

ED 224 - Cognition, Langage, Interaction

Thèse

présentée par **Vincent LEVORATO**

pour l'obtention du grade de

Docteur

Spécialité : Informatique

Contributions à la Modélisation des Réseaux Complexes : Prétopologie et Applications

soutenue publiquement le 5 Décembre 2008 devant le jury composé de :

Président	Pr Michel LAMURE	Professeur à l'Université Lyon I
Rapporteurs	Pr Jean-Paul AURAY	Directeur de Recherche au CNRS
	Pr Van-Dat CUNG	Professeur à l'INP Grenoble
Examinateurs	Pr Ivan LAVALLÉE	Professeur à l'Université Paris 8
	Pr Sébastien LIMET	Professeur à l'Université d'Orléans
Directeur	Pr Marc BUI	Professeur et Directeur d'Etudes Cumulant à L'EPHE

Laboratoire d'Informatique et des Systèmes Complexes (LaISC)

"Un bon maître a ce souci constant : enseigner à se passer de lui."
André Gide, extrait de *Journal*.

Remerciements

Je voudrais en premier lieu remercier tous les membres du jury pour m'avoir fait l'honneur d'accepter de juger ce travail de thèse :

- Michel Lamure, professeur à l'Université Lyon I, président du jury.
- Jean-Paul Auray directeur de recherche au CNRS, et Van-Dat Cung, professeur à l'Institut polytechnique de Grenoble, rapporteurs de thèse.
- Ivan Lavallée, professeur à l'Université Paris 8, et Sébastien Limet, professeur à l'Université d'Orléans, examinateurs.
- Marc Bui, Professeur et Directeur d'Etudes Cumulant à L'EPHE, directeur de thèse et examinateur.

Cette thèse n'aurait vu le jour sans la confiance, la patience et la générosité de mon directeur de recherche, Professeur Marc Bui, que je souhaite vivement remercier. Je voudrais également le remercier pour le temps qu'il m'a accordé et la patience dont il a fait preuve tout au long de ces années, d'avoir cru en mes capacités et de m'avoir fourni d'excellentes conditions logistiques et financières. De plus, les conseils qu'il m'a prodigués tout au long de mes recherches ont toujours été d'une grande aide, me facilitant grandement la tâche et me permettant d'aboutir à la production de ce mémoire. Qu'il sache que je le considère comme un de mes mentors et que je lui voue une vive estime.

Je ne saurai trop remercier mon autre mentor, le professeur Ivan Lavallée pour son soutien, sa sympathie et l'intérêt qu'il a manifesté à mon travail. Qu'il trouve ici ma profonde gratitude et mes remerciements les plus sincères.

Un merci particulier aux professeurs Michel Lamure et Gérard Duru, pour la qualité de leur collaboration, leurs nombreux conseils, et pour la façon amicale avec laquelle ils ont suivi ce travail, ainsi qu'au professeur Stéphane Bonnevay pour ses remarques pertinentes qui ont permis d'approfondir et d'enrichir les problématiques abordées au cours de ce travail.

Je voudrais remercier le professeur Alain Bui qui a su me montrer la voie à suivre, et sans qui je n'aurais sûrement jamais entrepris mon travail de thèse. Je remercie également les professeurs Michaël Krajecki et Olivier Flauzac qui m'ont initié aux travaux de recherche dès le commencement de mon cursus universitaire.

Mes remerciements vont bien sûr à tous ceux qui y ont participé de près ou de loin à la réussite de ce travail, et en particulier les membres du LRIA, MA2D, LaISC, CSMC, CHART, LICA pour leur présence et leur collaboration lors des moments de travail ainsi que lors des moments de rire et de détente.

Je tiens également à remercier mes collègues et amis Sofiane Benamor, Murat Ahat, Omar Bennani, pour leur aide, leur bonne humeur, leurs encouragements et leur confiance en moi. Ils ont rendu à mon sens plus facile la réalisation de ce travail.

Mes remerciements les plus profonds vont aussi à mes parents et à ma soeur, qui m'ont soutenu durant mes nombreuses années d'études, et amoureusement à ma compagne qui m'a motivé au quotidien durant toutes ces années.

Et à tous ceux qui étaient là, avec qui j'ai tant partagé. Merci.

Contributions à la Modélisation des Réseaux Complexes : Prétopologie et Applications

Résumé :

Un réseau complexe est un réseau d'interactions entre entités dont le comportement global n'est pas déductible des comportements individuels desdites entités, d'où l'émergence de nouvelles propriétés. Notre problème est l'analyse et la modélisation de ces réseaux. L' analyse nécessite un formalisme englobant à la fois structure (approche statique) et fonction (approche dynamique), afin d'avoir une meilleure compréhension des caractéristiques de ces réseaux. En premier lieu, nous présentons dans cette thèse les modélisations utilisées jusqu'à présent et basées sur la théorie des graphes, sensées simuler le comportement des réseaux complexes. En analysant les faiblesses de ces modèles quant à une représentation convaincante des réseaux du monde réel (réseaux sociaux, informatiques, biologiques, ...), nous apportons une définition formelle générale d'un réseau par le biais de la théorie de la prétopologie, laquelle permet d'exprimer au mieux la dynamique de ces systèmes. Associée à cette définition, nous proposons une série de structures de données permettant de développer toute une algorithmique autour du modèle. En second lieu, nous proposons de nouveaux algorithmes d'analyse basés sur la classification d'éléments et la recherche d'éléments centraux, afin de fournir des outils d'aide à la décision puissants. Enfin nous présentons une librairie logicielle permettant la mise en oeuvre de simulations efficaces de tout modèle basé sur la théorie de la prétopologie.

Mots-clés : Prétopologie, Algorithmique, Réseaux Complexes, Classification prétopologique, Méthodes des centres, PretopoLib

Complex Network Modeling Contributions : Pretopology and Applications

Abstract :

A complex network is a interaction network of entities where global behavior is not deductible from the individual behaviors of each entities, leading to new properties emergence. Our problem is the network analysis ad modeling. Network analysis needs a formalism to assemble together the structure (static approach) and the function (dynamic approach), and to have a better understanding of the networks caracteristics. First, in this thesis, we introduce common used network modeling based on graph theory, having the role to simulate complex networks. By analyzing weakness of this models about a convincing representation of real networks (social sciences, computers, biology), we bring a formal general definition of a network using pretopology theory, allowing us to have a better reproduction of system dynamics. With that definition comes series of data structures allowing us to develop a whole algorithmic surrounding the model. Secondary, we propose new analyzing algorithms based on element classification and on centers search, giving power tools for decision aid. To finish, we introduce a software library permitting efficient simulations of every models based on pretopology theory.

Keywords : Pretopology, Algorithmic, Complex Networks, Pretopological Classification, Centers Methods, PretopoLib

Table des matières

Remerciements **v**

Résumé **vii**

Table des Matières **ii**

Liste des Figures **vii**

Introduction **1**

1 Les Réseaux Complexes, du Réel à la Modélisation **5**
1.1 Introduction 5
1.1.1 Définitions 6
1.1.2 Problématique : modélisation d'un réseau complexe 7
1.2 La théorie des graphes 8
1.2.1 Définition et concepts de base 8
1.2.2 Représentations d'un graphe 11
1.2.3 Coloration des sommets d'un graphe 13
1.2.4 Connexité dans les graphes 13
1.2.5 Autres graphes 16
1.3 Modélisation des réseaux complexes par la théorie des graphes 17
1.3.1 Propriétés d'un graphe 17
1.3.2 Structure & dynamique des graphes 20
1.4 Modélisations existantes : graphes aléatoires, invariance d'échelle et petits-mondes 22
1.4.1 L'expérience de Milgram 22
1.4.2 Les graphe aléatoires 23
1.4.3 Modèle d'Albert et Barabási 27
1.4.4 Les petit-mondes (Small-World) de Watts & Strogatz 29
1.4.5 Modèle de petit-monde navigable de Kleinberg 32
1.4.6 Modèle de petit-monde de Lebhar basé sur le modèle de Kleinberg . 34
1.5 Conclusion 37

2 Modélisation d'un réseau par la prétopologie **39**
2.1 Introduction 39
2.2 La théorie de la prétopologie 40

2.2.1 Espace prétopologique 40
2.2.2 Fermés et Ouverts, Fermeture et Ouverture 41
2.2.3 Fermés élémentaires et fermés minimaux 42
2.2.4 Espace prétopologique de type $\mathcal{V}$ 42
2.2.5 Préfiltre et base de voisinage 44
2.3 Modélisation d'un réseau par la prétopologie 45
2.3.1 Définition d'un réseau en prétopologie 46
2.3.2 Structures de données adaptée à la prétopologie 46
2.4 Exemple de modélisation d'une problématique 52
2.4.1 Modélisation de flux de migrations inter-villes 52
2.4.2 Modélisation de réseau d'auteurs 55
2.5 Conclusion 57

3 Classification prétopologique 59
3.1 Introduction 59
3.2 Classification de données par partitionnement 60
3.2.1 Mesure de la distance 60
3.2.2 Méthodes de partitionnement hiérarchiques 61
3.2.3 Méthodes de partitionnement non-hiérarchiques 61
3.3 Méthodes de classification prétopologiques 62
3.3.1 MCP et MCPR 62
3.3.2 Encadrement d'un espace prétopologique 68
3.4 Classification prétopologique & complexité de Kolmogorov 71
3.4.1 Complexité de Kolmogorov 71
3.4.2 Estimation de la complexité d'un objet 71
3.4.3 Utilisation de la distance de compression avec la prétopologie 72
3.5 Conclusion 76

4 Centralité et distances à la frontière d'un espace prétopologique 79
4.1 Introduction 79
4.2 Distance à la frontière d'un ensemble 80
4.2.1 F-distance et pseudo-F-distance 80
4.2.2 Pseudo-distance Δ 81
4.2.3 F-distance prétopologique et B-distance 81
4.3 Méthode des ϵ-centres 83
4.3.1 Méthode des ϵ-centres en topologie 83
4.3.2 Méthode des ϵ-centres en prétopologie 83
4.4 Conclusion 89

5 PretopoLib : la librairie JAVA basée sur la prétopologie 91
5.1 Introduction 91
5.2 Structure de données de *PretopoLib* 92
5.2.1 PretopoNodes 93
5.2.2 PretopoEdges 96
5.2.3 PretopoSpace 98

5.2.4 Coûts théoriques de l'adhérence 99
5.2.5 Coûts empiriques de l'adhérence 100
5.3 Méthodes et Exemples basés sur *PretopoLib* 101
5.3.1 Méthodes 102
5.3.2 Autres méthodes 103
5.3.3 Exemples d'applications 104
5.4 Conclusion 107

Conclusion **108**

Bibliographie **110**

Annexes **110**

Table des figures

1.1 Réseau social d'un site Web en communauté 7
1.2 Ponts de Königsberg 8
1.3 Graphe orienté et non-orienté 10
1.4 Graphes complets avec 3, 4, 5 et 6 sommets 11
1.5 Clique (en rouge) d'un graphe 11
1.6 Matrice d'adjacence d'un graphe 12
1.7 Matrice d'incidence d'un graphe 12
1.8 Liste d'adjacence d'un graphe 14
1.9 Coloration d'un graphe en 3 couleurs 15
1.10 Graphe possédant 2 composantes connexes. 15
1.11 Exemples de coefficients de regroupement 17
1.12 Exemple du degré de corrélation d'un sommet i 18
1.13 Courbes de distribution des degrés d'un graphe 19
1.14 Distribution des degrés de réseaux réels 19
1.15 Exemple de biais du coefficient de regroupement 21
1.16 Les six degrés de séparation de Milgram 23
1.17 Exemple de graphe aléatoire $G_{n,p}$ avec sa composante géante au centre . . 24
1.18 Phase de transition de formation de la composante géante dans un graphe aléatoire $G_{n,p}$ 25
1.19 Comparaison du coefficient de regroupement entre réseaux réels et graphes aléatoires 25
1.20 Graphe aléatoire généralisé de Molloy & Reed suivant une loi de puissance 27
1.21 Graphe construit suivant le modèle d'attachement préférentiel d'Albert et Barabási 28
1.22 Modèle Small-World de Watts et Strogatz 30
1.23 Variante du modèle Small-World de Watts et Strogatz 31
1.24 Coefficient de regroupement C et longueur moyenne noeud à noeud l en fonction de p du modèle Small World Watts et Strogatz 32
1.25 Exemple de réseau Kleinberg $K^2_{1,n}$ 33
1.26 Extension du chemin π (en gras) à la fin d'une phase d'exploration. 35
1.27 Routage de Lebhar par explorations successives 36

2.1 Illustrations d'adhérence & d'intérieur 41
2.2 Illustration d'adhérences successives menant au fermé 41
2.3 Opérations représentant la dynamique d'un réseau (t : temps) 46

2.4 Hachage par adressage ouvert 49
2.5 Hachage par chaînage 49
2.6 Exemple de table de hachage avec $\mathcal{V}_s$ 51
2.7 Schéma des flux de migrations inter-villes 53
2.8 Résultats des adhérences et fermetures sur E 54
2.9 Illustration de la simulation basée sur l'article Largeron-Bonnevay 54
2.10 Réseau d'auteurs sous forme de graphe 56
2.11 Fermés (gauche) et Composantes fortement connexes (droite) 56

3.1 Exemple de dendogramme 61
3.2 Exemple de partitionnement d'un espace avec les k-moyennes 62
3.3 Exemple de découpage des ensembles utilisés pour MCP 68
3.4 Fonctions de surjection de Y dans X et de X dans Y. 73
3.5 Espace prétopologique des articles (G, a_G) 75
3.6 Fonctions de surjection f et g 75
3.7 Classification d'un petit réseau d'auteurs par MCP basé sur la distance normalisée de compression 77

4.1 Pseudo-distance de A à B égale à 3. 81
4.2 Famille des fermés élémentaires d'un espace E. 82
4.3 Bord d'un fermé élémentaire 83
4.4 Exemple d'ϵ-centres dans un espace topologique 84
4.5 ϵ-centres d'un fermé dans un espace prétopologique 86
4.6 ϵ-centres d'un espace prétopologique 87
4.7 Comparaison entre $\mathcal{F}_m$ et ϵ-centres 88
4.8 Cas où la méthode 2 (à droite) est plus fine que la méthode 1 (à gauche) . 90

5.1 Exemple de visualisation avec *PretopoLib* 92
5.2 Structure simplifiée de *PretopoLib* 93
5.3 Performances de l'adhérence en fonction du nombre de relations 101
5.4 Capture d'écran de l'application simulant MCP sur un réseau d'interactions entre protéines dans la levure 104
5.5 Capture d'écran de l'application simulant la méthode 1 des ϵ-centres sur un réseau de livres d'Amazon.com 105
5.6 Capture d'écran de l'application simulant la recherche des composantes fortement connexes sur un réseau de « blogs » traitant des élections présidentielles de 2004 aux Etats-Unis d'Amérique 106
5.7 Capture d'écran de l'application simulant la recherche de classes d'équivalence sur un réseau d'interactions sociales d'un groupe de moines dans un monastère 106

Introduction

Le travail de recherche effectué autour de cette thèse est principalement axé sur l'étude de la modélisation des réseaux complexes ainsi que sur la simulation de ces derniers. L'intérêt premier de ce travail est la formalisation théorique d'un réseau, permettant ainsi une modélisation générale abstraite. Il a également pour objectif de fournir des outils et concepts mathématiques visant à une meilleure compréhension du comportement du modèle.

Cette approche du problème de la modélisation des réseaux nous paraît nécessaire à une compréhension globale d'un système. Nous nous efforcerons de définir mathématiquement la notion de « réseau complexe » en nous appuyant sur des outils théoriques dont nous justifierons le choix. Tout au long de ce document, nous illustrerons la théorie appliquée à des réseaux de différentes catégories.

Le thème de la modélisation des réseaux complexes appartient au thème plus général de la modélisation des systèmes complexes. L'analyse de ces systèmes reste difficile, l'intérêt porté par les chercheurs aux réseaux complexes ayant seulement 50 ans, avec un regain récent ces 15 dernières années grâce à l'essor du Web et des communautés virtuelles. Le problème de la généralisation théorique des réseaux étant connu, aussi, est-il possible de donner une définition mathématique permettant d'unifier le concept de réseau ?

Problématique

Habituellement, pour représenter un réseau, on utilise la *théorie des graphes*. Cependant, les réseaux sont des structures évolutives et des systèmes dynamiques. Ainsi, la théorie des graphes semble ne pas être assez complète pour représenter toutes les propriétés d'un tel système, dit complexe, dans lequel la dynamique joue un rôle prépondérant.
En effet, un bon nombre de modèles de réseaux basés sur la théorie des graphes, existe déjà [32, 10, 66, 84, 6, 40, 69, 54]. Au fil des recherches, de nombreuses caractéristiques et propriétés ont été étudiées, pour au final, but ultime de la modélisation et de la simulation, mettre en évidence les propriétés du monde réel que l'on veut étudier. La question de la correspondance « réseaux du réel - réseaux modélisés » reste cependant sans réponse satisfaisante. La plupart des modèles présentés possèdent, il est vrai, une ou plusieurs caractéristiques appartenant aux réseaux réels. Cependant, même si les derniers modèles en date paraissent pertinents, en particulier pour les réseaux informatiques, il est évident qu'une grande part d'incertitude existe quant à une modélisation efficace et une compréhension globale de ces réseaux. Si les études autour de la structure sont les plus nombreuses, alors peut-être une meilleure compréhension de la fonction permettrait d'éclairer le problème posé, ou tout simplement en allant au-delà d'une modélisation uti-

lisant de simples graphes. Dès que le comportement fonctionnel des réseaux complexes sera compris, la modélisation de ceux-ci n'en deviendra que plus évidente et saisissable.

La théorie de la Prétopologie généralise la notion de proximité introduit par la topologie. Là où les chercheurs étudient les réseaux avec des modélisations centrées sur la structure du réseau et principalement la topologie des graphes les représentant, la prétopologie y ajoute la fonction, permettant ainsi la prise en compte de la dynamique, facteur indispensable dans une telle étude. En proposant une extension à la théorie des graphes, nous pourrons améliorer et adapter certaines mesures existantes sur les réseaux complexes, et même en trouver de nouvelles, les propriétés ensemblistes de la prétopologie nous permettant d'aller plus loin dans la modélisation, ainsi que dans la représentation des phénomènes du réel, tout en se basant sur un cadre théorique solide.

Plan du document et Apports

La présentation des recherches menées dans ce mémoire suit le schéma suivant :
le passage du modèle « réseau complexe » à la simulation. Tout d'abord nous nous sommes intéressé aux recherches actuelles et passées concernant la modélisation des réseaux, afin d'en tirer les enseignements nécessaires à une mise en avant des problèmes rencontrés lors de la modélisation et la simulation de tels systèmes. Puis nous amenons ce type de modélisation sur un « nouveau » terrain, celui de la modélisation des réseaux par la théorie de la prétopologie, permettant une approche de l'étude de la structure et de la dynamique, différente des études usuelles. Les problèmes auxquels se confrontent la plupart des travaux dans ce domaine sont d'une part la correspondance réalité-modèle souvent peu adéquate, et d'autre part l'analyse des caractéristiques de ces modèles qui, se basant sur la théorie des graphes, ne peut faire apparaître certaines propriétés globales inhérentes à la dynamique du système. Ainsi, nous avons essayé d'apporter une réponse unificatrice à ces problèmes de modélisation et de simulation, ainsi que des méthodes générales en permettant l'analyse.

Ce mémoire de thèse est composé de cinq chapitres :

- **Le chapitre 1** est un état de l'art des modélisations des réseaux, lesquelles sont basées sur la théorie des graphes. On énonce des rappels sur la théorie des graphes, théorie utilisée habituellement dans les modèles étudiés dont les fondements sont nécessaires pour comprendre les définitions des dits modèles, puis nous passons en revue les différentes modélisations faites par les scientifiques. Grâce à l'étude de différents travaux portant sur le sujet, il nous a été possible de mettre en évidence les problèmes liés aux modélisations proposées habituellement, et de mettre en évidence les faiblesses, ainsi que de mener une réflexion sur la définition de réseau complexe.
- **Le chapitre 2** permet de faire la transition entre les modèles existants, et la modélisation que nous proposons d'un réseau complexe. Etant basée sur la théorie de la prétopologie, cette dernière fera l'objet d'une description de ses notions mathématiques. On présentera les structures de données choisies et opérations associées visant la simulation de réseaux du réel de manière efficace. La complexité de ces opérations sera également étudiée pour conclure sur une série d'exemples.
- **Le chapitre 3** explicite comment à partir d'un modèle prétopologique il est possible d'analyser un réseau en y appliquant des algorithmes de classification. Nous

partirons de méthodes existantes dont nous étudierons la complexité. Puis nous proposerons un couplage des méthodes de classifications prétopologiques à un indice de similarité basé sur la complexité de Kolmogorov que nous illustrerons par un exemple d'application.

– **Le chapitre 4** a pour objectif de redéfinir la notion de centre dans un espace prétopologique, en prenant en compte la « distance » des éléments à la frontière de cet espace, nous amenant à de nouveaux résultats quant à la centralité des entités composant un réseau complexe et à l'analyse de ces derniers. Nous proposerons de nouvelles méthodes de recherche de centres avec une série d'illustrations.
– **Le chapitre 5** présente l'outil *PretopoLib* afin de permettre la simulation des modèles proposés dans les chapitres précédents. Il y est présentée une description détaillée de la librairie, ainsi qu'une étude de sa complexité quant à son utilisation sur des cas de réseaux du réel. Nous y expliquons comment est structuré le programme, puis nous étudions sur ses performances pour finir par une série d'exemples.
– **La conclusion** permet de faire le point sur le travail présenté, et donne également les nouvelles directions à suivre dans les recherches futures.

Chapitre 1

Les Réseaux Complexes, du Réel à la Modélisation

Dans ce chapitre, nous menons un questionnement sur la définition de réseau complexe : qu'est-ce qu'un *réseau complexe* ? Et de même, qu'est-ce que l'on appelle *réseau* ? Ce chapitre correspond à l'état de l'art de la modélisation mathématique des réseaux. Outre la réflexion sur la notion de réseau complexe, on énoncera des rappels sur la théorie des graphes, théorie utilisée habituellement dans les modèles étudiés dont les fondements sont nécessaires pour comprendre les définitions des dits modèles. Nous passerons en revue les différentes modélisations qui ont été faites jusqu'alors en en faisant une analyse critique quant à la pertinence de l'utilisation de ces modèles pour représenter les réseaux du réel, pour finir sur la nécessité d'une théorie unificatrice plus générale que la théorie des graphes.

1.1 Introduction

La définition de réseau varie selon le domaine ou le contexte dans lequel on se place : pour le chasseur du XIX$^{\text{ème}}$ siècle, un réseau est un filet destiné à capturer certains animaux alors que, dans un domaine opposé comme la physique, un réseau est une pièce d'optique destinée à fournir des spectres et dont le principe repose sur l'interférence d'un grand nombre d'ondes diffractées. Concernant l'informatique, une définition simple d'un réseau est qu'il est un ensemble de moyens matériels et logiciels mis en œuvre pour assurer les communications entre ordinateurs, stations de travail et terminaux informatiques. D'une manière plus générale, dans la langue française, on définit un réseau comme un ensemble de lignes entrecroisées.

Ceci nous amène tout naturellement à l'étymologie du mot « réseau » qui remonte au latin « rete » signifiant filet, donnant l'adjectif réticulé, caractérisant ainsi les objets ayant une structure de filet, notamment les réseaux.
Une des définitions se rapprochant le plus de ce que l'on souhaite expliquer est celle tirée du dictionnaire de la langue française qui, parmi ses définitions pour le mot réseau, nous donne celle-ci : *Répartition des éléments d'un ensemble en différents points ; ces points*

ainsi répartis - Réseau urbain : *ensemble des villes unies par des liens de nature variée (économique, politique, etc.).*
Cette définition se veut la plus générale possible, et on comprend par l'exemple donné, que la nature des interactions envisageables peuvent être totalement différentes.

Nous en venons donc à la définition d'un réseau complexe : pourquoi un réseau est-il complexe ? 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 [3]. Un *réseau complexe* est donc un réseau d'interactions entre entités dont le comportement global n'est pas déductible des comportements individuels desdites entités, d'où l'émergence de nouvelles propriétés [69]. C'est sur cette base de réflexion que nous amenons notre travail.

1.1.1 Définitions

Les réseaux complexes sont présents dans de nombreux domaines aussi divers les uns que les autres : biologie, sociologie, psychologie, informatique, ... Ils recouvrent ainsi des réseaux aussi divers que le réseau Internet, les réseaux d'humains, ou encore les réseaux de protéines. Aussi, nombreuses ont été les études les concernant. Ces réseaux peuvent être regroupés en quatre catégories [69] :

- les *réseaux sociaux*
- les *réseaux d'informations*
- les *réseaux technologiques*
- les *réseaux biologiques*

Un *réseau social* est un ensemble de personnes ou de groupes de personnes possédant des schémas de contacts ou d'intéractions entre eux [25] (Fig. 1.1). C'est à partir de ce type de réseaux que la modélisation du monde réel a été introduite de façon empirique grâce à l'expérience de Milgram [65] qui sera décrite plus tard. La plupart des études concernant ces réseaux souffrent de problèmes d'imprécisions, de subjectivité, et d'échantillons de petite taille [71].

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 noeuds, 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 hyper-liens est également un réseau d'informations (à ne pas confondre avec le 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 d'énergie. Les réseaux électriques, aériens, d'ordinateurs, en font

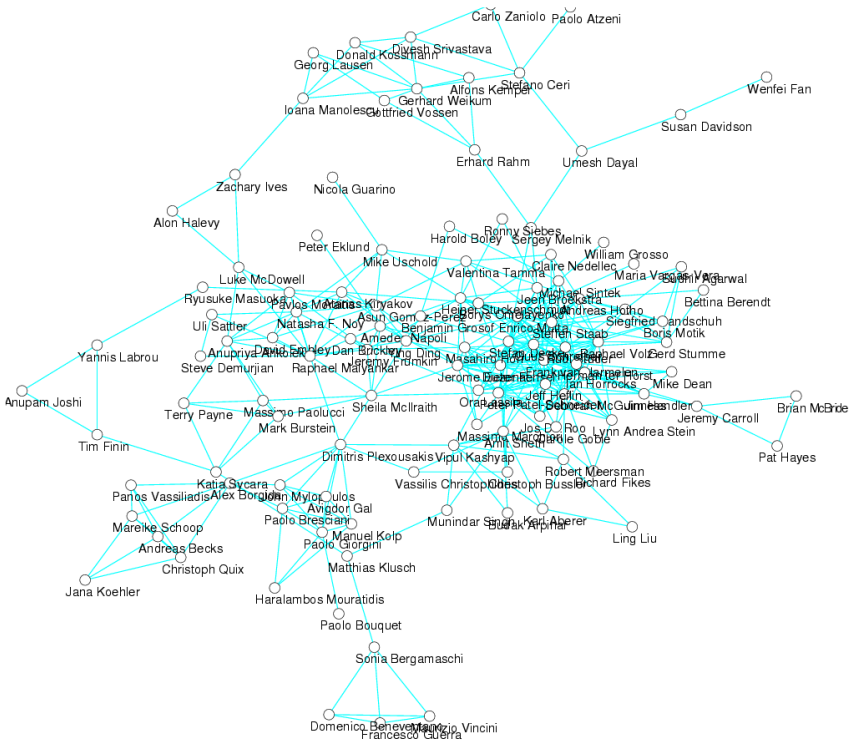


FIG. 1.1 – Réseau social d'un site Web en communauté

partie.

Un *réseau biologique* est un réseau d'éléments touchant au vivant. Un exemple de réseau biologique peut être un réseau d'intéractions entre protéines.

1.1.2 Problématique : modélisation d'un réseau complexe

Dans les réseaux réels, alors que les interactions locales sont généralement bien connues (la communication entre deux routeurs, la réaction entre deux protéines), le résultat global de l'ensemble des interactions est encore mal compris (propriété d'émergence) [45]. La compréhension de ces propriétés globales touche pourtant à des problématiques essentielles : la dynamique des interactions dans un réseau social ou un réseau informatique est par exemple liée à la problématique de la propagation des virus (informatiques ou biologiques), celle d'un réseau de distribution d'électricité, au problème de la robustesse d'un grand réseau. L'augmentation récente des capacités de traitement et de collecte d'un grand nombre de données statistiques sur ces réseaux a permis l'essor des études de ces objets. En particulier, on a observé expérimentalement que ces réseaux, a priori éloignés, partageaient des propriétés macroscopiques communes [79].

Le problème majeur réside dans la modélisation de tels réseaux. Or, pour pouvoir créer des modèles cohérents et reflétant au maximum les propriétés des réseaux réels, il

faut être en mesure de les caractériser. C'est seulement en réussissant à caractériser tel ou tel réseau que l'on pourra reproduire leur comportement lors de simulations. Pour le moment, ces simulations donnent certains résultats mais ceux-ci ne sont pas toujours très pertinents ou ne prennent pas en compte tous les facteurs d'intéractions nécessaires [71]. La caractérisation de l'anatomie d'un réseau fait référence à des définitions incomplètes, d'où le besoin de rechercher une (ou plusieurs) définition(s) générale(s) permettant la reconnaissance et/ou la construction d'un réseau complexe.

1.2 La théorie des graphes

Pour représenter les réseaux, la théorie des graphes paraît l'outil adéquat. C'est principalement cet outil qui a été utilisé dans les différentes études qui ont porté sur la modélisation de réseaux complexes. Dans cette partie, la théorie des graphes sera présentée, puis on étudiera les caractéristiques des propriétés structurelles d'un graphe pour arriver au problème de la dynamique, ainsi que les limites des études auxquelles on se confronte à l'heure actuelle.

1.2.1 Définition et concepts de base

La théorie des graphes est née en 1736 quand Euler démontra qu'il était impossible de traverser chacun des sept ponts de la ville russe de Königsberg (aujourd'hui Kaliningrad) une fois exactement et de revenir au point de départ. Les ponts enjambent les bras de la Pregel qui coulent de part et d'autre de l'île de Kneiphof. Sur la figure 1.2, les noeuds représentent les rives.

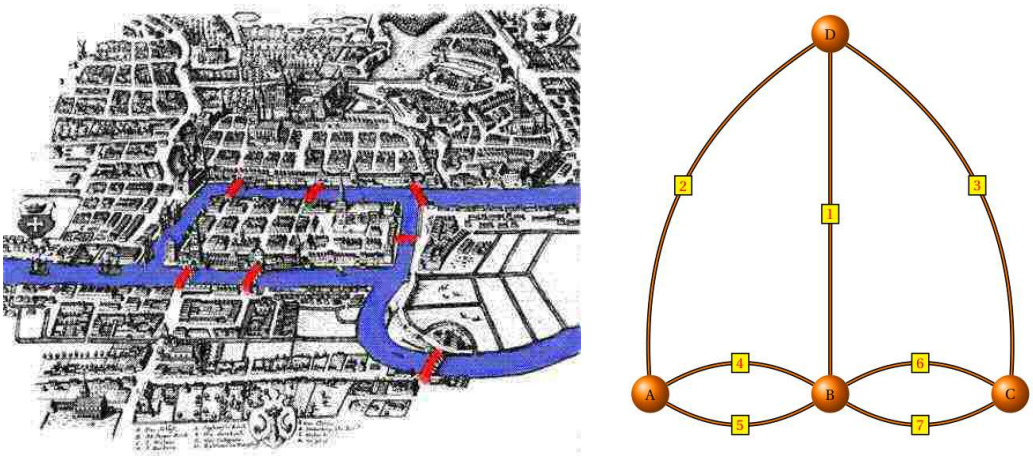


FIG. 1.2 – Ponts de Königsberg

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

des instruments les plus courants et les plus efficaces pour résoudre des problèmes discrets posés en Recherche Opérationnelle (RO). 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 (e.g., réseau de communication, réseaux ferroviaire ou routier, arbre généalogique, diagramme de succession de tâches en gestion de projet, ...). En plus de son existence purement mathématique, le graphe est aussi une structure de données puissante pour l'informatique [61].

Concepts orientés

Dans beaucoup d'applications, les relations entre éléments d'un ensemble sont orientées, i.e., un élément x peut être en relation avec un autre y sans que y soit nécessairement en relation avec x. On parle alors de graphe orienté (en Anglais *directed graph* ou plus simplement *digraph*).

Définition 1.1 *Un graphe $G = (X, U)$ est déterminé par :*
– un ensemble $X = \{x_1, x_2, ..., x_n\}$ dont les éléments sont appelés sommets *ou* noeuds *(ce dernier terme est plutôt utilisé dans le contexte des réseaux).*
– un ensemble $U = \{u_1, u_2, ..., u_m\}$ du produit cartésien $X \times X$ dont les éléments sont appelés arcs.

Pour un arc $u = (x_i, x_j)$, x_i est l'extrémité initiale, x_j l'extrémité finale (ou bien origine et destination). L'arc u part de x_i et arrive à x_j . Un arc (x_i, x_i) est appelé une boucle. Un $p-graphe$ est un graphe dans lequel il n'existe jamais plus de p arcs de la forme (i, j) entre deux sommets quelconques. On appellera communément « graphe »un $1-graphe$. La densité d'un graphe est donnée par le quotient m/n^2, rapport du nombre effectif d'arcs sur le nombre maximal théorique.

Concepts non orientés

Lors de l'étude de certaines propriétés, il arrive que l'orientation des arcs ne joue aucun rôle. On s'intéresse simplement à l'existence d'arc(s) entre deux sommets (sans en préciser l'ordre). Un arc sans orientation est appelé *arête*. U est constitué non pas de couples, mais de paires de sommets non ordonnés. Pour une arête (x_i, x_j), on dit que u est *incidente* aux sommets x_i et x_j .
Remarque : Dans le cas non-orienté, au lieu de noter $G = (X, U)$ et $u = (x_i, x_j)$, on préfère souvent $G = (X, E)$ et $e = [x_i, x_j]$.
Un multigraphe $G = (X, E)$ est un graphe pour lequel il peut exister plusieurs arêtes entre deux sommets.
Un graphe G = (X,E) est simple :

1. s'il n'est pas un multigraphe.
2. s'il n'existe pas de boucles.

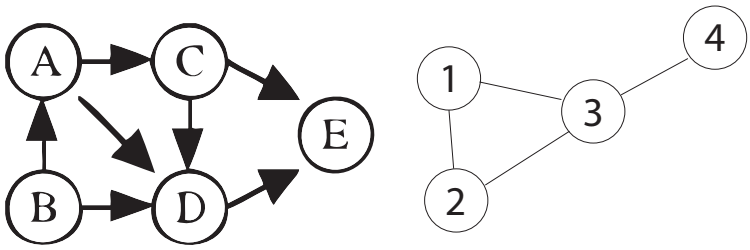


FIG. 1.3 – Graphe orienté et non-orienté

Principales définitions

- *Adjacence*
  - Deux sommets sont adjacents (ou voisins) s'ils sont joints par un arc.
  - Deux arcs sont adjacents s'ils ont au moins une extrémité commune.
- *Degrés*
  - Le demi-degré extérieur de x_i, $d^+(x_i)$, est le nombre d'arcs ayant x_i comme extrémité initiale ; $d^+(x_i) = |\omega^+(x_i)|$.
  - Le demi-degré intérieur de x_i, $d^-(x_i)$, est le nombre d'arcs ayant x_i comme extrémité finale ; $d^-(x_i) = |\omega^-(x_i)|$.
  - Le degré de x_i est $d(x_i) = d^+(x_i) + d^-(x_i)$. Le degré d'un sommet d'un graphe non orienté est le nombre d'arêtes qui lui sont incidentes.
- *Graphe complémentaire*
  $G = (X, U)$ et $G = (X, \bar{U})$. $(x_i, x_j) \in U \Rightarrow (x_i, x_j) \notin \bar{U}$ et $(x_i, x_j) \notin U \Rightarrow (x_i, x_j) \in \bar{U}$. $\bar{G}$ est le graphe complémentaire de G.
- *Graphe partiel*
  $G = (X, U)$ et $U_p \subset U$. $G_p = (X, U_p)$ est un graphe partiel de G. (On peut ainsi obtenir des sommets isolés...).
- *Sous-graphe*
  $G = (X, U)$ et $X_s \subset X$. $G_s = (X_s, V)$ est un sous-graphe de G, où V est la restriction de la fonction caractéristique de U à X_s. $V = \{(x, y)/(x, y) \in U \cap X_s \times X_s\}$.$\forall x_i \in X_s, \Gamma_s(x_i) = \Gamma(x_i) \wedge X_s$.
- *Sous-graphe partiel*
  Combine les deux définitions précédentes.
  Exemple : Réseau routier
  – que les autoroutes : graphe partiel
  – que la région Midi-Pyrénées : sous-graphe
  – que les autoroutes de Midi-Pyrénées : sous-graphe partiel

- Graphe *réflexif* : $\forall x_i \in X, (x_i, x_i) \in U$.
- Graphe *irréflexif* : $\forall x_i \in X, (x_i, x_i) \notin U$.
- Graphe *symétrique* : $\forall x_i, x_j \in X, (x_i, x_j) \in U \Rightarrow (x_j, x_i) \in U$.
- Graphe *asymétrique* : $\forall x_i, x_j \in X, (x_i, x_j) \in U \Rightarrow (x_j, x_i) \notin U$.(si G est asymétrique, G est irréflexif).
- Graphe *antisymétrique* : $\forall x_i, x_j \in X, (x_i, x_j) \in U$ et $(x_j, x_i) \in U \Rightarrow x_i = x_j$ (si G

est asymétrique, G est aussi antisymétrique).
- Graphe *transitif* : $\forall x_i, x_j \in X, (x_i, x_j) \in U, (x_j, x_k) \in U \Rightarrow (x_i, x_k) \in U$.
- Graphe *complet* : $\forall x_i, x_j \in X, (x_i, x_j) \notin U \Rightarrow (x_j, x_i) \in U$.
- *Clique* : ensemble des sommets d'un sous-graphe complet. Soit $C \subset X$ une clique de G non orienté : $\forall (x_i, x_j) \in C, (x_i, x_j) \in U$ (2 sommets distincts de G sont toujours adjacents).
  Notons qu'un graphe complet et antisymétrique s'appelle un « tournoi », car il symbolise le résultat d'un tournoi où chaque joueur est opposé une fois à chacun des autres joueurs.

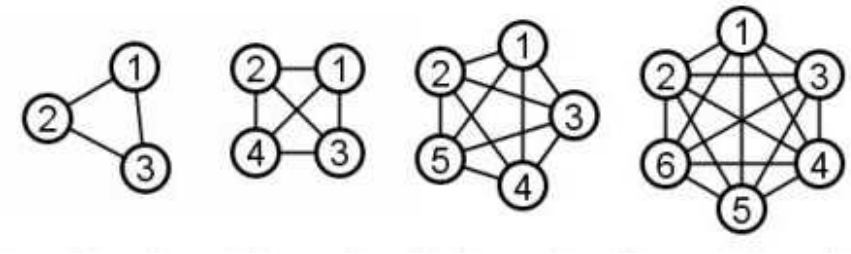


FIG. 1.4 – Graphes complets avec 3, 4, 5 et 6 sommets

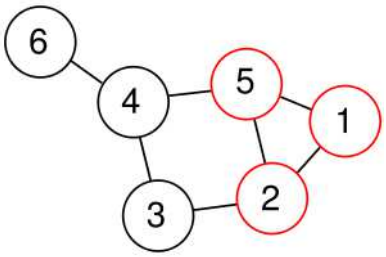


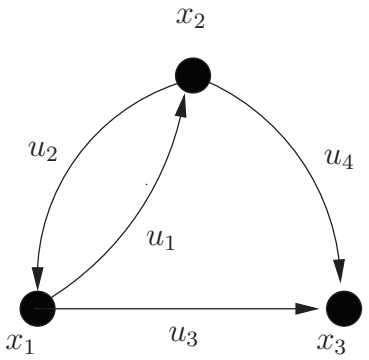
FIG. 1.5 – Clique (en rouge) d'un graphe

1.2.2 Représentations d'un graphe

Un certain nombre de représentations existent pour décrire un graphe. En particulier, elles ne sont pas équivalentes du point de vue de l'efficacité des algorithmes. On distingue principalement la représentation par matrice d'adjacence, par matrice d'incidence sommets-arcs (ou sommets-arêtes dans le cas non orienté) et par listes d'adjacence [13].

Matrice d'adjacence

Considérons un graphe. La matrice d'adjacence (Fig. 1.6) fait correspondre les sommets origine des arcs (placés en ligne dans la matrice) aux sommets destination (placés en colonne). Dans le formalisme matrice booléenne, l'existence d'un arc (x_i, x_j) se traduit par la présence d'un 1 à l'intersection de la ligne x_i et de la colonne x_j ; l'absence d'arc par la présence d'un 0 (dans un formalisme dit matrice aux arcs les éléments représentent le nom de l'arc).

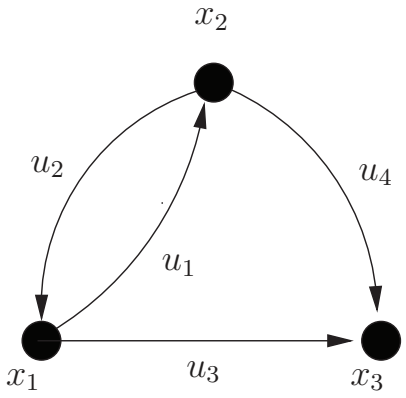


	x_1	x_2	x_3	destination
x_1	0	1	1	
x_2	1	0	1	
x_3	0	0	0	
origine				

FIG. 1.6 – Matrice d'adjacence d'un graphe

Matrice d'incidence sommets-arcs

Si $u = (i, j) \in U$, on trouve dans la colonne u : $a_{iu} = 1 : a_{ju} = -1$; tous les autres termes sont nuls (Fig. 1.7).



	u_1	u_2	u_3	u_4	arcs
x_1	1	-1	1	0	
x_2	-1	1	0	1	
x_3	0	0	-1	-1	
sommets					

FIG. 1.7 – Matrice d'incidence d'un graphe

Listes d'adjacence

Pour un graphe, l'avantage de la représentation par listes d'adjacence par rapport à celle par matrice d'adjacence, est le gain obtenu en place mémoire ; ce type de représentation est donc mieux adapté pour une implémentation [64]. Le but est de représenter chaque arc par son extrémité finale, son extrémité initiale étant définie implicitement. Tous les arcs émanant d'un même sommet sont liés entre eux dans une liste. A chaque arc sont donc associés le noeud destination et le pointeur au prochain sommet dans la liste (Fig. 1.8). On crée deux tableaux LP (tête de listes) de dimension (n + 1) et LS (liste de successeurs) de dimension m (cas orienté) ou 2m (cas non orienté). Pour tout i, la liste des successeurs de i est dans le tableau LS à partir de la case numéro LP(i).

1. On construit LS.
2. On construit LP qui donne pour tout sommet l'indice dans LS où commencent ses successeurs.
3. Pour un sommet $i \rightarrow$ 1er suivant : $LS(LP(i))$; 2ème suivant : $LS(LP(i)+1)$. L'ensemble des informations relatives au sommet i est contenue entre les cases $LP(i)$ et $LP(i+1)-1$ du tableau LS.
4. Si un sommet i n'a pas de successeur, on pose $LP(i) = LP(i+1)$ (liste vide coincée entre les successeurs de i - 1 et ceux de i + 1). Pour éviter des tests pour le cas particulier $i = n$ (le sommet i + 1 n'existant pas), on « ferme »par convention la dernière liste en posant $LP(n+1) = m+1$.

1.2.3 Coloration des sommets d'un graphe

Définition 1.2 *Soit $G = (X, U)$ un graphe non orienté. Un sous-ensemble $S \subset X$ est un ensemble stable (ou plus simplement un stable) s'il ne comprend que des sommets non adjacents deux à deux : $\forall i, j \in S \Rightarrow (i, j) \notin U$ [38].*

Comme tout sous-ensemble d'un ensemble stable est un ensemble stable, il est naturel de chercher le cardinal maximum d'un ensemble stable (un stable maximal). Ce nombre, noté $\alpha(G)$, est le nombre de stabilité.

Définition 1.3 *La coloration des sommets d'un graphe consiste en une affectation de couleurs à tous les sommets du graphe de telle sorte que deux sommets adjacents ne soient pas porteurs de la même couleur (Fig. 1.9).*

Le plus petit k pour lequel G est k-colorable est le nombre chromatique de G noté $\gamma(G)$; le nombre chromatique est donc défini comme le nombre minimum de couleurs distinctes nécessaires à la coloration des sommets de G. Une k-coloration des sommets est une partition $(S_1, S_2, ..., S_k)$ de l'ensemble des sommets en k ensembles stables.

1.2.4 Connexité dans les graphes

Chaîne - Cycle

Une chaîne est une séquence d'arcs telle que chaque arc ait une extrémité commune avec le suivant. Un cycle est une chaîne qui contient au moins une arête, telle que toutes les arêtes de la séquence soient différentes et dont les extrémités coïncident.

Chemin - Circuit

Ce sont les mêmes définitions que les précédentes mais en considérant des concepts orientés.
Le sous-ensemble de sommets atteignables à partir d'un sommet donné, grâce à des chemins, est appelé *fermeture transitive* de ce sommet. Le terme de parcours regroupe les chemins, les chaînes, les circuits et les cycles. Un parcours est :

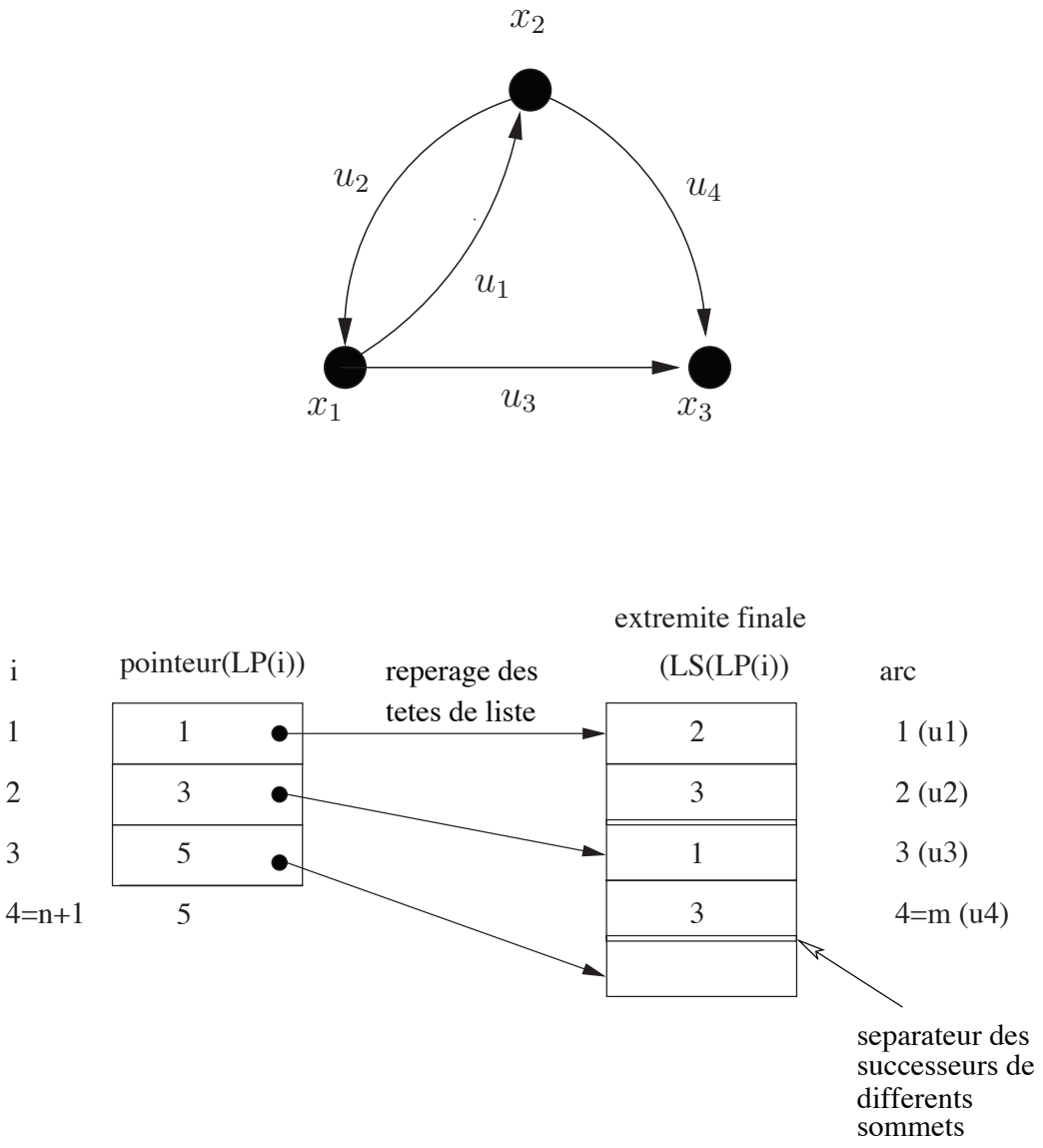


FIG. 1.8 – Liste d'adjacence d'un graphe

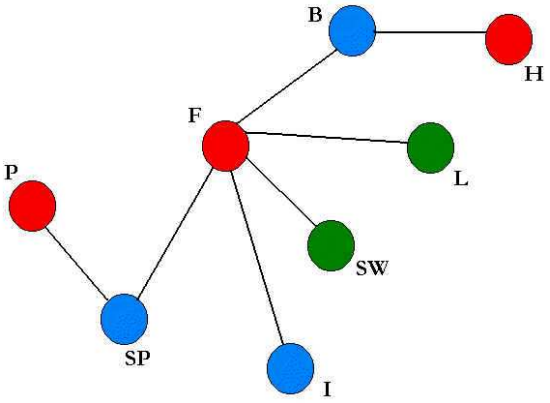


FIG. 1.9 – Coloration d'un graphe en 3 couleurs

- *élémentaire* : si tous les sommets qui le composent sont tous distincts ;
- *simple* : si tous les arcs qui le composent sont tous distincts ;
- *hamiltonien* : passe une fois et une seule par chaque sommet du graphe ;
- *eulérien* : passe une fois et une seule par chaque arc du graphe ;
- *préhamiltonien* : passe au moins une fois par chaque sommet du graphe ;
- *préeulérien* ou *chinois* : passe au moins une fois par chaque arc du graphe.

Connexité

Un graphe $G = (X, U)$ est *connexe* si $\forall i, j \in X$, il existe une chaîne entre i et j. On appelle composante connexe le sous-ensemble de sommets tels qu'il existe une chaîne entre deux sommets quelconques (Fig. 1.10). Un graphe est connexe s'il comporte une composante connexe maximale et une seule. Chaque composante connexe est un graphe connexe.

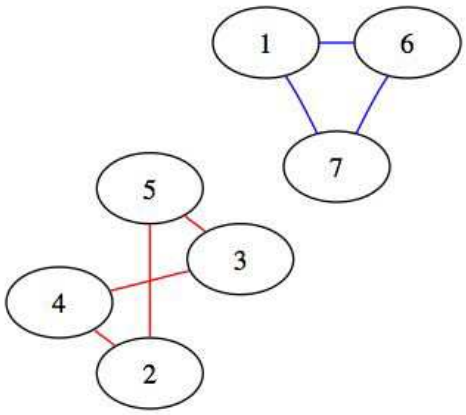


FIG. 1.10 – Graphe possédant 2 composantes connexes.

Forte connexité

Un graphe $G = (X, U)$ est fortement connexe si $\forall i, j \in X$, il existe un chemin entre i et j. Une composante fortement connexe (cfc) est un sous-ensemble de sommets tel qu'il existe un chemin entre deux sommets quelconques. Une cfc maximale (cfcm) est un ensemble maximal de cfc. Les différentes cfcm définissent une partition de X. Un graphe est fortement connexe s'il comporte une seule cfcm.

1.2.5 Autres graphes

Il existe d'autres classes de graphes particuliers comme les graphes sans circuit, les graphes bipartis ou les graphes planaires. Nous n'entrerons pas dans leur description ici, les notions présentées ci-dessus étant suffisantes à la compréhension de ce chapitre.

1.3 Modélisation des réseaux complexes par la théorie des graphes

1.3.1 Propriétés d'un graphe

Dans les différentes études concernant les réseaux complexes et la théorie des graphes, d'autres propriétés du graphe sont étudiées car les définitions précédentes, même si elles donnent une idée du type graphe auquel nous pouvons avoir à faire, ne sont pas suffisantes pour caractériser un réseau. Ci-dessous sont données les définitions de *coefficient de regroupement*, *assortativité* et *d'augmentation d'un graphe*.

Définition 1.4 ***(Coefficient de regroupement)*** *[84]* Le coefficient de regroupement C également appelé "clustering coefficient" est la probabilité que deux voisins d'un sommet donné soient voisins entre eux. Cela correspond à la densité locale d'un sommet.
Soit d_i le degré d'un sommet i,

$$C(i) = 2 \times \frac{|nbre\ de\ liens\ entre\ les\ voisins\ de\ i|}{d_i(d_i - 1)}$$

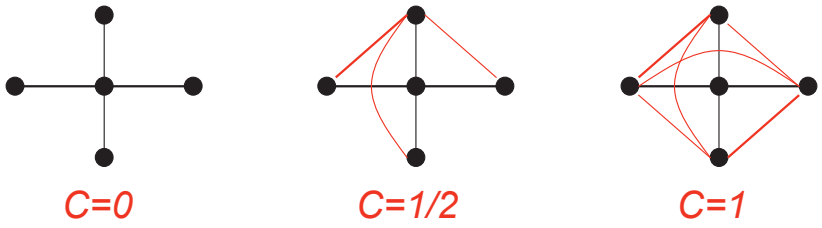


FIG. 1.11 – Exemples de coefficients de regroupement

Le degré de corrélation se calcule grâce à la moyenne des degrés des voisins d'un noeud. Pour faire simple, cela répond à la question : est-ce que dans un réseau, les sommets ayant un degré élevé sont de préférence connectés à d'autres sommets avec un degré élevé, ou sont plutôt connectés à des sommets ayant un faible degré ? La définition 1.5 permet de formaliser cette propriété.

Définition 1.5 ***(Degré de corrélation et assortativité)*** *[24]* Degré moyen des voisins d'un sommet i (avec arêtes pondérées) :

$$k_{nn,i} = \frac{1}{k_i} \sum_j a_{ij} k_j$$

Assortativité d'un graphe :

$$k_{nn}(k) = \frac{\sum_i \delta(k_i - k) k_{nn,i}}{\sum_i \delta(k_i - k)}$$

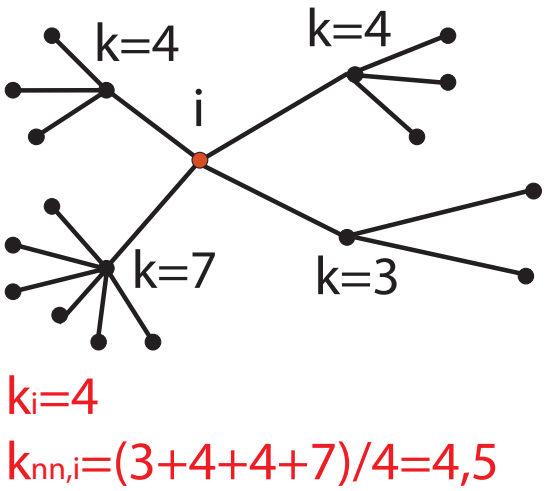


FIG. 1.12 – Exemple du degré de corrélation d'un sommet i

On peut remarquer deux cas distincts quant au comportement assortatif de $k_{nn}(k)$. Dans le cas où $k_{nn}(k)$ est *croissant*, on observe que les noeuds fortement connectés sont reliés aux noeuds fortement connectés. Cette caractéristique s'observe, par exemple, dans les réseaux sociaux (popularité). En revanche, dans le cas où $k_{nn}(k)$ est *décroissant*, on observe que les noeuds fortement connectés sont reliés aux noeuds faiblement connectés, ce qui engendre une structure hiérarchique. Internet est un réseau qui possède cette caractéristique.

Définition 1.6 ***(Graphe augmenté)*** *[42]* Un graphe augmenté $G = (V, E, E')$ est un graphe obtenu à partir d'un graphe $H = (V, E)$, en ajoutant un ensemble d'arêtes supplémentaires E' sur V. La distance sous-jacente de u à v dans G est la distance de u à v dans H.

D'autres propriétés moins connues et à priori non exploitées existent comme *l'excentricité* d'un sommet qui est la distance maximum de ce sommet aux autres sommets, et le *rayon* qui est la plus petite des excentricités. Dans le cas orienté, il faut distinguer excentricités entrantes et sortantes et, de ce fait, rayons entrants et sortants (le diamètre n'est pas sujet à ces différences).
La centralité de degré (la plus simple) considère le degré (pour un graphe non orienté, le demi-degré intérieur ou extérieur pour un graphe orienté) d'un nœud : c'est-à-dire le nombre de connexions directes irriguant ce nœud. Un noeud est d'autant plus central qu'il est directement lié à un grand nombre de noeuds. La centralité de degré mesure (localement) la capacité d'un noeud à communiquer, indépendamment de la centralité des noeuds auxquels il est directement lié. Or, intuitivement, un noeud lié à n noeuds périphériques n'est pas aussi central que celui qui est lié à n noeuds, eux-même centraux. Un noeud à la périphérie dépend des intermédiaires (plus centraux) pour sa communica-

tion avec les autres. Il ne contrôle pas sa communication [12].

Pour finir, on analyse souvent la *distribution des degrés* des sommets d'un graphe (Fig. 1.13). Deux distributions de degrés sont connues :

– une distribution homogène des degrés des noeuds (selon une loi de Poisson)

– une distribution hétérogène des degrés des noeuds (selon une loi de Puissance)

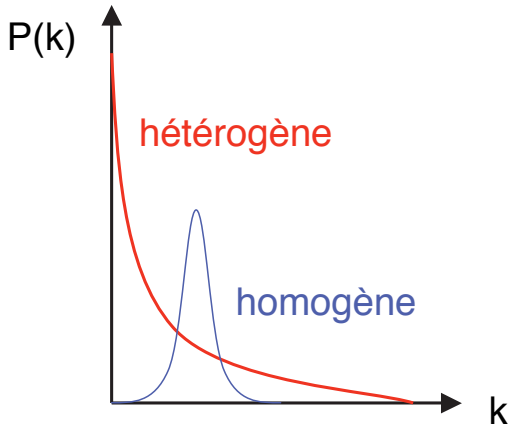


FIG. 1.13 – Courbes de distribution des degrés d'un graphe

Les graphes dont la distribution des degrés suit une loi de puissance ($p_k \sim k^{-\alpha}$) se retrouvent être la modélisation de la plupart des réseaux réels [82] comme le World Wide Web, Internet, les réseaux biologiques ou encore téléphoniques. La figure 1.14 donne la distribution des degrés d'un réseau d'acteurs, du World Wide Web, et d'un réseau électrique.

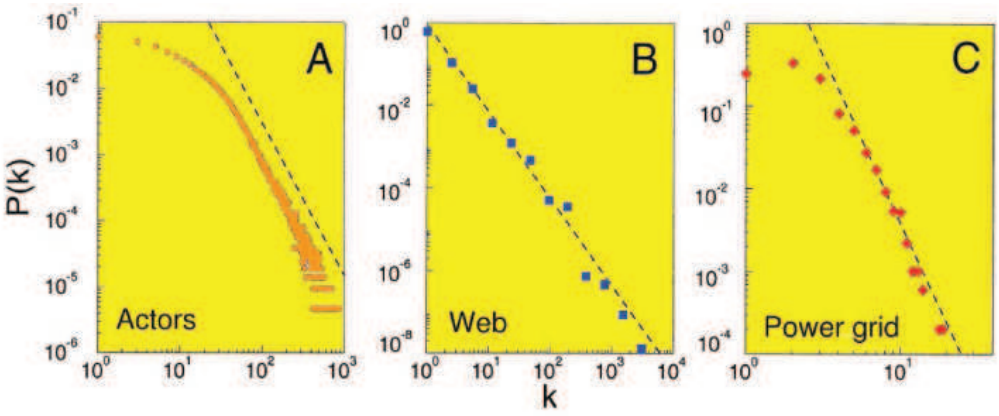


FIG. 1.14 – Distribution des degrés de réseaux réels

1.3.2 Structure & dynamique des graphes

L'étude conjointe des propriétés macroscopiques de différents types de réseaux contient plusieurs problématiques. Pour donner un sens à une propriété observée sur plusieurs ensembles de données statistiques, il faut s'assurer de sa pertinence. En particulier, il faut garantir que la méthode de collecte des données n'est pas biaisée et que la propriété est spécifique, c'est-à-dire qu'on ne l'observe pas sur tout ensemble de données arbitraire, ou aléatoire uniforme. Une fois qu'une propriété pertinente est isolée se pose la question de son apparition dans le réseau : est-ce dû à un processus dynamique de construction du réseau ? Comment peut-on la reproduire ? Révèle-t-elle un principe général sous-jacent ? Les réponses à ces questions peuvent avoir des applications intéressantes si la propriété présente un avantage pratique (on peut par exemple penser aux mécanismes de diffusion de rumeurs dans un réseau social pour une entreprise).

La structure d'un réseau peut être analysée en utilisant les définitions et propriétés liées aux graphes. La plupart des études sur les réseaux complexes se servent de propriétés globales du réseau pour en déterminer la nature. Le problème est que la dynamique du réseau a son rôle à jouer également. En effet, nous verrons, par la suite, que la structure d'un réseau ne suffit pas pour le caractériser de façon certaine. Le côté dynamique d'un réseau est souvent mis de côté, hors, c'est en couplant structure et dynamique que l'on pourra caractériser de manière précise un réseau. Certains articles traitent de la partie dynamique par l'étude d'algorithmes de routage, de navigabilité ou de construction de réseau.

Voici les trois principales propriétés qui ont été observées expérimentalement [54] :
- le *diamètre*
- la *distribution des degrés*
- la *densité locale (ou coefficient de regroupement)*

Diamètre

Le diamètre d'un réseau est le plus long des plus courts chemins entre deux entités, ou noeuds, du réseau, via ses connexions. Dans l'étude des grands réseaux d'interactions, il est souvent fait référence à un petit diamètre pour décrire l'observation d'une petite distance moyenne, le diamètre étant généralement trop coûteux à obtenir (puisqu'il s'agit du pire des cas). Dans de nombreux réseaux d'interactions, la distance moyenne observée est, de façon surprenante, de l'ordre du logarithme du nombre total de noeuds, alors que le nombre de connexions, ou liens, reste très inférieur au carré du nombre de noeuds.
Si les chemins sont très courts par rapport à la densité de liens des réseaux, la pertinence de cette propriété peut toutefois être remise en cause lorsque l'on remarque qu'un réseau aléatoire uniforme, où chaque paire de noeuds est reliée indépendamment avec une probabilité fixée, présente également un diamètre logarithmique en son nombre de noeuds lorsque le nombre d'arêtes est très faible (de l'ordre de $nlog(n)$ pour n noeuds) [84].

Distribution des degrés

Le modèle usuel utilisé au départ pour ce domaine d'étude était un réseau aléatoire uniforme, sur lequel on observe un effet de seuil pour la transmission d'un virus, c'est-à-dire qu'en dessous d'une fraction d'individus infectés, le virus cesse de se répandre. Mais une étude similaire menée sur un modèle présentant une distribution de degrés en loi de puissance a donné des résultats différents, en particulier l'effet de seuil disparaît. Une telle observation a donc remis en cause les mécanismes mis en place pour freiner la propagation des virus et les modèles utilisés jusqu'alors. Des modèles suivant une loi de puissance sont dorénavant utilisés, car c'est cette distribution qui est retrouvée la plupart du temps dans tous les réseaux réels [71].
Toutefois la pertinence de la loi de puissance comme propriété caractéristique a été mise en doute, en mettant en évidence un risque de biais lié à la méthode de collecte des données. En parcourant un réseau aléatoire uniforme selon un processus similaire à celui utilisé pour les parcours expérimentaux du réseau des routeurs Internet (le parcours par traceroute à partir d'un faible nombre de sources), on obtient des données statistiques présentant une distribution des degrés suivant une loi de puissance. Il est pourtant connu que le réseau aléatoire uniforme présente une distribution des degrés suivant une loi de Poisson [10]. Cette étude souligne le risque de baser la construction de modèles sur une propriété qui pourrait n'être qu'un biais de la mesure expérimentale.

Densité locale

On parle de forte densité locale d'un réseau lorsque les voisins d'un même noeud sont très reliés entre eux (Voir définition 1.4). Toutefois, cette propriété semble encore mal définie par le coefficient de regroupement. Il est en effet possible de construire des réseaux ayant un fort coefficient de regroupement alors que, majoritairement, les noeuds n'ont pas de voisins reliés entre eux. Sur la figure Fig. 1.15, le coefficient de regroupement vaut environ $1/2$ alors que $n/2$ noeuds sont disposés en chaîne et n'ont aucun de leurs voisins reliés entre eux.

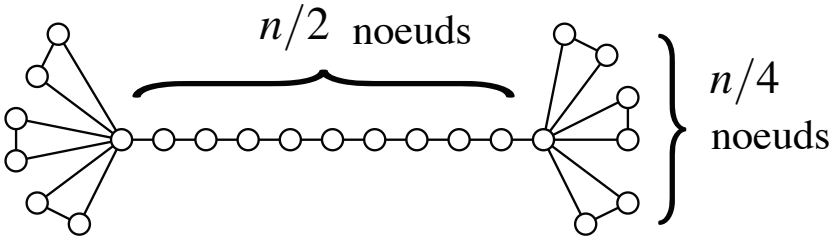


Fig. 1.15 – Exemple de biais du coefficient de regroupement

1.4 Modélisations existantes : graphes aléatoires, invariance d'échelle et petits-mondes

1.4.1 L'expérience de Milgram

Stanley Milgram était un psychologue social américain dont les recherches ont été menées sur de nombreux fronts. Milgram reprend une idée développée par Frigyes Karinthy, écrivain hongrois, qui est l'inventeur du concept des *six degrés de séparation*, dans sa nouvelle *« Chaînes »* de 1929. Cette théorie dit que chacun d'entre nous sur la planète peut être connecté à une autre personne en suivant une chaîne de connaissances ne contenant pas plus de six intermédiaires [65].
Dans sa première expérience décrite dans un papier appelé « Results of Communication Project », Milgram envoie soixante lettres à différents volontaires à Omaha dans le Nebraska, à qui il fut demandé de faire suivre les lettres à un agent de change vivant à Sharon, près de Boston, dans le Massachusetts. Les participants ne pouvaient faire transiter les lettres que de main à main à des personnes qu'ils jugeaient le plus susceptible de connaître la personne cible (directement ou par le biais de connaissances). Avec cinquante personnes relevant le challenge, seules trois lettres arrivèrent à destination. L'article de Milgram [65] de 1967 explique que l'une des lettres a atteint sa cible en quatre jours seulement, mais l'auteur n'a pas mentionné que seul 6% des lettres ont réussi à atteindre leur but.
Dans deux expériences ultérieures, la réussite de construction de chaîne était si faible que les résultats n'ont jamais été publiés. Pour expliquer cela, des chercheurs ont montré que nombre de facteurs secondaires peuvent affecter les résultats de ces expériences. Des études qui ont essayé de connecter des personnes de différentes couleurs ou origines ont montré de significatives asymétrie. Evidemment, l'article révélant que le taux de réussite d'une telle expérience était de 13% pour des personnes cibles noires et de 33% pour des personnes cible blanches (en dépit du fait que les participants n'aient pas su la couleur de la personne cible) fut co-écrit par Milgram lui-même.

Malgré ces complications, une variété de nouvelles découvertes a émergé des recherches de Milgram. Après de nombreux affinements du mécanisme de l'expérience (la valeur perçue de la lettre qui influençait le comportement des participants dans leur motivation à la transmettre ou pas), Milgram fut capable de pousser le taux de réussite à 35%, et plus tard, d'autres chercheurs poussèrent même le résultat à 97%. S'il y avait peu de certitudes pour que le monde entier soit un *small-world*, il y avait en revanche beaucoup de chance que le monde entier soit composé de plusieurs *small-worlds*. Pour toutes les chaînes qui ont réussi à faire transiter la lettre jusqu'à la cible, le nombre six émergea, étant la moyenne du nombre de personnes intermédiaires nécessaires. Ainsi, l'expression des *« six degrés de séparation »* était née (Fig. 1.16). De plus, Milgram identifia un effet « entonnoir » car le plus grand nombre de transits était souvent fait par un faible nombre de personnes, ayant une connectivité supérieure à la moyenne : même dans l'expérience avec 6% de réussite, Milgram avait noté que deux sur trois des chaînes avait été complétées par les mêmes personnes.

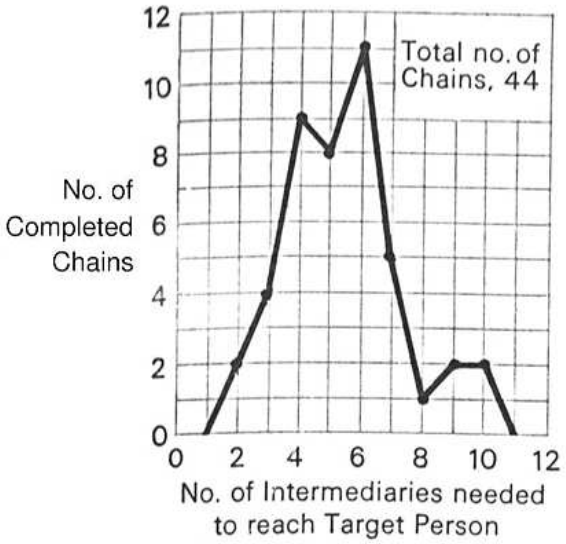


FIG. 1.16 – Les six degrés de séparation de Milgram

1.4.2 Les graphe aléatoires

Le modèle d'Erdös et Rényi

La définition d'un des tout premiers modèles théoriques de réseau a été établie en 1959 par Paul Erdös et Alfréd Rényi [32]. Ce modèle minimaliste consiste en un ensemble de n noeuds reliés par des arêtes qui sont placées de manière *aléatoire uniforme* entre des paires de noeuds. Erdös et Rényi ont donné plusieurs versions de leur modèle.
Le plus communément étudié est celui dénommé $G_{n,p}$ où chaque arête entre deux noeuds est indépendamment présente avec une probabilité p et absente avec une probabilité $1-p$. Souvent, la caractéristique d'un graphe $G_{n,p}$ n'est pas exprimée en fonction de p mais plutôt en fonction du degré moyen z des noeuds. Le nombre moyen d'arêtes est $\frac{1}{2}n(n-1)p$ et le nombre moyen de connexion correspond à 2 fois ce résultat puisque que chaque arête a deux extrémités. On peut donc en déduire le degré moyen des noeuds :

$$z = \frac{n(n-1)p}{n} = (n-1)p \simeq np$$

L'approximation np est d'autant plus précise que n est grand. Ainsi, une fois que l'on connaît n, chaque propriété exprimable en fonction de p peut l'être également en fonction de z.
Le modèle d'Erdös et Rényi possède plusieurs propriétés intéressantes. Par exemple, une caractéristique intéressante démontrée dans les papiers originaux de l'époque est qu'une *phase de transition* apparaît en fonction de z, ce qui provoque la formation d'une *composante géante* (Fig. 1.17) [11].

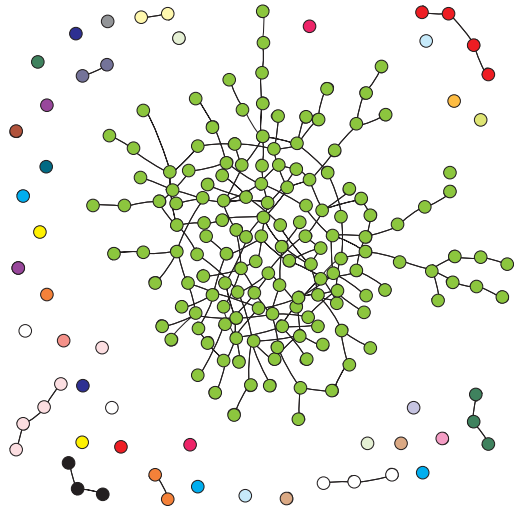


FIG. 1.17 – Exemple de graphe aléatoire $G_{n,p}$ avec sa composante géante au centre

Une composante est un sous-ensemble de noeuds du graphe qui s'avère être un sous-graphe *connexe* (c'est-à-dire que tous les noeuds sont connectés entre eux par au moins un chemin). Pour des petites valeurs de z, quand il y a peu de noeuds dans le graphe, il n'est pas surprenant de constater que la plupart des noeuds ne sont pas connectés entre eux, et que les composantes sont petites, avec une taille moyenne qui reste constante au fur et à mesure que le graphe augmente en taille. Cependant, il y a une valeur critique de z à partir de laquelle la plus grosse des composantes du graphe contient une fraction finie S du nombre total de noeuds, sa taille nS augmentant linéairement avec la taille du graphe entier. Cette composante est la *composante géante*. En général, en plus de celle-ci, il y a d'autres composantes, mais elles restent petites, ayant une taille moyenne qui reste constante au fur et à mesure que le graphe s'agrandit. La phase de transition à laquelle la composante géante se forme se déclenche à $z = 1$ [22] (Fig. 1.18).

La formation d'une composante géante dans un graphe aléatoire fait penser au comportement de nombreux réseaux réels. Par exemple, un réseau social où les gens ont connectés, est très dense et possède probablement une composante géante.
Pourtant, les graphes aléatoires diffèrent des réseaux réels sur de nombreux points. Deux différences notables ont été remarquées par Strogatz [79] et par Albert et Barabási [6]. Premièrement, les réseaux réels montrent de forts coefficients de regroupement (clustering) [84] alors que ce n'est pas le cas pour le modèle d'Erdös et Rényi. En effet, dans ce modèle, les probabilités qu'une paire de noeuds soit connectées par une arête sont indépendantes, il y a donc pas de différence au niveau probabilité de connexion entre deux noeuds ayant un voisinage commun et deux noeuds n'en ayant pas. Cela signifie que le coefficient de regroupement pour un graphe aléatoire est tout simplement égal à $C = p$ ou de manière équivalente $C = \frac{z}{n}$. La comparaison de coefficients de regroupement entre graphes aléatoires et réseaux réels se trouve sur la figure 1.19.

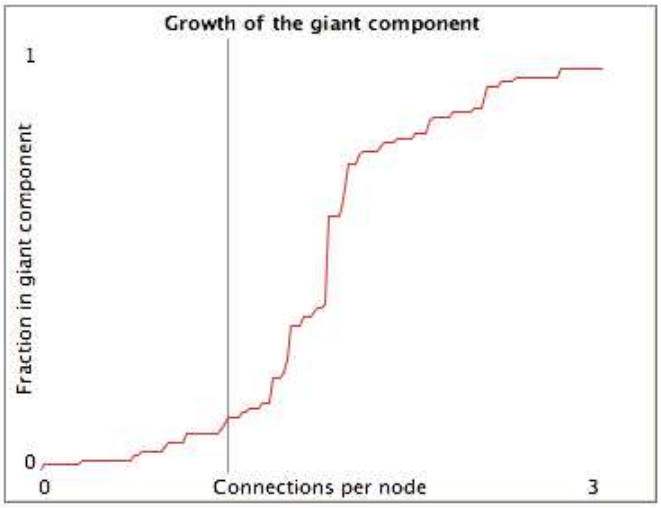


FIG. 1.18 – Phase de transition de formation de la composante géante dans un graphe aléatoire $G_{n,p}$

			clustering coefficient C	
network	n	z	measured	random graph
Internet (autonomous systems)	6 374	3.8	0.24	0.00060
World-Wide Web (sites)	153 127	35.2	0.11	0.00023
power grid	4 941	2.7	0.080	0.00054
biology collaborations	1 520 251	15.5	0.081	0.000010
mathematics collaborations	253 339	3.9	0.15	0.000015
film actor collaborations	449 913	113.4	0.20	0.00025
company directors	7 673	14.4	0.59	0.0019
word co-occurrence	460 902	70.1	0.44	0.00015
neural network	282	14.0	0.28	0.049
metabolic network	315	28.3	0.59	0.090
food web	134	8.7	0.22	0.065

FIG. 1.19 – Comparaison du coefficient de regroupement entre réseaux réels et graphes aléatoires

Comme on peut le constater, la correspondance du coefficient de regroupement entre réseaux réels et graphes aléatoires n'est pas bonne. Les graphes aléatoires ne reproduisent donc pas cette propriété importante des réseaux réels.
La seconde chose qui diffère entre les réseaux réels et les graphes aléatoires est la distribution de leur degré [6]. La probabilité p_k qu'un noeud dans un graphe aléatoire possède un degré égal à k est donnée par la distribution binomiale suivante :

$$p_k = \left(\begin{array}{c} n-1 \\ k \end{array} \right) p^k (1-p)^{n-1-k}$$

Quand $n \gg kz$, on obtient :

$$p_k = \frac{z^k e^{-z}}{k!}$$

qui s'avère être la distribution de Poisson (homogène). Quand on regarde la distribution de réseaux réels (Fig. 1.14), on peut aisément remarquer qu'elle ne correspond pas du tout à une distribution de Poisson (Fig. 1.13) mais bien à une distribution suivant une loi de puissance ([6]).
On peut donc en conclure que le graphe aléatoire d'Erdös et Rényi n'est pas un modèle permettant de représenter les réseaux réels.

Les graphes aléatoires généralisés

Il existe une classe de graphe proches des graphes aléatoires d'Erdös et Rényi à la seule différence que le degré de distribution des noeuds ne suit pas forcément une loi de Poisson. On peut, en théorie, lui faire suivre n'importe quelle loi de distribution. Abordée en premier lieu par Bender et Canfield en 1978 [8], la généralisation de ces graphes a été formalisée en 1995 par Molloy et Reed [66].
La méthode de construction de base est de se tenir à une séquence de degré spécifique, c'est-à-dire à un ensemble $\{k_i\}$ de degrés des noeuds $i = 1 \ldots n$. En fait, cet ensemble sera choisi de manière à ce que la fraction de noeuds ayant comme degré k tende vers la distribution de degrés désirée p_k au fur et à mesure que n augmente. Une fois que chaque noeud possède sa séquence de degré, la méthode pour générer le graphe est la suivante : on donne à chaque noeud i un nombre k_i de "points de connexion" (un point de connexion correspond à l'extrémité d'une arête), et on choisit des paires de ces points de connexion de manière aléatoire uniforme pour joindre les arêtes. Quand tous les points de connexion ont été utilisés, le graphe résultant est un membre choisi de façon aléatoire parmi l'ensemble des graphes possédant la séquence du degré désiré.

Plus récemment, une classe générale de graphes encore plus large intitulée « modèles de graphes aléatoires non-homogènes »a été formalisée par Söderberg en 2002 [81]. Ces graphes sont construits en imposant un type de structure aux noeuds. Les modèles de cette classe ne sont pas restreints au seul choix de la distribution du degré, cependant, une infinité de modèles existe pour chaque type de distribution. Ces modèles peuvent, dans certains cas, correspondre à certains types de réseaux dynamiques, même si leur

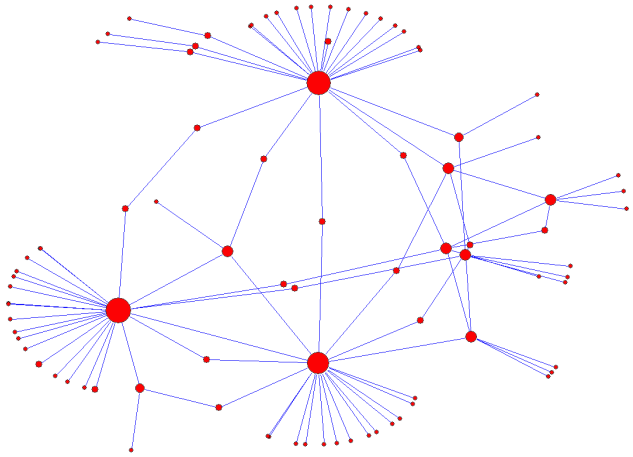


FIG. 1.20 – Graphe aléatoire généralisé de Molloy & Reed suivant une loi de puissance

défaut majeur est le côté artificiel de leur génération.

1.4.3 Modèle d'Albert et Barabási

La première personne à introduire les *réseaux à invariant d'échelle*[1] fût Price en 1965. Il mena une étude à propos des citations d'articles scientifiques, en étudiant les degrés entrants (nombres de fois qu'un papier a été cité) et sortants (nombre de papiers cités par un autre papier), il découvrit une distribution suivant une *loi de puissance*, ce qui fit l'objet d'un article [76]. En fait, cela correspondait au fait que plus le dégré d'un élément est élevé, plus ce degré augmente. Price appela cette propritété *avantage cumulatif*. Cette appellation est maintenant connue sous le terme *attachement préférentiel*, introduit par Albert et Barabási en 1999 [6]. L'idée de Price était que le nombre de citations reçues par un papier était proportionnel au nombre de citations qui le désignaient auparavant. On peut noter le même comportement pour d'autres réseaux comme par exemple le World Wide Web.

Le modèle d'Albert et Barabási (Fig. 1.21) est le même que celui de Price à la différence près que les arêtes d'Albert et Barabási sont *non orientées*. Dans un réseau de citations par exemple, le fait que les arêtes soient non orientées fait perdre une information cruciale au réseau. Mais d'un autre côté, cette caractéristique permet d'éviter le problème de Price de savoir comment un papier reçoit sa première citation (ou un site Web son premier lien). Chaque noeud dans le graphe apparaît avec un degré initial m, donc avec une probabilité non-nulle de recevoir de nouveaux liens. La probabilité qu'un nouveau noeud « s'attache » à un noeud de degré k est :

$$\frac{kp_k}{\sum_k kp_k} = \frac{kp_k}{2m}$$

[1] également appelés *scale-free networks*

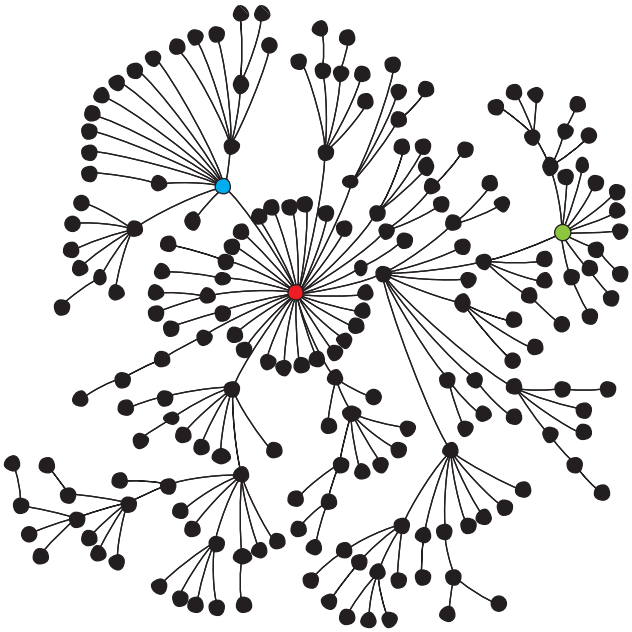


FIG. 1.21 – Graphe construit suivant le modèle d'attachement préférentiel d'Albert et Barabási

La somme du dénominateur est égale au degré moyen du réseau, qui est $2m$, puisqu'il y a m arêtes ajoutées pour chaque noeud ajouté, et comme celles-ci sont non orientées, chacune d'elle possède deux extrémités incrémentant les degrés des noeuds connectés. A partir de là, le nombre moyen de noeuds de degré k qui gagne une connexion d'arête quand un nouveau noeud ayant m arêtes est ajouté est : $m \times kp_k/2m = \frac{1}{2}kp_k$ (indépendamment de m). Le nombre np_k de noeuds de degré k décroît ainsi suivant ce même résultat, alors que les noeuds qui gagnent de nouvelles arêtes deviennent des noeuds de degré $k+1$. Le nombre de noeuds de degré k augmente également à cause de l'afflux des noeuds qui étaient précédemment de degré $k-1$ et qui venaient juste d'acquérir une nouvelle arête (sauf les noeuds de degré m, qui ont un afflux d'exactement 1).
Soit $p_{k,n}$ la valeur de p_k avec un graphe à n noeuds alors le comportement du réseau en fonction de np_k par noeud ajouté est :

– pour $k > m$:

$$(n+1)p_{k,n+1} - np_{k,n} = \frac{1}{2}(k-1)p_{k-1,n} - \frac{1}{2}kp_{k,n}$$

– pour $k = m$:

$$(n+1)p_{m,n+1} - np_{m,n} = 1 - \frac{1}{2}mp_{m,n}$$

– pour $k < m$: aucun noeud n'est ajouté

Si l'on regarde les solutions de $p_{k,n+1} = p_{k,n} = p_k$, on obtient les équations suivantes :

$$p_k = \begin{cases} \frac{1}{2}(k-1)p_{k-1} - \frac{1}{2}kp_k & \text{pour } k > m \\ 1 - \frac{1}{2}mp_m & \text{pour } k = m \end{cases}$$

Ainsi, voici ce qu'obtiennent Albert et Barabási :
$p_m = 2/(m+2)$ et $p_k = p_{k-1}/(k+2)$, soit :

$$p_k = \frac{(k-1)(k-2)...m}{(k+2)(k+1)...(m+3)}$$

$$p_m = \frac{2m(m+1)}{(k+2)(k+1)k}$$

Aux limites d'un grand k, cela donne une distribution suivant une loi de puissance $p_k \sim k^{-3}$, avec un exposant fixe $\alpha = 3$.

Un élément crucial de ce modèle est l'hypothèse que l'attachement préférentiel est *linéaire*. A priori, on peut le remarquer de manière empirique, mais aucune preuve formelle n'a été élaborée.
Un grand nombre d'études concernant le modèle d'Albert et Barabási ont été menées, ce qui aurait été impossible trente ans plus tôt. Ce modèle a attiré beaucoup d'attention dans la littérature. En plus des études analytiques proposées, un bon nombre d'auteurs ont suggéré des modifications ou des extensions du modèle, qui altèrent son comportement ou qui le rendent un peu plus réaliste vis à vis des réseaux réels, car même si ce modèle paraît pertinent, nous obtenons un modèle de réseau sous forme d'*arbre*, alors que les réseaux du réel n'arborent presque jamais cette forme.

1.4.4 Les petit-mondes (Small-World) de Watts & Strogatz

Watts et Strogatz ont proposé un modèle en 1998 [84] qui peut être décrit comme tel : c'est un graphe régulier dans lequel les arêtes originales ont été replacées comme des arêtes aléatoires suivant une probabilité $0 \leq \phi \leq 1$. Ils découvrirent que le fait de rediriger quelques arêtes transforme le réseau (graphe régulier), à partir d'un certain seuil [28], en *small world* (Fig. 1.22), ayant des chemins courts entre deux noeuds (comme pour la composante géante des graphes aléatoires).

Bien que le réseau soit bien plus regroupé qu'un réseau aléatoire, c'est-à-dire que si A est relié à B et B est relié à C, il y a une bien plus grande probabilité que A soit également relié à C (une propriété appelée par les sociologues *transitivité*). Watts et Strogatz ont conjecturé que ces deux propriétés (courts chemins et fort regroupement) étaient observées dans de nombreux réseaux naturels et technologiques. Leur intuition était que des chemins courts pourraient fournir des canaux de communication rapides entre deux parties éloignées dans le système, facilitant ainsi les processus dynamiques (comme la synchronisation ou le calcul) qui requiert une coordination globale ainsi que des flux d'informations.

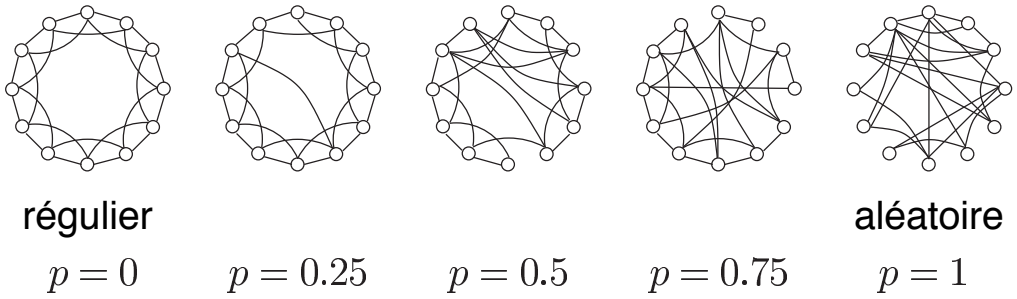


FIG. 1.22 – Modèle Small-World de Watts et Strogatz

De nombreux exemples de réseaux small-world ont été étudiés, dans des domaines allant de la biologie aux études de commerce. D'un point de vue théorique, les réseaux small-world sont utilisés de façon différente selon qu'il s'agit de problèmes différents et de scientifiques venant de disciplines différentes.
En informatique, les chercheurs se sont posés des questions à propos des algorithmes et de la complexité de ces small-world. Les épidémiologistes se sont demandés comment le regroupement local de noeuds et les contacts globaux influencent la diffusion de maladie infectieuse. Les neurobiologistes se sont interrogés sur l'évolution significative d'une architecture neuronale small-world.
Néanmoins, le modèle original proposé par Watts et Strogatz pose quelques problèmes. Le fait que seulement une seule des extrémités d'une arête soit redirigée, qu'aucun noeud ne puisse être connecté à lui-même, et qu'une arête ne soit jamais ajoutée entre des paires de noeuds où il en existe déjà une, rend difficile l'énumération ou le calcul de moyenne sur l'ensemble du graphe.

Il existe une variante intéressante de ce modèle proposée par Newman, Moore, et Watts [70]. Sous cette variante, le modèle est créé à partir d'un anneau de n noeuds, chacun connecté par des arêtes à ses voisins jusqu'à une distance de k (ce type de graphe est également appelé *graphe régulier* de degré k). Au lieu de les rediriger, des arêtes servant de « raccourcis » sont ensuite ajoutées entre des paires de noeuds choisis de manière aléatoire suivant une probabilité ϕ pour chaque arête, ce qui donne $nk\phi$ raccourcis dans le graphe (Fig. 1.23).

La question est : en moyenne, combien d'étapes sont nécessaires pour aller d'un noeud à l'autre en passant par le chemin le plus court ? Si on note l l'écart moyen entre deux noeuds, on observe que l diminue considérablement quand ϕ est proche de 0, confirmant ainsi qu'un petit nombre de raccourcis rétrécit énormément le « monde ». Un des résultats les plus frappants est la formule suivante dérivée par Newman, Moore, et Watts [70] :

$$l = \frac{n}{k} f(nk\phi)$$

avec :

$$f(x) = \frac{1}{2\sqrt{x^2 + 2x}} tanh^{-1} \frac{x}{\sqrt{x^2 + 2x}}$$

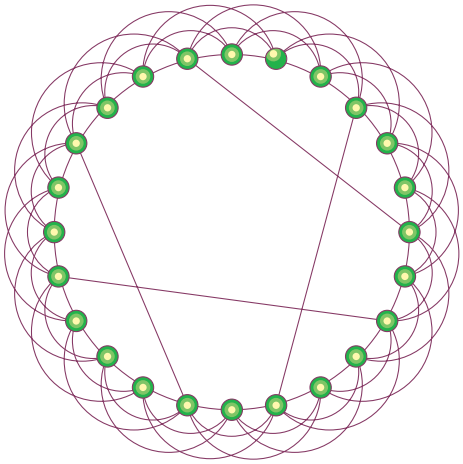


FIG. 1.23 – Variante du modèle Small-World de Watts et Strogatz

Cette solution est asymptotiquement exacte aux limites $n \to \infty$ (système de grande taille), $nk\phi \to \infty$ ou $nk\phi \to 0$ (grand nombre ou petit nombre de raccourcis).
Cette version du modèle possède une propriété intéressante : aucun noeud n'est déconnecté du reste du réseau, ainsi, la distance moyenne noeud à noeud reste toujours finie (graphe connexe). Ces deux modèles, l'original et sa variante, ont été longuement étudiés dans la littérature physique et mathématique.

Coefficient de regroupement

Pour les deux modèles présentés, le coefficient de regroupement peut être calculé [69] pour la version originale :

$$C = \frac{3(k-1)}{2(k-1)}(1-p)^3$$

ainsi que pour sa variante :

$$C = \frac{3(k-1)}{2(2k-1)+4kp(p+2)}$$

Distribution des degrés

La distribution des degrés du modèle small-world ne correspond pas à la plupart des réseaux réels [2]. Pour la version où on ne redirige pas les arêtes, chaque noeud possède un degré d'au moins $2k$, (idem pour les arêtes du graphe régulier sous-jacent), ainsi qu'une distribution binomiale du nombre de raccourcis.
Pour la version avec les arêtes redirigées, la distribution des degrés devient plus compliquée et s'approche plus de k que de $2k$.

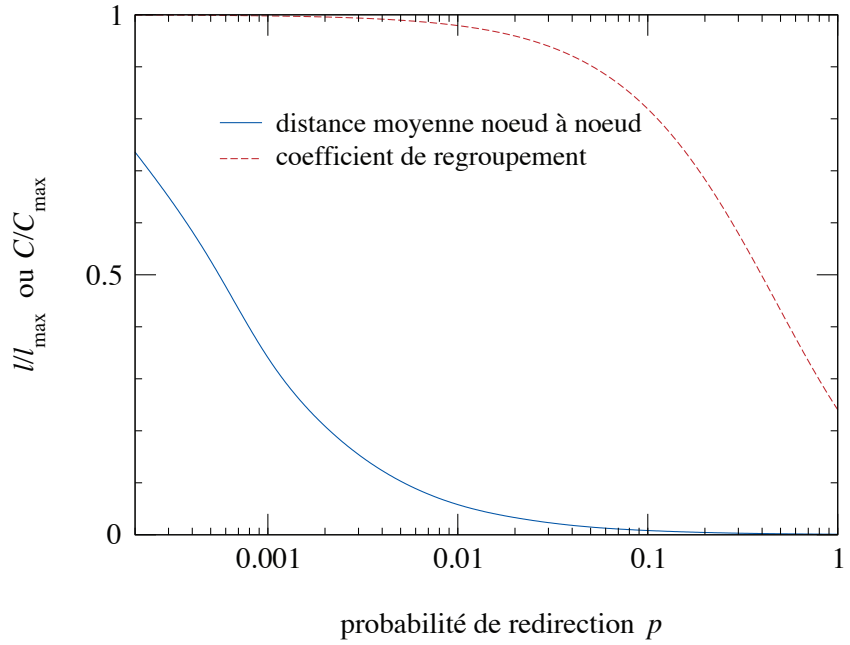


FIG. 1.24 – Coefficient de regroupement C et longueur moyenne noeud à noeud l en fonction de p du modèle Small World Watts et Strogatz

En cherchant à reproduire à la fois le petit diamètre présent dans les graphes aléatoires uniformes et le fort coefficient de clustering présent dans les grilles régulières, Watts et Strogatz ont eu l'idée d'introduire une part d'aléatoire sur un réseau régulier et ils ont obtenu les deux propriétés simultanées pour un assez grand intervalle de valeurs du paramètre p. Toutefois, ce modèle ne présente pas la propriété dynamique de navigabilité. Précisément, Kleinberg [40] a montré que tout algorithme de routage décentralisé calcule, entre deux noeuds de ce graphe, un chemin de longueur au moins polynomiale en n. Cela signifie que, même lorsque le paramètre p donne naissance à des chemins polylogarithmiques en n entre toute paire de noeuds, il n'existe pas d'algorithme qui puisse les découvrir en n'ayant qu'une vue locale du graphe, ce qui rend les travaux d'analyse sur ces modèles très délicats.

1.4.5 Modèle de petit-monde navigable de Kleinberg

Description

En 2000, à partir d'un article de l'année précédente [39], Kleinberg propose le premier modèle de petit monde présentant la propriété de *navigabilité*, c'est-à-dire le premier modéle de graphe dont *le diamétre est polylogarithmique en nombre de noeuds* et dont *des chemins polylogarithmiques peuvent être découverts par un algorithme décentralisé entre tout couple de sommets*. Il s'agit d'une grille carrée de dimension 2 et de côté n que l'on augmente en ajoutant à tout sommet u un nombre $k > 0$ constant d'arcs aléatoires tirés

indépendamment. La destination du j-ème arc de u, pour $1 \leq j \leq k$, est v avec une probabilité proportionnelle à $1/|u-v|^s$, où $|u-v|$ est la distance l_1 entre u et v dans la grille[2], et $s > 0$ est une constante. On remarquera que la grille est constituée d'arêtes (liens non-orientés) tandis que les arcs (liens orientés) ajoutés aléatoirement sont orientés.

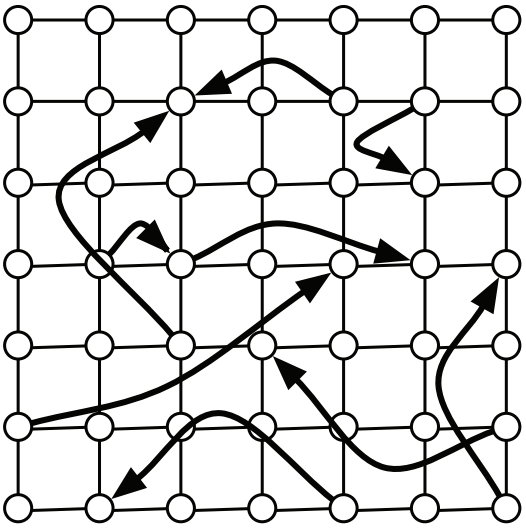


FIG. 1.25 – Exemple de réseau Kleinberg $K^2_{1,n}$

La figure 1.25 illustre un exemple de réseau de Kleinberg de dimension 2 ayant un lien aléatoire sortant par noeud (en gras). Les liens aléatoires ne sont pas tous représentés pour la lisibilité. Kleinberg démontre que seul le cas où $s = 2$ permet d'obtenir la propriété de navigabilité.
Il exhibe, pour ce cas, un algorithme de routage décentralisé[3] qui calcule des chemins dont l'espérance de la longueur est $O(log^2 n/k)$ entre tout couple de noeuds (on dit que la longueur est $O(log^2 n/k)$ en espérance).

Il s'agit de l'algorithme glouton G (Def. 1.7) qui transmet, en chaque noeud, le message au voisin le plus proche de la cible du routage.

Définition 1.7 ***(Algorithme de routage glouton de Kleinberg)***
Entrée *: la source s et la cible t.*
Initialisation *: $x \leftarrow s$. (x est le porteur courant du message)*
tant que $x \neq t$ faire
$x \leftarrow y$ où y est le contact local ou longue-distance de x qui minimise $|y-t|$ (choix aléatoire uniforme en cas d'égalité).
fin tant que

[2]distance de Manhattan
[3]l'expérience de Milgram est un bon exemple d'algorithme décentralisé

Ce résultat a été généralisé par la suite à toute grille de dimension $d > 0$, ainsi on obtient la propriété de navigabilité pour le seul exposant $s = d$. On obtient également que l'espérance de la longueur du chemin construit par l'algorithme glouton G est $\Theta(log^2 n/k)$. Lorsque $s \neq d$, tout algorithme de routage décentralisé calcule, entre deux noeuds, un chemin de longueur au moins polynomiale en espérance en le nombre de sommets. Ce résultat démontre en particulier qu'aucune distribution uniforme ($s = 0$) n'est navigable sur le tore $d - dimensionnel$, c'est le cas par exemple des modèles d'Erdös-Rényi et de Watts et Strogatz.

Classes de réseaux de Kleinberg

Définition 1.8 ***(Réseau de Kleinberg)***
Etant donnés $k, n, d \geq 1$, un réseau de Kleinberg de dimension d, noté $K^d_{k,n}$, est un tore régulier de dimension d et de côté n augmenté de k liens (ou arcs) longue-distance sortants par noeud, aléatoires et indépendants, tels que pour tout noeud u, la probabilité que la destination de son j-ème lien longue-distance soit v est proportionnelle à $1/|u - v|^d$.

On appelle lien local toute arête du tore sous-jacent. Sa destination est un contact local de son origine. L'ensemble des liens locaux représente l'information globale connue par tous les noeuds du réseau. Ces liens sont symétriques. On appelle lien longue-distance tout arc aléatoire du modèle. Sa destination est un contact longue-distance de son origine. Un lien longue-distance représente une information partielle, sans carte globale. Ces liens sont orientés. On note $L_j(u)$ le j-ème contact longue-distance d'un sommet u.

Le réseau de Kleinberg est le premier modèle à reproduire la propriété de petit monde navigable et il présente l'avantage d'être très simple, ce qui en fait un bon outil pour l'analyse de cette propriété. Toutefois, sa structure sous-jacente de tore régulier, qui représente la connaissance globale partagée par tous les noeuds, paraît peu vraisemblable pour un réseau réel. Si l'on fait l'hypothèse que la connaissance globale d'un réseau est l'ensemble des distances géographiques par exemple, on peut penser que celles-ci ne sont pas réparties selon une grille régulière. Des travaux récents se sont attachés à produire de nouveaux modèles de petit monde navigable, également construits à partir d'une structure sous-jacente que l'on augmente par des arcs aléatoires, pour chercher un modèle plus réaliste.

1.4.6 Modèle de petit-monde de Lebhar basé sur le modèle de Kleinberg

En 2005, Lebhar a soutenu une thèse sur les algorithmes de routage dans les petits mondes de Kleinberg [54], et de par son travail a soulevé les problèmes évoqués ci-dessus concernant les modèles petit-monde de Kleinberg. Ses recherches lui ont permis de développer un algorithme de routage bien plus efficace et proche de l'optimal (partie

dynamique), ainsi qu'une classe de petit monde de Kleinberg modifiée (partie statique) afin de faire correspondre d'avantage ce modèle aux réseaux réels.

Algorithme de routage optimisé de Lebhar

Pour une source et une cible donnée, l'algorithme va construire des structures d'explorations successives des contacts locaux et longue-distance, dont la profondeur dépend de la distance courante à la cible. A la fin de chaque exploration, le message à transmettre est envoyé de la racine de l'exploration au noeud le plus proche de la cible, parmi les noeuds explorés. Lebhar fait l'hypothèse que la découverte d'un lien longue-distance lors de l'exploration l'autorise à renvoyer une faible quantité d'information en sens inverse sur ce lien, ce qui permet à la racine de déterminer le noeud le plus proche de la cible dans l'ensemble exploré. En revanche, le message à transmettre emprunte les liens longue-distance suivant leur orientation. Cette hypothèse peut s'illustrer par le passage d'une lettre dans un réseau social : supposons que A connaît B mais que B ne connaît pas A ; si A envoie une lettre à B en indiquant son adresse au dos de l'enveloppe (découverte du lien longue-distance), B peut alors envoyer une lettre à A. On peut également illustrer ce fait par la navigation dans le réseau des pages web : supposons que la page web de C pointe sur celle de D, sans que celle-ci ait un hyperlien vers la page de C ; un internaute passant de la page de C à la page de D par l'hyperlien (découverte du lien longue-distance) peut revenir sur la page de C par la touche « back »du navigateur.

Soient s et t la source et la cible respectivement. On suppose que trois fonctions $h_{max}(x)$, $b_{max}(x)$ et $g(x)$, ainsi qu'un paramètre *stop* sont donnés. La fonction $h_{max}(x)$ représente la profondeur d'exploration pour une racine à distance x de t, $b_{max}(x)$ le nombre de feuilles que l'on cherche à obtenir, $g(x)$ la longueur des chaînes de liens locaux (Fig. 1.26). Enfin, *stop* est la distance seuil à t en dessous de laquelle on exécute l'algorithme glouton de Kleinberg car l'exploration devient inutile. La difficulté consiste à choisir les valeurs de ces fonctions de façon à minimiser la longueur du chemin $h_{max}(x) + g(x)$ tout en garantissant un nombre de feuilles suffisamment grand.

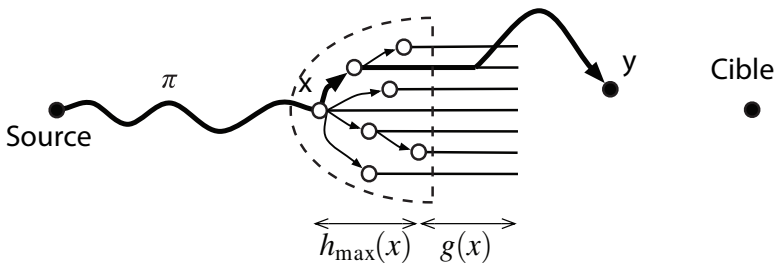


FIG. 1.26 – Extension du chemin π (en gras) à la fin d'une phase d'exploration.

Soit x le porteur courant du message et π le chemin courant de s à x :

1. Tant que $|x - t| > stop$:
– Explorer les noeuds à $h_{max}(x)$ liens valides de x suivant un parcours en profondeur, enregistrer dans un ensemble F tous les contacts longue distance valides visités, mais sauter dans l'exploration tous les contacts longue-distance à distance strictement inférieure à $h_{max}(x) + g(x)$ d'un noeud de l'ensemble F courant. Le parcours en profondeur est arrêté dès que $|F| = b_{max}(x)$, même s'il reste des noeuds à parcourir.
– A chaque fois qu'un noeud z, à exactement $h_{max}(x)$ liens valides de x, est rencontré, lire les adresses des contacts longue-distance des noeuds à $g(x)$ liens locaux valides de z, et enregistrer dans une variable y le noeud le plus proche de la cible parmi les noeuds visités et leurs contacts.
– A la fin de l'exploration en profondeur, router le message de x à y dans l'arbre d'exploration et étendre le chemin π jusqu'à y, suivant ces liens.

2. Lorsque $|x - t| \leq stop$: appliquer l'algorithme glouton de Kleinberg, transmettre le message au contact de x le plus proche de t et étendre le chemin π par ce lien, jusqu'à atteindre t.

La figure 1.26 illustre le principe de la structure visitée durant chaque phase d'exploration : les lignes droites représentent les liens locaux valides et les flèches représentent les liens longue-distance valides. La figure 1.27 illustre la succession des structures d'exploration durant un routage.

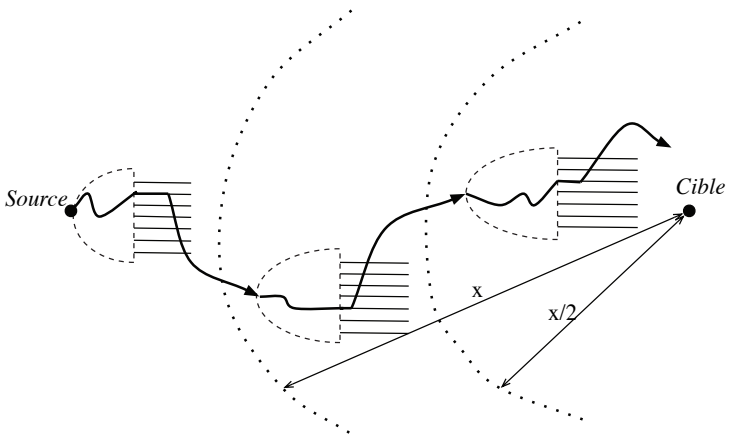


FIG. 1.27 – Routage de Lebhar par explorations successives

D'après Lebhar [54], il n'existe pas de meilleur algorithme décentralisé sur les réseaux de Kleinberg, c'est-à-dire qui calcule une longueur de chemin plus faible asymptotiquement pour tout degré sortant $O(log(n))$.

Généralisation du modèle de Kleinberg

Le modèle de petit monde de Kleinberg a montré que l'augmentation d'un tore régulier par des arcs aléatoires, dont la distribution est étroitement liée à la métrique de ce tore, produisait un graphe petit monde navigable. Ce fait pose une question essentielle pour la compréhension du phénomène petit monde : n'y a-t-il que des métriques spécifiques qui peuvent être transformées en petit monde par l'ajout de liens longue-distance ? Lebhar introduit ainsi le concept de processus de *petit-mondisation*.
Elle définit un processus d'augmentation de graphe en graphe petit monde navigable par l'ajout d'un nombre constant de liens longue-distance aléatoires par noeud, en exprimant la distribution des liens comme une fonction de la taille des boules[4] dans la métrique du graphe que l'on augmente, et démontre que l'on peut étendre le modèle de Kleinberg aux grilles infinies.
Une des principales critique faites au modèle de Kleinberg est qu'il ne modélise pas la distribution des degrés suivant une loi de puissance qui est observée sur de nombreux graphes réels, comme le graphe d'Internet. Pourtant, sur ce modèle comme sur celui que nous venons de présenter, il suffit de faire une modification mineure pour obtenir cette distribution des degrés et plus généralement, toute distribution de degrés qui assure un degré supérieur ou égal à 1 avec probabilité constante. C'est en fait le cas pour tout processus de petit mondisation construit en ajoutant 1 lien par noeud.
Lebhar énonce un théorème qui présente l'ensemble des distributions de degrés possibles pour la petit-mondisation et montre ainsi que la propriété de petit monde navigable ne semble pas corrélée à une distribution des degrés spécifique.
Le graphe est augmenté en ajoutant à chaque noeud, k liens indépendants distribués selon une famille de distribution suivant une certaine probabilité. En prenant une distribution suivant une loi de puissance, on obtient un modèle de petit monde avec un degré suivant cette distribution tout en restant navigable.
La question résultant de cette approche permet de comprendre qu'il existe un problème évident de cette modélisation de réseau : comment peut-on décomposer, dans un réseau réel, les liens qui jouent le rôle de raccourcis (liens longue-distance) des autre liens ? Ce problème est directement lié à la nature de la modélisation utilisée, c'est-à-dire la théorie des graphes.

1.5 Conclusion

Tout au long de ce chapitre, un bon nombre de modèles de réseaux complexes, basés sur la théorie des graphes, a été présenté, de manière plutôt chronologique. Au fil des recherches, de nombreuses caractéristiques et propriétés ont été étudiées, pour au final, but ultime de la modélisation et de la simulation, correspondre le plus au monde réel. La question de la correspondance réseaux réels - réseaux modélisés reste cependant sans réponse satisfaisante. La plupart des modèles présentés possèdent, il est vrai, une ou plusieurs caractéristiques appartenant aux réseaux réels. Cependant, même si le dernier modèle de Lebhar paraît pertinent, en particulier pour les réseaux informatiques, il est

[4]Sur un tore $d-dimensionnel$, $B(u,r)$ est une sorte de sphère centrée sur un noeud u et de rayon r.

évident qu'une grande part d'incertitude existe quant à une modélisation efficace et une compréhension globale des réseaux complexes. Si les études autour de la structure sont les plus nombreuses, alors peut-être une meilleure compréhension de la fonction permettrait d'éclaircir le problème posé [19], ou tout simplement en allant au-delà d'une modélisation utilisant de simples graphes, comme les hypergraphes ou les matroïdes par exemple, que la théorie présentée dans le prochain chapitre englobera. Car aussitôt que le comportement fonctionnel des réseaux complexes sera compris, la modélisation de ces réseaux n'en deviendra que plus évidente et compréhensible. Dans la suite du document, nous allons montrer qu'il est possible d'obtenir des modélisations plus fines se rapprochant d'avantage des cas du réel.

Chapitre 2

Modélisation d'un réseau par la prétopologie

Dans le présent chapitre, nous introduisons la *théorie de la Prétopologie* [7], afin de définir la notion de *réseau prétopologique* permettant de modéliser les réseaux complexes. Pour ce faire, nous rappelons les fondements de la théorie de la prétopologie, puis nous développons un argumentaire sur les structures de données choisies et les opérations afférentes avec le souci de construire des algorithmes efficaces permettant une implémentation ayant de bonnes performances, le tout pour aboutir à la simulation de réseaux du réel. On discute également de la complexité des opérations liées à ces structures de données. Pour finir, on présente une série d'exemples illustrant l'intérêt de la prétopologie dans la modélisation des réseaux complexes.

2.1 Introduction

Habituellement, on parle de modèle mathématique quand on met en oeuvre des outils et théories mathématiques permettant de simuler le comportement du modèle réel. Cependant, *il ne faut pas contraindre le réel à subir une axiomatique qui ne lui convient pas*, d'où l'idée de développer une topologie mieux adaptée aux problèmes rencontrés [7]. L'historique de cette idée débute dans les années 20 quand Fréchet visait à construire une topologie ayant moins d'axiomes [33]. Puis Appert reprit ces travaux dans les années 30, suivi par Monteiro dans les années 40 puis par Ky-Fan. Dans les années 60, Cech décrivait dans son ouvrage des structures topologiques privées de l'axiome idempotence [20]. C'est au début des années 70 que des chercheurs français (Marcel Brissaud, Gérard Duru, Jean-Paul Auray, Michel Lamure [...] pour ne citer qu'eux) débuteront leurs travaux sur la prétopologie et comment trouver une formalisation du concept de *proximité*, afin de résoudre des problématiques difficiles liées à la contrainte de l'outil mathématique utilisé, ici la topologie.
Dans le cadre de mes travaux, la prétopologie est à mon sens l'outil permettant d'aller plus loin que la théorie des graphes habituellement utilisée pour représenter les réseaux. Son cadre général, ainsi que sa capacité à représenter la dynamique d'un système en font l'outil adéquat pour la modélisation des réseaux complexes. Dans cette partie, des définitions

liées à la prétopologie sont données, ainsi qu'une définition générale d'un réseau. Nous proposerons ensuite des choix de structures de données adaptées dans le but de limiter la complexité des opérations afin de pouvoir construire des simulations exploitables, que nous appliquons sur plusieurs problématiques.

2.2 La théorie de la prétopologie

Cette section énonce les définitions de la prétopologie utilisées pour les travaux de recherche présentés dans ce mémoire, l'ouvrage de référence [7] contenant une présentation exhaustive de la prétopologie.

2.2.1 Espace prétopologique

Soit E un ensemble non vide, et soit $\mathcal{P}(E)$ l'ensemble des parties de E.

Définition 2.1 *Soit une application* $a : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$ *appelée* adhérence *et définie comme suit :* $\forall A$, $A \subseteq E$ *l'adhérence de* A, $a(A) \subseteq E$ *est telle que :*
- $a(\emptyset) = \emptyset$ (P_1)
- $A \subseteq a(A)$ (P_2)

L'adhérence est associée au processus de *dilatation*. De plus, $a(.)$ peut être appliquée à A selon une séquence : $A \subseteq a(A) \subseteq a^2(A) \subseteq ...$. Cela signifie que l'on peut suivre le processus pas à pas, ce qui n'est pas possible avec la topologie, qui conserve la propriété d'idempotence ($a(A) = a^2(A)$) [17]. Grâce à l'adhérence, on peut directement modéliser la notion de *proximité*. En terme de complexité, le coût d'une adhérence sera étudié lors de la partie concernant les structures de données utilisées en prétopologie.

Définition 2.2 *Soit une application* $i : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$ *appelée* intérieur *et définie comme suit :* $\forall A$, $A \subseteq E$ *l'intérieur de* A, $i(A) \subseteq E$ *est telle que :*
- $i(A) = [a(A^c)]^c$ (P_1)
- $i(A) \subseteq A$ (P_2)

avec A^c *le complémentaire de* A *soit* $E - A$.

L'intérieur est quant à lui associé au processus *d'érosion*. Notons que la propriété 1 de l'intérieur amenant la dualité n'est pas toujours vraie. Il est possible de définir une application intérieur indépendamment de l'adhérence.

On appelle *espace prétopologique* le triplet (E, i, a) dont les applications i et a sont définies précédemment.

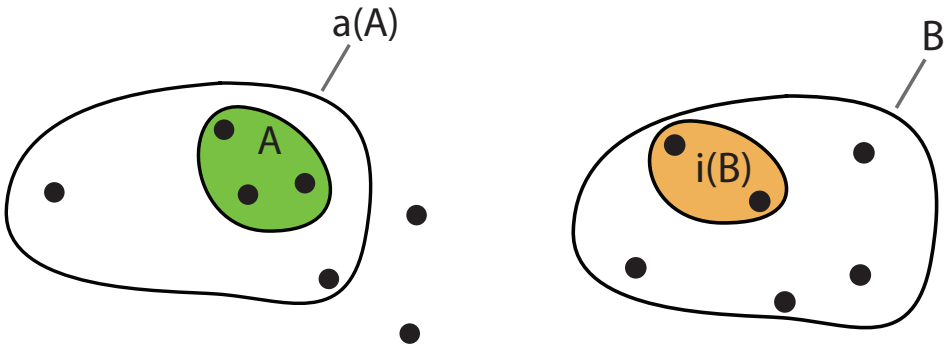


FIG. 2.1 – Illustrations d'adhérence & d'intérieur

2.2.2 Fermés et Ouverts, Fermeture et Ouverture

Le processus de dilatation généré par l'adhérence s'arrête à un instant donné et n'évolue plus. Dans ce cas, on a $a^{k+1}(A) = a^k(A)$. On nomme A comme étant un sous ensemble *fermé*.
De la même manière, l'évolution de l'intérieur va cesser, ce qui nous donne $i^{k+1}(A) = i^k(A)$. Cette fois, on nomme A comme étant un sous ensemble *ouvert*.
Respectivement, on utilise les notations $F(A)$ pour la fermeture de A et $O(A)$ pour l'ouverture de A. La complexité d'un fermé, si on prend l'adhérence comme opération de base, se fait au pire en $O(n \times$ coût d'une adhérence$)$.

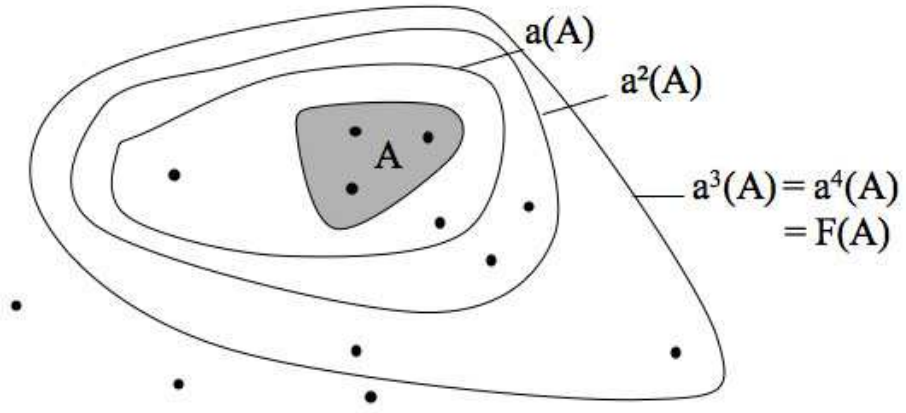


FIG. 2.2 – Illustration d'adhérences successives menant au fermé

2.2.3 Fermés élémentaires et fermés minimaux

On appelera fermé élémentaire et on notera F_x, la fermeture d'un singleton $\{x\}$ de E. On note $\mathcal{F}_e(E,a)$ ou $\mathcal{F}_e$, l'ensemble des fermés élémentaires de E :

$$\mathcal{F}e(E,a) = \{F_x, x \in E\}$$

L'algorithme (Alg. 2.2.3) permet le calcul de la famille des fermés élémentaires , et si on prend comme opération de base *l'adhérence*, la complexité de cet algorithme qui comprend deux boucles imbriquées est, dans le pire des cas, en $O(n^2 \times$ coût d'une adhérence$)$.

Fermés élémentaires

Méthode : $\mathcal{F}_e()$
Variables :
E : espace prétopologique
$\mathcal{F}_e$: famille d'ensembles
F : ensemble
Début
  $\mathcal{F}_e \leftarrow \emptyset$
  Pour tout $\{x\} \in E$ **Faire**
    $F_x \leftarrow a(\{x\})$
    Tant que $a(F_x) \neq F_x$ **Faire**
      $F_x \leftarrow a(F_x)$
    Fin Tant que
    $\mathcal{F}_e \leftarrow \mathcal{F}_e \cup \{F_x\}$
  Fin Pour
  Retourner $\mathcal{F}_e$
Fin

On appelle fermé minimal de E, tout élément de $F(E,a)$, minimal au sens de l'inclusion. L'ensemble des fermés minimaux est noté : $\mathcal{F}_m(E,a)$ ou $\mathcal{F}_m$.
Un résultat important est que tout fermé minimal est obligatoirement élément de $\mathcal{F}_e$, c'est à dire un fermé élémentaire. Déterminer les fermés minimaux revient donc à explorer les éléments de $\mathcal{F}_e$ et en extraire les éléments minimaux par la relation d'inclusion. L'algorithme [15] permettant le calcul de la famille des fermés minimaux (Alg. 2.2.3), comprend également deux boucles imbriquées, donc la complexité dans le pire des cas est en $O(n^2)$ (l'adhérence n'intervient pas dans ce cas).

2.2.4 Espace prétopologique de type $\mathcal{V}$

Un espace prétopologique général comme défini ultérieurement ne présente que peu d'intérêt en l'état, car il est difficile d'en faire une analyse. Il faut donc amener une nouvelle propriété pour rendre cet espace prétopologique plus «intéressant», d'où la définition d'un nouvel espace prétopologique : le type $\mathcal{V}$.

Fermés minimaux

Méthode : $\mathcal{F}_m(\mathcal{F})$
Variables :
E : espace prétopologique
$\mathcal{F}, \mathcal{F}T, \mathcal{F}_m$: famille d'ensembles
F, G : ensembles
$minimal$: booléen
Début
  $\mathcal{F}_m \leftarrow \emptyset$
  Tant que $\mathcal{F} \neq \emptyset$ **Faire**
    Prendre F tel que $F \in \mathcal{F}$
    $\mathcal{F} \leftarrow \mathcal{F} - \{F\}$
    $minimal \leftarrow vrai$
    $\mathcal{F}T \leftarrow \mathcal{F}$
    Tant que $\mathcal{F}T \neq \emptyset$ et $minimal = vrai$ **Faire**
      Prendre G tel que $G \in \mathcal{F}T$
      Si $G \subset F$ **Alors**
        $minimal \leftarrow faux$
      Sinon
        Si $F \subset G$ **Alors**
          $\mathcal{F} \leftarrow \mathcal{F} - \{G\}$
        Fin Si
        $\mathcal{F}T \leftarrow \mathcal{F}T - \{G\}$
      Fin Si
    Fin Tant que
    Si $minimal = vrai$ **Alors**
      $\mathcal{F}_m \leftarrow \mathcal{F}_m \cup \{F\}$
    Fin Si
  Fin Tant que
  Retourner $\mathcal{F}_m$
Fin

Définition 2.3 *Un espace prétopologique de type* $\mathcal{V}$ (E, i, a) *est défini comme suit :*

$$\forall A, B, A \subseteq E, B \subseteq E \text{ et } A \subset B \text{ avec } a(A) \subseteq a(B)$$

$$\forall A, B, A \subseteq E, B \subseteq E \text{ et } A \subset B \text{ avec } i(A) \subseteq i(B)$$

D'autres types d'espaces existent :

Espace prétopologique de type $\mathcal{V}_d$

Définition 2.4 *Un espace prétopologique de type* $\mathcal{V}_d$ (E, i, a) *est défini comme suit :*

$$\forall A, B, A \subseteq E, B \subseteq E \text{ et } A \subset B \text{ avec } a(A \cup B) = a(A) \cup a(B)$$

$$\forall A, B, A \subseteq E, B \subseteq E \text{ et } A \subset B \text{ avec } i(A \cap B) = i(A) \cap i(B)$$

Tout espace de type $\mathcal{V}_d$ est de type $\mathcal{V}$.

Espace prétopologique de type $\mathcal{V}_s$

Définition 2.5 *Un espace prétopologique de type* $\mathcal{V}_s$ (E, i, a) *est défini comme suit :*

$$\forall A, A \subseteq E, avec\ a(A) = \bigcup_{\forall\ x \in A} a(\{x\})$$

Un espace de type $\mathcal{V}_s$ est clairement de type $\mathcal{V}_d$. Les applications a et i ne sont pas forcément idempotentes. On ne doit pas confondre une prétopologie de type $\mathcal{V}_s$ et une topologie. Les types d'espaces les plus utilisés dans nos études sont les types $\mathcal{V}$ et $\mathcal{V}_s$.

2.2.5 Préfiltre et base de voisinage

Comme écrit plus haut, le concept de proximité en prétopologie est fondamental, c'est pour cela qu'il nous faut définir ce que l'on appelle «être proche » d'un élément, autrement dit être dans le *voisinage* de cet élément. Pour cela il faut définir ce voisinage par un préfiltre et une base de voisinage.

Préfiltre

Une notion essentielle de la topologie est la notion de filtres, qui conduit à celle de voisinage. En prétopologie, non seulement nous conservons cette notion, mais nous définissons en outre celle de préfiltre.

Définition 2.6 *Une partie $\mathcal{F}$ de $\mathcal{P}(E)$ est un préfiltre sur E si elle vérifie la propriété de stabilité par passage à tout sur-ensemble :*

$$\forall\ F \in \mathcal{F}, \forall\ H \in \mathcal{P}(E), F \subset H \Rightarrow H \in \mathcal{F}$$

Soit (E,i,a) un espace prétopologique de type $\mathcal{V}$. Pour tout x de E, on définit le préfiltre $V(x)$ de parties de E par :

$$\mathcal{V}(x) = \{V \subseteq E / x \in i(V)\}$$

Les éléments de $\mathcal{V}(x)$ sont appelés *voisinage* de x. Pour comprendre la relation avec les applications adhérence et intérieur, on les définit à partir d'un préfiltre $\mathcal{V}(x)$:

$$a(A) = \{x \in E / \forall\ V, V \in \mathcal{V}(x), V \cap A \neq \emptyset\}$$
$$i(A) = \{x \in E / \forall\ V, V \in \mathcal{V}(x), V \subseteq A\}$$

Base de voisinage

Définition 2.7 *Soit $\mathcal{B}(x)$ la base de $\mathcal{V}(x)$ définie comme suit :*

$$\forall\ V, V \in \mathcal{V}(x), \exists B, B \in \mathcal{B}(x), B \subseteq V$$

On définit ainsi l'adhérence comme suit :

$$a(A) = \{x \in E / \forall\ B, B \in \mathcal{B}(x), B \cap A \neq \emptyset\}$$

Il existe plusieurs manières de définir la base de voisinage selon la problématique étudiée.
Par exemple, dans [52], on définit une population où les éléments sont reliés par une ou plusieurs relations binaires R_i réflexives, soit pour tout i, la partie $\mathcal{B}_i(x)$ étant construite pour chaque x appartenant à E de la manière suivante : $\mathcal{B}_i(x) = \{y \in E / x\ R_i\ y\}$.
Un autre cas tiré de [48] où E est doté d'une métrique définie par une distance et où la base de voisinage pour chaque x appartenant à E et $r \in \mathbb{R}_+$ est définie comme suit : $\mathcal{B}(x, r) = \{y \in E, d(x, y) \leq r\}$

2.3 Modélisation d'un réseau par la prétopologie

Il y a plusieurs raisons pour utiliser la théorie de la prétopologie dans la modélisation des réseaux : les modèles utilisant la théorie des graphes ont certaines limites. Premièrement, on ne peut pas dissocier les liens : orientés ou non, ils sont tous de nature identique. Si l'on veut utiliser n différentes relations entre les noeuds, il faut construire n graphes différents ce qui semble peu pratique [54]. Deuxièmement, tous les modèles utilisant la théorie des graphes présentés jusqu'alors dans les études récentes ont au moins une propriété qui ne correspond pas aux réseaux du réel [56]. Et troisièmement, les relations se font par paires de noeuds, on ne peut donc pas avoir de relations entre un groupe de noeuds et un noeud

par exemple. Les hypergraphes répondent à ce problème, mais sont un cas particulier d'espace prétopologique [7].
Où les autres modèles ont échoué, la théorie de la prétopologie peut apporter une réponse.

2.3.1 Définition d'un réseau en prétopologie

Un réseau peut être défini comme une famille de relations binaires ou valuées définies sur une population donnée [24]. La dynamique du réseau est basée sur des opérations telle que l'arrivée de nouveaux éléments, l'éviction d'éléments existants, la formation de groupe ou la séparation en sous-groupes (voir Fig. 2.3). Ces phénomènes sont souvent observables dans les réseaux sociaux sous forme de communautés [5] mais également dans le cas des réseaux de manière plus générale.

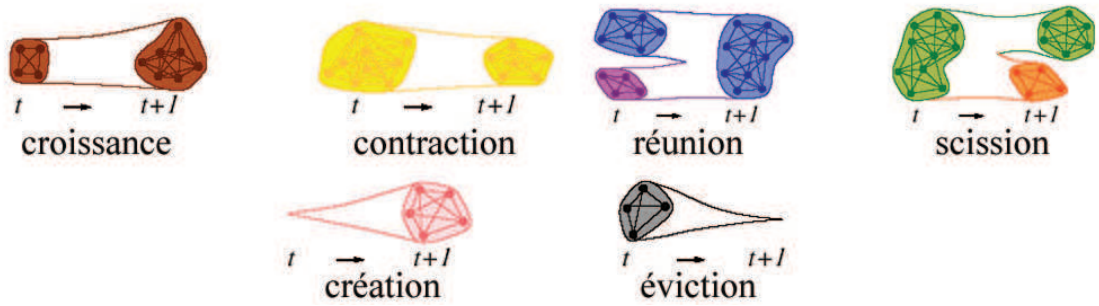


FIG. 2.3 – Opérations représentant la dynamique d'un réseau (t : temps)

Dans le cadre de la prétopologie, un réseau est une famille de prétopologies sur un ensemble donné [23], d'où la définition suivante :

Définition 2.8 *Soit X un ensemble :*
soit I une famille dénombrable d'indices ;
soit $\{a_i, i \in I\}$ une famille de prétopologies sur X_i ;
la famille d'espaces prétopologiques $\{(X, a_i), i \in I\}$ constitue un réseau sur X.

2.3.2 Structures de données adaptée à la prétopologie

Comme précisé précédemment, les deux types d'espaces prétopologiques qui nous intéressent sont le type $\mathcal{V}$ et le type $\mathcal{V}_s$. Ces deux espaces ne sont pas tout à fait basés sur le même type de structure de données. Mais quelle structure de données serait assez efficace pour permettre des opérations rapides sur de tels espaces ? A notre sens, la structure de données la plus efficace est *la table de hachage*.

La table de hachage

Comment faire des opérations de recherche et d'insertion dans une table en temps constant ? Les mémoires associatives, étudiées en intelligence artificielle, auraient cette

propriété : la position occupée par un objet est déterminée seulement par cet objet. Il suffit de calculer la position en mémoire de cet objet, donc de disposer d'une fonction qui associe à un objet x sa position $h(x)$. On veut que le temps de calcul de $h(x)$ soit indépendant de x et aussi petit que possible. On dit que $h(x)$ est la valeur de hachage associée à x. C'est l'idée des tables de hachage, dont l'organisation s'oppose radicalement à celle des tables ordonnées, dans lesquelles la position d'un objet dépend du nombre d'objets plus petits qui ont déjà été insérés dans la table. Son seul inconvénient est que l'ordre dans lequel les données sont stockées n'est pas maîtrisable. Si l'on doit énumérer les données par ordre croissant, il faut recourir à une autre implémentation des tables, à l'aide d'arbres, et renoncer au coût constant des opérations pour un coût logarithmique.

Si l'on devait insérer dans une table les éléments de l'ensemble $[0, N-1]$, il suffirait d'utiliser un tableau t de taille N et de ranger l'objet i dans $t[i]$, solution triviale, avec $h(i) = i$. Cela n'est déjà plus aussi simple si au lieu de l'intervalle $[0, N-1]$, on doit traiter un ensemble quelconque E de N entiers : comment faire pour construire « effectivement » une fonction injective[1] de E dans l'ensemble des indices du même tableau $[0, N-1]$? L'existence d'une fonction injective est sans doute rassurante mais pas suffisante. En effet, les fonctions injectives sont rares : il y a nm fonctions d'un ensemble de cardinal m dans un ensemble de cardinal n, parmi lesquelles il y a $\frac{n!}{(n-m)!}$ fonctions injectives, si $m \leq n$. Une estimation asymptotique du rapport $\frac{n!}{(n-m)!n^m}$, par la formule de Stirling [2] , quand $n \to \infty$, est $e^{-k^2/2n}$.
Le problème se complique encore, car on veut insérer dans une table des objets d'un ensemble E de cardinal plus grand, voire beaucoup plus grand que N. Par exemple, la table de hachage qu'un compilateur construit pour les noms figurant dans un programme donné doit contenir tout au plus quelques centaines de noms, mais ces noms appartiennent au minimum à un ensemble de cardinal 27×37^5, soit près de deux milliards, pour le langage C, dont la norme impose à tout compilateur d'accepter des identificateurs d'au moins 6 caractères, formés de chiffres, de lettres sans distinction de casse et de '_', et commençant par une lettre ou par '_'. Le problème est de construire une fonction, qui bien que non injective, a de bonnes propriétés de dispersion.
Voici un exemple simple de fonction de hachage sur des chaînes de longueur l à valeurs dans l'intervalle $[0, N-1]$:

$$h(x) = \sum_{i=0}^{l-1} x[i] B^{l-1-i} \bmod N$$

où B est une puissance de 2 et N est un nombre premier (pour éviter des collisions).

Comme la fonction de hachage h n'est pas injective, il faut savoir traiter les collisions, c'est-à-dire le cas de deux clés $x \neq y$ ayant la même valeur de hachage $h(x) = h(y)$. Il existe deux sortes de techniques de résolution : le hachage *ouvert* et le hachage *par*

[1] f est injective si et seulement si : $f(x_1) = f(x_2) \Rightarrow x_1 = x_2$

[2] cette formule du nom du mathématicien James Stirling, donne un équivalent de la factorielle au voisinage de l'infini réel (quand n tend vers l'infini) : $\lim_{n \to +\infty} \frac{n!}{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n} = 1$

chaînage.

Hachage par adressage ouvert

La table de hachage est implémentée par un tableau t dont les cases vont contenir les associations. Initialement, tous les éléments du tableau ont la valeur *null*. Puis des opérations d'insertion sont réalisées ; ni la clé ni l'information ne peut être nulle. Si une association de clé x doit être insérée, et que $t[h(x)]$ est vide, c'est-à-dire contient la valeur nulle, alors l'insertion se fait à cette place ; si par contre $t[h(x)]$ est déjà occupé, et que le contenu a une clé différente de x, alors on calcule des valeurs de hachage supplétives $h_1(x), h_2(x), ...$ jusqu'à ce que l'on trouve $t[h_i(x)]$ vide ou contenant une association de clé x. De même, pour chercher une association de clé x, on teste la clé de l'objet en $t[h(x)]$, et éventuellement en $t[h_1(x)], t[h_2(x)]$, etc, jusqu'à ce que la clé de l'objet qui s'y trouve soit égale à x, ou bien que la valeur soit nulle.
Dans le cas où la table permet aussi des suppressions, il faut remplacer un objet supprimé par un objet spécial supprimé, distinct de la valeur nulle ; en insertion, on utilisera la première case vide ou supprimée, tandis qu'en recherche, on ne s'arrêtera qu'à la première case vide. On a choisi de représenter un objet supprimé par l'association d'une clé égale à la chaîne vide et d'une information nulle ; par conséquent, on doit interdire la chaîne vide en tant que clé pour l'insertion.

Pour générer ces nouvelles valeurs de hachage, la méthode la plus simple est le sondage linéaire, qui choisit $h_i(x) = (h(x) + i) \bmod N$, ce qui revient à essayer les cases suivant $t[h(x)]$. Il y a d'autres méthodes (sondage quadratique, double hachage, hachage uniforme) qui ont de meilleures capacités de dispersion.

Il est clair que la complexité dans le pire des cas est de N opérations, N étant la taille de la table. On définit le facteur de charge d'une table de hachage de taille N contenant n objets comme la fraction $\alpha = n/N$: on a toujours $0 \leq \alpha \leq 1$. Pour donner une estimation asymptotique de la complexité des opérations, on fait tendre n et N vers l'infini, à α constant. On montre que, sous une hypothèse d'uniformité, le nombre moyen d'accès nécessaires pour une recherche infructueuse est d'au plus $1/(1-\alpha)$ et pour une recherche fructueuse de $\frac{1}{\alpha} \log \frac{1}{1-\alpha} + \frac{1}{\alpha}$. Il s'agit bien d'un algorithme en $\Theta(1)$ [44].

Hachage par chaînage

La méthode de hachage par adressage ouvert avec sondage linéaire a l'inconvénient d'effectuer des comparaisons avec plusieurs cases successives de la table, qui n'ont pourtant pas la même valeur de hachage que l'objet recherché. Pour éviter ces comparaisons inutiles, on va chaîner les objets ayant la même valeur de hachage. La table de hachage est également implémentée par un tableau, l'élément du tableau d'indice i étant une référence à un objet rassemblant des associations dont les clés ont i pour valeur de hachage.

Le facteur de charge $\alpha = n/N$ est la longueur moyenne des listes chaînées. Sous une hypothèse d'uniformité de la fonction de hachage, on montre que le nombre moyen d'accès

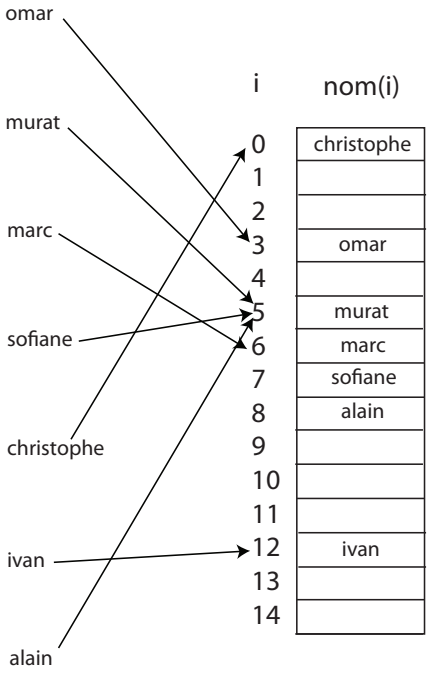


FIG. 2.4 – Hachage par adressage ouvert

nécessaires pour une recherche, négative ou positive, est d'au plus $1 + \alpha$.

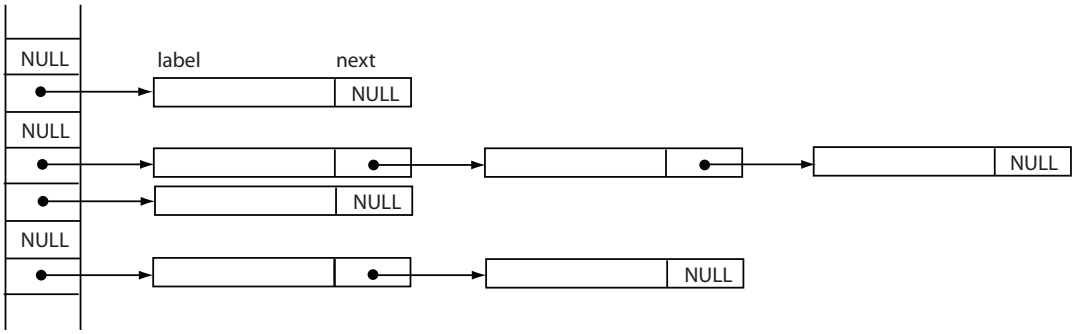


FIG. 2.5 – Hachage par chaînage

Structure de données : type $\mathcal{V}$

Un espace prétopologique de type $\mathcal{V}$ est un espace prétopologique général auquel on rajoute une propriété décrite dans la définition 2.3. Si cet espace paraît simple à représenter, c'est néanmoins cette simplicité qui fait tout son intérêt, permettant ainsi de représenter des relations plus complexes et plus fines que les autres espaces. Dans $\mathcal{V}$, pour chaque partie $\mathcal{B}(E)$ de $\mathcal{P}(E)$, il existe une adhérence associée. L'idée est donc de considérer une

table de hachage avec comme clé $\mathcal{B}(E)$, et comme valeur $a(\mathcal{B}(E))$, en écartant les cas où $a(\mathcal{B}(E)) = \mathcal{B}(E)$. L'opération adhérence se fait de cette manière de façon extrêmement rapide. Nous sommes bien conscients que dans ce cas, nous sacrifions la complexité spatiale à la complexité temporelle, cette dernière étant celle qui nous intéresse le plus. Ici, un hachage par adressage ouvert peut être suffisant. On peut voir un exemple d'adaptation de table d'adhérences en table de hachage ci-dessous :

Exemple 2.1 *Soit $E = \{x, y, z\}$ et a l'application adhérence décrite par le tableau suivant :*

$\mathcal{P}(x)$	$\boldsymbol{a}$
$\emptyset$	$\emptyset$
$\{x\}$	$\{x, z\}$
$\{y\}$	$\{y\}$
$\{z\}$	$\{z\}$
$\{x, y\}$	$\{x, y\}$
$\{x, z\}$	$\{x, z\}$
$\{y, z\}$	E
E	E

donne la table de hachage suivante :

Clé	***Valeur***
$\{x\}$	$\{x, z\}$
$\{y, z\}$	$\{x, y, z\}$

Adhérence (table de correspondance)

Méthode : a(A)
Variables :
E : espace prétopologique
A : ensemble passé en paramètre
tab : table de correspondance
Début
  Retourner résultat de *tab* avec comme clé A
Fin

Sachant que la plupart des résultats en prétopologie proviennent d'algorithmes basés sur l'adhérence, on calculera la complexité en temps d'un algorithme en prenant l'adhérence comme coût de base d'une opération, dans ce cas $\theta(1)$ (Alg. 2.3.2). On parlera ainsi de complexité *amortie*, car le pire des cas n'est pas impossible à envisager mais en gardant un facteur de charge performant, il est possible de l'éviter dans la majorité des configurations.

Structure de données : type $\mathcal{V}_s$

Les espaces de type $\mathcal{V}_s$ possèdent des axiomes plus forts que les espace de type $\mathcal{V}$ (voir Déf. 2.5). Leur définition se rapproche fortement des graphes, même si ces derniers ne leur sont pas totalement équivalents. Dans $\mathcal{V}_s$, comme l'union des adhérences de singletons revient à l'adhérence de l'union de ces même singletons, on ne prend donc en compte que les adhérences de singletons, puisqu'il suffit de faire des unions pour trouver le résultat de

l'adhérence d'un ensemble composé de plusieurs éléments. On retombe donc sur un cas connu de structure de données de graphe : *les listes d'adjacence* (voir chapitre 1.2.2). Nous utiliserons ce modèle avec des tables de hachage selon cette idée : les voisins d'un singleton seront stockés dans une table de hachage elle-même stockée dans la case valeur d'une table de hachage principale où la clé serait le singleton utilisé. C'est le principe du hachage par chaînage, sauf qu'à la place d'utiliser des listes dans une table de hachage, on utilise des tables de hachage dans une table de hachage. En voici un exemple :

Exemple 2.2 *Soit* $E = \{a, b, c, d, e\}$ *et a l'application adhérence décrite comme suit :*

$$\forall A \subseteq \mathcal{P}(E), a(A) = \{x \in E, R(x) \cap A \neq \emptyset\}$$

où $R(x) = \{y \in E, xRy\}$ *et* R *représentant une relation binaire entre deux singletons. Soit l'ensemble des relations dans* E *défini selon la figure 2.6, on obtient une table de hachage dont l'illustration est visible également Fig. 2.6.*

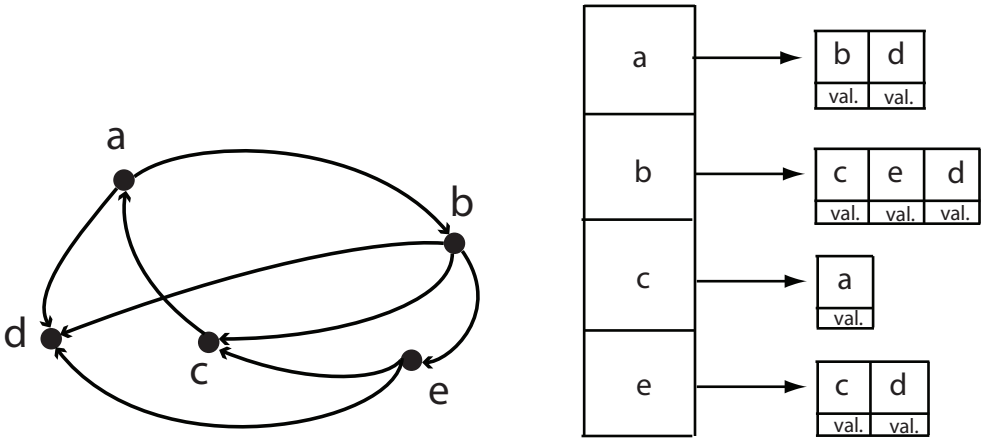


FIG. 2.6 – Exemple de table de hachage avec $\mathcal{V}_s$

Il faut noter que dans ce cas, comme il s'agit de relations binaires, qu'aucune information supplémentaire n'est renseignée. Les données contenues dans les cases « valeur » des tables de hachage représentant les voisins sont ici égales à NULL. *Si en revanche, nous nous placions dans un espace avec des relations valuées, il suffirait d'insérer la valeur du poids de la relation, que ce soit un réel, une chaîne de caractère ou tout autre type de donnée.*

Le coût d'une adhérence ici change : en effet, dans l'algorithme 2.3.2, on remarque que deux boucles s'imbriquent, donc la complexité de cet algorithme est en $O(n^2)$ dans le pire des cas, car les opérations de remplissage de liste et de test d'appartenance d'un singleton à un ensemble se font en $O(1)$ grâce à la structure de la table de hachage. Dans le cas où le nombre de relations est très grand, on peut combiner la structure de donnée utilisée

Adhérence (liste d'adjacence)

Méthode : a(A)
Variables :
E : espace prétopologique
A : ensemble passé en paramètre
B : ensemble résultat
liste_v : liste de singletons
Début
  $B \leftarrow \emptyset$
  $A^c \leftarrow E - A$
  Pour tout $\{y\} \in A^c$ **Faire**
    Remplir *liste_v* avec les voisins de $\{y\}$
    Pour tout $\{v\}$ de *liste_v* **Faire**
      Si $\{v\} \in A$ **Alors**
        Ajouter $\{v\}$ à B
      Fin Si
    Fin Pour
  Fin Pour
  Retourner B
Fin

pour $\mathcal{V}$ en y insérant au fur et à mesure les unions d'adhérence résultant de précédents calculs. On peut aussi générer la table de correspondance une fois pour toutes puis ne travailler qu'avec cette dernière. On sacrifie également dans ce cas la place mémoire à la vitesse d'exécution. Cela n'est possible que si l'adhérence reste statique.

2.4 Exemple de modélisation d'une problématique

Pour illustrer nos propos, nous allons appliquer nos structures de données à deux problématiques : un travail sur les flux de migrations inter-villes modélisée avec la prétopologie par C. Largeron et S. Bonnevay [47] et une approche de modélisation inter-auteurs d'articles scientifiques due à Dalud-Vincent.

2.4.1 Modélisation de flux de migrations inter-villes

Dans cet article [47], les auteurs proposent une méthode de structuration d'un ensemble E fini, dont les éléments sont reliés par une relation quelconque, afin de faire ressortir des groupes homogènes. Nous invitons le lecteur à prendre connaissance de cet article pour plus de détails. Pour notre part, nous rappellerons ici que l'espace utilisé pour la modélisation est un type $\mathcal{V}$, que cet espace est métrique défini par une distance d. Dans leur exemple, « *à partir des flux migratoires entre les agglomérations, décrits par le nombre d'actifs résidant dans une commune et allant travailler dans une autre sur le graphe donné*

en annexe, l'objectif est de mettre en évidence les liens de dépendance en matière de recrutement de la main d'oeuvre qui existent entre ces communes en considérant uniquement les flux jugés significatifs c'est-à-dire supérieurs à un seuil r préalablement fixé [...].
Ainsi, r étant un nombre réel strictement positif, pour tout x de E, on note B(x,r) la boule ouverte de centre x et de rayon r définie par :

- $B(x,r) = \{y \in E, f_{xy} > r\}$ *où* f_{xy} *représente le nombre d'actifs résidant en x et travaillant en y.*

- *et,* $\forall A \subseteq \mathcal{P}(E), a(A) = \{x \in E, B(x,r) \cap A \neq \emptyset\}$

»

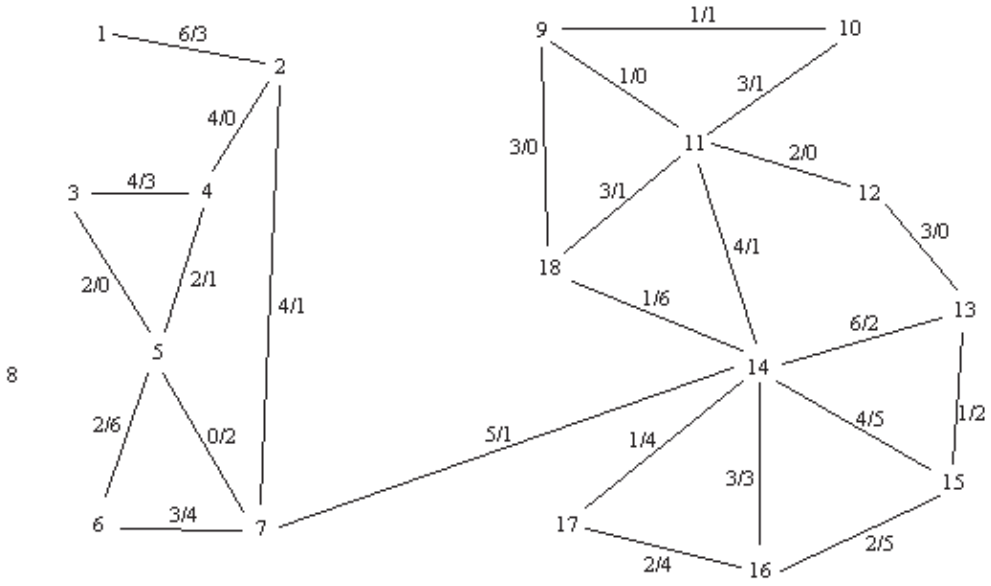


FIG. 2.7 – Schéma des flux de migrations inter-villes

La figure 2.7 illustre un exemple de flux migratoires. La valeur du seuil utilisé pour l'exemple est $r = 3$, d'où le tableau suivant (Fig. 2.8) où les * signifient que le fermé est minimal.

On se rend bien compte que si l'on veut implémenter un exemple tel que celui-ci, il faut avoir recours à des structures de données et des méthodes efficaces. L'idée serait donc de développer un programme évolutif et paramétrable selon la problématique pour être capable de mettre en oeuvre des simulations basées sur la prétopologie : c'est pourquoi nos travaux nous ont menés à l'écriture d'une librairie JAVA générique permettant la simulation de telle problématique. Son fonctionnement sera détaillé dans le chapitre qui lui est consacré.
Pour revenir à notre exemple, on peut voir le résultat de la simulation informatique sur la figure 2.9. Les flux n'ont pas été représentés sur l'image. Les fermés élémentaires sont en transparent et les fermés minimaux en rouge.

Ville x	B(x,3)	F_x : fermeture de x	Ville x	B(x,3)	F_x : fermeture de x
1	{1, 2}	{1} *	10	{10}	{10}*
2	{2, 4, 7}	{1, 2}	11	{11, 14}	{11}*
3	{3, 4}	{3} *	12	{12}	{12}*
4	{4}	{1, 2, 3, 4}	13	{13, 14}	{13}*
5	{5}	{1, 2, 5, 6, 7}	14	{14, 15}	{1, 2, 7, 11, 13, 14, 15, 16, 17, 18}
6	{5, 6}	{1, 2, 6, 7}	15	{14,15}	{1, 2, 7, 11, 13, 14, 15, 16, 17, 18}
7	{6, 7, 14}	{1, 2, 7}	16	{15, 16}	{16, 17}
8	{8}	{8}*	17	{14, 16, 17}	{17}*
9	{9}	{9}*	18	{14, 18}	{18}*

FIG. 2.8 – Résultats des adhérences et fermetures sur E

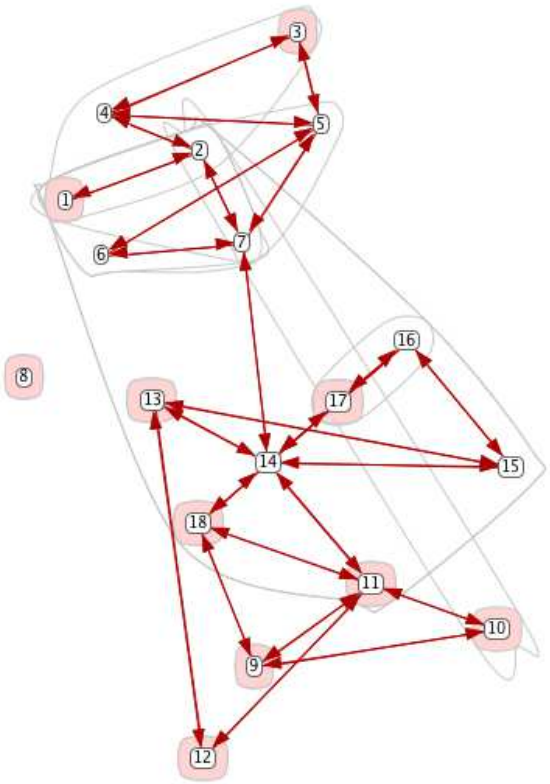


FIG. 2.9 – Illustration de la simulation basée sur l'article Largeron-Bonnevay

2.4.2 Modélisation de réseau d'auteurs

Cet exemple est issu d'une recherche initiée par la « mesure en éducation » dans un réseau social par Monique Dalud-Vincent de l'université Lyon I. L'objectif de cette étude est de montrer que la science se fait en réseau, que celle en particulier qui porte sur la mesure en éducation en est un exemple et de donner et analyser la configuration de ce réseau ainsi que l'analyse de son évolution dans le temps. Au niveau des données, on veut repérer les publications communes (livres, articles, rapports) en faisant une recherche bibliographique grâce à Google Scholar : on part d'auteurs qui font des études dans le domaine et à partir de ces auteurs on pratique la méthode « boule de neige ». On peut aussi partir aussi de mots-clés. Ainsi, une base de données est créée d'environ 2500 références et 1400 auteurs. Grâce aux méthodes développées par Dalud-Vincent [23], on fait une recherche des composantes et une décomposition de celles-ci en centres et périphéries.
Jusqu'à présent, les modèles liés à cette problématique sont basés sur les graphes. Notre objectif est de créer un modèle de cette problématique basé sur la théorie de la prétopologie. Pour une référence contenant plus de deux auteurs, on fait comme si les auteurs avaient écrit deux à deux ensemble. On construit donc des liens de type individu-groupe tout en tenant compte de la réalité des données mais aussi d'une certaine conception sociologique. Si X est la population composée de l'ensemble des auteurs, et si A est une partie de X, on peut construire une prétopologie de type $\mathcal{V}$ en posant :

$$a(A) = \{x \in X | \text{x a publié avec A}\} \cup A$$

tel que :
- $a(\emptyset) = \emptyset$
- $A \subset B \Rightarrow a(A) \subset a(B)$

Dans cet espace, notre rôle concerne principalement la recherche des composantes fortement connexes, c'est-à-dire regrouper les singletons dont les fermés sont identiques.

Nous allons expliquer par un exemple simple pourquoi on peut gagner en finesse d'expression en utilisant la prétopologie par rapport à une représentation de type graphe.

Exemple 2.3 *Soit X composé de quatre auteurs tel que $X = \{a, b, c, d\}$. L'adhérence est construite comme précédemment. Nous avons une liste d'articles dont les auteurs sont a, b, c, etd comme suit :*
- *Article 1 écrit par a et b*
- *Article 2 écrit par a et c*
- *Article 3 écrit par a, b et c*
- *Article 4 écrit par c et d*

On obtient donc le graphe de la figure 2.10.

On lie les auteurs par des arêtes, les relations inter-auteurs étant symétrique (si a a écrit avec b, b a forcément écrit avec a). Le problème dans cette représentation est le manque de finesse. On voit bien que d est en relation avec c puisqu'ils ont écrit ensemble. Cependant, on peut de ce fait trouver un chemin entre d et n'importe quel autre auteur, alors que ceux-ci n'ont jamais collaboré ensemble (a et b n'ont pas écrit d'article avec d).

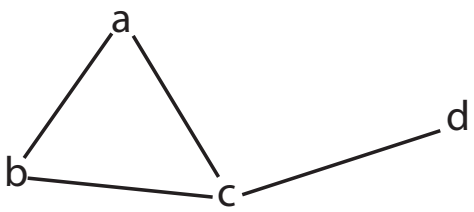


FIG. 2.10 – Réseau d'auteurs sous forme de graphe

A	a(A)	F(A)
$\{a\}$	$\{a, b, c\}$	$\{a, b, c\}$
$\{b\}$	$\{a, b, c\}$	$\{a, b, c\}$
$\{c\}$	$\{a, b, c, d\}$	$\{a, b, c, d\}$
$\{d\}$	$\{c, d\}$	$\{c, d\}$
$\{a, b\}$	$\{a, b, c\}$	$\{a, b, c\}$
$\{b, c\}$	$\{a, b, c\}$	$\{a, b, c\}$
$\{a, c\}$	$\{a, b, c\}$	$\{a, b, c\}$

TAB. 2.1 – Réseau d'auteurs sous forme d'espace prétopologique

Si on pense prétopologie, le fermé de d est X. On ne peut donc pas en dire grand chose.

Si on passe à une modélisation prétopologique, on obtient la table de correspondance 2.3.

On représente les adhérences en se basant sur les auteurs qui ont écrit ensemble et telle que $|a(A)| > |A|$, puis on calcule les fermés correspondants. Le réseau étant très petit, une seule adhérence est possible.

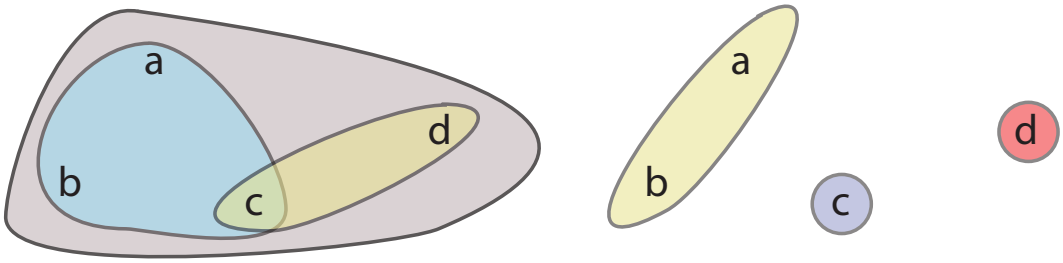


FIG. 2.11 – Fermés (gauche) et Composantes fortement connexes (droite)

On peut noter ici la différence entre une modélisation avec un espace prétopologique de type $\mathcal{V}$ et un graphe. Avec un graphe, on reste bloqué avec les fermés qui sont tous égaux et le fait qu'il n'y ait qu'une seule composante fortement connexe correspondant au graphe lui-même ne nous donne aucune information que nous n'ayons déjà. Avec un espace prétopologique, on trouve trois fermés différents, ainsi que trois composantes fortement

connexes, qui à la vue des résultats, paraît complètement sensé.

2.5 Conclusion

Ce chapitre expose comment la modélisation des réseaux peut être différente de celle que l'on rencontre habituellement. La théorie de la préopologie permet de travailler avec des outils mathématiques plus généraux, et ainsi de pouvoir créer des simulations plus réalistes, sinon plus fines. Jusqu'à présent, le plus gros inconvénient de ce type de modélisation était sa mise en oeuvre. Ce problème est maintenant en grande partie résolu grâce à nos travaux par la création d'une librairie basée sur les concepts prétopologiques permettant ainsi de construire des simulations de tout type. Aussi bonne toute théorie soit-elle, aussi faut-il les outils adéquates pour la mettre en application. Cela est ce à quoi ce chapitre apporte une réponse. Les perspectives de travail se situent dans la modélisation de réseaux complexes de grande taille, dans l'étude de l'évolution étape par étape de ces réseaux, et dans l'analyse de bases de données conséquentes comme les communautés virtuelles.

Chapitre 3

Classification prétopologique

Le problème de l'analyse d'un réseau peut faire appel au problème de la classification des éléments de celui-ci. Ce chapitre explicite quelles techniques connues sont utilisées habituellement pour classifier des entités et comment la théorie de la prétopologie peut permettre la classification d'ensembles d'éléments. Nous présentons les algorithmes de classification prétopologiques récemment développés [52] que nous spécifions afin de permettre une implémentation avec l'outil PretopoLib (voir chapitre 5), ainsi que l'introduction d'un nouvel algorithme original. Nous proposons en effet un algorithme de classification exploitant la notion de distance basée sur la complexité de Kolmogorov. Nous illustrerons celui-ci par une application permettant de saisir le degré de finesse que l'on peut obtenir avec une telle approche.

3.1 Introduction

Le problème de la classification peut être posé sous forme d'une interrogation : étant donné un ensemble d'observations dont on connaît une ou plusieurs caractéristiques, comment peut-on les regrouper en un certain nombre de groupes de manière à ce que les groupes obtenus soient constitués d'observations semblables et que les groupes soient les plus différents possible entre eux ?
Pour obtenir un bon partitionnement, il convient de :

- minimiser l'inertie intra-classe [**?**] pour obtenir des grappes (cluster en anglais) les plus homogènes possibles.
- maximiser l'inertie inter-classe afin d'obtenir des sous-ensembles bien différenciés.

Il existe de multiples méthodes de partitionnement des données, comme le regroupement hiérarchique, ou l'algorithme des k-moyennes . Nous allons nous pencher sur quelques méthodes existantes, puis nous nous focaliserons sur les récentes recherches de classification prétopologiques développées par Thanh Van Le [53] [52] que nous avons adaptées sous forme de pseudo-code, ce qui n'avait pas été fait jusqu'alors.
Dans la suite du chapitre, nous montrerons comment il est possible d'utiliser la complexité de Kolmogorov dans un modèle prétopologique afin d'introduire un indice de similarité.

3.2 Classification de données par partitionnement

La classification d'objets par partitionnement en différents groupes, ou si l'on se place dans un contexte mathématique, en sous-ensembles, dépend des caractéristiques que ces objets pourraient avoir en commun, et principalement en effectuant une mesure de la distance qui les sépare. Le partitionnement de données est une technique connue de l'analyse de données en statistique que l'on utilise dans divers domaines que ce soit en fouille de données ou en reconnaissance d'images par exemple. Habituellement, on scinde les méthodes de classification en deux classes [67] : les méthodes de classification *hiérarchiques* et *non-hiérarchiques*, que nous allons traité ci-après. Mais tout d'abord, il faut poser le problème de la mesure de la distance.

3.2.1 Mesure de la distance

La sélection de la mesure de la distance est primordiale car elle permet de mesurer la similarité mais aussi la dissimilarité entre deux objets. Voici quelques distances souvent utilisées :

- *la distance Euclidienne*, elle est définie comme la racine de la somme des carrés des différences des coordonnées.

$$d(x_i, x_k) = \left(\sum_{j=1}^{p} (x_i^j - x_k^j)^2 \right)^{1/2}$$

- *la distance de Manhattan* (ou City-Block) entre deux vecteurs mesure non pas l'hypoténuse du triangle rectangle P3P1P mais la longueur totale des deux autres côtés. On généralise à un espace à n dimensions :

$$d_M(P1P3) = \sum_{j=1}^{L} |a_j - c_j|$$

- *la distance de Hamming* soit le nombre de bits distincts dans une comparaison de mots binaires bit par bit :

$$d_H(a, b) = \sum_{i=0}^{n-1} (a_i \oplus b_i)$$

- *l'angle* entre deux vecteurs

$$u.v = |u||v|cos\varphi$$

- *la distance de Mahalanobis* est une distance d'une série de valeurs de moyenne $\mu = (\mu_1, \mu_2, \mu_3, \ldots, \mu_p)$ et possédant une matrice de covariance Σ pour un vecteur à plusieurs variables $x = (x_1, x_2, x_3, \ldots, x_p)$ définie comme suit :

$$D_M(x) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)}$$

3.2.2 Méthodes de partitionnement hiérarchiques

La méthode de classification hiérarchique soit construit des amas d'objets similaires (ascendante), soit divise un ensemble d'objets en sous-ensembles (descendante) [77]. La représentation traditionnelle d'une telle hiérarchie se fait grâce à un arbre appelé dendogramme. L'algorithme ascendant commence par les feuilles alors que l'algorithme descendant commence par la racine de l'arbre.

Ce type de classification se déroule en trois étapes :

1. trouver les similarités entre les paires d'objets
2. grouper les objets sous la forme d'un arbre, ceci est réalisé en considérant chaque élément dans une classe et regrouper les deux classes les plus proches au sens d'une distance ;
3. déterminer la coupe d'arbre en définissant le nombre de classes souhaitées. L'arbre binaire reflétant la structure des données et permettant de regrouper les données similaires s'appelle un dendogramme comme cité plus haut.

Cependant, l'inconvénient avec une telle approche est la détermination du seuil de la coupe, ainsi que l'absence de chevauchement des classes. Une donnée peut être proche thématiquement d'autres données appartenant à deux classes différentes.

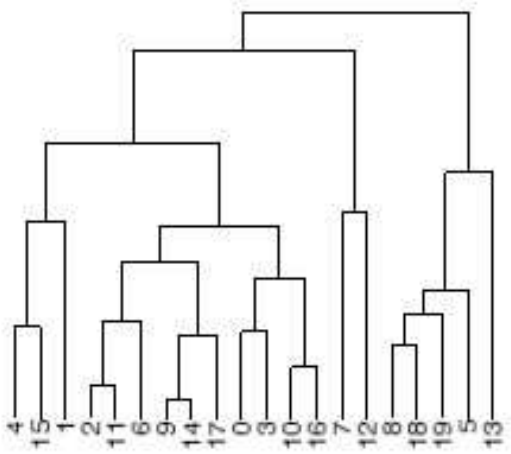


FIG. 3.1 – Exemple de dendogramme

3.2.3 Méthodes de partitionnement non-hiérarchiques

Algorithme des k-moyennes

Cet algorithme assigne chaque objet au sous-ensemble dont le centre est le plus proche de l'objet en question. L'algorithme de base de MacQueen [62] procède comme suit :

– Choisir le nombre de sous-ensembles k à obtenir.

- Choisir k groupes de manière aléatoire et en déterminer les centres, ou prendre aléatoirement k objets comme étant les centres initiaux.
- Assigner chaque objet au groupe dont le centre est le plus proche.
- Recalculer les centres de chaque groupe.
- Répéter les deux étapes précédentes tant que la composition des groupes change.

La performance de cette technique est proportionnelle à la qualité de la fonction de mesure de distance utilisée. La difficulté semble moindre quand les variables sont numériques, cependant, le nombre de possibilités est énorme. La difficulté s'accentue pour des valeurs énumératives non ordonnées. Lorsque aucune métrique n'est possible il est fréquent d'utiliser une distance égale à 0 pour des variables identiques et 1 dans le cas contraire. Une bonne fonction de distance donnera de bons résultats.

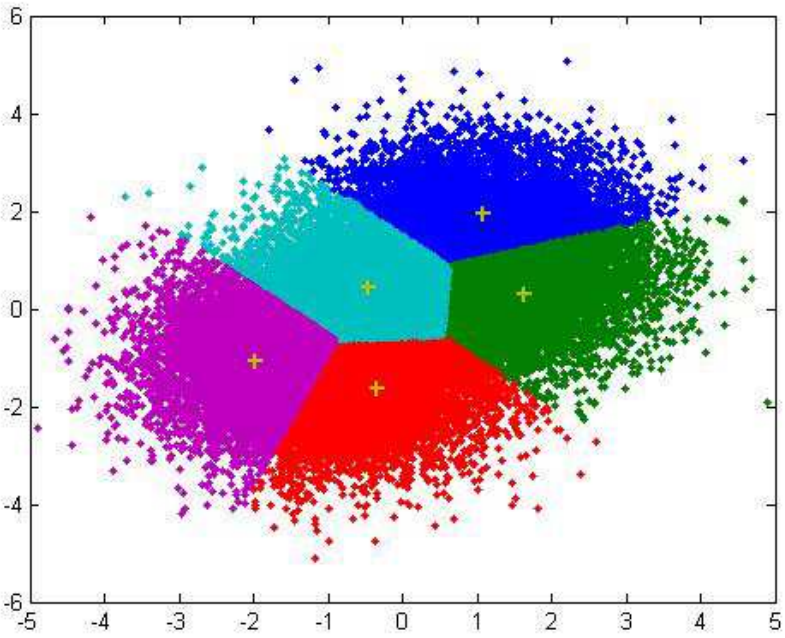


FIG. 3.2 – Exemple de partitionnement d'un espace avec les k-moyennes

3.3 Méthodes de classification prétopologiques

3.3.1 MCP et MCPR

Ces algorithmes ont été développés par Michel Lamure et Thanh Van Le, et apparaissent dans la thèse de cette dernière [52], lesquels nous proposerons sous une écriture de pseudo-code.
Concernant la notion de distance entre deux ensembles de E, nous appelons $\delta(A, B)$ la distance entre deux sous-ensembles A et B d'un ensemble fini E [52].

- $k_0 = min(min\{k|A \subset a^k(B)\}, \infty)$
- $k_1 = min(min\{k|B \subset a^k(A)\}, \infty)$
- $\delta(A, B) = min(k_0, k_1)$

Cette distance nommée pseudo-distance peut être calculée comme la distance bien connue de Hausdorff. Elle peut être utilisée pour mesurer la dissimilarité entre les objets représentés par différents types de variables. Concernant la complexité, on note deux boucles qui se suivent, nous avons donc dans le pire des cas $2 \times n$ adhérences. Nous prendrons tout au long de ce chapitre *l'adhérence comme opération de base*, la complexité de cet algorithme est donc en $O(n)$.

Pseudo-Distance

Méthode : δ(A, B)
Variables :
E : espace prétopologique
F_A, F_B, A', B' : ensembles
k_0, k_1 : entiers **Début**
  $A' \leftarrow A$
  $B' \leftarrow B$
  $F_A \leftarrow F(A)$
  $F_B \leftarrow F(B)$
  $k_0, k_1 \leftarrow \infty$
  Tant que $A \not\subset B'$ et $B \neq F_B$ **Faire**
    $B' \leftarrow a(B')$
    $k_0 \leftarrow k_0 + 1$
  Fin Tant que
  Tant que $B \not\subset A'$ et $A \neq F_A$ **Faire**
    $A' \leftarrow a(A')$
    $k_1 \leftarrow k_1 + 1$
  Fin Tant que
  Retourner $min(k_0, k_1)$
Fin

On détermine le centre d'un ensemble F comme suit :
On note x_0 le centre de l'ensemble.
Avec $F = \bigcup_{x \in F}\{x\}$, nous devons décider quel $\{x_i\}$ choisir. Pour cela, nous calculons $Card(a(x_i))$ avec $i \in [1, Card(F)]$. Nous choisissons x_0 tel que $Card(a(x_i))$ soit l'adhérence contenant le plus grand nombre de singletons.
Au cas où plus d'un x_0 existe, on choisit x_0 de manière à ce que celui-ci minimise la pseudo-distance avec *le plus grand fermé élémentaire qui le contient.*
En terme de complexité, on a tout au plus deux boucles imbriquées, sauf que dans l'une d'entre elles, on fait appelle à la pseudo-distance, qui est en $O(n)$. Nous somme donc en $O(n^2 \times n)$ soit $O(n^3)$ dans le pire des cas.

L'algorithme MCPR se base l'algorithme des *k-moyennes*. Celui-ci est souvent utilisé de par son efficacité et sa simplicité. Il fournit la possibilité de classifier des objets dans k classes par les k centres choisis a priori et qui sont employés comme germes afin de déterminer les k classes. Cependant, une des limites de cet algorithme est que le nombre de classes doit être prédéterminé et le procédé de choix des centres initiaux doit être effectué d'une certaine manière.
L'algorithme fondamental des méthodes de k-moyennes se déroule en 4 étapes (ci-dessous) :

1. Choisir k objets initiaux appelés centres des k classes en formant ainsi k classes.
2. (Ré)affecter chaque objet x à la classe C_i de centre M_i tel que $dist(O, M_i)$ soit minimale
3. Recalculer le centre M_i de chaque classe.
4. Aller à l'étape 2 jusqu'à ce que les objets ne changent plus de classe.

Au niveau de la complexité de l'algorithme MCPR, nous avons la première boucle *Tant que* qui ne dépend pas de n (le nombre d'éléments). On prendra donc k comme nombre d'itérations possibles. Dans cette boucle, nous trouvons au maximum deux boucles imbriquées dans lesquelles se trouve un calcul de pseudo-distance, soit une complexité en $O(n^3)$. En prenant en compte la première boucle, on obtient une complexité en $O(k \times n^3)$.

La méthode de classification MCP se fait selon deux processus :

1. *Processus de structuration :* il ressemble à celui de MCPR. Il consiste à chercher les familles des fermés et à déterminer des noyaux initiaux pour tous les fermés minimaux.
2. *Processus de partitionnement :* pour chaque élément qui n'est pas encore classé, nous choisissons une classe dans laquelle cet élément sera assigné de telle façon que deux contraintes ci-dessous soient satisfaites à la fois :
   – la distance entre cet élément et le noyau du groupe est la plus proche.
   – il y a des liens entre cet élément et tous les autres du groupe.

On utilise la notion de pseudo-distance pour la détermination originale des centres. Cependant, au moment de recalculer les centres pendant le déroulement de l'algorithme, on utilise une autre distance notée D basée sur la pseudo-distance et définie comme telle :

$$D(x, A) = \frac{\sum_{y \in A} \delta(x, y)}{Card(A)}$$

Initialisation du processus

Soit $\mathcal{G}$ la famille de l'ensemble des fermés minimaux sur E d'où :

$$\mathcal{G} = \mathcal{F}_m = \{F_{m_j}, j = 1, ..., n\} \text{ avec } n = Card(\mathcal{G})$$

Soit $\mathcal{F}_{em}$ la famille de l'ensemble des fermés élémentaires qui ne sont pas des fermés minimaux d'où :

Détermination des centres

Méthode : **Centres**($\mathcal{F}$)
Variables :
E : espace prétopologique
$\mathcal{F}$: famille d'ensembles en entrée
F : ensemble
$centres, liste_x0$: listes de singletons
max, min : entiers
Début
  Pour k de 0 à $Card(\mathcal{F})$ **Faire**
    $F \leftarrow \mathcal{F}_k$
    $max \leftarrow 0$
    Pour i de 0 à $Card(F)$ **Faire**
      Si $Card(a(\{F_i\})) > max$ **Alors**
        $max \leftarrow Card(a(\{F_i\}))$
      Fin Si
    Fin Pour
    Pour i de 0 à $Card(F)$ **Faire**
      Si $Card(a(\{F_i\})) = max$ **Alors**
        Ajouter $\{F_i\}$ à liste_x0
      Fin Si
    Fin Pour
    Si Card(list_x0) = 1 **Alors**
      Ajouter list_$x0_0$ à centres
    Sinon
      $min \leftarrow \infty$
      $ind_{min} \leftarrow 0$
      Pour i de 0 à Card(list_x0) **Faire**
        $sumdist \leftarrow 0$
        $sumdist \leftarrow sumdist + \delta(\{\text{list_}x0_i\}, F_e)$
        Si $sumdist < min$ **Alors**
          $min \leftarrow sumdist$
          $ind_{min} \leftarrow i$
        Fin Si
      Fin Pour
      Ajouter list_$x0_{ind_{min}}$ à centres
    Fin Si
  Fin Pour
  Retourner *centres*
Fin

MCPR

Méthode : **MCPR**()
Variables :
E : espace prétopologique
$\mathcal{F}_m$: famille des fermés minimaux
$classes$: liste d'ensembles
$centres$: liste de singletons
$stop$: booléen
C, O : ensembles
min, ind_{min} : entiers **Début**
  $stop \leftarrow FAUX$
  $O \leftarrow \emptyset$
  $centres \leftarrow \mathcal{F}_m$
  $classes \leftarrow \emptyset$
  Tant que stop = FAUX **Faire**
    $C \leftarrow \emptyset$
    Pour i de 0 à $|centres|$ **Faire**
      $classes_i \leftarrow centres_i$
      $C \leftarrow C \cup centres_i$
    Fin Pour
    $O \leftarrow E - C$
    Pour i de 0 à $Card(O)$ **Faire**
      $min \leftarrow \infty$
      $ind_{min} \leftarrow 0$
      Pour j de 0 à $|centres|$ **Faire**
        Si $\delta(\{O_i\}, centres_j) < min$ **Alors**
          $min \leftarrow \delta(\{O_i\}, centres_j)$
          $ind_{min} \leftarrow j$
        Fin Si
      Fin Pour
      $classes_{ind_{min}} \leftarrow classes_{ind_{min}} \cup \{O_i\}$
    Fin Pour
    Recalculer les $centres$ des $classes$
    Si centres identiques **Alors**
      $stop \leftarrow VRAI$
    Sinon
      $classes \leftarrow \emptyset$
    Fin Si
  Fin Tant que
  Retourner $classes$
Fin

$$\mathcal{F}_{em} = \mathcal{F}_e - \mathcal{F}_m = \{F_{em_i}, i = 1, ..., m\} \text{ avec } m = Card(\mathcal{F}_{em}).$$

On initialise la famille $\mathcal{G}$ avec $\mathcal{F}_m$, car celle-ci sera amenée à changer au cours de l'algorithme et cela sera notre résultat de partitionnement final.

Tri de $\mathcal{F}_{em}$

On trie la famille $\mathcal{F}_{em}$ de manière croissante au sens de la cardinalité tel que :

$$Card(F_{em_{i-1}}) < Card(F_{em_i}) < Card(F_{em_{i+1}})$$

Partitionnement

On exécute cette étape pour i de 1 à m, avec $m = Card(\mathcal{F}_{em})$. Pour visualiser les différentes appellations d'ensembles qui vont suivre, voir Fig 3.3.

On cherche les ensembles appartenant à $\mathcal{G}$ qui intersectent F_{em_i}.
Soit :

$$\mathcal{G}' = \{\forall G \in \mathcal{G} / G \cap F_{em_i} \neq \emptyset\}$$

$$H = F_{em_i} - \bigcup_{G'_j \in \mathcal{G}'} \bigcup_{x \in F_{em_i} \cap G'j} \{x\}$$

$\forall G' \in \mathcal{G}' / G' \subseteq F_{em_i}$, calculer la pseudo-distance de chaque singleton $\{x\}$ de H à chaque centre des ensembles G' qui satisfassent la condition précédente, puis affecter chaque {x} à l'ensemble G' qui donne la plus petite pseudo-distance.

Si en revanche, on a $\forall G' \in \mathcal{G}'$, $G' \not\subseteq F_{em_i}$, on note $\{x_0\}$ le centre de l'ensemble H calculé par le procédé de détermination des centres, à la différence qu'on utilise la fonction D à la place de la pseudo-distance δ (voir algorithme 3.3.1). Sélectionner l'ensemble $G' \in \mathcal{G}'$ tel que $\delta(x_0, centre(G'))$ soit minimale.

Un nouveau groupe G_{n+1} vide est créé et sera ajouté à $\mathcal{G}$ une fois le processus terminé. On sépare l'ensemble G' sélectionné en deux sous-ensembles :
- $Q_{G'} = G' - G' \cap F_{em_i}$
- $Q^c_{G'} = G' \cap F_{em_i}$

Si $D(centre(G'), G') > D(centre(G'), Q^c_{G'} \cup H)$

Alors $G_{n+1} = Q$ et $G' = Q^c_{G'} \cup H$ (on remplace l'ensemble G' que l'on a sélectionné dans $\mathcal{G}'$, et à fortiori dans $\mathcal{G}$).

Sinon $G_{n+1} = H$

Une fois que G_{n+1} est déterminé, on l'ajoute à notre famille $\mathcal{G}$ de partionnement ($\mathcal{G} = \mathcal{G} \cup \{G_{n+1}\}$) .

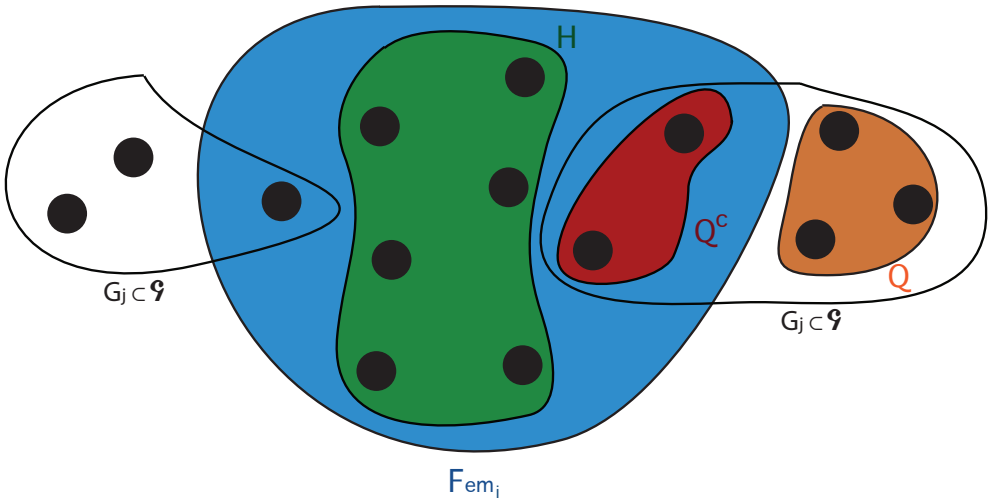


FIG. 3.3 – Exemple de découpage des ensembles utilisés pour MCP

L'algorithme MCP comprend au plus trois boucles imbriquées avec à l'intérieur un calcul de centres, ce qui donne une complexité de cette méthode en $O(n^3 \times n^2)$ soit $O(n^5)$.

3.3.2 Encadrement d'un espace prétopologique

Deux espaces prétopologiques moins fin et plus fin sont construits à partir d'un espace prétopologique quelconque associé à une famille de relations binaires.
Soit $B(x)$ la famille des voisinages de $\{x\}$ construite à partir des relations binaires réflexives initiales associées à un espace prétopologique quelconque. On appelle la base $G(x)$ obtenue par l'intersection des voisinages de chaque point comme suit :

$$G(x) = \bigcap_{B \in B(x)} B$$

Soit $G^s(x)$ une base plus fine que $G(x)$:

$$G^s(x) = G(x) - \{y / y \in G(x), x \notin G(y)\}$$

Soit $G^{s*}(x)$ une base moins fine que $G(x)$:

$$G^{s*}(x) = G(x) \cup \{h / h \notin G(x), x \in G(h)\}$$

On se base ensuite sur le calcul des fermés élémentaires comme initialisation des $G(x)$, puis on construit les voisinages encadrant ceux de départ. Les partitions finales seront déterminées par la recherche des fermés minimaux dans chacune d'entre elles.
L'algorithme ne comprend au maximum que deux boucles imbriquées, soit une complexité en $O(n^2)$ dans le pire des cas.

MCP

Méthode : **MCP**()
Variables :
$\mathcal{F}_m$: famille des fermés minimaux
$\mathcal{F}_e$: famille des fermés élémentaires
$\mathcal{G}, \mathcal{G}', \mathcal{G}'_{in}, \mathcal{G}'_{notin}, \mathcal{F}_{em}$: famille d'ensembles
$G, G', F, H, Q, Q^c, G'_{in}, G'_{notin}$: ensembles
i, j, k, t, min, ind : entiers
Début
  $\mathcal{G} \leftarrow \mathcal{F}_m$
  $\mathcal{F}_{em} \leftarrow \mathcal{F}_e - \mathcal{F}_m$
  Ordonner($\mathcal{F}_{em}$)
  Pour i de 1 à $Card(\mathcal{F}_{em})$ **Faire**
    $\mathcal{G}'_{in}, \mathcal{G}'_{notin} \leftarrow \emptyset$
    $F \leftarrow F_{em_i}$
    $\mathcal{G}' \leftarrow \{G \in \mathcal{G} / G \cap F \neq \emptyset\}$
    $H \leftarrow F - \cup_{G'_j \in \mathcal{G}'} \cup_{x \in F_{em_i} \cap G'j} \{x\}$
    Pour k de 1 à $Card(\mathcal{G}')$ **Faire**
      Si $G'_k \subseteq F$ **Alors**
        $\mathcal{G}'_{in} \leftarrow \mathcal{G}'_{in} \cup \{G'_k\}$
      Sinon
        $\mathcal{G}'_{notin} \leftarrow \mathcal{G}'_{notin} \cup \{G'_k\}$
      Fin Si
    Fin Pour
    Si $Card(\mathcal{G}'_{in}) > 0$ **Alors**
      Pour k de 1 à $Card(H)$ **Faire**
        $\{x\} \leftarrow H_k$
        $min \leftarrow \infty$
        $ind \leftarrow 0$
        Pour t de 1 à $Card(\mathcal{G}'_{in})$ **Faire**
          Si $\delta(\{x\}, centre(G'_{in_t})) < min$ **Alors**
            $min \leftarrow \delta(\{x\}, centre(G'_{in_t}))$
            $ind \leftarrow t$
          Fin Si
        Fin Pour
        $G'_{in_{ind}} \leftarrow G'_{in_{ind}} \cup \{x\}$
        Remplacer l'ancien $G'_{in_{ind}}$ de $\mathcal{G}$ par le nouveau $G'_{in_{ind}}$
      Fin Pour
    Sinon
      $\{x_0\} \leftarrow centreD(H)$
      $min \leftarrow \infty$
      $ind \leftarrow 0$
      Pour t de 1 à $Card(\mathcal{G}'_{notin})$ **Faire**
        Si $\delta(\{x_0\}, centreD(G'_{notin_t})) < min$ **Alors**
          $min \leftarrow \delta(\{x_0\}, centreD(G'_{notin_t}))$
          $ind \leftarrow t$
        Fin Si
      Fin Pour
      $G' \leftarrow G'_{notin_{ind}}$
      $Q \leftarrow G' - G' \cap F$
      $Q^c \leftarrow G' \cap F$
      Si $D(centreD(G'), G') > D(centreD(G'), Q^c \cup H)$ **Alors**
        $G' \leftarrow Q^c \cup H$
        Remplacer l'ancien G de $\mathcal{G}$ par G'
        $\mathcal{G} \leftarrow \mathcal{G} \cup \{Q\}$
      Sinon
        $\mathcal{G} \leftarrow \mathcal{G} \cup \{H\}$
      Fin Si
    Fin Si
  Fin Pour
  Retourner $\mathcal{G}$
Fin

Méthode d'encadrement

Méthode : **Encadrement**(E)
Variables :
E : espace prétopologique
$\mathcal{F}_e, \mathcal{F}_e^s, \mathcal{F}_e^{s*}$: ensembles
Début
$\mathcal{F}_e \leftarrow \cup F_e$
$\mathcal{F}_e^s \leftarrow \emptyset$
$\mathcal{F}_e^{s*} \leftarrow \emptyset$
Pour i de 1 à $Card(E)$ **Faire**
$\{x\} \leftarrow E_i$
$G(x) \leftarrow F(\{x\})$
$G^s(x) \leftarrow \emptyset$
$G^{s*}(x) \leftarrow G(x)$
Pour k de 1 à $Card(G(x))$ **Faire**
$\{y\} \leftarrow G(x)_k$
Si $\{x\} \in F(\{y\})$ **Alors**
$G^s(x) \leftarrow G^s(x) \cup \{y\}$
Fin Si
Fin Pour
Pour k de 1 à $Card(E - G(x))$ **Faire**
$\{h\} \leftarrow (E - G(x))_k$
Si $\{x\} \in F(\{h\})$ **Alors**
$G^{s*}(x) \leftarrow G^{s*}(x) \cup \{h\}$
Fin Si
Fin Pour
$\mathcal{F}_e^s \leftarrow \mathcal{F}_e^s \cup \{G(x)^s\}$
$\mathcal{F}_e^{s*} \leftarrow \mathcal{F}_e^{s*} \cup \{G(x)^{s*}\}$
Fin Pour
Retourner FermésMinimaux($\mathcal{F}_e, \mathcal{F}_e^s, \mathcal{F}_e^{s*}$)
Fin

3.4 Classification prétopologique & complexité de Kolmogorov

Dans cette partie, nous allons utiliser les méthodes de classification prétopologiques en se basant sur un indice de similarité, lequel sera calculé à partir de la description élémentaire des singletons de l'espace grâce la complexité dite de Kolmogorov.

3.4.1 Complexité de Kolmogorov

Description

Afin de caractériser un *objet*, on peut parler de sa complexité, mais plus précisément de la mesure de cette complexité. Andrei Kolmogorov était un mathématicien et un des fondateurs de la théorie de la complexité algorithmique. La complexité d'une chaîne selon Kolmogorov est la longueur de la description la plus courte de la chaine suivant un langage donné. L'exemple le plus parlant consiste à définir les deux chaînes de bits suivantes :

1. 01010101010101010101010101010101
2. 01011101011010010101010111110110

Dans cet exemple, la chaîne 1 peut être décrite comme 16 fois 01 alors que la chaîne 2 n'a pas de meilleure description qu'elle même, ce qui correspond à un nombre aléatoire. La complexité de Kolmogorov fournit de fait une mesure de la complexité de description de l'information.

Notions de base

Dans toute la suite les codages se feront sur l'alphabet $\Omega = \{0,1\}^*$. La complexité de Kolmogorov mesure la quantité d'information contenue dans un mot, et le définit comme la taille du plus petit programme qui engendre ce mot. On notera qu'elle ne fait pas intervenir le temps nécessaire à la production du mot, et que la machine considérée peut ne pas s'arrêter.

Définition 3.1 *Soit M une machine de Turing universelle. Pour $x \in \Omega$, on définit la complexité de x relativement à M ainsi :*

$$K_M(x) = min\{|y| : M(y) = x\}$$

Le problème majeur avec la complexité de Kolmogorov est qu'elle n'est pas calculable [26] et qu'on peut seulement approcher sa borne supérieure [72, 60]. Néanmoins, on peut interpréter la complexité de Kolmogorov ainsi : M est un algorithme de décompression, et y est une version compressée de x, ce qui nous amène au paragraphe suivant.

3.4.2 Estimation de la complexité d'un objet

Une des applications de la complexité de Kolmogorov est « *liée à la propriété de compressibilité, c'est elle qui permet d'optimiser la compression de données sans pertes* »[49]. Expliquons ce qu'est un compresseur "*normal*" d'après la définition de [21] :

Définition 3.2 *Un compresseur C est dit normal, possédant une complexité en $O(log(n))$ avec n la longueur binaire maximale d'un élément de Ω, s'il satisfait les conditions suivantes :*

1. Idempotence *: $C(xx) = C(x)$ et $C(\lambda) = 0$ avec λ étant le mot vide.*
2. Monotocité *: $C(xy) \geq C(x)$.*
3. Symétrie *: $C(xy) = C(yx)$.*
4. Distributivité *: $C(xy) + C(z) \leq C(xz) + C(yz)$.*

La notation $C(y|x)$ peut être vue comme l'excédant de bits contenus dans la version compressée de xy par rapport à la version compressée de x. On la définit comme telle :

$$C(y|x) = C(xy) - C(x)$$

On peut ainsi définir une *distance* de compression en se basant sur notre compresseur C. Celui-ci permet une approximation la distance d'information entre deux éléments x et y, noté $E(x, y)$ et basée sur la complexité de Kolmogorov, par la définition suivante [9] :

$$E_C(x, y) = C(xy) - min\{C(x), C(y)\}$$

où $E_C(x, y)$ est la distance de compression entre x et y. Dans cette définition, $C(x)$ représente la taille binaire de x compressé, $C(y)$ représente la taille binaire de y compressé, et $C(xy)$ représente la taille binaire de la concaténation de x et y, le tout compressé.
Grâce à l'approximation fournie par C, on voudrait le résultat approché de la normalisation de $E(x, y)$ nommée distance d'information normalisée et définie [59] par :

$$NID(x, y) = \frac{max\{K(x|y), K(y|x)\}}{max\{K(x), K(y)\}}$$

On peut donc réécrire cette distance avec notre compresseur C en remplaçant K par C, en simplifiant de la façon suivante [21] :

$$NCD(x, y) = \frac{C(xy) - min\{C(x), C(y)\}}{max\{C(x), C(y)\}}$$

On appellera cette distance *distance de compression normalisée*. Avec cette distance, on peut donc définir une distance entre deux objets quel qu'ils soient en comparant finalement la plus petite description de chacun.

3.4.3 Utilisation de la distance de compression avec la prétopologie

Nous allons dans cette partie illustrer par une application les notions théoriques évoquées ci-avant, en reprenant la modélisation d'auteurs vue dans le chapitre 2.4.2. Voici comment est construit le modèle : nous avons des auteurs, ainsi que l'intitulé des articles que ceux-ci ont écrits. L'objectif du modèle est de retrouver les auteurs en relation en fonction du titre de l'article qu'ils ont écrits, donc du sujet qu'ils ont abordé, puis de les classer selon leur domaine.
Chaque auteur correspond au moins un article, et pour chaque article correspond au

moins un auteur. Pour connaître la "*distance*" entre chaque article, on calcule toutes les distances de compression normalisées pour chaque paire d'articles en prenant comme mot à compresser la correspondance binaire de leur titre.
Ainsi comment procède-t-on pour effectuer l'adhérence d'un ensemble A d'auteurs ? Voici notre méthode :

1. On prend tous les articles auxquels les auteurs de l'ensemble A ont participé, soit l'ensemble B.
2. On recherche les voisins des articles de B au sens de la distance de compression normalisée et selon un seuil fixé, ce qui nous donne l'ensemble B'.
3. Pour finir, on recherche les auteurs ayant participé à l'écriture des articles de l'ensemble B', qui nous donne A'.

Nous avons donc ici : $a(A) = A'$. Pour formaliser un tel modèle, il faut procéder en plusieurs étapes : tout d'abord, nous devons construire deux espaces prétopologiques, un pour les articles, et un pour les auteurs. Pour passer des auteurs aux articles et vice-versa, on utilisera deux fonctions surjectives (voir Fig. 3.4) dont les définitions[29] sont données ci-après.

Définition 3.3 *Soient X et Y deux ensembles, et f une application de X dans Y. On dit que f est une application surjective ou que f est une surjection si et seulement si :*

$$\forall\ y \in Y, \exists\ x \in X, y = f(x)$$

Définition 3.4 *Soient X et Y deux ensembles, et f une application de X dans Y. Soit A une partie de X, on appelle image de A par l'application f la partie de Y notée $f(A)$ définie par :*

$$f(A) = \{y \in Y; x \in X, y = f(x)\}$$

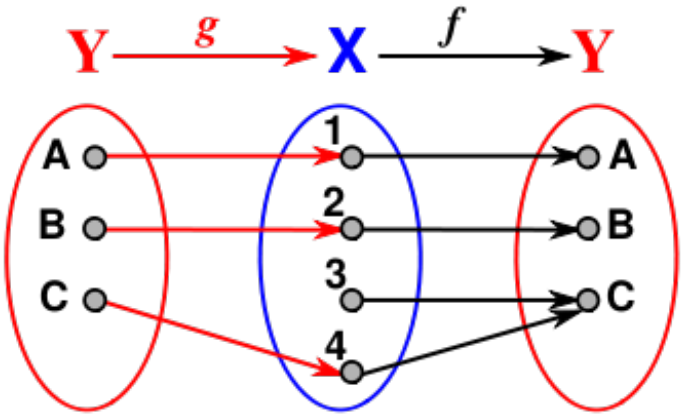


FIG. 3.4 – Fonctions de surjection de Y dans X et de X dans Y.

On définit l'ensemble E comme l'ensemble des auteurs et l'ensemble G comme l'ensemble des articles. Comment savoir si deux articles sont "*proches*" ou pas ? En utilisant la

définition de la distance de compression normalisée, on construit un espace prétopologique avec des relations valuées (à la manière d'un graphe complet), ces relations représentant la distance entre deux articles (calculée par la distance de compression normalisée appliquée au titre de l'article). Ainsi, les articles qui contiennent des mots similaires auront une distance qui les séparent plus faible que des articles qui ne possèdent aucun terme en commun, même s'il existe un risque de biais. Voici comment on le formalise :

$$\text{Soit } G \times G \to \mathbb{R}_+^*, (x,y) \to v(x,y)$$

$$\forall B \in \mathcal{P}(G), s \in \mathbb{R}_+^*, a_G(B) = \{y \in G - B, Moy(\sum_{x \in B} v(x,y)\ ) \leq s\} \cup B$$

L'application adhérence $a_G(.)$ nous permet d'exprimer la proximité entre articles. Le couple (G, a_g) correspond donc à l'espace prétopologique des articles. Nous allons nous servir de cette adhérence pour construire celle de l'espace prétopologique des auteurs. Définissons tout d'abord les deux fonctions surjectives nous permettant de passer de E à G et de G à E :

Soit f une application de E dans G, A une partie de E, on définit l'image de A par l'application f la partie de G notée $f(A)$ par $f(A) = \{y \in G; x \in E, y = f(x)\}$.
Soit g une application de G dans E, B une partie de G, on définit l'image de B par l'application g la partie de E notée $g(B)$ par $g(B) = \{y \in E; x \in G, y = g(x)\}$.

En somme, f nous permet de déterminer quels articles ont été écrits par un ensemble d'auteurs et g quels auteurs ont écrits un ensemble d'articles. Nous pouvons à présent construire l'adhérence sur E :

$$\forall A \in \mathcal{P}(E), a(A) = \{x \in E, g(a_G(f(A)))\}$$

Le couple (E, a) correspond à l'espace prétopologique qui nous intéresse, c'est-à-dire celui portant sur les auteurs.

Pour illustrer ce modèle, nous allons donner un exemple d'utilisation basé sur des données réelles. Prenons l'ensemble E des auteurs tel que :

$$E = \{aut_a, aut_b, aut_c, aut_d, aut_e, aut_f, aut_g, aut_h\}$$

et G l'ensemble des articles tel que :

$$G = \{art1, art2, art3, art4, art5, art6\}$$

Tout d'abord, construisons l'espace (G, a_G) en calculant les distances normalisées de compression entre chaque paire d'article. On peut voir le résultat sur la figure 3.5. Plus cette distance est grande, moins les articles sont proches.

Définissons ensuite nos deux fonctions de surjection f et g. La fonction f permet le passage de E à G et la fonction g le passage de G à E (Fig. 3.6).

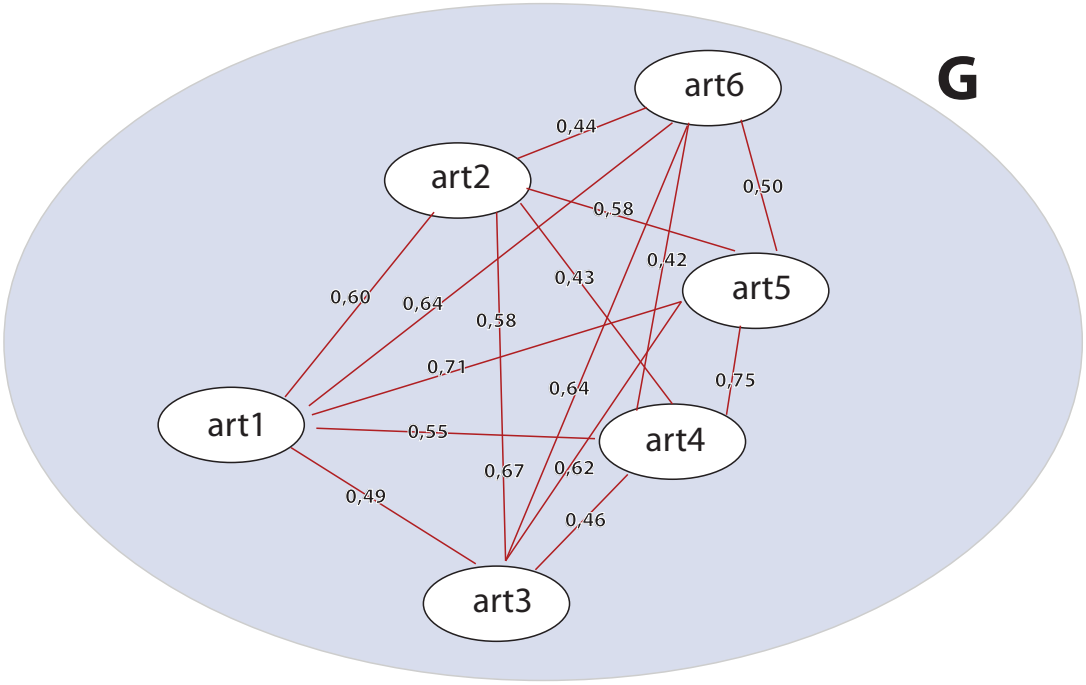


FIG. 3.5 – Espace prétopologique des articles (G, a_G)

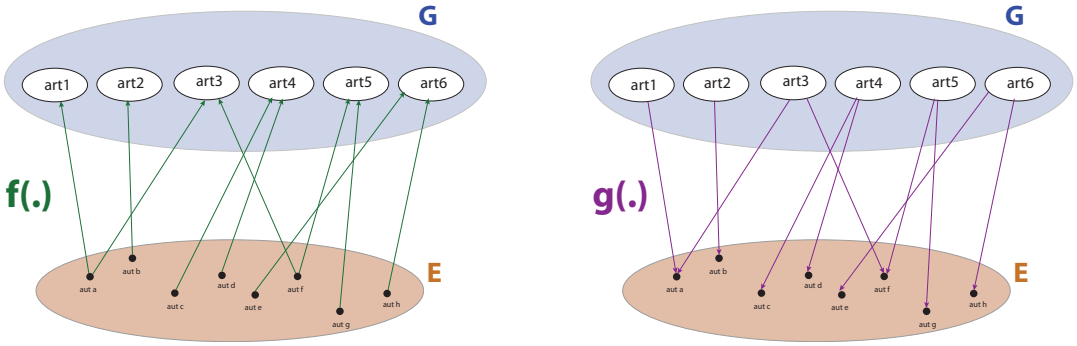


FIG. 3.6 – Fonctions de surjection f et g

Comment fonctionne le modèle si on veut faire l'adhérence d'un auteur, par exemple $a(\{aut_a\})$? Si on se réfère à la composition de $a(.)$, on a $a(\{aut_a\}) = g(a_G(f(\{aut_a\})))$, l'adhérence d'un auteur se faisant en trois étapes que voici :

1. On applique premièrement $f(.)$ à $\{aut_a\}$, ce qui nous donne (voir Fig. 3.6) :

$$f(\{aut_a\}) = \{art1, art3\}$$

   Cela signifie tout simplement que aut_a a participé à l'écriture de l'article 1 et 3.

2. On possédons maintenant une partie de G sur laquelle on peut appliquer l'adhérence $a_G(.)$. Prenons comme seuil $s = 0,6$. Si nous faisons les moyennes des distances normalisées de compression (NCD) entre $\{art1, art3\}$ et $G - \{art1, art3\}$, on obtient (voir Fig. 3.5) :
   - $Moy(NCD(art1, art2), NCD(art3, art2)) = 0,59$
   - $Moy(NCD(art1, art4), NCD(art3, art4)) = 0,5$
   - $Moy(NCD(art1, art5), NCD(art3, art5)) = 0,66$
   - $Moy(NCD(art1, art6), NCD(art3, art6)) = 0,64$

   On constate que seuls les articles 2 et 4 ont une distance moyenne à $\{art1, art3\}$ inférieure ou égal à s soit $0,6$. On a donc :

$$a_G(\{art1, art3\}) = \{art1, art3, art2, art4\} \text{ avec } s = 0,6$$

3. A présent, on allons appliquer $g(.)$ sur le résultat de $a_g(\{art1, art3\})$ pour finaliser le processus d'adhérence de $\{aut_a\}$. On applique donc $g(.)$ à $\{art1, art3, art2, art4\}$ (voir Fig. 3.6) :

$$g(\{art1, art3, art2, art4\}) = \{aut_a, aut_f, aut_b, aut_c, aut_d\}$$

Le résultat de ces trois étapes nous permet ainsi d'écrire :

$$a(\{aut_a\}) = \{aut_a, aut_f, aut_b, aut_c, aut_d\}$$

On peut constater par cet exemple de modélisation que la prétopologie peut aisément concevoir des représentations de réseaux complexes. Les méthodes de classifications vues plus haut s'intègrent parfaitement dans ce type de modèle, étant basé sur l'adhérence. On peut voir sur la figure 3.7 la capture d'écran d'une simulation d'application sur un petit réseau d'auteurs de MCP utilisant l'adhérence basée sur la distance normalisée de compression. Concernant la complexité de Kolmogorov, on pourrait être plus précis sur la distance d'un objet à un autre, car ici on ne se réfère qu'à une comparaison selon une taille en bits de mots compressés, perdant ainsi le côté qualitatif de cette comparaison. La prochaine étape concernant cet indice de similarité serait de prendre en compte une certaine « proximité sémantique ».

3.5 Conclusion

Durant la lecture de ce chapitre, on a pu constater que des méthodes de classification pouvaient être utilisées dans les espaces prétopologiques, et qu'elles pouvaient être

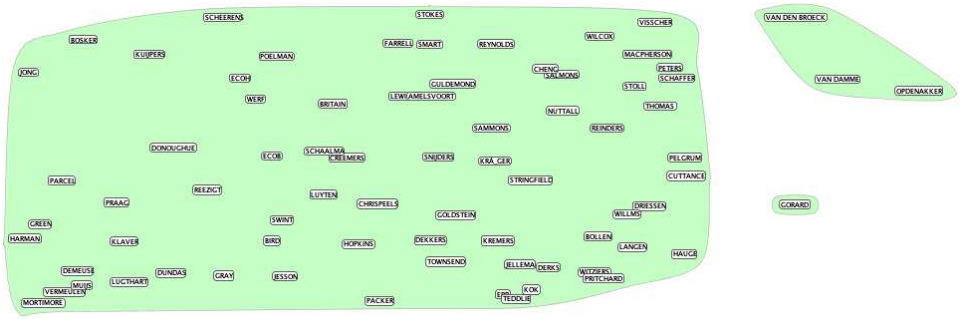


FIG. 3.7 – Classification d'un petit réseau d'auteurs par MCP basé sur la distance normalisée de compression

appliquées sur tout type de données. Etant basées sur l'adhérence comme la plupart des algorithmes prétopologiques car définissant la proximité, on a montré qu'il était possible d'utiliser la définition de cette adhérence pour définir différentes sortes de similarité, permettant de donner des résultats de classification différents selon le modèle étudié, faisant de la prétopologie un outil complet pour construire des modèles au comportement complexe. La modélisation d'auteurs présentée comme un cas particulier peut être utilisée de manière générique pour d'autres types de problèmes. Une des perspectives envisageables serait une comparaison de l'aspect théorique à l'aspect qualitatif d'une telle modélisation, ouvrant une direction vers des recherches intéressantes.

Chapitre 4

Centralité et distances à la frontière d'un espace prétopologique

L'analyse d'un réseau peut nécessiter que l'on s'intéresse à ses éléments dits « remarquables », dont le ou les centres en font partie. La recherche des centres d'un réseau est une question assez naturelle et générique. Dans ce chapitre, après une discussion autour de la notion de *centre*, nous nous intéresserons à la définition d'un centre et plus particulièrement en topologie mathématique. C'est à partir de travaux existants dans ce domaine que nous allons définir la notion de centre dans un espace prétopologique. Ainsi, nous proposerons de nouvelles méthodes de recherche de centres en découlant, permettant ainsi de nouvelles mesures dans les réseaux complexes, pour une meilleure analyse de ces derniers, illustrations à l'appui.

4.1 Introduction

La notion de *centre* diffère selon le domaine que l'on étudie. En sociologie, on parle de centre mais également de centralité [24, 12]. Plus on aura une structure des éléments du réseau qui suit une distribution en loi de puissance avec l'apparition de hubs, plus on aura affaire à une centralisation du réseau, où les quelques éléments centraux forment les points charnières du dit réseau [6].
En physique, la notion de centre renvoie à la notion de centre d'inertie ou de centre de masse. Les mathématiciens calculent par exemple des centres de gravité ou barycentres. On parle d'ailleurs de calcul barycentrique en chimie quand il s'agit de calculer la polarité d'une molécule.
Selon que l'on travaille avec les statistiques (calcul des moyennes pondérées), avec les probabilités (le barycentre entre des valeurs pondérées par leur probabilité est l'espérance mathématique) ou en géométrie analytique (géométrie euclidienne, géométrie affine), la signification de centre est différente. Cependant, il y a derrière toutes ces notions la même volonté de localiser un certain *point de convergence*.
Dans ce chapitre, nous nous penchons sur la définition de centre en topologie mathématique et plus particulièrement sur une adaptation de cette définition dans le cadre de la prétopologie. Plus précisément, nous voudrions pouvoir analyser un réseau quel qu'il soit afin de ca-

ractériser sa dynamique et ranger ses éléments dans des classes d'équivalences, à l'instar de travaux déjà réalisés [53].

Nous amènerons la notion de *pseudo-F-distance*, qui est une pseudo-distance d'un sous-ensemble à la frontière de l'ensemble le contenant, nous amenant à trouver le ou les centres de cet ensemble.
Puis nous présenterons deux méthodes de calcul des ϵ-centres prétopologiques avec quelques exemples illustratifs.

4.2 Distance à la frontière d'un ensemble

Pour déterminer le centre d'un sous-ensemble d'un espace prétopologique E, l'idée est que plus un élément x d'un ensemble A avec $\forall A \subseteq E, x \in A$ est éloigné de la frontière de A, plus il se situe au centre de celui-ci.

4.2.1 F-distance et pseudo-F-distance

Originellement, le concept de F-distance et d'ϵ-centres a été introduit et généralisé par Bui et Huard en 1966 [18] dans le but de résoudre des programmes mathématiques à contraintes non-linéaires, où $\mathbb{R}^n$ est substitué à un espace topologique. Avant de donner une définition des F-distances, rappelons ici la définition de la frontière d'un ensemble : la frontière de A, notée $Fr(A)$ est son adhérence privée de son intérieur.
En topologie, on la note habituellement comme ceci : $Fr(A) = \overline{A} \cap \overline{E \setminus A}$. En langage prétopologique, on écrit l'application frontière comme suit : $fr(A) = a(A) - i(A)$.

Définition 4.1 *Selon [18], soit E un espace topologique, et soit $\mathcal{B}$ un sous-ensemble de $\mathcal{P}(E)$.*
On appelle pseudo-F-distance sur $E \times \mathcal{B}$ une application $D : E \times \mathcal{B} \to \mathbb{R}$ telle que :

- $\forall B \in \mathcal{B}$ *et* $\forall x \in B$, $D(x, B) \geq 0$ (P_1)
- $\forall B \in \mathcal{B}$ *et* $\forall x \in Fr(B)$, $D(x, B) = 0$ (P_2)
- $\forall B$ *et* $B' \in \mathcal{B}$ *on a* $x \in B$ *et* $B \subset B' \Rightarrow D(x, B) \leq D(x, B')$ (P_3)

Définition 4.2 *On appelle F-distance sur $E \times \mathcal{B}$ une pseudo-F-distance D telle que $\forall B \in \mathcal{B}, x \in B$ et $D(x, B) = 0 \Rightarrow x \in Fr(B)$, soit $D(x, B) = d(x, Fr(B))$*

Il faut préciser ici qu'une F-distance n'est pas une métrique. En effet, si E est un espace métrique, la métrique d de E est une application de $E \times E$ dans $\mathbb{R}_+$ alors qu'une F-distance est une application de $E \times \mathcal{B}$ dans $\mathbb{R}$. Alors qu'une distance d'un point à un ensemble est définie par $d(x, A) = inf_{y \in A} d(x, y)$, une F-distance est une généralisation de la notion de distance d'un point à la frontière d'un ensemble [18].

Dans nos travaux, nous nous inspirons des définitions précédentes pour définir une F-distance en prétopologie. Plusieurs notions sont à définir :

– la pseudo-distance d'un ensemble à un autre
– une dérivée prétopologique de la F-distance qui correspondrait à nos besoins

4.2.2 Pseudo-distance Δ

Nous définissons notre pseudo-distance comme étant le nombre d'adhérences nécessaires pour qu'un ensemble intersecte un autre ensemble, autrement dit l'atteigne (Fig. 4.1).

Définition 4.3 *Nous notons $\Delta(A, B)$ la pseudo-distance entre deux sous-ensembles A et B d'un ensemble E tel que :*

$$\Delta(A, B) = \{k \in \mathbb{N}_+ / \ a^k(A) \cap B\}$$

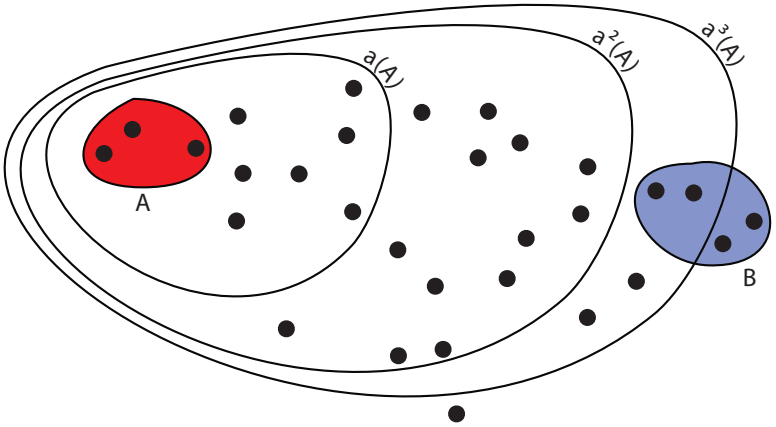


FIG. 4.1 – Pseudo-distance de A à B égale à 3.

Cette définition est extrêmement importante car elle nous permettra de calculer la distance des éléments d'un ensemble à sa frontière.

4.2.3 F-distance prétopologique et B-distance

Si on se base sur la définition d'une F-distance, calculer une pseudo-F-distance prétopologique (pFd) revient à calculer la pseudo-distance Δ d'un élément de l'ensemble à sa frontière, soit :

$$\forall x \in i(A), pFd(x, A) = \Delta(x, fr(A))$$

Un problème survient : comme on a $fr(A) = a(A) - i(A)$, il est clair que le calcul d'une pFd est inutile et inadapté avec une telle définition de la frontière. Quand on regarde la définition topologique de la frontière, on tombe sur un cas similaire : $Fr(A) = F(A) \cap F(A^c)$. Avec ces deux définitions, la valeur de la pFd sera toujours égale à 0 ou 1. En revanche, comme un espace prétopologique est structuré par ses fermés [48], il serait judicieux de calculer les distances entre les éléments qui composent un fermé et la frontière de celui-ci, et plus particulièrement la frontière d'un fermé élémentaire (Fig. 4.2).

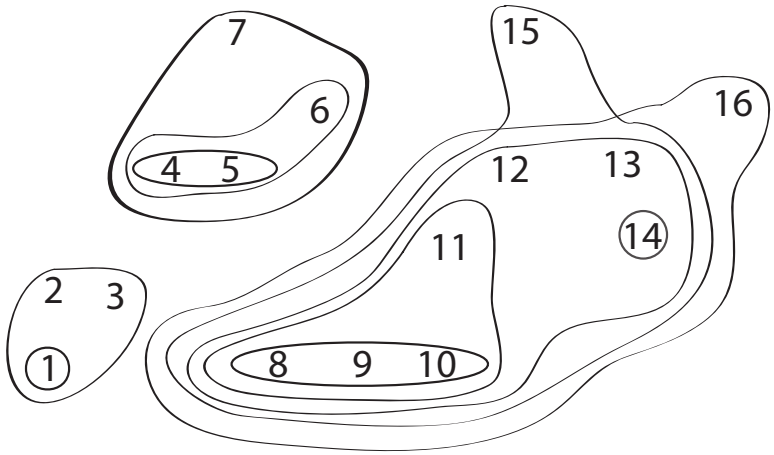


FIG. 4.2 – Famille des fermés élémentaires d'un espace E.

Voici ce qu'on obtient pour la frontière d'un fermé quelconque :

$$fr(F(A)) = a(F(A)) - i(F(A)),$$
$$soit\ fr(F(A)) = F(A) - i(F(A))$$

Ce résultat est tout à fait normal puisqu'un fermé ne peut plus évoluer, on a donc $a(F(A)) = F(A)$. La frontière prétopologique d'un fermé nous donne en fait la fonction de bord [7] définie comme tel :

Définition 4.4 *Le bord est la fonction de $\mathcal{P}(E)$ dans $\mathcal{P}(E)$ notée b définie par :*

$$b(A) = \{x \in A / \forall\ V, V \in \mathcal{V}(x), V \cap A^c \neq \emptyset\}$$
$$\forall A, A \in \mathcal{P}(E), \quad b(A) = A - i(A)$$

Cette fonction est également appelée semi-frontière intérieure. Dans notre cas, la notion de bord suffit car le fait d'atteindre la semi-frontière intérieure signifie atteindre la frontière, ce qui correspond à nos besoins.
On introduit donc une nouvelle pseudo-distance partant d'un singleton jusqu'au bord du fermé le contenant (Fig. 4.3), qui diffère donc d'une F-distance et que l'on appellera B-distance selon la définition qui suit :

Définition 4.5 *On appelle B-distance une application $D_b : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$ telle que :*

$$\forall\ x \in E, D_b(x, F(x)) = \Delta(x, b(F(x)))$$

Notre concept de B-distance nous permettra de trouver les centres des fermés élémentaires puis de les regrouper en groupes cohérents pour ainsi structurer notre espace prétopologique. Dans la section suivante, nous allons expliciter comment à partir des fermés élémentaires nous localiserons leurs centres amenant au processus de structuration.

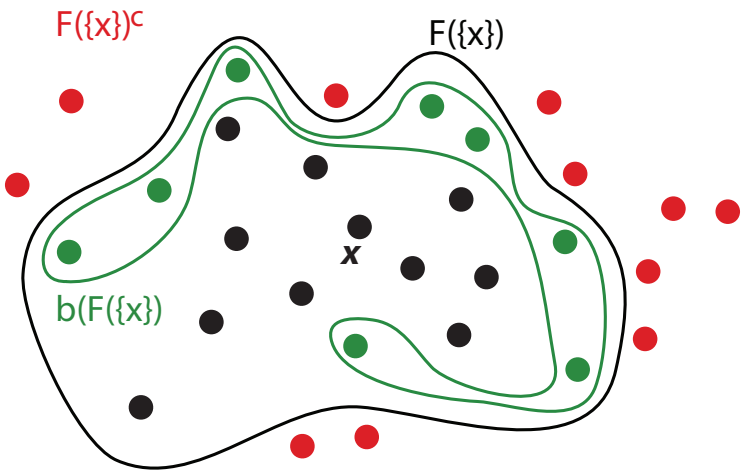


FIG. 4.3 – Bord d'un fermé élémentaire

4.3 Méthode des ϵ-centres

4.3.1 Méthode des ϵ-centres en topologie

La méthode des ϵ-centres a également été introduite par Bui et Huard [18] à l'instar des F-distances. D'ailleurs, on peut constater en faisant quelques recherches que cet algorithme continue d'être cité et utilisé jusqu'alors dans des articles en mathématiques ou en informatique. Voici comment est défini un ϵ-centre en topologie :

Définition 4.6 *Soit E un espace topologique, $\mathcal{B}$ un sous-ensemble de $\mathcal{P}(E)$, et soit D une pseudo-F-distance sur $E \times \mathcal{B}$. On appelle ϵ-centre d'un ensemble $B \in \mathcal{B}$ où ϵ est tel que $0 \leq \epsilon < sup_{x \in B} D(x, B)$ tout point $\hat{x}$ de B tel que :*

$$D(\hat{x}, B) \geq sup_{x \in B} D(x, B) - \epsilon$$

Avec cette définition, il est clair que si l'on ne devait garder qu'un seul centre de B, on choisirait le *0-centre* (s'il n'y en a qu'un seul bien-sûr). On peut se donner une représentation d'ϵ-centres d'un espace topologique délimité par plusieurs courbes avec l'illustration de la figure 4.4 pour différentes valeurs d'ϵ. Les *0-centres* sont situés au centre de l'espace, alors que les *100-centres* sont eux situés à la frontière de l'espace.

4.3.2 Méthode des ϵ-centres en prétopologie

L'idée derrière la transposition de la méthode topologique vers la méthode prétopologique est la suivante : le centre d'un fermé est le singleton le plus éloigné du bord de ce fermé

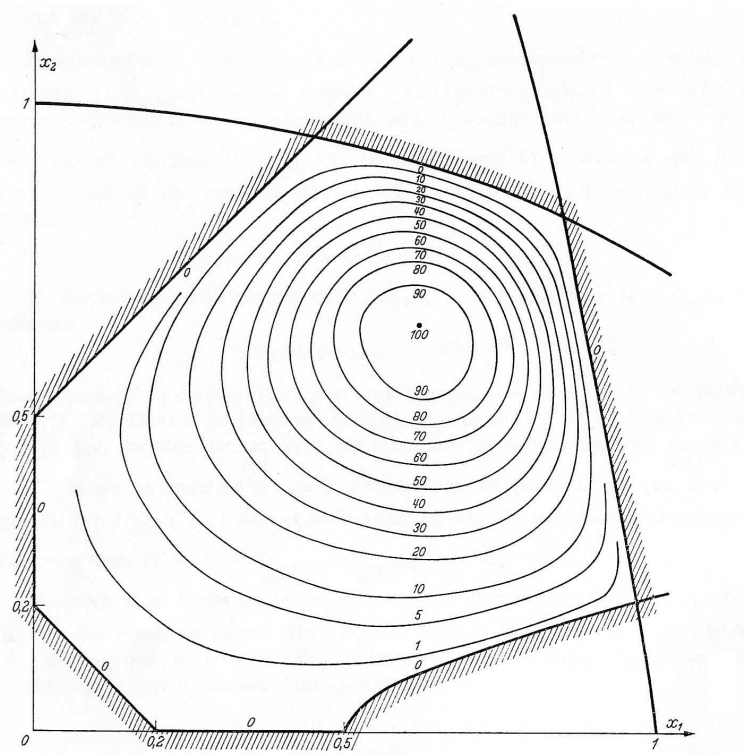


FIG. 4.4 – Exemple d'ϵ-centres dans un espace topologique

selon la pseudo-distance Δ. La définition d'un ϵ-centre est donc quasiment la même qu'en topologie sauf dans le cas où le fermé est égal à E. Le bord de E est l'ensemble vide car : $b(E) = a(E) - i(E) = E - E = \emptyset$. Nous devons déterminer une quasi-semi-frontière intérieure de E, qui ne peut être trouvée avec la fonction bord. Pour cela, nous introduisons la notion de *lisière d'un fermé*. Il suffit d'arrêter le processus de fermeture juste avant d'obtenir un fermé, d'où la définition suivante :

Définition 4.7 *Soit $\mathcal{F}_e$ la famille des fermés élémentaires, la lisière est la fonction de $\mathcal{F}_e$ dans $\mathcal{P}(E)$ notée l, définie par $\forall\ x \in E, \forall\ F_x \in \mathcal{F}_e$, avec $F_x = a^k(\{x\})$, on a :*

$$l(F_x) = F_x - a^{k-1}(\{x\})$$

Dans le cas où $\forall\ x_i \in E$ on a : $F_{x_1} = F_{x_2} = ... = F_{x_n}$ avec $i \in [1, Card(E)]$, on définit la lisière comme suit :

$$l(F_{x_i}) = \bigcap_{x_i \in E} (F_{x_i} - a^{k-1}(\{x_i\}))$$

On calcule ainsi la pseudo-distance des éléments du fermé jusqu'au bord de celui-ci, ou dans le cas où il n'y aurait pas de pas de bord, jusqu'à la lisière de ce dernier. Ce qui nous donne la définition suivante :

Définition 4.8 *Soit E un espace prétopologique, $\mathcal{P}(E)$ la famille de ses parties, et soit D_b une B-distance de $\mathcal{P}(E)$ dans $\mathcal{P}(E)$. On appelle ϵ-centre d'un fermé élémentaire $F_x \in \mathcal{F}_e$ où ϵ est tel que $0 \leq \epsilon < sup_{y \in F_x} D_b(y, F_x)$ tout point $\hat{y}$ de F_x tel que :*
Si $F_x \neq E$ on a :

$$D_b(\hat{y}, F_x) \geq sup_{y \in F_x} D_b(y, F_x) - \epsilon$$

Si $F_x = E$, on a :

$$D_l(\hat{y}, F_x) \geq sup_{y \in F_x} D_l(y, F_x) - \epsilon$$

$$où\ D_l(x, F_x) = \Delta(x, l(F_x))$$

On appellera B-distance la pseudo-distance d'un élément d'un fermé, peu importe que celle-ci mesure la distance jusqu'au bord ou jusqu'à la lisière. Grâce à ces définitions, il est possible de concevoir un algorithme permettant de trouver les centres d'un espace prétopologique E, en regroupant les centres trouvés pour chaque fermé élémentaire. Avant tout chose, il faut déjà établir un algorithme de calcul des ϵ-centres pour un fermé élémentaire, dont voici le pseudo-code :

Calcul des ϵ-centres d'un fermé élémentaire

Méthode : **centres**(F_x)
Variables :
F_x : fermé élémentaire du singleton x
B : ensemble représentant le bord ou la lisière de F_x
ldist : liste des distances des singletons de F_x à B
Début
  Si $F_x \neq E$ **Alors**
    $B \leftarrow b(F_x)$
  Sinon
    $B \leftarrow l(F_x)$
  Fin Si
  Pour i de 0 à $Card(F_x)$ **Faire**
    Ajouter à ldist $\Delta(x_i, B)$
  Fin Pour
  Retourner ldist
Fin

La figure 4.5 donne une illustration de l'algorithme. Dans cet exemple, {h} est le *0-centre*, {f}, {a}, {i} et {k} sont des *1-centres*, {l}, {j}, {e} et {b} sont des *2-centres*, {c}, {g} et {m} sont des *3-centres*, {d} et {n} sont des *4-centres* et {o} qui est un *5-centre* correspond également au bord du fermé dans ce cas. L'élément {p} n'appartient à aucune catégorie de centres car il n'est pas contenu dans le fermé.

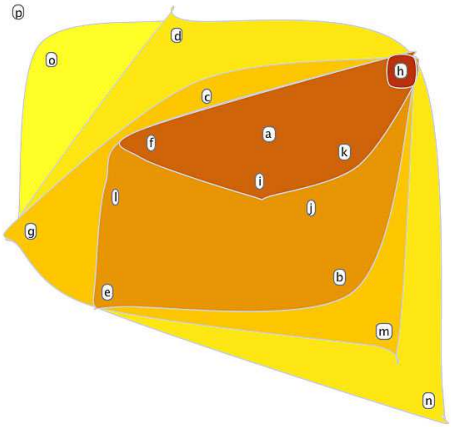


FIG. 4.5 – ϵ-centres d'un fermé dans un espace prétopologique

Dans cet article nous avons élaboré deux algorithmes basés sur les ϵ-centres.

Algorithme exhaustif de calcul des B-distances

La première méthode utilisée pour trouver les centres d'un espace prétopologique est la suivante :

- Calculer les fermés élémentaires de E.
- Pour chaque fermé élémentaire, calculer son bord (ou sa lisière si le fermé est égal à E), puis calculer la B-distance de chaque singleton le composant. Pour chaque singleton, si cette B-distance est plus grande que la B-distance qui lui était précédemment assignée (les éléments de E peuvent appartenir à plusieurs fermés), on la remplace par cette valeur plus élevée.
- Classer les éléments selon leur B-distance.

Voici le pseudo-code de la méthode décrite ci-dessus :

Les résultats de cet algorithme donne donc les éléments centres de l'espace prétopologique, des plus centraux aux plus éloignés. On peut voir un exemple d'ϵ-centres obtenu avec la librairie *PretopoLib* [58] sur la figure 4.6 dont voici la signification des couleurs : plus le groupe est rouge plus il est loin de la frontière de l'espace, et plus il s'éclaircit, plus il est proche de la frontière de l'espace. On peut remarquer ici la différence avec la topologie, où celle-ci se place plutôt dans $\mathbb{R}^n$, ainsi que la localisation des centres qui diffère complètement de ce que l'on aurait pu attendre en topologie (revoir Fig. 4.4).

Habituellement, on structure l'espace en se basant sur les fermés minimaux ([48], [53]) : les fermés minimaux sont des fermés élémentaires minimaux au sens de l'inclusion, c'est-à-dire qu'un fermé élémentaire est minimal s'il ne contient aucun autre fermé que lui-même.

Méthode n°1 du calcul des ϵ-centres d'un espace prétopologique E

Méthode : **e-centres1**(E)
Variables :
$\mathcal{F}_e$: famille des fermés élémentaires
e-centres : liste des singletons associés à leur B-distance
Début
  Pour i de 0 à $Card(\mathcal{F}_e)$ **Faire**
    Calculer centres(F_x)
    Actualiser la liste e-centres avec les nouvelles B-distances (si supérieures)
  Fin Pour
  Classer les singletons de e-centres selon leur B-distance
Fin

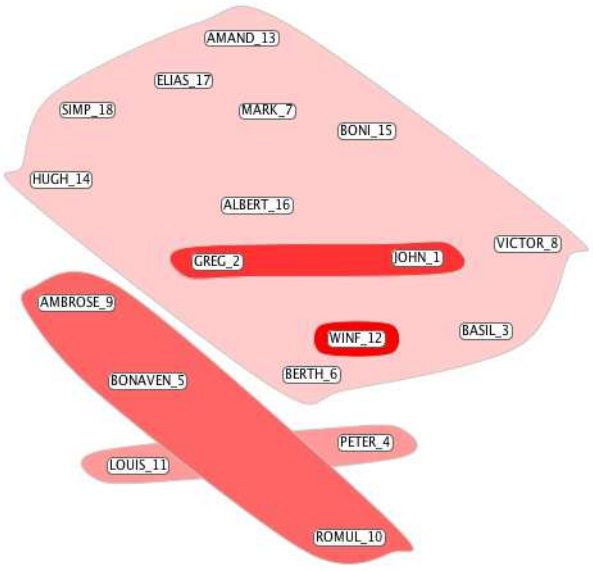


FIG. 4.6 – ϵ-centres d'un espace prétopologique

Ces fermés sont particulièrement intéressants dans le contexte de la structuration dans la mesure où ils permettent de caractériser des parties homogènes de E dans lesquelles il n'existe pas d'élément de $E - F$ appartenant à l'adhérence de F. Ainsi, on peut dire aussi, de façon plus intuitive, qu'une partie fermée F de E contient des éléments entre lesquels il existe des liaisons significatives vis-à-vis de l'adhérence et tels qu'il n'existe pas de liens significatifs entre ces éléments et ceux qui sont en dehors de F.
Notre approche des ϵ-centres étant sensiblement différente, ce que l'on cherche à obtenir n'est pas une structuration au sens des fermés minimaux qui représente les «germes » de l'espace, mais plutôt une structuration au sens de la centralité. On peut observer Fig. 4.7 le même espace que précédemment illustré avec ses fermés minimaux en gris sur l'image de gauche et une superposition de ceux-ci avec les ϵ-centres sur l'image de droite, donnant des résultats dissemblables.

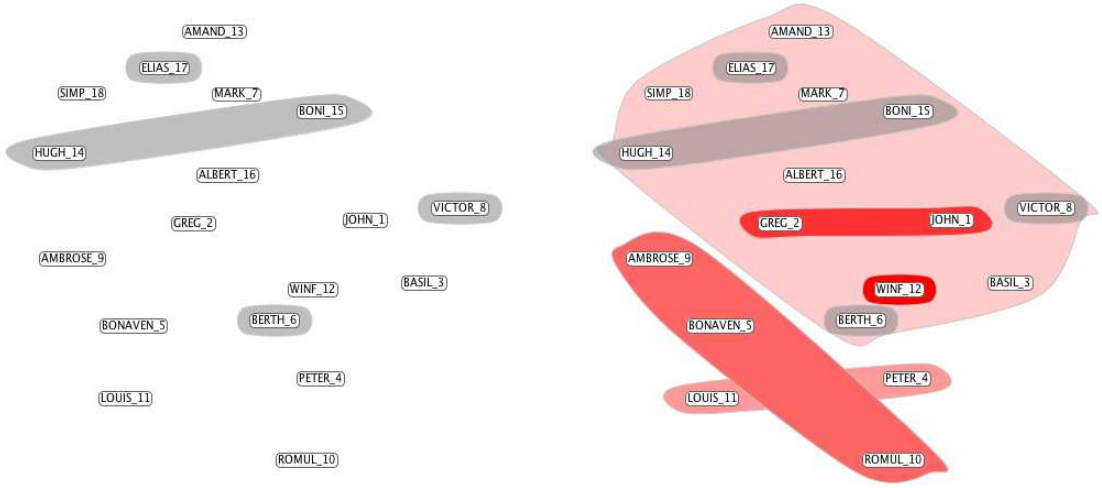


FIG. 4.7 – Comparaison entre $\mathcal{F}_m$ et ϵ-centres

Algorithme d'érosions successives des fermés

La deuxième méthode utilisée pour trouver les centres d'un espace prétopologique est la suivante :

- Calculer les fermés élémentaires de E.
- Pour chaque fermé élémentaire, calculer son bord (ou sa lisière), puis soustraire celle-ci du fermé en question. Seul le cas de la lisière est particulier, soustraire le bord revient à calculer l'intérieur du fermé.
- Calculer les intérieurs successifs pour chaque fermé jusqu'à l'ouvert.
- Classer les éléments selon le nombre d'intérieurs nécessaires pour les exclure du fermé (B-distance inversée).

Voici le pseudo-code de la méthode décrite ci-dessus :

Méthode n°2 du calcul des ϵ-centres d'un espace prétopologique E

Méthode : **e-centres2**(E)
Variables :
$\mathcal{F}_e$: famille des fermés élémentaires
B : ensemble représentant le bord ou la lisière de F_x
e-centres : liste des singletons associés à leur B-distance inversée
d : entier
Début
  Pour i de 0 à $Card(\mathcal{F}_e)$ **Faire**
    $d \leftarrow 0$
    Si $F_x \neq E$ **Alors**
      $B \leftarrow b(B)$
    Sinon
      $B \leftarrow l(B)$
    Fin Si
    Assigner dans la liste e-centres les valeurs des B-distances aux éléments de B à d (si non renseignées)
    Tant que $F_x \neq O(F_x)$ **Faire**
      $d \leftarrow d + 1$
      $B \leftarrow b(F_x)$
      Actualiser dans la liste e-centres les valeurs des B-distances des éléments de B à d (si d supérieur)
      $F_x \leftarrow i(F_x)$
    Fin Tant que
  Fin Pour
  Classer les singletons de e-centres selon leur B-distance inversée
Fin

Cette méthode ressemble à la première méthode à la différence près que celle-ci se base plutôt une distance calculée grâce à l'intérieur qu'avec l'adhérence, ce qui peut avoir son importance si a et i ne sont pas duales. Si la lisière n'est pas utilisée, dans la méthode 2, celle-ci donnera alors le même résultat que la méthode 1. Selon les cas, on peut utiliser les deux méthodes dans le cas où l'une de donnerait pas de résultats assez fins, comme on peut le remarquer sur la figure 4.8.

4.4 Conclusion

Tout au long de ce chapitre, nous avons axé notre réflexion sur la localisation des éléments centraux d'un espace prétopologique. Les algorithmes ainsi développés présentent

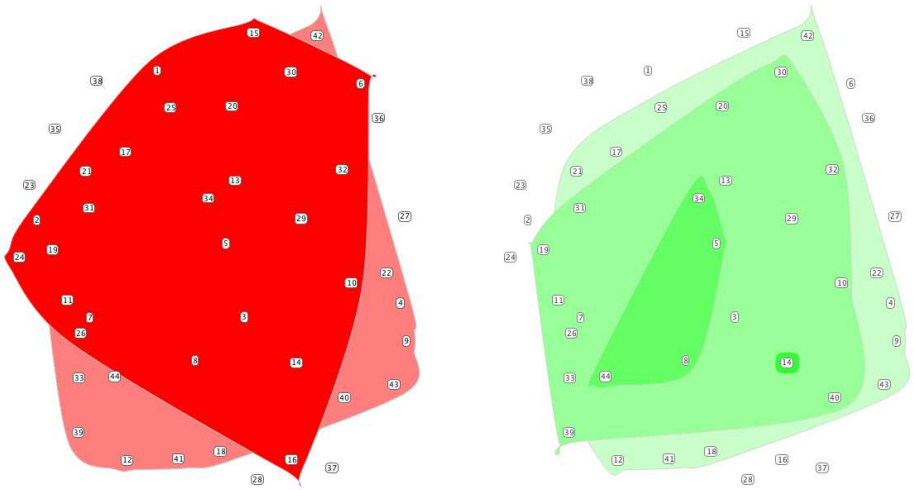


FIG. 4.8 – Cas où la méthode 2 (à droite) est plus fine que la méthode 1 (à gauche)

des résultats différents des algorithmes de structuration ou de classification existants actuellement (que ce soit en prétopologie ou non).
Tout d'abord, on observe la chose suivante dans nos expérimentations : des éléments qui habituellement seraient désignés comme ex-centrés, se trouvent être souvent présents dans les 0-centres (ou dans une classe centre proche des 0-centres). De prime abord, cela peut paraître étrange, mais c'est là tout l'intérêt de la méthode : il est tout à fait logique de trouver des éléments «difficiles » à atteindre comme étant des centres car ils se trouvent éloignés de la frontière de l'espace.
De ce fait, ce type de structuration par les ϵ-centres peut être utilisée dans des applications diverses, comme par exemple, déterminer dans une population donnée la propagation d'un virus, les 0-centres étant les individus atteints en dernier, ou à l'inverse, sur une population contaminée, être capable de déterminer l'origine de l'épidémie.
De nombreuses autres applications peuvent découler de cette méthode qui apporte à notre sens une nouvelle caractérisation des réseaux complexes.

Chapitre 5

PretopoLib : la librairie JAVA basée sur la prétopologie

Dans ce chapitre, nous présentons une contribution essentielle sous la forme d'une librairie logicielle implémentant les concepts de la prétopologie. Son intérêt en tant que librairie informatique générique permet d'effectuer des calculs sur des espaces prétopologiques ainsi que leurs visualisations avec toutes les opérations s'y appliquant, amenant les bases pour la simulation des réseaux complexes et leur dynamique, ainsi que toute application impliquant la théorie de la prétopologie (théorie des jeux, imagerie, ...). Nous expliquerons ici quelle est la structure du programme en présentant les grandes classes le composant, puis nous nous pencherons sur le problème des performances, comparant les résultats pratiques aux résultats théoriques. Nous finirons par une série d'exemples illustrant son apport et ses potentialités.

5.1 Introduction

PretopoLib est un package JAVA créé pour pouvoir utiliser toutes les opérations évoquées en prétopologie, que ce soit adhérence ou fermeture mais aussi les opérations de base sur les ensembles comme l'intersection et la réunion, et de rendre possible la visualisation de tels processus. Cette librairie est scindée en deux parties :

- la partie données et traitement permettant la manipulation d'ensembles
- la partie visualisation basée sur *Prefuse* permettant de représenter graphiquement ces mêmes ensembles

Plusieurs classes composent la librairie comme *PretopoSpace* qui définit l'espace prétopologique, lui-même composé de *PretopoNodes* contenant les éléments de E et *PretopoEdges* contenant les relations entre éléments. Toutes ces classes sont basées sur les classes standards *java.util.HashSet* et *java.util.HashMap* permettant de travailler sur les ensembles et possédant par défaut les opérations intersection et réunion.
La partie concernée par la visualisation se base sur la librairie *Prefuse* encore en ver-

sion bêta mais fournissant un très bon rendu. Habituellement utilisée pour représenter les graphes, elle permet également de représenter des polygones convexes dont nous nous servons pour représenter les ensembles (voir Fig. 5.1).
Dans ce chapitre, nous allons présenter la structure et les fonctionnalités de cette librairie composée d'environ 3400 lignes de code dont la conception s'est étalée sur plus d'un an. Celle-ci a d'ailleurs fait l'objet d'un dépôt logiciel.

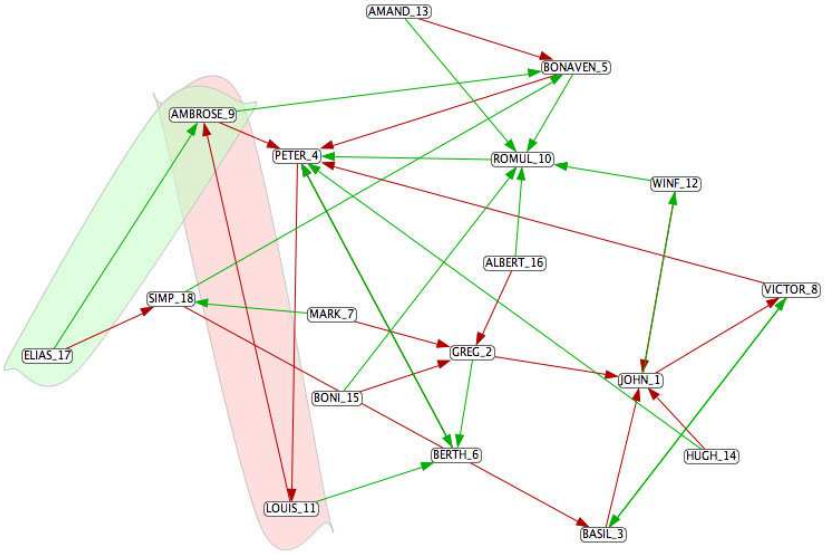


FIG. 5.1 – Exemple de visualisation avec *PretopoLib*

5.2 Structure de données de *PretopoLib*

Comme précisé dans l'introduction, *PretopoLib* est écrit en langage JAVA, lequel utilise une machine virtuelle pour exécuter un programme, nous n'avons donc pas à recompiler les sources pour passer d'un système d'exploitation à un autre (par exemple de Windows vers Linux ou MacOS). JAVA fournit une abstraction des données, un type de programmation orienté objet, la surcharge d'opérateurs, et bien d'autres fonctionnalités permettant de développer des structures de données. Pour plus de détails concernant le langage JAVA, voir la documentation de Sun[1] ou un ouvrage sur le sujet (par exemple [30]).
L'idée la plus importante derrière cette librairie, est sa disponibilité sous forme de « package » permettant son utilisation et sa réutilisation par divers utilisateurs. Evidemment, cela nécessite de spécifier les propriétés du type de données indépendamment de l'implémentation [37, 50], ce qui a été fait dans le chapitre 2.3.2. Nos structures de données se basent principalement sur la classe *HashMap* qui est une table de hachage et dont la fonction de hachage *hashCode()* de la classe Object, peut être utilisée pour tous les objets, et peut être redéfinie dans n'importe quelle classe.
PretopoLib est composé de deux sous-packages : *PretopoNotions* concerne la partie calcul

[1] http ://www.sun.com

et *PretopoVisual* concerne la partie visualisation. Un troisième est en cours de développement et sera composé d'outils utiles au programmeur désireux d'utiliser la bibliothèque.
Les trois grandes classes de PretopoNotions sont :

- **PretopoNodes** qui représente un ensemble composé de singletons.
- **PretopoEdges** qui contient toutes les relations dont l'espace est composé.
- **PretopoSpace** qui représente l'espace prétopologique avec E représenté par un objet PretopoNodes et les relations de l'espace par un objet PretopoEdges.

Ces trois classes principales sont détaillées ci-après.

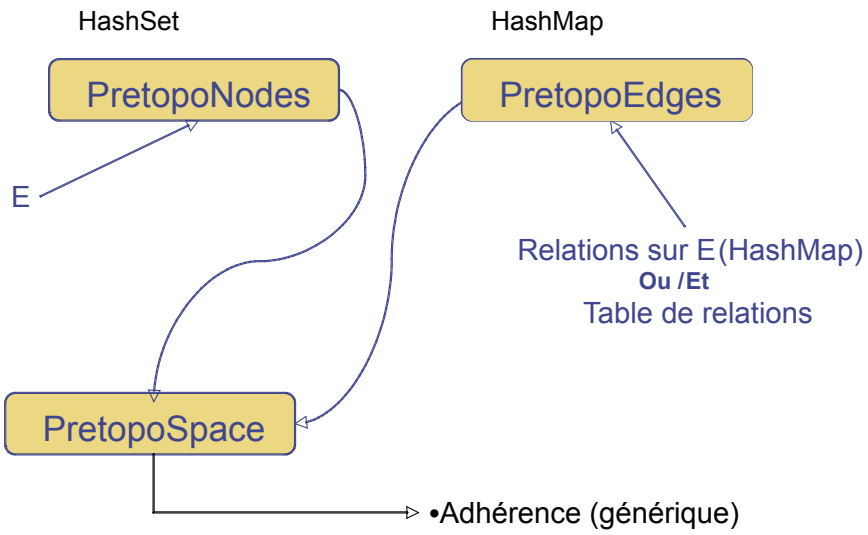


FIG. 5.2 – Structure simplifiée de *PretopoLib*

5.2.1 PretopoNodes

Description

Un objet PretopoNodes représente un ensemble d'éléments d'un espace prétopologique. Chaque élément de l'ensemble est unique, dont l'identifiant se rapporte au nom du dit élément (qu'on nommera son label), qui est utilisé pour générer les clés de la table de hachage. Le listing 5.1 montre une illustration simple d'utilisation de l'objet PretopoNodes.

Listing 5.1 – Création de deux ensembles de 3 et 4 éléments chacun, puis intersection de ceux-ci

```
1 import PretopoNotions.Elmt;
2 import PretopoNotions.PretopoNodes;
3
4 public class PretopoLibSample {
```

```
5
6       public static void main(String[] args)
7       {
8
9           //création des éléments
10          Elmt elt_a=new Elmt("a");
11          Elmt elt_b=new Elmt("b");
12          Elmt elt_c=new Elmt("c");
13          Elmt elt_d=new Elmt("d");
14          Elmt elt_e=new Elmt("e");
15
16          PretopoNodes A=new PretopoNodes();
17          PretopoNodes B=new PretopoNodes();
18          PretopoNodes C=new PretopoNodes();
19
20          //ajout des singletons dans les ensembles
21          A.addElmt(elt_a);
22          A.addElmt(elt_b);
23          A.addElmt(elt_c);
24
25          B.addElmt(elt_b);
26          B.addElmt(elt_c);
27          B.addElmt(elt_d);
28          B.addElmt(elt_e);
29
30          C=A.intersection(B);
31      }
32  }
```

Le programme créé deux ensembles A et B contenant divers éléments, et stocke le résultat de leur intersection dans l'ensemble C. Entre la ligne (10) et (14), on créé les éléments avec un label unique ; les lignes (16) à (18) créent les ensembles A, B, et C, puis les instructions ligne (21) à (28) insèrent les éléments dans A et B. En ligne (30), on fait ainsi l'intersection de A et B que l'on stocke dans C.

Declaration de l'objet PretopoNodes

Les constructeurs de PretopoNodes sont :
`public PretopoNodes()` créé un ensemble vide.
`public PretopoNodes(Elmt elt)` créé un singleton (ensemble contenant un seul élément)
`public PretopoNodes(PretopoNodes pn)` créé un ensemble par copie.

Méthodes de l'objet PretopoNodes

On retrouve ici les méthodes usuelles :
`boolean contains(Object o)` teste si un élément est contenu dans l'ensemble.
`boolean equals(Object o)` teste si deux ensembles contiennent les mêmes éléments.

`void addElmt(Elmt elmt)` ajoute un élément à l'ensemble (si celui-ci ne s'y trouve pas déjà).

`boolean intersects(PretopoNodes pn)` teste si l'ensemble intersecte un ensemble donné.

`PretopoNodes intersection(PretopoNodes pn)` donne le résultat de l'opération intersection entre deux ensembles.

`PretopoNodes union(PretopoNodes pn)` donne le résultat de l'opération union entre deux ensembles.

`PretopoNodes substract(PretopoNodes pn)` donne le résultat de l'opération soustraction entre deux ensembles.

`boolean isSubset(PretopoNodes pn)` teste si un ensemble est un sous-ensemble d'un ensemble donné.

Coûts des opérations

Notre implémentation des objets PretopoNodes type ensembliste se base sur les tables de hachage et plus précisément la classe *HashSet*. Celle-ci stocke les informations directement sous forme de clé, autrement dit, la donnée correspond à la clé. La classe *HashSet* est basée sur la classe *HashMap* laquelle nous utiliserons dans d'autres cas. Comme expliqué dans le chapitre 2.3.2, le facteur de charge doit être compris entre 0 et 1 si on veut garder une complexité des opérations en $O(1)$ (on ne s'intéresse qu'à la borne supérieure), qui est en fait une complexité amortie comme expliqué plus bas. Un facteur de charge de 0 correspond à une table vide, 0.5 correspond à une table à moitié pleine, etc. Une table faiblement chargée aura peu de collisions et sera donc optimale pour les insertions et les recherches (mais ralentira le processus de parcours avec un itérateur). $HashMap$ et $HashSet$ proposent des constructeurs qui permettent de spécifier un facteur de charge, ce qui veut dire que lorsque ce facteur de charge est atteint le conteneur augmentera automatiquement sa capacité (le nombre de seaux, bucket en anglais) en la doublant d'un coup (spécifique à l'implémentation de la classe $HashMap$ de Sun), et redistribuera les objets existants dans le nouvel ensemble de seaux (c'est ce qu'on appelle le rehachage). Cette opération coûteuse qui n'arrive que peu souvent nous permet donc de travailler avec une complexité amortie en $O(1)$.
D'après l'auteur Joshua Bloch ayant écrit la classe, le facteur de charge par défaut utilisé par $HashMap$ est 0.75 (il ne se rehache pas avant que la table ne soit aux trois-quarts pleine). Cette valeur est un bon compromis entre les performances et le coût en espace. Un facteur de charge plus élevé réduit l'espace requis par une table mais augmente le coût d'une recherche, ce qui est à surveiller car les recherches sont les opérations les plus courantes. Si un objet $HashMap$ est destiné à recevoir beaucoup d'entrées, le créer avec une grosse capacité initiale permettra d'éviter le surcoût du rehachage automatique. Pour nos simulations, les performances de cette classe sont acceptables. Si ces performances venaient à décroître, nous nous pencherions sur une implémentation plus efficace des tables de hachage. On peut voir le coûts des opérations effectuées par $PretopoNodes$ dans le tableau 5.2.1.

Méthode	Coût
contains	O(1)
equals	O(n)
addElmt	O(1)
intersects	O(n)
intersection	O(n)
union	O(n)
substract	O(n)
isSubset	O(n)

TAB. 5.1 – Coûts des opérations de l'objet *PretopoNodes*

5.2.2 PretopoEdges

Description

Cette classe représente les différentes relations composant un espace prétopologique. Cet objet contient une table de relations, qui peuvent être de différents types (métriques, valuées ou binaires). La table de relations est stockée sous la forme d'une table de hachage, utilisant la classe *HashMap*. Un objet de type *Relation* définit tous les liens existants entre les éléments de l'espace et surtout, il définit la *fonction de voisinage* utilisée pour construire l'application *adhérence*. Dans le listing 5.2, en se rapportant aux variables précédemment utilisées dans 5.1, nous construisons tout d'abord une relation binaire, puis une relation valuée et enfin une relation métrique :

Listing 5.2 – Création d'une relation binaire, valuée et métrique.

```
1 //création d'une relation binaire
2 Relation r= new Relation("binary");
3 r.addLink(elt_c, elt_d);
4 r.addLink(elt_e, elt_d);
5 r.addSymLink(elt_c, elt_a);
6 r.addLink(elt_b, elt_a);
7
8 //création d'une relation valuée
9 RelationV rv= new RelationV("valued");
10 rv.addSymLink(elt_a, elt_e,2);
11 rv.addSymLink(elt_b, elt_e,1);
12 rv.addSymLink(elt_b, elt_d,4);
13
14 //création d'une relation métrique
15 RelationB rb= new RelationB("metric");
```

A la ligne (2), une relation binaire est créée, laquelle est la classe mère de toute relation. De la ligne (3) à la ligne (6), des liens sont ajoutés entre les éléments, soit de façon orientée `addLink( source , target )` , soit de façon non-orientée `addSymLink( source , target )` . A la ligne (9), une relation valuée est créée, et de la ligne (10) à la ligne (12), des liens valués lui sont ajoutés. A la ligne (15), une relation métrique est créée. Notons qu'il n'y a

Type de méthode	Coût de la fonction de voisinage
Table de correspondance	O(1)
Liste d'adjacence	O(n)

TAB. 5.2 – Coûts du test de voisinage d'un singleton de E par rapport à un sous-ensemble de E

pas de liens utilisés pour la dernière relation, celle-ci utilisant directement les coordonnées des singletons de l'espace pour connaître leur distance par rapport aux autres (concept de boule).
La fonction de voisinage est implémentée dans la classe *Relation*. Le listing 5.3 montre comment la fonction $R(x) = x \in E, xRy$ est codée.

Listing 5.3 – Fonction de voisinage de la classe *Relation*

```
1 public boolean NeighbFunction(Elmt elt, PretopoNodes A)
2     {
3         boolean b=false;
4
5         HashMap<Elmt,Object> l=(HashMap<Elmt,Object>) rel_list.get(elt);
6         if(l!=null)
7         {
8             Iterator<Elmt> itr=l.keySet().iterator();
9             while(itr.hasNext() && !b)
10                if(A.contains(itr.next()))
11                    b=true;
12        }
13        return b;
14    }
```

L'algorithme examine si l'élément *elt* est ou non voisin de l'ensemble A. Dans ce cas, où la relation est de type binaire, on vérifie simplement si *elt* possède un lien qui pointe sur un élément de A. Si c'est le cas, on arrête la boucle de recherche et on renvoie *vrai*. Si ce n'est pas le cas, l'algorithme finit sa recherche quand tous les voisins de *elt* sont comparés à A puis il renvoie $faux$.

Coûts des opérations

Concernant la fonction de voisinage, si on travaille dans un espace dans lequel on utilise une table de correspondance, on sera en $O(1)$, la recherche des voisins étant immédiate. Si on doit vérifier les liens entre singletons (principalement type $\mathcal{V}_s$), on passe en $O(n)$ car pour un élément, on parcourt tous ses voisins de manière à vérifier si l'un d'entre eux appartient à l'ensemble sur lequel on travaille, celui possède donc au plus n voisins.

Remarque : dans le cas d'un espace métrique, par défaut le coût de la fonction de voisinage est supérieur à $O(n)$, car on doit pour chaque paire « singleton à tester, singleton de A » calculer la distance euclidienne. Si on prend k le coût de cette distance, on se place

en $O(kn)$, k étant linéaire. Pour éviter ce surcoût, la librairie propose une fonction qui transforme l'espace métrique en espace valué : cela calcule toutes les distances entre les éléments de E, prenant forcément du temps (de l'ordre de $O(n^3)$ car cela correspond à la recherche des plus courts chemins) mais permettant ensuite de repasser le coût de la fonction de voisinage d'un tel espace en $O(n)$.

5.2.3 PretopoSpace

Description

Cette classe représente l'espace prétopologique lui-même. Il est composé d'un objet *PretopoNodes* correspondant à E et d'un objet *PretopoEdges* regroupant toutes les relations et fonctions de voisinage de l'espace. C'est dans l'objet *PretopoSpace* que se trouve la méthode adhérence (implémentée *pseudoclosure*, voir listing 5.4), ainsi que d'autres méthodes basées sur l'adhérence, comme la fermeture par exemple.

Listing 5.4 – Extrait de code de l'adhérence pour les espaces valués.

```
1 public PretopoNodes pseudoclosure(PretopoNodes set, String rel_label, Object param
2  {
3
4 PretopoNodes pseudoCset=new PretopoNodes(set);
5 PretopoNodes E=new PretopoNodes(Nodes);
6 E=E.substract(set);
7
8           ...
9
10 if(param.getClass().equals(Double.class) || param.getClass().equals(Integer.class)
11               {
12                    RelationV r=(RelationV) Relations.getRelation(rel_label);
13                    if(!r.getuseTR())
14                    {
15                         Iterator<Elmt> itr=(Iterator<Elmt>) E.getNodeSet().iterator()
16                         while(itr.hasNext())
17                         {
18                              Elmt elt=itr.next();
19                           if(r.NeighbFunction(elt,set,(Double)param))
20                             pseudoCset.addElmt(elt);
21                         }
22                    }
23                    else
24                         pseudoCset=r.getTableRelation(set);
25               }
26          ...
27
28 return pseudoCset;
29 }
```

Dans le listing 5.4, on peut constater que la surcharge de la méthode *pseudoclosure* peut être faite de manière assez flexible. Celle-ci prend en entrée trois paramètres : l'ensemble sur lequel on applique l'adhérence, le nom de la relation, et un paramètre qui dépend de la nature de l'espace dans lequel on travaille utilisé par la fonction de voisinage. Dans le code précédent, seule la partie concernant les espaces valués est montrée. On extrait tout d'abord la relation passée en paramètre ligne (12), puis on recherche dans E les voisins de l'ensemble sur lequel on applique l'adhérence (lignes (15) à (21)). Pour finir, l'ensemble résultat de l'adhérence est retourné ligne (28).

5.2.4 Coûts théoriques de l'adhérence

Calcul du coût d'une adhérence

Concernant l'adhérence, deux méthodes peuvent être utilisées :
– l'adhérence *statique*
– l'adhérence *dynamique*

L'opération élémentaire est ici le temps d'accès à une case de la table où sont stockées les relations. Le cas le plus efficace est sans nul doute la méthode de la table de correspondance : comme l'on n'a pas à parcourir tout l'espace E, par exemple si on fait $a(\{a, b\})$, on aura tout de suite la réponse car dans la table, à $\{a, b\}$ correspond son adhérence, par exemple $\{a, b, d\}$. Les voisins étant déjà connus, on n'utilise pas la fonction de voisinage (ou alors une seule fois pour la construction de la table), et cela nous donne donc un coût d'adhérence d'une complexité amortie de l'ordre de $O(1) \times$ *temps d'accès mémoire*, identique à celui d'une table de hachage. Ce cas décrit une adhérence statique et cela fonctionne très bien avec un espace de type $\mathcal{V}$, si l'on n'a aucune modification de la fonction de voisinage. Cependant, cela peut poser problème avec un espace de type $\mathcal{V}_S$, en temps de calcul nécessaire à la génération de la table de correspondance (identique au coût de la reherche des fermés élémentaires) mais principalement en terme de taille mémoire de l'ordre de n^2 (on parcourt les n éléments de E sur lesquels on fait au plus $n-1$ adhérences, résultats que l'on garde en mémoire) dans le pire des cas, ce qui peut être très grand : par exemple, avec 100 000 éléments chacun représentés sur 16bits, cela nous donne une taille en mémoire pour la table de correspondance d'environ 18 Go ! On a donc tout intérêt à utiliser la fonction d'adhérence dynamique pour les grands espaces.

Ainsi, si l'adhérence dynamique est utilisée pour un ensemble $A/A \subseteq E$, il faut tester tous les éléments de $E - A$ avec la fonction de voisinage pour savoir quels sont les voisins de A. Nous avions évoqué le fait que si on utilisait une structure type liste d'adjacence dans la classe *Relation*, on aurait une complexité de la fonction de voisinage en $O(n)$. Cette fonction de voisinage est dynamique, car elle peut changer au cours du temps. L'optimisation de la simulation dépendra donc du cas étudié, si le modèle utilisé nécessite une fonction de voisinage dynamique ou statique. Dans le tableau 5.2.4, on peut lire un récapitulatif des coûts des adhérences dans le pire des cas utilisées par *PretopoLib*.

Espace	**Coût d'une adhérence**	
	Statique	Dynamique
Binaire	$O(1) \times$ *temps d'accès mémoire*	$O(n^2) \times$ *temps d'accès mémoire*
Valué	$O(1) \times$ *temps d'accès mémoire*	$O(n^2) \times$ *temps d'accès mémoire*
Métrique	$O(1) \times$ *temps d'accès mémoire*	$O(n^3) \times$ *temps d'accès mémoire*

Tab. 5.3 – Coûts d'une adhérence dans le pire cas avec *PretopoLib*

Méthode	**Coût**
Fermés élémentaires	$O(n^2)$
Fermés minimaux	$O(n^2)$

Tab. 5.4 – Coûts des fermés prétopologiques dans le pire cas avec *PretopoLib*

Calcul du coût des algorithmes prétopologiques dans le pire cas

Nous avons examiné la fonction adhérence, car elle est l'opération de base pour tous les autres algorithmes que l'on utilise habituellement en prétopologie. L'intérieur (même si ce n'est pas toujours le cas), la fermeture et donc les algorithmes de classifications ou de structuration se basent sur celle-ci. Il paraît donc judicieux de calculer la complexité de ces algorithmes en la prenant comme opération élémentaire.

Nous avions déjà étudié la complexité des algorithmes liés à la prétopologie nous pouvons donc dresser un tableau (5.2.4) des complexités associées à ces diverses méthodes, en lisant $O(n)$ par exemple comme $O(n) \times$ *coût d'une adhérence* sauf pour les fermés minimaux dont le calcul final est indépendant de l'adhérence.

5.2.5 Coûts empiriques de l'adhérence

La section précédente nous permet de fixer une borne supérieure de la complexité de l'exécution des algorithmes prétopologiques laquelle on est certain de ne pas dépasser. En pratique, on se rend compte que l'on est bien loin du pire cas, bien heureusement. Dans le tableau 5.2.5, on peut comparer les coûts théoriques et pratique[2] d'une adhérence (avec comme coût en temps de l'accès mémoire une durée de $1\mu s$) ; là où en théorie, l'adhérence coûte dans le pire des cas $O(n^2) \times$ *temps d'accès mémoire*, on se rend compte qu'en pra-

[2]coût moyen sur une centaine d'adhérences effectuées sur des éléments de E choisis aléatoirement.

Méthode	**Coût**
Pseudo-distance	$O(n)$
Recherche des centres	$O(n^3)$
MCPR	$O(k \times n^3)$
MCP	$O(n^5)$
Méthode d'encadrement	$O(n^2)$

Tab. 5.5 – Coûts des méthodes MCP, MCPR et Encradrement dans le pire cas avec *PretopoLib*

tique, on approche plûtot $O(n.log(n)) \times$ *temps d'accès mémoire*, donc que l'adhérence posséderait une complexité en temps plutôt quasi-linéaire que quadratique. Cela est dû principalement à la distribution des liens entre les éléments (voir Fig. 5.3). Le pire qui pourrait arriver serait que le nombres de liens atteigne le maximum possible, soit un total de n^2 liens. Dans les réseaux sur lesquels les essais ont été menés, ce nombre de liens est fortement réduit, alors que ces réseaux font partie de la classe des réseaux réels qui nous intéressent tant, puisqu'il s'agit par exemple de communautés sociales, de réseaux d'interactions de protéines ou encore de relations entre pages web d'un même campus universitaire. Même pour le plus grand que l'on ait testé (137 942 éléments et 1 497 134 liens), le nombre de liens représente à peine $0,008\%$ du nombre de liens envisageables, ce qui explique en grande partie les résultats obtenus lors des expérimentations.

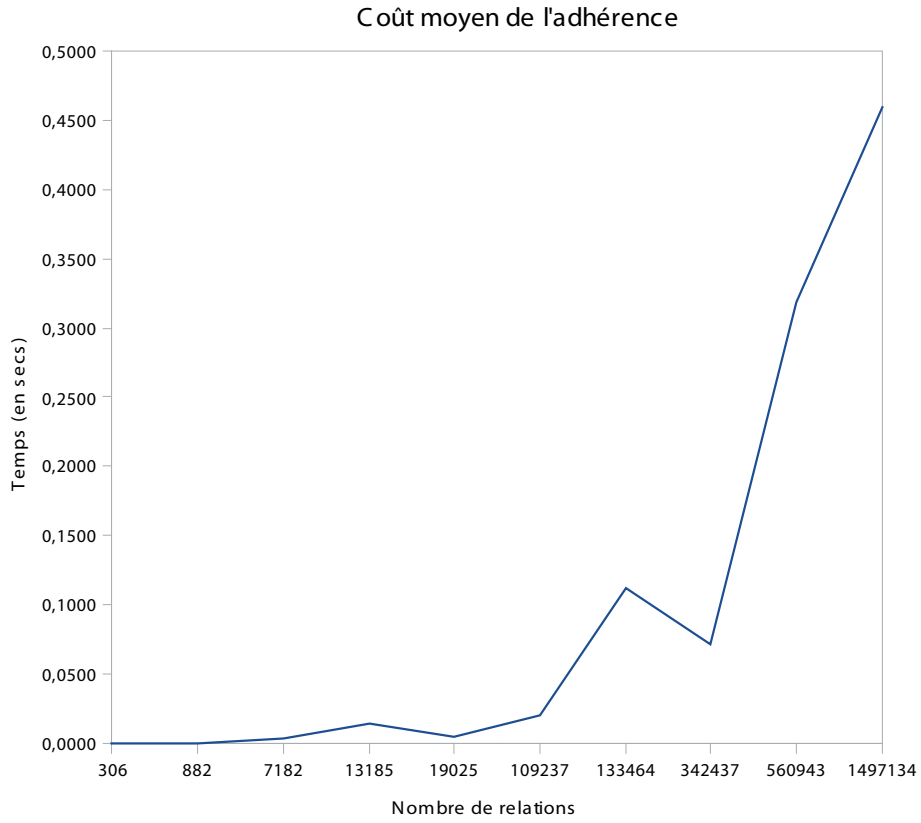


FIG. 5.3 – Performances de l'adhérence en fonction du nombre de relations

5.3 Méthodes et Exemples basés sur *PretopoLib*

Dans cette partie, une liste des méthodes utilisables avec *PretopoLib*, ainsi que quelques exemples illustratifs seront présentés.

Nombre d'éléments	Nombre de liens	Coût théorique (pire cas)	Coût en pratique
44	306	19 ms	2 ms
105	882	11 ms	3 ms
1428	7182	2 s	4 ms
1490	19025	2,22 s	5 ms
4940	13185	24,4 s	14 ms
5019	109237	25 s	20 ms
24541	342437	10 min	72 ms
76815	133464	1h38	112 ms
189076	560943	9h56	319 ms
137942	1497134	5h17	460 ms

TAB. 5.6 – Coûts d'une adhérence en théorie et en pratique

5.3.1 Méthodes

Adhérence, Interieur, Fermeture, Ouverture

Par défaut, l'intérieur, la fermeture et l'ouverture sont basés sur l'adhérence. Si la fonction adhérence est modifiée, les trois fonctions sus-citées le seront également. L'exemple suivant illustre leurs utilisations :

Exemple 5.1 *Exemple : $B = a(A), C = i(A), FA = F(A), OA = O(A)$*

```
B=monespace.pseudoclosure(A, "ma relation", null);
C=monespace.interior(A, "ma relation", null);
FA=monespace.closure(A, "ma relation", null);
OA=monespace.opening(A, "ma relation", null);
```

Fermés élémentaires et minimaux

Ces deux fonctions renvoient une liste d'ensembles avec le résultat escompté. En voici une petite illustration :

Exemple 5.2 *Exemple : liste des fermés élémentaires de E puis liste des fermés minimaux de E.*

```
ArrayList<PretopoNodes> listeFe,listeFm= new ArrayList<PretopoNodes>();
listeFe=monespace.getAllElementaryClosedSubsets(x,"ma relation",null);
listeFm=monespace.getAllMinimalClosedSubsets(listeFe);
```

MCPR, MCP, et Encadrement

Ici on peut comprendre comment se servir des méthodes de classifications étudiées dans [52].

Exemple 5.3 *Exemple : liste de classes formées par MCPR, liste de classes formées par MCP, puis listes de classes formées par encadrement.*

```
ArrayList<PretopoNodes> classesMCPR, classesMCP, classesEn1, classesEn2 = new
ArrayList<PretopoNodes>() ;
classesMCPR=monespace.MCPR("ma relation",null) ;
classesMCP=monespace.MCP("ma relation",null) ;
classesEn1=monespace.CoarsestPartition("ma relation",null) ;
classesEn2=monespace.FinestPartition("ma relation",null) ;
```

5.3.2 Autres méthodes

D'autres méthodes ont été ajoutées dans la librairie : les notions de séparabilités, certains opérations utiles comme le bord utilisées principalement en imagerie et la classification par équivalence.

Séparabilité

Les notions de séparabilité énoncées dans [7] peuvent être utilisées dans la librairie : on peut tester si deux ensembles sont séparés, selon le critère choisi. Les choix de la fonction sont : IS_WEAKLY_SEPARATED, ARE_WEAKLY_SEPARATED, IS_SEPARATED, ARE_SEPARA ARE_STRONGLY_SEPARATED. Si l'on veut savoir si A et B sont fortement séparés, alors voici comment il faut procéder :

Exemple 5.4 *Exemple : test pour savoir si A et B sont fortement séparés.*

```
if(separated(A, B, ARE_STRONGLY_SEPARATED, "ma relation",null))
System.out.println("A et B sont fortement séparés.") ;
```

Transformations morphologiques

Les fonctions comme le bord ou la frontière sont disponibles tout en gardant leurs notations usuelles et s'écrivent de la façon suivante :

Exemple 5.5 *Exemple : B = bord de l'ensemble A ($B = b(A)$)*

```
B=monespace.b(A, "ma relation", param) ;
```

Classes d'équivalence

Les méthodes développées dans [23] sont disponibles dans la librairie ; ainsi, on peut obtenir une classification par équivalence comme ceci :

Exemple 5.6 *Exemple : liste des classes d'équivalence de X*

ArrayList<PretopoNodes> classesEq= new ArrayList<PretopoNodes>();
classesEq=monespace.EquivalenceClasses(X, "ma relation",null);

5.3.3 Exemples d'applications

Réseau d'interactions entre protéines dans la levure [80]

Nombre d'éléments : 1428
Nombre de relations : 5200
Méthode appliquée : MCP avec 215 classes trouvées
Résultats : voir annexes et figure 5.4

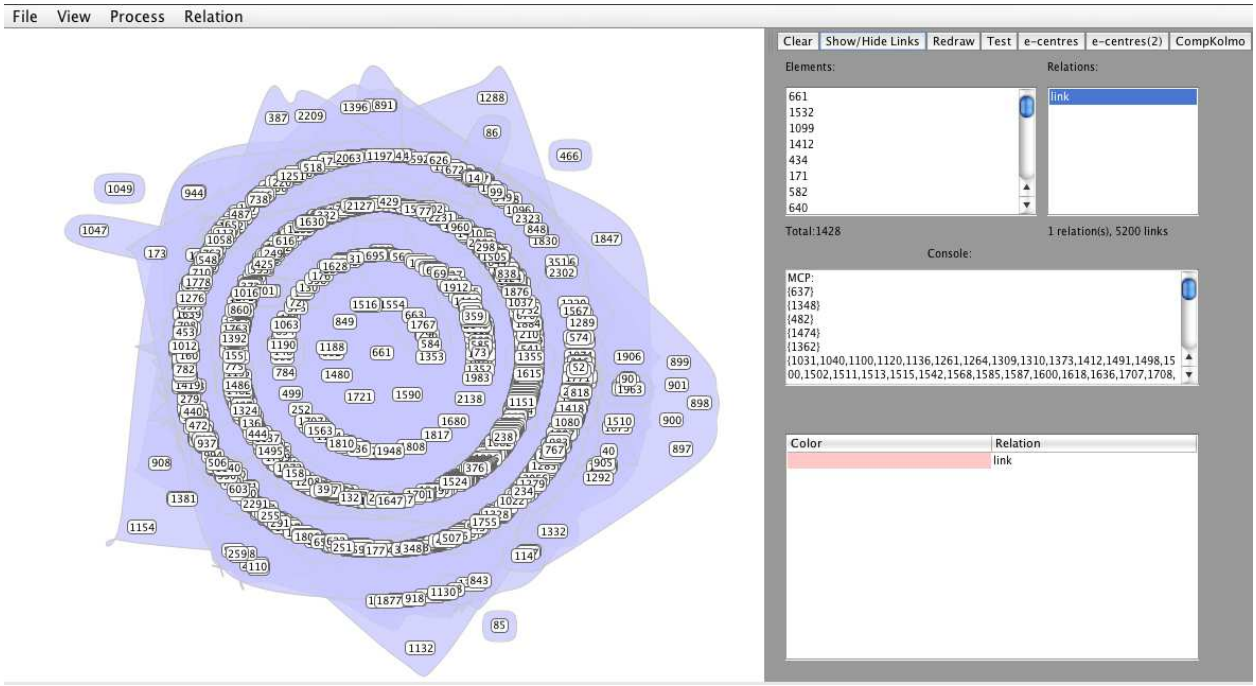


FIG. 5.4 – Capture d'écran de l'application simulant MCP sur un réseau d'interactions entre protéines dans la levure

Réseau de livres sur Amazon.com « le client qui a acheté ce livre a aussi acheté ... » (http ://www.orgnet.com/)

Nombre d'éléments : 105
Nombre de relations : 882
Méthode appliquée : Méthode 1 des centres avec 2 classes trouvées
Résultats : voir annexes et figure 5.5

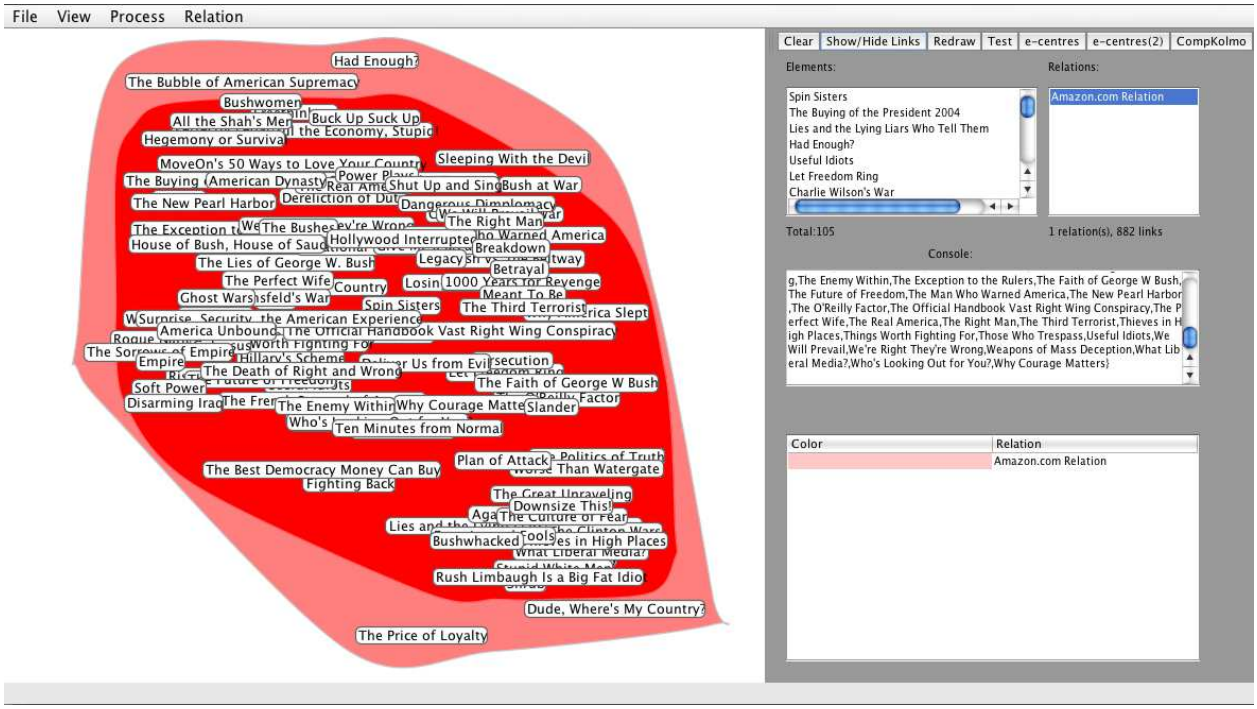


FIG. 5.5 – Capture d'écran de l'application simulant la méthode 1 des ϵ-centres sur un réseau de livres d'Amazon.com

Réseau de « blogs » traitant des élections présidentielles de 2004 aux Etats-Unis d'Amérique [1]

Nombre d'éléments : 1490
Nombre de relations : 19090
Méthode appliquée : Recherche des composantes fortement connexes avec 688 composantes trouvées
Résultats : voir annexes et figure 5.6

Réseau d'interactions sociales d'un groupe de moines dans un monastère [78]

Nombre d'éléments : 18
Nombre de relations : 40
Méthode appliquée : Recherche de classes d'équivalence avec 7 classes trouvées
Résultats : voir annexes et figure 5.7

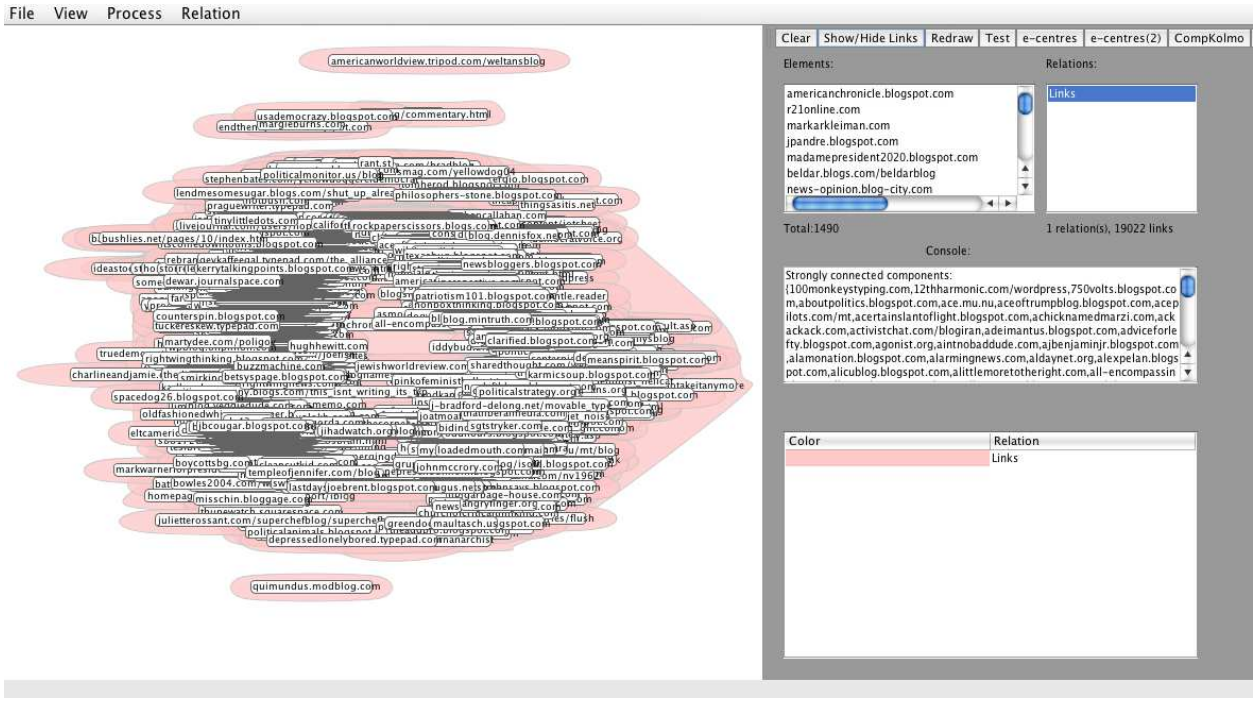


FIG. 5.6 – Capture d'écran de l'application simulant la recherche des composantes fortement connexes sur un réseau de « blogs » traitant des élections présidentielles de 2004 aux Etats-Unis d'Amérique

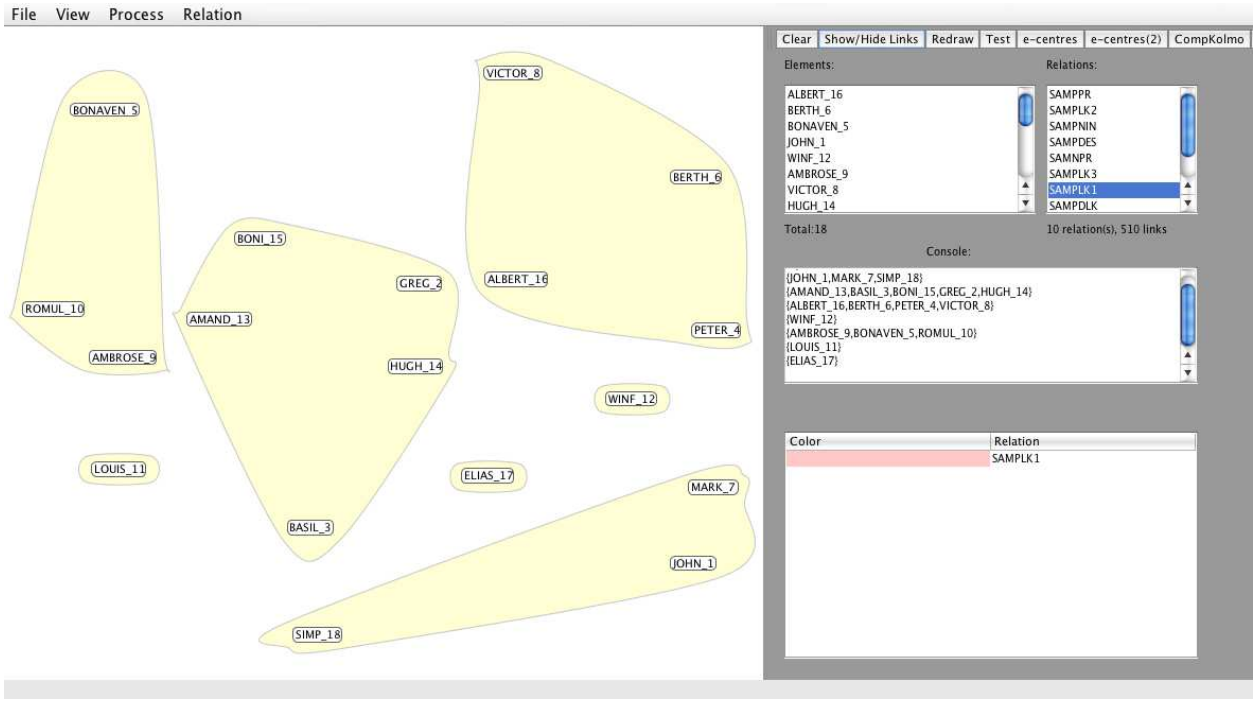


FIG. 5.7 – Capture d'écran de l'application simulant la recherche de classes d'équivalence sur un réseau d'interactions sociales d'un groupe de moines dans un monastère

5.4 Conclusion

Durant ce chapitre, il a été exposé le travail fourni pour le développement d'une librairie permettant la mise en oeuvre de problèmes utilisant la théorie de la prétopologie de manière générale, afin que tous puissent construire des modèles et des simulations, rapidement et efficacement. *PretopoLib* met à disposition des structures de données et des algorithmes permettant de faire tourner une simulation lambda sur une machine de bureau actuelle. Les avantages que nous voyons à cette librairie sont les suivants :

- *séparation entre opérations de calcul et visualisation* : en effet, rien n'empêche d'utiliser un système graphique complètement autonome, que l'on liera à la partie calcul de la librairie (par exemple pour faire de la simulation multi-agents avec MASON ou RePast, deux outils qui possèdent déjà leur propre visualisation).

- *implémentation efficace des structures de données liées à la prétopologie* : même si les concepts prétopologiques peuvent nécessiter un nombre important d'opérations, la plupart des calculs effectués par des modèles basés sur un modèle prétopologique sont possibles et efficaces.

- *inclusion d'une grande part des méthodes prétopologiques existantes à ce jour* : ainsi, nombre de méthodes prétopologiques utilisées dans des articles ou des livres traitant de la prétopologie sont inclus dans la librairie ; d'autres suivront rapidement.

Le développement de cette librairie continuera évidemment d'évoluer par le futur, proposant ainsi encore plus de fonctionnalités et à n'en point douter, des structures de données encore plus adaptées, même si les résultats actuels demeurent corrects, tout bonnement grâce à un fait simple : « *le pire n'est pas toujours certain* » [49]. La prochaine étape sera d'exécuter des simulations avec de grands jeux de données.

Conclusion

L'étude des réseaux complexes est une activité en plein essor [71]. Nous avons donné un aperçu des progrès que l'on peut réaliser grâce aux apports de la théorie de la prétopologie. On se rend bien compte de la possibilité de l'existence de principes sous-jacents allant au-delà des particularités de chaque domaine, et donc l'intérêt de développer un cadre général de modélisation, problématique que nous avons en partie résolue, ou tout du moins à laquelle nous apportons une réponse possible.

Effectivement, la théorie de la prétopologie nous permet de construire de puissants outils d'analyse. En effet, un autre enjeu de la modélisation d'un réseau consiste à faire « parler » les données liées à ce réseau. En effet, une fois les observations recueillies, des questions d'interprétation peuvent se poser qui suscitent le développement de solutions mathématiques, solutions que nous proposons dans ce recueil et qui sont, à notre sens, originales.

De nouvelles directions peuvent ainsi se développer, grâce à un cadre théorique général, permettant une meilleure compréhension de l'émergence de la topologie des réseaux et de la dynamique de processus se déroulant sur ceux-ci, l'interaction réciproque entre topologie du réseau et dynamique des processus restant évidemment un sujet à explorer.

Dans les travaux futurs, une prochaine étape sera d'appliquer notre modélisation et nos nouvelles mesures à de très grands réseaux et d'en estimer ainsi la limite, en terme de complexité et d'analyse. Si les calculs sont assez rapides sur un réseau composé d'environ 350.000 éléments et possédant 1.500.000 relations, il n'en sera pas de même pour des réseaux ayant un million d'éléments et dix millions de relations. Avec nos méthodes de classification par la complexité de Kolmogorov, et de structuration par les centres, il sera très intéressant d'évaluer la pertinence des résultats, principalement sur les grands réseaux.

Concernant la librairie, l'optimisation de certaines fonctions permettant de meilleures performances de simulation sont à envisager : les futurs développements permettront de la compléter, de distribuer les calculs quand ils commenceront à être conséquents et de créer de nouveaux modèles algorithmiques plus complets, afin de garantir des temps d'exécution accrus, qui seront néanmoins tributaires de la complexité des opérations engendrées.

Annexes

Extraits de code JAVA de PretopoLib

Listing 5.5 – Fermeture.

```
1 public PretopoNodes closure(PretopoNodes subsn, String rel\_label,
2 Object param){
3
4         PretopoNodes clsub=new PretopoNodes(subsn);
5         PretopoNodes tmpset=new PretopoNodes(subsn);
6         do{
7             tmpset=new PretopoNodes(clsub);
8             clsub = pseudoclosure(tmpset, rel\_label, param);
9         }while (clsub.getNodeSet().size()>tmpset.getNodeSet().size());
10
11        return clsub;
12    }
```

Listing 5.6 – Fonction bord.

```
1 public PretopoNodes b(PretopoNodes A, String rel\_label, Object param)
2     {
3         PretopoNodes bset=new PretopoNodes();
4         bset=A.substract(interior(A,rel\_label,param));
5
6         return bset;
7     }
```

Listing 5.7 – Calcul de la famille des fermés élémentaires.

```
1 public ArrayList<PretopoNodes> getAllElementaryClosedSubsets(PretopoNodes A,
2 String rel\_label,Object param)
3     {
4         ArrayList<PretopoNodes> Fe=new ArrayList<PretopoNodes>();
5
6         Iterator<Elmt> itr=A.getNodeSet().iterator();
7         while(itr.hasNext())
8         {
9             Elmt elmt=new Elmt(itr.next());
10            PretopoNodes tmpset=closure(new PretopoNodes(elmt),rel\_label,
```

```
11              param);
12              tmpset.label=elmt.label;
13
14              Fe.add(tmpset);
15          }
16
17          return Fe;
18      }
```

Listing 5.8 – Calcul de la famille des fermés minimaux.

```
1  public ArrayList<PretopoNodes> getAllMinimalClosedSubsets(ArrayList<PretopoNodes>
2  Fe)
3      {
4          HashSet<PretopoNodes> Fm=new HashSet<PretopoNodes>();
5          ArrayList<PretopoNodes> F=new ArrayList<PretopoNodes>();
6          F.addAll(Fe);
7
8          boolean minimal;
9
10         while(F.size()>0)
11         {
12             PretopoNodes Fx=new PretopoNodes(F.remove(0));
13             minimal=true;
14
15             ArrayList<PretopoNodes> FTemp=new ArrayList<PretopoNodes>();
16             FTemp.addAll(F);
17
18             while(FTemp.size()>0 \&\& minimal)
19             {
20                 PretopoNodes Gx=new PretopoNodes(FTemp.remove(0));
21                 if(Gx.isSubset(Fx))
22                     minimal=false;
23                 else
24                 {
25                     if(Fx.isSubset(Gx))
26                         F.remove(Gx);
27                 }
28             }
29
30             if(minimal)
31                Fm.add(Fx);
32         }
33
34         ArrayList<PretopoNodes> Fm\_list=new ArrayList<PretopoNodes>();
35         Fm\_list.addAll(Fm);
36
37         return Fm\_list;
38
```

```
39      }
```

Listing 5.9 – Calcul des composantes fortement connexes.

```
1 public ArrayList<PretopoNodes> getAllStronglyConnectedComponents(PretopoNodes
2 A, String rel\_label, Object param)
3     {
4         ArrayList<PretopoNodes> SC=new ArrayList<PretopoNodes>();
5         ArrayList<PretopoNodes> F=getAllElementaryClosedSubsets(A, rel\_label,
6         param);
7         ArrayList<PretopoNodes> FT=new ArrayList<PretopoNodes>();
8
9
10        FT.addAll(F);
11
12        while(FT.size()>0)
13        {
14            PretopoNodes set\_c=new PretopoNodes();
15            PretopoNodes temp\_set=FT.remove(0);
16            for(int i=0;i<F.size();i++)
17            {
18                if(temp\_set.equals(F.get(i)))
19                {
20                    set\_c.addElmt(Nodes.getElmtbyLabel(F.get(i).label));
21                    FT.remove(F.get(i));
22                }
23            }
24
25            SC.add(set\_c);
26        }
27
28        return SC;
29    }
```

Résultats de méthodes appliquées sur des données de réseaux

Méthode MCP appliquée à un réseau d'interactions entre protéines dans la levure [80] (215 classes)

{637}
{1348}
{482}
{1474}
{1362}
{1031, 1040, 1100, 1120, 1136, 1261, 1264, 1309, 1310, 1373, 1412, 1491, 1498, 1500, 1502, 1511, 1513, 1515, 1542, 1568, 1585, 1587, 1600, 1618, 1636, 1707, 1708, 1709, 1714, 1787, 1810, 1841, 1870, 1871, 2010, 2056, 2058, 483, 484, 499, 512, 550, 784}
{1001, 1019, 1023, 1028, 1033, 1034, 1037, 1052, 1063, 1070, 1071, 1076, 1077, 1080, 1081, 1087, 1109, 1146, 1147, 1152, 1200, 1221, 1406, 1418, 1429, 1434, 1452, 1456, 1476, 1518, 1525, 1529, 1545, 1548, 1550, 1551, 1605, 1614, 1619, 1624, 1631, 1637, 1673, 1680, 1682, 1685, 1689, 1690, 1691, 1710, 1739, 1741, 1742, 1769, 1771, 1779, 1785, 1817, 1820, 1824, 189, 190, 1915, 1920, 1990, 2001, 2013, 2046, 2054, 2126, 2144, 318, 428, 439, 440, 441, 447, 453, 461, 462, 508, 509, 574, 582, 597, 601, 627, 628, 641, 644, 648, 659, 671, 672, 674, 678, 716, 721, 722, 723, 724, 725, 726, 728, 735, 824, 848, 867, 870, 872, 875, 905, 907, 908, 910, 960, 966, 967}
{1301}
{922}
{1364}

{842}
{1405}
{1465}
{636}
{931}
{527}
{528}
{1006, 102, 103, 1067, 1069, 1072, 1073, 1093, 1153, 1154, 1188, 1190, 1202, 1377, 1378, 1381, 1384, 1407, 1443, 1444, 1449, 1477, 1480, 1482, 1509, 1524, 1534, 1541, 1544, 1549, 1558, 1594, 1606, 1607, 1608, 1630, 1639, 1641, 1642, 1645, 1667, 1678, 1702, 1733, 1763, 1765, 1776, 1778, 1781, 1782, 1802, 1858, 1880, 1882, 1884, 1897, 1899, 1942, 1972, 1976, 2030, 2084, 2143, 2291, 2336, 450, 456, 457, 458, 459, 460, 485, 486, 487, 526, 566, 570, 583, 673, 679, 680, 681, 688, 704, 845, 854, 855, 887, 893, 93, 959, 98, 988, 989, 992, 995}
{1, 1131, 1350, 1563, 1703, 1828, 1829, 1830, 1900, 1905, 1906, 2006, 3, 463}
{464}
{1008, 107, 110, 111, 1111, 1116, 1139, 1151, 1159, 1161, 1172, 1213, 1281, 1353, 1376, 1395, 1403, 1484, 1485, 1486, 1494, 1495, 1496, 1497, 1501, 1520, 1554, 1565, 1616, 1628, 1646, 1725, 1805, 1831, 1923, 1983, 1989, 2063, 2065, 2068, 2078, 425, 426, 549, 551, 554, 555, 556, 558, 565, 568, 584, 586, 593, 630, 651, 661, 663, 669, 697, 699, 700, 709, 710, 837, 852, 865, 874, 877, 934, 937, 938, 939, 940, 947, 962, 972, 974, 978}
{67}
{1004, 1065, 1098, 1137, 1140, 1145, 1167, 1171, 1236, 1249, 1279, 1289, 1292, 1295, 1333, 1337, 1341, 1345, 1385, 1420, 1475, 1516, 160, 1625, 1627, 1665, 1666, 167, 170, 1727, 1838, 1846, 1847, 1910, 1944, 1962, 1963, 1981, 1994, 20, 200, 202, 21, 2164, 2189, 227, 229, 230, 231, 232, 233, 239, 283, 294, 303, 304, 315, 350, 352, 353, 355, 356, 366, 368, 375, 393, 396, 397, 398, 399, 400, 416, 417, 419, 422, 429, 430, 431, 435, 436, 442, 443, 569, 571, 575, 578, 581, 587, 616, 620, 622, 625, 626, 649, 664, 665, 668, 670, 687, 689, 703, 705, 708, 727, 729, 730, 731, 732, 733, 734, 736, 737, 738, 739, 740, 741, 744, 745, 746, 752, 802, 818, 829, 849, 850, 853, 882, 885, 904, 920, 943, 961, 963, 965, 970, 973, 996}
{840}
{529}
{1472}
{1467}
{1058, 1112, 1178, 1194, 1218, 1232, 1233, 1240, 1259, 1273, 1277, 1324, 1326, 1346, 1379, 1526, 1577, 1589, 1596, 1738, 1788, 1789, 188, 2097, 2273, 2275, 537, 539, 611, 748, 811, 827, 828, 830, 831, 833, 835, 838, 839, 844, 90, 91, 92, 929}
{1322}
{327}
{1473}
{1471}
{1038, 1039, 1053, 1174, 1175, 1177, 1227, 1267, 1268, 1272, 1358, 1479, 1590, 1595, 1908, 2005, 2135, 2138, 2220, 2299, 494, 495, 826}
{326}
{1425}
{1448}
{1062}
{841}
{2338}
{1007, 1017, 1042, 1055, 1096, 1106, 121, 122, 1229, 123, 1235, 125, 1256, 126, 128, 129, 130, 133, 1388, 1469, 1478, 149, 150, 152, 153, 1532, 154, 156, 157, 158, 1580, 159, 172, 174, 175, 176, 177, 178, 1795, 1849, 1851, 1922, 1953, 1954, 197, 198, 209, 210, 214, 2270, 237, 242, 245, 246, 247, 249, 250, 251, 254, 255, 256, 257, 258, 259, 260, 261, 262, 264, 267, 272, 273, 274, 275, 276, 278, 279, 280, 282, 298, 300, 321, 322, 323, 324, 325, 328, 329, 341, 342, 344, 351, 365, 370, 382, 383, 406, 407, 409, 454, 481, 523, 530, 541, 542, 547, 548, 631, 638, 639, 653, 765, 766, 767, 768, 79, 80, 82, 83, 843, 903}
{2158}
{1557, 1610, 1656, 1721, 2060, 66}
{1380}
{1025, 1041, 1044, 1084, 1130, 1303, 1417, 1669, 1877, 2007, 207, 2231, 357, 358, 359, 647, 873, 982}
{106}
{1332}
{1047, 1048}
{1050}
{1061}
{434}
{1064}
{1463}
{1022, 1045, 1046, 1066, 1078, 1079, 1086, 1288, 1423, 1629, 1886, 1966, 2043, 2057, 2110, 2114, 217, 218, 222, 224, 225, 2268, 316, 389, 391, 472, 480, 595, 753, 754, 756, 758, 759, 760, 761, 763, 764, 823, 825, 836, 857, 860}
{1043, 1114, 1115, 1124, 1276, 1336, 1507, 1510, 1579, 1582, 1583, 1599, 1664, 1744, 1957, 2206, 395, 690}
{1015}
{1349}
{1948, 1949, 446, 691}
{1099, 1180, 1546, 1621, 1775, 330, 332, 491, 492, 493, 505, 546, 642, 652, 864, 964}
{866}
{945}
{410}
{1113, 1117, 1122, 1123, 117, 1182, 1250, 1275, 1287, 1372, 1386, 1387, 1392, 1393, 1408, 1419, 1421, 1424, 143, 144, 1523, 1530, 1531, 1650, 1651, 1670, 1747, 1793, 18, 1834, 1835, 1836, 1837, 184, 1852, 186, 187, 1912, 1945, 1971, 215, 2176, 228, 2323, 234, 236, 238, 252, 266, 269, 270, 285, 286, 287, 288, 289, 295, 307, 349, 372, 373, 374, 4, 444, 478, 490, 498, 500, 501, 502, 516, 517, 518, 519, 57, 633, 685, 7, 71, 713, 76, 777, 782, 785, 786, 789, 791, 792, 796, 798, 800, 801, 803, 804, 805, 883, 884, 888, 891, 894, 896, 897, 898, 899, 900, 901, 927, 975, 983}
{1016, 1389, 1422, 1440, 146, 1489, 151, 1517, 155, 1593, 161, 162, 1755, 1757, 1772, 2157, 263, 265, 367, 369, 371, 388, 437, 445, 497, 503, 504, 506, 545, 632, 640, 656, 658, 846}
{1399}
{1400}
{1567}
{1466}
{1401}
{1003, 1102, 1104, 1105, 1129, 1132, 1254, 1334, 1396, 1527, 1535, 1537, 1695, 17, 1730, 1814, 2019, 2053, 2193, 277, 290, 291, 292, 293, 390, 427, 433, 474, 475, 507, 543, 544, 643, 657, 686, 783, 917, 918, 921, 925, 926, 928}
{240}
{119}
{1451}
{1011, 1021, 1059, 1060, 1144, 1210, 1212, 1225, 1255, 1375, 1432, 1442, 1555, 1718, 1720, 1724, 1767, 1808, 194, 195, 2216, 302, 361, 799, 919}
{1647, 750, 751}
{580}
{1464}
{1308}
{706}
{1431}
{1331}
{1404}

{15, 1521, 1919, 1950, 1960, 2202, 2353, 588, 589, 936, 946}
{2142}
{514}
{1411}
{603, 609}
{749}
{473}
{1398}
{694}
{1751, 1753}
{139, 142}
{951}
{1197, 1198, 1368, 1726, 2045, 2048}
{8}
{1335}
{104}
{1460, 1468, 693}
{1012, 1581, 16, 1640, 2022, 2040, 2122, 941}
{1299}
{1010, 1181, 1187, 1294, 1352, 1354, 1602, 1674, 1719, 1806, 183, 1861, 1866, 2100, 30, 31, 714, 715}
{1360, 1361}
{1179, 1183, 1207, 1208, 1211, 1220, 1271, 1394, 1435, 1770, 2070, 2119, 2120, 2209, 2308, 385, 387, 585, 617, 834}
{862, 863}
{1135, 1192, 145, 2011, 997}
{515}
{1896, 2334, 40, 41}
{1917, 2127, 51}
{9}
{1503}
{1397}
{1560, 1564, 2096}
{1409}
{915}
{572}
{1426}
{1437}
{573}
{602}
{1447}
{1162}
{1121}
{942}
{1014, 115, 116, 1165, 118, 1247, 127, 131, 132, 134, 135, 136, 137, 138, 147, 1487, 1505, 163, 164, 165, 166, 168, 171, 1876, 212, 2162, 2302, 241, 243, 248, 253, 33, 339, 343, 345, 346, 347, 348, 377, 39, 401, 402, 403, 418, 420, 421, 423, 540, 591, 592, 594, 596, 60, 61, 62, 63, 64, 65, 654, 684, 712, 775, 778, 779, 780, 781, 788, 790, 793, 794, 795, 944, 985, 990}
{1169, 1311, 1318, 1357, 1402, 1446, 1493, 1506, 1570, 1638, 173, 1812, 199, 201, 2254, 226, 268, 299, 301, 306, 338, 44, 46, 48, 49, 598, 676, 677, 683, 806, 930, 932, 933}
{2297}
{12, 14, 1508, 2173, 2257, 2261}
{1170, 1283, 1285, 1321, 1323, 179, 180, 181, 182, 2038, 208, 334, 335, 336, 337, 510, 520}
{1195}
{19, 980}
{1947, 524}
{2073, 599}
{1238}
{24, 27}
{77}
{85}
{1756, 1974, 88}
{536}
{525}
{1251, 1615}
{522}
{50}
{1470}
{535}
{1298}
{1206}
{513}
{1328, 1410, 1439, 1450, 169, 203, 284, 319, 52, 53, 55, 56, 58, 59, 68, 69, 70, 717, 72, 73, 74, 75, 762, 78, 976, 984}
{405}
{465, 471}
{701}
{1445}
{114}
{411}
{534}
{29}
{10, 11, 994}
{1633}
{211, 213, 296}
{2150}
{720}
{86, 87}
{600}
{1051}
{902}
{378, 381}
{521}
{489}
{2004}
{94}

{97}
{23}
{204}
{112}
{25}
{219}
{84}
{185}
{386}
{380}
{1562}
{379}
{1199}
{101}
{223}
{1752}
{22}
{5}
{113}
{2}
{376}
{196}
{469}
{13}
{28}
{96}
{37}
{100}
{911}
{1049}
{45}
{6}
{148}
{466}
{99}
{95}
{35}
{220}
{192}
{54}
{191}
{1355}
{496}
{120}
{221}

Méthode 1 des ϵ-centres appliquée à un réseau de livres sur Amazon.com « le client qui a acheté ce livre a aussi acheté ... »(2 classes du plus central au moins central)

{A National Party No More,Against All Enemies,American Dynasty,Big Lies,Bush Country,Bush at War,Bushwhacked,Bushwomen,Deliver Us from Evil,Dereliction of Duty,Disarming Iraq,Dude, Where's My Country ?,Endgame,Fanatics and Fools,Fighting Back,Ghost Wars,Had Enough ?,Hating America,House of Bush, House of Saud,Legacy,Let Freedom Ring,Lies and the Lying Liars Who Tell Them,Losing Bin Laden,Off with Their Heads,Persecution,Plan of Attack,Rise of the Vulcans,Rogue Nation,Rumsfeld's War,Sleeping With the Devil,Spin Sisters,Ten Minutes from Normal,The Bubble of American Supremacy,The French Betrayal of America,The Great Unraveling,The Lies of George W. Bush,The Politics of Truth,The Price of Loyalty,The Savage Nation,The Sorrows of Empire,Why America Slept,Worse Than Watergate}

{1000 Years for Revenge,All the Shah's Men,Allies,America Unbound,Arrogance,Betrayal,Bias,Breakdown,Buck Up Suck Up,Bush vs. the Beltway,Charlie Wilson's War,Colossus,Dangerous Dimplomacy,Downsize This !,Empire,Freethinkers,Give Me a Break,Hegemony or Survival,Hillary's Scheme,Hollywood Interrupted,It's Still the Economy, Stupid !,Living History,Meant To Be,MoveOn's 50 Ways to Love Your Country,Perfectly Legal,Power Plays,Rush Limbaugh Is a Big Fat Idiot,Shrub,Shut Up and Sing,Slander,Soft Power,Stupid White Men,Surprise, Security, the American Experience,Tales from the Left Coast,The Best Democracy Money Can Buy,The Bushes,The Buying of the President 2004,The Choice,The Clinton Wars,The Culture of Fear,The Death of Right and Wrong,The Enemy Within,The Exception to the Rulers,The Faith of George W Bush,The Future of Freedom,The Man Who Warned America,The New Pearl Harbor,The O'Reilly Factor,The Official Handbook Vast Right Wing Conspiracy,The Perfect Wife,The Real America,The Right Man,The Third Terrorist,Thieves in High Places,Things Worth Fighting For,Those Who Trespass,Useful Idiots,We Will Prevail,We're Right They're Wrong,Weapons of Mass Deception,What Liberal Media ?,Who's Looking Out for You ?,Why Courage Matters}

Recherche des composantes fortement connexes sur un réseau de « blogs » traitant des élections présidentielles de 2004 aux Etats-Unis d'Amérique [1] (688 composantes)

{100monkeystyping.com, 12thharmonic.com/wordpress, 750volts.blogspot.com, aboutpolitics.blogspot.com, ace.mu.nu, aceoftrumpblog.blogspot.com, acepilots.com/mt, acertainslantoflight.blogspot.com, achicknamedmarzi.com, ackackack.com, activistchat.com/blogiran, adeimantus.blogspot.com, adviceforlefty.blogspot.com, agonist.org, aintnobaddude.com, ajbenjaminjr.blogspot.com, alamonation.blogspot.com, alarmingnews.com, aldaynet.org, alexpelan.blogspot.com, alicublog.blogspot.com, alittlemoretotheright.com, all-encompassingly.com, allanjenkins.typepad.com, allspinzone.blogspot.com, alphapatriot.com, alphapredator.com/wdtgw.htm, althippo.blogspot.com, althouse.blogspot.com, alvintostig.typepad.com, americablog.blogspot.com, americablog.org, americanchronicle.blogspot.com, americandigest.org, americandinosaur.blogspot.com, americanfuture.typepad.com, americanpolitics.com, americanreport.blogspot.com, americansforbayh.blogspot.com, americanthinker.com, americanwarmonger.blogspot.com, amigo13.blogspot.com, amleft.blogspot.com, amptoons.com/blog, anarchyxero.robfindlay.org, andifeelfine.blogspot.com,

andymatic.com, angrybear.blogspot.com, angrydesi.typepad.com, angrygwn.blogspot.com, angryhomo.blogspot.com, anncoulter.org, annika.mu.nu, anoldsoul.blogspot.com, anonymoussources.blogspot.com, answerguy.blogspot.com, antijohnkerry.blogspot.com, antiprotester.blogspot.com, apologiesdemanded.blogspot.com, arancaytar.blogspot.com, archpundit.com, arkansastonight.com, arkanssouri.blogspot.com, armchairgeneralist.typepad.com, asmallvictory.net, astrosoup.blogspot.com, atease.blogspot.com, athena.blogs.com, atrios.blogspot.com, awideawake.blogspot.com, ayerdis.com/rants.php, babytrollblog.com, baldilocks.typepad.com, balloon-juice.com, barkingdingo.blogspot.com, bartcop.com, beautifulatrocities.com, beefalwayswins.blogspot.com, beldar.blogs.com/beldarblog, belmontclub.blogspot.com, bensworld.patriotforum.org, bernardmoon.blogspot.com, betsyspage.blogspot.com, bidinotto.journalspace.com, bigfatroid.blogspot.com, billhobbs.com, billmon.org, billroggio.com/easongate, billsrants.blogspot.com, bitheads.blogspot.com, bittersweetme.net/blog/index.php, bkennelly.com/vox, blackmanforbush.blogspot.com, blindmindseye.com, blithered.blogspot.com, blog.dccc.org, blog.kenlan.net, blog.ncsweb.us, blog01.kintera.com/dnccblog, blogforamerica.com, blogfreeordie.blogspot.com, blogicus.com, blogodob.blogspot.com, blogs.salon.com/0002874, blogs.salon.com/0003364, blogsagainsthillary.com, blogsforbush.com, blogslut.com, blogsofwar.com, blogspirator.blogspot.com, blubberhollow.com/blog, blue6golf.blogspot.com, bluegoldfish.blogs.com/surface, bluegrassroots.blogspot.com, blueoregon.com, bluestateredstate.blogspot.com, bluestates.blogspot.com, bobhayes.net, bodyandsoul.typepad.com, boffo.typepad.com, bolo-boffin.blogspot.com, boottotheskull.com, bopnews.com, borgardblog.blogspot.com, brainshavings.com, brilliantatbreakfast.blogspot.com, brown-cross.com/usualsuspects, btcnews.com/btcnews, buckeyepundit.blogspot.com, bullmooseblog.com, burntorangereport.com, busybusybusy.com, buzzmachine.com, c-pol.blogspot.com, californiaconservatives4truth.blogspot.com, calpoly.edu/ doleary/blog.htm, captainsquartersblog.com/mt, catscape.com, cayankee.blogs.com, cbaker.org/blog, cbcbcbcb.blogspot.com, ccs178.com, cdharris.net, celluloid-wisdom.com/pw, cemonitor.blogspot.com, centinel.blogspot.com, chepooka.com, chiefsblog.blogspot.com, chrenkoff.blogspot.com, christian-patriot.blogspot.com, civilcommotion.com, clarified.blogspot.com, claudialong.com/blog, claycalhoun.com, clubforgrowth.org/blog, coffee.ondragonswing.com, collegedems.com/blog, commonsenserunswild.typepad.com, confederateyankee.blogspot.com, conservativeblogger.com, conservativecat.com, conservativedialysis.com, conservativeeyes.blogspot.com, conservativelife.com/blog, conservativeobserver.blogspot.com, conservativepush.blogspot.com, conservativeresponse.com, conservativerevolution.com, conservativessuck.blogspot.com, conservativezone.blogspot.com, contrapositive.blogspot.com, corrente.blogspot.com, coxandforkum.com, crankybastard.blogspot.com, crankyneocon.com/crankyneocon, crookedtimber.org, crooksandliars.com, cybermenace.blogspot.com, cynicalnation.com, daddypundit.blogspot.com, dailykos.com, dailypundit.com, dailyrecycler.com/blog, dailywarnews.blogspot.com, daisycutter.blogspot.com, dalythoughts.com, damntheman.net, danieldrezner.com/blog, danzfamily.com, dartblog.com, davidsirota.com, dawnofnewamerica.blogspot.com, deadcantrant.com/blog, deanesmay.com, deansjournal.blogspot.com, deathpenaltyusa.blogspot.com, decision08.blogspot.com, dedspace.blogspot.com, demagogue.blogspot.com, demleft.blogspot.com, democraticunderground.com, democrats.org/blog, denbeste.nu, denisdekat.com, deserttusk.blogspot.com, dewar.journalspace.com, digbysblog.blogspot.com, digestiblenews.blogspot.com, diggersrealm.com/mt, dimmykarras.blogspot.com, diplomadic.blogspot.com, discerningtexan.blogspot.com, discourse.net, disintegrator.blogspot.com, dissectleft.blogspot.com, diypolitics.blogdrive.com, dizzy-girl.net/index.php, dmobley.blogspot.com, dneiwert.blogspot.com, dogfight04.typepad.com, dogwooddreams.com/serendipity, dohiyimir.typepad.com, dolphinsdock.com/blog, donaldsensing.com, donkey2004.blogspot.com, donspoliticalblog.blogspot.com, downtownlad.blogspot.com, drudgereport.com, drumwaster.com, drunkensamurai.blogspot.com, dummocrats.com, dynamist.com/weblog, economicswithaface.com/weblog, eeg3.com, ejectejecteject.com, elayneriggs.blogspot.com, elderbearden.blogspot.com, electablog.com, electionprojection.com, electricvenom.com, elemming2.blogspot.com, elevendayempire.com/movabletype, emergingdemocraticmajority.com, emergingdemocraticmajorityweblog.com/donkeyrising, emilythedog.blogspot.com, evangelicaloutpost.com, everythingiknowiswrong.com, evilpundit.com, fafblog.blogspot.com, farleft.blogspot.com, flagrancy.net, folkbum.blogspot.com, fraterslibertas.com, frederickclarkson.com, freedomofthought.com, fringeblog.com, fritzs-thoughts.blogspot.com, fromtheroots.org, gaypatriot.blogspot.com, gcruse.typepad.com, gevkaffeegal.typepad.com/the_alliance, ghostofaflea.com, gindy.blogspot.com, gisleson.com/blog, goose3five.blogspot.com, gopandcollege.blogspot.com, gopandthecity.blogspot.com, gopbloggers.org, gopinsight.com, gotdesign.blogspot.com, grassrootsmom.blogspot.com, greatestjeneration.com, grupo-utopia.com/blog/isou, gullyborg.typepad.com, headland.blogspot.com, healingiraq.blogspot.com, hectorvex.blogspot.com, hellafaded.blogspot.com, hereswhatsleft.typepad.com, hiddennook.blogspot.com, higherpieproductions.com, hillcountryviews.blogspot.com, holecard.blogspot.com, home.earthlink.net/ fsrhine, home.earthlink.net/ kevin.omeara, home.houston.rr.com/skeptical, home.midsouth.rr.com/joefish, homespunbloggers.blogspot.com, homicidalmaniak.com, horologium.net, hotflashreport.com, hughhewitt.com, hugozoom.blogspot.com, iddybud.blogspot.com, idiosyncratictendencies.com, idlehandsmag.com/yellowdog04, iflipflop.com, ignatz.blogspot.com, ihatepatrobertson.com, ikilledcheguevara.blogspot.com, imao.us, incite1.blogspot.com, indcjournal.com, indepundit.com, instapundit.com, interested-participant.blogspot.com, interestingtimes.blogspot.com, interestingtimes.typepad.com, inthebullpen.com, iowageek.blogspot.com, iraqthemodel.blogspot.com, itscomedowntothis.blogspot.com, itznewstome.blogspot.com, ivotedforgeorge.com, j-bradford-delong.net/movable_type, jacklewis.net, jadbury.com, jadbury.com/blog, jadedreality.blogspot.com, jameswolcott.com, janegalt.net, jdeeth.blogspot.com, jeffblanco.blogdrive.com, jewssansfrontieres.blogspot.com, jg-spot.blogspot.com, jihadwatch.org, jimgilliam.com, jimtreacher.com, jinkythecat.blogspot.com, jlynnsmith30.blogspot.com, joatmoaf.typepad.com/i_love_jet_noise, joepublicus.blogspot.com, john.hoke.org, johnmccrory.com, jokerstotheright.blogspot.com, joshferguson.blogspot.com, journals.aol.com/breuvers/madashellnotgoingtotakeitanymore, jperspective.com, jquinton.com, jrogg.blogspot.com, juancole.com, juliusblog.blogspot.com, justalkin.us, kalblog.com, karmicsoup.blogspot.com, kcdems.blogspot.com, keithy1028.bravejournal.com, kellipundit.blogspot.com, kensain.com, kerryforpresident2008.blogspot.com, kevincourser.blogspot.com, kickthelefttist.blogspot.com, kimdutoit.com/dr/weblog.php, king-of-fools.com, kiosan.blogharbor.com, kitchennetwork.blogspot.com, kydem.blogspot.com, la4israel.org/wordpress, lashawnbarber.com, laughatliberals.com, laughingwolf.net, lawdork.blogspot.com, lawguy.blogspot.com, leanleft.com, left2right.typepad.com, leftcoastconservative.blogspot.com, lefterer.com, lefti.blogspot.com, leftthought.blogspot.com, legendgames.net/myblog.asp, leonards-digest.blogspot.com, liberalavenger.com, liberaloasis.com, libertarianjackass.blogspot.com, limericksavant.blogspot.com, liquidlist.com, littlegreenfootballs.com/weblog, littletinylies.com, livejournal.com/users/jmhm, livinginbabylon.blogspot.com, loadedmouth.com, loud-liberal.blogspot.com, loveamericahatebush.com, lowrentrat.blogdrive.com, lt-smash.us, luxeword.com, madanthony.org, made4theinternet.blogspot.com, madkane.com/notable.html, madtechspeaks.blogspot.com, mahablog.com, manifestcontent.net, mantic.squarespace.com, margaretcho.com/blog/blog.htm, mariestwocents.blogspot.com, markadams.blogdrive.com, markarkleiman.com, markheimonen.blogspot.com, marknicodemo.blogspot.com, markschmitt.typepad.com/decembrist, martinipundit.com, maruthecrankpot.blogspot.com, mathewgross.com, mathewgross.com/blog, maxspeak.org/mt, maybeithinktoomuch.com, mayflowerhill.blogspot.com, mdcbowen.org/cobb, meateatingleftist.com, mediajunkie.com/edgewise, mementomoron.blogspot.com, metallicity.com/mb.asp, metropol47.blogspot.com, mhking.mu.nu, michaelberube.com, michaeltotten.com, michellemalkin.com, midguard.blogspot.com, midwestrightwingers.blogspot.com, mirocat.com, misschin.bloggage.com, mixtersmix.blogspot.com, montages.blogspot.com, motgs.com, msnbc.msn.com/id/3449870, mstabile.blogspot.com, mtvirtus.blogspot.com, mudandphud.blogspot.com, mudvillegazette.com, musicfromthebluelight.com/b, mwilliams.info, my2cent.blogspot.com, mycatharsis.com, mydd.com, myleftbrain.com, mypetjawa.mu.nu, mysterypollster.com/main, nashvilletruth.blogspot.com, nastyboys.blogspot.com, nathancallahan.com, nathannewman.org/log, nationalcenter.org/blog.html, nationalreview.com/thecorner, nearouterheaven.blogspot.com, needlenose.com, nerepublican.blogspot.com, netpolitik.blogspot.com, netweed.com/postmodernanarchist, neverwrongnever.blogspot.com, newappeal.blogspot.com, newcounterculture.com/log, newdemocrat.blogdrive.com, newdonkey.com, newleftblogs.blogspot.com, newsbloggers.blogspot.com, nicedoggie.net, nickiegoomba.blogspot.com, nielsenhayden.com/electrolite, nielsenhayden.com/makinglight, node707.com, nofearoffreedom.blogspot.com, noguru.tblog.com, nomayo.mu.nu, nomoremister.blogspot.com, nonboxthinking.blogspot.com, norbizness.com, noseyonline.com, notgeniuses.com, o4tom.blogspot.com, obamablog.com, obsidianwings.blogs.com, oddhours.blogspot.com, officesmileman.com, offthekuff.com/mt, ofthemind.blogspot.com, ogresview.blogspot.com, oipp.blogspot.com, okieonthelam.com, oldfashionedwhippersnapper.blogspot.com, oliverwillis.com, one38.org, opiniontimes.blogspot.com, opiniontimes.com, oppb.com, ospolitics.org, outsidethebeltway.com, oxblog.blogspot.com, pacdaily.com, pacificviews.org, pajamaeditors.blogspot.com, pajamahadin.com, palominas.blogspot.com, pandagon.net, papadoc.net/pinkflamingobar.html, parabasis.typepad.com, pardonmyenglish.com, patrickruffini.com, patriot-paradox.com, patriotblog.com, patriotboy.blogspot.com, patriotism101.blogspot.com, patriotsforbush.com, patterico.com, peacegarden.blogspot.com, pejmanesque.com, pennsylvanianinexile.blogspot.com, pennywit.com/drupal, permanentred.com, perryvsworld.blogspot.com, phillybri.blogspot.com, philomathean.blogs.com, philosophers-stone.blogspot.com, phoblographer.com, piano-tuning.net/blog, pinkofeministhellcat.typepad.com/pinko_feminist_hellcat, poeticvalues.blogspot.com, poliblogger.com, polipundit.com, politicalmusings.net, politicalstrategy.org, politicaltracker.com, politicalvicesquad.blogspot.com, politics.feedster.com, polizeros.com, polstate.com, poorjustin.blogspot.com, portapulpit.com, powerageambleson.blogspot.com, powerlineblog.com, powerpundit.com, presidentboxer.blogspot.com, presidentreagan.info/discussion, professorbainbridge.com, profgoose.blogspot.com, progressivesociety.com/blog, prolifeblogs.com, propagandamachine.blogspot.com, proponentofreason.blogspot.com, prosandcons.us, prospect.org/weblog, pudentilla.blogspot.com, punditguy.com, punditz.journalspace.com, purplestates.blogspot.com, qando.net, qoae.net, quante.blogspot.com, quibbles-n-bits.com, radgeek.com, radioequalizer.blogspot.com, raedinthemiddle.blogspot.com, randomjottings.net, ratherbiased.com/news, rathergate.com, reachm.com/amstreet, realclearpolitics.com, realitybasednation.com, realordeal.blogspot.com, reckersau.blogspot.com, red-harvest.blogspot.com, redmindbluestate.blogspot.com, redstate.org, redstatealliance.blogspot.com, reggiescorner.blogspot.com, regimechangeiran.com, religiousleft.blogspot.com, republicoft.typepad.com, resistoppression.blogspot.com, respublica2004.blogspot.com, retiredwarrant.blogdrive.com, retrogrouch.net, rfburnhertz.net/rf_blog, rhetoricrhythm.blogspot.com, righthandthief.blogspot.com, rightmoment.blogspot.com, rightonred.net, rightrainbow.com, rightstate.blogspot.com, rightvoices.com, rightwingjournal.blogspot.com, rightwingnews.com, rightwingthinking.blogspot.com, rightwingtroll.blogspot.com, rightword.blogspot.com, rising-hegemon.blogspot.com, risinghegemon.blogspot.com, riteturnonly.blogspot.com, rittenhouse.blogspot.com, riverbendblog.blogspot.com, roadtosurfdom.com, robot-invasion.com, robschumacher.blogspot.com, rockpaperscissor.blogspot.com, rogerailes.blogspot.com, rogerlsimon.com, romeocat.typepad.com, rooksrant.com, rosenblog.com, roseoftheknights.com/yank/pride.html, roughstock.blogspot.com, rowhouselogic.com,

roxanne.typepad.com, rudepundit.blogspot.com, ruminatethis.com, ryanshead.com, s88172659.onlinehome.us/spockosbrain.html, sadlyno.com, samizdata.net/blog, sayanythingblog.com, scha-den-freu-de.blogspot.com, schadelmann.com, scoobiedavis.blogspot.com, scottcsmith.net, scrutinyhooligans.blogspot.com, sebourn.com/weblog, section-31.blogspot.com, secureliberty.org/, seeingtheforest.com, seetheforest.blogspot.com, sgtstryker.com, shadydowns.blogspot.com, shakespearessister.blogspot.com, shakingspears.blogspot.com, shallows.blogspot.com, shamelessantagonist.blogspot.com, sharedthought.com/weblog, shininglight.us, shotofpolitics.blogspot.com, simbaud.blogspot.com, simplyappalling.blogspot.com, sinfactor.blogspot.com, sistertoldjah.com, sixmeatbuffet.com, slantpoint.com, slingsnarrows.erudite-absurdity.com, slipofthepen.blogspot.com, slowplay.com, slublog.blogspot.com, smithersmpls.com, sneakeasysjoint.com, snunes.blogspot.com, sondrak.com, soundingthetrumpet.blogspot.com, southernappeal.blogspot.com, southernwatch.blogspot.com, southknoxbubba.net/skblog, spreadingunderstanding.blogspot.com, squiggler.braveblog.com, stagefour.typepad.com/commonprejudice, stalinism.com/shot-by-both-sides, stan-istan.blogspot.com, stcfoundation.blogspot.com, stealthebandwagon.blogspot.com, stevecopy.blogharbor.com, stevegilliard.blogspot.com, stevemacek.blogspot.com, stillmanvalley.org/aoo, stix1972.blogspot.com, stmonk.blogspot.com, stolenthunder.blogspot.com, strangedoctrines.typepad.com, strategicintelligence.blogspot.com, suburbanguerrilla.blogspot.com, sweetgrrrl.blogspot.com, swfwithattitude.blogspot.com, swingstateproject.com, tagorda.com, takebackthemedia.com, talkingpointsmemo.com, talkleft.com, talkshowamerica.blogspot.com, tbogg.blogspot.com, techievampire.net/wppol, techtrend.com/blog/taranicole, teenageenthusiast.blogspot.com, templeofjennifer.com/blog, texasbug.blogspot.com, thatcoloredfellasweblog.bloghorn.com, thatgoodnight.com, thatliberalmedia.com, the-goddess.org, the-hamster.com, theamericanmind.com, theangryliberal.blogspot.com, thebloghill.blogspot.com, theblueview.blogspot.com, thecarpetbaggerreport.com, thecommonvirtue.com, theconservativezone.com, thecore4.com, thecrankfiles.blogspot.com, thedonovan.com, thefulcrum.blogspot.com, theglitteringeye.com, thejessefactor.blogspot.com, theleftcoaster.com, theliberalslies.blogspot.com, thelucideye.blogspot.com, theneo-progressive.blogspot.com, thenudnikfile.blogspot.com, thepatriette.com, thepaytons.org/essays/considerettes, thepoliticalteen.net, thepoorman.net, therightblog.blogspot.com, therightcoast.blogspot.com, therightwingconspiracy.org, therinsecycle.com, therogueangel.com/blog, therostra.blogspot.com, thesnooter.blogspot.com, thespoonsexperience.com, thetalentshow.org, thetalkingdog.com, thetarpit.blogspot.com, thetemplarpundit.blogspot.com, theunderscore.blogspot.com, theurbangrind.blogspot.com, thevaticanofliberalism.com, thewashingtonnote.com, theworldaroundyou.com, thewritewing.blogspot.com, theyetimn.blogspot.com, thinkingright.us, thinkingrocks.blogspot.com, thisliberal.com, thismodernworld.com, thomasblog.typepad.com/bloggg, thomasgalvin.blogspot.com, timblair.net, timblair.spleenville.com, tom-hanna.org, tomburka.com, tomcoburnisabigfatjerk.blogspot.com, toneofvoice.blogspot.com, toteota.blogspot.com, townhall.com/clog, travalex.blogspot.com, tristero.blogspot.com, truebluegal.blogspot.com, truthlaidbear.com, truthliesandcommonsense.blogspot.com, tsrightdominion.blogspot.com, turnpikefeminist.com, twentyfirstcenturyrepublican.blogspot.com, txfx.net, uggabugga.blogspot.com, unconventionalwisdom.typepad.com, unfetteredmusings.blogspot.com, unodemocrats.com/blog, upper-left.blogspot.com, usdlaw.blogspot.com, valuerightnews.blogspot.com, vernsblog.thegillfamily.us :8180, victorhanson.com, viewfromaheight.blogspot.com, viewfromtonka.blogspot.com, viewpointjournal.com, vikingpundit.blogspot.com, virginiapolitics.blogspot.com, virginiaprogressive.blogspot.com, vodkapundit.com, volokh.com, waitingfordorothy.blogspot.com, wampum.wabanaki.net, warandpiece.com, washingtonmonthly.com, watchingthewatchers.org, watdanook.blogspot.com, waveflux.net, wehavesomeplanes.blogspot.com, whatsakyer.blogspot.com, wholewheatblogger.com, whopundit.blogspot.com, whoviating.blogspot.com, whyareweback.blogspot.com, williamwebbdotorg.blogspot.com, winceandnod.blogspot.com, windingroad.typepad.com, windsofchange.net, wizbangblog.com, wonkette.com, worldonfire.typepad.com/world_on_fire, wpblog.ohpinion.com, writingcompany.blogs.com/this_isnt_writing_its_typ, wtpm.blogspot.com, xnerg.blogspot.com, xtremerightwing.net, yarsrevenge.blogspot.com, yglesias.typepad.com/matthew, youngconservative.blogspot.com, zebrax.blogs.com, zeke01.typepad.com}
{r21online.com}
{jpandre.blogspot.com}
{madamepresident2020.blogspot.com}
{news-opinion.blog-city.com}
{civicactions.com}
{asiegeofherons.blogspot.com}
{thedailygoodnews.com}
{blogforarizona.com}
{lennonreport.blogspot.com}
{cleancutkid.com, thunewatch.squarespace.com}
{imprescindibile.ilcannocchiale.it}
{moorewatch.com, right-thinking.com}
{tokenyoungconservative.blogspot.com}
{aurelientt.blogspot.com}
{niagarabuzz.com}
{xanga.com/home.aspx ?user=phknrocket1k}
{ltobs.blogspot.com}
{cptnwinky.com}
{jospehsmind.blogspot.com}
{absurd-canadian.blogspot.com}
{erikack.blogspot.com}
{scrappleface.com}
{rightwingbob.com}
{swingleft.blogspot.com}
{r2korn.blogdrive.com, thelastword.blogdrive.com}
{anybodybutbushyall.blogspot.com}
{digitaldissent.blogspot.com}
{criticalviewer.com}
{ladymac.com}
{cnle.net}
{drugwarmovie.com/drugwarmovie/weblog/blogger.html}
{massachusetts-liberal.com}
{elfoflight.blogspot.com}
{politicalpie.modblog.com}
{neoconservatives.blogspot.com}
{marcbrazeau.blogspot.com}
{hellskitchennyc.blogspot.com}
{mercreport.blogspot.com}
{progressivedialogue.com}
{georgewbuy.blogspot.com}
{freerepublic.com}
{atomicairship.com}
{grantb88.com}
{battleground-wisconsin.blogspot.com}
{iamaugustus.blogspot.com}
{tseytlin.blogspot.com}
{fiercepoet.com}
{decksitters.my-expressions.com}
{batr.net, batr.org/commentary.html}
{njsoccermom.blogspot.com}
{kausfiles.com}
{estropundit.blogspot.com}
{hobsonschoice.us}
{franklog.com}
{philoponia.blogspot.com}
{happyhomeschooler.blog-city.com}
{home.earthlink.net/ cdowner}
{radio.weblogs.com/0131587/categories/flush}

{longdukdong44.blogspot.com}
{neoconswatch.blogspot.com}
{cynscorion.com/v-web/b2/index.php}
{politicaldogs.org}
{tomherod.blogspot.com}
{blog.so-amazing.net}
{nationalreview.com/frum/frum-diary.asp}
{question-dean.blogspot.com}
{louisville.edu/ bebusc02/blog/notarobot.html}
{tallone.org}
{streetlog.typepad.com}
{senorpez.com}
{watchandwait.blogspot.com}
{smellyduck.blogspot.com}
{life.laptoplobbyist.com}
{magnificentfreedom.us}
{prayforbush.blogspot.com}
{myworldview034.blogspot.com}
{messopotamian.blogspot.com}
{enlightenedbelle.blogspot.com}
{thewilson.blogspot.com}
{angrychineseblogger.blog-city.com}
{sickofliberals.blogspot.com}
{etherealgirl.blogspot.com}
{dygital.blogspot.com}
{livejournal.com/ ladywriter_mts}
{think2004.com}
{stillfighting.blogspot.com}
{hathaby.net/weblog.php}
{adonai.cc}
{theantibush.org}
{sulliwatch.blogspot.com}
{kippsblog.com}
{onetrueblog.com}
{gmscorner.blogspot.com}
{orient-lodge.com}
{thedumbdemocrat.blogspot.com}
{rmcdaniel.blogspot.com}
{bushmisunderestimated.blogspot.com}
{laylasblog.com}
{woneffe.typepad.com}
{bigjourney.blogspot.com}
{bushblog.us}
{prestoncaldwellblog.blogspot.com}
{theblogwarrior.blogspot.com}
{itsabitmuch.blogspot.com}
{catchingsmoke.blogspot.com}
{stitt.blogspot.com}
{fearfulsymmetry.blogspot.com}
{turnspit.com}
{wanderingmind.blog-city.com}
{zeke01.blogspot.com}
{boston.com/news/blogs/dnc}
{xanga.com/allegrad}
{heavylifting.blogspot.com}
{millers_time.typepad.com}
{gerberatetra.blogspot.com}
{dailyblatt.blogspot.com}
{urbanecoinc.com/wordpress1}
{richmar.blogspot.com}
{poorandwhite.blogspot.com}
{allahpundit.com}
{claytoncorner.blogspot.com}
{rockthrower.blogs.com/rockthrower}
{nationalnewsregister.com/modules/wordpress}
{lettherebepeace.blogspot.com}
{sarcasticsoutherner.blogspot.com}
{lightupthedarkness.org/blog/default.asp}
{liberalangst.org}
{mickikrimmel.com/redcarpet}
{orthogonian.blogspot.com}
{democrants.blogspot.com}
{anothervrwc.blogspot.com}
{usliberals.about.com}
{brianlarry.com}
{warriorsblog.blogspot.com}
{rncnyc2004.blogspot.com}
{anywhichway.net}
{takeaction.blog-city.com}
{theagitator.blogspot.com}
{vocis.com/frank}
{jenjuskhan.blogspot.com}
{gccpundit.blogspot.com}
{uwaythinks.blogspot.com}
{ocd-gx-liberal.blogspot.com}
{americanjihad.blogspot.com}
{sequiturnon.blog-city.com}
{blogs.salon.com/0004024}
{onthewire.blogspot.com}
{blog.johnkerry.com}
{blog.nam.org}
{alamodispatches.blogspot.com}

{dilusionl.blogspot.com}
{nakedrepublic.com}
{discountblogger.com}
{erickohn.com}
{wsrblood.typepad.com/wiserblog}
{americancommentator.blogs.com/the_american_commentator_}
{xianleft.blogspot.com}
{forewarned.blogspot.com}
{thebarrexam.blogspot.com}
{bullmooserepublicans.com/blog}
{isaacspolitics.blogspot.com}
{lashawnbarber.blogspot.com}
{skipperstyle.com}
{chak.org/blog}
{thesakeofargument.com}
{dystopia.blog-city.com}
{hasidicgentile.org}
{nftl.blogspot.com}
{b-patton.blogspot.com}
{journals.aol.com/lily990016/pissoffliberals}
{askhoudari.blogspot.com}
{younglibs.com}
{winthewest.blogspot.com}
{4lina.tblog.com}
{leavethematrix.blogspot.com}
{atrios.blogspot.com/}
{ludovicspeaks.typepad.com}
{nycitystoop.com}
{rightperspective.net}
{leftinthereign.blogspot.com}
{peskytherat.com/pesky}
{asian-nation.org/headlines/index.php}
{tom_thinks.blogspot.com}
{thomaswharry.com}
{hoosierparliament.com}
{urbandemocracy.blogspot.com}
{lorenjavier.blogspot.com}
{robertp17.blogspot.com}
{yoder.ru}
{americanmuslim.blogs.com}
{rkook.blogspot.com}
{truedemocrat.blogspot.com}
{isdl.blogspot.com, isdl.weblogs.us}
{xanga.com/kerry4president}
{americanworldview.tripod.com/weltansblog}
{blog.veggiedude.com}
{fooswah.com}
{nationalbird.net}
{writehouse.us}
{conservativecitizen.blogspot.com}
{devildawg.blogspot.com}
{literatevalues.org/virtue.htm}
{restoreamerica.blogspot.com}
{homepage.mac.com/elisa_camahort/iblog, sccdp.org/blog.php}
{democratvoice.org}
{endthenightmare.blogspot.com}
{pattymurray.com/blog.php}
{lightupthedarkness.org}
{literarybargainville.net}
{philvbblog.blogspot.com}
{katyggls.blogspot.com}
{werenotallincompetent.blogspot.com}
{mrstevebeers.blogspot.com}
{battlegroundstates.blogspot.com}
{datingrepublicans.com}
{beastsbelly.blogspot.com}
{debunker.net}
{politicalthought.net}
{openeyesmemo.com}
{conservativenation.us}
{rightwingbastardo.blogspot.com}
{lonewacko.com}
{maggiespants.co.uk}
{undercaffeinated.com}
{bamapachyderm.blogspot.com}
{shoegal.modblog.com}
{radikal_papi.pitas.com}
{mikesamerica.blogspot.com}
{shrillblog.blogspot.com}
{robbernard.com}
{charlineandjamie.com/dotnetweb01a/blogdisplay.aspx ?logname=jamie&logcatid=48}
{ideastockexchange.com/essays/laubs-mike-interest-politics-web-log.htm}
{henerd.com}
{freedompress.blogspot.com}
{blog01.kintera.com/emilysblog}
{thelonedem.com}
{michael124.blogspot.com}
{jeremybrendan.blogspot.com}
{messengerpuppet.com/blog}
{mydamascusroad.blogspot.com}
{michaelsavagelistener.blog-city.com}
{bryan08.tk}

{stonyfield.com/weblog/strongwomendailynews/index.html}
{calirep.blogspot.com, rednyc.blogspot.com}
{sharpeblog.cybercatholics.com}
{chiron.typepad.com}
{scott-niven.com/saltwaterpizza}
{thirdredeye.blogspot.com}
{masononline.biz}
{conservativedhc.blogspot.com}
{stirs-the-pot.blogspot.com}
{myleftthoughts.blogspot.com}
{progressivecyclone.blogspot.com}
{conservativepunk.com}
{americasteen.tk}
{progressivetrail.org/blog.shtml}
{kenshepherd.blogspot.com}
{95theses.blogspot.com}
{andrewsullivan.com}
{duckdaotsu.blogspot.com}
{dir.salon.com/topics/joe_conason/index.html}
{popstalin.com/news}
{thelefthand.blogspot.com}
{digital-democrat.blogspot.com}
{chadster.capitalfocus.org}
{bensgtusmc.com}
{home.earthlink.net/ techservant}
{scrawlville.com}
{brianhull.net}
{getdexview.barrennorth.com}
{jason.unitedworkers.org}
{charleypatton.blogspot.com}
{blogs.blogosphere.ca/brock}
{bigbloosky.blogspot.com}
{savetheus.blogspot.com}
{speculativepolitics.blogspot.com}
{votekerryedwards2004.blogspot.com}
{mostlycajun.com}
{gillaand.blogspot.com}
{vote4georgew.blogspot.com}
{raincrossblog.com}
{liberalthunderdome.blogspot.com}
{truthbenotproud.blogspot.com}
{myechochamber.blogspot.com}
{kiersten.connersax.com}
{enemykombatant.blogspot.com}
{swamiuptown.com}
{charlesking.blogspot.com}
{margieburns.com}
{conpro.blogspot.com}
{tuibguy.messagemonster.com}
{itsafterglow.blogspot.com}
{mediaprima.com/nv1962}
{nerofiddled.blogspot.com, saveoursenate.blogspot.com}
{howienfriends.blogspot.com}
{doublespeakez.blogspot.com}
{ergio.blogspot.com}
{dailyhowler.com}
{hategun.com}
{jchristopherfeng.com}
{littlegreenfootballs.com/weblog/weblog.php}
{hereswhatsleft.typepad.com/home}
{brianscott.blogspot.com}
{americankernel.com}
{wynterstoner.blogspot.com}
{americaindenial.com/ ?q=node/feed}
{whirlingbuffalo.com/creative-spirits.htm}
{itlooksliketh is.blogeasy.com}
{small-hands.blogspot.com}
{hipublican.com}
{liberationlearning.blogspot.com}
{journals.aol.com/mark24609/myblog}
{jointhecrowd.blogspot.com}
{politicaltracker.com/blog/weblog.php}
{rudyforfreeholder.org}
{spectatorsport.blogspot.com}
{home.earthlink.net/ juris_doctor_szh}
{posptect.org/weblog}
{tesibria.typepad.com/democracy_is_coming}
{airjun.blogspot.com}
{strangesemantics.blogspot.com}
{politicalwire.com}
{bentcorner.blogspot.com}
{mrleft.org}
{babylon.squarespace.com}
{abbadabbaduo.blogspot.com}
{bradcarson.com/blog}
{jetage.blogspot.com}
{dailyblurb.blogspot.com}
{freakinliberals.com/blog_a}
{uberdude.blogspot.com}
{homepage.mac.com/georgia.nesmith/iblog/index.html}
{tuckereskew.typepad.com}
{dems2004.com/blog}

{republicansbite.com}
{bigenchilada.org}
{kurtkk.blogspot.com}
{matthewyglesias.com}
{lefttimes.com}
{rightcrusade.blogspot.com}
{stinkbait.blogspot.com}
{workingpapers.blogspot.com}
{liberalreview.blogspot.com}
{summation.net}
{underscorebleach.net/content/jotsheet}
{thepoliticalanimal.com}
{renpro.kwcom.net}
{churchofcriticalthinking.com}
{achingandspinning.blogspot.com}
{x-pollen.com/many/blog.html}
{fuckthisblog.blogspot.com}
{reynardfox.com}
{njfordemocracy.org}
{probitus.obrien4.net}
{islanddave.blogspot.com}
{thepeach.blogspot.com}
{markmeador.blogspot.com}
{epubguy.blogs.com/epubguy}
{home.att.net/ dbarnhill}
{reporterette.com}
{demoliberal.blogspot.com}
{ironmouth.com}
{barcidhomily.com}
{smearbody.blogspot.com}
{johnnbrown.blogspot.com}
{wrighthot.com}
{tonyknowles.com/blog.html}
{conniptions.net}
{inez2004.com/blogs}
{cadat.nl}
{blackwingleftbird.blogspot.com}
{gloomanddoom.blogspot.com}
{therustybarrel.com/david/blog}
{allthingsmark.com}
{southernrightwingextremist.blogspot.com}
{paulchasman.com}
{flyinspacemonkey.mu.nu}
{crapiscommon.blogspot.com}
{newsnik.com}
{mc256.blogspot.com}
{polialaska.blogspot.com}
{planetdave.com/blogs.php3 ?mid=1}
{home.earthlink.net/ ur_blog}
{awards5.tripod.com/tarasblog}
{vandyconservative.blogspot.com}
{progressiveink.com/bloggers.php}
{dashnier.blogspot.com, janm.blogspot.com, michaelphillips.blogspot.com}
{blog.willrobinson47.com}
{ibrutus.com}
{johnkerrymustlose.com}
{zeph1z.tripod.com/blog}
{thislousytshirt.net}
{spydrzweb.blogspot.com}
{dems2004.org/blog}
{irregulartimes.com/blogger.html}
{praguewriter.typepad.com}
{bloghogger.blogspot.com}
{cecile.squarespace.com}
{quimundus.modblog.com}
{readingpost.com}
{rantingred.blogspot.com}
{uberddue.blogspot.com}
{ludovicspeaks.typepad.com/real_deal}
{dadougster.blogspot.com}
{mcwil.blogspot.com}
{elissa.typepad.com}
{quimundus.squarespace.com}
{americandreamslost.blogspot.com}
{marksteyn.com}
{nicoladellarciprete.blogs.com}
{rebelrouser.net/v-web/b2/index.php}
{patrickmcclarty.blogspot.com}
{soonerthought.com}
{winnegar.blog-city.com}
{leftwingnation.com}
{themeofabsence.com}
{rabner.blogspot.com}
{thedispatch.blogspot.com}
{larsonreport.com}
{emptyisform.com}
{johngalt.us}
{juandejesus.50megs.com/juancomments.html}
{steveguy.me.uk}
{politicalinsanity.com}
{njn-blogogram.blogspot.com}
{bluelemur.com}

{asilentrevolution.blogspot.com}
{pajamajournalist.blogspot.com}
{jewishworldreview.com}
{lucianne.com}
{blotts.org/polilog}
{astroturf.uni.cc}
{amradio.blogspot.com}
{johnnynobody.blogspot.com}
{home.earthlink.net/ jasider}
{aldaron.modblog.com}
{cruelkev.blogspot.com}
{politicalanimals.blogspot.com}
{senatorbarb.typepad.com}
{juntoboys.blogspot.com}
{jcchurch.blogspot.com}
{asmodeus-space.com/weblog/blog/index.php}
{vondollens.us}
{badattitudes.com/mt}
{xanga.com/lockjawtheogre}
{ndnblog.org}
{saxa.blogspot.com}
{markwarnerforpresident2008.blogspot.com}
{thesiliconmind.blogspot.com}
{changeforamerica.com/blog}
{amliberal.com/blog}
{jmbzine.com}
{gracefotography.com/graceblog/index.php}
{kristyandotherwise.blogspot.com}
{billrice.org}
{nykola.com}
{matthewstinson.com/blog}
{dividedloyalties.net}
{mindspring.com/ gsb3/blogger.html}
{sedenteanimo.com}
{weblogs.bakshi.us/redskunk}
{votervirginblog.blogspot.com}
{republog.net}
{annatopia.com/home.html}
{grownups.blogspot.com}
{fidlet.com}
{nuntius.blogspot.com}
{bellusbellum.info}
{a100wwe.blogspot.com}
{brunon.blogspot.com}
{julietterossant.com/superchefblog/superchefblog.html}
{politicswithrichard.blogspot.com}
{crankylittleblog.blogspot.com}
{revoltnow-us.blogspot.com}
{right-magazine.com/blog}
{gregpalast.com}
{smarthomeschool.com}
{newstrolls.com}
{martydee.com/poligov}
{home.att.net/ gentle.reader}
{translucence.org}
{warriorscreed.000k3.com}
{lahargis.blogspot.com}
{gadflyer.com}
{poptext.blogspot.com}
{politicalhyperbole.blogspot.com}
{thinkingright.blogspot.com}
{hellofromparis.blogspot.com}
{stephenbates.com/yellowdoggereldemocrat}
{theimmoralminority.blogspot.com}
{blosteam.blogspot.com}
{unusualhabits.blogspot.com}
{takeactionmichigan.blog-city.com}
{courtney5.us/drew}
{hotflashesfromthecampaigntrail.blogspot.com}
{the-torch.com}
{salestaxusa.com}
{powervacuum.org}
{heysucka.blogspot.com}
{in-sub-ordination.blogspot.com}
{dgci.blog-city.com}
{georgewbush.com/blog}
{snooty.binaryclub.com/wp-rdf.php}
{deanocrats.com}
{nebursworld.blogspot.com}
{ron-james.com/blog}
{dgci.net}
{web.utk.edu/ agroves}
{gaijincory.com}
{mcpheet.com}
{ilovecynics.com}
{theradiomansays.blogspot.com}
{hal9000report.blogdrive.com}
{bakshi.us/redskunk}
{marylandpolitics.blogspot.com}
{nelslindahl.net}
{eastbaykerry.com}
{liberaleric.blogspot.com}

{home.earthlink.net/ goodnoughp/index.html}
{revone.net/jarhead}
{markch.blogspot.com}
{84rules.blog-city.com}
{jamiefeyrer.com}
{justmark.com}
{gotzfamily.org/weblog}
{futuraworkz.com}
{carpetblogger.com}
{swiftreport.blogs.com}
{bushlies.net/pages/10/index.htm}
{kushreport.com}
{centerpiece.blogspot.com}
{randz.net}
{punditician.blogspot.com}
{ludis.blogspot.com}
{livejournal.com/users/hoptkov}
{trainorphans.org}
{ibe.blogspot.com}
{homepage.mac.com/jpbarr/myblog}
{gwbblows.blogspot.com}
{livejournal.com/ dailyapple}
{sonsoftherepublic.com}
{theapologist.blogspot.com}
{criticalbill.blogspot.com}
{totels.com/bush04}
{blog.glinka.com}
{hoeffelforsenate.com/blog}
{missourikid.blog-city.com}
{eltcamerica.blogspot.com}
{neonbrainiac.nu/mt/blog}
{brendonsspiderhole.blogspot.com}
{livejournal.com/users/bambenek}
{theaggpro.blogspot.com}
{bowles2004.com/weblog}
{smashingzero.blogspot.com}
{truthprobe.blogspot.com}
{onhappiness.crimsonblog.com}
{bradfriedman.com/bradblog}
{newsjungle.blogspot.com}
{trueconservative.typepad.com}
{cjh9999.blog-city.com}
{nickgerman.blogspot.com}
{someonetakethewheel.blogspot.com}
{chainik_hocker.blogspot.com}
{blog.mytechaid.com}
{roil.blogspot.com}
{georgewbush.com}
{usademocrazy.blogspot.com}
{tweetpetite.blogspot.com}
{notbush.com}
{pollsterseek.blogspot.com}
{cmillertime.blogspot.com}
{garrulus.ilcannocchiale.it}
{campaignofone.org/blog/index.asp}
{penheaded.blogspot.com}
{godubya.com}
{dems2000.net}
{topviewnews.com/blog}
{firstdownpolitics.com}
{smirkingchimp.com}
{voiceofthewebb.blogspot.com}
{iwantmycountryback.org}
{doubter.blogspot.com}
{vapolitics.info}
{boycottsbg.com}
{wildweasel.blog-city.com}
{blogitics.com}
{tinylittledots.com}
{myducksoup.com/blog/bloglib/bloglib.shtml}
{xanga.com/eugene3}
{nosmallplans.com/rants}
{labandadelbuko.ilcannocchiale.it}
{rant.st}
{andrightlyso.blogspot.com}
{brutallyhonest.org}
{tobyemden.blogspot.com}
{variousorthodoxies.com}
{conservativepolitics.blogspot.com}
{angryfinger.org}
{greendogdemocrat.blogspot.com}
{stzman.org/blog}
{mattresspolice.com}
{counterspin.blogspot.com}
{minagahet.blogspot.com}
{sethdmichaels.com}
{gabeanderson.com/life}
{etalkinghead.com}
{lendmesomesugar.blogs.com/shut_up_already}
{adamtalib.blogspot.com}
{40ozblog.blogspot.com}
{talesofawanderingmind.blogspot.com}

{karlhenderson.blogspot.com}
{martadonayre.com}
{democratreport.blogspot.com}
{mobile.blogcastlive.com}
{myaisling.blogspot.com}
{bettyblog.com}
{rightwingdude.squarespace.com}
{skinmeister.name}
{sarebabe.blogdrive.com}
{shetquaker.blog-city.com}
{bloganonymous.com}
{scripting.com}
{thingsasitis.net}
{thebluestate.com}
{limecoconut.blogspot.com}
{global-equality.org/news/blog/index.shtml}
{meckz.net}
{greaterdemocracy.org}
{insanepreschoolmom.blogspot.com}
{alternateworlds.blogspot.com}
{spacedog26.blogspot.com}
{hijkselur.blogspot.com}
{republicanpower.blogspot.com}
{garbage-house.com}
{depressedlonelybored.typepad.com}
{tepid-observer.blogspot.com}
{rightreadingroom.com/index.php}
{republitarian.org}
{wtf.mindsay.com}
{tedlinczak.com}
{fairshot.org}
{delraysteve.com/blog}
{d-42.com}
{obrien4.net/probitus}
{tomdaschle.com/tom/index.php ?id=34}
{grassrootsnation.com}
{conservativevor.blogdrive.com}
{halcyondays.info}
{lagernik.blogspot.com}
{loosecoin.blogspot.com}
{inmyrightmind.com}
{nationalreview.com/thecorner/corner.asp}
{digitaljay.blogspot.com}
{moxiegrrrl.com}
{progressiveink.com}
{blog.mintruth.com}
{stonyfield.com/weblog/bovinebugle/index.html}
{republicanresearchcouncil.blogspot.com}
{chestnut_tree.blogspot.com}
{home.earthlink.net/ heavy-duty}
{celluloid-wisdom.com/pw/}
{selfish-souls.com/blog}
{maultasch.us}
{nephandus.com/blog.php}
{lancehaun.com}
{heathenfolk.warblogs.com/blog}
{oliphantparts.org/blog}
{meanspirit.blogspot.com}
{politicalmonitor.us/blog}
{leftnuts.com/runcms/modules/news}
{cubanstories.blogspot.com}
{rebrand-democrat.txfx.net}
{members.cox.net/aaronjohn}
{planetsuz.blogspot.com}
{greentit.com}
{edmondthehun.blogspot.com}
{members.aol.com/ourprez08}
{shortfatcorporal.blogspot.com}
{multitudes.typepad.com}
{jbcougar.blogspot.com}
{blog.dennisfox.net}
{miss-sassafras.com}
{lastdaysoftherepublic.fatoprofugus.net}
{rockpaperscissors.blogs.com}
{blog.iainlindley.com}
{joebrent.blogspot.com}
{kerrytalkingpoints.blogspot.com}

Recherche des classes d'équivalence appliquée à un réseau d'interactions sociales d'un groupe de moines dans un monastère [78] (7 classes)

{JOHN_1, MARK_7, SIMP_18}
{AMAND_13, BASIL_3, BONI_15, GREG_2, HUGH_14}
{ALBERT_16, BERTH_6, PETER_4, VICTOR_8}
{WINF_12}

{AMBROSE_9, BONAVEN_5, ROMUL_10}
{LOUIS_11}
{ELIAS_17}

Bibliographie

[1] Lada A. Adamic and Natalie Glance. The political blogosphere and the 2004 us election. 2005.

[2] E. Almaas, R. V. Kulkarni, and D. Stroud. Characterizing the structure of small-world networks. *Phys. Rev. Letters*, vol 88 numero 9, 2002.

[3] Sofiane Ben Amor. *Percolation, prétopologie et multialéatoires,contributions à la modélisation des systèmes complexes : exemple du contrôle aérien*. PhD thesis, EPHE, Paris, 2008.

[4] Soufian Ben Amor, Vincent Levorato, and Ivan Lavallée. Generalized percolation processes using pretopology theory. In *Proceedings IEEE of RIVF'07*, pages 130–134, Hanoï, Vietnam, 2007. IEEE Society.

[5] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks : Membership, growth, and evolution. *Proc. 12th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2006.

[6] Albert-Laszlo Barabasi and Réka Albert. Emergence of scaling in random networks. *SCIENCE*, Vol 286 :509–512, 1999.

[7] Z. Belmandt. *Manuel de prétopologie et ses applications : Sciences humaines et sociales, réseaux, jeux, reconnaissance des formes, processus et modèles, classification, imagerie, mathématiques*. Hermes Sciences Publications, 1993.

[8] Edward A. Bender and E. Rodney Canfield. The asymptotic number of labeled graphs with given degree sequences. *Comb. Theory,*, 3 :296–307, 1978.

[9] Charles H. Bennett, Peter Gacs, Ming Li, Paul M.B. Vitanyi, and Wojciech H. Zurek. Information distance. pages 21–30, San Diego, California, USA, 1993.

[10] Béla Bollobás. Random graphs. *Academic Press*, 1985.

[11] Béla Bollobás and Oliver Riordan. Slow emergence of the giant component in the growing m-out graph. 2004.

[12] Phillip Bonacich. Power and centrality : a family of measures. *American Journal of Sociology*, 92 :1170–1182, 1987.

[13] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. Waterloo, Canada, 1976.

[14] Stéphane Bonnevay. *Analyse de texture : marqueurs et prétopologie*. PhD thesis, Université Lyon 1, 1997.

[15] Stéphane Bonnevay, Michel Lamure, Christine Largeron, and Nicolas Nicoloyannis. A pretopological approach for structuring data in non-metric spaces. *Electronic Notes in Discrete Mathematics*, 2, 1999.

[16] A. Bounekkar. *Analyse de texture et statistique spatiale*. PhD thesis, Université Lyon 1, 1997.

[17] N. Bourbaki. *Topologie générale*. Hermann, 1971.

[18] Trong-Lieu Bui and P. Huard. La méthode des centres dans un espace topologique. *Numerishe Mathematik*, 8 :56–67, 1966.

[19] Geoffrey Canright, Andreas Deutsch, Mark Jelasity, and Frederick Ducatelle. *Structures and Functions of Dynamic Networks*. PhD thesis, 2004.

[20] Ed. Cech. *Topological Spaces*. John Wiley and Sons, 1966.

[21] Rudi Cilibrasi and Paul M.B. Vitanyi. Clustering by compression. *TRANSACTIONS ON INFORMATION THEORY*, 51 :1523–1545, 2005.

[22] Colin Cooper and Alan Frieze. The cover time of the giant component of a random graph. 2006.

[23] Monique Dalud-Vincent. *Modèle prétopologique pour une méthodologie d'analyse des réseaux : concepts et algorithmes*. PhD thesis, Université Claude Bernard - Lyon 1, Lyon, 1994.

[24] Alain Degenne. Entre outillage et théorie, les réseaux sociaux. *Réseaux Sociaux de l'Internet*, 2004.

[25] Alain Degenne and Michel Forsé. *Les réseaux sociaux*. Armand Colin, Paris, 1994.

[26] J.P. Delahaye. *Information, complexité et hasard*. Paris, 1994.

[27] Narsingh Deo and Pankaj Gupta. Graph-theoritic web algorithms : an overview. In *Lecture Notes In Computer Science*, volume Vol 2060, pages 91 – 102. Springer Berlin / Heidelberg, 2001.

[28] Rick Durrett and Paul Jung. Two phase transitions for the contact process on small worlds. 2006.

[29] Gérard Duru, Jean-Paul Auray, and Michel Mougeot. Mathématiques appliquées aux sciences sociales - tome 1 : Relations et applications. 1, 1979.

[30] Bruce Eckel. *Thinking in Java*. Prentice Hall PTR, 1998.

[31] *Pierre Mercklé*. **Les réseaux sociaux :** *les origines de l'analyse des réseaux sociaux*. 2004.

[32] Paul Erdös and Alfréd Rényi. On random graphs. *Publicationes Mathematicae*, 6 :290–297, 1959.

[33] M. Fréchet. *Espaces abstraits*. Hermann, Paris, 1928.

[34] Thomas Fuhrmann. Small-world networks revisited. In *Lecture Notes in Computer Science*, volume Vol 2877, pages 80–92, 2003.

[35] Michelle Girvan and M. E. J. Newman. Community structure in social and biological networks. 2001.

[36] Roger Guimerà and Marta Sales-Pardo. Form follows function : the architecture of complex networks. *Molecular Systems Biology*, 2006.

[37] Tag Gon Kim Herbert Praehofer Bernard P. Zeigler. *Theory of Modeling and Simulation*. 2000.

[38] Tommy R. Jensen and Bjarne Toft. Graph coloring problems. *Wiley-Interscience*, 1995.

[39] Jon M. Kleinberg. The small-world phenomenon : An algorithmic perspective. 1999.

[40] Jon M. Kleinberg. Navigation in a small world. *NATURE*, Vol 406 :845, 2000.

[41] Jon M. Kleinberg. Bursty and hierarchical structure in streams. 2001.

[42] Jon M. Kleinberg. Small-world phenomena and the dynamics of information. 2001.

[43] Jon M. Kleinberg. The small-world phenomenon and decentralized search. *The Mathematics of Networks*, Vol 37, 2004.

[44] Donald Knuth. Section 6.4 : Hashing. In Third Edition. Addison-Wesley, editor, *The Art of Computer Programming, Volume 3 : Sorting and Searching*, pages 513–558. 1997.

[45] Annick Lesne Michel Laguës. Chapitre 3 : L'universalité comme conséquence de l'invariance d'échelle. In Belin, editor, *Invariance d'échelle - Des changements d'états à la turbulence*, pages 58–105. Paris, 2003.

[46] Michel Lamure. *Espaces abstraits et reconnaissance des formes, application aux images digitales*. PhD thesis, Université Lyon 1, 1987.

[47] Christine Largeron and Stéphane Bonnevay. Une méthode de structuration par recherche de fermés minimaux : application à la modélisation de flux de migrations inter-villes. Lyon, France, 1997.

[48] Christine Largeron and Stéphane Bonnevay. A pretopological approach for structural analysis. *Information Sciences*, 144 :169 – 185, 2002.

[49] Ivan Lavallée. *Complexité et algorithmique avancée : une introduction*. Hermann, Paris, 2008.

[50] W. David Kelton Averill M. Law. *Simulation Modeling and Analysis*. 2000.

[51] Emmanuel Lazega. *Réseaux sociaux et structures relationnelles*. Presses Universitaires de France, Versailles, 1998.

[52] Thanh Van Le. *Classification prétopologique des données Application à l'analyse des trajectoires patients*. PhD thesis, Université Lyon 1, Lyon, 2007.

[53] Thanh Van Le, Nadia Kabachi, and Michel Lamure. A new clustering method associated pretopological concepts and k-means algorithm. In *Proceedings of the International Conference on Research Innovation and Vision for the Future*, Hanoï, 2007.

[54] Emmanuelle Lebhar. *Algorithmes de routage et modèles aléatoires pour les graphes petits mondes*. PhD thesis, Ecole Normale Supérieure de Lyon, 2005.

[55] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Laws of graph evolution : Densification and shrinking diameters. 2006.

[56] Vincent Levorato and Murat Ahat. Modélisation de la dynamique des réseaux complexes associée à la prétopologie. In *Proceedings of ROADEF'08*, pages 299–300, Clermont-Ferrand, France, 2008.

[57] Vincent Levorato and Marc Bui. Modeling the complex dynamics of distributed communities of the web with pretopology. In *Proceedings of the 7th International Conference on Innovative Internet Community Systems*, Munich, Germany, 2007.

[58] Vincent Levorato and Marc Bui. Data structures and algorithms for pretopology : the java based software library pretopolib. In *Proceedings of the 8th International Conference on Innovative Internet Community Systems*, pages 122–134, Schoelcher, Martinique, 2008. IEEE.

[59] Ming Li, Xin Chen, Xin Li, Bin Ma, and Paul M.B. Vitanyi. The similarity metric. *Transactions on Information Theory*, 50(12) :3250 – 3264, 2004.

[60] Ming Li and Paul Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. 1997.

[61] Pierre Lopez. Graphes. 2005.

[62] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *University of California Press*, pages 281–297, Berkeley, 1967.

[63] Michel Lamure Marcel Brissaud Monique Dalud-Vincent. Fermés et fermetures en prétopologie. 2002.

[64] Kurt Mehlhorn and Stefan Näher. Leda : A library of efficient data types and algorithms. *GI Jahrestagung*, 1 :35–39, 1989.

[65] Stanley Milgram. The small world problem. *Psychology Today*, 1, 1967.

[66] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 6 :161–180, 1995.

[67] Jardine N. and Sibson R. The construction of hierarchic and non-hierarchic classifications. *The Computer Journal*, 11 :117, 1968.

[68] M. E. J. Newman. Random graphs as models of networks. *Handbook of Graphs and Networks*, 2002.

[69] M. E. J. Newman. The structure and function of complex networks. 2003.

[70] M. E. J. Newman, C. Moore, and D. J. Watts. Mean-field solution of the small-world network model. *Physical Review Letters*, Vol. 84, No. 14 :3201–3204, 2000.

[71] Mark Newman, Albert-László Barabási, and Duncan J. Watts. *The Structure and Dynamics of Networks*. Princeton University Press, 2006.

[72] Aurel Page. Complexité de kolmogorov et notion de mot aléatoire. 2007.

[73] Jean paul Auray. *Contribution à l'analyse des structures pauvres*. PhD thesis, Université Lyon 1, Lyon, 1982.

[74] Gilbert Babin Peter Kropf Jean Vaucher and Thierry Jouve. Experimenting with gnutella communities. *Scientific series*, 55, 2002.

[75] John Plaice, Peter G. Kropf, Peter Schulthess, and Jacob Slonim. Distributed communities on the web. *DCW 4th International Workshop*, 2002.

[76] Solla Price and Derek J. Networks of scientific papers. *SCIENCE*, 149 :510–515, 1965.

[77] Stan Salvador and Philip Chan. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. pages 576–584, 2004.

[78] S.F. Sampson. *A Novitiate in a Period of Change. An Experimental and Case Study of Social Relationships.* PhD thesis, Cornell University, 1968.

[79] Steven H. Strogatz. Exploring complex networks. *NATURE*, Vol 410 :268–276, 2001.

[80] Shiwei Sun, Lunjiang Ling, Nan Zhang, Guojie Li, and Runsheng Chen. Topological structure analysis of the protein-protein interaction network in budding yeast. *Nucleic Acids Research*, 31 :2443–2450, 2003.

[81] Bo Söderberg. A general formalism for inhomogeneous random graphs. *Phys. Rev. E*, 2002.

[82] Duncan J. Watts. Kevin bacon, the small-world, and why it all matters. *SANTA FE INSTITUTE BULLETIN • FALL*, 1999.

[83] Duncan J. Watts, Peter Sheridan Dodds, and M. E. J. Newman. Identity and search in social networks. 2002.

[84] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *NATURE*, Vol 393 :440–442, 1998.