes informations. Sans surprise, les nœuds les plus exposés sont les plus susceptibles de diffuser à leur tour l'information, dans un délai très court. De plus, le fait qu'un nœud ait beaucoup de ses voisins "contaminés" augmente nettement sa probabilité d'être atteint à son tour.

[Kitsak et al., 2010] proposent une méthode pour détecter les personnes importantes dans la diffusion en utilisant la décomposition du réseau en "k-shells". Ces k-shells déterminent, pour chaque nœud du réseau, à quel point il est bien connecté, et à quel point ses voisins sont bien connectés. Ils montrent que, quel que soit le graphe considéré, les nœuds les plus influents dans la diffusion (superspreaders) sont ceux qui ont un indice de k-shell très élevé. Pour cela, ils choisissent des noeuds sources de la diffusion appartenant à des k-shells aux indices différents, et observent la vitesse de diffusion et le nombre de nœuds atteints dans chaque cas.

#### 3.1.3.2 Diffusion et graphes dynamiques

Si l'étude de la diffusion sur des graphes statiques est un domaine étudié de longue date, en revanche l'étude de la diffusion sur des graphes dynamiques est très récente. L'état de l'art de [Masuda and Holme, 2013] fait le point sur les travaux qui ont été menés jusqu'à présent concernant les diffusions épidémiques sur les réseaux temporels. Parmi ces travaux, on peut distinguer trois catégories d'études.

**Diffusion sur un graphe dynamique aléatoire :** Tout d'abord, certains travaux s'intéressent à l'étude de la diffusion sur des graphes dynamiques via des études théoriques. [Volz and Meyers, 2009] étudient le modèle SIR sur des graphes aléatoires dynamiques, et compare les résultats obtenus avec ceux connus sur des graphes statiques. Ils montrent que le seuil à partir duquel une diffusion peut devenir épidémique dépend de plus de paramètres que dans le cas d'un graphe statique. En effet, pour un graphe statique, ce seuil est lié aux taux de transmission et de guérison. Dans le cas d'un graphe dynamique, la distribution de degrés ainsi que les taux de contacts entre les différents nœuds entrent en jeu également. [Rodriguez et al., 2011] étudient la modélisation de la diffusion sur des réseaux aléatoires dynamiques, au travers des cascades (diffusion de nœud en nœud). En connaissant le comportement de la diffusion, ils cherchent à inférer la dynamique du réseau sous-jacent, et proposent un algorithme. Ils testent ensuite cet algorithme sur des jeux de données où la diffusion et le graphe sont connus, et observent le taux de réussite, qui varie entre 60% et 95% de prédictions correctes.

**Diffusion sur un graphe dynamique réel :** Une autre branche de l'étude de la diffusion sur des graphes dynamiques s'intéresse à la simulation de modèles épidémiologiques sur des réseaux dynamiques réels. Dans [Stehlé et al., 2011], les auteurs étudient la simulation du modèle épidémiologique SEIR sur un réseau de contacts extrait d'une mesure réalisée sur les participants à une conférence. Dans cette étude, en plus d'analyser le comportement de ce modèle et l'impact de la dynamique du graphe sur la diffusion, les auteurs proposent une méthode pour prolonger artificiellement la durée de mesure du graphe d'après son comportement observé pendant un certain laps de temps.

Les travaux de [Eames et al., 2012] s'intéressent à la diffusion du virus H1N1v, et l'étudient sur le réseau de contacts d'une école. Dans cette étude, on observe l'impact du changement de topologie du réseau sur la diffusion, et notamment, on peut observer que pendant les vacances, la contamination des enfants de l'école ralentit fortement.

[Takaguchi et al., 2013] étudient la vitesse de diffusion sur deux réseaux dynamiques réels : un réseau de contacts et un réseau d'échange d'e-mails. Ils choisissent un modèle où un nœud a besoin de plusieurs contacts avec un nœud infecté pour pouvoir être contaminé à son tour, le but de ce modèle étant de reproduire un comportement réaliste de diffusion d'information. Les auteurs montrent qu'avec ce modèle, l'hétérogénéité de la dynamique (rafales de liens et de nœuds apparaissant dans le graphe en un court intervalle de temps) accélère nettement la diffusion par rapport à un réseau statique, ou par rapport à une dynamique régulière.

[Vernon and Keeling, 2009] cherchent à comprendre le rôle de la dynamique du réseau. Ils étudient un réseau de bétail sur un an. Afin d'étudier l'impact de la dynamique du réseau sur la diffusion, ils simulent un modèle SIR sur le graphe dynamique, le graphe agrégé par petites fenêtres de temps, et le graphe statique. Leurs résultats montrent que la diffusion garde une forme sigmoïde dans tous les cas, mais on observe dans le cas dynamique des effets jours / nuits dans la vitesse de diffusion. De plus, les auteurs montrent que le comportement de la diffusion sur le graphe dynamique est impossible à reproduire sur le graphe statique même en ajustant la probabilité de contamination.

**Diffusion sur graphe dynamique réel et comparaison avec un graphe aléatoire :** Une autre approche pour étudier le problème de la diffusion sur des graphes dynamiques consiste à observer une simulation de diffusion sur un graphe dynamique réel, puis à modifier ce graphe réel pour rendre les temps d'inter-arrivée aléatoires. Cette méthode permet par exemple de faire disparaître les rafales d'arrivées de liens, et d'uniformiser la dynamique. Ensuite, on compare une diffusion simulée sur les deux graphes, et cela permet d'observer l'impact d'une propriété du graphe liée à la dynamique sur la diffusion. Ces graphes rendus aléatoires sont appelés "null model" dans la littérature. Les travaux de [Karsai et al., 2011]

détaillent cette méthodologie sur un jeu de données téléphoniques dynamique, et sur un jeu de données e-mail. Les auteurs exposent plusieurs façons de faire un "null model". En comparant leurs résultats, ils en déduisent que la dynamique réelle d'un graphe ralentit la diffusion (si l'on compare la diffusion observée à celle sur un "null model").

[Kivelä et al., 2012] confirment ce résultat en étudiant un réseau dynamique de téléphonie mobile : ils simulent un modèle SI (avec une probabilité de contamination égale à 1) sur ce réseau, ainsi que sur six "null model" différents. Parmi ces "null model", une partie conserve les corrélations temporelles, et une partie conserve les corrélations topologiques. Les auteurs montrent que la diffusion est plus lente lorsque la dynamique est irrégulière. Leur étude permet de comprendre l'impact de la topologie et de la dynamique sur la diffusion.

[Merler and Ajelli, 2009], en appliquant la méthodologie décrite précédemment, trouvent quant à eux que la dynamique réelle accélère la diffusion dans le cas de réseaux de transport. Selon la dynamique du graphe, on peut donc changer radicalement le comportement de la diffusion. Les travaux de [Tabourier et al., 2012] utilisent le même principe de "null model" sur un réseau téléphonique dynamique : ils cherchent à déterminer la corrélation entre des événements, c'est-à-dire un appel qui en déclenche un autre. L'étude de ces corrélations se traduit donc par l'étude des cascades. Les auteurs comparent leurs résultats obtenus sur le jeu de données réels à ceux obtenus sur un "null model", afin de déterminer la corrélation entre deux appels. Ils trouvent au final peu de cascades dans ce jeu de données.

Enfin, [Gauvin et al., 2013] comparent une diffusion avec un modèle SI sur un réseau de contacts avec un "null model". Les auteurs définissent une unité de temps indépendante pour chaque nœud. Dans le graphe, chaque nœud possède une horloge, et enregistre le temps passé en contact avec des nœuds (on distingue les nœuds infectés et les nœuds sains) et le temps écoulé depuis le début de la contamination. Ces travaux montrent que l'hétérogénéité de la dynamique n'est pas le seul paramètre à étudier pour comprendre le comportement d'une diffusion : les hétérogénéités au niveau des contacts jouent également un rôle très important.

La diffusion sur des graphes dynamiques est étudiée également dans d'autres contextes, par exemple, dans le cas du routage opportuniste sur un réseau. Les travaux de [Fall, 2003] sont fondateurs dans le domaine, et expliquent comment on peut exploiter la dynamique du graphe pour faire transiter un paquet le plus rapidement possible. Cette étude constitue un domaine de recherche à part entière actuellement, on peut par exemple citer les travaux de [Yoneki et al., 2008] ou de [Whitbeck et al., 2011].

### 3.1.4 Notre approche

Dans nos travaux, nous nous sommes intéressés à l'impact de la dynamique d'un graphe sur une diffusion. Nous ne cherchons pas à accélérer ou ralentir une

diffusion, ni à créer de nouveaux modèles. Nous voulons comprendre le rôle d'une modification de topologie dans le graphe sur la diffusion. Toutefois, nous avons choisi une approche différente de celle du "null model", décrite ci-dessus : en effet, la façon de générer le null model modifie la dynamique du graphe pour en faire une dynamique "lisse". On peut donc comparer la perte d'information entre la dynamique réelle et la dynamique aléatoire. Dans nos travaux, nous étudions l'impact de la dynamique sur la diffusion en travaillant sur l'échelle de temps. Selon l'échelle choisie, nous verrons que l'on peut également "lisser" la dynamique, mais que nous obtenons aussi d'autres informations sur la dynamique, comme par exemple l'impact d'un événement particulier dans les changements de topologie. Une fois que l'on comprend en détail le rôle de la dynamique sur la diffusion, le fait d'utiliser la dynamique pour accélérer ou ralentir la diffusion devient plus simple. De la même façon, le changement d'échelle de temps permettra aussi de mieux appréhender l'impact d'un événement comme une rafale sur la diffusion.

Le but de ce chapitre est de proposer une méthodologie afin de comprendre, lorsque l'on observe un processus de diffusion, ce qui est intrinsèquement lié au processus de diffusion, et ce qui est dû uniquement à l'évolution du graphe. Bien évidemment, ce n'est pas possible de complètement dissocier les deux phénomènes car ils sont très fortement liés, mais l'on peut en revanche normaliser nos observations. Par exemple, si à un moment dans la diffusion, on observe une accélération soudaine, on peut regarder si cette accélération correspond à un changement de topologie radical dans le graphe. Dans cette thèse, nous proposons pour étudier l'impact de la dynamique du graphe sur la diffusion, une approche basée sur l'observation de ce phénomène à différentes échelles de temps.

Lorsque que l'on étudie un processus de diffusion, on l'observe spontanément selon une échelle de temps absolue (par exemple, la seconde, ou le jour) que nous qualifions d'extrinsèque. Ici, nous proposons d'observer ce même phénomène mais selon une autre échelle de temps que nous appelons intrinsèque, liée directement à l'évolution du graphe.

## 3.2 Méthodologie

### 3.2.1 Modèle de diffusion utilisé

Nous avons choisi un modèle de diffusion très simple pour nos simulations : le modèle Susceptible - Infected (SI). Ce modèle, détaillé dans la section 3.1.2, est largement utilisé depuis sa création en 1927. Il est particulièrement intéressant dans notre cas, car il requiert très peu de paramètres, à savoir, la probabilité de contamination $p$ et le choix des nœuds infectés au départ (c'est-à-dire ceux par qui démarre la contamination). Dans la suite, les valeurs de ces deux paramètres seront fixées, afin de se concentrer sur la corrélation entre la dynamique du graphe et le

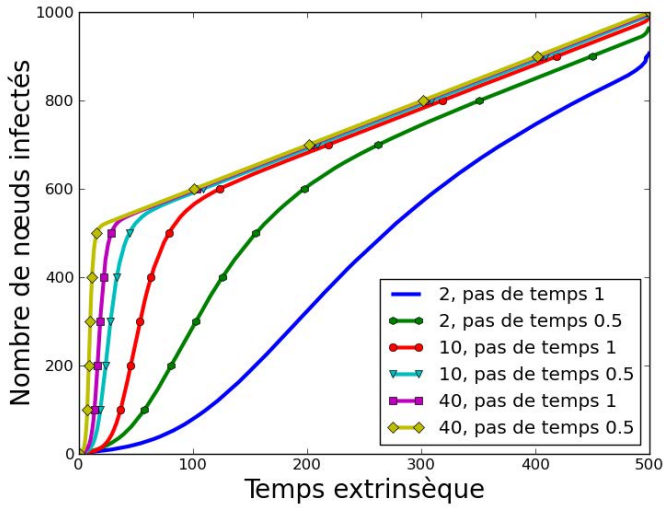



**Figure 3.4** – *Diffusion avec un modèle SI sur graphe BA dynamique évoluant de 500 à 1000 nœuds, pour une probabilité de contamination* $p = 1/500$. *Le paramètre* $m$, *respectivement égal à 2, 10 et 40, correspond au nombre de connexions établies par chaque nouveau nœud dans le graphe. Les simulations sont effectuées avec deux valeurs de pas de temps :* $timestep = 0,5$ *signifie qu'il y a deux tests de contaminations par unité de temps, contre un seul quand* $timestep = 1$. *Ces courbes correspondent à la moyenne des résultats sur 500 simulations.*

processus de diffusion. Nous avons fixé la probabilité d'être infecté $p = 0.005$ : avec cette valeur de $p$, une partie importante des nœuds est infectée, mais il reste malgré tout suffisamment de nœuds sains pour éviter un phénomène de saturation. Quelle que soit l'unité de temps considérée, nous réalisons un test de contamination du modèle SI par unité de temps. Il sera par conséquent plus facile par la suite de distinguer les observations directement liées au modèle de celles liés à la topologie du graphe ou de la notion de temps utilisée.

Pour illustrer cette notion, la figure 3.4 montre le comportement du modèle SI sur un graphe Barabási-Albert (décrit dans la section 2.4.1) quand le paramètre d'attachement préférentiel $m$ varie et avec différentes valeurs de pas de temps, et confirme des résultats connus. Dans tous les cas, le nombre de nœuds infectés continue d'augmenter avec le modèle SI, car les nœuds infectés le restent définitivement. Nous pouvons observer pour la plupart des courbes une croissance initiale rapide au début du processus de diffusion. À partir d'un certain point, le nombre de nœuds infectés augmente plus lentement, puis au final, la croissance observée est liée uniquement à la taille croissante du graphe, et donc à l'arrivée de nouveaux nœuds à contaminer. Comme prévu, la valeur de $m$ a un impact sur le comportement de la diffusion : cette dernière est beaucoup plus rapide avec $m = 40$ qu'avec $m = 2$. Quand $m = 2$, le point de saturation n'est pas atteint durant la simulation.

La figure 3.4 montre aussi l'importance de la valeur du pas de temps. En effet,

diviser le pas de temps par deux accélère la diffusion. En revanche, la diffusion n'est pas deux fois plus rapide.

### 3.2.2 Analyse de la diffusion en temps intrinsèque

Dans la suite, nous représenterons et comparerons pour chaque simulation de diffusion le nombre total de nœuds infectés en fonction de trois échelles de temps distinctes :

- **temps extrinsèque** qui correspond à la notion usuelle de temps.
- **temps extrinsèque converti en temps intrinsèque**, qui résulte de la conversion du temps extrinsèque en temps intrinsèque pour observer la diffusion, comme montré sur la figure 2.6. Notons que la diffusion observée ici est la même dans les deux cas.
- **temps intrinsèque** (défini dans la section 2.3), c'est-à-dire que nous représentons la vitesse de contamination en fonction du nombre de liens créés. Cette courbe correspond à un processus de diffusion différent des précédents, simulé directement en temps intrinsèque.

### 3.2.3 Exemple sur des graphes synthétiques Barabási-Albert

Afin d'illustrer notre méthodologie, nous étudions à présent une diffusion sur un graphe BA en utilisant les notions de temps extrinsèque et intrinsèque.

#### 3.2.3.1 Simulation en temps extrinsèque

Nous étudions d'abord une simulation de diffusion avec un modèle SI, sur les trois graphes BA générés. Les résultats sont montrés sur la figure 3.5 (en haut à gauche). Cette figure montre des comportements de diffusion très différents, et la variation des temps inter-arrivée modifie beaucoup le comportement de la diffusion. Ce résultat est naturel : en effet, il est très probable qu'à chaque génération du temps inter-arrivée, on obtienne un résultat supérieur à $1$. En temps extrinsèque, si nous générons par exemple un délai de $10$ unités de temps entre l'arrivée de deux nœuds, nous effectuons dix tests de contaminations en un seul pas de temps intrinsèque.

#### 3.2.3.2 Diffusion représentée en temps extrinsèque converti

Afin de comparer les notions de temps extrinsèque et intrinsèque et de mieux comprendre le comportement de diffusion en temps extrinsèque, nous convertissons en temps intrinsèque les résultats précédents obtenus en temps extrinsèque. Rappelons que si le second lien est créé à l'instant extrinsèque $15$, son temps

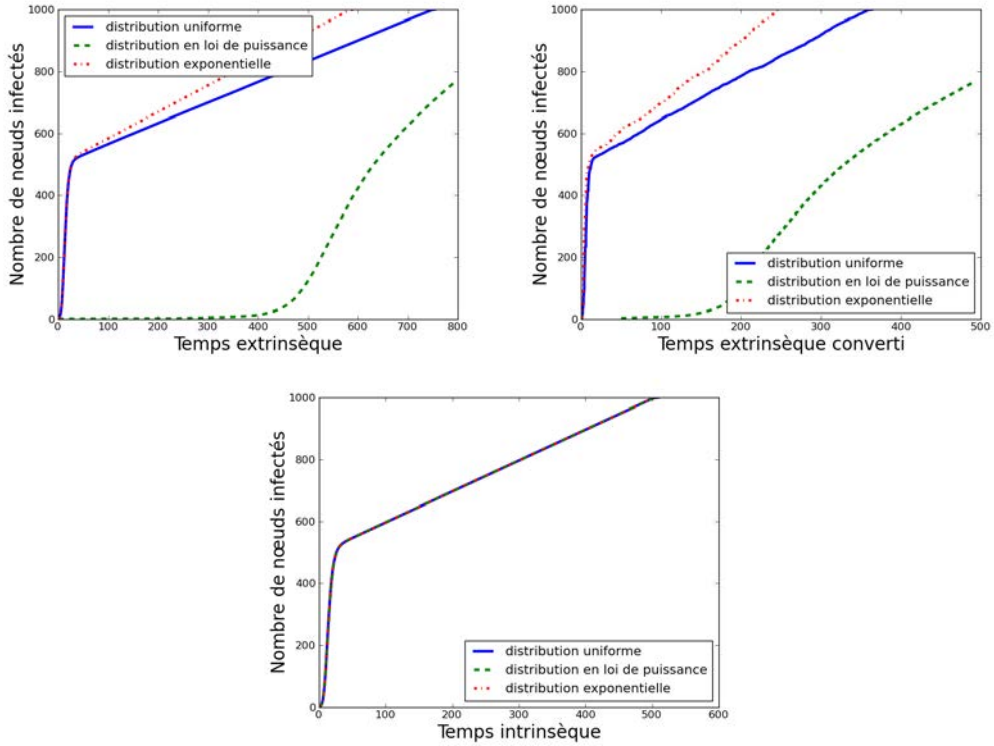



**Figure 3.5** – *Diffusion avec un modèle SI sur un graphe BA, de 500 à 1000 nœuds. La durée entre la création de deux nœuds suit trois lois de probabilité distinctes : uniforme, exponentielle, et loi de puissance.* ***En haut à gauche*** *: Diffusion simulée et observée sur une échelle de temps extrinsèque.* ***En haut à droite*** *: Diffusion simulée en temps extrinsèque et observée en temps intrinsèque.* ***En bas*** *: Diffusion simulée et observée en temps intrinsèque.*

intrinsèque est de 2. Dans ce cas, nous traçons par conséquent le nombre de nœuds infectés à l'instant extrinsèque 15, qui correspond au même nombre de nœuds infectés à l'instant intrinsèque 2. On réalise donc une mise à l'échelle non uniforme de la figure 3.5 (en haut à gauche) pour obtenir la figure 3.5 (en haut à droite).

Nous voyons que la conversion du temps extrinsèque en temps intrinsèque mène à des observations semblables à celles obtenues en temps extrinsèque (voir figure 3.5 en haut à gauche). La contamination démarre vite au début, jusqu'au moment où l'on observe un ralentissement important (sauf pour la distribution en loi de puissance). Ce ralentissement commence au moment où tous les nœuds du graphe sont contaminés. Il ne reste alors à contaminer que les nœuds arrivant dans le graphe au fur et à mesure. Ceci explique que toutes les courbes ont une allure linéaire par la suite. Cette observation est liée à la dynamique du réseau et au modèle SI. En effet, au début, beaucoup de nœuds sont susceptibles d'être contaminés, ce qui explique la phase de croissance rapide. Cette phase peut être observée en temps extrinsèque sur la figure 3.5 (en haut à gauche). Après cette phase, la plupart des nœuds sont contaminés, et l'infection se répand au rythme

de création des nouveaux nœuds, qui est de un par unité de temps intrinsèque.

Les trois courbes converties changent très peu pendant la conversion. En effet, le temps extrinsèque converti étire les rafales de création de liens, et raccourcit les phases avec peu d'activité. Comme nous étudions ici un graphe aléatoire sans rafale ni phase stable, il est par conséquent normal d'observer un comportement similaire entre les courbes en haut à gauche et en haut à droite de la figure 3.5.

#### 3.2.3.3 Simulations en temps intrinsèque

Le but ici est de montrer qu'un modèle SI simulé en fonction du temps intrinsèque est en quelque sorte imperméable à la dynamique. Nous simulons des diffusions avec un modèle SI sur les trois graphes BA décrits précédemment. La figure 3.5 (en bas) montre les résultats de ces simulations. Nous observons sur cette figure que ces trois courbes ont des comportements quasiment identiques. En effet, la notion de temps intrinsèque reflète seulement la modification de la topologie et pas le délai entre ces modifications. Par conséquent les différentes dynamiques des temps d'inter-arrivée n'ont aucun impact quand on observe la diffusion en temps intrinsèque. Cette courbe est très différente à la fois du temps extrinsèque et du temps extrinsèque converti (voir figures 3.5 en haut), ce qui confirme l'intérêt d'étudier la diffusion en utilisant l'échelle de temps intrinsèque.

Cet exemple montre l'impact très important de la dynamique du graphe sur le comportement de la diffusion. Dans la suite, nous réalisons les mêmes expériences sur les jeux de données décrits dans la section 2.4.

## 3.3 Résultats expérimentaux

Cette section est dédiée aux résultats de notre analyse sur la diffusion en temps intrinsèque, appliquée respectivement aux réseaux Github, blogs et Infectious SocioPatterns décrits dans les sections 2.4.2, 2.4.3 et 2.4.4 respectivement. Sur chacun de ces graphes, nous observons le nombre cumulé de nœuds contaminés en fonction du temps extrinsèque, du temps extrinsèque converti en intrinsèque, et du temps intrinsèque.

### 3.3.1 Résultats sur le réseau Github

#### 3.3.1.1 Simulation en temps extrinsèque

Nous commençons l'étude de notre diffusion en utilisant le temps extrinsèque, ici en seconde. Pour la simulation, nous faisons un test de contamination toutes les 30 secondes. En effet, si l'on effectue un test plus souvent, cela alourdit beaucoup

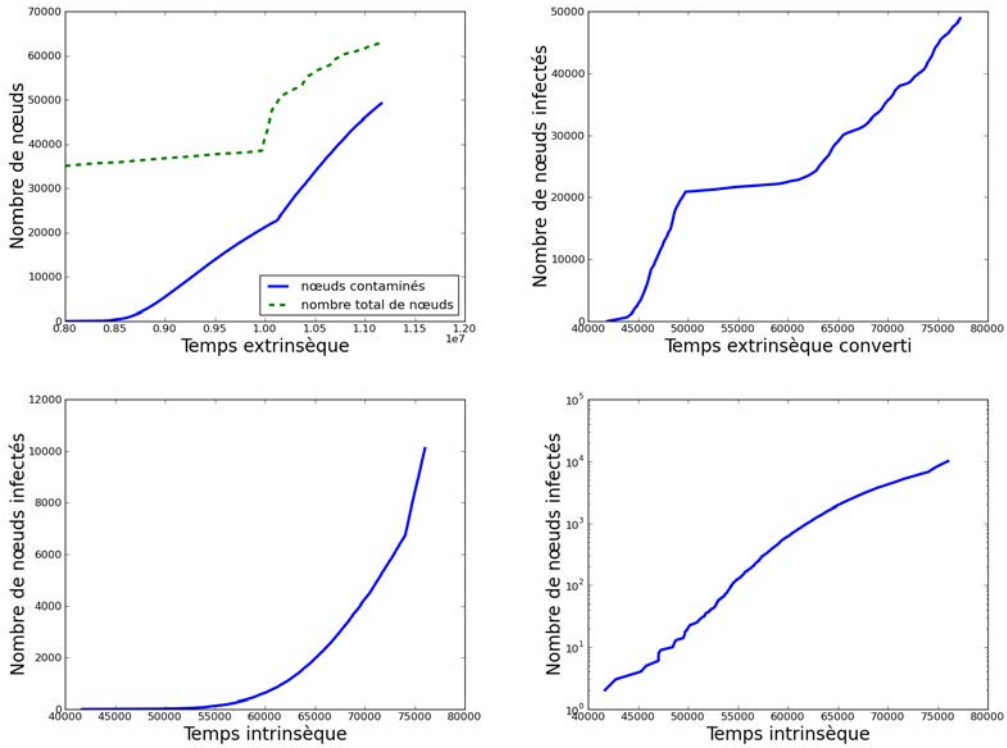



**Figure 3.6 –** ***En haut à gauche*** *: en bleu, diffusion avec un modèle SI sur le réseau Github, simulé et observé en temps intrinsèque. L'espace entre deux pas de temps est de 30 secondes. En vert : nombre de nœuds dans le graphe à chaque pas de temps.* ***En haut à droite*** *: diffusion avec un modèle SI sur le réseau Github, simulé en temps extrinsèque, et observé en temps extrinsèque converti en temps intrinsèque.* ***En bas*** *: diffusion avec un modèle SI sur le réseau Github, simulé et observé en temps intrinsèque, avec une échelle linéaire (à gauche), et avec une échelle logarithmique sur les ordonnées (à droite).*

le calcul, et n'apporte pas beaucoup plus d'information. En moyenne, il y a un nouveau lien toutes les 91 secondes dans le réseau Github. Ainsi, il y aura moins de pas de temps en intrinsèque qu'en temps extrinsèque, puisque chaque création de lien correspond à un pas de temps intrinsèque.

Les résultats sont montrés dans la figure 3.6 (en haut à gauche). Dans le but de faciliter l'interprétation, nous avons ajouté le nombre de nœuds au cours du temps. Nous remarquons que le processus de diffusion démarre lentement, puis accélère rapidement, ce qui correspond à la forme classique du modèle SI, exponentielle en début de diffusion. Il est très facile de voir sur cette courbe l'impact de la création du tutoriel *Try-Git* le 4 Juillet 2012 : le nombre de nœuds dans le réseau augmente brusquement plus vite, et simultanément, la diffusion accélère elle aussi. Cette accélération est due à l'apparition d'un nœud de très haut degré, qui facilite la diffusion.

#### 3.3.1.2 Simulation en temps extrinsèque converti en intrinsèque

Nous réalisons à présent sur la diffusion montrée dans le paragraphe précédent la conversion du temps extrinsèque en temps intrinsèque. Les résultats de cette conversion sont montrés sur la figure 3.6 (en haut à droite). Nous observons un comportement vraiment différent par rapport à celui de la diffusion en temps extrinsèque (voir figure 3.6, en haut à gauche). À l'instant de temps intrinsèque $t = 50000$, qui correspond à l'événement du 4 Juillet 2012, la croissance du nombre de nœuds ralentit brusquement. On peut expliquer ce phénomène de la façon suivante : après l'événement, la création de liens s'accélère nettement, c'est-à-dire qu'il y a beaucoup plus de nouveaux liens par seconde qu'avant l'événement.

Le temps intrinsèque ne prend pas en compte ce phénomène, puisque les pas de temps ne dépendent pas de la vitesse de création des liens. En temps extrinsèque, on a donc beaucoup de créations de liens en un petit nombre de pas de temps, alors qu'en temps intrinsèque, on crée beaucoup de pas de temps qui correspondent à un instant (extrinsèque) court. Ainsi, quand on convertit la diffusion en temps intrinsèque, on observe naturellement un plateau, qui correspond à l'accélération de la création des liens.

Pour expliquer ce phénomène d'une autre façon, on peut observer le nombre de créations de liens au cours du temps (extrinsèque) sur la figure 2.8. Nous remarquons une importante accélération de la croissance à l'instant $1 \times 10^7$, et des accélérations plus légères aux instants $1.04 \times 10^7$ et $1.07 \times 10^7$. Chaque accélération de la croissance du nombre de liens correspond à un plateau dans le nombre de nœuds infectés, aux temps intrinsèques $50000$, $65000$ et $71000$ respectivement. L'ampleur de l'accélération est corrélée à la longueur du plateau en temps intrinsèque.

Ainsi, observer une diffusion simulée en temps extrinsèque avec une échelle de temps intrinsèque aide à repérer les moments importants dans l'évolution du graphe. Nous avons vu que la définition du temps intrinsèque (en fonction de l'apparition des liens), est pertinente pour l'étude du réseau Github : en effet, la création de liens permet de caractériser correctement les événements majeurs dans le graphe. Nous allons dans la section suivante simuler la diffusion en temps intrinsèque, afin de confirmer que cette définition est pertinente.

#### 3.3.1.3 Simulation en temps intrinsèque

La diffusion simulée à l'échelle de temps intrinsèque sur le réseau Github est représentée sur la figure 3.6 (en bas). Nous n'observons pas de changement radical dans la diffusion à l'instant de la création du tutoriel *Try-Git* (qui correspond à l'instant $50000$ en temps intrinsèque. En effet, la courbe présente un comportement de diffusion assez standard pour le modèle SI : nous observons une croissance exponentielle (confirmée par la courbe en échelle logarithmique en bas à droite de

la figure 3.6), avec un début lent, et ensuite une croissance du nombre de nœuds qui accélère rapidement. Nous n'observons pas ici de phénomène de saturation à la fin de la simulation de diffusion, car le nombre de nœuds infectés dans le graphe est assez petit par rapport au nombre total de nœuds.

Ces résultats confirment le fait que la diffusion en utilisant une échelle de temps intrinsèque est insensible aux changements de topologie dans le réseau. Par conséquent ici, la définition du temps intrinsèque en utilisant la création de liens est pertinente : elle nous permet d'isoler le phénomène de diffusion et la dynamique du graphe.

### 3.3.2 Résultats sur le réseau de blogs (Webfluence)

#### 3.3.2.1 Simulation en temps extrinsèque

Nous avons réalisé sur le réseau Webfluence les mêmes expériences que pour le réseau Github. L'unité de temps extrinsèque est le jour. Le temps intrinsèque correspond à une nouvelle citation entre deux blogs.

Nous commençons l'étude de la diffusion en utilisant le temps extrinsèque (le jour). Pour la simulation, nous avons effectué dix tests de contamination par jour. Il y a par conséquent $1510$ tests de contaminations en temps extrinsèque, contre $12000$ en temps intrinsèque. Nous avons donc ici un plus grand nombre de pas de temps de diffusion en temps intrinsèque qu'en temps extrinsèque, contrairement à l'expérience sur Github. Les résultats de la simulation de diffusion en temps extrinsèque sont montrés sur la figure 3.7 (en haut à gauche).

Afin de faciliter l'interprétation, nous avons ajouté le nombre de nœuds en fonction du temps. Nous observons que le processus de diffusion a un début de croissance lent, et une phase de croissance rapide, comme prévu pour une propagation exponentielle. Au moment des vacances de Pâques (jours 65 à 80), nous observons un très léger ralentissement de la croissance.

#### 3.3.2.2 Simulation en temps extrinsèque converti en temps intrinsèque

Nous réalisons la même conversion du temps extrinsèque vers le temps intrinsèque que dans la section 3.2. Les résultats de cette conversion sont montrés sur la figure 3.7 (en haut à droite). Nous observons un comportement différent de celui de la représentation en extrinsèque (figure 3.7, en haut à gauche). Aux alentours du temps intrinsèque $5400$, qui correspond aux vacances de Pâques, la croissance du nombre de nœuds infectés augmente plus rapidement. On peut expliquer ce phénomène de la façon suivante : pendant les vacances de Pâques, la création de liens ralentit beaucoup, et on a donc moins de nouveaux liens par jour. Le temps intrinsèque ne prend pas en compte la vitesse de création des liens.

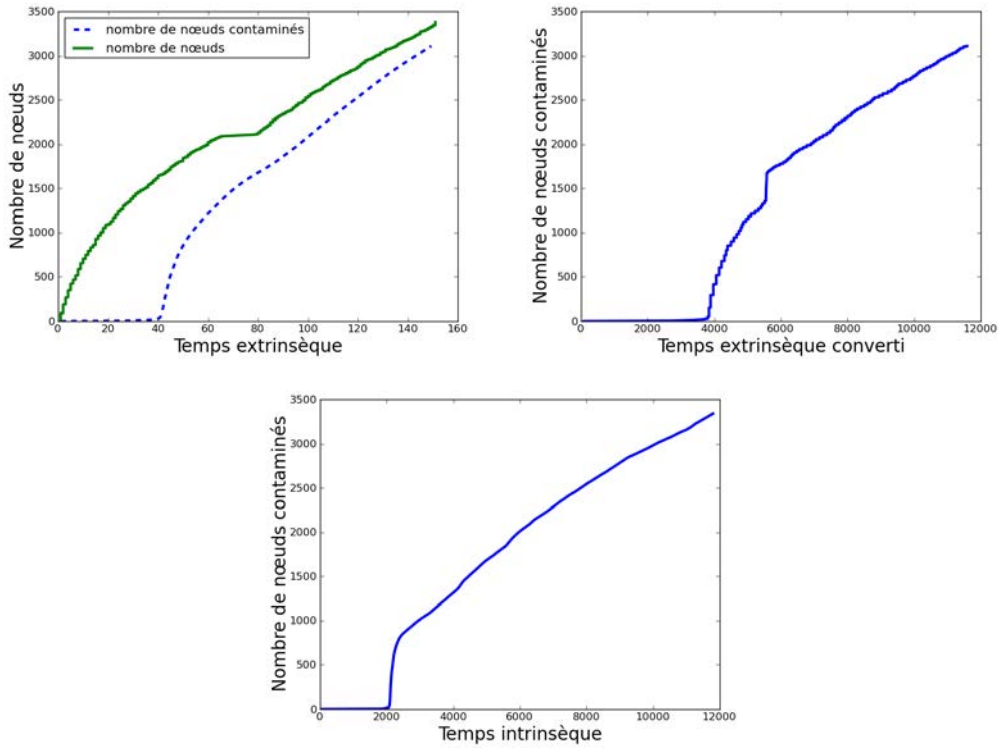



**Figure 3.7** – ***En haut à gauche*** *: en vert, diffusion avec un modèle SI sur le graphe des blogs simulée et observée en temps extrinsèque. Il y a dix tests de contaminations entre deux jours. En bleu : nombre de nœuds dans le graphe à chaque pas de temps.* ***En haut à droite*** *: diffusion avec un modèle SI sur le graphe de blog simulée en temps extrinsèque et observée en temps intrinsèque.* ***En bas*** *: diffusion avec un modèle SI sur le graphe de blogs simulée et observée en temps intrinsèque.*

Dans le réseau de blogs, le fait d'observer la diffusion en fonction du temps intrinsèque accentue les changements dans la dynamique du réseau, alors que dans le réseau Github, les modifications de topologie étaient déjà visibles en temps extrinsèque.

#### 3.3.2.3 Simulation en temps intrinsèque

La diffusion simulée en temps intrinsèque sur le réseau Webfluence est présentée en figure 3.7 (en bas). La diffusion démarre très lentement, et augmente soudainement à l'instant 2000. Quand nous regardons le jeu de données, nous voyons que ce temps intrinsèque correspond au vingtième jour de la mesure. Cette augmentation est corrélée au comportement naturel du modèle SI, et n'est pas liée à un événement dans le graphe. Après une augmentation très importante, le nombre de nœuds infectés augmente régulièrement, jusqu'à la fin de la mesure. Aucun

changement n'est observé pendant les vacances de Pâques (entre les instants 5100 et 5600 en temps intrinsèque).

Après cela, le nombre de nœuds infectés se rapproche du nombre total de nœuds, et nous observons une croissance plus linéaire. Comme dans le réseau Github, la diffusion en temps intrinsèque est moins affectée par les changements dans le réseau, ce qui signifie que la définition du temps intrinsèque utilisée ici (l'apparition de nouveaux liens) est valide.

### 3.3.3 Résultats sur le réseau de contacts Infectious SocioPatterns : impact de la définition du temps intrinsèque

Nous avons vu sur les jeux de données Github et Webfluence deux cas pour lesquels notre définition d'unité de temps intrinsèque liée à l'apparition d'un nouveau lien était très pertinente, puisqu'elle permettait de capturer la dynamique du graphe et de l'isoler de la diffusion. Dans cette section, nous allons nous intéresser à un autre type de réseau : un réseau de contacts. Comme expliqué dans la suite, ce jeu de données va nous permettre d'étudier les limites de la définition de temps intrinsèque.

#### 3.3.3.1 Simulation en temps extrinsèque

Nous étudions en premier lieu sur ce réseau de contacts des simulations de diffusion en temps extrinsèque. Nous faisons une simulation SI sur les quatre jours montrés sur la figure 2.10. Les résultats de ces simulations sont illustrés sur la figure 3.8 (colonne de gauche).

Nous voyons sur cette figure que les diffusions ont des comportements différents : la première (coin en haut à gauche) ressemble à un comportement standard de modèle SI avec la forme en S. Les trois autres sont plus irrégulières, et sont plus affectées par les événements du graphe. De plus, quand nous regardons la figure 2.10, il est difficile de trouver une corrélation entre les irrégularités dans la diffusion et les événements dans le nombre de nœuds et de liens : par exemple, pour la diffusion du 4 Juillet, nous observons une corrélation avec le nombre de liens, mais pour le 16 Juin, le comportement de diffusion est assez inattendu.

Dans la suite, nous utilisons la notion de temps intrinsèque pour voir si nous pouvons expliquer ce comportement.

#### 3.3.3.2 Simulation en temps extrinsèque converti en temps intrinsèque

Nous étudions maintenant les même processus de diffusion que précédemment, simulés en temps extrinsèque, mais convertis en temps intrinsèque, de la même

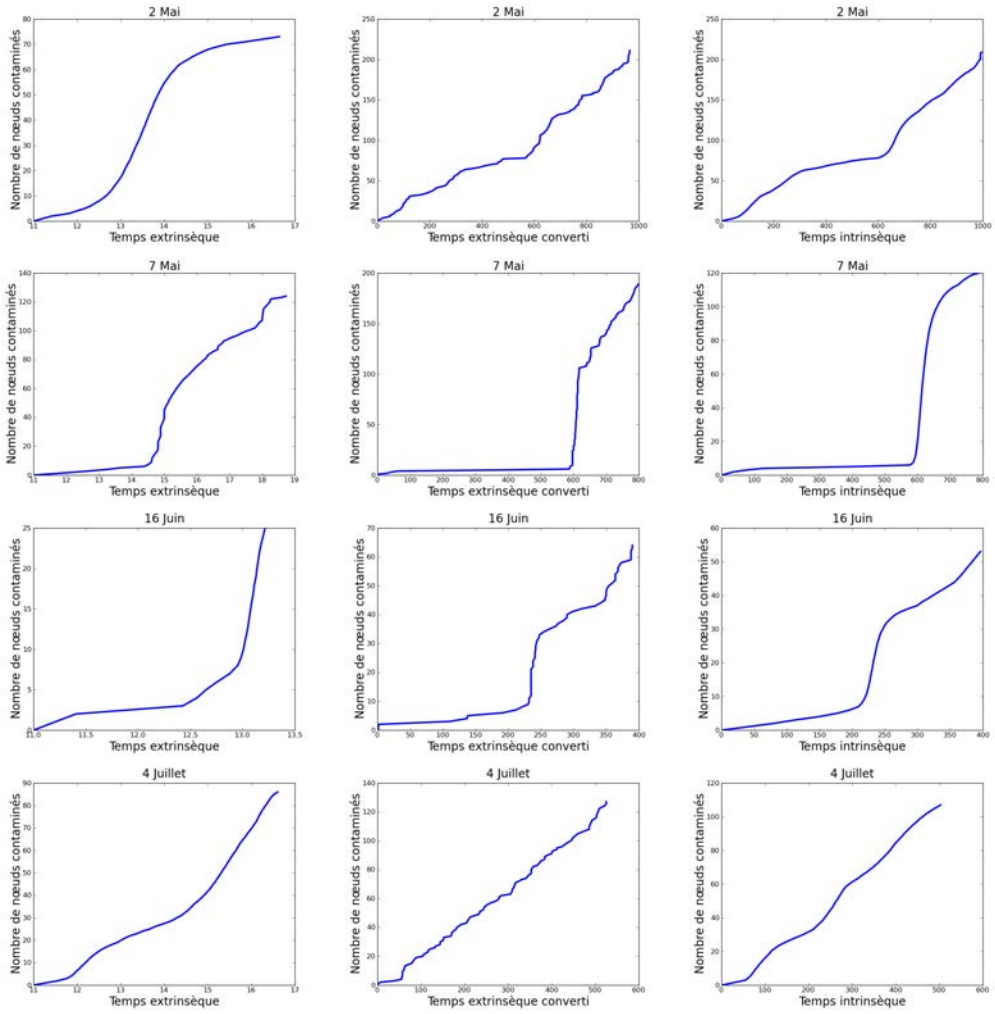



**Figure 3.8 –** ***Colonne de gauche*** *: diffusion avec un modèle SI simulée et observée en temps extrinsèque sur le réseau Infectious SocioPatterns.* ***Colonne de milieu*** *: diffusion avec un modèle SI diffusion simulée en temps extrinsèque et observée en temps intrinsèque sur le réseau Infectious SocioPatterns.* ***Colonne de droite*** *: diffusion avec un modèle SI simulé et observé en temps intrinsèque (défini par le nombre de nouveaux liens) sur le réseau Infectious SocioPatterns).*

façon que dans la section 3.2. Les résultats sont représentés dans la figure 3.8 (colonne du milieu). Le comportement de diffusion en temps extrinsèque converti est assez surprenant : le 2 Mai, on observe une croissance très irrégulière, alors qu'en temps extrinsèque, la diffusion est assez lisse. Le 7 Mai et le 16 Juin, les comportements sont globalement similaires, même si l'on peut observer quelques événements supplémentaires en temps extrinsèque converti. Le 4 Juillet, les deux courbes sont très similaires.

Pour chacun de ces jours, les événements observés en temps extrinsèque

converti sont difficiles à expliquer en utilisant le nombre de nœuds et de liens dans le graphe au cours du temps (figure 2.10). Nous simulons alors une autre diffusion en temps intrinsèque, afin de mieux comprendre ces résultats.

#### 3.3.3.3 Simulation en temps intrinsèque

Nous simulons maintenant une diffusion avec le modèle SI en temps intrinsèque sur ces quatre jours. Les résultats de ces simulations sont réunis sur la figure 3.8 (colonne de droite). Premièrement, nous voyons que les quatre courbes de diffusion ont des comportements différents : par exemple, sur la courbe en haut à droite, nous voyons un début de diffusion très lent, et une accélération brutale par la suite. En revanche, pour le 16 Juin, la croissance est lente au début, puis s'accélère, et ralentit de nouveau vers la fin de la diffusion. Quand nous comparons ces quatre simulations de diffusion avec le nombre de nœuds et de liens dans le réseau (figure 2.10), nous voyons que le comportement de diffusion est assez surprenant car très irrégulier. De plus, les comportements de diffusion en temps intrinsèque et extrinsèque sont très similaires.

Ces observations nous montrent que la définition de temps intrinsèque basée sur l'apparition de liens au cours du temps ne semble pas appropriée ici : en effet, nous voyons clairement que la diffusion en temps intrinsèque est très sensible aux événements dans le graphe, et par conséquent, nous ne pouvons pas isoler correctement la diffusion de la dynamique du graphe pour ce réseau. On peut imaginer une autre définition de temps intrinsèque qui capturerait mieux la dynamique du graphe et serait donc plus appropriée ici : par exemple, on pourrait prendre comme mesure de temps la création d'un nouveau triangle. Si le nombre de triangles fournit plus d'information que le nombre de liens, alors la nouvelle définition de temps intrinsèque sera plus appropriée sur ce réseau.

Les expériences menées sur ces différents réseaux nous montrent donc que l'utilisation du temps intrinsèque permet d'effectuer des observations complémentaires sur la diffusion, à la condition que la définition utilisée pour le temps intrinsèque soit appropriée.

## 3.4 Conclusion

Dans ce chapitre, nous avons utilisé la notion de temps intrinsèque pour étudier des phénomènes de diffusion dans des graphes dynamiques. Nous avons comparé nos résultats avec une diffusion étudiée à une échelle de temps extrinsèque, ce qui nous a permis de montrer l'intérêt de cette notion de temps. Nous avons tout d'abord illustré l'impact de l'échelle de temps sur des diffusions sur des graphes synthétiques Barabási-Albert. Cette étude a montré que le temps intrinsèque nous

permet en quelque sorte d'isoler la dynamique du graphe du phénomène de diffusion. Par conséquent, nous avons utilisé ce concept pour observer des simulations de diffusion sur trois jeux de données réels issus de domaines très différents : un réseau social, un réseau de blogs et un réseau de contacts.

Nos résultats sur ces jeux de données ont mis en évidence des différences importantes dans l'analyse de la diffusion en temps extrinsèque, extrinsèque converti en temps intrinsèque, et en temps intrinsèque, pourvu que la définition du temps intrinsèque soit pertinente. En effet, quand c'est le cas, la diffusion en temps intrinsèque est indépendante de la dynamique du graphe, comme nous avons pu le constater avec le comportement de la diffusion pendant l'événement du 4 Juillet dans le réseau Github. En temps extrinsèque en revanche, la dynamique de la topologie joue un rôle majeur.

Nous avons montré qu'il est très intéressant d'étudier une même diffusion en temps extrinsèque et en temps extrinsèque converti en temps intrinsèque, car la conversion de l'échelle de temps fournit des informations supplémentaires pour l'interprétation de la diffusion en temps extrinsèque : en effet, si l'on observe une accélération importante de la diffusion en temps extrinsèque pendant un événement (comme c'est le cas dans le réseau Github), cette accélération correspond à un ralentissement très important si l'on étudie la même diffusion représentée en temps extrinsèque converti en temps intrinsèque. Ce ralentissement signifie que la diffusion ralentit proportionnellement au nombre de liens, même si l'on observe une accélération en temps extrinsèque. Les trois échelles de temps utilisées ici nous donnent donc chacune des informations différentes, et la compréhension d'un phénomène de diffusion est améliorée par l'utilisation du temps intrinsèque.

En revanche, nous avons vu également avec le réseau Infectious SocioPatterns que le temps intrinsèque ne nous donne pas toujours plus d'information que le temps intrinsèque. Dans ce cas, la définition du temps intrinsèque (un pas de temps à chaque apparition de lien) n'est pas adaptée au graphe étudié : l'unité choisie n'est pas suffisamment représentative de la dynamique du graphe, et n'arrive donc pas à capturer correctement les événements dans le graphe. La simulation d'une diffusion dans une échelle de temps intrinsèque peut alors être un bon moyen pour valider une définition de temps intrinsèque et sa pertinence par rapport à un jeu de données.

La première perspective pour la suite de ces travaux consiste à étudier en temps intrinsèque des phénomènes de diffusion simulés sur d'autres jeux de données. Nous avons vu ici, sur trois jeux de données différents, que les observations peuvent changer de manière significative. Dans les deux premiers cas, l'échelle de temps intrinsèque utilisée a permis de mieux comprendre le comportement de la diffusion. Dans le dernier cas, la définition de temps intrinsèque utilisée n'était pas pertinente. L'étude d'autres graphes pourrait permettre de comprendre dans quel cas il est pertinent d'utiliser une échelle de temps intrinsèque pour étudier un phénomène de diffusion.

Une deuxième perspective consiste à chercher à étudier des diffusions réelles en utilisant une échelle de temps intrinsèque : on pourrait par exemple observer le cas de la diffusion d'une maladie, en supposant qu'elle se produit en milieu fermé. L'utilisation du temps intrinsèque pourrait alors permettre de voir à quel point la diffusion est affectée par les événements du graphe. Cela permettrait de déterminer si la diffusion a subi des influences extérieures au graphe. En effet, comme le temps intrinsèque permet de rendre la diffusion indépendante de la dynamique du graphe, si l'on observe des événements dans la diffusion, on sait alors qu'ils sont liés soit à un changement de topologie important, soit à une influence extérieure sur la diffusion.

Chapitre 4

# Application du temps intrinsèque aux communautés dynamiques

## Sommaire



Dans le chapitre précédent, nous avons mis en évidence l'impact très important que pouvait avoir l'échelle de temps utilisée pour l'analyse de la diffusion. Dans ce chapitre, nous mettons en œuvre la notion de temps intrinsèque pour la détection de communautés dynamiques. Plus précisément, nous cherchons à détecter des périodes de temps pendant lesquelles les communautés sont stables. Ce chapitre présente deux contributions. Nous proposons en premier lieu une nouvelle

façon de calculer ces communautés stables en utilisant une échelle de temps intrinsèque. Par la suite, nous montrons comment la visualisation en échelles de temps intrinsèque et extrinsèque peut aider à valider et interpréter les communautés obtenues.

## 4.1 Détection de communautés : définitions

### 4.1.1 Communautés de nœuds sur un graphe statique

Les graphes de terrain sont en général peu denses : si l'on choisit deux nœuds au hasard, ils ont une faible probabilité d'être connectés par un lien. En revanche, si l'on ne sélectionne plus deux nœuds au hasard, mais deux nœuds ayant un voisin en commun, alors la probabilité qu'ils soient reliés est beaucoup plus élevée. Ce phénomène, mesuré par le coefficient de clustering, s'explique par la présence de *communautés* dans le graphe, c'est-à-dire de groupes de nœuds très liés entre eux et moins liés aux autres nœuds du graphe.

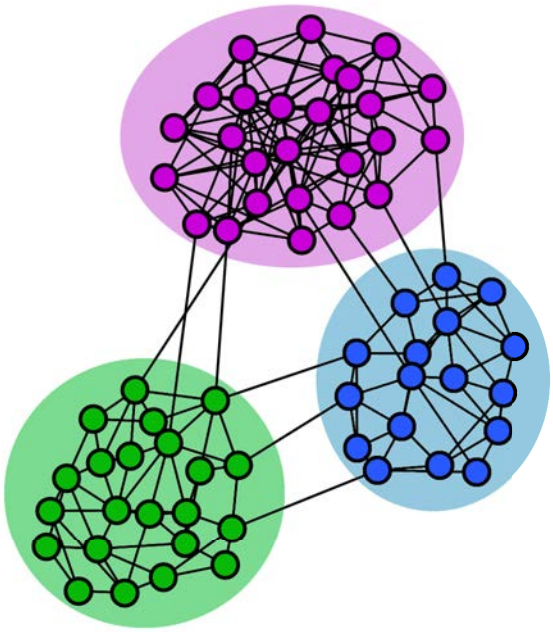


**Figure 4.1** – *Exemple de communautés de nœuds sur un graphe statique.*

Le problème de la détection de communautés dans un graphe consiste donc à chercher de tels groupes de nœuds. Pour des raisons de simplicité, de nombreux algorithmes partitionnent le graphe en communautés disjointes, et même si des

approches existent pour calculer des communautés recouvrantes, nous ne les utilisons pas dans nos travaux [1].

La figure 4.1 montre un exemple fictif de partition en trois communautés : on observe qu'il existe un grand nombre de liens au sein de chacune de ces communautés et très peu de liens entre ces communautés. Il existe dans la littérature plusieurs fonctions de qualité permettant d'estimer la pertinence d'un partitionnement d'un graphe, mais la plus utilisée est la *modularité* définie par [Newman and Girvan, 2004]. La modularité $Q(\pi)$ d'une partition $\pi$ pour un graphe $G = (V, E)$ est définie de la façon suivante :

$$Q(\pi) = \sum_{s \in \pi} \left( \frac{l_s}{L} - \left( \frac{d_s}{2L} \right)^2 \right)$$

où $L = |E|$ est le nombre de liens du graphe, $d_s$ le degré total d'une partie $s$ de $\pi$ (c'est-à-dire la somme des degrés des sommets dans s) et $l_s$ le nombre de liens à l'intérieur de $s$ (c'est-à-dire dont les deux extrémités sont dans s).

La valeur de la modularité, qui est toujours comprise entre -1 et 1, mesure la différence entre la proportion de liens à l'intérieur de chaque communauté et la proportion de liens que devrait avoir le même groupe de sommets dans un graphe aléatoire ayant la même distribution de degrés. Une partition est donc de bonne qualité s'il y a plus de liens au sein des communautés qu'attendu, et donc globalement peu entre ces communautés (l'implication n'est pas automatique et d'autres fonctions de qualité intègrent effectivement ce paramètre [Céspedes and Marcotorchino, 2013])). La partition triviale regroupant tous les nœuds dans une seule communauté ayant une modularité de 0, les seules partitions intéressantes ont donc une modularité positive.

### 4.1.2 Communautés de nœuds sur un graphe dynamique

Les graphes de terrain étant la plupart du temps dynamiques, la question de la détection de communautés de nœuds dynamiques se pose naturellement. La figure 4.2 illustre sur un petit exemple la notion de communautés dynamiques, ainsi que la notion de suivi de ces communautés.

Nous voyons que deux communautés existent au premier pas de temps. À l'instant suivant, la communauté bleue (en haut) se divise en deux, et la communauté verte (en bas) reste identique. Au dernier instant, une nouvelle communauté apparaît, et la communauté verte récupère deux nœuds de la communauté rouge (au milieu). Cet exemple met en évidence deux problématiques majeures pour la

1. Il est toutefois possible d'appliquer notre méthodologie à n'importe quel algorithme de détection de communautés, donc en particulier pour des communautés recouvrantes.

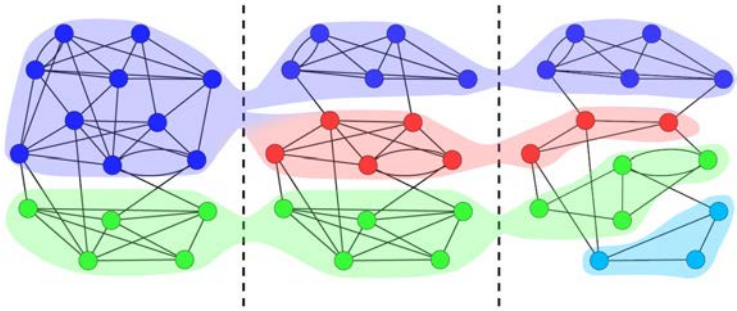


**Figure 4.2** – *Communautés calculées à trois instants (pas de temps) consécutifs, avec un suivi de l'évolution des différentes communautés.*

détection de communautés dynamiques : la détection de communautés à chaque instant, et le suivi de ces communautés au cours du temps (effectué ici grâce à la couleur des communautés).

Ce chapitre est organisé de la façon suivante : après une présentation des travaux existants sur la détection de communautés dynamiques dans les réseaux complexes ainsi que notre approche dans la section 4.2, nous détaillons les résultats de notre expérience sur un réseau de contacts dans la section 4.3. La section 4.4 est dédiée à la validation et l'interprétation de nos résultats à l'aide de la visualisation.

## 4.2 Travaux existants sur la détection de communautés dynamiques

[Aynaud et al., 2012] dressent un état de l'art détaillé sur la détection de communautés dynamiques. Dans cette section, nous nous limitons à exposer les principales approches, et pour plus de détails, nous invitons le lecteur à se reporter directement à cet état de l'art.

### 4.2.1 Algorithmes statiques sur instantanés de graphes dynamiques

Les premiers travaux sur la détection de communautés dynamiques ont appliqué des algorithmes de détection de communautés statiques, en considérant un graphe dynamique comme une succession de graphes statiques, et en appliquant sur chaque instantané un algorithme de détection classique. Cependant, cette méthode soulève un nouveau problème : le suivi des communautés au cours du temps. En effet, si l'on a $n$ communautés à un instant $t$, et $m$ communautés à un instant $t+1$, comment peut-on faire correspondre ces communautés entre elles ? Pourtant, à moins que les nœuds et les liens du réseau n'aient été complètement renouvelés, il

est très probable que la majorité des communautés à l'instant $t+1$ soit l'évolution des communautés à l'instant $t$.

[Hopcroft et al., 2004] ont été parmi les premiers à essayer de résoudre le problème de suivi des communautés. Pour cela, ils étudient la taille de l'intersection des communautés sur des pas de temps successifs. Si cette intersection est suffisamment grande, alors les communautés concernées sont considérées comme étant l'évolution l'une de l'autre. Cependant à cause de l'instabilité de l'algorithme de détection utilisé (qui n'est pas déterministe), ainsi qu'à cause d'autres problèmes, les communautés évoluent énormément entre deux pas de temps. Les auteurs restreignent donc leur étude à des communautés stables, c'est-à-dire à des communautés qui restent identiques même si le réseau est légèrement modifié. Cette contrainte diminue fortement le nombre de communautés étudiées, mais permet tout de même une première approche pour le problème du suivi des communautés.

Par la suite, cette approche a été généralisée dans les travaux de [Spiliopoulou et al., 2006], afin de prendre en compte des règles d'évolution des communautés plus complexes, comme la fusion, la séparation, l'apparition et la disparition. Les travaux de [Asur et al., 2007], [Falkowski et al., 2006], [Greene and Doyle, 2010], [Oliveira and Gama, 2010a] et [Oliveira and Gama, 2010b] ont introduit d'autres techniques très similaires en utilisant différentes façons de calculer la similarité entre les communautés.

Les travaux de [Wang et al., 2008] utilisent le même type d'approche pour le suivi des communautés, mais proposent de suivre chaque communauté avec uniquement quelques nœuds, au lieu de l'ensemble des nœuds de la communauté. Ils choisissent quelques nœuds centraux pour chaque communauté (la mesure de centralité choisie est débattue dans l'article) qui identifient la communauté à laquelle ils appartiennent à chaque pas de temps. Cette approche a été développée par la suite dans d'autres travaux [Beiró and Busch, 2010], [Chen et al., 2010], [Wang and Fleury, 2010].

Une autre direction de recherche consiste à construire un réseau représentant les relations entre les communautés trouvées à différents instants. On peut ensuite calculer des communautés sur le graphe des communautés (qui est donc un graphe statique contenant des informations temporelles). Les communautés obtenues sont donc directement des communautés dynamiques, ce qui permet de les suivre au cours du temps. Les travaux de [Chi et al., 2007], [Falkowski and Spiliopoulou, 2007] et [Tantipathananandh et al., 2007] utilisent cette approche.

### 4.2.2 Algorithmes directs de détection de communautés dynamiques

Une autre piste pour étudier les communautés dynamiques consiste à mettre au point des algorithmes qui prennent en entrée directement des graphes dynamiques,

ce qui évite de se ramener à une succession de graphes statiques. Les travaux de [S.-Y. Chan and Xu, 2009], [R. Kumar and Chakrabarti, 2006] et [Song et al., 2007] proposent pour cela de modifier la fonction de qualité afin qu'elle prenne en compte l'évolution temporelle des communautés. La nouvelle fonction de qualité est donc constituée de deux parties : une partie, qui ressemble à la modularité classique, permet de mesurer la pertinence d'une partition (d'un point de vue topologique), et l'autre partie mesure la stabilité de la partition au cours du temps, en la comparant aux partitionnements aux instants précédents et suivants.

Les travaux de [M. B. Jdidia and Fleury, 2007] et de [Mucha et al., 2010] représentent un graphe dynamique comme un graphe avec deux types de liens : des liens classiques et des liens temporels. Par exemple, il peut exister un lien temporel entre un nœud à l'instant $t$ et ce même nœud à l'instant $t+1$. Le graphe est donc observé sous forme de "tranches", avec des liens temporels entre les différentes tranches. Si l'on calcule des communautés sur ce type de graphe, on n'a plus de problème de suivi des communautés, car on obtient à la fois une partition du graphe et le suivi des communautés au cours du temps. En revanche, cette approche fait perdre de l'information concernant la dynamique du graphe : on ne connaît plus l'ordre d'arrivée des nœuds et des liens dans le graphe.

### 4.2.3 Algorithmes de détection de communautés en mode flux

Si l'on souhaite calculer à présent des communautés sur un flux de données, il faut trouver un moyen beaucoup plus efficace que tout recalculer à chaque pas de temps. De plus, les approches nécessitant une comparaison d'un pas de temps avec les instants futurs deviennent impossible à utiliser, et on ne se permet pas, en général, de stocker l'historique complet des évolutions. L'idée principale pour traiter ce genre de données est de mettre à jour les communautés à chaque instant, au lieu de les calculer à chaque fois.

Plusieurs approches ont été proposées pour mettre à jour les communautés. [Ning et al., 2007] choisissent d'utiliser les valeurs propres du réseau. Dans [Falkowski et al., 2008], les auteurs définissent une notion de distance entre les nœuds, ainsi qu'un voisinage des nœuds (ensemble de nœuds à une distance inférieure à un certain seuil). Quand un nouveau nœud ou lien arrive dans le graphe, il impacte donc seulement son voisinage, ce qui permet de délimiter une zone du graphe sur laquelle re-calculer des communautés.

[Görke et al., 2010] proposent une optimisation de la modularité et de l'algorithme de Louvain ([Blondel et al., 2008]) afin de gérer facilement de petites modifications dans le graphe. Le principe de base de l'algorithme de Louvain est de changer de façon gloutonne les communautés des nœuds. S'il existe déjà une partition, on peut utiliser le même algorithme, en fixant toutefois les nœuds non

affectés par le changement dans le graphe dans leur communauté précédente. Ainsi, seuls quelques nœuds se verront attribuer une nouvelle communauté, ce qui accélère largement l'algorithme par rapport à un calcul sur l'ensemble des nœuds.

[Dinh et al., 2010] proposent de travailler sur un réseau en groupant les nœuds en communautés. Ensuite, ils observent le réseau constitué de ces méta-nœuds. Quand un changement se produit, on divise les méta-nœuds concernés par le changement en rétablissant les nœuds orignaux, et on calcule des communautés sur le réseau obtenu, qui est très petit.

### 4.2.4 Détection de communautés stables au cours du temps

[Aynaud and Guillaume, 2011] ont proposé une approche très prometteuse. L'hypothèse de base de cette approche est que si une partition en communautés a été trouvée à l'instant $t$ et que le réseau n'évolue que très peu entre $t$ et $t+1$, alors la partition trouvée à l'instant $t$ est certainement encore valable à l'instant $t+1$ (sans aucune modification). Les auteurs généralisent ensuite cette idée et tentent de trouver des intervalles de temps de plusieurs pas de temps consécutifs, tels qu'une seule partition en communauté puisse représenter correctement les communautés sur l'intervalle de temps.

Afin de déterminer la qualité d'une telle partition les auteurs définissent la modularité moyenne $Q_{avg}$ d'une partition $\pi$ d'un graphe dynamique $G$ sur un ensemble de pas de temps $T$ de taille $n$ comme

$$Q_{avg}\left(G, \pi, T\right) = \frac{1}{n} \sum_{t \in T} Q\left(G_t, \pi\right)$$

et essayent ensuite de trouver des partitions et des intervalles de temps (fenêtres) pour lesquels la modularité moyenne est bonne.

La recherche de ces fenêtres et de ces partitions est effectuée en utilisant la définition de similarité entre les fenêtres, qui repose sur les partitions associées. L'idée consiste à dire que si le graphe est similaire sur deux pas de temps (ou deux fenêtres) distincts, alors la décomposition en communautés sur un pas de temps (ou une fenêtre) sera aussi une bonne décomposition pour l'autre. Les auteurs évaluent la pertinence de la décomposition pour deux fenêtres de temps $T_i$ et $T_j$, pour lesquelles les partitions associées sont $\pi_i$ et $\pi_j$, par la mesure de similarité $Sim$

$$Sim(T_i, T_j) = Q_{avg}(G, \pi_i, T_j) + Q_{avg}(G, \pi_j, T_i).$$

L'algorithme proposé par [Aynaud and Guillaume, 2011] et que nous utilisons dans la suite, est un algorithme de clustering hiérarchique classique, décrit par l'algorithme 1. Si le graphe dynamique est constitué de $n$ pas de temps, alors l'algorithme nécessite d'effectuer $n$ détections de communautés, une par pas de

temps. Ensuite, il recherche les deux pas de temps consécutifs les plus similaires (selon la fonction de similarité $Sim$), les agrège (si leur modularité est supérieure à un certain seuil [2]) et il calcule une nouvelle partition associée (en utilisant $Q_{avg}$). Cette procédure est répétée tant que la modularité demeure plus élevée que le seuil, et fournit ainsi un regroupement hiérarchique des fenêtres de temps.

**Algorithme 1** Algorithme de fusion des fenêtres de temps.

$G$ le graphe dynamique initial
$L$ la liste des instantanés potentiels, initialement vide
**pour tout** les instantanés $t$ de $G$ **faire**
    Calculer les communautés $\pi_{\{t\}}$ sur l'instantané $t$
    Insérer $\{t\}$ dans $L$
**fin pour**
**tantque** $L$ n'est pas vide **faire**
    Chercher $t_i$ et $t_j$ dans $L$ qui maximisent $Sim(T_i, T_j)$
    Enlever $t_i$ et $t_j$ de $L$
    Ajouter $t = t_i \cup t_j$ dans $L$
    Calculer $\pi_t$ les communautés de $G$ sur la fenêtre $t$
**fin tantque**

Cet algorithme peut fusionner n'importe quelle paire de fenêtres de temps, et pas seulement des fenêtres consécutives. Le résultat de l'algorithme fournit un arbre des intervalles de temps. À la racine de l'arbre, on trouve la fenêtre de temps la plus longue, si le seuil de similarité nécessaire pour grouper les fenêtres a été atteint jusqu'au bout. Sinon, on obtient une forêt, et les fenêtres de temps à la racine de chaque arbre ne peuvent plus être groupées. Selon la taille de fenêtre voulue, il est possible de considérer différents niveaux dans l'arbre. Une version plus simple de l'algorithme ne fusionne que les fenêtres ou les pas de temps consécutifs. Cette version est beaucoup plus rapide à calculer, puisque la maximisation de la similarité a uniquement besoin d'être calculée entre les paires de fenêtres de temps consécutives, et non entre toutes les paires. Cependant, si les communautés peuvent apparaître et disparaître plus tard, alors la version non adjacente de l'algorithme est nécessaire pour identifier la résurgence de ces communautés. Dans la suite, nous utilisons principalement la version adjacente pour des raisons d'efficacité, mais nous présenterons aussi certains résultats utilisant la version non adjacente.

La figure 4.3 montre un exemple de résultat de l'algorithme 1. Le graphe dynamique décrit en entrée est défini sur 6 pas de temps. La première étape consiste à calculer des communautés statiques sur chaque pas de temps du graphe. On insère dans la liste $L$ tous les pas de temps, donc de 1 à 6. Ensuite, pour l'ensemble des pas de temps dans $L$, on cherche le couple $(t_i, t_j)$ qui maximise la similarité. Sur la figure 4.3, ce sont les pas de temps 5 et 6 qui ont été trouvés. On ajoute alors

2. La question du seuil est abordée plus loin dans ce chapitre.

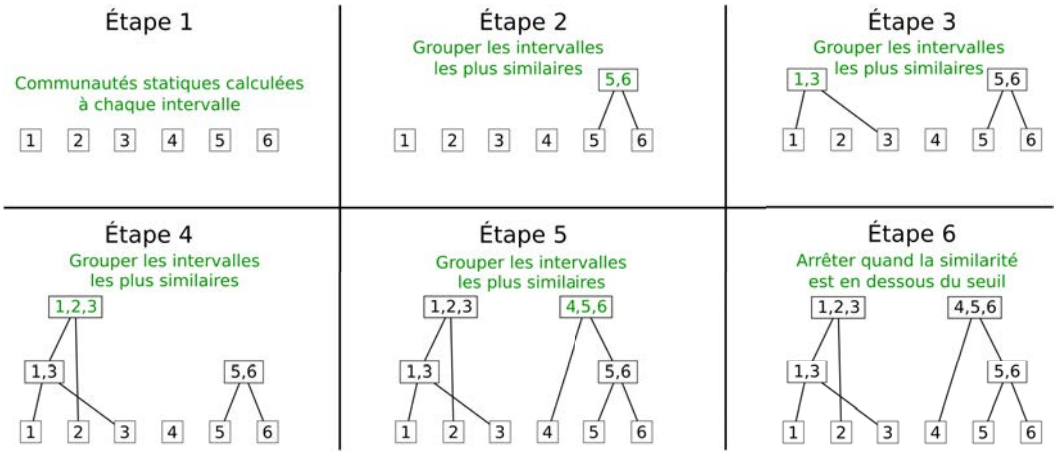



**Figure 4.3** – *Exemple d'arbres produits par l'algorithme de classification hiérarchique du temps.*

dans $L$ le couple $(5, 6)$, et on retire les instants 5 et 6. On recommence le même processus tant que la liste n'est pas vide. À l'étape 3, l'algorithme regroupe les intervalles de temps 1 et 3. Aux étapes 4 et 5, l'algorithme groupe les instants 1, 2 et 3, puis 4, 5, 6. Enfin, à l'étape 6, on n'observe plus de groupement car les intervalles dans $L$ maximisant la similarité ont une similarité inférieure à un certain seuil (dans cet exemple, le seuil est fixé à 0).

Cet algorithme, que ce soit dans sa version simple ou dans sa version non adjacente, souffre de plusieurs limitations. La première est le choix du niveau approprié dans l'arbre résultant du regroupement hiérarchique des fenêtres de temps. Aux niveaux les plus bas, seules quelques fusions ont été effectuées et les fenêtres de temps obtenues sont donc très petites. À l'inverse, pour les niveaux les plus élevés, il y a beaucoup de fusions, ce qui donne donc un plus petit nombre de fenêtres, mais la structure en communautés n'est peut-être pas très stable, et par conséquent pas forcément très significative. Le choix de la valeur du seuil de modularité en dessous duquel nous considérons que les communautés ne sont pas suffisamment stables est par conséquent stratégique.

Cette valeur a été fixée à 0 dans la version originale de l'algorithme, c'est-à-dire que l'on arrête de fusionner des fenêtres de temps quand la modularité moyenne obtenue devient négative. Cependant, cette valeur n'est pas forcément suffisamment restrictive. En particulier, quand la dynamique du réseau est très élevée, l'algorithme continue parfois à fusionner des périodes de temps alors qu'il ne devrait pas (si les fenêtre obtenues n'ont plus de sens pour la structure en communautés). Nous avons décidé dans ce chapitre de ne pas approfondir ce problème, et de garder la valeur initiale de 0 pour nos expériences. La conséquence directe est que l'algorithme ne produit pas un arbre de fenêtres de temps, mais une forêt, comme montré sur la figure 4.3. Nous nous concentrerons par la suite uniquement sur les niveaux élevés des arbres de cette forêt.

Un autre problème vient du fait que l'algorithme utilisé pour détecter les communautés stables n'est pas déterministe, puisqu'il est basé sur la méthode de Louvain décrite dans [Blondel et al., 2008]. Deux calculs sur le même jeu de données peuvent donc donner des résultats complètement différents. L'utilisation d'un algorithme déterministe de détection de communautés est un domaine de recherche à part entière, et nous n'avons aucune raison de préférer une exécution par rapport à une autre. Nous avons donc choisi de considérer une sortie aléatoire de l'algorithme. Ceci signifie que nous aurions pu obtenir de meilleurs résultats (ou de moins bons résultats) en utilisant une autre sortie. Notons que les travaux de [Diday, 1971] et de [Seifi et al., 2013] donnent des pistes pour gérer cet aspect non déterministe en combinant plusieurs exécutions de l'algorithme.

### 4.2.5 Notre approche

Dans ce chapitre, nous utilisons cet algorithme de classification hiérarchique du temps pour la détection de communautés stables en nous intéressant à l'importance de l'échelle de temps. La section suivante est dédiée à l'expérimentation de l'algorithme de détection des communautés stables en temps intrinsèque sur un jeu de données issu d'un réseau de contact. En effet, nous avons vu que la forêt produite en sortie par l'algorithme utilisé en temps extrinsèque peut être compliquée à analyser. De plus, comme cet algorithme permet de grouper des pas de temps, il est crucial de considérer les pas de temps que l'on utilise pour décrire le graphe dynamique. Nous appliquons donc cet algorithme à une échelle de temps intrinsèque sur un réseau de contacts dynamique, et nous comparons nos résultats avec ceux obtenus en temps extrinsèque. Par la suite, nous visualisons le graphe sur des fenêtres de temps pour valider les résultats que nous obtenons en temps intrinsèque.

Les résultats présentés dans ce chapitre correspondent à une exécution de l'algorithme 1. Nous avons effectué plusieurs exécutions, et avons vérifié que les résultats ne variaient pas de manière significative d'une exécution à l'autre. En effet, en exécutant une dizaine de fois l'algorithme, nous obtenons des fenêtres de temps très similaires (elles varient au maximum de cinq minutes).

## 4.3 Temps intrinsèque et communautés stables

Nous détectons des communautés sur le jeu de données Infocom 2006, décrit dans la section 2.4.5. Sur ce genre de réseau, la détection d'une unique structure en communautés sans prendre en compte la dynamique n'a pas vraiment de sens, puisque le graphe agrégé perd beaucoup d'information. Par exemple, les liens présents pendant une journée sont très rarement présents les autres jours. Par conséquent, la structure du graphe est complètement différente d'un jour à l'autre.

A l'inverse, détecter des communautés stables sur des fenêtres de temps peut être très pertinent ici, puisque l'on observe clairement différentes phases dans la dynamique. De plus, l'évolution du nombre de liens en fonction du temps extrinsèque donne une bonne compréhension de la dynamique du graphe. Sur la figure 2.11, nous pouvons en effet observer clairement les phases jour / nuit, ainsi que de nombreux événements pendant les journées. Nous avons par conséquent choisi comme définition du temps intrinsèque l'apparition et la disparition d'un lien (c'est-à-dire qu'à chaque apparition ou disparition d'un lien, on crée un pas de temps intrinsèque).

### 4.3.1 Communautés stables calculées et analysées en temps extrinsèque

Nous utilisons l'algorithme 1 détaillé dans la section 4.2 pour trouver des fenêtres de temps extrinsèques sur lesquelles les communautés sont stables. Nous calculons donc une seule partition du graphe, qui est pertinente sur toute la durée de chaque fenêtre de temps. Dans ce chapitre, nous cherchons à valider et interpréter ces fenêtres de temps en utilisant des échelles de temps différentes (extrinsèque dans cette section et intrinsèque dans la suivante).

Nous appliquons notre algorithme sur le graphe original ainsi que sur le graphe où le temps a été converti en temps intrinsèque. En effet, afin de pouvoir appliquer l'algorithme dans le cas du temps intrinsèque, nous devons considérer le graphe dans cette échelle de temps. Pour cela, nous le convertissons comme expliqué sur la figure 2.6. Sur ce graphe converti, nous utilisons ensuite le même algorithme de détection de communautés pour obtenir des fenêtres de temps. Nous n'attendons pas le même nombre de fenêtres de temps dans le cas intrinsèque et extrinsèque. En effet, comme le temps intrinsèque ralentit les événements et agrège les périodes stables, on peut s'attendre à obtenir plus de fenêtres en temps intrinsèque pendant les périodes avec de nombreuses créations et suppressions de liens.

Un calcul des communautés stables en temps intrinsèque identifie neuf grosses fenêtres de temps au niveau le plus élevé de l'arbre. Nous avons enlevé toutes les fenêtres de temps de durée inférieure à 500 pas de temps, c'est-à-dire 33 minutes.

La figure 4.4 représente ces neuf fenêtres de temps, ainsi que le nombre de liens. Les première et quatrième fenêtres correspondent aux deux premières nuits. Nous nous attendions à ce que l'algorithme trouve ces fenêtres, car elles correspondent à des périodes longues, avec peu d'activité. De plus, ces périodes sont très différentes des jours précédent et suivant, qui sont très actifs. Les troisième et quatrième nuits ne sont pas clairement détectées, ce qui peut être lié au fait que la journée entre ces deux nuits est moins active ; l'algorithme n'arrive pas à distinguer ces phases, au moins au niveau le plus élevé de l'arbre.

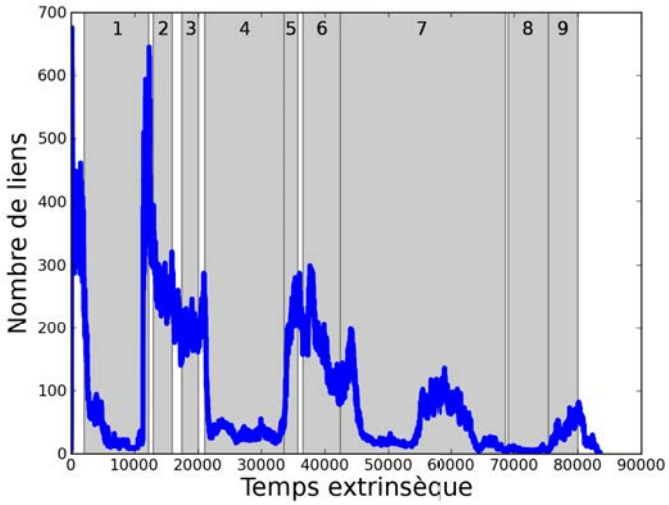



**Figure 4.4** – *Nombre de liens en fonction du temps extrinsèque dans le réseau Infocom 2006. Les communautés stables en haut de l'arbre sont représentées par les bandes verticales grises.*

Pendant les phases correspondant aux journées, il y a plus d'activité, et la structure du graphe évolue donc plus rapidement. Nous nous attendons donc à obtenir des fenêtres plus petites. Les fenêtres 2, 3, 5 et 6 sont effectivement très courtes, ce qui confirme notre intuition. À l'inverse, la fenêtre 7 est très grande et la phase de nuit (troisième nuit) est regroupée avec le jour suivant. Ce résultat est assez étonnant, car le nombre de liens semble tout de même varier suffisamment pour modifier la structure du graphe. L'absence de séparation est due au problème de seuil de modularité dans l'algorithme mentionné dans la section 4.2.4. Si nous prenons un seuil bas, nous risquons de grouper des fenêtres pendant lesquelles la structure du graphe évolue beaucoup. À l'inverse, en utilisant un seuil élevé, nous risquons de ne pas effectuer des regroupements de fenêtres de temps qui auraient pu être pertinents.

La structure hiérarchique des fenêtres de temps donnée par l'algorithme est montrée dans la figure 4.5 (pour les quatre niveaux les plus élevés) pour les fenêtres 2 et 7. Pour la fenêtre 2 (en haut sur la figure), les trois niveaux les plus hauts contiennent des fenêtres de très courte durée : au premier niveau, il y a seulement un pas de temps séparé du reste de la fenêtre (instant 15 931), et au second niveau, la nouvelle fenêtre (entre les instants 15 871 et 15 930) a une taille de moins de cent pas de temps. La seule division pertinente est au quatrième niveau, avec les fenêtres entre 12 917 et 14 700, et entre 14 701 et 15 870 (les fenêtres pertinentes sont colorées en vert). Si nous comparons ces résultats avec l'évolution du nombre de liens montrée sur la figure 4.4, nous voyons qu'il y a beaucoup de rafales, qui ne sont pas identifiées dans les divisions de l'arbre. Si nous considérons la période, contenant de nombreuses rafales, couverte par la fenêtre extrinsèque 2, le fait de

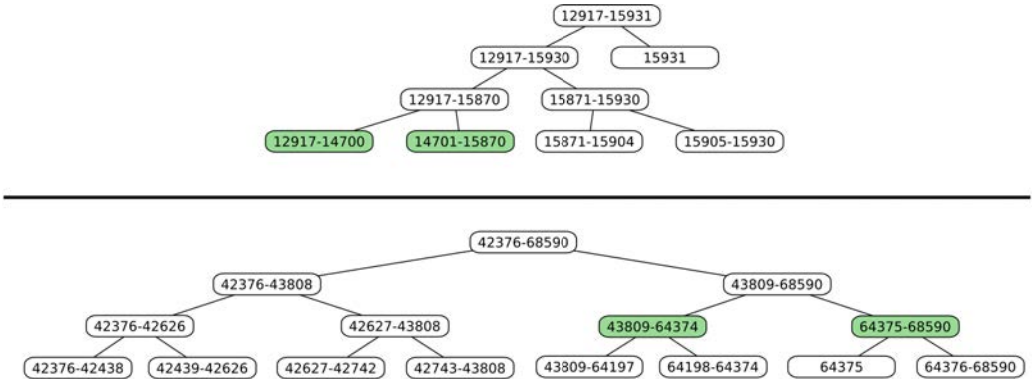



**Figure 4.5** – *Structure hiérarchique pour les fenêtres de temps 2 (en haut) et 7 (en bas).*

ne pas trouver de divisions plus significatives signifie que l'algorithme a du mal à calculer une structure en communautés stable sur ce type de réseau dynamique.

Pour la fenêtre 7 (en bas de la figure), nous observons des résultats similaires : la branche gauche de l'arbre contient uniquement des petites divisions. Sur la branche droite, on observe une division significative à l'instant 64 374 (colorée en vert). Sur la figure montrant le nombre de liens (figure 4.4), cet instant correspond au début de la phase de nuit. Les autres divisions sont trop petites pour être pertinentes.

L'algorithme utilisant le temps extrinsèque nous renvoie des résultats intéressants, mais la division de l'arbre montre que les fenêtres que nous nous attendions à trouver ne peuvent pas être vues dans les niveaux inférieurs de l'arbre. Cependant, même si nous considérons toutes les fenêtres à un niveau donné de l'arbre, elles ne sont pas toutes pertinentes ; la sélection des fenêtres intéressantes est par conséquent une question complexe. L'utilisation du temps intrinsèque dans la section suivante vise à rendre la dynamique du réseau plus régulière. L'objectif est d'obtenir directement grâce à cette échelle de temps les fenêtres de temps les plus pertinentes au niveau le plus élevé de l'arbre, sans avoir à effectuer un traitement a posteriori sur l'arbre pour y trouver les informations pertinentes.

### 4.3.2 Communautés stables calculées et analysées en temps intrinsèque

Nous convertissons maintenant notre graphe en temps intrinsèque afin de calculer les communautés stables et leurs fenêtres de temps intrinsèque associées. Cette conversion devrait nous donner des informations sur les phases avec beaucoup d'activités, puisqu'elle les étire. Nous obtenons 19 grandes fenêtres de temps (de plus de 500 pas de temps intrinsèque) au niveau le plus élevé de l'arbre.

La figure 4.6 représente ces fenêtres de temps, superposées avec le nombre de liens en fonction du temps intrinsèque. Nous voyons tout d'abord que le nombre

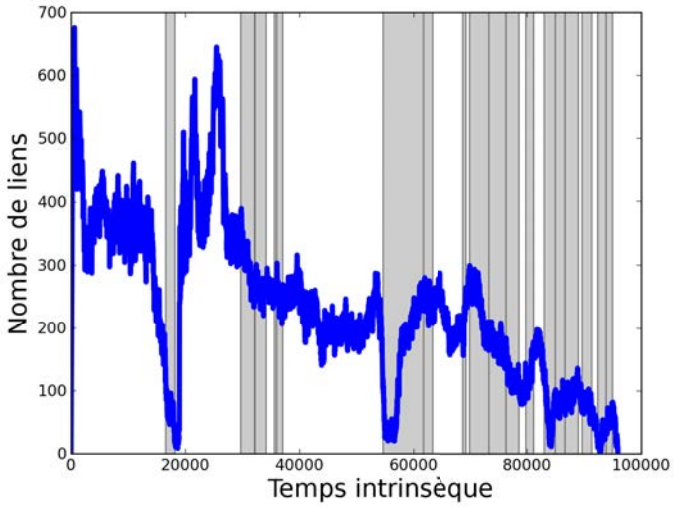



**Figure 4.6** – *Nombre de liens en fonction du temps intrinsèque dans le réseau Infocom 2006. Les communautés stables sont représentées par les bandes verticales grises.*

de liens représenté en temps intrinsèque est très différent de sa représentation en temps extrinsèque. Les phases de nuit sont naturellement raccourcies, alors que les phases de jour avec beaucoup d'activité occupent la plus grande partie du temps intrinsèque.

De plus, les fenêtres pertinentes en temps intrinsèque couvrent moins de pas de temps qu'en temps extrinsèque : un plus grand nombre de fenêtres sont détectées, mais elles sont plus courtes. La première fenêtre couvre la première nuit de la conférence. Après cela, la plupart des fenêtres correspondent à des phases de jour, avec beaucoup de variations dans le niveau d'activité. La fin de la mesure est couverte par beaucoup de petites fenêtres de temps, alors qu'il y en a une seule en temps extrinsèque. De manière générale, les fenêtres de temps intrinsèques sont plus pertinentes que les fenêtres de temps extrinsèques, puisqu'elles ne regroupent pas les phases de nuit et de jour. De plus, elles divisent les périodes de jour de façon plus précise : il y a plus de fenêtres de temps intrinsèque que de fenêtres de temps extrinsèque sur chaque jour.

Le calcul des communautés stables en utilisant le temps intrinsèque nous donne des résultats très différents de ceux calculés en temps extrinsèque. Dans la section suivante, nous comparons ces résultats de façon plus systématique.

### 4.3.3 Comparaison des notions de temps

Nous avons vu dans les sections précédentes que le temps intrinsèque ralentit les rafales et agrège les périodes stables. Par conséquent, il existe un plus grand

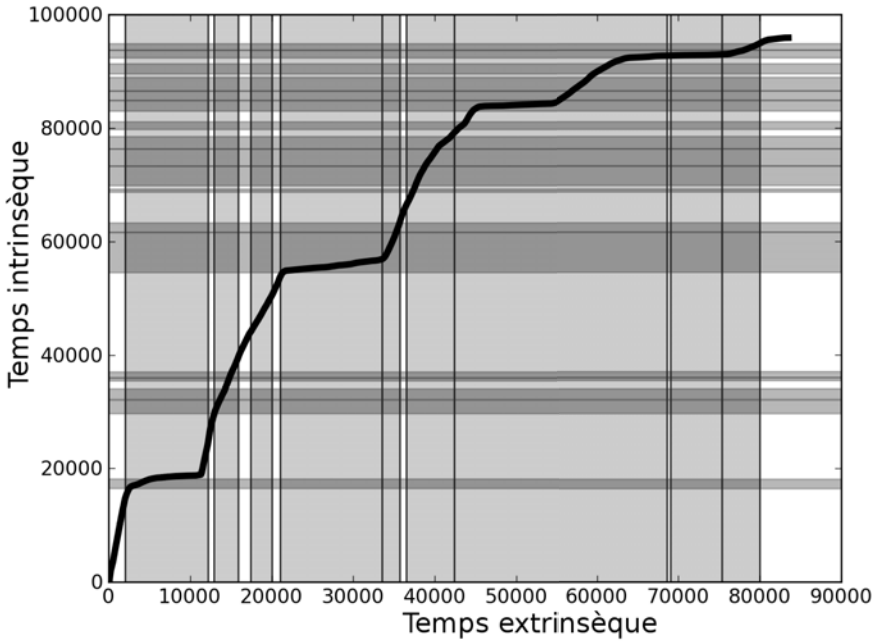



**Figure 4.7** – *Temps intrinsèque représenté en fonction du temps extrinsèque. Les bandes gris clair verticales représentent les fenêtres de temps extrinsèque, et les bandes gris foncé horizontales représentent les fenêtres de temps intrinsèque.*

nombre de fenêtres de temps différentes en temps intrinsèque. Pour valider cette intuition, nous comparons les résultats obtenus avec les deux méthodes grâce à une courbe représentant le temps intrinsèque en fonction du temps extrinsèque. Pour chaque échelle de temps, nous traçons les fenêtres de temps. Ainsi, nous comparons les fenêtres obtenues dans les deux échelles. La figure 4.7 montre cette représentation croisée des courbes présentées dans les figures 4.4 et 4.6, c'est-à-dire une représentation du temps intrinsèque en fonction du temps extrinsèque.

#### 4.3.3.1 Changement d'échelle

Nous voyons tout d'abord sur cette figure que le changement d'échelle du temps extrinsèque vers le temps intrinsèque n'est absolument pas homogène. Certaines phases sont beaucoup plus courtes, alors que d'autres sont allongées. En particulier, les quatre plateaux de la courbe correspondent aux phases de nuit : en temps extrinsèque, la nuit dure longtemps (environ 28 800 secondes, ce qui correspond à 7200 pas de temps), et donc s'étale sur de nombreux pas de temps, alors qu'en temps intrinsèque, la nuit est très courte car il y a très peu d'activité (environ 500 pas de temps). Au contraire, les phases de jour sont étirées en temps intrinsèque, en raison de l'activité importante pendant les journées de conférence. Cela correspond aux portions où la croissance est la plus rapide de la courbe.

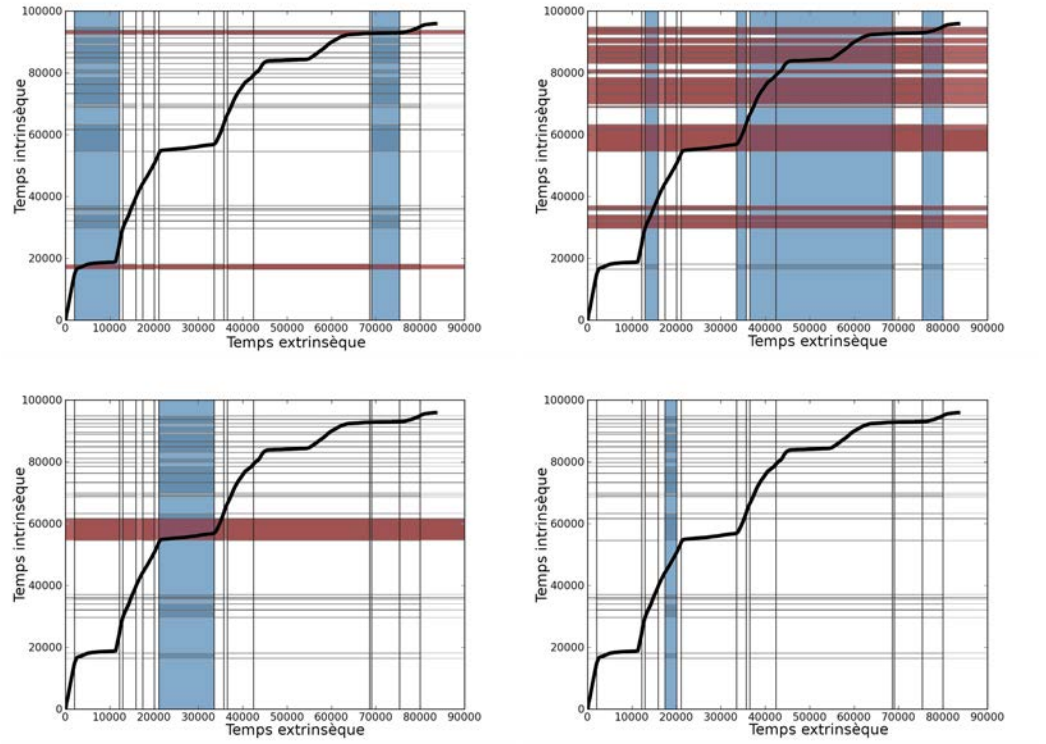



**Figure 4.8** – *Temps intrinsèque représenté en fonction du temps extrinsèque. Les bandes bleues verticales représentent les fenêtres de temps extrinsèques, et les bandes rouges horizontales représentent les fenêtres de temps intrinsèques.*

La figure 4.7 nous permet analyser chaque fenêtre selon les deux échelles de temps. Nous distinguons quatre types de croisement.

#### 4.3.3.2 Validation des fenêtres identiques

Premièrement, certaines fenêtres couvrent la même période, comme on peut le voir sur la figure 4.8 (en haut à gauche) pour la première et la huitième fenêtres extrinsèques. Dans cette situation, les deux notions de temps donnent les mêmes périodes sur lesquelles les communautés sont stables, ce qui confirme la pertinence de ces fenêtres et des partitions associées dans le réseau. Dans ces cas-là, le seuil utilisé par l'algorithme pour arrêter de fusionner les fenêtres de temps est par conséquent bien adapté.

#### 4.3.3.3 Fenêtres plus nombreuses en temps intrinsèque

Une fenêtre de temps extrinsèque peut être divisée en plusieurs fenêtres de temps intrinsèques (figure 4.8 en haut à droite). C'est le cas pour les fenêtres extrinsèques 2 (divisée en 4 fenêtres intrinsèques), 5 (divisée en 2), 6 (divisée en 4),

7 (divisée en 5) et 9 (divisée en 2). Les fenêtres extrinsèques 2, 5 et 6 sont toutes pendant des phases de jour, alors que la fenêtre 7 regroupe une phase de nuit et une phase de jour. Cette division pendant les phases de jour est très intéressante. En effet, pendant une journée, il y a beaucoup d'activité, et la structure en communauté évolue vraisemblablement assez vite. La division en plusieurs fenêtres de temps intrinsèque nous fournit une information plus détaillée sur l'évolution des communautés. Pour la fenêtre 7, cette division est aussi potentiellement pertinente, puisque la fenêtre extrinsèque couvre une phase de nuit et une phase de jour.

Dans ce cas de figure, le temps intrinsèque divise les périodes extrinsèques en un grand nombre de fenêtres intrinsèques. Nous avons expliqué précédemment que l'algorithme utilisé pour détecter les communautés stables retourne une hiérarchie des fenêtres de temps : au sommet de la forêt, on trouve les plus grandes fenêtres. Il est par conséquent naturel de penser que les fenêtres de temps extrinsèques que nous avons trouvées peuvent être divisées dans des niveaux inférieurs de la forêt de la même façon que les fenêtres intrinsèques obtenues. Si nous comparons nos résultats calculés avec le temps intrinsèque à la structure de l'arbre (figure 4.5), nous voyons que dans l'arbre, les sous-fenêtres n'apparaissent pas clairement.

Pour la fenêtre extrinsèque 2, dans l'arbre, la seule division correspondant au temps intrinsèque est au quatrième niveau de l'arbre (aux alentours de l'instant 14 700). Les autres fenêtres dans l'arbre divisent le temps seulement en périodes très courtes. Les 3 autres fenêtres intrinsèques ne peuvent pas être trouvées dans l'arbre.

Le cas de la fenêtre extrinsèque 7 est très similaire : nous ne pouvons pas trouver dans l'arbre une division qui correspond à une fenêtre de temps intrinsèque. Toutes les fenêtres dans l'arbre sont très courtes. Nous voyons pour ces deux cas que considérer un niveau plus bas dans l'arbre nous donne des résultats moins pertinents que de calculer directement les communautés stables en temps intrinsèque. En effet, en étirant les événements, le changement d'échelle en temps intrinsèque autorise l'algorithme à calculer des meilleures fenêtres qu'en temps extrinsèque, quand la dynamique du réseau est trop irrégulière. Le temps intrinsèque nous permet donc d'obtenir directement des fenêtres de temps pertinente, et qui sont pour la plupart absentes dans la forêt obtenue en temps extrinsèque, même à des niveaux bas.

Nous voulons alors déterminer si l'algorithme utilisé en version non adjacente nous permet de déterminer des fenêtres de temps extrinsèques plus pertinentes. Nous l'appliquons sur les fenêtres extrinsèques 2 et 7, en autorisant le regroupement de pas de temps ou de fenêtres de temps non consécutifs. Cette version non adjacente est utile pour regrouper des phases qui sont similaires, mais pas contiguës en temps, comme par exemple des phases de nuits. Cependant, la complexité de la version non adjacente de l'algorithme est beaucoup plus élevée que la version adjacente, et est trop importante pour calculer la structure en communautés sur

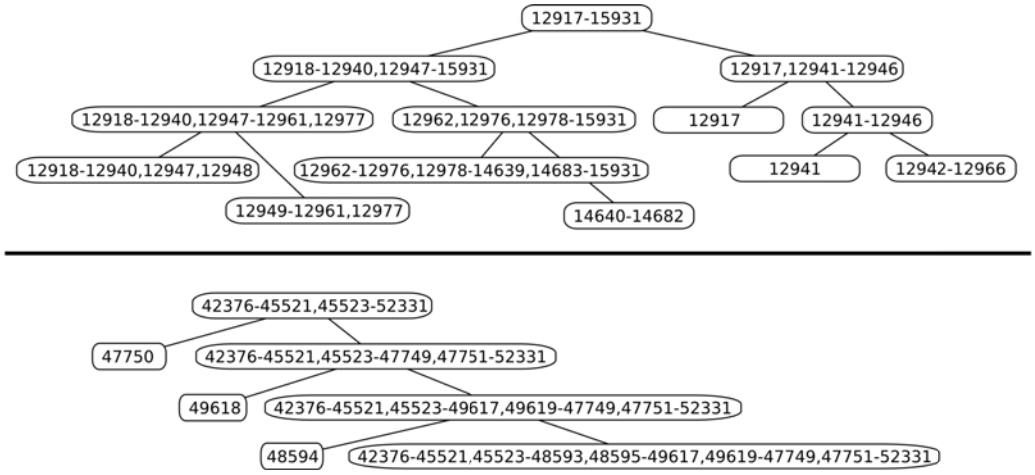



**Figure 4.9** – *Structure hiérarchique pour les fenêtres extrinsèques 2 (en haut) et 7 (en bas).*

l'ensemble du graphe. C'est pourquoi nous l'utilisons uniquement sur deux fenêtres de temps.

Les arbres obtenus de cette manière sont montrés sur la figure 4.9 : l'arbre correspondant à la figure 2 est en haut, et celui de la fenêtre 7 est en bas. Pour la fenêtre 2, sur la branche droite de l'arbre, nous voyons qu'à chaque niveau, seul un pas de temps est séparé du reste. Nous cherchons alors à déterminer si la structure du graphe à cet instant de temps est très différent du reste. Le pourcentage de liens en commun entre le pas de temps isolé et les pas de temps autour est compris entre 99% et 100%. De plus, c'est le cas pour chaque pas de temps isolé dans le graphe (et on observe la même chose pour la fenêtre 7). Ces divisions ne sont par conséquent pas pertinentes pour la structure en communautés.

Sur la branche gauche de l'arbre, les divisions sont un peu plus longues (environ 20 pas de temps), mais elles sont également très similaires aux pas de temps autour, si nous regardons le pourcentage de liens en commun. Dans ce calcul de fenêtres de temps non adjacentes pour les communautés stables, nous n'obtenons pas de résultat pertinent. En particulier, nous ne trouvons aucune division proche de celles trouvées en intrinsèque, ni aucune fenêtre pertinente.

Pour l'arbre de la fenêtre 7, nous avons un scénario quasiment identique : chaque fenêtre est séparée avec seulement un pas de temps exclu. La similarité entre ce pas de temps et le reste est très élevée, et la séparation de ces fenêtres ne nous donne pas d'information à propos de la structure en communautés. Comme dans le cas la fenêtre extrinsèque 2, la version non adjacente de l'algorithme ne nous aide pas à distinguer des fenêtres significatives, alors qu'étant donné la période étudiée, il existe clairement différentes phases. En temps intrinsèque, nous pouvons donc au final calculer de manière plus simple et moins coûteuse des fenêtres de temps plus pertinentes et plus détaillées. Ceci confirme donc l'intérêt de la notion de temps intrinsèque, qui est simple à appliquer, et plus pertinente.

#### 4.3.3.4 Fenêtres absentes en temps intrinsèque

Sur la figure 4.8 (en bas à gauche), nous observons que la fenêtre extrinsèque 3 ne croise aucune fenêtre intrinsèque, même partiellement. Cette fenêtre correspond à une période avec beaucoup d'activité (comme montré sur la figure 4.4). Le fait de pouvoir trouver des communautés stables sur cette fenêtre est donc un défaut de l'algorithme en temps extrinsèque. De plus, si l'on remonte le seuil minimal de similarité pour fusionner des pas de temps à 0,3 au lieu de 0, cette fenêtre n'est plus détectée par l'algorithme. En revanche, le temps intrinsèque nous donne automatiquement un résultat plus pertinent.

#### 4.3.3.5 Limites de l'algorithme en temps intrinsèque

La dernière possibilité, montrée par la figure 4.8 (en bas à droite), est d'avoir une fenêtre de temps extrinsèque plus pertinente que la fenêtre de temps intrinsèque qu'elle croise. C'est le cas pour la fenêtre extrinsèque 4. Dans ce cas, l'algorithme nous donne un meilleur résultat en temps extrinsèque qu'en temps intrinsèque : en effet, en temps intrinsèque, la nuit est groupée avec le début du jour suivant, malgré des changements de topologie très importants, alors qu'en temps extrinsèque, la fenêtre couvre juste la durée de la nuit. La mauvaise détection de la fenêtre en temps intrinsèque est liée à un problème de seuil. En effet, si l'on remonte légèrement le seuil de similarité nécessaire pour grouper, alors la nuit n'est plus groupée avec le début de la journée.

#### 4.3.3.6 Conclusion

En conclusion de cette analyse, nous voyons que pour 18 des 19 fenêtres obtenues, le calcul en fenêtres de temps intrinsèque est très pertinent, en particulier quand les fenêtres de temps extrinsèques sont divisées en beaucoup de fenêtres intrinsèques. De plus, il y a quelques fenêtres qui sont identiques en temps intrinsèque et extrinsèque, ce qui permet de valider ces périodes de temps pour la détection de communautés stables. Dans notre étude, nous avons un seul cas de fenêtre où le temps intrinsèque donne de moins bons résultats que le temps extrinsèque.

## 4.4 Visualisation et interprétation des communautés

Nous avons calculé et comparé dans la section précédente les fenêtres de temps dans les deux échelles. Dans cette section, nous étudions leur pertinence. Par exemple, pourquoi pouvons-nous observer cinq fenêtres de temps intrinsèque

pendant la fenêtre extrinsèque 7 ? Pour répondre à ces questions, nous utilisons la visualisation du graphe sur ces différentes fenêtres : nous nous intéressons ici au cas des fenêtres extrinsèques 2 et 7, qui sont divisées en de nombreuses fenêtres en temps intrinsèques. Pour chacune des fenêtres extrinsèques, nous visualisons le graphe agrégé sur l'ensemble de la fenêtre, puis le graphe agrégé uniquement sur chaque fenêtre intrinsèque. Cette méthode nous permet de comparer la pertinence des fenêtres, et de déterminer si la division en petites fenêtres données par le temps intrinsèque correspond bien à des changements de structure important dans le graphe.

La validation des communautés stables à l'aide de la visualisation est efficace seulement si le graphe n'est pas trop gros : sinon, la visualisation deviendrait confuse et inexploitable. Pour des graphes de plus grande taille, il est plus judicieux d'utiliser des mesures statistiques, comme la distance entre deux graphes agrégés.

### 4.4.1 Premier exemple d'interprétation

Dans la section 4.3, nous avons montré que la fenêtre extrinsèque 2 était divisée en quatre fenêtres intrinsèques. De plus, nous avons vu que pendant la période de temps couverte par cette fenêtre (qui correspond à une phase de jour), il y a beaucoup de rafales dans l'évolution du nombre de liens. Afin d'évaluer la pertinence des fenêtres de temps intrinsèques trouvées précédemment, nous représentons sur la figure 4.10 le graphe agrégé sur la totalité de la fenêtre extrinsèque (en haut à droite), puis ensuite agrégé seulement sur chaque fenêtre intrinsèque (fenêtres 3, 4, 5 et 6).

Le graphe agrégé sur la fenêtre extrinsèque est très dense, ce qui est naturel puisqu'il couvre une phase de jour. On observe que chaque graphe correspondant à une fenêtre intrinsèque est très différent du graphe extrinsèque. Entre les fenêtres intrinsèques 3 et 4, nous pouvons observer beaucoup de différences dans la structure du graphe : sur la fenêtre 3, le graphe est plus dense. Il correspond à la fenêtre qui contient le plus grand nombre de liens (figure 4.6). De plus, une partie du graphe très dense sur la première visualisation intrinsèque devient très peu dense pendant la seconde fenêtre. Par contre, la partie du graphe en bas à droite, malgré une baisse de densité, reste assez bien connectée.

Les graphes des fenêtres 4 et 5 sont un peu plus similaires, mais présentent malgré tout des différences importantes. La partie en bas à droite devient plus dense à nouveau, mais la partie en haut à gauche n'évolue pas énormément. Enfin, entre les graphes des fenêtres 5 et 6, la partie en bas à droite ne change pas, mais en revanche dans le reste du graphe, on peut observer beaucoup de créations de liens, ce qui explique le changement dans la structure en communautés entre ces deux fenêtres.

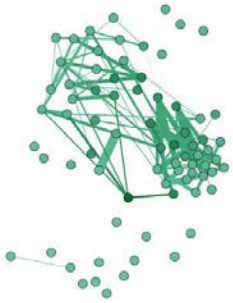
Cette visualisation nous montre que toutes les fenêtres de temps intrinsèques trouvées pendant la fenêtre extrinsèque 2 sont pertinentes. Entre chacune d'entre

Fenêtre extrinsèque 2

Fenêtre intrinsèque 3

Fenêtre intrinsèque 4

Fenêtre intrinsèque 5



Fenêtre intrinsèque 6

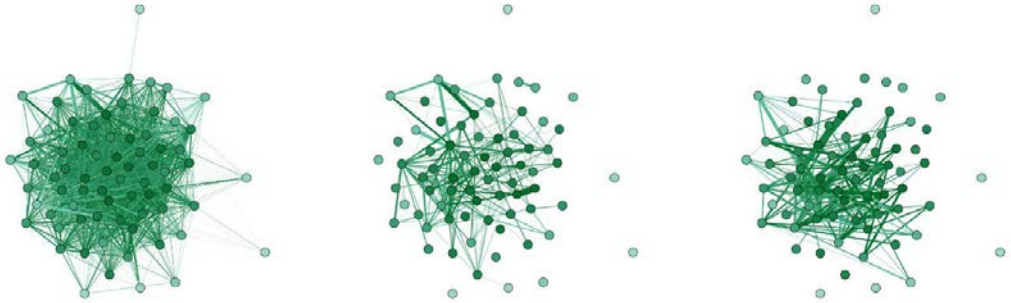
**Figure 4.10** – *Graphe agrégé sur la fenêtre de temps extrinsèque 2, et graphe agrégé sur chaque fenêtre de temps intrinsèque croisant la fenêtre 2 (fenêtres 3 à 6).*

elles, la structure du graphe change de manière importante. La division intrinsèque nous permet donc d'obtenir une analyse plus détaillée de l'évolution de la structure en communautés qu'en temps intrinsèque. Sur une courte période de temps (correspondant à la fenêtre extrinsèque 2), le temps intrinsèque peut donc permettre une division pertinente pour la structure en communautés.

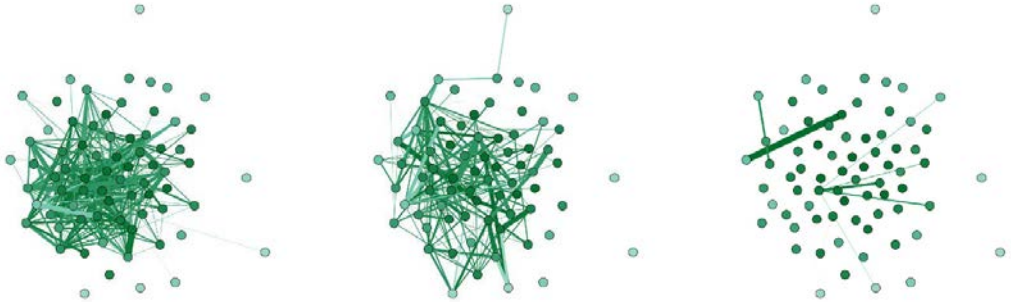
### 4.4.2 Second exemple d'interprétation

Nous nous intéressons à présent à la fenêtre extrinsèque 7. Cette fenêtre groupe une phase de nuit et une phase de jour, et est divisé en cinq fenêtres de temps intrinsèques. Nous cherchons à savoir ici si ces fenêtres sont pertinentes. Les visualisations de ces fenêtres sont montrées dans la figure 4.11.

Sur cette figure, le graphe agrégé en temps extrinsèque est en haut à gauche. Ensuite, nous voyons les graphes agrégés sur chaque fenêtre de temps intrinsèque. Nous observons sur cette figure que le graphe agrégé en temps extrinsèque est très dense : excepté deux nœuds qui ont un degré égal à 1, tous les nœuds ont beaucoup de voisins. Malgré la très petite taille du graphe, cette visualisation est très confuse. Il y a trop de liens, et nous ne pouvons rien voir, à part que le graphe est dense.



Fenêtre extrinsèque 7 | Fenêtre intrinsèque 14 | Fenêtre intrinsèque 15



Fenêtre intrinsèque 16 | Fenêtre intrinsèque 17 | Fenêtre intrinsèque 18

**Figure 4.11** – *Graphe agrégé sur la fenêtre de temps extrinsèque 7, et graphe agrégé sur chaque fenêtre intrinsèque croisant la fenêtre 7 (entre 14 et 18).*

Sur chaque fenêtre de temps intrinsèque, les graphes agrégés sont plus lisibles. Le premier graphe correspond à la phase de nuit, et est par conséquent très peu dense. Sur les trois graphes suivants (les fenêtres de temps intrinsèques 15, 16 et 17), il y a plus de liens. La différence entre les deux premières fenêtres de temps intrinsèque est très claire, et valide la division de ces deux périodes.

Les fenêtres 15, 16 et 17 sont toutes assez denses, mais chacune est différente des autres. Les nœuds de haut degré changent entre chaque fenêtre, et certains nœuds connectés deviennent pratiquement inactifs. Il n'est par conséquent pas surprenant que la structure en communautés change entre chacune de ces fenêtres. En effet, la topologie du graphe montre des évolutions importantes. La fenêtre 18 est un cas particulier : elle est très peu dense, et différente des autres fenêtres. Cela est dû au fait que cette fenêtre est entre les fenêtres extrinsèques 7 et 8. Sur la visualisation, elle couvre seulement la fin de la fenêtre 7, et est donc coupée avant la fin.

Nous avons vu avec cette visualisation que les fenêtres de temps extrinsèques peuvent être découpées de manière plus fine et plus pertinente en utilisant le temps intrinsèque. La conversion du graphe et le calcul de nouvelles fenêtres sont très pertinents dans ce cas. La visualisation sur les deux fenêtres extrinsèques confirme l'intérêt du temps intrinsèque pour calculer les communautés stables sur des fenêtres de temps.

## 4.5 Conclusion

Dans ce chapitre, nous avons proposé une méthode efficace pour détecter les communautés stables dans un réseau dynamique en étendant un algorithme existant avec la notion de temps intrinsèque. Nous avons pour cela converti le temps dans notre jeu de données d'une manière non linéaire avant d'appliquer l'algorithme. Avec cette méthode, nous obtenons des résultats prometteurs : les fenêtres de temps trouvées sont au moins aussi pertinentes que dans le cas extrinsèque, et très souvent avec un niveau de détail plus adapté. De plus, nous identifions ces fenêtres sans avoir besoin de paramétrer l'algorithme de détection de communautés. L'utilisation du temps intrinsèque est donc dans ce cas très simple et très enrichissante.

Nous avons également proposé une méthode facile et efficace pour comparer les résultats obtenus avec les deux échelles de temps, en représentant le temps intrinsèque en fonction du temps extrinsèque. Cela permet à la fois d'observer l'impact du changement d'échelle et de comparer les fenêtres de temps obtenues dans les deux cas. De plus, cette approche permet de lister de manière exhaustive toutes les configurations observables et leur interprétation.

Nous avons ensuite utilisé une version de l'algorithme de détection de communautés stables plus complexe, en autorisant le regroupement de pas de temps non adjacents. En comparant les résultats obtenus avec cette version en temps extrinsèque à ceux qui utilisent la version simple de l'algorithme en temps intrinsèque, nous constatons que les fenêtres de temps trouvées avec l'échelle de temps intrinsèque restent plus pertinentes. Cela confirme donc la simplicité et l'intérêt de l'échelle de temps intrinsèque.

Enfin, nous avons visualisé notre graphe en utilisant les deux échelles de temps (intrinsèque et extrinsèque) afin de valider l'intérêt des fenêtres de temps trouvées en temps intrinsèque. Avec cette visualisation, nous avons observé des changements de structure très importants entre les différentes fenêtres, ce qui a validé leur pertinence.

La perspective immédiate de ces travaux est d'appliquer cette méthodologie à d'autres jeux de données. Ceux-ci doivent néanmoins vérifier plusieurs conditions : étant donné que la définition du temps intrinsèque est basée sur un changement de topologie, comme l'apparition ou la disparition d'un lien par exemple, le jeu de données doit avoir des pas de temps détaillés (à la seconde près par exemple). En effet, si les pas de temps étaient plus larges (comme la journée), il y aurait beaucoup d'activité pendant un seul pas de temps, et cela impliquerait un ordonnancement artificiel de l'arrivée des liens. Par conséquent, pour être pertinent le jeu de données doit être un flot de liens. De plus, ce jeu de données doit avoir une structure en communautés (ce qui est le cas dans la plupart des réseaux complexes). L'étude de structures communautaires stables sur d'autres jeux de données pourra montrer

l'intérêt et les limites de cette approche selon les différents types de données considérés (réseaux sociaux, réseaux biologiques, etc.).

Une deuxième perspective consiste à s'intéresser plus en détail à la méthode de détection de communautés choisie. Pour tous nos calculs de structure en communautés, nous avons utilisé la méthode de Louvain qui n'est pas déterministe. Par conséquent, en relançant cet algorithme sur le même jeu de données, nous observerons des différences, même si nous avons vérifié la pertinence globale de nos résultats. Il serait donc intéressant de combiner l'approche décrite ici avec la notion de cœurs de communautés : l'idée est de calculer de nombreuses fois la structure en communautés dynamiques avec un algorithme, et de placer un nœud dans une communauté seulement s'il y apparait assez souvent afin de garantir la stabilité des communautés identifiées.

Chapitre 5

# Conclusions et perspectives

Nous nous sommes intéressés dans cette thèse à l'étude de la dynamique des graphes de terrain, c'est-à-dire à la façon dont ces graphes évoluent au cours du temps. La contribution majeure de cette thèse est de proposer et d'évaluer une méthodologie reposant sur la notion de temps intrinsèque pour l'étude des phénomènes de diffusion et du calcul de communautés dans des graphes dynamiques. Cette notion de temps est basée sur l'évolution du graphe et est indépendante de l'écoulement classique du temps (seconde, heure, etc.). Elle est donc intrinsèque au graphe étudié.

Nous avons montré que l'utilisation d'une échelle de temps intrinsèque permet de mieux comprendre le comportement d'une diffusion dans un graphe dynamique. Plus précisément, une diffusion simulée en temps intrinsèque devient indépendante de la mesure utilisée pour la définition du temps intrinsèque, et nous pouvons alors distinguer ce qui est lié à la dynamique du graphe de ce qui est lié au processus de diffusion lui-même. Nous avons validé notre méthodologie sur trois jeux de données issus de contextes variés. Enfin, nous avons vu que la simulation d'un phénomène de diffusion en temps intrinsèque sur un graphe dynamique permet de valider ou d'invalider l'unité de temps choisie pour la définition du temps intrinsèque (nouveau pas de temps à chaque création de lien, nœud, etc.).

Nous avons également montré que le temps intrinsèque permet de détecter des communautés plus détaillées et plus pertinentes dans un graphe dynamique que le temps extrinsèque. Nous avons appliqué notre méthodologie à un algorithme de détection de communautés dynamiques existant sur un réseau de contacts, et nous avons obtenu de bons résultats. Enfin, nous avons proposé une méthodologie pour comparer les communautés obtenues selon les deux échelles de temps.

Ces travaux ouvrent de nombreuses perspectives de recherche. Nous avons présenté, à la fin des chapitres 3 et 4, les perspectives immédiates découlant directement de nos travaux sur la diffusion et les communautés respectivement. Nous présentons ici les pistes de recherche plus générales soulevées par nos travaux.

La première perspective consiste à approfondir la définition du temps intrinsèque. Dans cette thèse, nous avons associé l'apparition ou la disparition d'un lien à un nouveau pas de temps intrinsèque. Nous avons vu que cette définition est pertinente sur certains jeux de données, mais inadéquate dans d'autres cas. Par exemple, dans un graphe peu dense contenant beaucoup de triangles (c'est-à-dire

trois nœuds tous liés entre eux), le nombre de liens peut ne pas suffire à caractériser correctement les évolutions de la topologie. En effet dans ce cas, la création ou la disparition d'un lien n'est pas entièrement représentative de l'évolution du graphe. Ce lien a beaucoup de chances de faire partie d'un triangle, et donc d'impacter de manière significative la diffusion au sein du graphe ou sa structure en communautés (en changeant le coefficient de clustering notamment). En revanche dans un graphe aléatoire sans structure en communautés, l'évolution des liens semble plus représentative de l'évolution de la topologie.

Il serait donc intéressant de développer une méthodologie pour pouvoir capturer, pour chaque graphe dynamique, la définition du temps intrinsèque qui reflète correctement la dynamique de ce graphe. Afin de déterminer si une définition de temps intrinsèque est pertinente, une stratégie possible est de l'utiliser pour observer une diffusion, ou d'étudier la structure en communautés dynamiques. Nous avons en effet montré dans cette thèse qu'une échelle de temps intrinsèque pertinente mène à des observations très différentes de celles obtenues en temps extrinsèque. Si l'on observe des résultats similaires dans les deux échelles, cela signifie que la définition du temps intrinsèque utilisée n'est pas pertinente pour représenter la dynamique du graphe étudié, ou que le phénomène est correctement décrit en temps extrinsèque. Cette méthodologie peut être mise au point en utilisant dans un premier temps des graphes aléatoires dynamiques de différents types (avec une distribution de degrés homogène ou hétérogène, une structure communautaire présente ou absente, etc.). Dans chaque cas, on pourra alors déterminer avec précision la définition du temps intrinsèque à utiliser.

Un autre champ d'application du temps intrinsèque est la visualisation d'un graphe dynamique. Ce sujet constitue un problème ardu, car si le graphe est gros et que la durée d'observation est élevée, la visualisation devient trop complexe pour être exploitable : on ne sait plus où regarder ni quels intervalles de temps considérer pour analyser la dynamique. Dans ce contexte, l'utilisation du temps intrinsèque peut permettre de simplifier l'exploitation de la visualisation, en raccourcissant les phases stables dans la dynamique, et en étirant les phases où le graphe évolue très rapidement. On peut alors se concentrer sur les phases très dynamiques, qui sont en général les plus complexes à visualiser.

De plus, si l'on est intéressé uniquement par l'analyse d'un phénomène précis, comme par exemple l'arrivée de nouveaux utilisateurs dans un réseau social, ou bien l'apparition d'une nouvelle personne contaminée dans le cas de la diffusion d'une maladie, on peut alors choisir une définition du temps intrinsèque qui mesure précisément ce phénomène et "ignore" les autres changements. Dans les deux exemples cités ici, on peut choisir comme définition d'une unité de temps intrinsèque l'apparition d'un nœud et l'apparition d'un nœud contaminé respectivement.

Une troisième piste de recherche consiste à utiliser le temps intrinsèque pour la détection d'événements. Des travaux ont déjà utilisé le temps intrinsèque pour

observer l'évolution du nombre de nœuds et de liens d'un graphe, et mettre en évidence des événements non observés en temps extrinsèque. Une autre possibilité pour chercher des événements dans la dynamique d'un graphe est d'étudier le comportement d'un processus dynamique sur ce graphe en utilisant des définitions différentes du temps intrinsèque. Par exemple, on peut observer une diffusion sur le graphe dynamique, en définissant le temps intrinsèque comme dépendant de l'apparition d'un nœud, ou bien de l'apparition et de la disparition d'un nœud, ou encore de l'apparition d'un triangle, etc. On peut même coupler ces définitions : par exemple, créer un pas de temps intrinsèque à chaque fois qu'un nœud ou un lien apparaît.

On peut ensuite étudier le phénomène dynamique considéré selon ces multiples échelles de temps, et comparer les résultats obtenus, comme nous l'avons fait dans l'étude de la diffusion. Ainsi, les événements dans le graphe vont ressortir naturellement, et uniquement à certaines échelles de temps, ce qui permettra de plus de déterminer plus précisément la nature de l'événement (par exemple, un événement lié à la création de liens).

De manière générale, le temps intrinsèque peut être utilisé pour l'observation de n'importe quel processus sur les graphes dynamiques : nous avons montré son intérêt pour les phénomènes de diffusion et pour le calcul de communautés de nœuds stables dans un réseau dynamique. Par exemple, le temps intrinsèque peut être utilisé de la même manière que nous l'avons fait dans cette thèse pour la détection de communautés de liens dynamiques (les communautés de liens sont définies de la même façon que les communautés de nœuds, c'est-à-dire des groupes de liens denses, et peu connectés avec le reste du réseau). La notion de temps intrinsèque est pertinente pour analyser des phénomènes dynamiques variés comme la dynamique d'Internet, la dynamique d'un automate cellulaire, et tout système dynamique en général.

# Annexes

Annexe A

# Diffusion de fichiers dans un réseau pair-à-pair

## Sommaire



## A.1 Introduction

Parmi les nombreux travaux sur la diffusion dans la littérature, quelques-uns se sont intéressé plus spécifiquement à la diffusion de fichiers sur les réseaux pair-à-pair. Ainsi, [Han et al., 2004] proposent un modèle pour optimiser la diffusion d'un fichier sur un réseau pair-à-pair. Ils ont donc naturellement utilisé un modèle avec un comportement exponentiel. Dans [Leibnitz et al., 2006], les auteurs créent un modèle de diffusion en tenant compte des détails du protocole pair-à-pair utilisé (notamment pour la découpe des fichiers), et le modèle proposé montre une croissance exponentielle du nombre de nœuds infectés.

Toutefois, l'observation d'une diffusion réelle de fichiers sur un réseau pair-à-pair n'a été que très peu étudiée. Les travaux de [Bernardes et al., 2012] s'intéressent à la diffusion de fichiers, sur le même jeu de données que nous utilisons dans ce chapitre [1], en le considérant toutefois de façon statique. Ils observent le comportement d'un modèle SIR et montrent que ce celui-ci ne permet pas de reproduire

1. décrit dans la section A.2.1

des cascades de diffusion avec un nombre de nœuds et une profondeur proches de celles observées.

Dans ce chapitre, nous étudions la diffusion réelle sur un jeu de données pair-à-pair, en prenant en compte la dynamique du graphe, et nous proposons un modèle pour reproduire le comportement de la diffusion observée.

## A.2 Jeu de données et diffusion observée

### A.2.1 Données utilisées

Le jeu de données sur lequel nous étudions la diffusion est issu d'un réseau pair-à-pair décrit par [Guillaume et al., 2004]. Le protocole eDonkey est un protocole de pair-à-pair semi centralisé : les pairs clients effectuent des requêtes pour des fichiers, et le serveur envoie ensuite en réponse au client un ensemble de sources pour un fichier donné. Ce sont ces réponses du serveur qui ont été enregistrées. Le client se charge ensuite de récupérer le fichier souhaité auprès des sources que le serveur lui a fourni, ce qui n'est pas visible dans les traces puisque le serveur n'intervient plus à ce stade.

Le jeu de données nous fournit de plus des informations de session sur les nœuds, indiquant les périodes de connexion de chaque client au serveur eDonkey. Ces informations permettent de savoir, à chaque instant, quels sont les nœuds présents dans le réseau, et de connaître ainsi la dynamique du réseau. La mesure a duré 48 heures, pendant lesquelles le serveur a reçu 210 millions de requêtes, et géré environ 1,5 millions de connexions et déconnexions.

Nous avons également utilisé un second jeu de données issu d'un autre réseau pair-à-pair utilisant le protocole eDonkey. Ce jeu de données est enregistré sur le même principe que le précédent, mais dure 10 semaines.

### A.2.2 Graphe d'intérêt

Afin de pouvoir étudier les diffusions sur ce jeu de données, nous devons tout d'abord construire le graphe sur lequel elle se produit, et déterminer les voisins des nœuds. Pour cela, nous construisons ce que nous appelons le *graphe d'intérêt* : quand un client reçoit une réponse du serveur, celui-ci lui indique des sources potentielles. Le client et les sources ont alors un point commun : l'intérêt pour le fichier concerné (respectivement en tant que demandeur ou fournisseur). Nous lions le client ainsi que les sources citées dans le graphe d'intérêt, puisqu'ils ont un intérêt commun pour un fichier. Les informations de sessions sur les nœuds rendent ce graphe dynamique : à un instant $t$, on ne considère dans le graphe d'intérêt que les nœuds qui sont effectivement connectés au serveur.

| temps | client | fichier | source 1 | source 2 |
|---|---|---|---|---|
| t1 | 1 | fichier1 | 2 | 4 |
| t2 | 2 | fichier2 | 3 | 5 |
| t3 | 4 | fichier3 | 5 | |

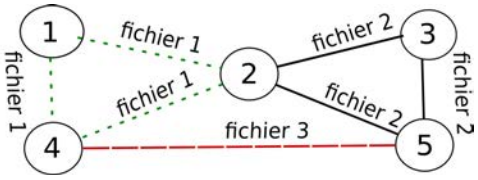



**Figure A.1** – *Construction du graphe d'intérêt à partir de 3 réponses du serveur avec les sources potentielles.*

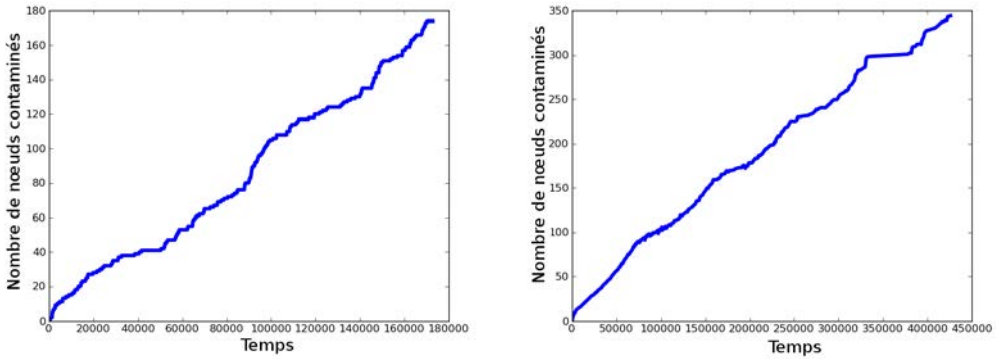



**Figure A.2** – *Diffusion observée sur les deux jeux de données. À gauche, jeu de données court. À droite, jeu de données long.*

Un exemple de construction du graphe d'intérêt à partir de réponses du serveur est donné dans la figure A.1 : à l'instant $t_1$, le serveur indique au pair client 1 les pairs sources 2 et 4 pour le fichier 1. Les pairs 1, 2 et 4 sont reliés deux à deux dans le graphe d'intérêt. Ensuite, à l'instant $t_2$, le serveur indique au client 2 les sources 3 et 5. Les nœuds 2, 3 et 5 seront donc reliés deux à deux dans le graphe d'intérêt. On utilise le même processus à l'instant $t_3$, et on obtient alors le graphe montré dans la figure A.1. Notons qu'un pair peut être demandeur (client) de certains fichiers, et fournisseur (source) d'autres. C'est le cas du pair 2, fournisseur du fichier 1 et demandeur du fichier 2.

### A.2.3 Diffusion observée

Le jeu de données utilisé nous permet d'observer la diffusion des fichiers. Pour obtenir le comportement moyen de la diffusion d'un fichier, nous sélectionnons au hasard un échantillon de mille fichiers sur l'ensemble des fichiers. Nous regardons ensuite la diffusion de chaque fichier, c'est-à-dire l'évolution du nombre de nœuds possédant un fichier donné au cours du temps, et nous calculons une moyenne sur la diffusion. Nous constatons (figure A.2 à gauche) que le comportement de cette diffusion observée moyenne évolue de manière linéaire.

Afin de valider ce comportement, nous utilisons le même procédé sur le jeu de données durant dix semaines. Nous observons que le comportement moyen de la diffusion réelle a aussi une allure linéaire (figure A.2 à droite).

En observant le comportement des nœuds dans notre jeu de données, nous constatons que la majorité des clients ne partagent pas ensuite le fichier qu'ils viennent de télécharger, c'est-à-dire qu'ils ne deviennent jamais fournisseurs. La proportion des nœuds clients qui deviennent fournisseurs est de l'ordre de 8% (sur les deux jeux de données). Dans les travaux qui ont été réalisés sur la diffusion sur un réseau pair-à-pair, les diffusions simulées ont un comportement exponentiel, comme nous l'avons expliqué dans l'introduction de ce chapitre.

Dans le cas étudié ici, nous souhaitons avoir un modèle cohérent avec le comportement de la diffusion observée, qui est globalement linéaire. Dans la suite, nous testons tout d'abord un modèle simple inspiré par le modèle SI, puis nous proposons une amélioration de ce modèle, plus proche de la diffusion observée.

## A.3 Modélisation

### A.3.1 Premier modèle

Nous testons tout d'abord sur notre jeu de données un modèle de diffusion inspiré du modèle SI. Les nœuds peuvent être dans un des deux états suivants : sain ou infecté. Dans le contexte de la diffusion d'un fichier, un nœud est sain lorsqu'il ne possède pas le fichier et il est infecté s'il le possède. Lorsqu'un nœud sain est voisin d'un nœud infecté, il a une probabilité $p$ d'être infecté à son tour. En revanche, et c'est ici que notre modèle est différent du modèle SI classique, lorsqu'un nœud sain a plusieurs voisins contaminés, il n'a pas plus de chance d'être contaminé à son tour : on ne fait qu'une seule fois par nœud le test de contamination à un instant $t$, toujours avec la même probabilité $p$.

L'idée sous-jacente est que dans le cas de la diffusion de fichiers, la contamination n'est pas passive : un nœud dans l'état S qui veut un fichier va lui-même le chercher et le télécharger. Ce n'est pas le nœud dans l'état I qui contamine un nœud passif dans l'état S.

Les résultats d'une simulation de diffusion avec ce modèle sont montrés sur la figure A.3 (figure de gauche, courbe du bas), comparés à la diffusion observée (courbe du haut). Pour cette simulation, la probabilité $p$ est égale à $\frac{1}{245000}$. Nous avons choisi dans un premier temps ce paramètre de façon empirique afin d'obtenir un nombre de nœuds contaminés à la fin de la simulation proche de celui de la diffusion observée. Par la suite, nous avons vérifié cette valeur en mesurant la distance entre les courbes et en choisissant le paramètre minimisant la distance.

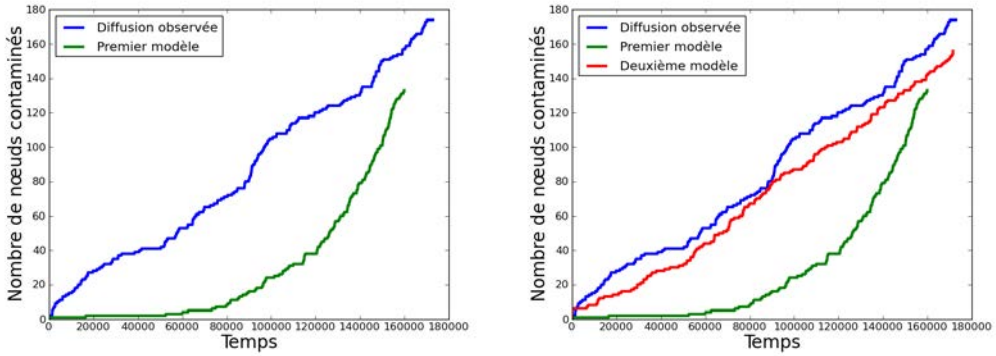



**Figure A.3** – *À gauche : diffusion observée (en haut) et simulation du premier modèle (en bas). À droite : diffusion observée, simulation du premier modèle et simulation du deuxième modèle (au milieu).*

Nous observons sur notre simulation que, contrairement à la diffusion observée, l'allure de la diffusion est ici exponentielle. Ce résultat n'est pas surprenant compte-tenu du modèle utilisé : plus il y a de nœuds contaminés dans le réseau, plus le nombre de nœuds en contact de nœuds contaminés augmente. Ainsi, la croissance du nombre de nœuds contaminés est exponentielle.

### A.3.2 Deuxième modèle

Nous cherchons alors à expliquer cette différence de comportement entre la diffusion réelle et la diffusion simulée avec le modèle précédent. Le fait que la majorité des clients ne deviennent pas fournisseurs est à prendre en compte : en effet, cela ralentit de façon significative la diffusion.

Nous avons donc adapté le modèle proposé ci-dessus pour tenir compte de ce paramètre. Nous introduisons dans le modèle un nouveau paramètre : la probabilité qu'un nœud contaminé soit contagieux. Il y a donc deux cas possibles lorsqu'un nœud est contaminé : soit il devient contagieux avec une probabilité $q$, soit il ne l'est pas et ne peut pas dans ce cas transmettre à son tour le fichier.

Ce modèle présente des différences importantes avec le modèle SIR, où les nœuds à l'état I ont une chance de "disparaître" et de passer à l'état R. Ils ne sont alors plus comptabilisés comme des nœuds contaminés. Dans le modèle que nous proposons ici, les nœuds qui sont contaminés et non contagieux sont tout de même comptabilisés dans les nœuds infectés. De plus, dans le modèle SIR, un nœud à l'état I peut à chaque instant passer à l'état R avec une certaine probabilité. Dans notre modèle, les nœuds bloquants, c'est-à-dire contaminés mais pas contagieux, restent dans cet état jusqu'à la fin. Ce nouveau modèle permet de décrire le fait que dans le réseau pair-à-pair que nous étudions, la plupart des nœuds qui possèdent un fichier, donc à l'état I, ne deviennent pas source et donc ne sont pas contagieux.

Lorsque l'on analyse les données, on constate que la proportion de clients qui deviennent fournisseurs est de 8%. Nous simulons donc une diffusion avec ce nouveau modèle, en prenant comme probabilité qu'un nœud contaminé soit contagieux $q = 0.08$. Les résultats de cette simulation sont montrés sur la figure A.3 (figure de droite), pour une probabilité de contamination $p = \frac{1}{30000}$. Pour déterminer ce dernier paramètre, nous avons testé comme précédemment différentes valeurs de $p$, tout en gardant la probabilité $q$ constante, et nous avons choisi la valeur de $p$ pour laquelle la distance entre la courbe de diffusion observée et la simulation est la plus faible.

Nous observons cette fois-ci que la diffusion simulée suit un comportement qui se rapproche beaucoup du comportement de la diffusion réelle. Ce modèle semble donc approprié pour modéliser le type de diffusion que nous observons ici.

### A.3.3 Interprétation du nouveau modèle

Nous cherchons à expliquer comment ce modèle, pourtant inspiré du modèle SI qui produit des courbes généralement exponentielles, donne ici une croissance linéaire.

Le premier modèle observé présente un comportement exponentiel, qui s'explique facilement : lorsqu'un nœud est contaminé, tous ses voisins ont un risque d'être contaminés à l'instant $t$. Ensuite, à l'instant $t + 1$, les nœuds contaminés à l'étape $t$ diffusent également la maladie, et le nombre de contaminés augmente ainsi de plus en plus vite. En revanche, l'aspect linéaire du deuxième modèle est a priori étonnant. Pourquoi l'introduction de nœuds bloquants (les nœuds contaminés mais pas contagieux) change-t-elle radicalement l'allure de la diffusion ?

#### A.3.3.1 Influence de la probabilité d'être contagieux

En premier lieu, nous cherchons à observer l'influence de la probabilité $q$ d'être contagieux quand un nœud est contaminé. En effet, si $q = 1$, c'est-à-dire qu'un nœud contaminé est systématiquement contagieux, alors le deuxième modèle devient identique au premier. Pour déterminer l'impact de ce paramètre, nous avons donc effectué plusieurs simulations de diffusion en le faisant varier.

Les résultats de ces simulations sont montrés sur la figure A.4. Les paramètres utilisés pour ces différentes simulations correspondent à des probabilités de devenir contagieux égales à $0.01$, $0,08$ $0.25$, $0.5$, $0.75$ et $0.9$. Nous observons que lorsque cette probabilité est très faible, le comportement de la diffusion semble linéaire, comme ce que nous avons observé dans le cas du modèle 2 avec un paramètre de $0.08$. En revanche, quand la probabilité augmente, la diffusion reprend un caractère exponentiel : au début de la simulation, la croissance du nombre de nœuds contaminés est lente, et à partir d'un certain seuil, elle accélère brusquement. Lorsque la probabilité est très élevée, ce modèle se rapproche alors très fortement du premier modèle.

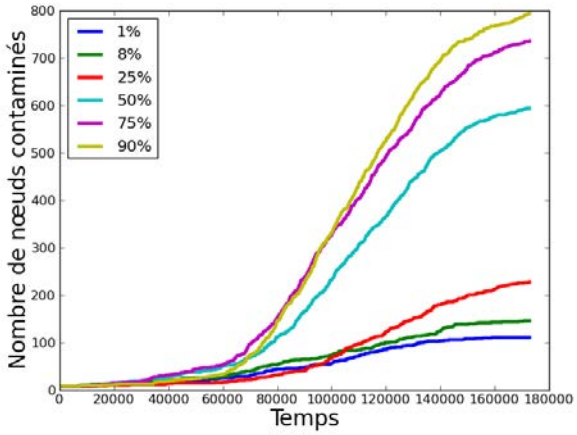



**Figure A.4** – *Simulation de diffusion avec le modèle 2 en faisant varier la probabilité q d'être contagieux.*

### A.3.3.2 Influence de la contamination active

La contamination sur un réseau pair-à-pair est de type "actif", c'est-à-dire que ce sont les nœuds sains qui vont chercher un fichier et le télécharger. Dans nos modèles, nous représentons ce phénomène en gardant la même probabilité de contamination pour un nœud, indépendamment du nombre de voisins contaminés qu'il possède.

Ainsi, si l'on compare notre premier modèle à un modèle SI classique, la diffusion, même si elle est exponentielle, est plus lente puisque la probabilité de contamination est invariante.

### A.3.3.3 Influence de la structure du graphe

Nous nous sommes intéressés, à chaque instant, au nombre de nœuds potentiellement "contaminables", c'est-à-dire à l'ensemble des nœuds voisins de nœuds contaminés. Nous avons constaté que ce nombre est quasiment égal au nombre de nœuds présents dans le graphe à chaque instant. Le graphe que nous étudions est très dense, un nœud contaminé peut donc atteindre très rapidement l'ensemble du réseau. Ainsi, dans le cas de réseaux pair-à-pair eDonkey, la diffusion est en quelque sorte indépendante de la structure du graphe, puisque l'ensemble des nœuds peuvent être atteints à chaque instant. Seul l'aspect dynamique du graphe influe sur le comportement de la diffusion, en modifiant l'ensemble des nœuds présents à chaque instant (et donc potentiellement "contaminables").

Nous avons vu dans l'ensemble des travaux de cette thèse que la dynamique est un paramètre extrêmement important à prendre en compte, et qu'elle influe de façon très significative sur le comportement de la diffusion.

## A.4 Conclusion

Dans ce chapitre, nous avons observé un phénomène de diffusion de fichiers sur un réseau pair-à-pair, et nous avons pour cela dû faire des hypothèses, comme par exemple, décider de la façon de construire le graphe d'intérêt. La diffusion est paramétrée de telle façon que nous observons a un comportement linéaire, alors que dans les travaux déjà effectués dans le domaine, les diffusions traitées sont de type exponentiel.

Nous avons proposé un modèle, inspiré du modèle SI, qui présente un comportement linéaire grâce à l'introduction de nœuds contaminés mais pas contagieux, et qui permet d'approximer correctement la diffusion réelle. Nous avons également expliqué pourquoi le modèle obtenu convient, et nous avons montré son comportement en faisant varier ses paramètres. Nous avons de plus mis en évidence l'impact très important de la dynamique du graphe sur la diffusion.

Annexe B

# Étude d'un réseau dynamique de l'activité cérébrale

## Sommaire



## B.1 Introduction

Nous nous intéressons dans cette annexe à un autre type de réseau complexe : les réseaux biologiques. Nous étudions en particulier un graphe modélisant l'activité cérébrale. Les détails sur la façon dont le jeu de données a été obtenu sont présentés par [Doucet et al., 2011]. L'activité cérébrale de quelques personnes au repos a été observée pendant huit minutes en mesurant le taux d'oxygène par IRM dans différentes zones du cerveau. Ce réseau a déjà été étudié par des biologistes et nous nous intéressons ici à une étude de ce réseau sous l'angle d'un graphe dynamique. [Hutchison et al., 2013] présentent un état de l'art sur les travaux existants dans ce domaine. Nous n'utilisons pas dans cette annexe la notion de temps intrinsèque, car la dynamique du graphe que nous étudions est très particulière, comme nous l'expliquons dans la section suivante.

## B.2 Description du jeu de données

Le graphe représentant l'activité cérébrale considérée est un graphe dynamique complet pondéré, constitué de 384 nœuds. Il est mesuré sur 208 pas de temps. À chaque pas de temps, la totalité des liens est présente. La raison pour laquelle ce

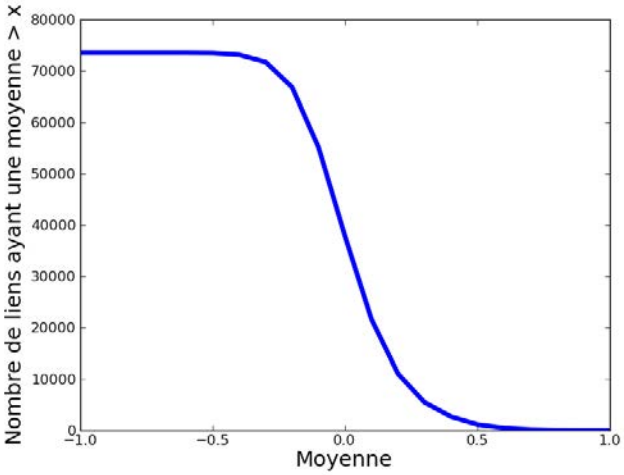



**Figure B.1** – *Distribution cumulée de la moyenne des poids des liens.*

graphe est dynamique est que les poids des liens varient au cours du temps (entre -1 et 1) et représentent la co-activation des zones concernées. Si le poids entre deux zones est positif, alors celles-ci sont activées simultanément. Sinon, le poids est négatif.

Nous avons calculé la moyenne des poids au cours du temps pour chaque lien et représentons la distribution cumulée sur la figure B.1. Sur cette courbe, on indique pour chaque valeur $x$ de poids moyen (compris entre -1 et 1), le nombre de liens dont la moyenne des poids est supérieure à $x$. Nous observons que la moité des liens environ a un poids positif en moyenne. La distribution est symétrique pour les poids positifs ou négatifs.

La variance des poids au cours du temps varie très peu pour chaque lien : elle est comprise entre 0 et 0.1 pour tous les liens. En revanche, le poids de chaque lien varie en permanence au cours des 208 pas de temps. Ainsi, pour chaque lien, le poids varie à chaque pas de temps, mais très peu.

La figure B.2 représente les deux liens pour lesquels la variation est respectivement minimale et maximale. Dans les deux cas, la somme totale des variations des poids des liens au cours du temps est très proche : 6,5 pour le maximum, contre 6,34 pour le minimum, sur les 208 pas de temps. En moyenne, le poids varie donc d'environ 0.03 par pas de temps.

Ce graphe a donc une dynamique très particulière : il est complet et la dynamique est représentée uniquement par la variation des poids des liens. Nous avons vu que ces poids varient en permanence, mais très peu, ce qui rend la dynamique assez subtile. Enfin, la signification des poids négatifs est très particulière, puisque dans ce cas, un lien entre deux nœuds ne représente pas une interaction, mais un

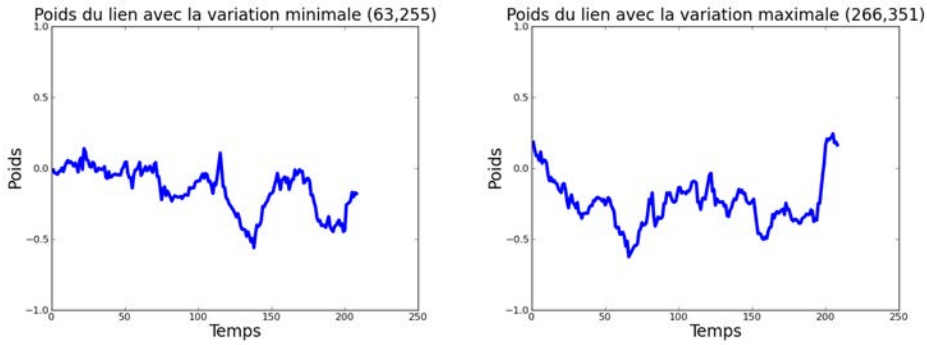



**Figure B.2** – *Évolution de la variance au cours du temps pour le lien où elle est minimale (à gauche) et le lien où elle est maximale (à droite).*

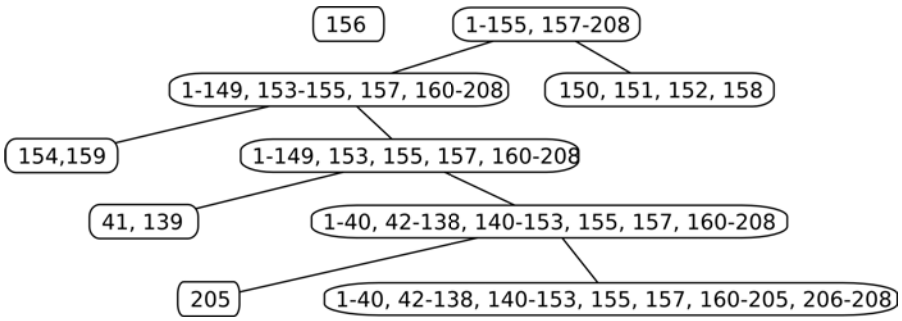



**Figure B.3** – *Arbre des fenêtres de temps.*

éloignement entre ces deux nœuds. Dans la suite, nous étudions ce graphe et sa dynamique du point de vue de sa structure en communautés.

## B.3 Détection de communautés

### B.3.1 Communautés stables au cours du temps

Nous utilisons l'algorithme 1 (en version non-adjacente) de [Aynaud and Guillaume, 2011] détaillé dans le chapitre 4 afin de détecter des communautés stables au cours du temps dans le graphe étudié. Comme cet algorithme utilise la méthode de Louvain, il ne gère pas les poids négatifs sur les liens. Nous avons donc effectué un filtrage pour conserver uniquement les liens avec un poids positif. L'arbre obtenu en sortie de l'algorithme appliqué à notre graphe est représenté sur la figure B.3.

Nous voyons sur cette figure qu'au plus haut niveau de la hiérarchie, l'algorithme regroupe la totalité des pas de temps dans la même fenêtre, à l'exception

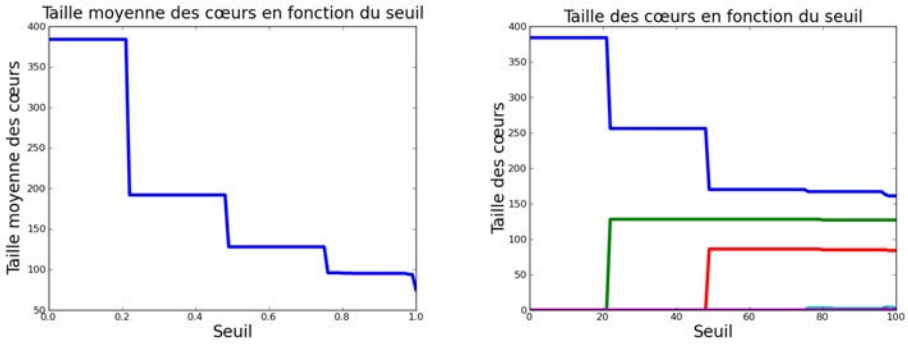



**Figure B.4** – *Taille moyenne des cœurs en fonction du seuil(à gauche). Taille des cœurs en fonction du seuil (à droite).*

de l'instant 156. Dans les niveaux inférieurs de l'arbre, quelques pas de temps supplémentaires sont exclus, mais la majorité des pas de temps restent groupés dans la même fenêtre. Cela signifie que l'algorithme détecte une structure en communautés stables sur la quasi-totalité de la mesure. Comme nous avons vu que le poids des liens varie très peu au cours du temps, cela confirme le fait que la dynamique est très faible sur ce graphe. Dans la section suivante, nous considérons donc le graphe statique à un instant donné et nous cherchons des communautés stables sur ce graphe (malgré le non-déterminisme de la méthode de Louvain).

### B.3.2 Cœurs de communautés

Nous utilisons un algorithme de détection de communautés basé sur la méthode des cœurs, décrite par [Seifi et al., 2013]. Le principe est de calculer plusieurs fois les communautés avec un algorithme de détection de communautés (nous utilisons ici la méthode de Louvain). Pour chaque paire de nœuds, on compte le nombre de fois où ils apparaissent dans la même communauté. Un cœur contient alors les nœuds qui ont été dans la même communauté suffisamment de fois (selon un seuil à fixer).

La figure B.4 (à gauche) représente la taille moyenne des cœurs en fonction du seuil choisi. Jusqu'à un seuil de 20% de présence dans la même communauté, on observe un seul cœur contenant tous les nœuds. Ensuite, certaines valeurs de seuils entrainent des paliers sur la taille moyenne des cœurs : 45%, 75%, et 95% (environ).

La figure B.4 (à droite) montre pour la taille des cœurs en fonction de la valeur du seuil. Il n'y a jamais plus de 5 cœurs dans le graphe. À partir d'un seuil de 45%, on observe trois grosses communautés. On a deux communautés très petites (2 et 4 nœuds), pour un seuil plus grand. Cette courbe peut donc nous aider à fixer la valeur du seuil.

## B.4 Perspective : modularité et poids négatifs

Nous avons étudié dans ce chapitre un graphe avec une dynamique très particulière consistant en une faible variation des poids des liens sur un graphe complet. De plus, ce graphe contient des poids négatifs, qui indiquent à quel point les zones cérébrales associées ne fonctionnent pas en même temps. Dans nos expériences de détection de communautés, nous nous sommes restreints aux liens avec un poids positif, car la modularité n'est pas conçue pour gérer les poids négatifs. Or ce filtrage a supprimé environ la moitié des liens, dont le poids négatif a une signification particulière car cela signifie que deux nœuds se repoussent. Afin de calculer des communautés en intégrant ces poids négatifs, il faut donc définir une nouvelle modularité. L'idée est que des liens avec un poids négatif entre deux nœuds d'une communauté doivent faire baisser la modularité. Au contraire, des liens avec un poids négatif entre des nœuds de deux communautés différentes doivent augmenter la modularité.

[Traag and Bruggeman, 2009] proposent une définition pour une fonction de qualité qui intègre les poids négatifs comme nous venons de l'expliquer. Cette fonction de qualité $Q$, est définie en deux parties, une fonction de qualité $Q^+$ pour les poids positifs et une fonction de qualité $Q^-$ pour les poids négatifs. On a :

$$Q^+(\pi) = \sum_{i,j} \left( A^+_{ij} - \frac{k^+_i k^+_j}{2m^+} \right) \delta(c_i, c_j)$$

et

$$Q^-(\pi) = \sum_{i,j} \left( A^-_{ij} - \frac{k^-_i k^-_j}{2m^-} \right) \lambda(c_i, c_j)$$

avec $k^+_i$ la somme des poids positifs liés au sommet $i$, $A_{ij}$ le poids du lien $(i, j)$ s'il est positif et 0 sinon, $m^+ = \frac{1}{2} \sum_{i,j} A^+_{ij}$, $c_i$ la communauté du nœud $i$, $\delta(c_i, c_j)$ égal à 1 si $c_i = c_j$ et 0 sinon, et $\lambda(c_i, c_j)$ égal à 1 si $c_i \neq c_j$ et -1 sinon (pour tous les symboles avec le signe -, on a la même définition en remplaçant positif par négatif).

La modularité globale est définie de la façon suivante :

$$Q(\pi) = \frac{H^+(\pi) + H^-(\pi)}{2}$$

La partie positive de la modularité ajoute donc un bonus si un nœud a plus de liens avec un poids positif vers l'intérieur de sa communauté que vers l'extérieur. La partie négative en revanche, permet de pénaliser chaque lien au poids négatif contenu dans la communauté, et d'ajouter un bonus pour chaque lien au poids

négatif sortant de la communauté. C'est la fonction $\lambda$ qui traduit ce système de bonus et de malus.

Ces travaux sur l'analyse d'un réseau biologique peuvent donc être poursuivis en remplaçant la modularité utilisée dans la méthode de Louvain par cette fonction de qualité et ainsi pouvoir gérer les poids négatifs. Cela permettra de prendre en compte la totalité des liens du graphe et de tenir compte de la répulsion entre les nœuds (qui peut affecter beaucoup la structure en communautés). De nombreux travaux devraient donc être menés sur ce thème.

# Bibliographie


[Adar and Adamic, 2005] Adar, E. and Adamic, L. A. (2005). Tracking information epidemics in blogspace. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 207–214. (Cité en page 44.)

[Albano, 2011] Albano, A. (2011). Phénomènes de diffusion dans les réseaux dynamiques : simulation et modélisation. In *Proceedings of Journée Thématique de Fouille de Grands Graphes (JFGG), in conjunction with Modèles et Analyse des Réseaux : Approches Mathématiques et Informatique (MARAMI)*. (Cité en page 14.)

[Albano et al., 2013] Albano, A., Guillaume, J.-L., Heymann, S., and Le Grand, B. (2013). A matter of time - intrinsic or extrinsic - for diffusion in evolving complex networks. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. (Cité en page 14.)

[Albano et al., 2014a] Albano, A., Guillaume, J.-L., Heymann, S., and Le Grand, B. (2014a). Studying graph dynamics through intrinsic time based diffusion analysis. In *Applications of Social Media and Social Network Analysis*. (Cité en page 14.)

[Albano et al., 2012] Albano, A., Guillaume, J.-L., and Le Grand, B. (2012). File diffusion in a dynamic peer-to-peer network. In *Proceedings of the Mining Social Network Dynamics workshop (MSND), in conjunction with the World Wide Web conference (WWW)*, pages 1169–1172. (Cité en page 14.)

[Albano et al., 2014b] Albano, A., Guillaume, J.-L., and Le Grand, B. (2014b). On the use of intrinsic time scale for dynamic community detection in social networks. In *Proceedings of the 8th IEEE International Conference on Research Challenges in Information Science (RCIS)*. (Cité en page 14.)

[Asur et al., 2007] Asur, S., Parthasarathy, S., and Ucar, D. (2007). An event-based framework for characterizing the evolutionary behavior of interaction graphs. In *Proceedings of the 13th international conference on Knowledge discovery and data mining (KDD)*, page 921. (Cité en page 67.)

[Aynaud and Guillaume, 2011] Aynaud, T. and Guillaume, J.-L. (2011). Multi-step community detection and hierarchical time segmentation in evolving networks. In *Proceedings of the 5th SNA-KDD workshop*. (Cité en pages 69 et 103.)

[Aynaud et al., 2012] Aynaud, T., Guillaume, J.-L., Fleury, E., and Wang, Q. (2012). *Dynamics On and Of Complex Networks*, volume 2, chapter Communities in evolving networks : definitions, detection and analysis techniques, pages 159–200. (Cité en page 66.)

[Bailey et al., 1975] Bailey, N. T. et al. (1975). *The mathematical theory of infectious diseases and its applications*. Charles Griffin & Company Ltd. (Cité en page 41.)

[Bajardi et al., 2011] Bajardi, P., Barrat, A., Natale, F., Savini, L., and Colizza, V. (2011). Dynamical patterns of cattle trade movements. *PloS one*, 6. (Cité en page 19.)

[Bakshy et al., 2012] Bakshy, E., Rosenn, I., Marlow, C., and Adamic, L. (2012). The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web*, pages 519–528. (Cité en page 46.)

[Barabási and Albert, 1999] Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439) :509–512. (Cité en page 31.)

[Barthelemy et al., 2005] Barthelemy, M., Barrat, A., Pastor-Satorras, R., and Vespignani, A. (2005). Dynamical patterns of epidemic outbreaks in complex heterogeneous networks. *Journal of theoretical biology*, 235(2) :275–288. (Cité en page 43.)

[Bass, 1969] Bass, F. (1969). A new product growth for model consumer durables. *Management Science*, 15 :215–227. (Cité en page 42.)

[Beiró and Busch, 2010] Beiró, M. and Busch, J. (2010). Visualizing communities in dynamic networks. In *Latin American Workshop on Dynamic Networks*, volume 1. (Cité en page 67.)

[Benamara and Magnien, 2010] Benamara, L. and Magnien, C. (2010). Estimating properties in dynamic systems : the case of churn in p2p networks. In *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*, pages 1–6. (Cité en page 17.)

[Bernardes et al., 2012] Bernardes, D. F., Latapy, M., and Tarissan, F. (2012). Relevance of sir model for real-world spreading phenomena : Experiments on a large-scale P2P system. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 327–334. (Cité en page 93.)

[Blondel et al., 2008] Blondel, V., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics : Theory and Experiment*. (Cité en pages 68 et 72.)

[Blonder et al., 2012] Blonder, B., Wey, T. W., Dornhaus, A., James, R., and Sih, A. (2012). Temporal dynamics and network analysis. *Methods in Ecology and Evolution*, 3 :958–972. (Cité en pages 17 et 24.)

[Cardillo et al., 2013] Cardillo, A., Petri, G., Nicosia, V., Sinatra, R., Gómez-Gardeñes, J., and Latora, V. (2013). Evolutionary dynamics of time-resolved social interactions. *NetSci*. (Cité en page 27.)

[Casteigts et al., 2011] Casteigts, A., Flocchini, P., Quattrociocchi, W., and Santoro, N. (2011). Time-varying graphs and dynamic networks. In *Ad-hoc, Mobile, and Wireless Networks*, pages 346–359. (Cité en pages 17 et 23.)

[Céspedes and Marcotorchino, 2013] Céspedes, P. C. and Marcotorchino, J.-F. (2013). Comparing different modularization criteria using relational metric. In *Geometric Science of Information*, pages 180–187. (Cité en page 65.)

[Cha et al., 2010] Cha, M., Haddadi, H., Benevenuto, F., and Gummadi, P. K. (2010). Measuring user influence in twitter : The million follower fallacy. *ICWSM*, 10 :10–17. (Cité en page 44.)

[Chen et al., 2010] Chen, Z., Wilson, K. A., Jin, Y., Hendrix, W., and Samatova, N. F. (2010). Detecting and Tracking Community Dynamics in Evolutionary Networks. In *IEEE International Conference on Data Mining Workshops*, pages 318–327. (Cité en page 67.)

[Cheng et al., 2014] Cheng, J., Adamic, L. A., Dow, P. A., Kleinberg, J., and Leskovec, J. (2014). Can cascades be predicted ? *Proceedings of the 23rd international conference on World Wide Web*. (Cité en page 44.)

[Chi et al., 2007] Chi, Y., Zhu, S., Song, X., Tatemura, J., and Tseng, B. L. (2007). Structural and temporal analysis of the blogosphere through community factorization. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 163. (Cité en page 67.)

[Clauset and Eagle, 2007] Clauset, A. and Eagle, N. (2007). Persistence and periodicity in a dynamic proximity network. *Workshop on Computational Methods for Dynamic Interaction Networks*. (Cité en page 21.)

[Clementi et al., 2010] Clementi, A. E., Macci, C., Monti, A., Pasquale, F., and Silvestri, R. (2010). Flooding time of edge-markovian evolving graphs. *SIAM journal on discrete mathematics*, 24(4) :1694–1712. (Cité en page 24.)

[de la Convention du Mètre, 2006] de la Convention du Mètre, O. I. (2006). The international system of units (SI). Technical Report 8, Bureau International des Poids et Mesures. (Cité en page 30.)

[Delre et al., 2010] Delre, S. A., Jager, W., Bijmolt, T. H., and Janssen, M. A. (2010). Will it spread or not ? The effects of social influences and network topology on innovation diffusion. *Journal of Product Innovation Management*, 27(2) :267–282. (Cité en page 45.)

[Diday, 1971] Diday, E. (1971). Une nouvelle méthode en classification automatique et reconnaissance des formes la méthode des nuées dynamiques. *Revue de Statistique Appliquée*, 19(2) :19–33. (Cité en page 72.)

[Dinh et al., 2010] Dinh, T., Shin, I., Thai, N., and Thai, M. (2010). A General Approach for Modules Identification in Evolving Networks. *Dynamics of Information*, 40(4) :83–100. (Cité en page 69.)

[Doucet et al., 2011] Doucet, G., Naveau, M., Petit, L., Delcroix, N., Zago, L., Crivello, F., Jobard, G., Tzourio-Mazoyer, N., Mazoyer, B., Mellet, E., et al. (2011). Brain activity at rest : a multiscale hierarchical functional organization. *Journal of neurophysiology*, 105(6) :2753–2763. (Cité en page 101.)

[Eames et al., 2012] Eames, K. T., Tilston, N. L., Brooks-Pollock, E., and Edmunds, W. J. (2012). Measured dynamic social contact patterns explain the spread of H1N1v influenza. *PLoS Computational Biology*, 8(3) :e1002425. (Cité en pages 45 et 47.)

[Eckmann et al., 2004] Eckmann, J.-P., Moses, E., and Sergi, D. (2004). Entropy of dialogues creates coherent structures in e-mail traffic. *Proceedings of the National Academy of Sciences of the United States of America*, 101(40) :14333–14337. (Cité en pages 25, 28 et 30.)

[Erdos and Rényi, 1961] Erdos, P. and Rényi, A. (1961). On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5 :17–61. (Cité en page 23.)

[Falkowski et al., 2008] Falkowski, T., Barth, A., and Spiliopoulou, M. (2008). Studying community dynamics with an incremental graph mining algorithm. In *Proc. of the 14 th Americas Conference on Information Systems (AMCIS)*, pages 1–11. (Cité en page 68.)

[Falkowski and Spiliopoulou, 2007] Falkowski, T. and Spiliopoulou, M. (2007). Users in volatile communities : Studying active participation and community evolution. *Lecture Notes in Computer Science*, 4511 :47. (Cité en page 67.)

[Falkowski et al., 2006] Falkowski, T., Spiliopoulou, M., and Bartelheimer, J. (2006). Community dynamics mining. In *Proceedings of 14th European Conference on Information Systems (ECIS)*. (Cité en page 67.)

[Fall, 2003] Fall, K. (2003). A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34. (Cité en page 48.)

[Gauvin et al., 2013] Gauvin, L., Panisson, A., Cattuto, C., and Barrat, A. (2013). Activity clocks : spreading dynamics on temporal networks of human contact. *Scientific reports*, 3. (Cité en pages 30 et 48.)

[Girvan et al., 2002] Girvan, M., Callaway, D. S., Newman, M. E., and Strogatz, S. H. (2002). Simple model of epidemics with pathogen mutation. *Physical Review E*, 65(3) :031915. (Cité en page 43.)

[Goldenberg et al., 2001] Goldenberg, J., Libai, B., and Muller, E. (2001). Talk of the network : A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3) :211–223. (Cité en page 43.)

[González-Bailón et al., 2011] González-Bailón, S., Borge-Holthoefer, J., Rivero, A., and Moreno, Y. (2011). The dynamics of protest recruitment through an online network. *Scientific reports*. (Cité en page 44.)

[Görke et al., 2010] Görke, R., Maillard, P., and Staudt, C. (2010). Modularity-Driven Clustering of Dynamic Graphs. *Experimental Algorithms*, Cl(1). (Cité en page 68.)

[Greene and Doyle, 2010] Greene, D. and Doyle, D. (2010). Tracking the evolution of communities in dynamic social networks. In *Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1–13. (Cité en page 67.)

[Gruhl et al., 2004] Gruhl, D., Guha, R., Liben-Nowell, D., and Tomkins, A. (2004). Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web*, pages 491–501. (Cité en page 44.)

[Guillaume et al., 2004] Guillaume, J.-L., Latapy, M., and Le-Blond, S. (2004). Statistical analysis of a p2p query graph based on degrees and their time-evolution. *IWDC*. (Cité en page 94.)

[Han et al., 2004] Han, P., Hosanager, K., and Tan, Y.-W. (2004). Diffusion of digital products in peer-to-peer networks. In *Proceedings of the ICIS conference*. (Cité en page 93.)

[Heymann and Le Grand, 2013] Heymann, S. and Le Grand, B. (2013). Monitoring user-system interactions through graph-based intrinsic dynamics analysis. In *Proceedings of the 7th IEEE International Conference on Research Challenges in Information Science (RCIS)*. (Cité en pages 22 et 27.)

[Holme and Saramäki, 2012] Holme, P. and Saramäki, J. (2012). Temporal networks. *Physics reports*, 519 :97–125. (Cité en pages 15 et 24.)

[Hopcroft et al., 2004] Hopcroft, J., Khan, O., Kulis, B., and Selman, B. (2004). Tracking evolving communities in large linked networks. In *National Academy of Sciences of the United States of America*, volume 101, page 5249. National Acad Sciences. (Cité en page 67.)

[Hutchison et al., 2013] Hutchison, R. M., Womelsdorf, T., Allen, E. A., Bandettini, P. A., Calhoun, V. D., Corbetta, M., Della Penna, S., Duyn, J. H., Glover, G. H., Gonzalez-Castillo, J., et al. (2013). Dynamic functional connectivity : promise, issues, and interpretations. *Neuroimage*, 80 :360–378. (Cité en page 101.)

[Isella et al., 2011] Isella, L., Stehlé, J., Barrat, A., Cattuto, C., Pinton, J.-F., and Van den Broeck, W. (2011). What's in a crowd ? analysis of face-to-face behavioral networks. *Journal of theoretical biology*, 271(1) :166–180. (Cité en page 35.)

[Jensen, 1982] Jensen, R. (1982). Adoption and diffusion of an innovation of uncertain profitability. *Journal of economic theory*, 27(1) :182–193. (Cité en page 42.)

[Kant, 1781] Kant, I. (1781). *Kritik der reinen Vernunft*. (Cité en page 30.)

[Karkada et al., 2011] Karkada, U. H., Adamic, L. A., Kahn, J. M., and Iwashyna, T. J. (2011). Limiting the spread of highly resistant hospital-acquired microorganisms via critical care transfers : a simulation study. *Intensive care medicine*, 37(10) :1633–1640. (Cité en page 44.)

[Karsai et al., 2011] Karsai, M., Kivelä, M., Pan, R. K., Kaski, K., Kertész, J., Barabási, A.-L., and Saramäki, J. (2011). Small but slow world : How network topology and burstiness slow down spreading. *Physical Review E*, 83(2). (Cité en pages 42 et 47.)

[Karsai et al., 2014] Karsai, M., Perra, N., and Vespignani, A. (2014). Time varying networks and the weakness of strong ties. *Scientific reports*. (Cité en page 25.)

[Keeling and Eames, 2005] Keeling, M. J. and Eames, K. T. (2005). Networks and epidemic models. *Journal of the Royal Society Interface*, 2(4) :295–307. (Cité en page 43.)

[Kemeny and Snell, 1960] Kemeny, J. G. and Snell, J. L. (1960). *Finite markov chains*, volume 356. (Cité en page 30.)

[Kempe et al., 2000] Kempe, D., Kleinberg, J., and Kumar, A. (2000). Connectivity and inference problems for temporal networks. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 504–513. (Cité en pages 16 et 25.)

[Kempe et al., 2003] Kempe, D., Kleinberg, J., and Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. (Cité en page 42.)

[Kermark and Mckendrick, 1927] Kermark, M. and Mckendrick, A. (1927). Contributions to the mathematical theory of epidemics. part I. In *Proceedings of the Royal Society*, volume 115, pages 700–721. (Cité en pages 41 et 43.)

[Kitsak et al., 2010] Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., and Makse, H. A. (2010). Identification of influential spreaders in complex networks. *Nature Physics*, 6(11) :888–893. (Cité en page 46.)

[Kivelä et al., 2012] Kivelä, M., Pan, R. K., Kaski, K., Kertész, J., Saramäki, J., and Karsai, M. (2012). Multiscale analysis of spreading in a large communication network. *Journal of Statistical Mechanics : Theory and Experiment*, 2012. (Cité en page 48.)

[Leibnitz et al., 2006] Leibnitz, K., Hoßfeld, T., Wakamiya, N., and Murata, M. (2006). Modeling of epidemic diffusion in peer-to-peer file-sharing networks. In *Biologically Inspired Approaches to Advanced Information Technology*, pages 322–329. (Cité en page 93.)

[Leskovec et al., 2005] Leskovec, J., Kleinberg, J., and Faloutsos, C. (2005). Graphs over time : densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187. (Cité en page 20.)

[M. B. Jdidia and Fleury, 2007] M. B. Jdidia, C. R. and Fleury, E. (2007). Communities detection and analysis of their dynamics in collaborative networks. In *IEEE ICDIM*, pages 744–749. (Cité en page 68.)

[Madar et al., 2004] Madar, N., Kalisky, T., Cohen, R., ben Avraham, D., and Havlin, S. (2004). Immunization and epidemic dynamics in complex networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2) :269–276. (Cité en page 45.)

[Mahajan et al., 1990] Mahajan, V., Muller, E., and Bass, F. M. (1990). New product diffusion models in marketing : A review and directions for research. *The Journal of Marketing*, pages 1–26. (Cité en page 45.)

[Masuda and Holme, 2013] Masuda, N. and Holme, P. (2013). Predicting and controlling infectious disease epidemics using temporal networks. *F1000prime reports*. (Cité en page 46.)

[Merler and Ajelli, 2009] Merler, S. and Ajelli, M. (2009). The role of population heterogeneity and human mobility in the spread of pandemic influenza. *Proceedings of the Royal Society B : Biological Sciences*, 277(1681) :557–565. (Cité en page 48.)

[Miorandi and De Pellegrini, 2010] Miorandi, D. and De Pellegrini, F. (2010). K-shell decomposition for dynamic complex networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 488–496. (Cité en page 25.)

[Mucha et al., 2010] Mucha, P., Richardson, T., Macon, K., and Porter, M. (2010). Community structure in time-dependent, multiscale, and multiplex networks. *science*, 876 :10–13. (Cité en page 68.)

[Myers et al., 2012] Myers, S. A., Zhu, C., and Leskovec, J. (2012). Information diffusion and external influence in networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 33–41. (Cité en page 43.)

[Neiger et al., 2012] Neiger, V., Crespelle, C., and Fleury, E. (2012). On the structure of changes in dynamic contact networks. In *Signal Image Technology and Internet Based Systems (SITIS)*, pages 731–738. (Cité en page 26.)

[Newman, 2002] Newman, M. E. (2002). Spread of epidemic disease on networks. *Physical review E*. (Cité en page 43.)

[Newman and Girvan, 2004] Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, 69. (Cité en page 65.)

[Newton, 1687] Newton, I. (1687). *PhilosophiæNaturalis Principia Mathematica*. (Cité en page 30.)

[Nicosia et al., 2013] Nicosia, V., Tang, J., Mascolo, C., Musolesi, M., Russo, G., and Latora, V. (2013). Graph metrics for temporal networks. In *Temporal Networks*, pages 15–40. (Cité en page 24.)

[Ning et al., 2007] Ning, H., Xu, W., Chi, Y., Gong, Y., and Huang, T. (2007). Incremental spectral clustering with application to monitoring of evolving blog communities. *In SIAM Int. Conf. on Data Mining*. (Cité en page 68.)

[Oliveira and Gama, 2010a] Oliveira, M. and Gama, J. (2010a). Bipartite graphs for monitoring clusters transitions. *Advances in Intelligent Data Analysis IX*, M :114–124. (Cité en page 67.)

[Oliveira and Gama, 2010b] Oliveira, M. and Gama, J. (2010b). Understanding Clusters Evolution. In *Workshop on Ubiquitous Data Mining*, volume D, pages 16 – 20. (Cité en page 67.)

[Pan and Saramäki, 2011] Pan, R. K. and Saramäki, J. (2011). Path lengths, correlations, and centrality in temporal networks. *Physical Review E*, 84. (Cité en page 25.)

[Pastor-Satorras and Vespignani, 2001] Pastor-Satorras, R. and Vespignani, A. (2001). Epidemic spreading in scale-free networks. *Physical review letters*, 86(14) :3200–3203. (Cité en page 43.)

[R. Kumar and Chakrabarti, 2006] R. Kumar, A. T. and Chakrabarti, D. (2006). Evolutionary clustering. In *12th ACM SIGKDD international conference on Knowledge discovery and data mining*. (Cité en page 68.)

[Ribeiro et al., 2013] Ribeiro, B., Perra, N., and Baronchelli, A. (2013). Quantifying the effect of temporal resolution on time-varying networks. *Scientific reports*, 3. (Cité en page 20.)

[Rocha and Blondel, 2013] Rocha, L. E. and Blondel, V. D. (2013). Flow motifs reveal limitations of the static framework to represent human interactions. *Physical Review E*, 87. (Cité en page 26.)

[Rodriguez et al., 2011] Rodriguez, M. G., Balduzzi, D., and Schölkopf, B. (2011). Uncovering the temporal dynamics of diffusion networks. *Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA*. (Cité en page 46.)

[Rogers, 1962] Rogers, E. (1962). The diffusion of new innovations. *The Free Press*. (Cité en page 42.)

[S.-Y. Chan and Xu, 2009] S.-Y. Chan, P. H. and Xu, K. (2009). Community Detection of Time-Varying Mobile Social Networks. *Complex Sciences*, pages 1154—-1159. (Cité en page 68.)

[Saito et al., 2010] Saito, K., Kimura, M., Ohara, K., and Motoda, H. (2010). Selecting information diffusion models over social networks for behavioral analysis. In *Machine Learning and Knowledge Discovery in Databases*, pages 180–195. (Cité en page 42.)

[Salah Brahim et al., 2011] Salah Brahim, A., Le Grand, B., Tabourier, L., and Latapy, M. (2011). Citations among blogs in a hierarchy of communities : Method and case study. *Journal of Computational Science*, 2 :247–252. (Cité en page 44.)

[Scherrer et al., 2008] Scherrer, A., Borgnat, P., Fleury, E., Guillaume, J.-L., and Robardet, C. (2008). Description and simulation of dynamic mobility networks. *Computer Networks*, 52(15) :2842–2858. (Cité en pages 23 et 26.)

[Seidman, 1983] Seidman, S. B. (1983). Network structure and minimum degree. *Social networks*, 5 :269–287. (Cité en page 25.)

[Seifi et al., 2013] Seifi, M., Junier, I., Rouquier, J.-B., Iskrov, S., and Guillaume, J.-L. (2013). Stable community cores in complex networks. In *Complex Networks*, pages 87–98. (Cité en pages 72 et 104.)

[Song et al., 2007] Song, X., Chi, Y., Tseng, B., Zhou, D., and Hino, K. (2007). Evolutionary spectral clustering by incorporating temporal smoothness. In *13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 153–162. (Cité en page 68.)

[Spiliopoulou et al., 2006] Spiliopoulou, M., Ntoutsi, I., Theodoridis, Y., and Schult, R. (2006). Monic : modeling and monitoring cluster transitions. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 706–711. ACM New York, NY, USA. (Cité en page 67.)

[Stehlé et al., 2011] Stehlé, J., Voirin, N., Barrat, A., Cattuto, C., Colizza, V., Isella, L., Régis, C., Pinton, J.-F., Khanafer, N., Van den Broeck, W., et al. (2011). Simulation of an SEIR infectious disease model on the dynamic contact network of conference attendees. *BMC medicine*, 9(1) :87. (Cité en page 47.)

[Sulo et al., 2010] Sulo, R., Berger-Wolf, T., and Grossman, R. (2010). Meaningful selection of temporal resolution for dynamic networks. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, pages 127–136. (Cité en page 21.)

[Tabourier et al., 2012] Tabourier, L., Stoica, A., and Peruani, F. (2012). How to detect causality effects on large dynamical communication networks : a case study. In *Communication Systems and Networks (COMSNETS)*, pages 1–7. (Cité en page 48.)

[Takaguchi et al., 2013] Takaguchi, T., Masuda, N., and Holme, P. (2013). Bursty communication patterns facilitate spreading in a threshold-based epidemic dynamics. *PloS one*, 8. (Cité en page 47.)

[Tantipathananandh et al., 2007] Tantipathananandh, C., Berger-Wolf, T., and Kempe, D. (2007). A framework for community identification in dynamic social networks. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, page 717. (Cité en page 67.)

[Traag and Bruggeman, 2009] Traag, V. and Bruggeman, J. (2009). Community detection in networks with positive and negative links. *Physical Review E*, 80. (Cité en page 105.)

[Valente, 1995] Valente, T. W. (1995). *Network models of the diffusion of innovations*, volume 2. (Cité en page 42.)

[Van Berkel et al., 1999] Van Berkel, C., Josephs, M. B., and Nowick, S. M. (1999). Applications of asynchronous circuits. *Proceedings of the IEEE*, 87 :223–233. (Cité en page 30.)

[Vernon and Keeling, 2009] Vernon, M. C. and Keeling, M. J. (2009). Representing the UK's cattle herd as static and dynamic networks. *Proceedings of the Royal Society B : Biological Sciences*, 276(1656) :469–476. (Cité en pages 20 et 47.)

[Viard and Latapy, 2014] Viard, J. and Latapy, M. (2014). Identifying roles in an ip network with temporal and structural density. *In Sixth IEEE International Workshop on Network Science for Communication Networks (NetSciCom 2014)*. (Cité en page 25.)

[Volz, 2008] Volz, E. (2008). SIR dynamics in random networks with heterogeneous connectivity. *Journal of mathematical biology*, 56(3) :293–310. (Cité en page 43.)

[Volz and Meyers, 2009] Volz, E. and Meyers, L. A. (2009). Epidemic thresholds in dynamic contact networks. *Journal of the Royal Society Interface*, 6(32) :233–241. (Cité en page 46.)

[Wang and Fleury, 2010] Wang, Q. and Fleury, E. (2010). Mining time-dependent communities. In *Latin American Workshop on Dynamic Networks*. (Cité en page 67.)

[Wang et al., 2008] Wang, Y., Wu, B., and Du, N. (2008). Community Evolution of Social Network : Feature, Algorithm and Model. *Science And Technology*, (60402011). (Cité en page 67.)

[Whitbeck et al., 2011] Whitbeck, J., Conan, V., and de Amorim, M. D. (2011). Performance of opportunistic epidemic routing on edge-markovian dynamic graphs. *Communications, IEEE Transactions*, 59(5) :1259–1263. (Cité en page 48.)

[Yoneki et al., 2008] Yoneki, E., Hui, P., and Crowcroft, J. (2008). Wireless epidemic spread in dynamic human networks. In *Bio-Inspired Computing and Communication*, pages 116–132. (Cité en page 48.)