

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Jijel

Faculté des sciences exactes et d'informatique

Département d'informatique



Mémoire

Pour obtenir le diplôme de

Master en Informatique

Spécialité : **Réseaux et sécurité**

**Algorithme de détection de communautés
dans les graphes temporels**

Présenté par :

BOUDJELLABA IMEN

LAOUICI ILHAM

Sous la direction de :

BOUDEBZA SOUÀAD



— Année universitaire 2015-2016 —

M. inf. RS 02/16

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Jijel

Faculté des sciences exactes et d'informatique
Département d'informatique



Mémoire

01
02

Pour obtenir le diplôme de
Master en Informatique
Spécialité : **Réseaux et sécurité**

**Algorithme de détection de communautés
dans les graphes temporels**

Présenté par :
BOUDJELLABA IMEN
LAOUICI ILHAM

Sous la direction de :
BOUDEBZA SOUÀAD

— Année universitaire 2015-2016 —

Dédicaces

Avec un énorme plaisir,

un cœur ouvert et une immense joie, que

je dédie mon travail à mes très chers, respectueux et

Magnifiques parents qui m'ont soutenus tout au long de ma vie,

Que dieu les garde pour nous.

Ainsi à mes sœurs: Faten, Manel, Rania

Je leurs souhaite une belle vie.

A mon amour, mon bébé, mon frère Othmane.

A mon amie depuis l'enfance,

Amina.

A mes amies Chahira & Soumia.

A mon binôme Isham

A toute ma grande famille

A toute personne qui m'ont encouragé ou aidé.

A la fin pour toutes personnes que j'aime et qui m'aiment.

Imen

Dédicaces

Je dédie ce mémoire :

À ceux qui me sont les plus chers sur cette terre, ma raison de vivre,

*Ma source de succès, ceux qui m'ont apporté une tendresse
incomparable et Une affection inouïe « mes parents » pour leur
patience, leur amour, leur Soutien et leurs encouragements.*

À toutes mes sœurs, en particulier ma grande sœur

Ahlam

À mon frère Nourelislam que j'adore

À tout ma famille.

À mon binôme Imen et mes amies et mes camarades.

*Enfin je voudrais dédier ce mémoire à tout personnes ayant
participé de loin ou De près à la réalisation ce travaille*

Islam

Remerciement

Ce mémoire n'aurait été possible sans l'aide de dieu que nous Remercions de nous avoir donnée le courage, la volonté nécessaire et la Patience durant toutes ces années d'études et que grâce a lui ce travail a pu être réalisé

Nous tenons à remercier très vivement Madame Souaad Boudabza, Professeur à l'université de Tijel qui nous a proposé ce sujet. Pour avoir dirigé ce travail, pour la confiance et l'intérêt qu'elle a témoigné tout au long de la réalisation de ce travail, pour sa disponibilité ainsi de nous avoir fait bénéficier de ses compétences

Nous adressons notre vif remerciement aux membres de jury d'avoir accepté de juger ce travail.

Nous tenons également à remercier tous ceux qui nous ont aidés de près ou de loin à l'élaboration de ce travail.

Nous tenons à remercier Nos parents pour tout ce qui ont fait pour nous.

Inen et Itham

Résumé

Les systèmes complexes issus de nombreuses disciplines peuvent être modélisés par des réseaux, plus précisément par des graphes de nœuds connectés par des arrêtes. Ces réseaux exhibent une structure microscopique dite « structure communautaire ». Une communauté est vue comme un sous-graphe composé de nœuds densément liés entre eux et faiblement liés aux autres nœuds du réseau. La détection de cette structure communautaire est cruciale pour comprendre la topologie et le fonctionnement de ces réseaux. La majorité des travaux dans la littérature concernant la détection de communautés portent sur des réseaux statiques. Or, beaucoup de réseaux évoluent au cours du temps. L'approche traditionnelle de la détection de communautés réutilise les algorithmes statiques sur différents instantanés du réseau et souffre du problème d'instabilité. Dans le cadre de ce travail, nous proposons un algorithme de détection de communautés dynamiques baptisé TCCD (Temporel Community Clique Détection), basé sur la notion de clique et de graphes temporels. Nous optons pour une approche à base de clique pour permettre une détection de communautés recouvrantes. Nous exploitons l'information temporelle dans les graphes temporels dans le but de suivre le réseau au fur et à mesure de son évolution. Afin de valider notre proposition nous avons effectué des expérimentations sur des réseaux de test statiques et dynamiques.

Mots clés :

Algorithme de détection, détection de communautés, communauté, graphe temporel, communautés dynamiques.

Abstract

Complex systems from many disciplines can be modeled by networks, specifically by nodes connected by edges graphs. These networks exhibit a microscopic structure called "community structure". A community is seen as a subgraph composed of densely linked nodes together and weakly linked to other network nodes. The detection of this community structure is crucial to understanding the topology and operation of these networks. The majority of work in the literature concerning the community detection relate to static networks. However, many networks evolve over time. The traditional approach to community detection reuses static algorithms on different snapshots of the network and suffer instability problem. As part of this work, we propose a dynamic communities detection algorithm called TCCD (Click Detection Temporal Community), based on the concept of clicks and time graphs. We opt for a click-based approach to allow detection of overlapping communities. We use the time information in the time charts in order to monitor the network as and when it evolves. To validate our proposal we conducted experiments on static and dynamic test systems.

Keywords: detection algorithm community detection, community, temporal graph, dynamic communities.

Liste des abréviations

- CPM** Clique percolation Méthod
- iLCD** intrinsic Longitudinal Community Detection.
- TCCD** Temporel Community Clique Détection
- LND** 3D Landscape Data
- TNF** Communauty Transport Neutral encapsulation Format File

Table de matières

Introduction générale	14
Chapitre 1 : Notions de base	16
1. Les réseaux complexes.....	17
1.1.Définition d'un réseau complexe.....	17
1.2.Domaines des réseaux complexes.....	17
1.3.Types des réseaux complexes.....	18
1.3.1.Réseaux évoluant fortement.....	18
1.3.2.Réseaux petit monde (Small-world network).....	19
2. La théorie de graphes.....	19
2.1.Concepts et notations.....	19
2.2.Les graphes temporels.....	23
2.2.1.Les séquences de contact.....	23
3.Conclusion.....	24
Chapitre 2 : Etat de l'art	25
1.Définition d'une communauté.....	26
2.Détection de communautés.....	26
3.Détection de Communautés dynamiques.....	26
3.1.Opérations sur les communautés dynamiques.....	27
4.Approches de détection de communautés.....	28
4.1.Approches basées sur des algorithmes statiques.....	28
4.1.1.Approches hiérarchiques.....	28
4.1.2.Approches basées sur l'optimisation d'une fonction objective.....	29
4.1.3.Approches à base de cliques.....	30
4.2.Approches dynamiques.....	31
4.2.1.Approches travaillant sur des instantanés du réseau.....	31

4.2.2. Approche travaillant sur les réseaux temporels.....	33
5. Synthèse de l'état de l'art.....	35
6. Conclusion.....	37
Chapitre 3 : Proposition.....	37
1. Les exigences de TCCD.....	39
2. Principe de TCCD.....	39
2.1. Ajout d'un lien dans le graphe.....	40
2.1.1. Lien externe.....	40
2.1.2. Lien interne.....	43
2.1.3. Lien intra-communautés.....	43
2.2. Suppression d'un lien dans le graphe.....	44
2.2.1. Le lien externe.....	44
2.2.2. Le lien interne.....	44
2.2.3. Le lien intra-communautés.....	45
3. Pseudo algorithme TCCD.....	46
4. Conclusion.....	48
Chapitre 4 : Implémentation et validation.....	49
1. Développement de l'outil TCCD.....	50
1.1. Environnement de développement.....	50
1.1.1. IDE Netbeans.....	50
1.1.2. Langage Java.....	50
1.1.3. La bibliothèque GraphStream.....	51
1.2. Formats des fichiers utilisés.....	51
1.2.1. Format des fichiers d'entrée.....	51
1.2.2. Format du fichier de sortie.....	52
1.3. Présentation de l'outil.....	52
1.3.1. Vue générale de l'outil.....	52

1.3.2.Exemples explicatifs.....	57
2.Teste et validation.....	63
2.1.Les Réseaux de teste.....	63
2.1.1.Club de karaté du Zachary.....	63
2.1.2. Réseau de dauphins de Lusseau	64
2.1.3.Le réseau temporel d'isards.....	65
3.Choix des algorithmes à comparer.....	67
3.1.Test sur des réseaux statiques.....	67
3.2.Test sur le réseau dynamique d'Izard.....	67
4.Bilan.....	68
5.Conclusion.....	68
Conclusion et perspective.....	69
Bibliographie.....	70

Liste des Figures

Figure 1 : Réseau social d'un site web en communautés.....	18
Figure 2 : Représentation d'un graphe G.....	20
Figure 3 : Représentation d'un sous-graphe G'	22
Figure 4 : Une clique de $k=3$	22
Figure 5 : Représentation de stable.....	23
Figure 6 : Opérations possibles sur les communautés.....	27
Figure 7 : dendrogramme pour un algorithme agglomératif et divisif.....	28
Figure 8 : Exemple de l'algorithme de percolation de cliques avec $k=3$	31
Figure 9 : Représentation d'une méthode par détections statiques informées successives.....	33
Figure 10 : Ajout d'un lien externe pour les cas (a, b, c, d).....	41
Figure 11 : Ajout d'un lien externe.....	42
Figure 12 : Ajout d'un lien interne.....	43
Figure 13 : Ajout d'un lien intra-communautés.....	43
Figure 14 : Suppression d'un lien externe.....	44
Figure 15 : Suppression d'un lien interne.....	45
Figure 16 : Suppression d'un lien intra-communautés.....	45
Figure 17 : Fichier d'entrée.....	51
Figure 18 : Fichier de sortie.....	52
Figure 19 : Menu principale de l'outil TCCD.....	52
Figure 20 : Le menu fichier de TCCD.....	53
Figure 21 : Choix du réseau à traiter.....	53
Figure 22 : Visualisation du fichier d'entrée.....	54
Figure 23 : Le menu détection.....	54
Figure 24 : Début de la détection.....	55
Figure 25 : Résultat final de la détection.....	55
Figure 26 : La visualisation du fichier résultat de la détection.....	56
Figure 27 : Le fichier résultat des communautés.....	56
Figure 28 : Fichier d'entrée de l'exemple 1.....	57
Figure 29 : Résultat de l'exemple 1.....	57
Figure 30 : Fichier d'entrée pour exemple 2.....	58

Figure 31: Résultat de l'exemple 2.....	58
Figure 32: Fichier d'entrée de l'exemple 3	59
Figure 33: Après l'ajout d'un lien interne.....	59
Figure 34: Fichier d'entrée pour la suppression de lien externe	60
Figure 35 : Avant la suppression du lien(10,11)	60
Figure 36 : Après la suppression du lien(10,11)	60
Figure 37: Fichier d'entrée pour un lien intra-communauté	61
Figure 38: Avant de supprimer le lien(3,11)	61
Figure 39: Après la suppression du lien (3,11)	61
Figure 40 : Le fichier d'entrée pour l'exemple 5	62
Figure 41 : Avant de supprimer le lien(9,10)	62
Figure 42 : Après la suppression du lien (9,10)	62
Figure 43 : Partie du fichier d'entrée pour le réseau du club de Zachary	63
Figure 44: Résultat d'exécution du réseau de Zachary	64
Figure 45: Partie du fichier d'entrée de TCCD pour le réseau des Dauphins de Lusseau	64
Figure 46: Résultat de TCCD pour le réseau des Dauphins de Lusseau	65
Figure 47: Réseau temporel d'isards sous la forme TNF	65
Figure 48: Fichier d'entrée pour le réseau d'isards	66
Figure 49: Résultat de TCCD pour le réseau d'isards	66

Liste des Tableaux

Tableau 1: Comparaison entre les approches de détection de communautés.....36

Tableau 2: Résultats d'exécution pour les réseaux statiques.....67

Tableau 3: Résultats d'exécution pour le réseau dynamique d'izard.....67



Introduction

Les réseaux complexes sont utilisés dans tous les domaines. En informatique, l'internet peut être vu comme un ensemble de routeurs interagissant via des câbles. Alors qu'en biologie, le cerveau est un ensemble de neurones interagissant entre eux. En ce qui concerne le domaine de sociologie, l'étude des réseaux sociaux conduit à étudier comment interagissent divers agents entre eux. De façons générales, ces réseaux peuvent être définis comme un ensemble d'entités reliées. Et leur modélisation se fait par des graphes appelés graphes complexes.

La présence des réseaux complexes dans tous les domaines est la cause essentielle qui a mené les chercheurs à vouloir comprendre et analyser ces réseaux. L'axe qui a reçu beaucoup d'intérêt dans l'analyse des réseaux est la détection de communautés. Une communauté est vue comme une partie du graphe composé de nœuds fortement liés entre eux et faiblement liés avec le reste des nœuds du réseau. La détection de communauté nous permet de comprendre le fonctionnement du réseau.

La plupart des travaux concernant la détection de communautés portent sur des réseaux statiques. Or, la majorité des grands graphes représentent des objets dynamiques. Donc, beaucoup d'informations sont négligées en n'étudiant que le réseau dans son état statique.

La détection de communautés dynamiques peut être traitée de plusieurs façons. Elles peuvent être définies comme une succession de communautés statiques. Ces approches souffrent cependant du problème d'instabilité. En effet, elles peuvent donner des résultats très différents pour des réseaux semblables. Les communautés dynamiques peuvent être aussi détectées directement en utilisant les réseaux temporels. Ces approches assurent la stabilité de la détection, toutefois la majorité d'entre elles ne permettent pas une détection de communautés recouvrantes, c'est-à-dire les communautés partageant des nœuds en commun.

Dans le cadre de ce mémoire notre travail vise à proposer un nouvel algorithme pour la détection de communautés dynamiques, en se basant sur une approche à base de clique travaillant sur des réseaux temporels. En effet, l'utilisation des réseaux temporels permet de suivre le réseau au fur et à mesure de son évolution. Notre choix est porté sur cette notion de clique en premier lieu, à cause de sa simplicité. Puis en second lieu, car les approches à base de clique sont parmi les premières à pouvoir détecter les communautés recouvrantes.

Ce mémoire est organisé autour de quatre chapitres auxquelles s'ajoutent cette introduction et une conclusion générale.

Dans le premier chapitre, nous définissons qu'est ce qu'un réseau complexe, ses types et ses domaines d'application ensuite nous définissons quelques concepts jugés nécessaires sur la théorie des graphes.

Le deuxième chapitre présente dans un premier temps les définitions relatives à la notion de communauté. Une revue des recherches liées à la détection de communautés dynamique sera présentée par la suite. Ce chapitre se termine par une synthèse de l'état de l'art.

Dans le troisième chapitre, nous présenterons les exigences de notre algorithme, ensuite nous donnerons une description de notre proposition TCCD (Temporel Community Clique Détection) avec son pseudo algorithme.

Dans le quatrième chapitre, nous présenterons les outils de développement que nous avons utilisés. Nous fournissons ensuite des exemples de petits graphes qui illustrent le fonctionnement de notre outil de détection. La dernière partie présentera les résultats des expérimentations de l'outil TCCD sur des réseaux de test statiques et dynamiques.

Enfin, nous terminerons ce mémoire en revenant sur notre proposition et en proposant quelques perspectives et travaux futurs.

Chapitre 1

Notions de base

Ce chapitre a pour objectif de présenter les notions de base qui seront utilisées tout au long de ce travail. Il est divisé en deux parties. Dans la première, nous introduirons la notion des réseaux complexe, leurs différents types avec leur caractéristique. La seconde partie présentera les notions liées à la théorie de graphes qui est principalement utilisée pour l'étude des réseaux complexes.

1. Les réseaux complexes

1.1. Définition d'un réseau complexe

Réseau complexe désigne tous les graphes comprenant un nombre élevé de sommets et de liens. Il se différencie des graphes traditionnellement étudiés en analyse des réseaux sociaux où il est rare que le nombre de sommets dépasse la quarantaine [VL08].

Les réseaux complexes sont en fait une sous-classe des systèmes complexes. Un système complexe est généralement considéré comme un ensemble d'éléments en interactions mutuelles, où le comportement global du système ne peut être déduit de la somme de ses parties et de leurs propriétés [VL08].

1.2. Domaines des réseaux complexes

Les réseaux complexes sont présents dans de nombreux domaines aussi divers les uns que les autres comme la biologie, la sociologie, la psychologie et l'informatique. Ces réseaux peuvent être regroupés en quatre catégories :

1. les réseaux sociaux.
2. les réseaux d'informations.
3. les réseaux technologiques.
4. les réseaux biologiques.

Un réseau social est un ensemble de personnes ou groupes de personnes possédant des schémas de contacts ou d'interactions entre eux. Il se définit par différents éléments qui sont les individus, leurs contacts, les affinités avec et entre ces contacts, ainsi que l'environnement qui les entoure. Il s'agit donc d'un ensemble de relations entre des individus [AO12].

Un réseau d'informations peut être rapporté à l'exemple classique d'un réseau de citations entre papiers scientifiques. La structure des informations étant stockée dans les nœuds, c'est pour cela que l'on utilise le terme réseau d'informations. Le World Wide Web avec ses pages web (contenant des informations) et ses hyperliens est également un réseau d'informations. Les réseaux d'information sont différents du réseau Internet qui est le réseau physique reliant les ordinateurs du monde entier entre eux.

Un réseau technologique est un réseau créé par l'homme principalement pour la distribution d'un service ou de l'énergie. Les réseaux électriques, aériens, et les réseaux

d'ordinateurs, en font partie.

Un réseau biologique est un réseau d'éléments touchant au vivant. Un exemple De réseau biologique peut être un réseau d'interactions entre protéines. [VL08]

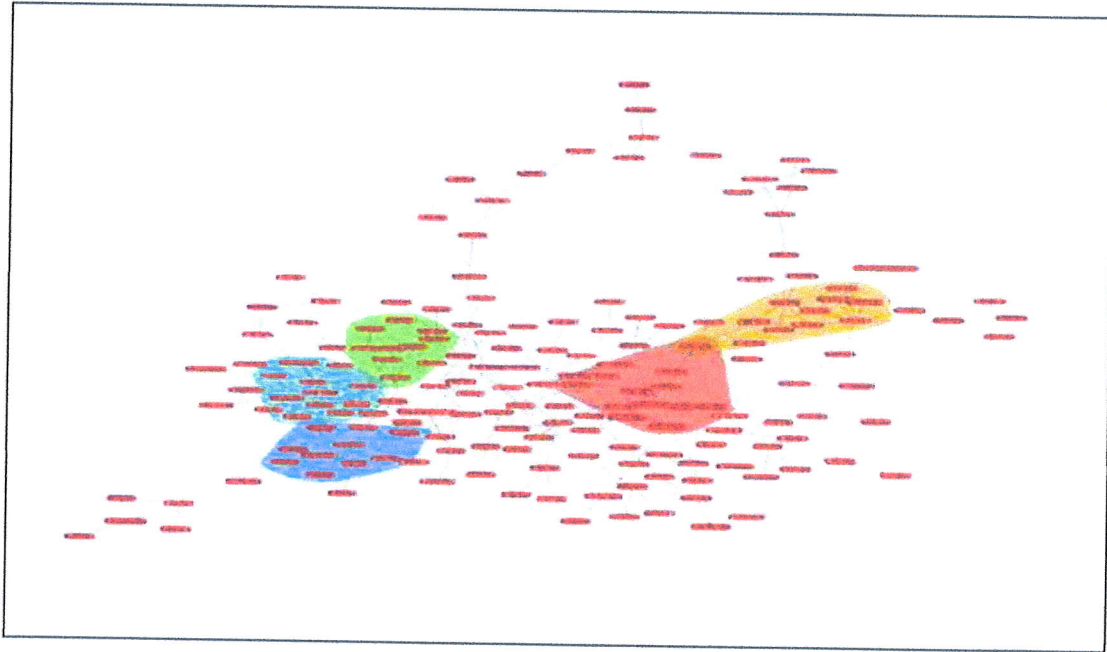


Figure 1 : Réseau social d'un site web en communautés

1.3. Types des réseaux complexes

On distingue deux familles des réseaux complexes qui sont les réseaux évoluant fortement et les réseaux petits-monde.

1.3.1. Réseaux évoluant fortement

Nous utilisons ce terme pour faire la distinction entre, d'un côté, un réseau pour lequel nous disposons d'un petit nombre d'étapes d'évolution, par exemple un réseau de citation pour lequel nous avons un instantané par an, et, d'un autre côté, des réseaux pour lesquels nous disposons de tous les détails de son évolution. S'il est toujours arbitraire de donner une définition précise de cette notion, on peut définir un réseau évoluant fortement comme un réseau pour lequel le nombre d'étapes d'évolution est supérieur au nombre total de nœuds du réseau. [Caz13]

1.3.2 Réseaux petit monde (Small-world network)

La notion de réseau petit monde peut être légèrement différente. Le consensus général est de les caractériser par les propriétés suivantes :

- **Faible distance entre les nœuds** : le nombre moyen pour lier deux nœuds du réseau choisis aléatoirement par la plus courte chaîne reste très faible, quelle que soit la taille du réseau. Cette distance moyenne est de l'ordre de $\ln(n)$. En pratique, des études récentes sur le réseau Facebook [UKB11] tendraient à montrer que ce nombre peut être encore petit.
- **Fort clustering** : cette propriété signifie que les nœuds ont tendance à créer des structures locales denses. Elle peut venir notamment de la transitivité : s'il existe un lien (a, b) et un lien (b, c), alors la probabilité qu'un lien (a, c) existe est renforcée. [Caz13]
- **Structure en communauté** : cette propriété est fortement liée à la précédente. Dans un réseau petit monde, on observe des structures microscopiques, c'est à dire des ensembles de nœuds fortement liés entre eux et plus faiblement liés au reste du réseau.

2. La théorie de graphes

La théorie des graphes est née en 1736 quand Leonhard Euler démontra qu'il était impossible de traverser chacun des sept ponts de la ville russe de Königsberg (aujourd'hui Kaliningrad).

La théorie des graphes constitue par la suite un domaine des mathématiques qui, historiquement, s'est aussi développé au sein de disciplines diverses telles que la chimie (modélisation de structures), la biologie (génomique), les sciences sociales (modélisation des relations) ou en vue d'applications industrielles (problème du voyageur de commerce).

De manière générale, un graphe permet de représenter simplement la structure, les connexions, les cheminements possibles d'un ensemble complexe comprenant un grand nombre de situations, en exprimant les relations, les dépendances entre ses éléments.

2.1. Concepts et notations

La théorie de graphes est principalement l'outil le plus utilisé dans les différentes études des réseaux complexes. Dans ce qui suit, nous donnerons un certain nombre de définitions des concepts de cette théorie et de notations les concernant. Nous présenterions

également les graphes temporels qui représentent une catégorie particulière pour modéliser les réseaux dynamiques.

Définition 1 (Graphe) : Un graphe G est un couple formé de deux ensembles, un ensemble $V = \{v_1, v_2, \dots, v_n\}$ dont les éléments sont appelés sommets ou nœuds, et un ensemble $E = \{e_1, e_2, \dots, e_p\}$, dont les éléments sont appelés arêtes ou arcs. On notera le graphe $G = (V, E)$. Lorsque $e = \{v_1, v_2\}$, on dit que e est l'arête de G d'extrémités v_1 et v_2 , ou que e relie v_1 et v_2 , ou que e passe par v_1 et v_2 . Les sommets v_1 et v_2 sont dits adjacents dans G . Un exemple de graphe est illustré dans la figure 2.

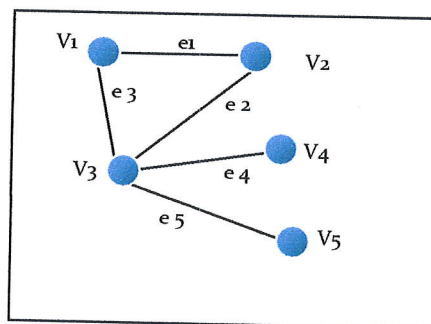


Figure 2 : Représentation d'un graphe G

Définition 2 (Une arête). Est une liaison entre deux sommets. Donc on parle d'arc ou de lien qui peut être:

- Une **hyperarête** qui relie deux ou plusieurs sommets.
- Une arête **orientée** qui ne s'utilise que dans une direction unique.
- Une arête **non orientée** qui s'utilise dans les deux directions.
- Une arête **pondérée** lorsqu'on lui attribue un poids de liaison (arête ou arc) entre les sommets, dont ce poids est un nombre réel positif ou nul.
- Une arête **étiquetée** lorsqu'on lui attribue un label qui est une lettre, un mot, un nombre ou un symbole.

Définition 3 (Le degré de sommet) .Soit $G = (V;E)$ un graphe et v un sommet de ce graphe. Le degré de v , noté $d(v)$, est le nombre d'arêtes incidentes à v ; c'est-à-dire contenant v . Lorsque $d(v) = 0$, on dit que le sommet v est isolé, lorsque $v = 1$, il est dit que le sommet v est regroupé.

Chapitre 1 : Notions de base

Définition 4 (graphe orienté). Si deux sommets consécutifs v_1 et v_2 sont reliés par une arête, on dira que l'on peut passer de v_1 à v_2 , à moins qu'un sens de parcours soit imposé. Les arêtes sont alors fléchées dans le sens de parcours autorisé et on parle d'arcs pouvant s'interpréter comme un couple (v_1, v_2) à distinguer de (v_2, v_1) . Dans ce cas, on parle de graphe orienté.

Définition 5 (chemin, chaîne, circuit, cycle). Un chemin dans un graphe est une succession d'arcs parcourus dans le même sens. Le nombre d'arcs parcourus s'appelle la longueur du chemin. On parle de chaîne si l'on ne tient pas compte de la direction des arcs ; on parle ainsi de chaîne dans les graphes non orientés. Si un chemin revient à son point de départ, on parle de circuit dans un graphe orienté, ou de cycle dans un graphe non orienté.

Définition 6 (distance). La distance entre deux sommets est la longueur du plus court chemin entre ces deux sommets.

Définition 7 (diamètre). Le diamètre d'un graphe est la plus grande distance séparant deux sommets de ce graphe.

Définition 8 (graphe connexe). Un graphe $G = (V, E)$ est dit connexe si quels que soit les sommets i et j de V , il existe une chaîne entre ces deux sommets.

Définition 9 (graphe complet). Un graphe non orienté $G = (V, E)$ est dit complet si quelque soit la paire $(i, j) \in V$, il existe une arête appartenant à E reliant les deux sommets i et j . Un graphe complet de n sommets contient $\frac{n(n-1)}{2}$ arêtes.

Définition 10 (graphe pondéré). Un graphe est dit pondéré s'il désigne un graphe avec des arêtes pondérées.

Définition 11 (graphe étiqueté). Un graphe est dit étiqueté s'il désigne un graphe (orienté ou non) avec des arêtes étiquetées. Dans cette représentation structurelle, les individus sont des sommets du graphe tandis que les arêtes décrivent la nature de la relation topologique.

Définition 12 (Adjacence) Deux sommets sont adjacents (ou voisins) s'ils sont joints par un arc. Et deux arcs sont adjacents s'ils ont au moins une extrémité commune.[VL08]

Définition 13 (Sous-graphe). Un sous-graphe de G est un graphe $G'=(W,E(W))$ tel que W est un sous-ensemble de V , et $E(W)$ c'est les arêtes de E dont les 2 extrémité sont des sommets de W . $E(W) = \{(v_2, v_1) \in E \mid v_2, v_1 \in W\}$. La figure 3 illustre un sous-graphe G' de graphe G de l'exemple précédent.

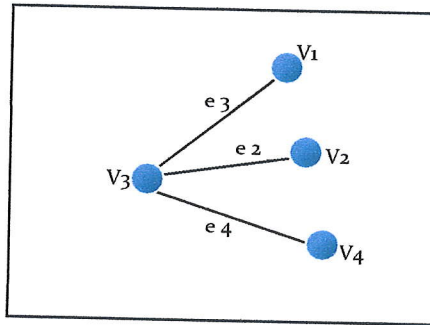


Figure 3 : Représentation d'un sous-graphe G'

Définition 14 (graphe partiel). Un graphe partiel de G est un graphe $I=(V,F)$ tel que F est un sous-ensemble de E . Soit $G=(V,E)$ un graphe. Le graphe $G'=(V,E')$ est un graphe partiel de G , si E' est inclus dans E . Autrement dit, on obtient G' en enlevant une ou plusieurs arêtes au graphe G .

Définition 15 (communauté). Est un ensemble de nœuds fortement liées entre eux et faiblement liées avec les autres nœuds du graphe.

Définition 16 (Clique). Est l'ensemble de sommets d'un sous-graphe complet. Soit C une clique de G non orienté. On peut dire aussi qu'une clique de G est un sous graphe complet de G , où chaque nœud est lié avec tous les autres, il est évident qu'une clique possède les propriétés requises pour une communauté, mais elle impose aussi des restrictions, on parle de k -clique pour désigner un graphe complet de k sommets. (Voir figure4)

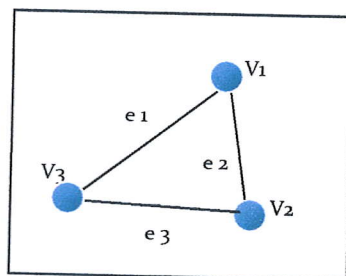


Figure 4 : Une clique de $k=3$

Définition 17 (graphe stable). Un stable est un sous-graphe de G sans arêtes. (Voir la figure5)

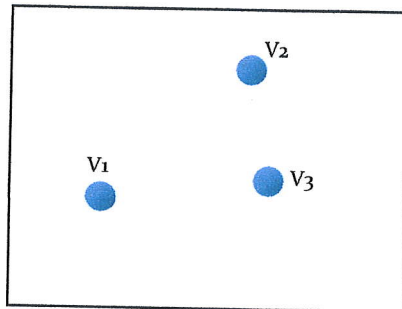


Figure 5 : Représentation de stable

Définition 18 (voisinage). L'ensemble des voisins d'un sommet $j \in V$ noté $N(j)$, est défini comme $N(j) = \{i \in V \mid (i, j) \in E\}$.

Définition 19 (graphe statique). Un graphe $G = (V, E)$ est dit statique si l'ensemble des nœuds V et l'ensemble des liens E ne changent pas au cours du temps.

Définition 20 (nœud chevauché). Un nœud chevauché est un nœud qui appartient à plusieurs communautés en même temps.

Définition 21 (degré externe d'un nœud). Soit v un sommet de graphe G , le degré externe de v est le nombre d'arêtes incidentes à v et qui n'appartiennent à aucune communauté.

Définition 22 (degré interne d'un nœud). Soit v un sommet de graphe G , le degré interne de v est le nombre d'arêtes incidentes à v et qui appartiennent à une communauté.

2.2. Les graphes temporels

Les graphes temporels sont des graphes acycliques orientés. La présence d'un arc entre deux nœuds n_1 et n_2 , traduit le fait que n_2 se situe temporellement après n_1 . On distingue deux types de graphes temporels. On va s'intéresser au type qui s'appelle séquences de contact.

2.2.1. Les séquences de contact :

Dans ce type de graphe, un lien est représenté par un triplet (i, j, t) , tel que i et j sont des nœuds du graphe, et t est l'instant où le lien a été activé. Ce triplet représente donc l'interaction entre les individus i et j à l'instant t . Les séquences de contacts sont particulièrement utilisées pour représenter les relations entre individus pour lesquelles la

durée de l'interaction est ignorable, comme les réseaux de communications asynchrones (e-mails, courriers), ou réseaux de communications téléphoniques, etc.

3. Conclusion :

Dans ce chapitre nous avons présenté les notions de base concernant les réseaux complexes, leurs types et leurs propriétés. Nous avons également décrit les concepts et la notation liée à la théorie de graphes. Celle-ci est considérée comme l'outil le plus adéquat pour la représentation des réseaux complexes. En particulier, nous avons introduit le concept de graphe temporel qui est dédié à l'étude des réseaux complexes évoluant dans le temps. Le chapitre suivant présentera une des problématiques essentielles de ces réseaux qui est la détection de communautés.

Chapitre 2

Etat de l'art

Dans ce chapitre nous introduisons la notion de communauté aussi que la problématique de détection de communautés. Nous présenterons en revanche les différentes approches pour la détection. Ainsi, nous porterons notre attention aux approches de détection dynamiques, auxquelles se rapporte notre travail.

1. Définition d'une communauté

Une caractéristique des graphes complexe est la possibilité de les diviser en communautés. D'après Newman et Girvan en 2004 [New04] une communauté est définie comme un sous-graphe composé de nœuds densément liés entre eux et faiblement liés aux autres nœuds du graphe.

Les communautés peuvent avoir des interprétations différentes suivant le type de réseau considéré. Pour des réseaux d'informations, elles correspondent à des pages web traitant un même sujet. Pour les réseaux métaboliques, les communautés correspondent par exemple à des fonctions biologiques de la cellule, etc. La détection de communauté est donc un outil important pour la compréhension de la structure et du fonctionnement des réseaux complexes.

2. Détection de communautés

La détection de communautés est une problématique cruciale lors de l'analyse des systèmes complexes, par exemple, on peut souhaiter étudier les interactions entre des individus, entre des protéines ou bien les liens entre différents sites internet. Ces données peuvent se représenter sous forme de graphe où chaque nœud représente un individu et une arête une interaction entre deux individus. Ces réseaux ont la propriété de se diviser en communauté. Ce problème est posé pour la première fois dans l'article de Girvan et Newman en 2002. [GN02]

La détection de communautés s'approche des deux thématiques classiques de Partitionnement de graphe et de regroupées les données. La première, initialement introduite pour la parallélisations des processus, cherchent à répartir des tâches représentées par les sommets d'un graphe tout en minimisant les échanges, représentés par les arêtes. La seconde thématique de regroupement de données est une thématique générale plus vaste dans laquelle on cherche à regrouper des données possédant des caractères communs.

3. Détection de Communautés dynamiques

Beaucoup de travaux portant sur la définition et la détection de communauté ont été proposés ces dernières années. Néanmoins, la plupart d'entre elles ne valent que pour des graphes statiques. Or, dans les réseaux complexes les interactions entre communautés évoluent dynamiquement au cours du temps. Par exemple, dans le réseau social Facebook les utilisateurs ajoutent et/ou suppriment des amis. Les données sont très souvent dynamiques et

une grande quantité d'informations est donc complètement ignorée. Ce dynamisme constitue un nouveau challenge qui a tout récemment fait son apparition dans la détection de communautés.

3.1. Opérations sur les communautés dynamiques

Lorsque les communautés sont amenées à évoluer, elles peuvent le faire de plusieurs façons. Palla et al [PBV07]. (Voir la Figure 6)

- **Croissance et Contraction** : signifie l'ajout et le retrait d'un nœud dans une communauté existante.
- **Naissance et Mort** : signifie à l'apparition de nouvelles communautés et la disparition d'anciennes communautés.
- **Fusion et Division** : sont deux opérations un peu plus complexes, correspondant au regroupement de deux communautés en une seule communauté, ou la décomposition d'une communauté en deux nouvelles communautés.

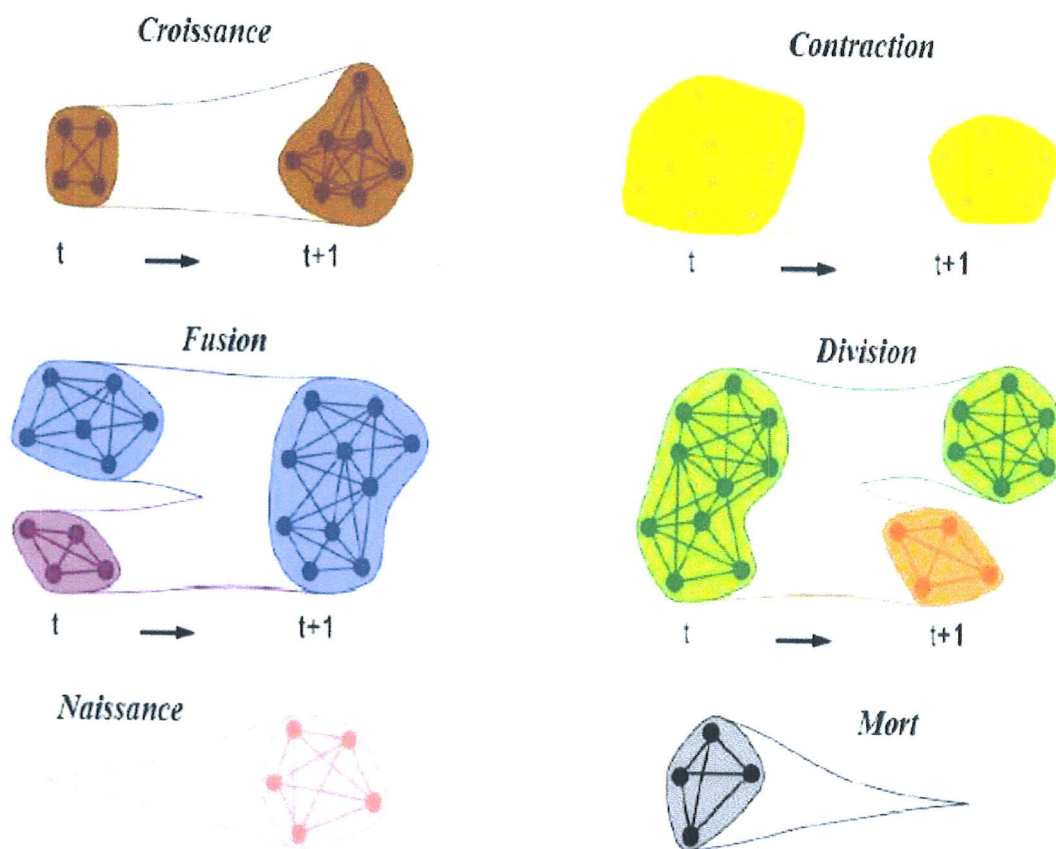


Figure 6: Opérations possibles sur les communautés.

4. Approches de détection de communautés

4.1. Approches basées sur des algorithmes statiques

Les méthodes de détection de communautés statiques sont classées en trois catégories. La première contient les méthodes de classification hiérarchiques qui permettent de choisir une structure de communautés parmi plusieurs niveaux hiérarchiques représentant différentes structures possibles. La deuxième catégorie quant à elle englobe les méthodes d'optimisation d'une fonction objective, qui identifient les communautés en maximisant une fonction de qualité. Finalement, la dernière catégorie porte sur les méthodes à base d'heuristiques dont des formalismes s'appliquent sur les nœuds du réseau d'une manière itérative jusqu'à avoir une structure de communautés stable. [SR08]

4.1.1 Approches hiérarchiques

Les méthodes hiérarchiques travaillent sur plusieurs niveaux. Elles sont divisées en deux types, le premier est les algorithmes hiérarchiques ascendants (agglomératifs) et le second est les algorithmes hiérarchiques descendants (divisifs).

L'idée du premier type est que les sommets d'un graphe sont regroupés itérativement en communautés en partent d'une partition de 'n' communautés composées d'un seul sommet. Les regroupements de communautés sont poursuivis jusqu'à obtenir une seule communauté regroupant tous les sommets et une structure hiérarchique de communautés qu'on appelle dendrogramme comme le montre la figure 7.

Le second type des méthodes hiérarchiques consiste à partager le réseau en plusieurs communautés en éliminant itérativement les liens entre les nœuds. En commençant par une seule communauté (tous le réseau), en haut du dendrogramme, jusqu'à avoir 'n' communautés à un seul nœud représentant les feuilles du dendrogramme. Dans chaque itération, tout réseau connexe est considéré comme une communauté. Les deux méthodes sont illustrées dans la figure 7.

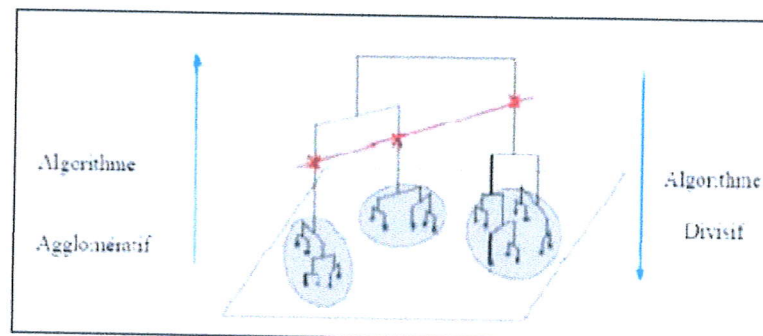


Figure 7: Dendrogramme pour un algorithme agglomératif et divisif

Les méthodes hiérarchiques ont pour avantage de trouver de petites communautés même dans un très grand graphe. Cependant, l'inconvénient avec ces approches est la détermination du seuil de la coupe du dendrogramme, ainsi que l'absence de chevauchement des classes.

4.1.2. Approches basées sur l'optimisation d'une fonction objective

La première méthode moderne pour la détection de communautés, encore utilisée dans plusieurs domaines est celle proposée en 2002 par Girvan et Newman. [GN02]

Cette méthode reprend le principe des algorithmes divisifs. Au départ, on considère que tous les nœuds du réseau appartiennent à une seule communauté. On va ensuite retirer successivement les liens, un à un, en retirant toujours celui de centralité d'intermédierité maximale. Petit à petit, le graphe devient donc non-connexe, et chacune des composantes connexes ainsi formées est une communauté. Le résultat est donc un dendrogramme, contenant à sa racine une communauté, deux à l'étape suivante, puis trois, et ainsi de suite jusqu'à ce qu'à ce que chaque nœud forme sa propre communauté (racines de l'arbre).

Pour le découpage du dendrogramme de façons optimal, il y'a eu pénétration de la modularité, qui est une métrique utilisée dans le domaine de la détection de communauté. Cette métrique pouvait être considérée comme une définition de ce qu'est une «bonne communauté». Des lors, ils eurent l'idée de chercher directement le découpage en communautés correspondant à la valeur maximale de la modularité pour un graphe donné. Le problème de la détection de communauté devenait donc un problème mathématique d'optimisation, consistant à explorer un espace de solutions pour trouver celle correspondant au maximum de modularité, celle-ci a été définie par Girvan et Newman de 2002 [GN02]. Elle se calcule comme la différence entre la proportion de liens internes au communautés et la proportion de liens qu'auraient des communautés aléatoires de même taille, cette fonction de qualité est la suivante :

$$M = \sum C_i \frac{e(C_i)}{a(C_i)} - a(C_i)^2$$

Où : $e(C_i)$ représente la proportion d'arêtes ayant les deux extrémités dans la communauté comporte le nœud i . et $a(C_i)$: est la probabilité pour qu'une arête ait une extrémité dans la communauté C_i .

Cette fonction a été utilisée dans plusieurs travaux comme [New04], [CNM04], [DDA06], [WT07].

Le problème mis en évidence est que la modularité présuppose certaines propriétés des communautés à partir des propriétés du réseau à étudier. Pour une taille de réseau donné ayant une densité donnée, la modularité ne pourra pas trouver de communautés inférieures à $\sqrt{\frac{M}{2}}$. Toute communauté plus petite, même nettement séparée du réseau, sera fusionnée avec d'autres communautés pour obtenir des communautés de la taille attendue par la modularité.

En plus, dans les grands graphes il existe un grand nombre de partitions qui ont des valeurs de modularité qui sont très proches de la modularité maximale et correspondent pourtant à des partitions très différentes.

4.1.3 Approches à base de cliques

Dans cette catégorie d'approches une communauté est définie comme une chaîne de *k*-cliques adjacentes. Une *k*-clique est un sous-ensemble de *k* nœuds tous adjacents les uns aux autres, et deux *k*-cliques sont adjacentes si elles partagent *k-1* nœuds. L'idée de ces algorithmes est que, à partir de *k*-cliques, de construire petit à petit les communautés. L'avantage immédiat d'une telle approche est la détection de communautés avec recouvrement, un nœud pouvant appartenir à plusieurs *k*-cliques non forcément adjacentes. CPM (clique méthode percolation) est la première méthode proposée. Elle constitue la base de nombreux travaux comme [palla07]. Par la suite comme l'algorithme EGALÉ [SCC09].

La méthode CPM [PBV07] :

Palla et al proposent l'algorithme CFinder qui est structuré en trois principales étapes:

- Calculer l'ensemble de cliques de taille *k* (paramètre de l'algorithme) dans le graphe cible *G*.
- Construire un graphe de cliques où chaque clique est représentée par un nœud. Deux nœuds sont connectés par un lien si les deux cliques associées partagent *k-1* nœuds dans le graphe *G*.
- Les communautés dans le graphe *G* sont alors les composantes connexes identifiées dans le graphe de cliques construit à l'étape 2. (Voir figure 8)

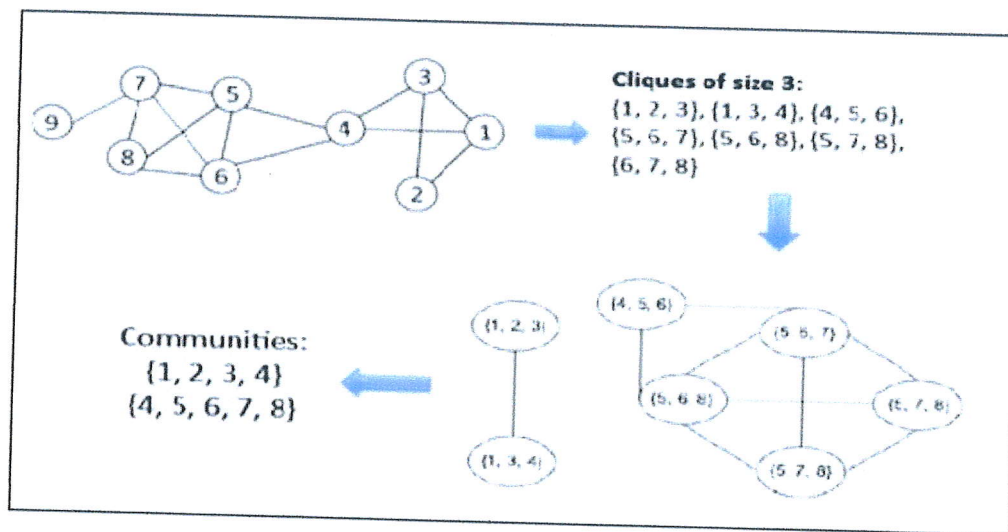


Figure 8 : Exemple de l'algorithme de percolation de cliques avec $k=3$

Cette approche est conceptuellement simple. Aussi son avantage est qu'elle fonctionne remarquablement bien dans les réseaux de terrain. Cependant, sur des graphes ayant une structure arborescente, la méthode ne donne pas des résultats fiables. Une autre limitation est liée au paramétrage : la valeur de k (la taille des communautés à considérer). De plus, cette méthode est sensible à certaines configurations. Par exemple, si l'on imagine une suite de cliques de taille k , ayant $(k-1)$ nœuds en commun, et formant une chaîne, cette méthode les détectera comme une communauté, alors que cela n'est généralement pas pertinent.

4.2. Approches dynamiques

L'étude des communautés dynamiques est menée dans deux grandes directions : étudier les communautés parmi différentes captures en utilisant les algorithmes adaptés pour les graphes statiques ou utiliser directement l'information temporelle lors de la détection.

4.2.1 Approches travaillant sur des instantanés du réseau

Il existe deux catégories des approches travaillant sur des instantanés du réseau. La première catégorie est celle des approches par détection statique successive. L'idée est de considérer le graphe dynamique comme une succession d'instantanés indépendants. Une première étape consiste donc à appliquer un algorithme statique sur chacun de ces instantanés, ce qui permet d'obtenir une série de partitions, une pour chaque instantané. Ensuite de trouver une correspondance (association) entre les communautés existantes dans des instantanés consécutifs. De nombreux travaux rentrent dans cette catégorie [PBV07] [WF10] [Chen10].

Algorithme de Chen et al. [Chen10]

Chen et al utilisent des nœuds cœurs définis comme les nœuds existant aux instants $t - 1$, t et $t + 1$ pour réduire le nombre de communautés à considérer. Ils définissent les communautés initialement comme les cliques maximales et peuvent donc avoir des communautés recouvrantes. Néanmoins, le nombre de communautés peut être élevé et ils utilisent la notion de nœuds cœurs pour ne considérer que les communautés contenant des nœuds cœurs et donc diminuer leur nombre.

Evidemment, la faiblesse principale de cette méthode est qu'elle définit les communautés comme étant des cliques maximales, ce qui, d'une part, donne souvent des communautés peu pertinentes et, d'autre part, conduit à un nombre bien trop important de communautés dans des grands graphes ayant une densité élevée.

La deuxième catégorie est celle des approches par détection statique informées successives. Elles proposent de prendre en compte les résultats obtenus à l'instant t lors de la détection des communautés à l'instant $t + 1$. Ceci permet de réduire l'instabilité des algorithmes c-à-dire que l'algorithme donne des résultats assez éloignés pour des graphes très proches. En imposant le choix entre deux découpages différents, et pourraient prendre le plus semblable au découpage précédent, cette approche est appliquée dans les travaux. [Lin09] [XSK13] [Wan12]. Par la suite nous avons détaillé un algorithme qui utilise la deuxième catégorie.

Algorithme Lin et al. [Lin09]:

Cet algorithme propose une solution basée sur un modèle génératif probabiliste, constituant à formuler une fonction de qualité comme un problème de factorisation de matrices non négatives qui optimise conjointement la qualité et la stabilité de communautés.

Bien que cette méthode ait l'avantage de permettre la détection de communautés recouvrantes Elle impose cependant de fortes contraintes : le nombre de communautés doit être connu à l'avance, et il n'est a priori pas possible d'ajouter ou de supprimer de nœuds au cours du temps. Elle ne permet pas non plus d'opération telle que la fusion ou la division de communautés. (Voir la figure 9)

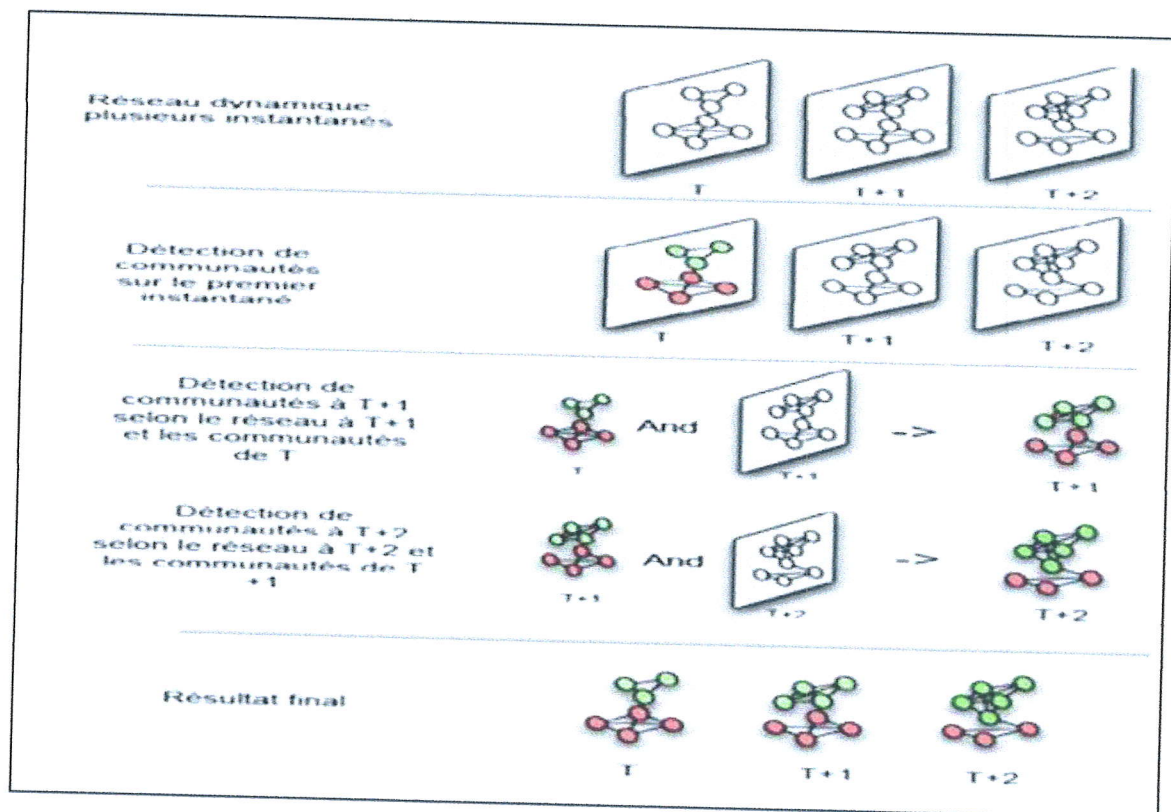


Figure 9 : Représentation d'une méthode par détections statiques informées successives

4.2.2 Approche travaillant sur les réseaux temporels

Dans les algorithmes travaillant directement sur le réseau temporel, l'évolution du réseau est considérée comme une succession de modifications sur le réseau. L'idée est donc de prendre en compte la ou les dernières modifications effectuées sur le réseau, et de modifier les communautés existantes en conséquence. Son avantage est qu'il y'a plus de problème d'instabilité. Il y'a deux algorithmes qui travaillent a base de cette méthode et qui traitent le cas de recouvrement .Nous allons les détailler dans la section suivante.

L'algorithme de Falkowski et al. [FBS08] :

Falkowski et al définissent tout d'abord une distance entre les nœuds d'un réseau puis définissent le voisinage d'un nœud comme la boule topologique de rayon R (variant entre 0 et 1). Ils ne considèrent que les nœuds dont le voisinage est plus grand qu'une limite donnée S et les considèrent comme des nœuds cœurs. Les nœuds présents dans le voisinage d'un nœud cœur sont des nœuds frontières. Ils définissent ensuite les communautés comme les unions des voisinages partageant des nœuds.

Ils proposent ensuite des techniques pour mettre à jour les voisinages et les communautés : à chaque nouvelle étape, les valeurs de distance entre les nœuds sont mises à jour. Si l'une de ces modifications fait apparaître un nouveau nœud cœur, celui-ci est intégré à une nouvelle communauté, ou on crée une nouvelle s'il n'a pas de nœud cœur dans son voisinage. De même, si les nœuds qui étaient dans le rayon r d'un nœud cœur passent au-delà de ce rayon, ils quittent la communauté, et ainsi de suite.

Cependant, le problème de cette solution est que la définition de communauté utilisée est très particulière, et est assez éloignée de ce que l'on considère généralement comme une bonne communauté. De plus, elle dépend toujours des valeurs choisies de paramètres R et S .

L'Algorithme de Cazabet [Caz13] :

L'algorithme ILCD (Détection Longitudinale de Communautés) proposé par Cazabet [Caz13], cet algorithme est innovateur parce qu'il était l'un des premiers algorithmes à pouvoir détecter les communautés dynamiques dans un réseau temporel, c-à-dire, les communautés qui changent d'une façon réaliste quand le réseau évolue.

ILCD, décrit la façon dont la détection de communautés doit être effectuée et il représente l'ensemble des actions effectuées (Te) sur les réseaux temporels. La séquence des actions Te est composée de quadruplets (i, j, a, t) avec i et j les nœuds affectés, a l'action ayant lieu, pouvant être un ajout ou une suppression de lien, et t l'instant auquel cette modification a lieu. Ces actions sont ordonnées par t croissants. Le principe d'ILCD est le suivant :

- Pour chaque ajout d'un lien (i, j) dans le réseau, si i est dans une communauté C et j n'appartient pas à C , on regarde si j doit être intégré à C .
- On trouve ensuite l'ensemble des nouvelles communautés formées par l'apparition de ce nouveau lien. Les nouvelles communautés n'étant pas incluses dans une communauté existante sont conservées.
- Pour chaque suppression d'un lien (i, j) , si ce lien se trouve à l'intérieur d'une Communauté $(i \in C \wedge j \in C)$, on calcule si C perd un ou plusieurs nœuds, ce qui peut l'amener à se diviser.
- Après chaque modification du réseau, si une ou plusieurs communautés ont été modifiées. On regarde si elles doivent être fusionnées avec d'autres communautés.

Les communautés candidates sont celles qui partagent des nœuds avec la communauté modifiée.

L'algorithme ILCD utilise deux métriques :

Représentativité : est une métrique qui indique à quel point un nœud est représentatif d'une communauté. Elle est définie par la fonction suivante.

$$RP(i, c) = \frac{d^{int}_{c(i)}}{d(i)}$$

Où : $d^{int}_{c(i)}$: est le degré interne du nœud i dans la communauté c .

$d(i)$: est le degré total du nœud i .

Force d'appartenance : elle est quantifiée par un nœud i et une communauté C . Sa valeur est bornée entre 0 et 1. Elle est définie par la fonction suivante.

$$FA(i, c) = \sum_{j \in N_c(i)} RP(j, c)$$

Où : $N_c(i)$: sont les nœuds de la communauté où se trouve le nœud i .

ILCD n'est pas basé sur la modularité, mais au contraire sur l'idée que les communautés sont définies localement. Cet algorithme permet de suivre les opérations des communautés (naissance, mort, division, mort). L'avantage de cet algorithme est qu'il n'y a plus de problème d'instabilité. Cependant cet algorithme présente quelques limitations. Premièrement il ne prend pas en compte l'aspect hiérarchique de plus il est applicable que sur les graphes non orientés. En plus, ILCD ne fait pas d'opération de division au sens réel du mot mais il la traite de façon partielle.

5. Synthèse de l'état de l'art

Une comparaison entre les différentes approches est fournie dans le tableau ci-dessous (Tableau 1). Nous avons considéré les critères suivants :

1. **Instabilité** : l'algorithme donne des résultats assez éloignés pour des graphes très proches.
2. **Hiérarchique** : l'algorithme permet une détection de communauté à plusieurs niveaux.
3. **Grand graphe** : l'algorithme est applicable sur des graphes complexes.
4. **Itératif** : le résultat final fourni par l'algorithme n'apparaît qu'après plusieurs exécutions.
5. **Paramétrage** : l'algorithme nécessite un ou plusieurs paramètres d'entrée.
6. **Recouvrement de communautés** : l'algorithme permet de détecter l'appartenance multiple d'un nœud à plusieurs communautés.

Chapitre 2 : Etat de l'art

On a utilisé aussi les symboles suivants :

K : taille des communautés initiales

R : rayon du nœud

S : limite voisinage

Algorithmes Critères	Approches statiques				Approches dynamiques			
	Girvan Et Newman 2002	CPM 2007	Lin Et All 2008	Eagle 2009	Palla 2007	Falko- wski 2008	Li Et All 2012	ILCD 2013
Recouvrement	+	+	+	+	+	+	-	+
Instabilité	-	-	-	-	+	+	+	-
Hiérarchique	-	-	-	-	-	-	-	-
Grand graphe	-	+	-	+	+	+	+	+
Itératif	+	+	-	+	-	+	+	-
Paramétrage	-	k	-	-	-	S, R	+	-

Tableau 1 : comparaison entre les approches de détection de communautés

À partir de l'état de l'art, nous avons constaté qu'il existe un grand nombre d'algorithmes itératifs pour la détection de communautés. Ces algorithmes sont souvent coûteux en exécution.

En outre, dans les méthodes nécessitant un paramétrage comme CPM [PBV07], les résultats dépendent des paramètres en entrée. Cependant, cette méthode (CPM) est parmi les premières à pouvoir détecter les communautés recouvrantes et à être appliquée dans les grands graphes à l'opposée des approches traditionnelles comme Girvan et Newman [GN02].

L'état de l'art montre qu'il ya deux approches pour la détection des communautés dynamiques. La première approche considère le graphe comme une succession d'instantanés, son avantage certain est la réutilisation des algorithmes statiques. Cependant, cette approche a une limitation très claire c'est l'instabilité. Dans la deuxième approche travaillons sur des graphes temporels, il n'ya plus de problème d'instabilité. Néanmoins, l'évolution des

communautés n'est pas prise en compte. Par exemple, dans le travail de Cazabet [Caz13] l'opération de division est partialement implémentée.

6. Conclusion :

Dans notre chapitre nous avons examinés les différentes approches statiques et dynamiques de la détection de communautés. Trois approches statiques ont été présentées : les approches hiérarchiques, les approches basées sur l'optimisation de la modularité et les approches à base de clique. Cette dernière est très simple et elle est la première à pouvoir détecter les communautés recouvrantes.

Pour les approches dynamiques, nous avons distingué deux catégories. La première considère le réseau comme instantané et réutilise les algorithmes statiques. Elle souffre du problème d'instabilité. La deuxième approche utilisant les réseaux temporels élimine le problème d'instabilité. Cependant, elle ne permet pas de suivre toutes les étapes d'évolution des communautés. Notre proposition qui sera décrite dans le chapitre suivant, s'intéresse à cette dernière catégorie et tente toutefois à suivre toute l'évolution de communautés.

Chapitre 3

Proposition

Nous présentons dans ce chapitre notre proposition qui consiste à une nouvelle approche de détection de communautés au sein des réseaux temporels. Nous décrivons tous d'abord les exigences de notre algorithme baptisé TCCD (Temporel Communauté Clique Détection) et son principe de fonctionnement. Puis, nous présentons le pseudo algorithme sur le quel se base notre proposition.

1. Les exigences de TCCD

Avant de présenter notre proposition qui consiste à un algorithme de détection de communautés baptisé TCCD (Temporel Communauté Clique Détection), nous allons d'abord décrire les exigences auxquelles répondra notre algorithme.

➤ **Réseaux temporels :**

Pour suivre les réseaux au fur et à mesure de leur évolution, on va se baser sur les réseaux temporels, qui sont définis dans la section (2.2.2) du premier chapitre. Nous avons choisi de modéliser le réseau sous forme d'une suite d'événements. Un événement est un quadruplet :

- i, j : deux nœuds affectés par l'action.
- **action** : l'action pouvant être la création ou la suppression de lien.
- t : l'instant où l'action aura lieu.

Ainsi, l'événement (i, j, action, t) décrira l'action **action** (création ou suppression de lien) entre les nœuds i et j à l'instant t . Nous avons choisi de nous limiter aux réseaux non dirigés et non pondérés. Les actions (i, j, action, t) et (j, i, action, t) sont donc équivalentes.

➤ **Détection recouvrante :**

Notre algorithme sera capable de détecter l'appartenance multiple des nœuds à plusieurs communautés. Pour cela nous avons repris le principe de détection de communautés à base de cliques. Cette méthode est capable de détecter les communautés recouvrantes.

➤ **Traitement des grands graphes :**

Notre algorithme sera destiné pour traiter les grands graphes de terrains.

2. Principe de TCCD

Notre algorithme TCCD est basé sur la notion de clique. On considère que les communautés initiales sont des *k-clique*. Le paramètre k désigne la taille de la clique, sa valeur par défaut égale à trois. TCCD exploite l'information temporelle dans le graphe temporel pour suivre son évolution. Celle-ci comprend deux événements : l'ajout ou la suppression d'un lien. Nous avons distingué trois types de liens.

Chapitre 3 : Proposition

1. Lien interne : représente un lien entre deux nœuds appartenant à la même communauté.
2. Lien externe : il s'agit d'un lien dont une ou les deux extrémités ne sont pas encore intégrées à une communauté.
3. Lien intra-communautés : il s'agit d'un lien dont les deux extrémités appartiennent à des communautés différentes.

Pour chaque événement et par type de lien, nous avons distingué les cas suivants :

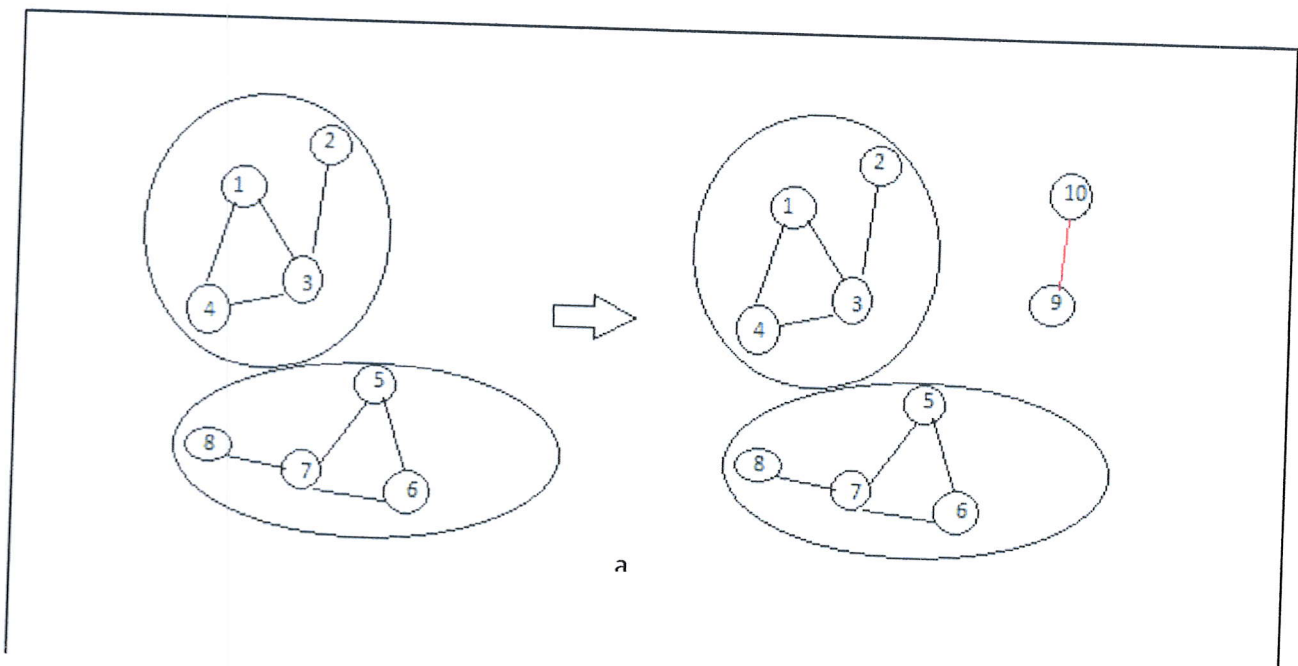
2.1. Ajout d'un lien dans le graphe

Lors de l'ajout d'un lien on doit vérifier l'existence de ses deux extrémités. Si ces dernières n'existent pas, alors on les ajoute d'abord au graphe G puis on ajoute le lien.

2.1.1. Lien externe :

- a. Si les deux extrémités du lien n'appartiennent à aucune communauté, on vérifie s'il y'a une formation de k -clique avec l'ajout de ce nouveau lien. Si oui, il y'a naissance de nouvelles communautés contenant la k -clique. Puis, on vérifie le degré de tous les nœuds. Si le degré de chaque nœud relié à cette communauté est supérieur ou égale à son degré externe, ce nœud sera intégré à cette communauté. (Voir figure 10)

Exemple :



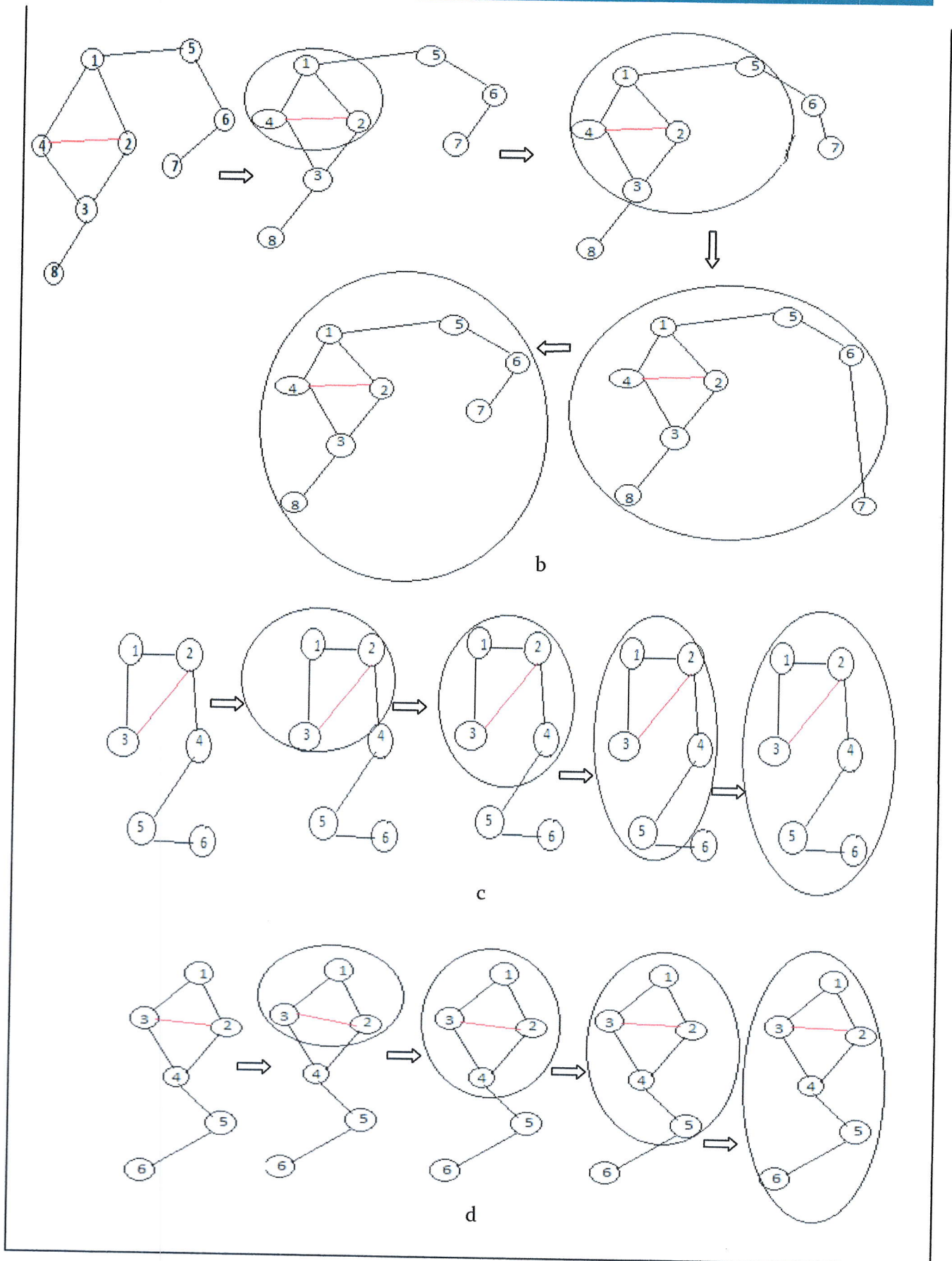


Figure 10: Ajout d'un lien externe pour les cas (a, b, c, d)

Chapitre 3 : Proposition

- b. Si l'une des extrémités du lien appartient à une ou plusieurs communautés et l'autre extrémité n'appartient à aucune communauté, on vérifie s'il y a formation de k -clique avec l'ajout de ce nouveau lien. Si oui on crée une nouvelle communauté. Ensuite, on vérifie le degré des nœuds externes reliés à cette communauté, en commençant par les voisins puis les voisins de voisins etc. si le degré du nœud est supérieur ou égale à son degré externe, ce nœud sera intégré à la communauté. Dans le cas contraire où il n'existe pas une k -clique, on vérifie le degré de l'extrémité externe puis de chacun de ses voisins dans les communautés de l'autre extrémité, s'il est supérieur ou égale à son degré externe, ce nœud sera intégré à la communauté examinée. (Voir figure 11)

Exemple :

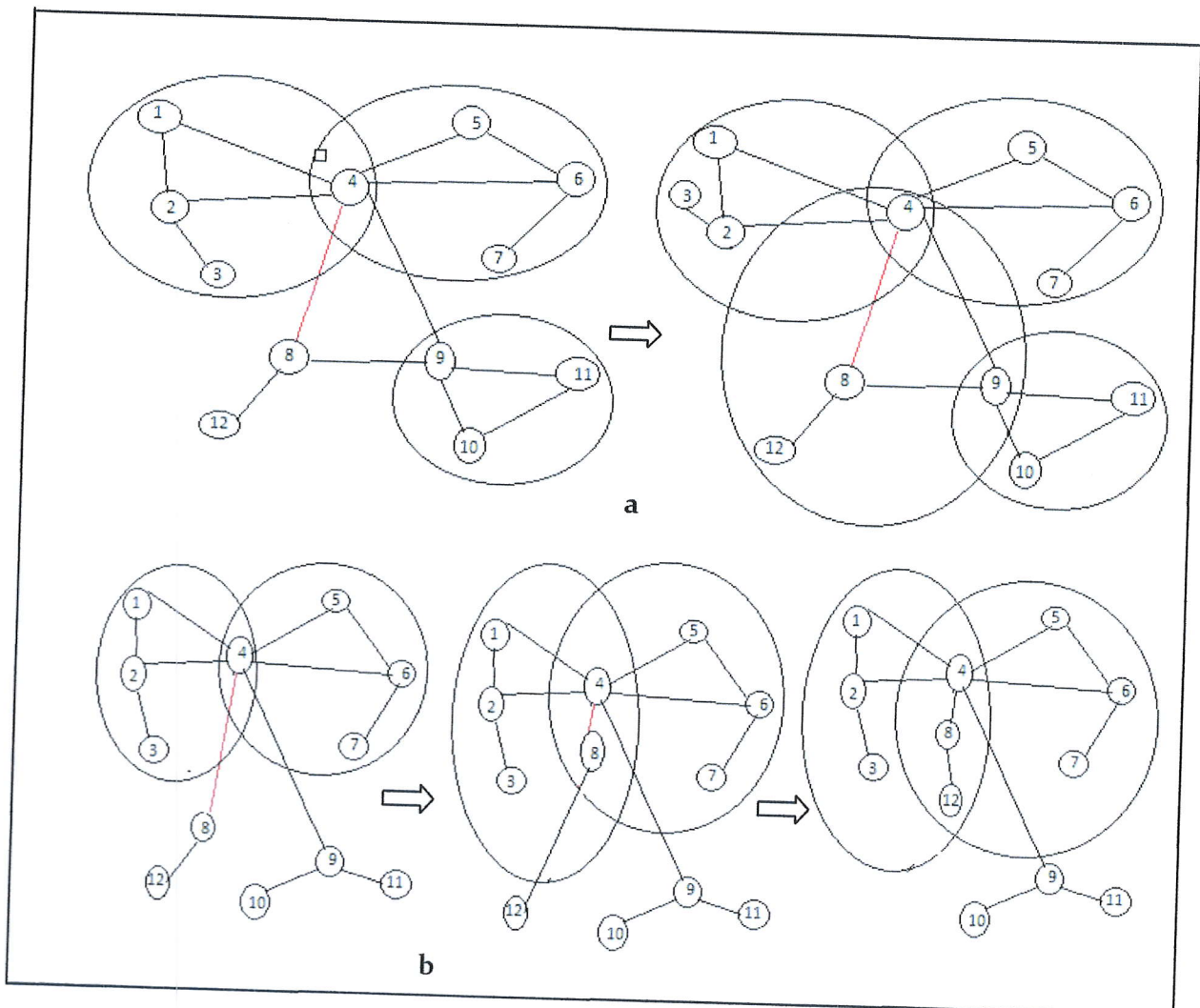


Figure 11: Ajout d'un lien externe

Chapitre 3 : Proposition

2.1.2. Lien interne :

Si les deux extrémités du lien appartiennent à une ou plusieurs mêmes communautés. Il y'aura une croissance de cette ou de ces communautés. (Voir figure 12)

Exemple :

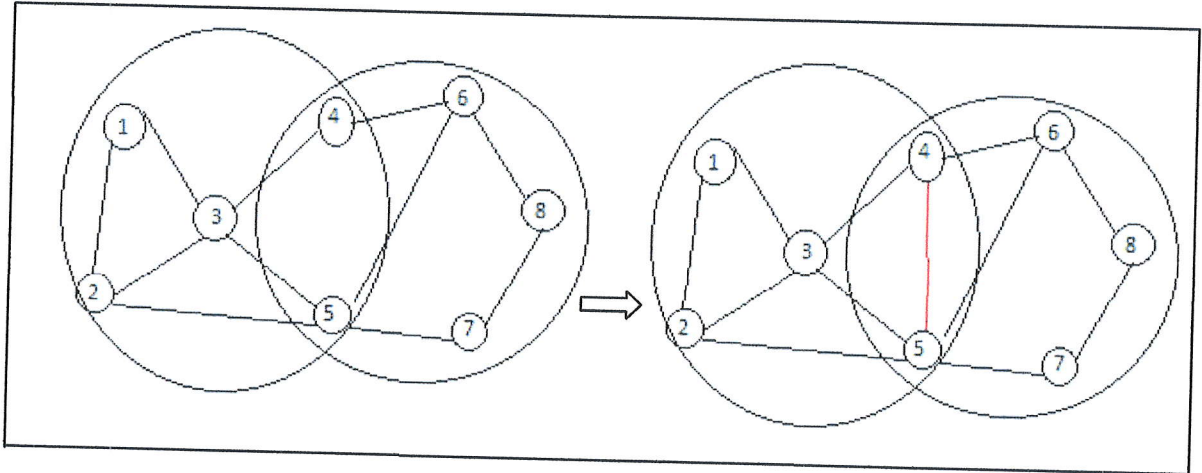


Figure 12: Ajout d'un lien interne

2.1.3. Lien intra-communautés :

Si les deux extrémités du lien appartiennent à des communautés différentes, alors on vérifie le degré des deux extrémités par rapport à toutes les communautés c'est-à-dire ; le degré de la première extrémité dans toutes les communautés de la deuxième extrémité. Si ce degré est supérieur ou égale à son degré interne alors on ajoute cette extrémité dans la communauté examinée. On fait la même chose avec la deuxième extrémité. (Voir figure 13)

Exemple :

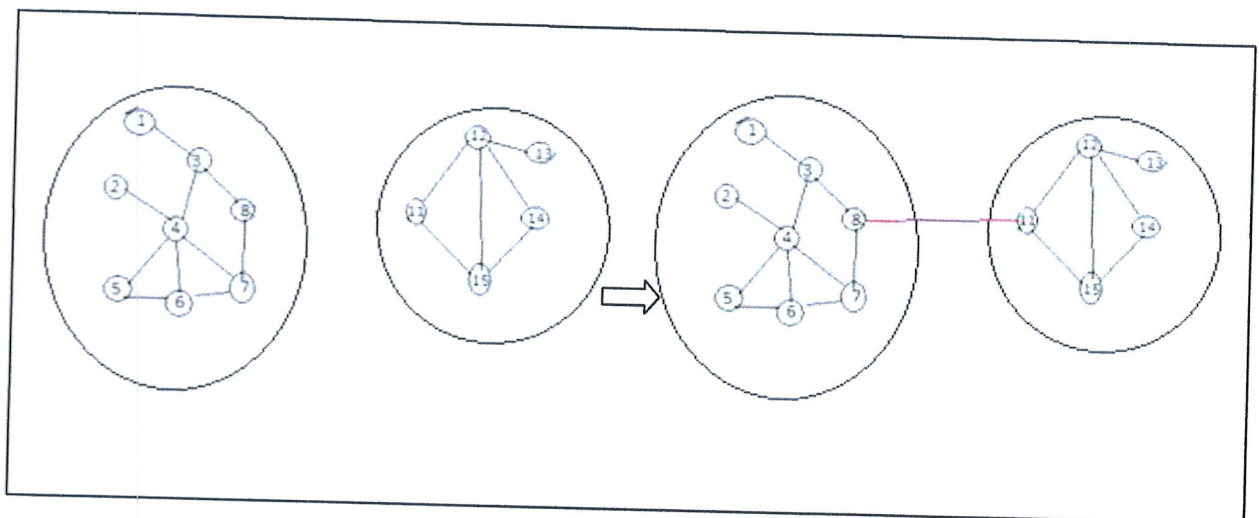


Figure 13: Ajout d'un lien intra-communautés

2.2. Suppression d'un lien dans le graphe :

2.2.1. Le lien externe :

Si les deux extrémités d'un lien n'appartiennent à aucune communauté ou l'une appartient et l'autre extrémité n'appartient à aucune communauté, alors on supprime le lien. On vérifie ensuite le degré de chaque extrémité de ce lien, s'il est égal à zéro alors, on supprime cette extrémité. (Voir la figure 14)

Exemple:

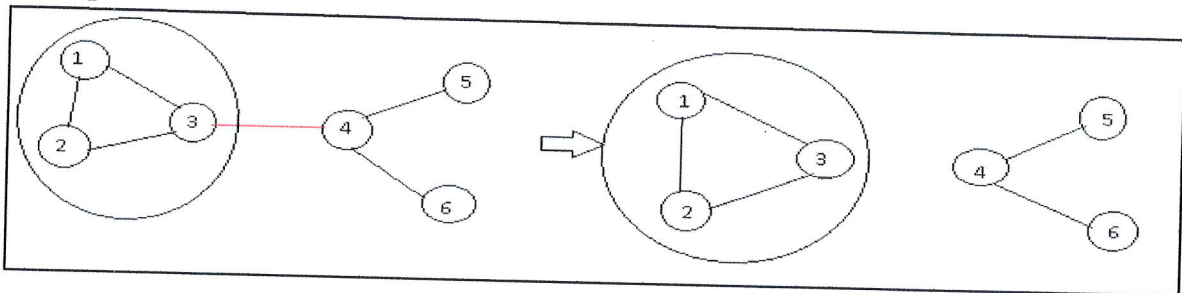
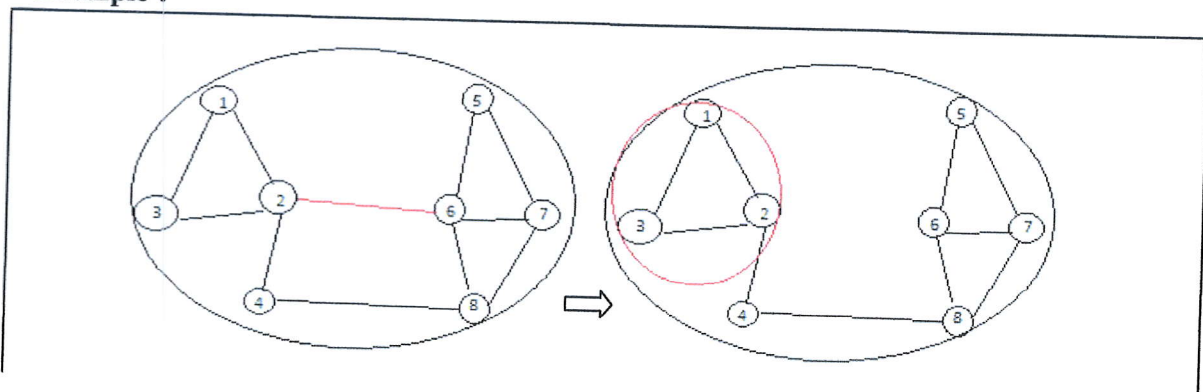


Figure 14: Suppression d'un lien externe

2.2.2. Le lien interne :

Si les deux extrémités du lien appartiennent aux mêmes communautés, alors on supprime le lien. Ces communautés sont amenées à se diviser comme suit : dans chaque communauté (C) on vérifie s'il existe une k -clique. Cette k -clique sera affectée à une nouvelle communauté (CI). Puis, on vérifie le degré de chacun des nœuds voisins de CI appartenant à C . Si le degré du nœud est supérieur ou égale à son degré externe, on va l'intégrer à CI et on refait le même traitement (vérification de degré) pour les voisins de voisins. Ensuite, lorsqu'il y aura plus de voisins à intégrer, on fait une contraction de la communauté C avec CI . On continue avec le même traitement (recherche de k -clique) sur c jusqu'il n'aura pas de k -clique. Après, on vérifie le degré des nœuds restants dans la communauté C . Si le degré du nœud est égal à zéro alors, on supprime ce nœud. Ensuite il y aura la mort de la communauté C et le traitement se passe aux autres communautés. (Voir figure 15)

Exemple :



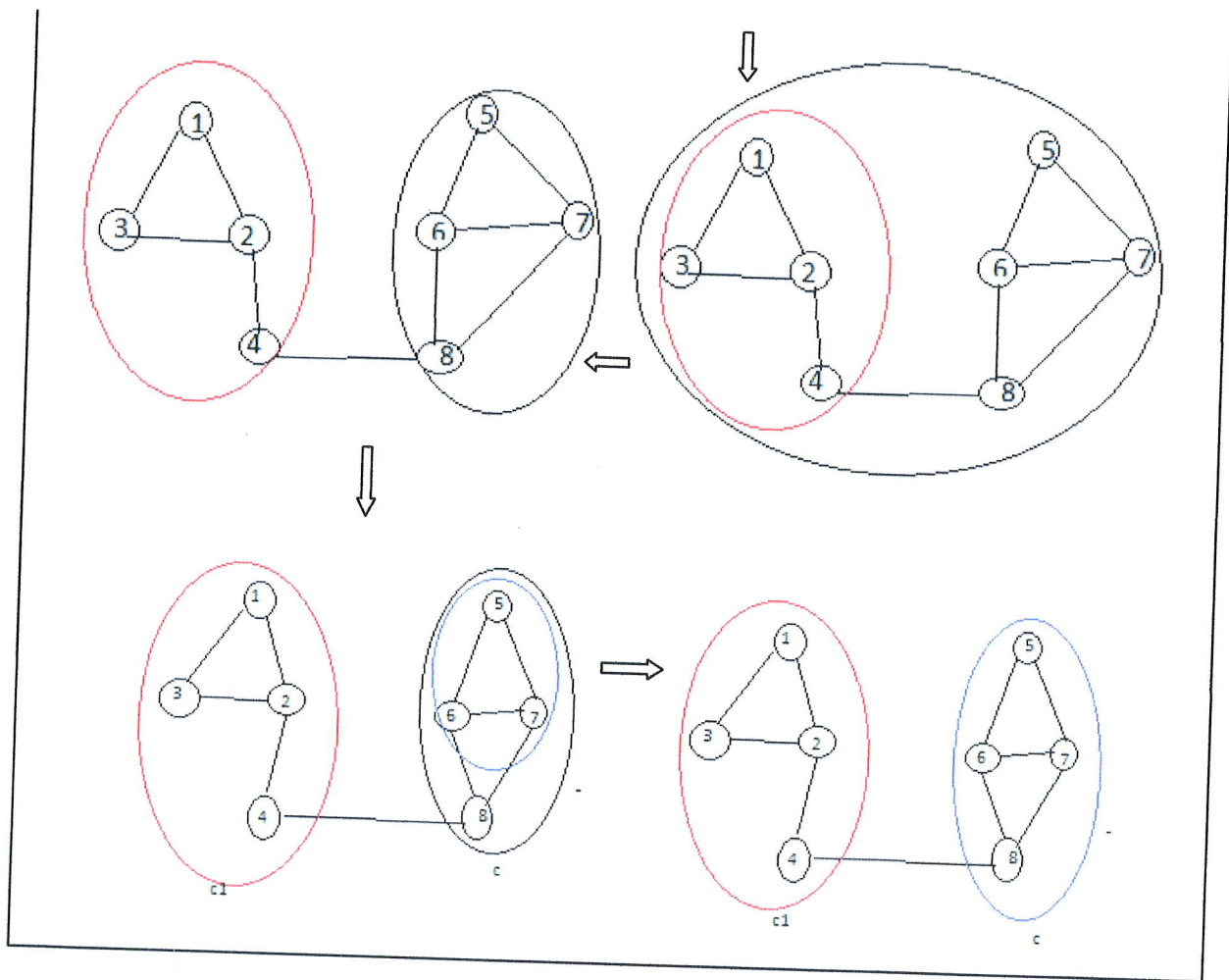


Figure 15: Suppression d'un lien interne

2.2.3. Le lien intra-communautés :

Si les deux extrémités du lien appartiennent à des communautés différentes, alors on supprime le lien. (Voir figure 16)

Exemple :

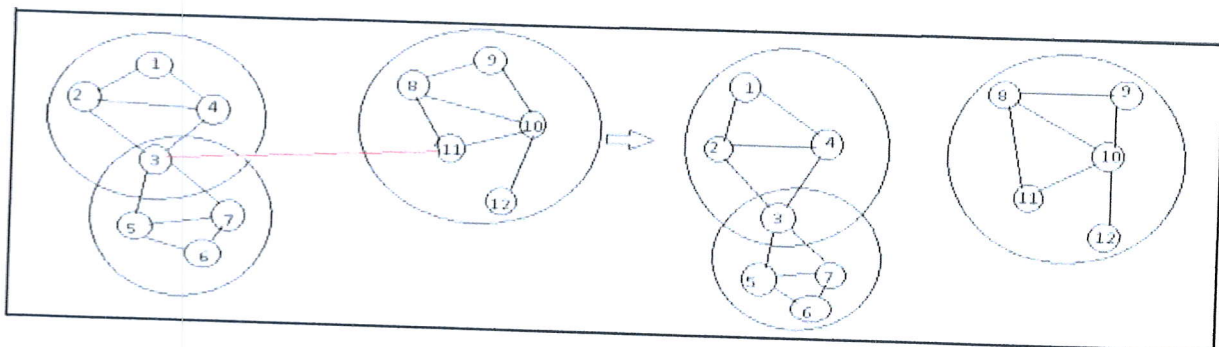


Figure 16 : Suppression d'un lien intra-communautés

3. Pseudo algorithme TCCD

Dans cette section nous décrivons le pseudo algorithme de notre outil de détection de communauté TCCD. Nous donnerons aussi la signification des symboles utilisés dans cet algorithme.

action : l'action entre deux nœuds, comprenant l'ajout ou la suppression d'un lien.

E : l'ensemble des liens du graphe G .

V : l'ensemble initial des nœuds du graphe G .

V_i : les voisins des nœuds de la k -clique.

C : l'ensemble des communautés vivantes.

M : l'ensemble des communautés mortes.

$e(i, j)$: un lien ayant les extrémités i et j .

$d(i)$: le degré de nœud i .

$d(i)^{in}$: degré interne du nœud i par rapport à une communauté c .

$d(i)^{out}$: degré externe du nœud i dans une communauté c

k -clique : est une fonction qui cherche une k -clique de la taille k égale à trois, contenant le lien $e(i, j)$.

i : un nœud du graphe.

c : une communauté.

Nous avons comme entrée pour notre algorithme le graphe ' G ', l'ensemble des nœuds du graphe ' V ' et nous avons aussi l'ensemble des liens du graphe ' E '.

Nous avons comme sortie pour notre algorithme l'ensemble des communautés vivantes ' C ' et aussi l'ensemble des communautés mortes ' M '.

Début

Si (action= ajouter lien e (i, j)) alors

Si (i \notin V) alors

 V := V U {i};

Fin si

Si (j \notin V) alors

 V := V U {j};

Fin si

 E := E U {i, j};

Si (k_clique (V) pas vide) alors *\ \ recherche les k-clique contenant les nœuds i, j.*

 c := k-clique ;

 Naissance (c) ;

 Intégré _ voisins (c);

Fin si

Fin si

Si (action= supprimer lien e (i, j)) alors

 E := E /{e (i, j)} ;

Si (d(i) = 0) alors

 V := V / {i};

Fin si

Si ((d(j)= 0) alors

 V := V / {j};

Fin si

Pour (c \in C) faire

Si (k_clique(V) vide) alors

 Mort (c) ;

Sinon

Pour chaque (k_clique) faire

 c' :=k-clique ;

 Naissance (c') ;

 Intégré _ voisins (c') ;

Fin pour

Chapitre 4

Implemetation

et

validation

Nous allons décrire dans ce chapitre, l'environnement de développement et les différentes technologies et les outils utilisés pour développer notre outil de détection de communauté. Par la suite, nous montrons le développement, puis nous présentons une vue d'ensemble de son fonctionnement. Au final, nous allons présenter les résultats des expérimentations de l'algorithme proposé sur des réseaux de test réels. La performance de notre algorithme est comparée avec celles des algorithmes connus dans la littérature.

1. Développement de l'outil TCCD

1.1. Environnement de développement

1.1.1. IDE Netbeans



Netbeans est un environnement open-source de développement intégré (IDE) java .En 1997, Netbeans naît de Xelfi, un groupe d'étudiants de l'Université Charles de Prague, racheté par Sun Microsystems en 1999. Il présente toutes les caractéristiques Indispensables à un environnement de qualité, que ce soit pour développer en Java, C/C++ ou même PHP. Netbeans comprend toutes les caractéristiques d'un IDE moderne (coloration syntaxique, projets multi-langage, éditeur graphique d'interfaces et de pages Web, auto complétion, gestion multi-projets, intégration des documents de référence, support de solutions de gestion de versions ...). Il est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), MacOS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java).Un environnement JDK est requis pour les développements en Java.

1.1.2. Language Java



JAVA est un langage de programmation orienté objet et une plateforme informatique, qui a vu le jour en 1995, comme résultat d'un projet interne SUN¹ de création d'un langage de haut niveau. Le projet stipulait que le langage à créer devrait être sûr, portable, facile à apprendre, interprété, et suffisamment ouvert pour permettre son extension aisée. Il est utilisé sur plus de 850 millions d'ordinateurs de bureau et plus d'un milliard de périphériques dans le monde, dont des périphériques mobiles et des systèmes de diffusion télévisuelle.

¹ De La Société Sun Microsystems.

1.1.3. La bibliothèque GraphStream



La partie affichage du graphe temporel fut la plus aisée car la bibliothèque GraphStream permet de générer un «Viewer» (un panel où sera dessiné le graphe) qui consiste à représenter l'évolution des communautés sous la forme d'une vidéo. GraphStream² qui est une librairie Java pour la modélisation, le traitement et la visualisation de graphes dynamiques. Elle offre l'opportunité d'afficher les modifications apportées sur le graphe directement dans le viewer sans avoir recours à une méthode de rafraîchissement de celui-ci. La visualisation est automatiquement générée à partir des fichiers CTNF, que nous décrivons dans la section suivante. L'inconvénient de cet outil est qu'il ne permet pas de visualiser les graphes très denses et comportant beaucoup de nœuds.

1.2. Formats des fichiers utilisés

1.2.1. Format des fichiers d'entrée

L'algorithme TCCD prend en entrée un réseau qui est représenté dans un fichier. Ce fichier de type LND (3D Landscape Data). Ce fichier contient des actions effectuées sur le réseau à une date au format « AAAAMMJJ ». Le « + » représente l'apparition d'un lien, et le « - » définit la disparition d'un lien. Le lien est représenté avec deux nœuds. (Voir figure 17)

Date de l'action	Opération	Nœuds du lien
19930309		mamie cohe
19930314		narcis camou
19930314		mamie cohe
19930326		pitou azul
19930326		narcis camou
19931202		calla azul
19931209		neige fraise
19931210		pitou calla
19931210	Ajouter lien	sky pitou
19931213		sky calla
19940111		sky azul
19940111		neige luz
19940115		luz fraise
19940115		verace eglant
19940115		verace julia
19940124		julia eglant
19940213		pom goto
19940213		pom may
19940218		julia eglant
19940218	Supprimer lien	may fraise
19940218		nice blou
19940218		nice may
19940218		may blou

Figure 17: Fichier d'entrée

² <http://graphstream-project.org/>

Chapitre 4 : Implémentation et validation

1.2.2. Format du fichier de sortie

Le fichier de sortie de notre application contient le résultat de la détection de communautés dynamiques dans le réseau temporel. Ce fichier présente les communautés vivantes et les communautés mortes. Chaque ligne de ce fichier définit une communauté, celle-ci est décrite avec son identifiant, sa date de naissance ou de mort et ses nœuds. (Voir la figure 18)

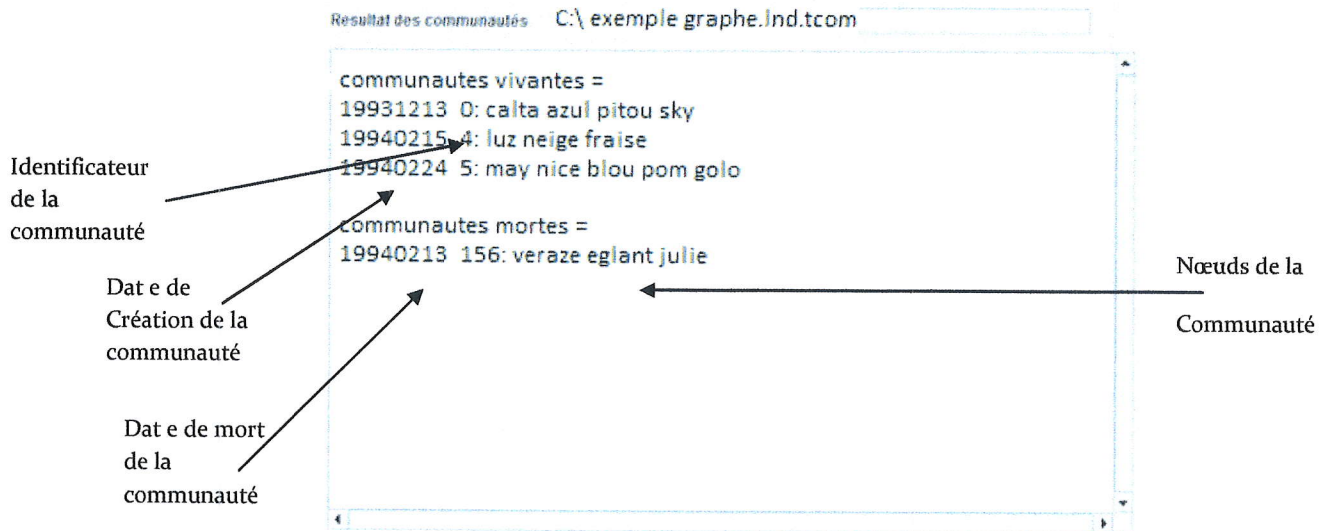


Figure 18: Fichier de sortie

1.3. Présentation de l'outil

1.3.1. Vue générale de l'outil

La figure 19 illustre le menu général de Notre outil.



Figure 19: Menu principale de l'outil TCCD

Chapitre 4 : Implémentation et validation

L'exécution de l'outil TCCD se fait selon les étapes suivantes :

- 1) La première étape consiste à ouvrir le graphe temporel. (voir figure 20 et 21)



Figure 20: Le menu fichier de TCCD

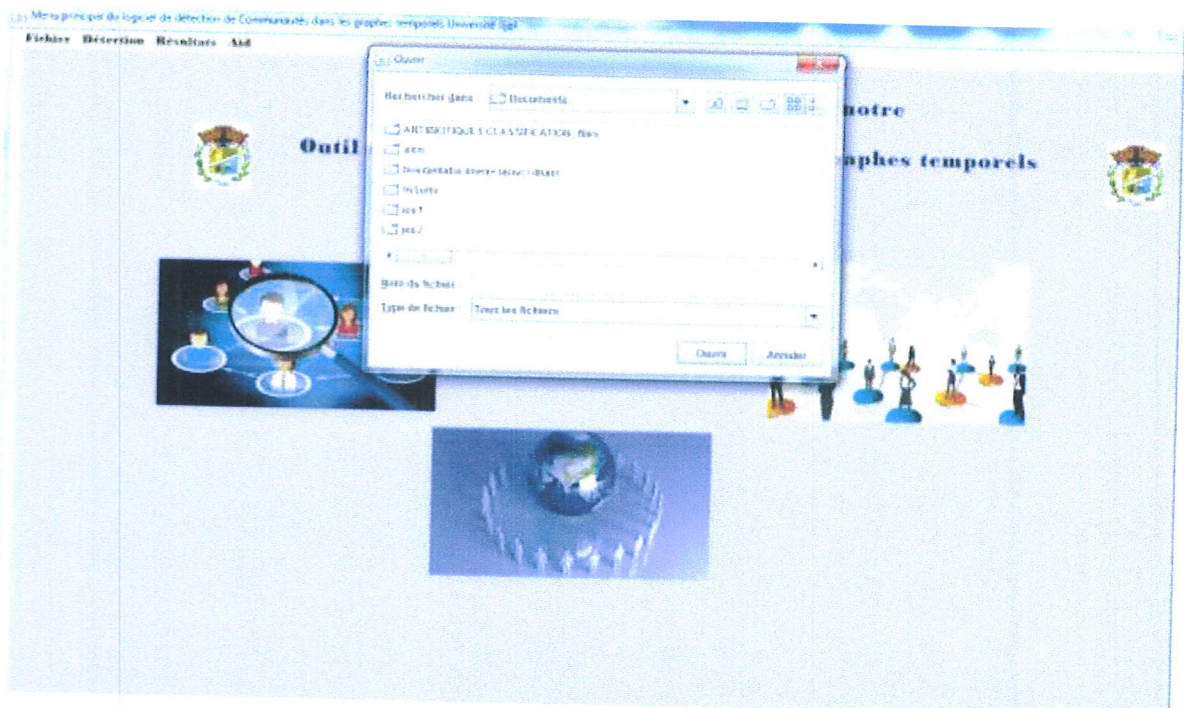
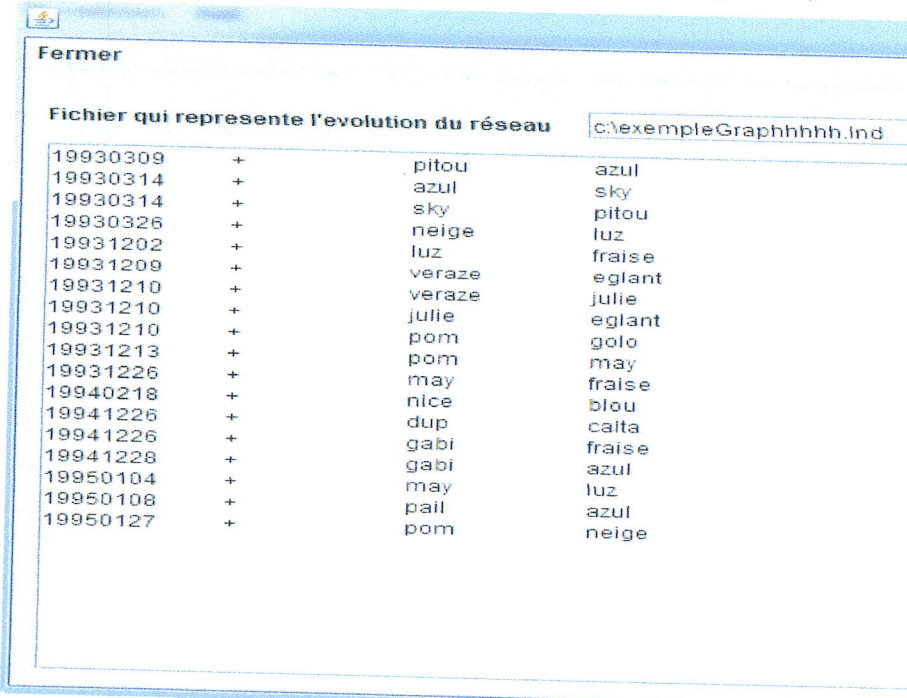


Figure 21: Choix du réseau à traiter

Chapitre 4 : Implémentation et validation

Voici un exemple de fichier d'entrée à analyser. (Voir figure 22).



Fichier qui représente l'évolution du réseau

19930309	+	pitou	azul
19930314	+	azul	sky
19930314	+	sky	pitou
19930326	+	neige	luz
19931202	+	luz	fraise
19931209	+	veraze	eglant
19931210	+	veraze	julie
19931210	+	julie	eglant
19931210	+	pom	golo
19931213	+	pom	may
19931226	+	may	fraise
19940218	+	nice	blou
19941226	+	dup	caita
19941226	+	gabi	fraise
19941228	+	gabi	azul
19950104	+	may	luz
19950108	+	pail	azul
19950127	+	pom	neige

Figure 22: Visualisation du fichier d'entrée

- 2) La deuxième étape consiste à détecter les communautés dans le réseau choisi. L'outil fournit ensuite à l'aide de la bibliographie GraphStream une visualisation graphique de communautés détectées. (Voir figure 23 et 24 et 25)



Figure 23: Le menu de détection

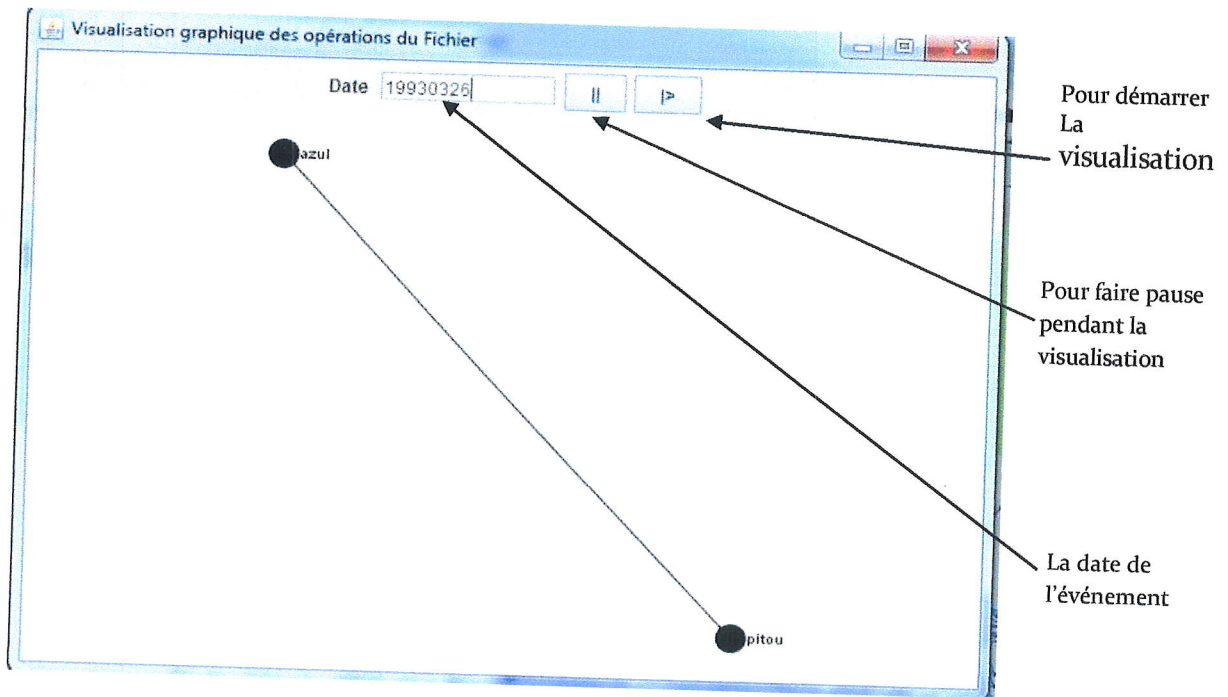


Figure 24: Début de la détection

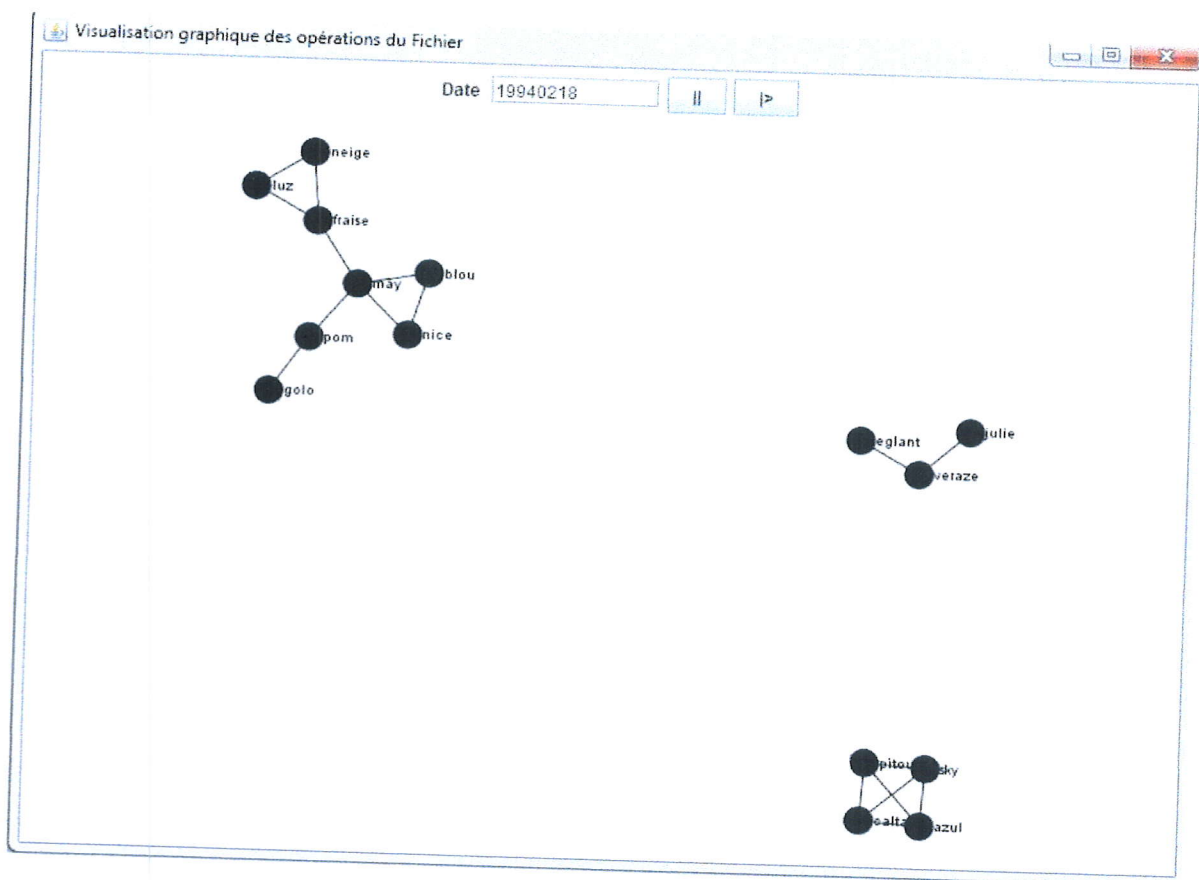


Figure 25: Résultat final de la détection

Chapitre 4 : Implémentation et validation

- 3) La dernière étape consiste à afficher le fichier de sortie qui représente les communautés vivantes et les communautés mortes détectées. (Voir figure 26 et 27)

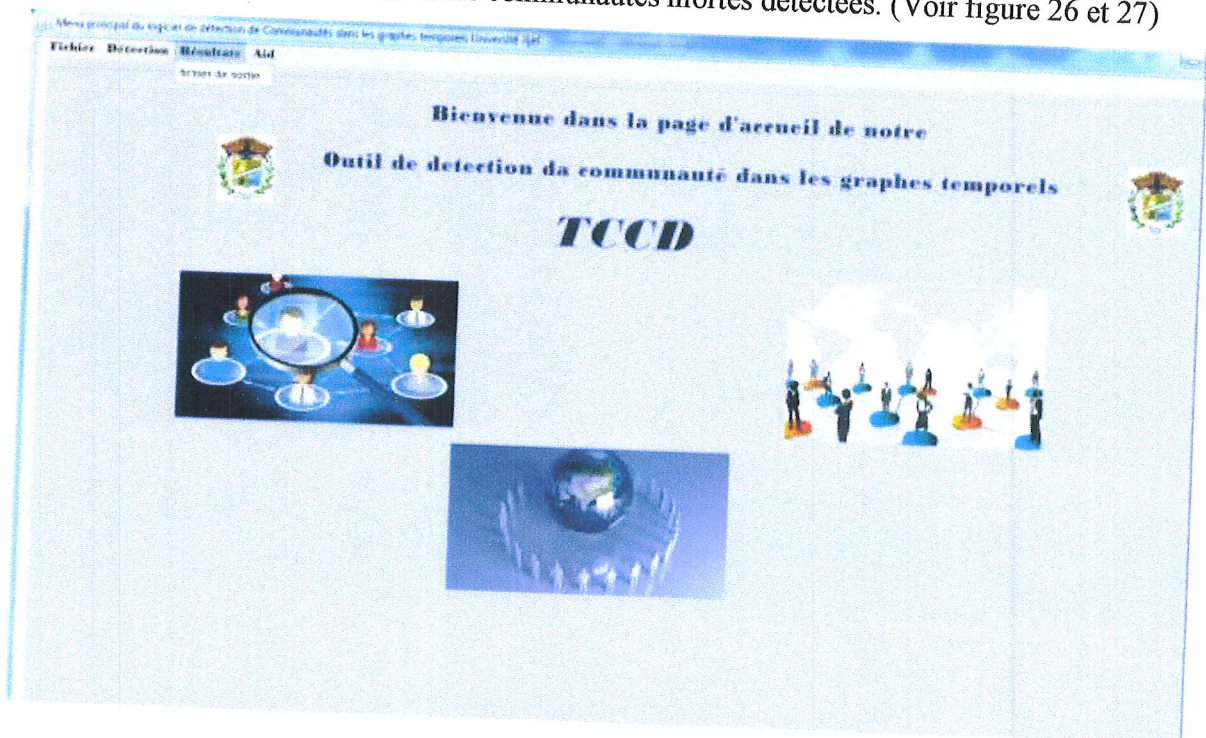


Figure 26: La visualisation du fichier résultat de la détection

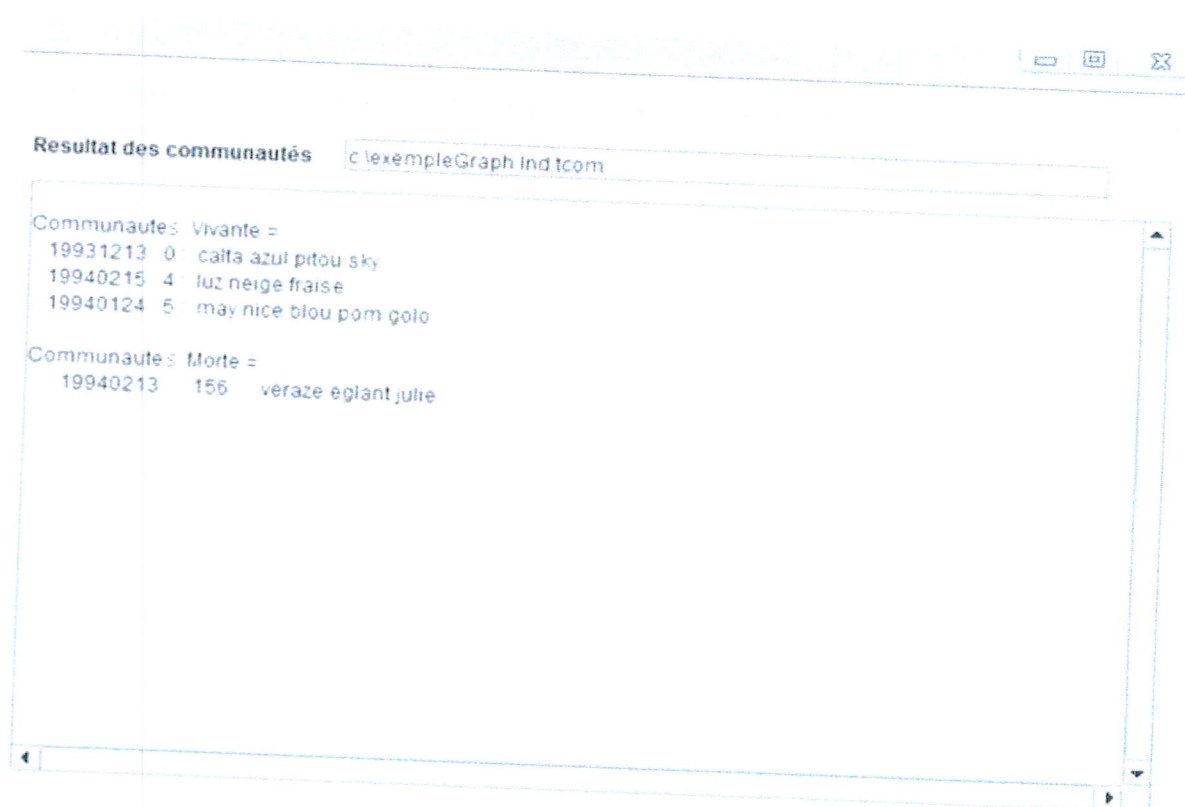


Figure 27 : Le fichier résultat des communautés

Chapitre 4 : Implémentation et validation

1.3.2. Exemples explicatifs

Nous illustrons dans cette section le fonctionnement de notre outil TCCD à travers des exemples de petits graphes temporels. Nous considérons les cas suivants :

a) Ajout des liens :

Exemple1 : Ajout d'un lien externe

Soit le fichier en entrée suivant. (Voir figure 28)

19930326	+	1	2
19931210	+	3	4
19931213	+	4	5
19940111	+	3	5
19940111	+	6	7
19940115	+	7	9
19940115	+	9	8
19940115	+	8	6
19940124	+	8	7

Figure 28: Fichier d'entrée d'exemple 1

Après l'exécution du fichier d'entrée on aura le résultat suivant. (Voir figure 29)

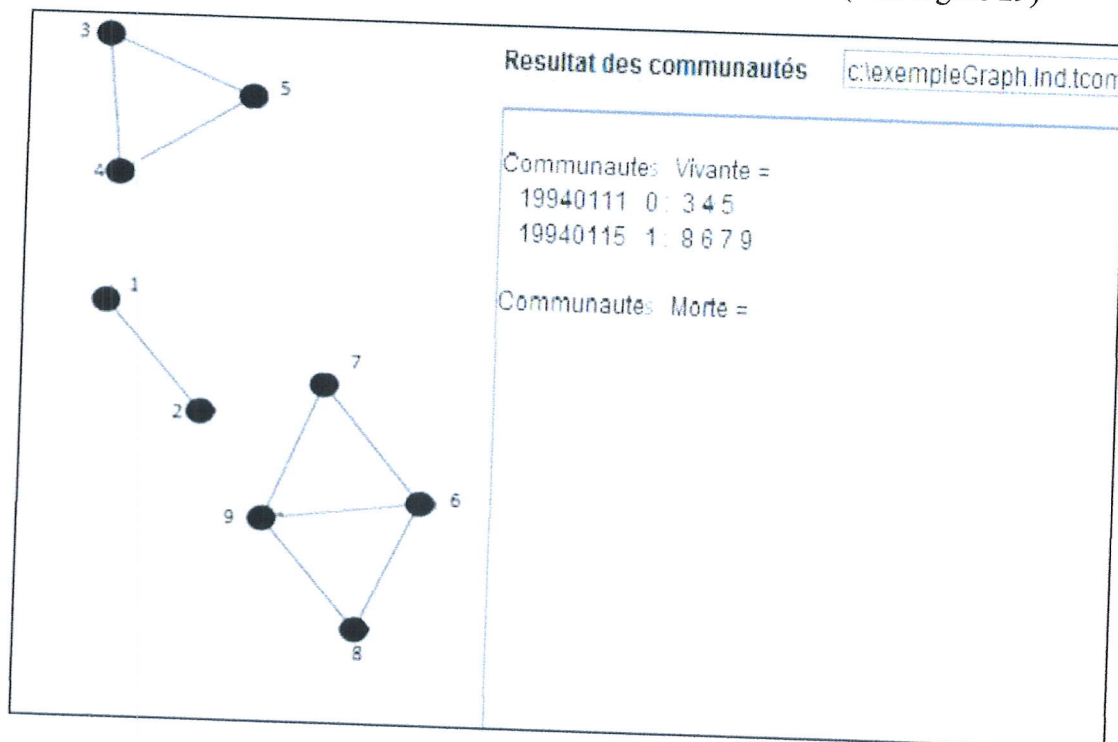


Figure 29 : Résultat exemple 1

Chapitre 4 : Implémentation et validation

Exemple2 : Ajout des liens intra-communautés

Soit le fichier en entrée suivant. (Voir figure 30)

19930326	+	1	2
19931210	+	2	3
19931213	+	1	3
19940111	+	4	2
19940111	+	4	3
19940115	+	5	6
19940115	+	6	7
19940115	+	5	7
19940116	+	5	8
19940124	+	8	7
19940213	+	8	4

Figure 30 : Fichier d'entrée pour exemple 2

Après l'exécution du fichier d'entrée on aura le résultat suivant. (Voir figure 31)

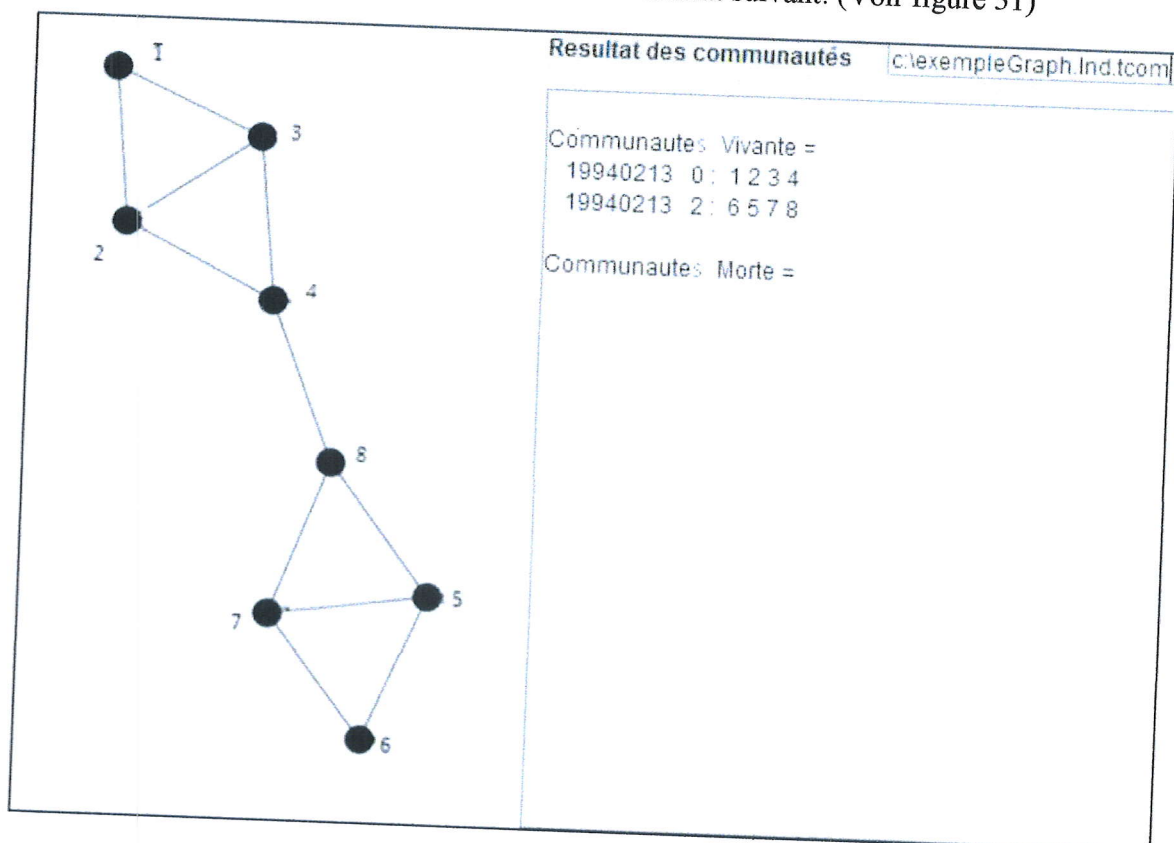


Figure 31: Résultat de l'exemple 2

Chapitre 4 : Implémentation et validation

Exemple3 : Ajout des liens internes

Soit le fichier en entrée suivant. (Voir figure 32)

19930326	+	1	2
19931210	+	2	3
19931213	+	1	3
19940111	+	4	5
19940111	+	3	5
19940115	+	2	5
19940115	+	4	6
19940115	+	5	6
19940124	+	5	7
19940124	+	6	7
19940124	+	7	8
19940125	+	6	8

Figure 32: Fichier d'entrée de l'exemple 3

Après l'exécution du fichier d'entrée on aura le résultat suivant illustré dans la figure 33.

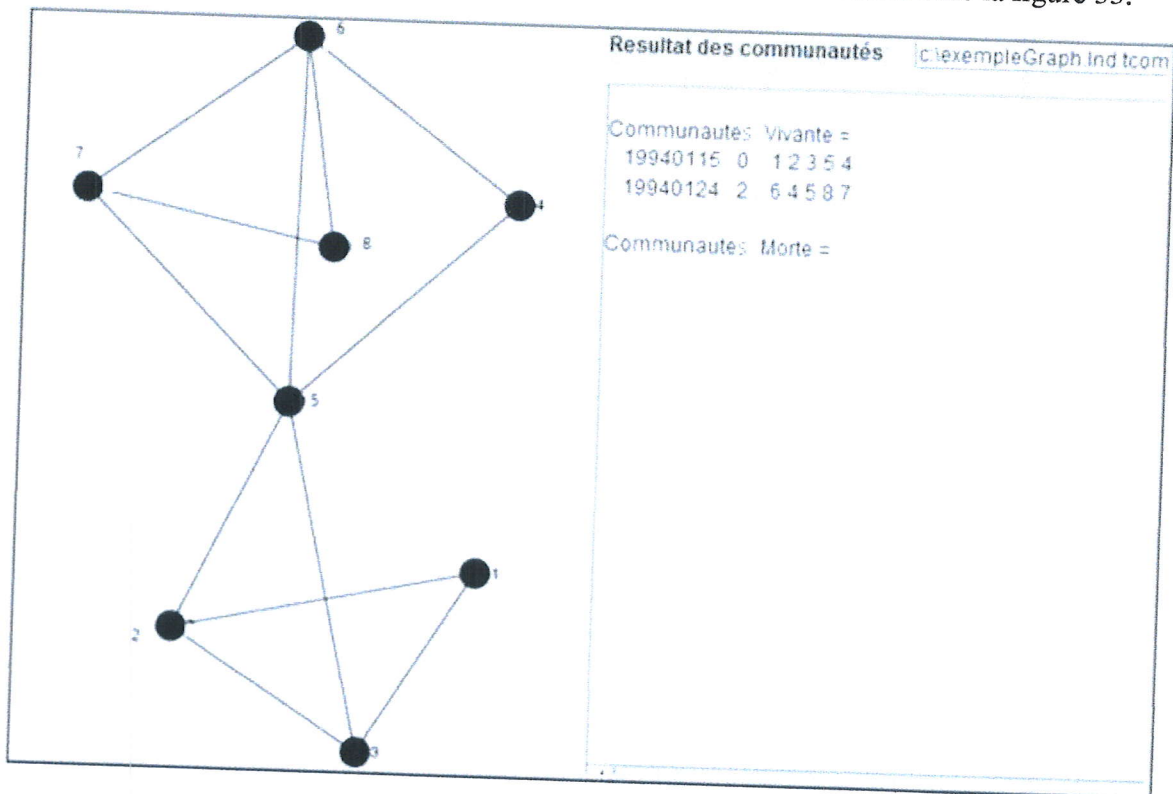


Figure 33: Après l'ajout d'un lien interne

Chapitre 4 : Implémentation et validation

b) suppression de lien :

Exemple 4 : suppression lien externe

Soit le fichier en entrée suivant. (Voir figure 34)

19930326	+	1	2
19931210	+	2	3
19931213	+	1	3
19940115	+	3	4
19940115	+	4	5
19940116	+	5	6
19940117	+	6	4
19940117	+	7	8
19940117	+	8	9
19940120	+	9	7
19940120	-	1	2
19940125	+	11	10
19940126	-	11	10

Figure 34: Fichier d'entrée pour la suppression de lien externe

On aura le résultat suivant. (Voir figure 35 et figure 36).

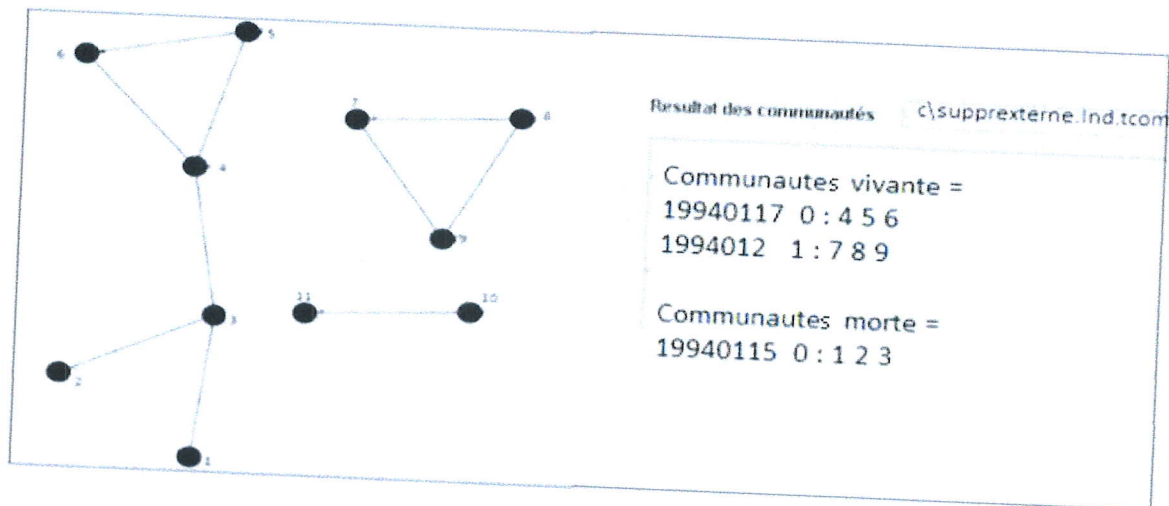


Figure 35 : Avant la suppression du lien(11,10)

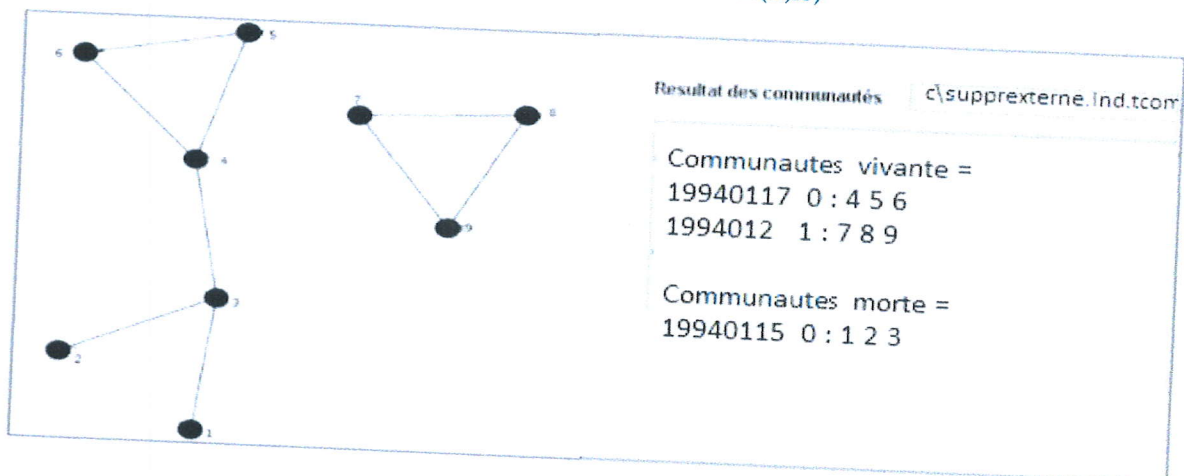


Figure 36 : Après la suppression du lien(11,10)

Chapitre 4 : Implémentation et validation

Exemple 6 : suppression des liens internes

Soit le fichier en entrée suivant. (Voir figure 40)

19930326	+	1	2
19931210	+	2	3
19931213	+	3	1
19940111	+	2	4
19940111	+	2	6
19940115	+	6	8
19940115	+	8	4
19940116	+	8	7
19940116	+	7	6
19940116	+	7	5
19940117	+	5	6
19940118	-	2	6
19940120	+	9	10
19940121	+	10	11
19940122	+	11	9
19940123	-	9	10

Figure 40 : le fichier d'entrée pour exemple 6

On aura le résultat suivant. (Voir figure 41 et figure 42)

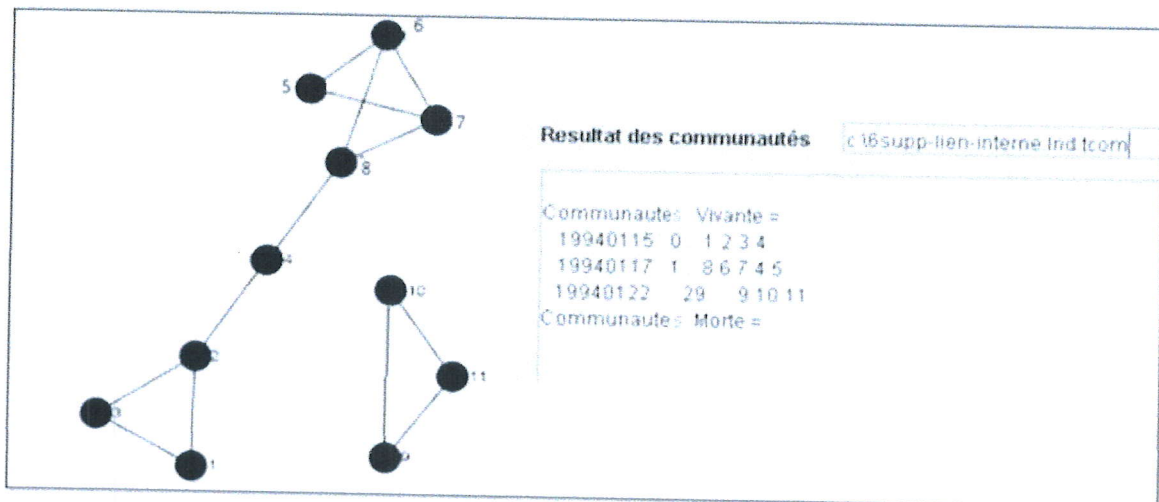


Figure 41 : Avant de supprimer le lien (9,10)

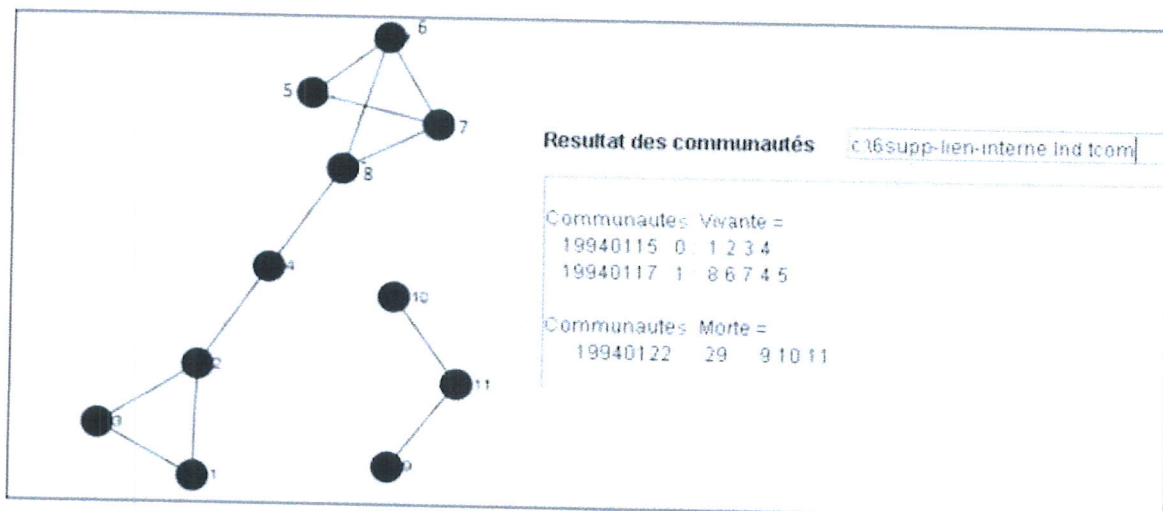


Figure 42 : Après la suppression du lien (9,10)

2. Teste et validation

2.1. Les Réseaux de teste

Notre outil TCCD est capable de détecter les communautés à la fois dans des réseaux dynamiques et dans les réseaux statiques en mettant la même date pour tous les événements. Pour cela nous allons effectuer des expérimentations sur des réseaux réels statiques et dynamiques. Notre choix est porté sur deux réseaux réels pour le cas statique qui sont le réseau de club de Zachary et celui des dauphins de Lusseau. Pour les réseaux dynamiques on remarque qu'il n'y a pas beaucoup de réseaux temporels. A cet effet, nous avons repris le même réseau de test utilisé par l'algorithme ILCD [Caz13].

2.1.1. Club de karaté du Zachary [Zach77]

Le réseau de Zachary est un réseau social statique des membres d'un club de karaté de l'université San Francisco aux Etats Unis. Le club de karaté compte 78 membres. Zachary a fait une étude sur les membres du club qui ont des relations d'amitié en dehors du club. Parmi les 78 membres du club, seuls 34 ont des relations d'amitié. Le réseau d'amitié issu de ce club est représenté dans la figure 43. Ce réseau donne comme résultat deux communautés. Nous avons représenté le graphe selon le format d'entrée de notre algorithme (voir la figure 43). On mettant une date similaire pour tous les événements du réseau. Le résultat est illustre dans la (figure 44).

19940111	+	1	2
19940111	+	2	3
19940111	+	3	1
19940111	+	4	1
19940111	+	7	5
19940111	+	5	11
19940111	+	11	7
19940111	+	8	3
19940111	+	8	4
19940111	+	9	1
19940111	+	7	1
19940111	+	9	3
19940111	+	3	10
19940111	+	11	6
19940111	+	12	1
19940111	+	13	1

Figure 43 : partie du fichier d'entrée de TCCD pour le réseau du club de Zachary

Chapitre 4 : Implémentation et validation

La figure 44 illustre le résultat d'exécution de TCCD sur le réseau statique de Zachary donnée comme résultat trois communautés.

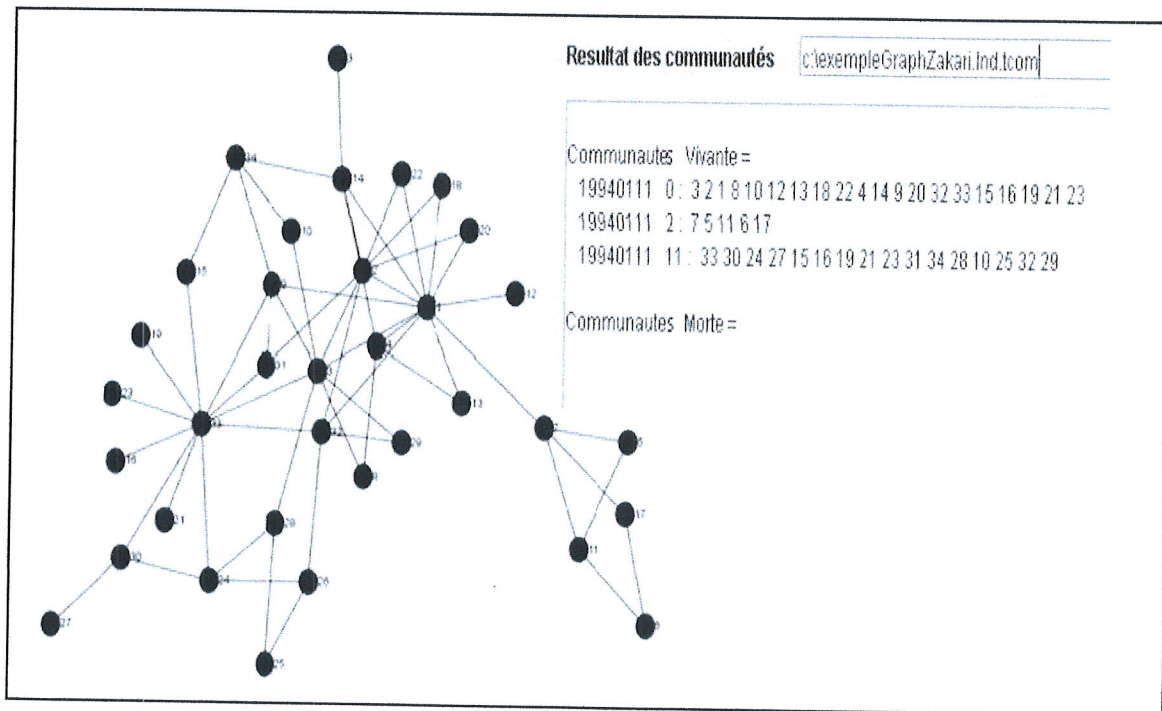


Figure 44: Résultat d'exécution du réseau de Zachary par TCCD

2.1.2. Réseau de dauphins de Lusseau [Lus 03]

Il est un réseau statique créé en 2003 par Lusseau qui avait pour but d'étudier les fréquentes associations entre les dauphins. Ce réseau contient 62 nœuds et 159 liens. Ce réseau donne comme résultat deux communautés. Nous avons représenté le graphe selon le format d'entrée de notre algorithme (voir la figure 45). On met une date similaire pour tous les événements du réseau. Le résultat est illustré dans la (figure 46).

19940115	+	8	3
19940115	+	9	5
19940115	+	9	6
19940115	+	10	0
19940115	+	10	2
19940115	+	13	5
19940115	+	13	9
19940115	+	14	0
19940115	+	14	3
19940115	+	15	0
19940115	+	16	14
19940115	+	17	1
19940115	+	17	13
19940115	+	18	15
19940115	+	19	1
19940115	+	19	7

Figure 45: Partie du fichier d'entrée de TCCD pour le réseau des dauphins de Lusseau

On a le résultat retourné par TCCD qui est illustré dans la figure 46.

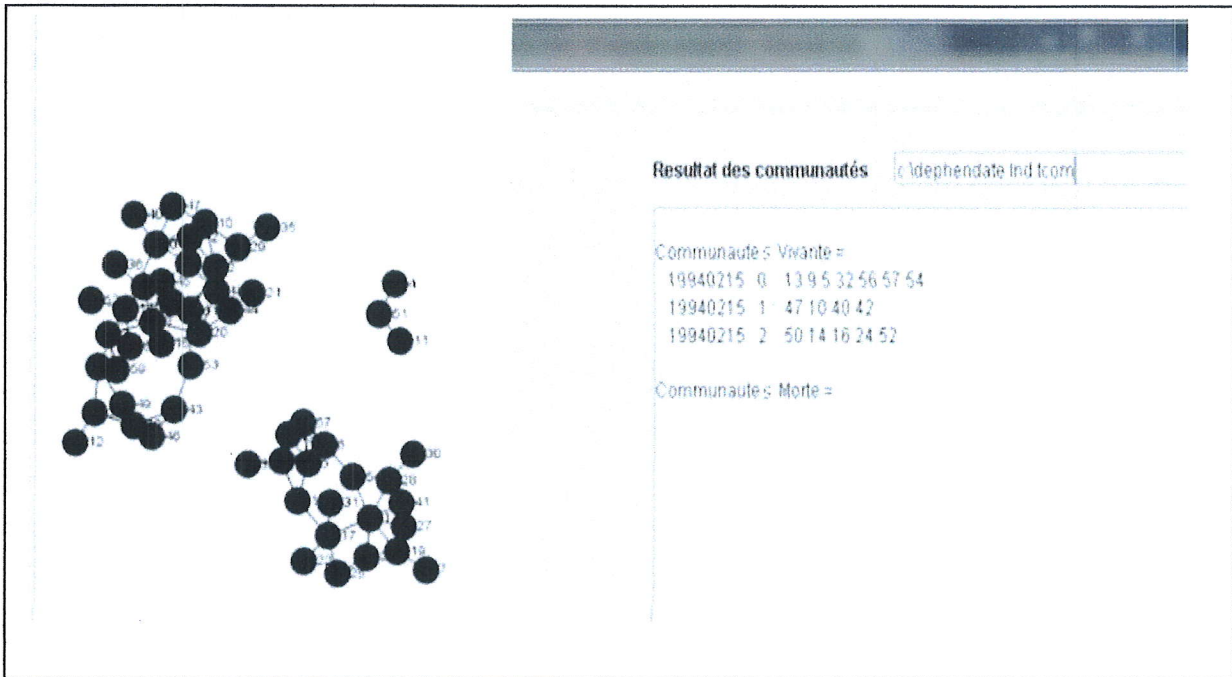


Figure 46: Résultat de TCCD pour le réseau des dauphins de Lusseau

2.1.3. Le réseau temporel d'isards

C'est un réseau temporel d'animaux qui ressemblent à des chamois, dont les femelles en particulier ont un comportement social. La population étudiée est une population vivant en liberté dans le parc national des Pyrénées. Il est constitué de 2500 relations entre eux. Ce réseau est disponible sous la forme TNF (Community Transport Neutral encapsulation Format File) c-à-dire un fichier que chaque ligne commence par le symbole '#' suivi d'une date et dans la quel il y'a une succession d'événement. On va voir un fragment de la structure de ce fichier TNF dans la figure 46.

```
#19930309
+   mamie   cohe
#19930314
+   narcis  camou
-   mamie   cohe
#19930326
+   pitou   azul
-   narcis  camou
#19931202
+   calta   azul
#19931209
+   neige   fraise
#19931210
-
```

Figure 47: Réseau d'isard sous la forme TNF

Chapitre 4 : Implémentation et validation

Nous avons convertie le fichier TNF en format LND qui conforme au format d'entrée de l'outil TCCD. (Voir figure 47)

```
19930309 + mamie cohe
19930314 + narcis camou
19930314 - mamie cohe
19930326 + pitou azul
19930326 - narcis camou
19931202 + calta azul
19931209 + neige fraise
19931210 + pitou calta
19931210 + sky pitou
19931210 + sky calta
19931213 + sky azul
19940111 + neige luz
19940111 + luz fraise
19940115 + veraze eglant
19940115 + veraze julie
19940115 + julie eglant
19940124 + pom golo
19940213 + pom may
19940215 + may fraise
19940218 + nice blou
```

Figure 48: Fichier d'entrée pour le réseau isards

On a le résultat retourné par TCCD qui est illustré dans la figure 49



Figure 49 : Résultat de TCCD pour le réseau d'izard.

3. Choix des algorithmes à comparer

Pour tester les performances de notre algorithme nous allons comparer nos résultats obtenus avec ceux des algorithmes de la littérature. Notre choix est porté sur l'algorithme ILCD travaillant sur des réseaux temporels et en dernier sur l'illustre algorithme de Girvan et Newman.

3.1. Test sur des réseaux statiques

Algorithme \ Réseaux	Nombre de communautés détectées	
	Girvan Et Newman	TCCD
Zachary karaté club	4	3
Dauphins de Lusseau	5	3

Tableau 2 : Résultats d'exécutions pour les deux réseaux statiques

Le réseau du club de Zachary donne en réalité deux communautés en résultat, et notre outil TCCD a donné en résultat trois communautés et par rapport à l'algorithme de Girvan et Newman il donne quatre communautés en résultats. On constate que les résultats donnés par TCCD sont très intéressants et très proche des résultats réel et mieux de Girvan et Newman. Alors que pour Le réseau de Dauphin qui donne en réalité deux communautés en résultat, et notre outil TCCD a donné trois communautés et par rapport à l'algorithme de Girvan et Newman il donne cinq communautés en résultats. On constate que les résultats donnés par TCCD sont très intéressants et très proche des résultats réel et mieux de Girvan et Newman.

3.2. Teste sur le réseau dynamique d'Izard

Algorithme \ Critères	Cazabet	TCCD
Nombre de communautés vivantes	5	8
Nombre de communautés mortes.	21	10

Tableau 3 : Résultats d'exécutions pour le réseau dynamique d'Izard.

Chapitre 4 : Implémentation et validation

Le réseau d'Izard donne en réalité cinq communautés vivantes et vingt et une communauté morte, et notre outil TCCD a donné en résultat huit communautés vivantes et dix mortes. On constate que les résultats donnés par TCCD sont satisfaisants mais non efficace car le temps de calcul est très long.

4. Bilan :

Les résultats obtenus par TCCD pour les deux réseaux statiques sont intéressants et très proches à la réalité que ceux de Girvan et Newman. TCCD donne des résultats satisfaisants mais non efficaces car le temps de calcul est très long.

5. Conclusion :

Pour valider notre outil nous avons effectué des expérimentations sur des réseaux de tests statiques et dynamiques. L'outil donne des résultats efficaces dans le cas des réseaux statique. Cependant, les résultats de tests sur les réseaux dynamiques sont moins efficaces et le calcul prend beaucoup de temps.

Conclusion et perspectives

Le travail réalisé dans le cadre de ce mémoire se situe dans le domaine de la détection de communautés dans les réseaux complexes. Nous nous sommes particulièrement intéressés à la détection de communautés dynamiques dans les graphes temporels. Celle-ci constitue un outil puissant pour comprendre la structure et le fonctionnement du réseau, et aussi ses mécanismes dynamiques.

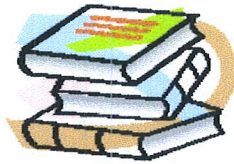
La détection de communautés dynamiques est un domaine qui est encore dans une phase d'exploitation. Cette relative jeunesse du domaine a, d'une part représenté un challenge et d'autre part, été un facteur de motivation très important.

Malgré la présence de plusieurs algorithmes conçus pour trouver des communautés dynamiques, ils existent peu d'algorithmes travaillant directement sur des réseaux temporels, qui sont pourtant les plus adaptés pour représenter l'évolution d'un réseau.

Notre proposition était de réaliser un outil de détection de communautés dynamiques. Pour cela nous avons utilisé les graphes temporels pour pouvoir suivre l'évolution du réseau dans le temps. On voulait aussi traiter l'aspect recouvrement pour cela nous avons utilisé la notion de clique.

Afin de valider notre outil TCCD de détection de communautés dynamiques nous l'avons testé sur deux réseaux réels statiques qui sont celui du club de karaté de Zachary et celui des dauphins de Lusseau et sur un réseau dynamique qui est le réseau d'isards. D'après les résultats des expérimentations faites, nous avons constaté que TCCD nous a fournit des bons résultats. Cependant, pour les grands graphes les résultats sont moins efficaces car le temps d'exécution est très long.

Dans le but d'améliorer ce travail, trois suggestions peuvent être émises. La première consiste à rendre notre taille initiale de clique standard pour envisager la détection de clique de façon directe. La deuxième, consiste à diminuer le temps d'exécution qui est long. La troisième, consiste à traiter l'opération de fusion qui peut intervenir lors de l'ajout des liens intra-communautés.



Bibliographie

- [AO12] A.Oussama, «Utilisation des technologies du web sémantique dans les réseaux sociaux», mémoire de master, Université de OUARGLA, 2012.
- [Caz13] Rémy Cazabet, « Détection de communautés dynamiques dans des réseaux temporels», thèse de doctorat, Université Paul Sabatier - Toulouse III, 2013.
- [Chen10] Z. Chen, K. a. Wilson, Y. Jin, W. Hendrix, and N. F. Samatova, «Detecting and Tracking Community Dynamics in Evolutionary Networks», In proceedings of the 8th IEEE international conference on data mining ,Minnesota au États-Unis ,2010.
- [CNM04] A. Clauset, M.E.J. Newman, and C. Moore, «Finding community structure in very large networks», Physical review E, 70 p066111, 2004.
- [DDA06] L. Danon, A. Diaz-Guilera, and A. Arenas, «The effect of size heterogeneity on community identification in complex networks», Journal of Statistical Mechanics :Theory and Experiment , 11 p11010, 2006.
- [FBS08] T. Falkowski, A. Barth, and M. Spiliopoulou, « Studying community dynamics with an incremental graph mining algorithm »,In proceedings of the 14th Americas conference on information systems AMCIS, Toronto, 2008.
- [GE 09] G.Erétéo , « Analyse des Réseaux Sociaux et Web Sémantique », ISICIL-ANR-08-Cord-011-05, 2009.
- [GN02] M. Girvan et MEJ Newman, «community structure in social and biological network» , In proceedings of national academy of sciences, United States, 2002.

- [JHB12] J. Li, L. Huang, T. Bai, Z. Wang, and H. Chen, «A dynamic community detection method based on incremental analysis », In proceeding of Cdbia, 2012.
- [Lin09] Y.R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B.L. Tseng, «Analyzing communities and their evolutions in dynamic social networks», ACM, In proceedings of 17th international conference on journal of the American statistical association, 104 p 704-717 , 2009.
- [Lus 03] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson, «The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations», In proceedings of Behavioral Ecology and Sociobiology, 54 p 396–405, 2003.
- [New 04] M. E. J. Newman, « Fast algorithm for detecting community in networks», Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, 2004.
- [PBV07] G. Palla, A. Barabasi et T. Vicsek, « Quantifier l'évolution du groupe social», Nature 446 p 664-667, 2007.
- [Pon07] P. Pons, «détection de communautés dans les grands graphes de terrain »,thèse de doctorat, Université de paris, 2007.
- [SLX12] J. Shang, L. Liu, F. Xie, Z. Chen, J. Miao, X. Fang, and C. Wu, « A real time detecting algorithm for tracking community structure of dynamic networks», SNAKDD Workshop, Hong Kong ,2012 .
- [SCC09] H. Shen, X. Cheng, K. Cai, and M.B, «effect overlapping and hierarchical community structure in networks», Physical A : Statistical Mechanics and its Applications, 388 p1706-1712, 2009.
- [TM13] T. Mouhamed,«une nouvelle approche de détection de communautés dans les réseaux sociaux», mémoire de master, Université du Québec en Outaouais 2013.

- [UKB11] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow, «The anatomy of the facebook social graph »,arXiv Preprint arXiv, p1111.-4503,2011.
- [VL08] V. Levorato, « Contributions à la Modélisation des Réseaux Complexe Prétopologie et Applications Modélisation et simulation», thèse de doctorat, Université de Paris VIII Vincennes-Saint Denis, 2008.
- [Wan12] Q. Wang,« Détection de communautés recouvrantes dans des réseaux de terrain dynamique », thèse de doctorat, Université de Lyon 2012.
- [WF10] Q. Wang and E. Fleury, «Mining time-dependent communities», In LAWDN-Latin American Workshop on Dynamic Network, 2010.
- [WT07] K. Wakita, T. Tsurumi, « Finding community structure in mega-scale social NetworksIn », Proceedings of the 16th international conference on World Wide Web ACM, p1275-1276, New York ,2007.
- [XSK11] Xie, J .Symanski, B. K and Liu .X, « MPA: Uncovering overlapping communities In social networks via a speaker-listener interaction dynamic process», in Proceedings of international journal of emerging technologies in applied sciences, p 299-303, india, 2011.
- [Zach 77] Zachary, «An information flow model for conflict and fission in small groups», In proceedings of small Journal of anthropological research of 33,452-473, 1977.