Network partitioning algorithms with scale-free objective

Nicolas Martin

▶ **To cite this version:**

Nicolas Martin. Network partitioning algorithms with scale-free objective. Automatic Control Engineering. Université Grenoble Alpes [2020-..], 2020. English. NNT : 2020GRALT001 . tel-02532058v3

HAL Id: tel-02532058
https://hal.science/tel-02532058v3

Submitted on 17 Jul 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ GRENOBLE ALPES

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE ALPES

Spécialité : **Automatique**

Arrêté ministériel : 7 août 2006

Présentée par
Nicolas MARTIN

Thèse dirigée par **Carlos CANUDAS-DE-WIT** et
codirigée par **Paolo FRASCA**

préparée au sein du
GIPSA-lab
dans **l'école doctorale Electronique Electrotechnique Automatique & Traitement du signal (EEATS)**

Network partitioning algorithms with scale-free objective

Thèse soutenue publiquement le **19 Février 2020**,
devant le jury composé de:

Myriam PREISSMANN
Directrice de recherche, G-SCOP, Présidente du jury
Pierre-Alexandre BLIMAN
Directeur de recherche, INRIA, Rapporteur
Christophe CRESPELLE
Maitre de conférence, Université Claude Bernard Lyon 1, Rapporteur
Jacquelien Scherpen
Professor, University of Groningen, Examinatrice
Paolo FRASCA
Chargé de recherche, GIPSA-lab, Co-directeur de thèse

UNIVERSITÉ DE GRENOBLE ALPES

EEATS

Electronique Electrotechnique Automatique & Traitement du signal

T H È S E

pour obtenir le titre de

docteur en sciences

de l'Université de Grenoble Alpes

Mention : AUTOMATIQUE

Présentée et soutenue par

Nicolas MARTIN

Network partitioning algorithms with scale-free objective

Thèse dirigée par Carlos CANUDAS-DE-WIT

GIPSA-lab

soutenue le 19/02/2020

Jury :

Rapporteurs :	Pierre-Alexandre BLIMAN	-	Directeur de recherche, INRIA
	Christophe CRESPELLE	-	Maitre de conférence, Université Claude Bernard
Co-directeur :	Paolo FRASCA	-	Chargé de recherche, GIPSA-lab
Présidente :	Myriam PREISSMANN	-	Directrice de recherche, G-SCOP
Examinatrice :	Jacquelien SCHERPEN	-	Professor, University of Groningen

Résumé

L'analyse, l'estimation et le contrôle de systèmes dynamiques évoluant sur des réseaux se confronte à des problèmes de complexité lorsque le système considéré devient trop grand. Pour faire face à ces problèmes d'échelles des méthodes de réduction de modèle sont utilisées. Cette thèse s'intéresse particulièrement à une méthode de réduction de réseaux en particulier : le partitionnement. Cette méthode consiste à trouver une partition des noeuds du réseaux induisant un réseau de dimension réduite ayant certaines caractéristiques. Afin d'utiliser le réseau réduit comme modèle du système initial, toute méthode de réduction vise à préserver certaines propriétés du réseaux initial. La particularité de notre travail est d'imposer en plus au réseau réduit certaines propriétés de notre choix. Cette approche permet de bénéficier des propriétés du réseau réduit pour l'application qui en sera faite par la suite en particulier l' observation, l'estimation ou le contrôle du système. En particulier une propriété d'intérêt est appelée *scale-free*[1]. Un réseau est dit scale-free si il possède quelques noeuds, appelés *hub*, avec un très grand nombre de connexions mais que la plupart de ces noeuds ont très peu de connexions. Précisément un réseau est dit scale-free si sa distribution de degrés suit une loi de puissance. Cette thèse s'articule autour de cette problématique et est composée de six parties principales détaillés ci-dessous: Tout d'abord une introduction permet de définir un cadre formel à ce travail, de le recontextualiser et d'apporter certains préliminaires sur les propriétés du réseaux que l'on veut préserver ou imposer et notamment la propriété scale-free. Dans le premier chapitre nous étudions l'impact d'une contrainte de connexité dans un problème de partitionnement. Le deuxième chapitre explore le problème de partitionnement assurant que le réseau réduit soit scale-free. Dans le troisième chapitre nous proposons d'imposer au réseau réduit une certaine propriété assurant que l'évolution du système peut être reconstruite dans une certaine mesure. Enfin les quatrième et cinquième chapitres sont des applications des résultats précédents respectivement au réseau de trafic et au réseau d'épidémie. La thèse se termine sur une conclusion revenant sur les principales contributions et offre quelques perspectives.

Le premier chapitre concerne un problème que nous avons dénommé le *prix de la connexité*. Lorsqu'on considère un problème de partitionnement de réseau où la meilleur partition doit être trouvée afin d'optimiser un certain objectif il est possible d'y ajouter une contrainte de connexité. Dans ce cas, les noeuds au sein d'une même partie de la partition doivent être connectée entre eux. Cette contrainte est pertinente par exemple lorsqu'on considère un réseau de trafic (ou n'importe quel réseau avec une géographie sous-jacente) et que l'on souhaite que chaque partie de la partition corresponde à une zone géographique. Ajouter cette contrainte va nécessairement diminuer la qualité de la réduction au regard de l'objectif fixé. La dégradation de la solution est ce que l'on appelle le prix de la connexité. Nous proposons ensuite d'estimer sa valeur pour un réseau donné et pour n'importe quel

[1]Certains ouvrages en français parlent d'*invariance d'échelle* mais le terme anglais est le plus souvent conservé.

problème de partitionnement. Pour ce faire, on compare la taille des ensembles de recherche du problème et du problème contraint qui sont respectivement l'ensemble des partitions du réseau et l'ensemble des partitions connexes du réseaux. Le rapport entre la cardinal de ces deux ensembles est appelé *rapport de connexité*. La suite du chapitre vise à estimer cette valeur et en particulier nous obtenons une borne supérieure dans le cas de réseau Erdős–Rényi. Finalement, nous montrons expérimentalement que cette borne est très proche de la vraie valeur du rapport de connexité calculée expérimentalement.

Le second chapitre pose le problème de partitionnement de réseau induisant une structure scale-free. Comme dit précédemment, il est nécessaire de préserver également des caractéristiques du réseau initiale. Étant donné un réseau initial, nous proposons donc un *méta-problème* consistant à trouver la meilleure partition en terme de scale-free sous les contraintes de préservations de propriété. Nous proposons ensuite un *méta-algorithme* donnant le squelette d'une méthode pour obtenir une solution suboptimal à ce type de problème. Nous résolvons ensuite un premier cas simple de ce type de problème pour des raisons essentiellement didactiques. Puis nous nous intéressons un problème plus complexe préservant certaines propriétés dynamiques du réseau initial à savoir: la conservation de masse, la centralité de vecteur propre et la masse totale du système. Des résultats mathématiques sont établis permettant d'établir un algorithme sub-optimal pour ce problème.

Le troisième chapitre s'intéresse à la possibilité de reconstruire l'évolution d'un état agrégé d'un réseau. En considérant un système linéaire invariant dans le temps, on défini la notion de détectabilité moyenne comme la possibilité de pouvoir reconstruire la valeur moyenne de certaines zones du réseau à partir de quelques mesures. Le problème qui nous intéresse est donc de trouver une partition du réseau entre une partie observée et d'autres parties non-observées assurant la détectabilité moyenne. Un premier résultat montre que pour un type de système particulier une condition suffisante à la détectabilité est la *régularité* des parties non-observées ce qui signifie qu'au sein de chacune de ces parties les noeuds doivent tous avoir le même degrés sortant. Un algorithme est ensuite proposé pour trouver une telle partition. Cependant ce problème de détection de sous-graphes réguliers est fondamentalement complexe entre autre parce que c'est un problème NP, que les meilleurs solutions sont souvent des réseaux trop petits et que les hypothèses sur le système de départ sont très contraignante. Ceci étant dit, nous proposons ensuite une version relaxée du problème où l'on ne cherche plus la régularité exacte du sous-graphe mais une certaine forme de quasi-régularité. En effet, nous montrons que l'erreur de reconstruction de la moyenne de chaque sous-graphe peut être borné par une fonction de l'erreur de la régularité. A partir de ce résultat nous proposons un algorithme fournissant une partition en sous-graphe quasi-régulier.

Les deux derniers chapitres sont des applications des résultats théoriques présentés dans les trois premiers chapitres. Le quatrième chapitre s'intéresse au trafic routier de l'agglomération grenobloise. Nous montrons qu'à partir du grand réseau initial (environ 19000 noeuds), les résultats du chapitre deux permettent d'obtenir un réseau réduit avec une structure scale-free.

Des simulations montrent ensuite qu'il est possible d'utiliser ce réseau réduit pour reconstituer efficacement la dynamique du système initial. Le cinquième chapitre propose deux applications en lien avec l'épidémiologie, la science qui étudie la propagation de maladie dans une population et que nous présentons dans une première partie du chapitre. La première application s'intéresse à la question de stratégie de vaccination. Étant donnés un réseau représentant une population et un nombre limité de vaccin, on cherche les individus les plus intéressant à vacciner (et donc retirer du réseau) afin d'enrayer la propagation de la maladie. Des résultats bien connus pour les réseau scale-free propose de vacciner les hubs c'est à dire les individus fortement connectés. Pour les raisons plus homogènes, nous proposons d'utiliser l'algorithme du deuxième chapitre pour obtenir un réseau réduit scale-free et ainsi identifié certaines zone ayant un rôle de hub au sein du réseau initiale. Cette stratégie est comparé à d'autres stratégies et nous montrons qu'elle est jusqu'à deux fois plus efficace pour une large plage de paramètre du modèle. Enfin la dernière application considère la propagation d'une maladie au sein d'un large réseau d'individu recouvrant le territoire français. En utilisant les résultats du troisième chapitre, on montre qu'il est possible de diviser la population en plusieurs zones dont l'évolution moyenne peut être estimée efficacement à partir de l'observation d'une petite partie des individus.

Remerciements

Je tiens à remercier sincèrement celles et ceux qui directement ou indirectement ont permis la réalisation de ce travail. Une thèse c'est une douce tempête que l'on traverse et qu'il est difficile de décrire sans l'avoir vécu. Aujourd'hui, au crépuscule de cette épreuve, nombreux sont ceux que je veux citer ici, pour leur accompagnement, leur amitié, leur soutien durant cette thèse d'une part, mais également au long du chemin qui m'a mené jusqu'ici.

Et puisqu'ils sont à l'origine de tout, tout d'abord merci Papa et Maman pour votre amour inconditionnel. Je comprends mieux aujourd'hui les sacrifices que vous avez fait pour nous donner à Manon et moi toutes les chances dans la vie. Chaque page de cette thèse vous est due puisque je n'aurais rien fait sans vous. Manon, chère soeur, ma thèse t'intrigue autant qu'elle te parait douteuse, je te remercie aussi de me l'avoir fait remarquer. Tu as raison, c'est probablement pas normal de rester 3 ans devant un ordinateur à chercher d'obscures algorithmes. Tu te pares de toutes les couleurs du monde, laisse aux autres le choix de trouver celle qui leur plaît. Merci au reste de ma famille, oncles et tantes, cousines et cousins, grand-parents, *los de qui cau*. Si ma thèse m'a confié aux Alpes bienveillantes, mon coeur est toujours dans les Pyrénées.

Puisqu'ils sont ma seconde famille, ou ma famille de second degré, merci aux Chevaliers. Vous seuls comprenez qu'une vanne peut durer trois ans. Nos retrouvailles, à califourchon sur le grand sud ou en amazone dans les Balkans, à aiguiser nos vannes, à griffer la route et à assécher nos écuelles sont toujours un grand ressourcement, à défaut d'être d'un quelconque repos. Après tant de temps en exil, ma lame reviendra très vite s'aligner aux vôtres. David, Maxime, Mélanie, Cyril, Botch, Cauchy[2], Mael, Crevette, Elsa, Mariette, Simon, Doat, Pinoges, Elsa, Doralice, Paul, Stynen et les autres nos aventures ne font que commencer...

Merci aussi à celles et ceux que j'ai croisé durant mes études et qui ne m'ont jamais dit quand il fallait s'arrêter. En classe préparatoire au lycée Louis Barthou d'abord, il y a au moins un siècle, où je découvrais de nouvelles notions : le travail, la fatigue, les espaces pré-hilbertiens, la solidarité et la mauresque. Les moments de répit y étaient rares mais intenses. À l'INSA ensuite, où les moments de répit étaient fréquents et intenses. J'y ai découvert l'émancipation, la légèreté, les responsabilités de la vie de jeune adulte et les irresponsabilités de la vie étudiante. Merci à celles et ceux qui ont partagé mes journées studieuses. Je pense aux membres du GMM, professeurs et étudiants, qui offraient une formidable ambiance de travail. Merci également à mes binômes successifs qui ont écopé de la lourde tâche de me mettre au travail. Merci à celles et ceux, plus nombreux, qui ont partagé mes nuits festives. Notamment la *Grande Famille* qui est née parmi les NE, s'est déployée dans la Team ONE et perdure en ANE: Paul, Marine, Yoann, Etxe,

[2]Et hop ! Un justificatif pour l'état civil

Laureen, Aurélie, Pauline, Mambu, Youssouf, Quentin, Seb, Mike, Steevens, Soizick et tant d'autres. Celles et ceux qui sont toujours à mes côtés, celle et ceux qui n'y sont plus. Si vous avez contribué à bien des facteurs d'échec dans ma scolarité, ce n'était pour sur pas la solitude.

S'ils sont si nombreux ceux que j'ai quitté, c'est que d'autres ont su m'accueillir quand je suis arrivé à Grenoble. Seul avec mon gros sac à dos. Ils sont nombreux ceux que je dois remercier et qui font qu'aujourd'hui je suis (presque) un vrai Grenoblois qui monte régulièrement à la Bastille, qui va *grimper* à la salle et qui boit de la Chartreuse au petit-déj'. Evidemment tout d'abord un grand merci à mes premières colocs, Margaux et Noémie, qui m'ont accueilli, fait découvrir la ville et fait faire de magnifique rencontres. Puis il y a eu de nouveaux colocs et de nouvelles rencontres. Merci à vous tous, ceux qui sont là depuis trois ans ou ceux que je n'ai croisé que quelques mois. Dur de faire un remerciement global tant ma relation à chacun est différente, mais merci à tous ceux qui ont fait que cette ville est devenu mon chez moi: Simon, Bébou, MP, Leïla, Nina&Maé, Mouss, Anna, Sarah A., Totti, Chloé, Tristan, Sarah V., Baptiste, Gemma, Robin, Anne, Myrtille, Cantalou, Flo, Lulue, Mel, ... je vous disais coloc, pote de coloc, pote de pote de coloc ou encore coloc de pote de coloc mais maintenant je dis mes copains, et c'est quand même beaucoup plus court. Certains savent qu'il y a eu des moments difficiles où la tempête soufflait plus fort, ils se reconnaîtront et je suis profondément reconnaissant pour leur présence. Et à ceux qui traversent aussi leurs orages, restez fort, tout fini par passer. Au milieu de ces trois ans en terre dauphinoise, il y a aussi eu une escapade de quelques mois au Japon où j'ai rencontré une sympathique horde de doctorant français avec qui on a partagé quelques galères mais surtout beaucoup de bonheur, de voyages et une deuxième étoile ! Je les en remercie chaleureusement ! Parmi eux, je dois remercier tout particulier Léa-san. Si il y a une personne envers qui l'achèvement de cette thèse est redevable c'est toi. Tu as su me remotiver à aller au bout quand l'air Japonnais me donnait envie de prendre le large (je verrais dans quelques années si je dois te remercier pour ça). Merci pour ta présence, ta douceur et nos grandes discussions qui j'en suis sur sont loin d'être finies.

Enfin, le parcours de ces remerciements se terminent au plus près de ma thèse, au bout du couloir F de l'INRIA Grenoble où est installé l'essentiel de mon équipe de recherche NECS. Merci à vous tous d'voir été là au quotidien. D'une part, pour mes recherches et d'autre part pour tous les autres bons moments autour. Merci tout d'abord à mes directeurs de thèse Carlos et Paolo, vos conseils et vos soutiens m'ont guidé tout au long de mon travail. Votre complémentarité m'a fourni le juste équilibre entre rigueur et autonomie et m'a permis de fournir un travail que je crois de qualité. Merci également aux autres permanents qui m'ont offert un cadre de travail bienveillant. Merci enfin à tous les non-permanents que j'ai croisés pendant ces trois ans et avec qui j'ai partagé, au delà d'un lieu de travail, de très bons moments, des heures de jeux et beaucoup de discussions: Rémy, Hannah, Vadim, Pietro, Andres, Giacomo, Baptiste, Sebin, Thibault, Thibaud, Martin, Julie, Bassel, Umar, Denis, Liudmila, Ujjwal, Diego, Tianyi vous m'avez fait parcourir le monde et je ressors grandi de chacune de vos rencontres. Un merci particulier à Stéphane avec qui j'ai partagé mon bureau, trois ans de thèse et de longues discussions. Bon vent à toi, docteur, pour tes nouvelles

aventures.

J'ai profité de cette tribune que je me suis offerte au prix de trois an de thèse pour dire à l'écrit ce que je dis mal à l'oral. Je me suis permis de sortir du cadre de la thèse mais je crois que je suis (et que mon travail est) le fruit de toutes ces rencontres et de tous ces moments. Merci encore à tous ceux que j'ai cité et ceux que j'ai certainement oubliés. Les quelques lignes ci-dessus comptent pour moi au moins autant que la centaine de pages à venir.

I wish to sincerely thank the jury members of my thesis: Myriam Preissmann the president of the jury, Jacquelien Scherpen examiner, Paolo Frasca who was also the co-supervisor of my thesis and the reviewers Pierre-Alexandre Bliman and Christophe Crespelle for generously offering their time and support for the last steps of my thesis. In particular, I thank the two reviewers whose careful reading and pertinent comments allowed me to significantly improve the quality of my work.

Au monde qui suit.

Contents

List of Figures

List of Tables

Introduction

L'art de donner le même nom à des choses différentes

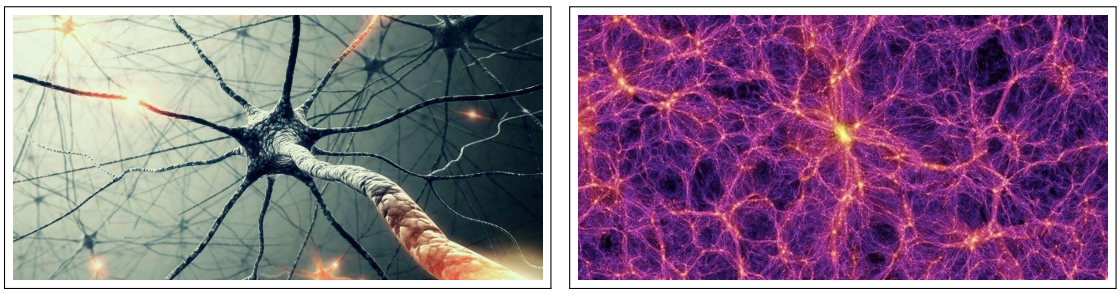
Henri Poincaré

0.1 Motivation to network reduction

0.1.1 The world as a network

A network is a system composed of elements interacting together. This definition is not precise as the terms *system*, *elements* and *interacting* are in a certain way blurry. However, it seems to be a good definition and if the terms are not clear it is because networks include a very vast range of systems. *Networks are everywhere.* This statement is so used in literature that it has almost lost its sense, but, literally, networks are everywhere:
Humans are connected within societies forming networks where the links may be friendships, family ties or work relationships. The brains of these humans are composed of billions of connected neurons, and each of these neurons are composed by molecules interacting together via chemical reactions. Besides the brain, the human body contains several other networks such as circulatory, nervous and lymphatic systems. Thanks to their brains, humans build road networks, electrical networks and telecommunication networks. In addition to humanity, animal interactions and physical phenomenon can also be interpreted as networks. Networks are everywhere, inside and around us.

While it includes a large variety of different systems, networks can be modeled and studied



(a) The brain network is composed of billions of interconnected neurons

(b) The cosmic web describes the distribution of matter in the universe

Figure 1: From the hidden recesses of our brain to the structure of the space, a large variety of systems can be viewed as networks.

within a same mathematical formalism. The parenthood of this formalism is often attributed to the Swiss mathematician Leonard Euler and its problem of the seven bridges of Königsberg in 1736. This problem asks us to find a route in the city of Königsberg which crosses once and only each of the seven bridges. While at first view the problem seems far from the network theory, Euler used mathematics to bring a solution leading to the foundation of the graph theory[3]. However, this primary work remained isolated and confined to recreational mathematics. In 1878, this formalism found an application in chemistry when the English mathematician J.J. Sylvester introduced the *chemicograph*. But it is especially during the twentieth century that network found an application in sociology, with the introduction of *sociogram*. In the second part of the twentieth century several models aimed to explain the recurrent structure found in networks. In particular, Paul Erdős and Alfred Rényi introduced in 1958 a model bearing their name. This model will be detailed further as it is still commonly used in network theory. In 1998, Duncan Watts and Steven Strogatz proposed their small-world model explaining the result of the experiment proposed by the psychologist Stanley Milgram in 1960[4]. Finally, in 1999 Albert-Lazlo Barabási and Réka Albert introduced the scale-free model which explains the structure of a large variety of networks. This last model will be widely used throughout the thesis and extensively presented in Section 0.3. Today, network theory is ubiquitous in many different areas of research. We give hereafter some examples of works using this formalism:

Biology and medicine Biology is probably one of the fields which has benefited the most from the network theory. We can cite for example the protein/protein interaction networks which help to empower our knowledge about the biochemical events between proteins within a cell; the metabolic networks describes more generally all the physical interactions in a biological system (from a cell to an organism); in a macroscopic point of view food webs are also represented as networks and help to better understand ecological interconnections; finally networks are widely used to model the propagation of a disease within a population. Throughout this thesis, we will precisely present applications in this last field.

Linguistic and art Semantic networks represent the semantic relations between different words in a language. They help to better understand the structure of a language, to implement high-level interfaces on computers or to design efficient translation engines. Narrative networks link together different concepts of a story which can be fictions, historic events, or even personal stories. In the same idea, there are networks linking the characters of fictional universes such as the Marvel Cinematic Universe[5], Game of Thrones[6] or the Shakespeare's tragedies[7].

Computer and information science Obviously networks are ubiquitous in computer sci-

[3]We will say a word further on the difference between graphs and networks

[4]In this experiment Stanley Milgram aimed to show that any individual is connected to any other through few intermediates. This groundbreaking work emphasized that the human society is a *small-world* and is commonly associated with the idea of the *six-degrees of separation*.

[5]See `https://studentwork.prattsi.org/infovis/projects/marvel-universe-visualization/`

[6]See `https://networkofthrones.wordpress.com/`

[7]See `http://www.martingrandjean.ch/network-visualization-shakespeare/`

ence at different levels. From the physical layer with the design of computer networks to the application layer with the structure of the internet in which websites are connected via hyperlinks. From the algorithmic point of view, the functioning of a web engine is also based on network theory. For example, Google uses, among other tools, the PageRank metric to find the most relevant website within the network. From a more theoretical point of view, bayesian networks are used in information theory to understand the relations between different pieces of information.

Sociology As said, sociology is the application which partly encouraged the development of network theory. Today, the research in this area studies social networks online (Facebook, Twitter,...) and offline (work, school, friendship), to understand, for example, how an information spreads in a population or how communities emerge. In criminalitics (also named forensic science), network theory is also used to gather information and detect fraud.

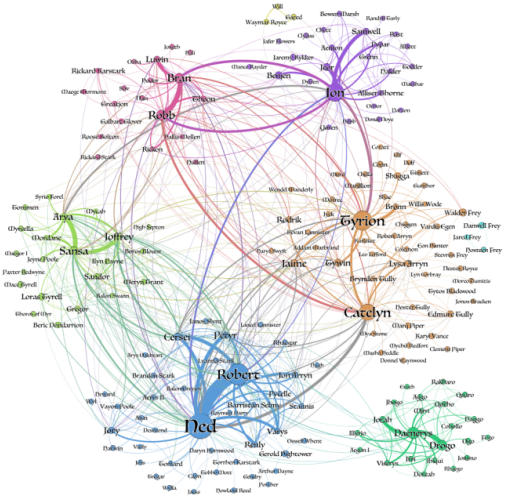
This list is far from exhaustive and aims only to give an overview on the wide range of applications of network theory. One could have also evoked power distribution networks, neural networks, applications in climatology, in risk-assessment, in economics and in traffic modelisation. This last application will be considered and discussed in the thesis. [29] draws up a vast review of the applications and research areas concerned by network theory. Figure 2 illustrates this variety of utilization.

In particular, a network may be used to represent the evolution of a dynamical system. In this case a state is associated to any element of the network and equation governs the evolution of the state of each element in function of the state of its neighborhood. Along this thesis, we will consider such evolution occurring on networks. We will present further the details of the dynamical equations in which we are interested.
While the study of networks is abundant, in some instances the size of the network makes it more challenging to analyze. From this issue flows the question of the reduction of networks which is the guideline of this work.

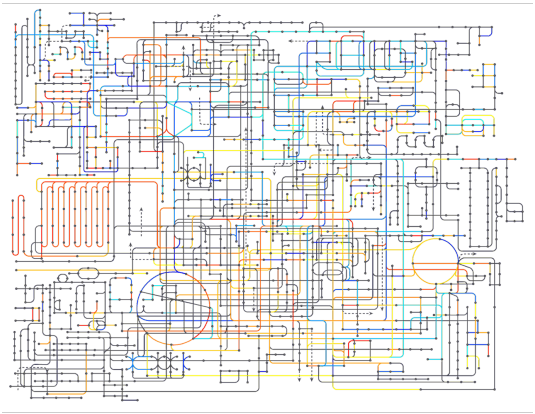
0.1.2 Reduction of complexity

Large networks (with thousands of nodes) are common in several fields like transportation, power grid or biology among others. The Stanford Large Network Dataset [71] lists such large systems and can give an order of magnitude: Internet peer-to-peer networks have around 10^4 nodes, traffic networks have around 10^6 nodes and social networks have between 10^4 and 10^7 nodes. At these scales, the analysis of these large networks becomes very costly or even impossible. This complexity motivates network reduction methods also known as coarse-graining or summarization methods. Works on the network reduction are profuse (see [77] for an extensive survey) and the ERC-granted project *Scale-Freeback*[8], in which this thesis takes place, aims precisely, among other things, to reduce the complexity of large-scale networks.

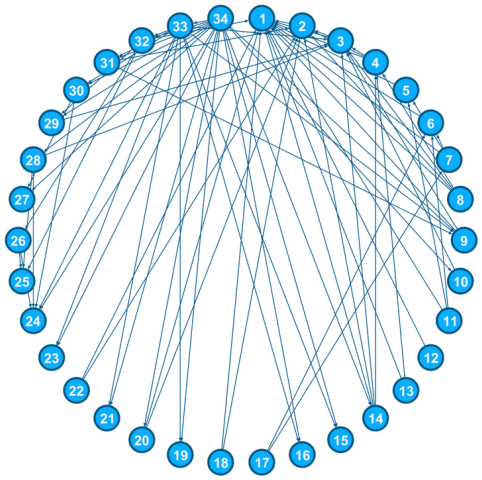
[8] See `http://scale-freeback.eu/`



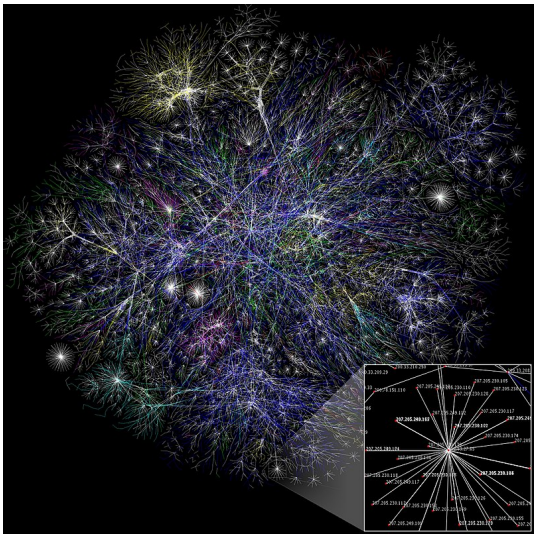
(a) Network of the connection between the characters of the Game Of Thrones universe (season 1)



(b) Simplified network of the human metabolism



(c) The Zachary's karate club network is an example of social network commonly used



(d) Visual representation of a portion of the Internet structure from the Opte Project

Figure 2: Use of networks in various research areas.

The objectives of the network reduction are different depending on the application. However, the different purposes of network reduction methods have the following form: cutting the complexity (e.g. volume of data, redundancy, visualization) of a network while preserving some properties (e.g. topological, dynamical, patterns). The techniques used differ but essentially there are four categories: partitioning (merging nodes in super-node and/or edges in super-edges), compression (exploiting redundancy in the patterns of the network) and simplification (removing unimportant nodes and/or edges). In this thesis, we consider only partitioning approaches as it is the most common and the most rich method. This terminology, as well as the approaches, may differ according to the fields of study. This thesis falls in the scope of a dynamical system vision which aims to reduce networks by preserving a consistency, in particular, in the dynamical behavior. Precisely, besides the preservation of dynamical characteristics, our work proposes to endow the reduced network with a particular shape known as *scale-free*. The two following sections aims to introduce a variety of network-related notions, among them the scale-free property, allowing to introduce more formally the objectives of the thesis.

0.2 Introduction to network theory

This section provides a partial introduction to the network theory and to the notions used along the thesis and necessary for the formulation of the objective of our work.

0.2.1 Generalities on networks and network partitioning

We introduce formally in this section some generalities and notations related to network and network partitioning. A network G is a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes[9] and $\mathcal{E}$ is a set of edges verifying $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The graph is said undirected if for all edges (v, w) in $\mathcal{E}$, the edge (w, v) is also in $\mathcal{E}$, which means that the edges has no *direction*. In the other case, the graph is said directed and the direction of the edges matters.
In this thesis, we consider directed[10] networks G, represented by the triple $(A, \mathcal{V}, \mathcal{E})$ where $A \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is the adjacency matrix, whose non-zeros values indicate the edges: $A_{i,j} \neq 0$ implies $(i, j) \in \mathcal{E}$. In particular if the non-zero values of A are different, the graph is said weighted. In this case, $A_{i,j} = w \neq 0$ indicated that a weight w is assigned to the edge (i, j). We may denote $G = (\,\cdot\,, \mathcal{V}, \mathcal{E})$ if only the structure of G (and not the weights) is relevant. A visual representation is associated to any network. In such representation, dots represent the nodes of the network and links between the dots represent the edges. In the case of a directed network, an arrow is added to the link to represent the direction of the edge and in the case of a weighted network the weight of each edges may be displayed.
We present now some definitions related to networks.

[9] While we use the terms of *network* and *node* throughout the thesis the notation G and $\mathcal{V}$ inherits from the terms *graph* and *vertex*. In the next section we will briefly discuss this terminology

[10] However, for readability concern we may illustrate some notions with undirected networks.

Definition 0.1 (Degree)
In general, the degree of a node is the number of connections the node has. When considering directed network we distinguish indegree and outdegree:

- *The indegree of a node v, denoted $\deg_{\text{in}}(v)$, is the number of nodes preceding v (these nodes are called the predecessors of v) which is:*

$$\deg_{\text{in}}(v) = |\{w \in \mathcal{V},\ (w,v) \in \mathcal{E}\}| \tag{1}$$

  where $|\cdot|$ gives the cardinality of a set. The set of predecessors is denoted $\mathcal{N}_{in}$:

$$\mathcal{N}_{in}(v) = \{w \in \mathcal{V},\ (w,v) \in \mathcal{E}\} \tag{2}$$

- *The outdegree of a node v, denoted $\deg_{\text{out}}(v)$, is the number of nodes following v (these nodes are called the successors of v) which is:*

$$\deg_{\text{in}}(v) = |\{w \in \mathcal{V},\ (v,w) \in \mathcal{E}\}| \tag{3}$$

  The set of successors is denoted $\mathcal{N}_{out}$:

$$\mathcal{N}_{out}(v) = \{w \in \mathcal{V},\ (v,w) \in \mathcal{E}\} \tag{4}$$

In the following, when there is no need to precise if we consider indegree or outdegree, we will use the general term *degree*.

Definition 0.2 (Degree distribution)
The degree distribution of a network G, denoted by Π_G, is a vector giving the distribution of the degree over the whole network which is:

$$\Pi_{G,k} = |\{v \in \mathcal{V}, \deg(v) = k\}|$$

Figure 3 gives an illustration of the degree distribution of a network.

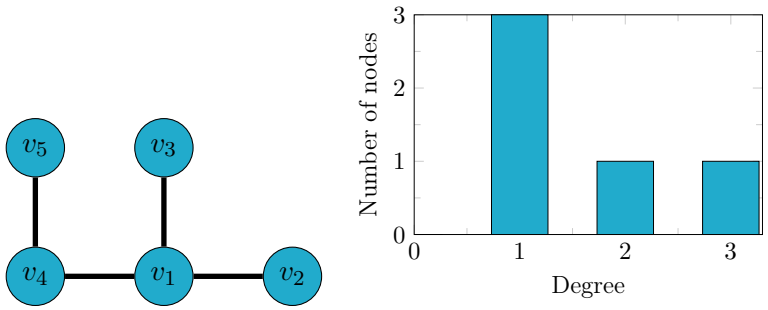
Definition 0.3 (Network partition)
A partition $\mathcal{S} = \{S_1, S_2, ..., S_n\}$ of a network is a partition of the set of vertices $\mathcal{V}$ which is:

$$\bigcup_i S_i = \mathcal{V} \tag{5}$$

$$\forall\ i,\ j\ \in [1,n]\ i \neq j \implies S_i \cap S_j = \emptyset \tag{6}$$

Each element S_i is named a *part*[11] of the partition $\mathcal{S}$. Let us remark that this definition does not imply that the nodes inside a part have to be *somehow* connected. From a partition of a network, we obtain a new network defined hereafter.

[11] In the literature, the term *cluster* is also used. We prefer here *part* for the consistency with *partition* and to avoid the polysemy of *cluster*.



(a) (b)

Figure 3: A network of 5 nodes (a) and the corresponding degree distribution (b): Three nodes (v_2, v_3, v_5 has degree 1, one node (v_4) has degree 2 and one node (v_1) has degree 3.

Definition 0.4 (Network coming out of a partition)
Let $G_0 = (\cdot, \mathcal{V}_0, \mathcal{E}_0)$ be a network, let $\mathcal{S}$ be a partition of this network. We denote $G_1 = (\cdot, \mathcal{V}_1, \mathcal{E}_1)$ the network coming out the partition $\mathcal{S}$ of G_0. Which is:

$$\begin{aligned} &\mathcal{V}_1 = \{1, \ldots, |\mathcal{S}|\} \\ &(i,j) \in \mathcal{E}_1 \Leftrightarrow (S_i \times S_j) \bigcap \mathcal{E}_0 \neq \emptyset \end{aligned} \tag{7}$$

If G_1 is a network obtained from a partition of G_0 we denote $G_0 \succ G_1$ or $G_0 \overset{\mathcal{S}}{\succ} G_1$ to emphasize the partition. Let us remark that since this relation only determines the structure of the reduced network and not its weights, there is an infinite number of weighted networks coming out of the partition $\mathcal{S}$ of G_0.
Finally, we also define a particular type of partition that we will need throughout our development: the merging[12].

Definition 0.5 (Merging)
A merging is a partition in which only two nodes are merged. Let $\{1, ..., n\}$ be the set of vertices, the merging of the vertices v and w is denoted by $S_{v,w}$ and:

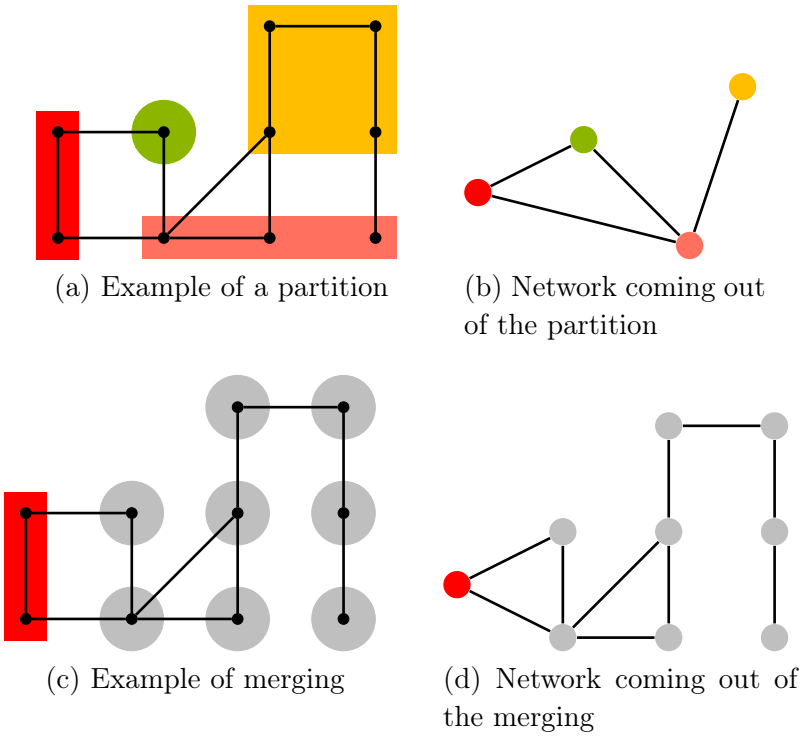
$$\begin{aligned} S_{v,w} = &\{\{1\}, \{2\}, \ldots, \{v-1\}, \{v+1\}, \ldots \\ &\ldots, \{w-1\}, \{w+1\}, \ldots \{n\}, \{v,w\}\} \end{aligned}$$

Figure 4 illustrates the notions of *partition*, *merging* and *network coming out of a partition*.

0.2.2 Network properties

This section aims to define a list of network properties that will be used throughout the thesis. These properties are roughly divided into three parts to help readability. This section is not

[12] We coined this term and it is not used in the literature



(a) Example of a partition

(b) Network coming out of the partition

(c) Example of merging

(d) Network coming out of the merging

Figure 4: Illustration of partition and merging

expected to be thoroughly examined at the first read but to be used as a glossary to which the reader can refer all along the thesis. A particularly interesting property for us is the scale-freeness and Section 0.3 is fully dedicated to present it.

Remark 0.1 (Terminology). The terms used in this thesis are not always uniquely employed in the literature, and due to the wide range of areas concerned by the network theory it is common that several terms coexist for a same concept. We will try to give these different terms and indicate the one we will use.
First of all, we say a word on the difference between *graph* and *network*: basically, these terms correspond to the same object but with a different connotation following the authors. Intuitively, *graph* refers to the mathematical object while *network* refers to the physical object. Said otherwise, a network is a graph endowed with a physical model. It is in this sense that the graph theory and network theory are often divided. For the sake of simplicity, in this thesis we do not make a real difference between these two terms, and we use mainly the term *network* to remain consistent. However in some cases, the term *graph* may be chosen if it appeared more natural and it should not preoccupy the reader. In the same way, the couples of terms *node*/*vertex* and *edge*/*arc* coexist in the literature and we will mainly use here *node* and *edge*.

Structural properties

In this part, we define notion regarding only the structure and the weights of the network without considering any dynamics. We first define connectedness, also known as connectivity. To introduce properly the different notions of connectedness we first introduce the notions of walk and path.

Definition 0.6 (Walk and path)
A walk is a sequence of edges of the form (v_{i_0}, v_{i_1}), (v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), $\dots (v_{i_{m-1}}, v_{i_m})$.
A path is a walk in which all nodes are distinct.
An undirected walk is a sequence of couples of the form (v_{i_0}, v_{i_1}), (v_{i_1}, v_{i_2}), $\dots (v_{i_{m-1}}, v_{i_m})$ *verifying* $(v_{i_k}, v_{i_{k+1}}) \in \mathcal{E}$ *or* $(v_{i_{k+1}}, v_{i_k}) \in \mathcal{E}$.
An undirected path is an undirected walk in which all nodes are distinct.

Definition 0.7 (Strongly connected network)
A network is said to be strongly connected if it exists a path from any node to any other node.

Definition 0.8 (Weakly connected network)
A network is said to be weakly connected if it exists a an undirected path from any node to any other node.

Roughly, a directed network is strongly connected if any node can be connected to any other node following the edges and it is weakly connected if the network forms an unique *island*. If the network is not weakly connected then it is disconnected. Figure 5 illustrates the three different cases: strongly connected, weakly connected and disconnected.

We now define the *characteristics lengths* of a network. In general two metrics are referred as the characteristics lengths: the diameter and the radius. Both aims to measure how much the nodes are *close* together. To define these two concepts, we first define eccentricity.

Definition 0.9 (Eccentricity)
In a network, the eccentricity of a node v*, denoted* ϵ_v*, is the greatest distance between* v *and any other node:*

$$\epsilon_v = \max_{w \in \mathcal{V}} \delta(v, w) \tag{8}$$

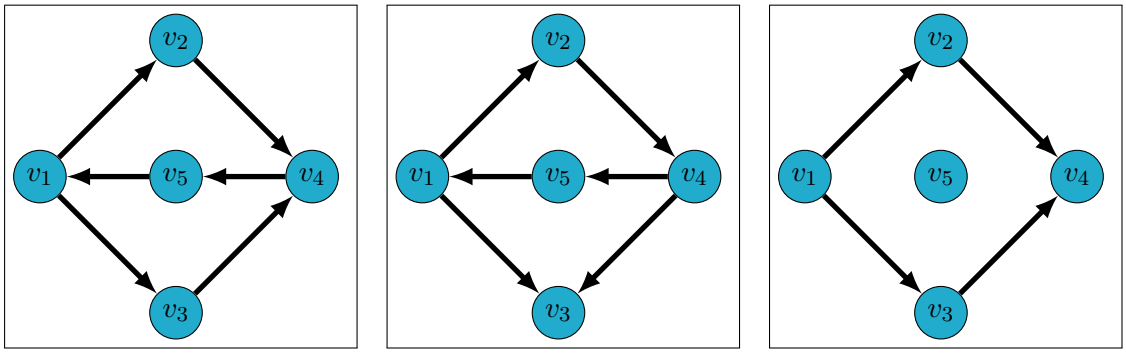
where $\delta(v, w)$ *is the length of the shortest path between* v *and* w.

Definition 0.10 (Radius)
The radius r *is the minimal eccentricity:*

$$r = \min_{v \in \mathcal{V}} \epsilon_v \tag{9}$$

Definition 0.11 (Diameter)
The diameter d *is the maximal eccentricity:*

$$d = \max_{v \in \mathcal{V}} \epsilon_v \tag{10}$$



(a) Strongly connected network: it is possible to go from any node to any other node following the edges

(b) Weakly connected network: it is not possible to go from node v_3 to any other node, the network is not strongly connected. However, the network forms an unique island.

(c) Disconnected network: node v_5 is not linked to the rest of the network.

Figure 5: Network connectedness

fig. 5	ϵ_1	ϵ_2	ϵ_3	ϵ_4	ϵ_5	r	d
(a)	3	4	4	3	3	3	4
(b)	3	4	$+\infty$	3	3	3	$+\infty$
(c)	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$

Table 1: Eccentricities, radius and diameters for networks of fig. 5

The diameter is then the longest distance between any two nodes of the graph. The radius is somehow the distance between the most central node and the node, which is the furthest from it. Note that if the graph is not strongly connected the diameter is infinite and if it is disconnected the radius and all the eccentricities are also infinite. Table 1 presents the eccentricities, radius and diameter for the three networks in fig. 5.

The clustering coefficient measures how much a network tends to form clusters. Different versions of the clustering coefficient exist. The one we use requires first the computation of the local clustering coefficient:

Definition 0.12 (Local clustering coefficient)
The local clustering coefficient of a node v, denoted c_v is the following quantity:

$$c_v = \frac{|\{i,\ j\ \in \mathcal{N}_{out}(v),\ (i,j)\ \in \mathcal{E}\}|}{n_{out}(n_{out}-1)} \tag{11}$$

where $n_{out} = |\mathcal{N}_{out}(v)|$

The local clustering coefficient measures the tendency of the successors of a node to be

connected together.

Definition 0.13 (Cluster coefficient)
The clustering coefficient of a network, denoted C is the average value of the local clustering coefficients:

$$C = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} c_v \tag{12}$$

By considering only few parts of the network we can extract a new network. This refers to the notion of *subgraph* defined hereafter.

Definition 0.14 (Subgraph)
Given $G_0 = (\cdot, \mathcal{V}_0, \mathcal{E}_0)$, a subgraph $G_1 = (\cdot, \mathcal{V}_1, \mathcal{E}_1)$ of G_0 is a graph verifying

$$\mathcal{V}_1 \subset \mathcal{V}_0 \tag{13}$$

$$\mathcal{E}_1 \subset \mathcal{E}_0 \tag{14}$$

Therefore, a subgraph G_1 of a given graph G_0 is obtained by considering only a part of the nodes and edges of G_0. An induced subgraph is a specific case of subgraph:

Definition 0.15 (Induced subgraph)
Given $G_0 = (\cdot, \mathcal{V}_0, \mathcal{E}_0)$, the subgraph induced by the subset $I \subset \mathcal{V}_0$, denoted G_I, is the graph consisting of the nodes in I and all the edges between nodes of I. Therefore, we have $G_I = (\cdot, I, \mathcal{E}_I)$ with:

$$\mathcal{E}_I = (I \times I) \cap \mathcal{E}_0 \tag{15}$$

Figure 6 illustrates the difference between the two notions.

Definition 0.16 (Regular network)
A network is said regular if all its nodes have the same degree:

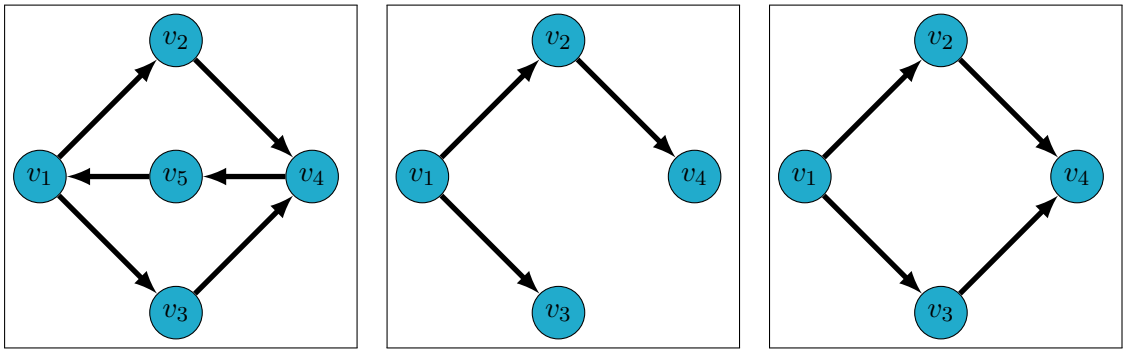
$$\exists\, d \in \mathbb{N},\ \forall\, v \in \mathcal{V},\ \deg(v) = d \tag{16}$$

d is called the degree of regularity.

When the degree of regularity is equal to 0 the network is said empty (or null) and when its equal to $n - 1$, where n is the number of nodes, the network is said complete. Figure 7 illustrates this.

Definition 0.17 (Mass conservation)
A weighted network is said to have the mass conservation property if, for each node, the sum of the weights coming in equals the sum of the weights coming out:

$$\forall\, i \in [1, ..., |\mathcal{V}|],\ \sum_{j=1}^{|\mathcal{V}|} A_{i,j} = \sum_{j=1}^{|\mathcal{V}|} A_{j,i} \tag{17}$$

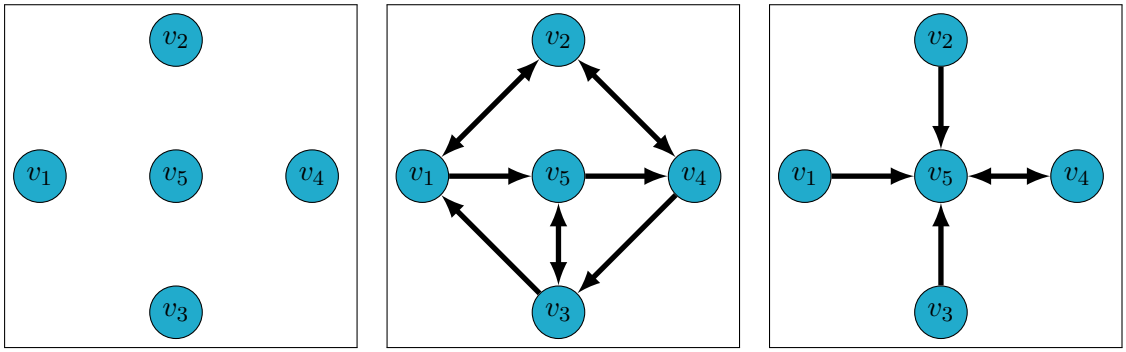


(a) Initial graph G_0

(b) Subgraph of G_0: the nodes $\{1, 2, 3, 4\}$ and *some* edges between them are considered

(c) The induced subgraph $G_{\{1,2,3,4\}}$: the nodes $\{1, 2, 3, 4\}$ and *all* edges between them are considered

Figure 6: Subgraph and induced subgraph



(a) An empty network is regular with a degree of regularity equal to 0.

(b) Regular network both for indegree and outdegree with a degree of regularity equal to 2

(c) Network regular for outdegree (degree of regularity equal to 1) and not regular for indegree

Figure 7: Network regularity

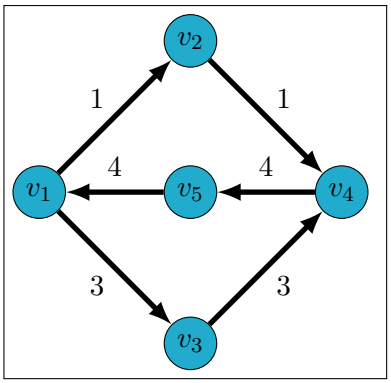


Figure 8: Network with the mass conservation property. For each node the sum of the weights coming in equal the sum of the weights coming out.

The term *flow network* is also used for networks having this property. In this thesis, for such a network we will say that it is mass conserving, it has the mass conservation property or it is a flow network. Figure 8 illustrates this property.

Definition 0.18 (Total mass)
The total mass of a weighted network, denoted M, is the sum of all the weights in the network:

$$M = \sum_{i=1}^{|\mathcal{V}|} \sum_{j=1}^{|\mathcal{V}|} A_{i,j} \tag{18}$$

In fig. 8 the total mass is 16.

Dynamical properties of networks : a control systems perspective

This second part presents more specifically notions related to the dynamical evolution of a network. Through the thesis, we will use linear equations governing the evolution of networks defined hereafter[13]. For that we associate to the network $G = (A, \mathcal{V}, \mathcal{E})$ a state vector $x \in \mathbb{R}^{|V|}$ assigning a value to each node of the network. The equation governing its evolution is:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ \dot{y}(t) = Cx(t) \end{cases} \tag{19}$$

where A is the weighted adjacency of the network G, $u(t) \in \mathbb{R}^p$ is a control (or input) vector, $B \in \mathbb{R}^{|\mathcal{V}| \times p}$ is the input matrix pointing the nodes to control, $y(t) \in \mathbb{R}^m$ is the output vector and $C \in \mathbb{R}^{m \times |\mathcal{V}|}$ is the output matrix. If, all along this thesis, we consider dynamical system evolving network, the notions defined below remain true for any system with or without an underlying network structure. Notions related to controllability and observability will be evoked throughout the thesis. Basically, they refer, respectively, to the possibility to steer the

[13]Note that in Chapter 2 we will consider a different type of evolution. We present here the most commonly used dynamical equations from which we can define notions of control theory.

state of the network towards a desired state and the possibility to determine the state of the network through only few measurements. We define properly these properties hereafter. The bulk of this section is inspired from the book *Linear Systems Theory* of João Hespanha [53] to which the reader is invited to refer for further information on the subject.

System (19) drives the state $x(0) = x_0$ at time t_0 to the state $x(T) = X_1$ at time T given by:

$$x_1 = e^{AT}x_0 + \int_0^T e^{A(T-\tau)}Bu(\tau)d\tau \tag{20}$$

The notion of controllability refers to the possibility to transfer the state between two states. The following definition allows to express the range of accessible state.

Definition 0.19 (Controllable subspace)
The controllable subspace $\mathcal{C}$ of the system (19) *consists of all states x_0 for which, for all $T > 0$, there exists an input $u : [0, T] \longrightarrow \mathbb{R}^p$ that drives the state from x_0 at time 0 to the origin at time T, which is:*

$$\mathcal{C} = \left\{ x_0 \in \mathbb{R}^{|\mathcal{V}|}, \forall T > 0, \exists u, \; e^{AT}x_0 + \int_0^T e^{A(T-\tau)}Bu(\tau)d\tau = 0 \right\} \tag{21}$$

The matrix C of (19) play no role in this definition. Therefore one talk about the controllable subspace of the pair (A, B) and thus, when considering the underlying network, one may talk about the controllable subspace associated to a weighted network and a set of controlled node.

Definition 0.20 (Controllable system)
The system (19) *is controllable if $\mathcal{C} = \mathbb{R}^{|\mathcal{V}|}$, which is if any state can be driven to the origin.*

Note that targeting the origin is not a special case: if we can go from any state in finite time to the origin, then we can go from that state to any other state in finite time as well.
To any control input $u : [0, T] \longrightarrow \mathbb{R}^p$ we associate the following measure of the energy:

$$\int_0^T \|u(\tau)\|_2^2 d\tau \tag{22}$$

The controllability Gramian is a matrix associated to control and used, among other things, to measure how much energy is needed to control a given system.

Definition 0.21 (Controllability Gramian)
The controllability Gramian[14] *of system* (19) *is defined by:*

$$\mathcal{W}_C = \int_0^{+\infty} e^{A\tau}BB^\top e^{A^\top(\tau)}d\tau \tag{23}$$

[14] This definition corresponds to the Gramian in infinite time. Another version in finite time exist also but will not be considered here.

Observability is a notion dual to controlability. It refers to the possibility to determine $x(0)$ from the future inputs and outputs $u(t)$ and $y(t)$ for $t \leq 0$. When considering system (19), the evolution of the output is described by:

$$y(t) = Ce^{At}x_0 + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau \quad \forall\, t \leq 0 \tag{24}$$

To study the system's observability, we need to determine under which condition we can solve

$$Ce^{At}x_0 = y(t) - \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau \quad \forall\, t \leq 0 \tag{25}$$

for the unknown $x_0 \in \mathbb{R}^{|\mathcal{V}|}$. This lead to the following definition:

Definition 0.22 (Unobservable subspace)
The unobservable subspace $\mathcal{UO}$ consists of all states $x_0 \in \mathbb{R}^{|\mathcal{V}|}$ for which

$$Ce^{At}x_0 = 0 \forall t > 0 \tag{26}$$

This leads to the the following definition

Definition 0.23 (Observable system)
The system (19) *is observable if the unobservable subspace is reduced to the zero vector which is $\mathcal{UO} = 0$. This implies that the initial state x_0 can be uniquely determined.*

The matrix B of (19) play no role in this definition. Therefore one talk about the observability of the pair (A, C).

A weaker notion than the observability is the detectability and will be considered in the thesis. Basically, a system is said detectable if the components of unobservable subspace are all asymptotically stable and therefore it is possible to determine asymptotically the state of the system. To define it properly we introduce first the following decomposition.

Theorem 0.1 (Observable decomposition - Theorem 16.2 in [53])
For every system (19)*, there is a similarity transformation that takes the system to the form*

$$\begin{cases} \begin{pmatrix} \dot{x}_o \\ \dot{x}_u \end{pmatrix} = \begin{bmatrix} A_o & 0 \\ A_{21} & A_u \end{bmatrix} \begin{pmatrix} x_o \\ x_u \end{pmatrix} + \begin{bmatrix} B_o \\ B_u \end{bmatrix} u \\ y = \begin{bmatrix} C_o & 0 \end{bmatrix} \begin{pmatrix} x_o \\ x_u \end{pmatrix} \end{cases} \tag{27}$$

for which:

- *The pair (A_o, C_o) is observable*
- *The unobservable subspace of* (0.24) *is given by*

$$\bar{\mathcal{UO}} = Im \begin{bmatrix} 0 \\ I_{\bar{n}\times\bar{n}} \end{bmatrix} \tag{28}$$

where Im is the image of a vector and $\bar{n}$ is the dimension of the unobservable subspace of the original system.

Based on this decomposition, the detectability is defined as follows:

Definition 0.24 (Detectable system)
The system (19) *is detectable if its decomposition verifies $n = \bar{n}$ (the system is observable) or if A_u is a stability matrix*

These notions of control theory will be used along the thesis both in the development of our work and in the motivation for our interest in scale-free networks presented in the following section.

0.3 Scale-free networks

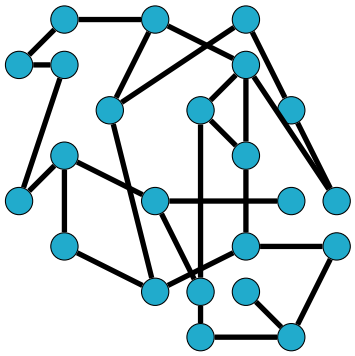
We pay particular attention to a special type of networks: scale-free networks. These mathematical objects have been extensively studied at the beginning of this century and have aroused a lot of interest. In one hand because it appeared that a lot of systems are well captured by the definition of scale-free networks, and in the other hand because these networks exhibit properties and behavior differing from other networks and potentially beneficial for some applications. In this section we propose a presentation of scale-free networks from different point of view: first we give a mathematical definition, then a historical approach and finally a non-exhaustive list of properties and applications.

0.3.1 Definition

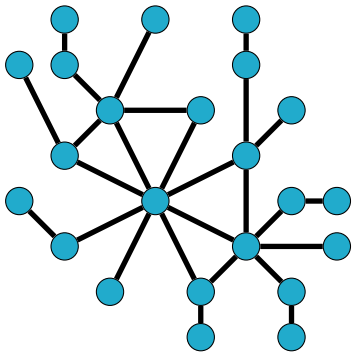
While it exists a wide range of networks, it is possibly to classify them according to different criteria. In particular, a property of interest is the degree distribution of the network. The degree distribution says a lot about the structure of the network and it is often used to classify networks. Based on this distribution, it is interesting to distinguish two different structures: the first one is when the degree distribution is homogeneous: all nodes have a degree close to the average degree. This type of degree distribution is obtained, for example, with random Erdős-Rényi networks [39]. The second noteworthy structure is the heavy-tailed distribution: in this case few nodes have a very large degree while most of the nodes have a small degree. In the case where the distribution is a power-law we refer to it as a *scale-free network*. Figure 9 gives a representation of this two types of network and their respective degree distributions.

Definition 0.25 (Scale-free network)
Let G be a network and Π_G its degree distribution. G is said to be scale-free if it exists $\alpha > 0$ such that:

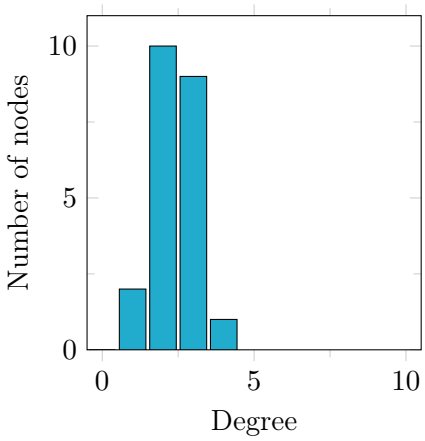
$$\forall\, k \in \mathbb{N}, \quad \Pi_G(k) \propto k^{-\alpha} \tag{29}$$



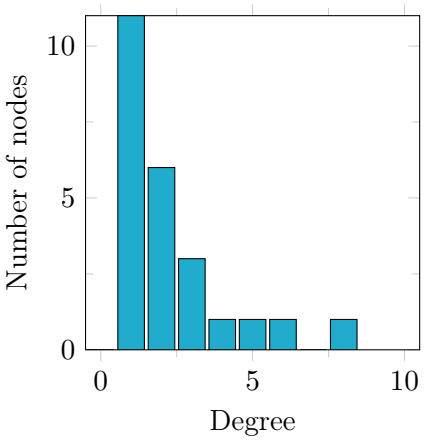
(a) Network with a homogeneous degree distribution: all nodes have more or less the same degree



(b) Scale-free network: some nodes are highly-connected and most are poorly-connected



(c) The degree distribution is bell-shaped



(d) The degree distribution of a scale-free network is heavy-tailed: the right side is extended further.

Figure 9: Comparison of two types of networks and their respective degree distribution

Figure 10: The international air transportation network has a scale-free structure. The small-world property (see Section 0.3.3) of the scale-free networks ensures that it is possible to travel the world relatively quickly. This image comes from [48].

α is called the scale-free coefficient of G.
In practice G is called scale-free if its degree distribution is relatively close to a power law.

For a directed network, the definition is the same but it is necessary to precise if we consider indegree, outdegree, or both. In the latter case, one may have two different scale-free coefficients, α_{in} and α_{out}.
Figure 10 shows a famous example of scale-free network: the international air transportation network. Indeed some airports are huge platforms and are highly connected while there is a majority of national or local airport with few connections.

0.3.2 Historic of scale-free networks

The studies of these networks started with Derek Price in 1965 [105] when he looked at the network of citation between scientific papers. Since the most a paper is cited the most he tends to be cited again, he discovered that there are few papers with a very large degree. In [105], Derek Price exhibited the heavy-tailed degree distribution of the citation networks and proposed a model capturing its properties. This was the first-time that a scale-free network is noticed, but the name was not coined yet.
Beyond this network, it is common, in a lot of different contexts, that the most connected item attracts more connections. This mechanism has been discovered in different fields and at different time, hence it is known by different names: preferential attachment in network theory, Yule effect in evolution, cumulative advantage or again Matthew effect in sociology. This is almost always this mechanism which leads to real-world scale-free networks. As an example, the World Wide Web network, constituted of the webpages connected via hyperlink, is subject

to preferential attachment. Indeed, when a new web-page is created, it will probably propose links towards some websites which are already highly-connected. In 1999, it is precisely by studying the topology of the World Wild Web that Albert-László Barabási and its collaborators rediscovered scale-free networks forgotten since Derek Price [9]. In their work, they proposed a model of scale-free network, the Barabási-Albert model, based on the preferential attachment mechanism. This model is explained hereafter. Since their rediscovery it has been shown that scale-freeness is ubiquitous in a wide range of fields such as biological networks [64], social networks [73] and internet network [125] among others. Above the highlighting of the scale-free nature of a variety networks, a lot of works has been done to study the properties of scale-free networks. We will detail further some of these properties.

The Barabási-Albert model To explain scale-free structure of the World Wild Web, Albert-László Barabási and Réka Albert proposed a model allowing to generate networks with a power-law degree distribution, which was not possible with the classical Erdős-Rényi model. The iterative algorithm they proposed relies on the preferential attachment mechanism. Hereafter is an explannation of the algorithm for the undirected case.

- At the beginning a small network, the seed, is considered:

$$G_0 = (\,\cdot\,, \mathcal{V}_0, \mathcal{E}_0) \tag{30}$$

- At each step k a new node is added to the network G_k:

$$\mathcal{V}_k = \mathcal{V}_{k-1} \cup \{v_k\} \tag{31}$$

- The new node is then connected to m nodes. These nodes are chosen via preferential attachment[a]: the most connected is a node, the most likely it is to be chosen:

$$E_k = E_{k-1} \cup \{(v_k, v_{i_1}), \dots (v_k, v_{i_m})\} \tag{32}$$

  where $i_1, \dots, i_m \leq k$ and the probability that nodes v_i is selected is proportional to $\deg_{G_{k-1}}(v_i)$.

No matter the value of m, if the number of iteration is large enough, the model leads to a network with a degree distribution $P(k) \propto k^{-3}$ which corresponds to a scale-free network. This model is the most commonly used to generate scale-free network. A version for directed network can be used. In this case, at each step links are created from m_{in} nodes to the new node, and from the new node to m_{out} nodes with a preferential attachment depending respectively on the indegree and outdegree of the nodes.

[a] This preferential attachment has to be linear to obtain a power-law distribution [43]

After the early 2000s and the promising discovery of scale-free networks, several works claimed that scale-free networks are actually rare in the real-world [25, 66]. These claims took a new extent in 2019 with the publication of the article "Scale-Free are rare" [19] leading

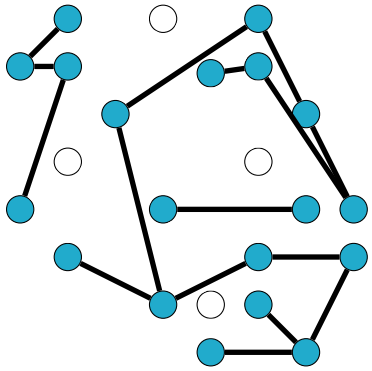
to several outcomes in scientific press. These works actually success to prove clearly something: there is almost no networks with a power-law degree distribution. This is true and noteworthy, the power-law is an idealized case for infinite-size network. As said before, two phenomena are used to explain the heavy-tailed degree distribution of some networks: preferential attachment and growing. These phenomena are used in the Barabási-Albert leading to a power law degree distribution: $P(k) \propto k^{-3}$. In real-world a lot more different processes may occur such as disappearance of nodes, disappearance of edges, attraction due to other factors, noise... Since then, these other mechanism has been incorporate in another scale-free generating model [37]. Therefore, results based on the power-law degree distribution and the Barabási-Albert model are not straightforwardly applicable to real-world network. However, they often furnish a very good approximation. This is a textbook case of the famous aphorism of Georges Box: "*All models are wrong, but some are useful*". The two articles [8, 54] comes back on this controversy and explains that indeed power-law is never reached but this does not disqualify scale-free as a very powerful model.
That being said, the omnipresence of scale-free networks in natural and technological systems encourages a better understanding of their properties and the potential advantages, inconvenient and opportunities that they present. We propose in the next section to present some properties flowing naturally from Definition 0.25. Some applications taking advantage of these properties are presented in Appendix 0.3.4. In order to structure the discussion we decided to distinguish roughly properties and applications in the following but sometimes these notions are closed and the distinction can be fuzzy.

0.3.3 Properties of scale-free network

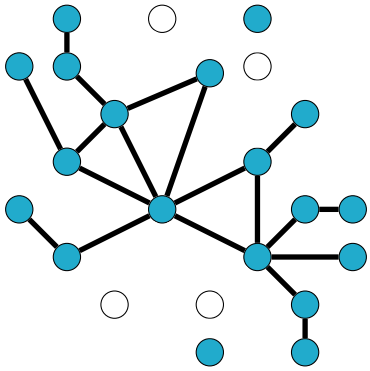
While some properties of scale-free networks depends on the model used to generate them, like the Barabási-Albert model or other models [31, 52, 70], some properties flows naturally from the power-law degree distribution. Here we present some of the most common properties of scale-free network.

Presence of hubs The main feature of the scale-free network is the presence of highly-connected nodes the so-called *hubs*. The other properties flow more or less directly from this fundamental property. On the contrary, in a random network, the presence of nodes with a large number of connection is very unlikely. Thus, before the emergence of the scale-free models, the models based on random networks struggled to capture the effect of the presence of hubs. Taking the World Wide Web as an example, it appears that more than 80 percent of the webpages are referenced by 4 links or less, these are the poorly-connected nodes, and 0.01 percent are referenced by 1000 links or more, these are the hubs [10].

Robustness to random failure A common concern when studying network is the robustness, or resilience, of the system. Does a computer network still works if a part of the equipment crashed ? How longer would be the route from place A to place B if some roads are closed ? How the brain would react if some neurons are inhibited ? Of course each of



(a) Random network with 4 random failures: the resulting network is fragmented in three components



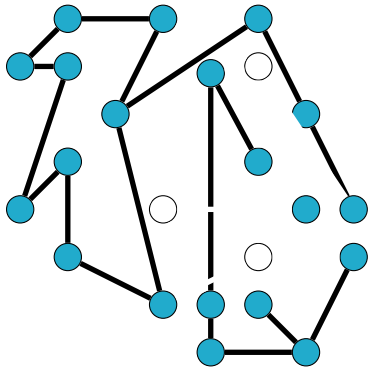
(b) Scale-free network with 4 random failures: a large component remains well-connected along with two isolated nodes

Figure 11: We compare here the robustness to random failure between random and scale-free networks. As most of their nodes are unimportant, the connectedness resists better to failure in scale-free networks.

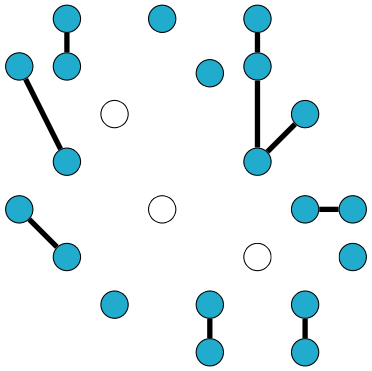
this question depends on the interaction between the nodes and are specific to the different problems. However, knowing if the network remains connected or if it is fragmented depends only on the structure of the network. It appears that scale-free networks are very resilient to accidental failures. An accidental failure, by opposition with targeted attacks, developed further, concerns the case where a node is *randomly* shut down for any reason. Noticing that a large majority of nodes in a scale-free networks are poorly-connected nodes, it appears that if a random fraction of the system fails, it will touch very likely these poorly-connected nodes, and so it will not impact so much the structure of the network. See fig. 11 for a comparison of the robustness of a random and a scale-free network.

Vulnerability to targeted attack The counterpart of the previous property is that scale-free networks can easily be broken apart if the nodes shutting down are aimed to the hub. As the hubs maintain the structure of the network, the removal of the hub would cause an important fragmentation of the network. Figure 12 illustrates this.
The consequences of this phenomenon are double: in some cases this represents a threat for the sustainability of the system, in other cases it is a chance to prevent a danger. Indeed, if a network describes a flow of interest, such as information on internet, persons on a traffic network or goods on a delivery network, the presence of such weak spot is a major risk for the durability of the system. On the contrary, if what flows over the network is harmful, such as a disease through a social network or a virus through a computer network, the hubs are strategic points to remove to prevent the spreading.

Small-world Intuitively, the small-world property says that you can go from any node to any other node in few steps in a scale-free network. In a scale-free network, by passing through



(a) Random network after removing the three most connected nodes. The resulting network is composed of two big components and an isolated nodes.



(b) Scale-free networks after removing the three most connected nodes. The network is fragmented in 11 parts.

Figure 12: Scale-free networks are sensitive to targeted attack: the hubs are the Achille's heel of this network. Random networks do not have such weakness and the vulnerability is distributed through the whole network.

the hubs you can connect quickly any part of the network. The Barabási-Albert model (and other models leading to scale-free network) is said to be small-world because the characteristic lengths (radius and diameter) of the generated network grow no faster than the logarithm of the number of nodes[15]. This explains, for example, that the air transportation network, exhibiting a scale-free structure [50] (as shown in fig. 10), allows to travel anywhere with few connections.

These properties among others (hyperbolic embedding, clustering distribution) flows directly from the definition of scale-free network. In some applications, it is possible to take advantage of these properties and gain efficiency with respect to other networks. In the next section, we present three different applications of scale-fee networks.

0.3.4 Applications of scale-free network

In this section, we present three applications taking advantage of the scale-free properties presented in Section 0.3.3. Actually, these are not direct applications to real-world but computations or approaches benefiting from scale-freeness.

[15]The model is even ultra small-world which means that the characteristic lengths grow as the logarithm of the logarithm of the number of nodes [26]

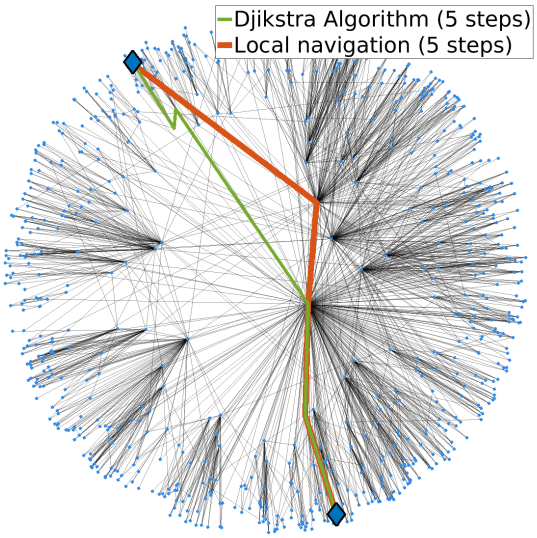
Network navigation

The problem to find the shortest path between two nodes in a network is solved, in particular, thanks tothe Dijkstra algorithm [35]. However another problem, known as network navigation, is to go from a node to another using only local information: the geographical position of the neighbors and the position of the destination [13]. The most obvious strategy is to go at each step as close as possible to the destination. In random network, the navigation performs badly since it is not possible to see distant shortcuts. By contrast, in scale-free networks when the nodes are arranged with the right method, the network navigation performs almost as well as the shortest path method [12]. This is due to the hyperbolic nature of scale-free networks and the best method to arrange nodes follows a hyperbolic metric[16]. Figure 13 illustrates this propensity of scale-free network for navigation. In this example, the local navigation gives a path as short as the optimal path (computed with the Djikstra algorithm) in a majority of cases: 88% of success. Moreover, in average the path found is 5% longer than the shortest path. By contrast, in a random network the results are way lower: 44.5% of success and 24% longer in average. In spite of the non-optimality, the local navigation presents two advantages in comparison with the Djikstra algorithm which ensures optimality. First, in some cases, the knowledge of the whole network is either impossible or too costly to obtain. Secondly and most important, the time complexity in the case of the Djikstra algorithm is $O(n\log(n))$ where n is the number of nodes in the network while for local navigation it is $O(log(log(n))$. As an example, for a network with a million nodes the local navigation is about five millions time faster. Thus, when designing the architecture of a network, this feature is a good argument to design with a scale-free structure if a lot of transmission has to been done between nodes (as communications networks).

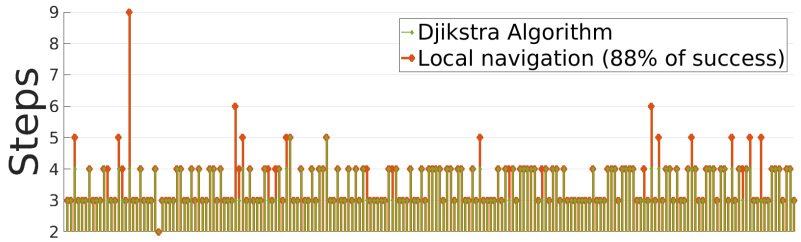
Strategy of vaccination

To study the evolution of a disease in a population a wide range of model uses network to model the connection among the population. These models form a field of research called *network epidemiology* presented in Chapter 5. These models can also be used to model the spreading of a computer virus [104] or to model information diffusion in a population [49].
A question of interest is the influence of the structure of the social network on the propagation of the disease. In scale-free networks, the presence of hubs is detrimental as their high-connectivity make them very susceptible to infection, and once infected they can contaminate a lot of individuals: hubs act as epidemic relay. For example, it is shown that, within a particular model (Susceptible-Infected-Susceptible), a disease may eventually disappear totally of a random network [6], while it always survives in a scale-free network [15] for any value of the model's parameters. This sounds like a bad news for scale-free networks: if a social network has a scale-free structure it will be difficult to eradicate a disease, and if a computer network is scale-free, as Internet is [28], a virus will easily survive. However, it is possible to take advantage of the sensitive position of the hubs by eliminating them. As seen in fig. 12

[16]Here is a video showing different manners to arrange the nodes of a network: `https://www.youtube.com/watch?v=sRujricoGQY`



(a) We compared shortest path and local navigation on a scale-free network where the nodes are arranged with a hyperbolic metric. The local navigation performs in this case as good as the shortest path computed with the Djikstra algorithm (although the two routes are different).



(b) Over 200 tests, the local navigation performs as the shortest path in 88% of cases and in average the path found is 5% longer.

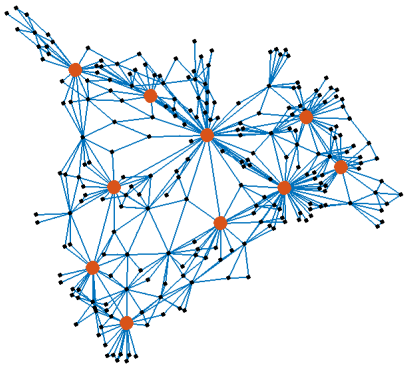
Figure 13: Due to its underlying hyperbolic nature, a scale-free networks is particularly adapted to network navigation. The computation time needed for it is several order of magnitude lower than the classical Djikstra algorithm.

in Introduction when the hubs are removed, a scale-free network is quite fragmented and restrains the diffusion which is not the case in a random network. By chance, with vaccine (or an antivirus in the computer case), it is actually possible to remove a node from the epidemic network. As expected in scale-free network it is very efficient to vaccinate the hubs (which is the persons with the most acquaintances), in priority [102]. Of course, this question arises if the number of vaccine is limited or in the case of voluntary vaccination [126], and raises ethical dimension making it essentially academic. Figure 14 proposes a simulation illustrating this result. This phenomenon may be an incentive when the structure of a network has to be chosen (for a computer network for example). If it is possible to protect few nodes, then a scale-free structure is preferable. Whereas if there is no protection, a random network is a better choice to restrain the propagation. When the structure of the network is already fixed (as social networks), an abstracting network with a scale-free structure can still be used to determine a good vaccination strategy. This point is developed in Section 5.2

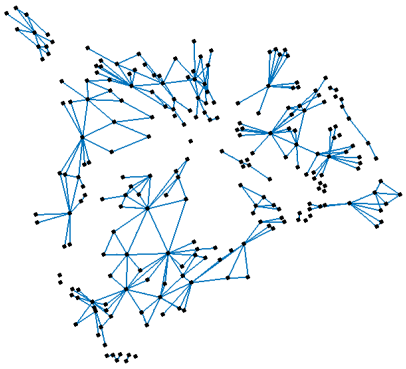
Network controllability

The notion of network controllability refers to the ability to steer the state of the network endowed with a dynamical equation as 19 towards a desired state using some inputs. Several notions of controllability can be considered according to the problem. The question of the controllability of scale-free network has divided the opinion of researchers in particular because of these different notions of controllability. We propose here a brief review of these discussions. Network controllability has first been introduced by Lin in 1974[74] where he defined structural controllability. A system is structurally controllable if the placement of the inputs (which is matrix B in (19)) and the configuration of the network (which is the placement of the non-zeros value of A in (19)) allow the inputs to reach any node of the system. As an example we consider a very basic model of information transmission. In this model, nodes represent individuals, edges are communication channels between individuals, and the inputs are information media (a newspaper for example). In this case, the system is said structurally controllable if the information coming from the inputs (newspaper) can reach any node (individuals). Figure 15 illustrates this example.

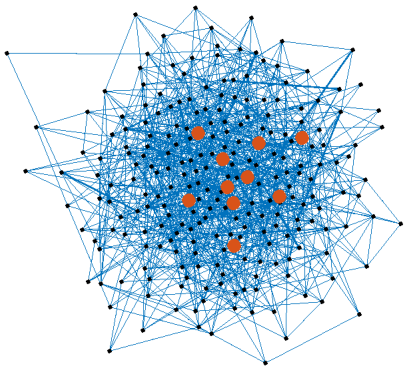
Some results [45, 111] have been obtained about structural controllability especially on how to determine if a network is structurally controllable and where to place the inputs to make a given network structurally controllable. However this notion of controllability does not take into account the intensity of the connection between the nodes, which is the weights in the matrix A in (19). The structural controllability only refers to the structure and not to the weighted network. In practice most networks have weighted edges which can represent a strength, a capacity or an intensity for example. To emphasize why this is essential, let us reconsider the network model introduced previously and add a weight on the edges. We consider that these weights correspond to the probability of transmition of the piece of information. Figure 16 presents two scenarios for this model. The two networks are structurally controllable as the information can reach everyone. However, one can feel that the transmission of the signal is much more difficult in the first scenario. If Bill consults only once the information of the input and if the individuals talk only once, the probabilities that



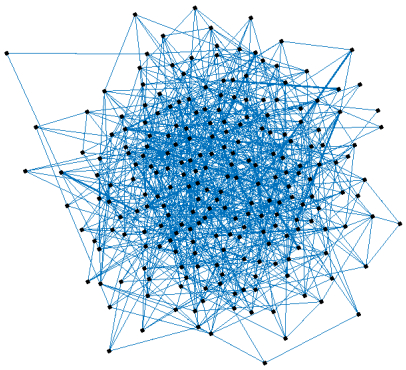
(a) Scale-free network. The ten most connected nodes are plotted in red.



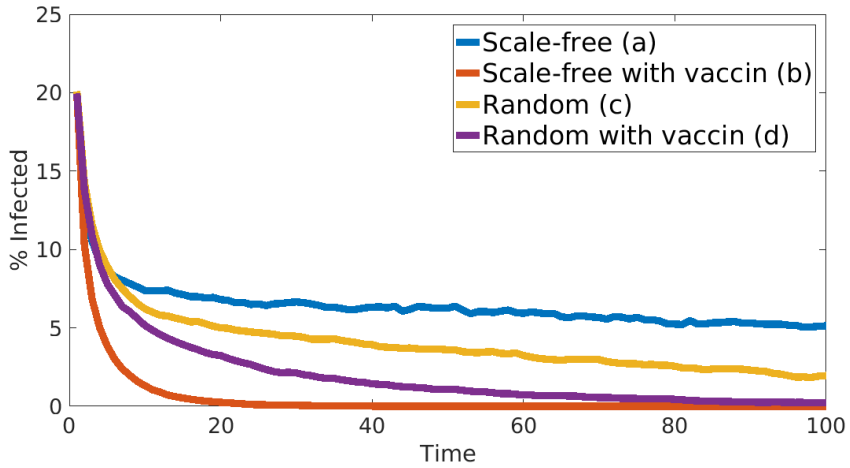
(b) Scale-free network after *vaccination*: the ten most connected nodes has been removed.



(c) Random network. The ten most connected nodes are plotted in red.

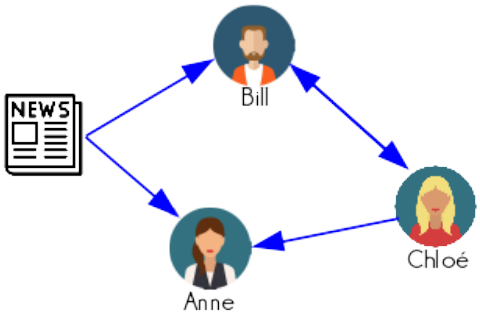


(d) Random network after *vaccination*: the ten most connected nodes has been removed.

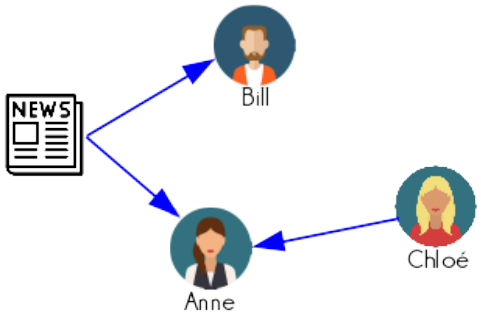


(e) Evolution of the percentage of infected people in the four networks. Initially 20% of people are infected and the evolution is driven by the SIS model. The scale-free network *performs* worse than the random networks. But after the *vaccination* of the hubs, the epidemic disappears totally in the scale-free network faster than in the random network.

Figure 14: Strategy of vaccination for scale-free networks



(a) Bill and Anne read the paper. Chloé and Bill can speak together while Chloé speak with Anne but not the inverse. In this case the information contained the input newspaper can reach everyone



(b) The situation is the same as the first configuration except that Chloé and Bill do not talk together. In this case, the information will never reach Chloé

Figure 15: We present two different configurations of a basic information propagation model. In (a) the system is structurally controllable: every node is reached by the input. In (b) the system is not structurally controllable: a node can not be reach by the input.

(a) Bill read the paper with a probability of 0.1. Then the information is transmitted to Anne and to Chloé with probability 0.1.

(b) Bill read the paper with a probability of 0.9. Then the information is transmitted to Anne and to Chloé with probability 0.9.

Figure 16: Even if the two networks are structurally controllable one can feel that the information is much more easily transmitted in the second case.

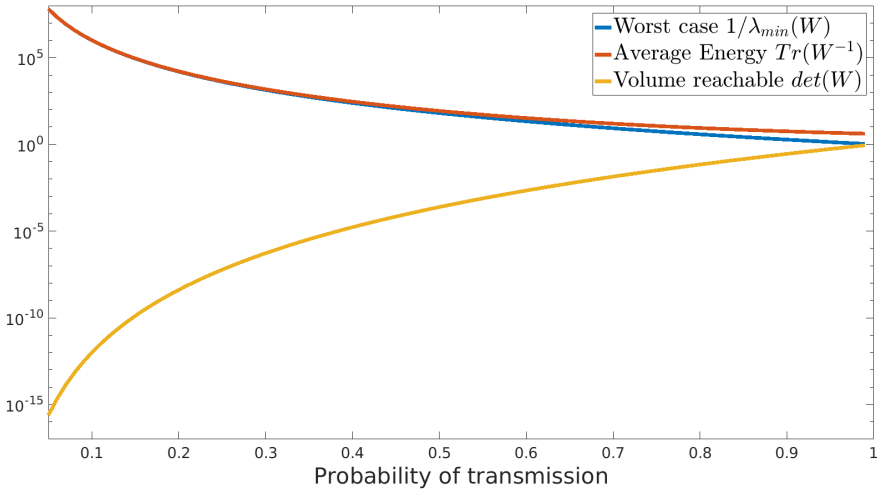


Figure 17: Energy needed to transmit the information up to Chloé as a function of the probability of transmission. Here we plot the three metrics measuring the control energy

Chloé accepts the information are respectively 0.1% and 72.9%. This example emphasizes a fundamental notion in network control which is not captured by the structural controllability: the energy required as defined in (22). Indeed, it may be theoretically possible to steer a system towards any desired state while the amount of energy needed make it impossible. It is particularly the case in large-scale network for which the structural controllability is not sufficient in practice. A lot of studies in this domain focus on the placement of the input in view to reduce the energy needed for control. As said before the controllability Gramian defined in (17) allow to estimate the energy needed to control a given system. There is, at least, three ways to quantify the energy needed to control a system [92] based on this matrix.

- The energy needed to steer the system from the origin along the most energy-consuming direction. This energy corresponds to the inverse of the smallest eigenvalue of the Gramian $1/\lambda_{min}(\mathcal{W})$
- The average energy needed to steer the system from the origin to any direction. It is proportional to the trace of the inverse of the Gramian $Tr(\mathcal{W}^{-1})$. If the system is not controllable, the Gramian is singular and $\mathcal{W}^{-1}$ is not computable.
- The volume of the space-state reachable from the origin with a fixed amount of energy. This volume is proportional to the determinant of the Gramian $\det(\mathcal{W})$

Figure 17 shows the value of these energy measure for the model presented in fig. 16 with values of the probability of transmission varying from 0.05 to 1. The three metrics confirm that the highest are the weights in the network the less energy is needed to control the system.

We can now raise the question of the effect of scale-freeness in the controllability of network. Intuitively, it seems that by controlling the hubs which are highly-connected it is easy to steer the state of the whole node. If the information propagation model proposed before is applied

on a scale-free network, then it seems that if the individuals with a lot of relation are targeted by the input, then the information would easily spread in the population. However scale-free networks also imply that a large number of nodes are poorly-connected and so are difficult to reach.
Liu, Slotline and Barabási proposed in [76] a major and one of the first contribution on this question. They proposed an algorithm to find a minimal number of input to control to reach controllability. In their study, they consider the structural controllability and not the notion of control energy which has been developed later. They showed that in scale-free network the number of nodes to control is relatively high and their algorithm seems to avoid to place input on hubs. This result has been received with surprise by the community engendering, in particular, two answering papers: The first one [91] written by Müller and Schuppert, two biologists, argued that from their empirical result, it appears that scale-free biological networks can be reprogrammed with few inputs (0.02% of controlled nodes against 80% according to the algorithm of *Liu et al.*). The second article [30] explain why the result of [76] contradicts the intuition and the empirical results: in the model they proposed, nodes evolves as a function of their neighborhood and do not have a proper dynamic. In most practical cases, though, the evolution of a node is influenced by its current state.
It is clear that the link between controllability and scale-freeness is not clear. We propose hereafter several factors which may affect this relation:

- The measure of the controllability: As seen before some consider structural controllability which is a binary measure giving sometimes unrealistic result due to energy requirement. Other measures consider the energy required but several metrics co-exist. Other measures can also be considered: the minimum dominating set, for example, which also conceals the notion of energy. With this measure, scale-free network appears to be easily controllable as concluded in [94] and [89].

- The system considered: The system may have a linear or non-linear dynamics, and as said before the nodes may have, or not, a proper dynamics. Several generating models leading to scale-free networks can be used and leads to slightly different properties. [36] presents the difference of controllability between two models of scale-free networks.

- Type of control: Several question appears: where is the control applied ? It may be on the nodes, or on edges. How is it applied ? It may be open-loop or closed-loop controller. What is the objective of the control ? It may be the whole state of the system, a unique state (single output control) or an aggregated value of the states (output control). Is the control constrained or unconstrained ?

It appears therefore that the question of the controllability of scale-freeness is not unique and indissociable from the scenario, the network system, and the control strategy considered which explains the dissensus on this question.
All that being said, we present now briefly an approach developed in [21] which is particularly interesting in our case. It shows that scale-freeness is somehow a good feature for an abstracting network used to design a control input of a large-scale network. The objective of the article is to control an aggregation (e.g. the average) of the uncontrolled nodes. Based on

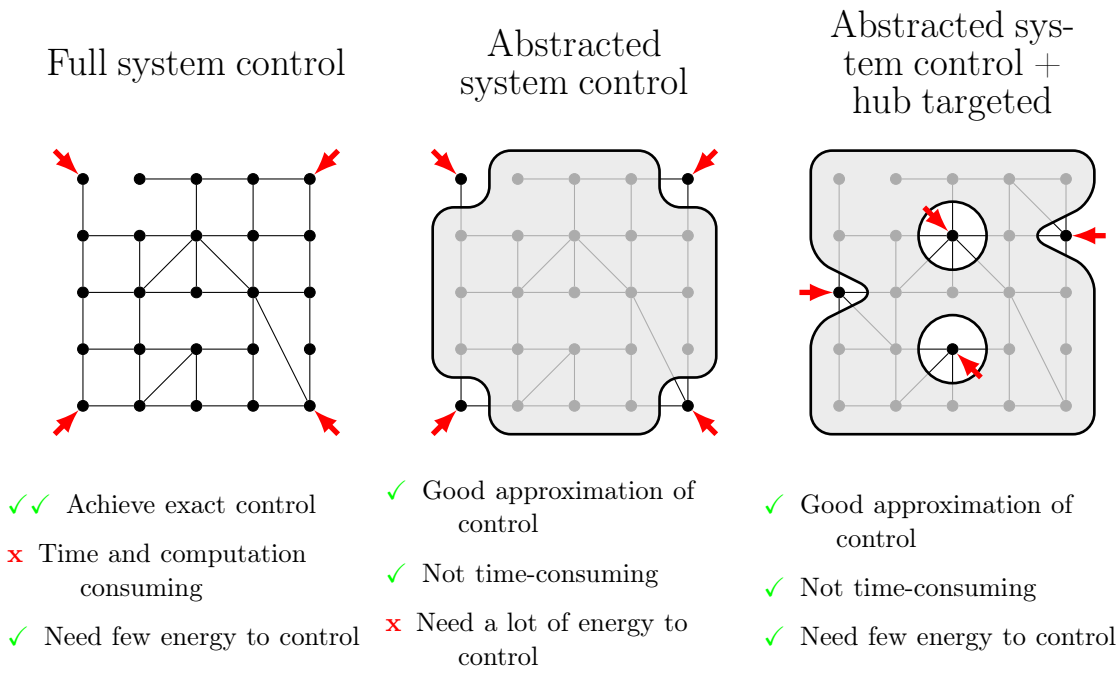


Figure 18: In [21] several control strategy for network are compared. In the first scenario, the whole system is considered. A control design can be found to reach a desired state with low energy. However, for large-scale network the computation needed to find this control may be tremendous. The second and third scenarios uses then an abstraction of the large-scale network to reduce the computation requirements and preserve a good approximation of the desired state. Moreover, in the third scenario it is emphasized that if the controlled nodes are the hubs of the network, the energy needed is much lower than for random controlled node.

the observation that for large-scale networks the computation of the controllability Gramian and the optimal control is very time-consuming, the authors propose to considers a smaller network abstracting the large one. They show that using this abstraction, the computation of the optimal control is much faster and preserves a good performance as the output is close to the desired state. Finally, and essentially for us, they show that the control energy needed decreases if the degree of the measured nodes increases. Figure 18 illustrates the main result of this article. Therefore, the abstracting network allows to design a low-computation control directed on few nodes and driving the average state of other nodes close to a desired state. By providing the abstracting network a scale-free structure, the control is moreover low energy-consuming.

0.3.5 Synthesis

Through this presentation of scale-free networks we have seen that it is useful to group together networks with a heavy-tail degree distribution. Indeed, a lot of real-world systems have this tendency and share common properties. We have highlighted some properties flowing from the definition, and some applications taking advantages of these properties. Therefore, when

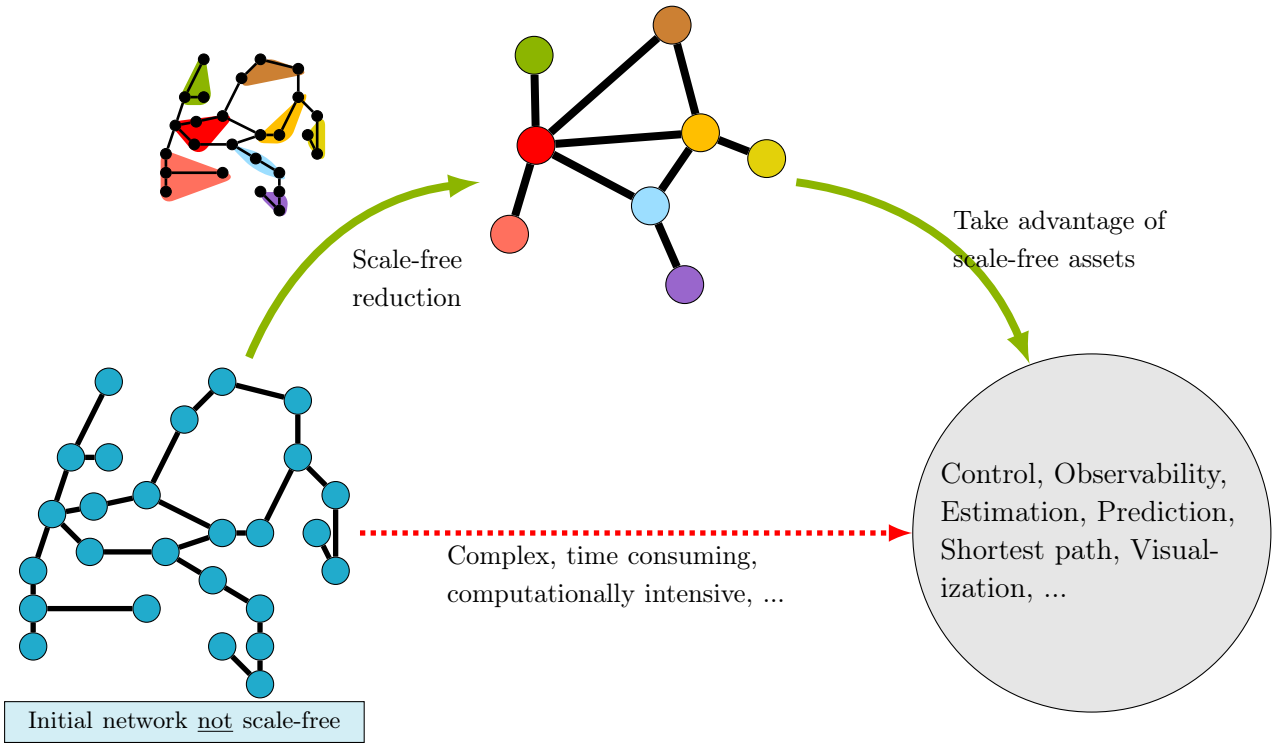


Figure 19: Principle of the Scale-Free reduction approach.

designing a new network it may be useful to endow it with a scale-free architecture if these properties may benefit to the applications. In contrast, when considering a pre-existing system, the properties of the network are already fixed and there is no benefit to consider the network as scale-free. However, when a large-scale network is reduced it is possible to impose a scale-free structure to the reduced network in order to benefit from the assets of the scale-freeness by using a reduced network with a scale-free structure. This idea is the very essence of our work and will be developed in particular in Chapter 2. Figure 19 illustrates this principle. Of course, in order to use the reduced system it is essential to preserve the behavior and properties of the initial system. Some properties which are particularly interesting to preserve through the reduction have been presented in Section 0.2.2. In the next section we discuss more precisely the problems we want to tackle and how they are arranged within the thesis.

0.4 Problematics and contributions

In order to reduce the complexity or large-scale network we aim through this thesis to develop tools and algorithms to partition a network with the following criteria: i) preserve some properties (dynamical behavior, structural characteristics, ...) of the original network ii) impose some properties (scale-freeness) to the reduced network.
The design of such network partitioning algorithms is the cornerstone of our work to which we

graft different related problematics. As evoked, the major one is the design of a partitioning algorithm inducing a scale-free network which will be treated in Chapter 2. Within this chapter we will also think about which properties of the initial network have to be preserved in order to preserve the characteristics of the initial network and how to do so. Before that, we raise a more theoretical question on the structure of the partition itself: as the parts may gather disconnected nodes, what are the drawbacks of imposing the nodes inside each part to be connected ? This question will be explored in Chapter 1. The third major work, treated in Chapter 3, concerns a reconstruction issue. We will investigate how to partition a network between measured and unmeasured nodes in order to ensure that the average of the unmeasured nodes can be efficiently reconstructed. Then, three applications are presented: in Chapter 4, we apply the partitioning algorithm towards scale-freeness to a large-scale urban traffic network. We show then that, thanks to the properties preserved through the partition, the reduced network can be used as an abstraction of the initial network. Finally, Chapter 5 will propose two independent applications in epidemiology. In the first one, we show that the scale-freeness of the abstracting network can be used to build a cure-allocation strategy. In the second application, we take advantage of the result on average reconstruction to estimate the evolution of a disease on a large-scale network. Figure 20 gives an overview of the structure of the thesis.

0.5 Publications

Here is a list of the works published or under review by the author.

Conferences

Nicolas Martin, Paolo Frasca, and Carlos Canudas-de Wit. "Network reduction towards a scale-free structure preserving physical properties". In: *Complex Networks 2017-The 6th International Conference on Complex Networks and Their Applications*. 2017, pp. 294–296

Nicolas Martin, Paolo Frasca, and Carlos Canudas-de Wit. "A network reduction method inducing scale-free degree distribution". In: *2018 European Control Conference (ECC)*. IEEE. 2018, pp. 2236–2241

Nicolas Martin, Paolo Frasca, and Carlos Canudas-de Wit. "MergeToCure: a New Strategy to Allocate Cure in an Epidemic over a Grid-like network Using a Scale-Free Abstraction". In: *IFAC-PapersOnLine* 51.23 (2018), pp. 34–39

Nicolas Martin et al. "The price of connectedness in graph partitioning problems". In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 2313–2318

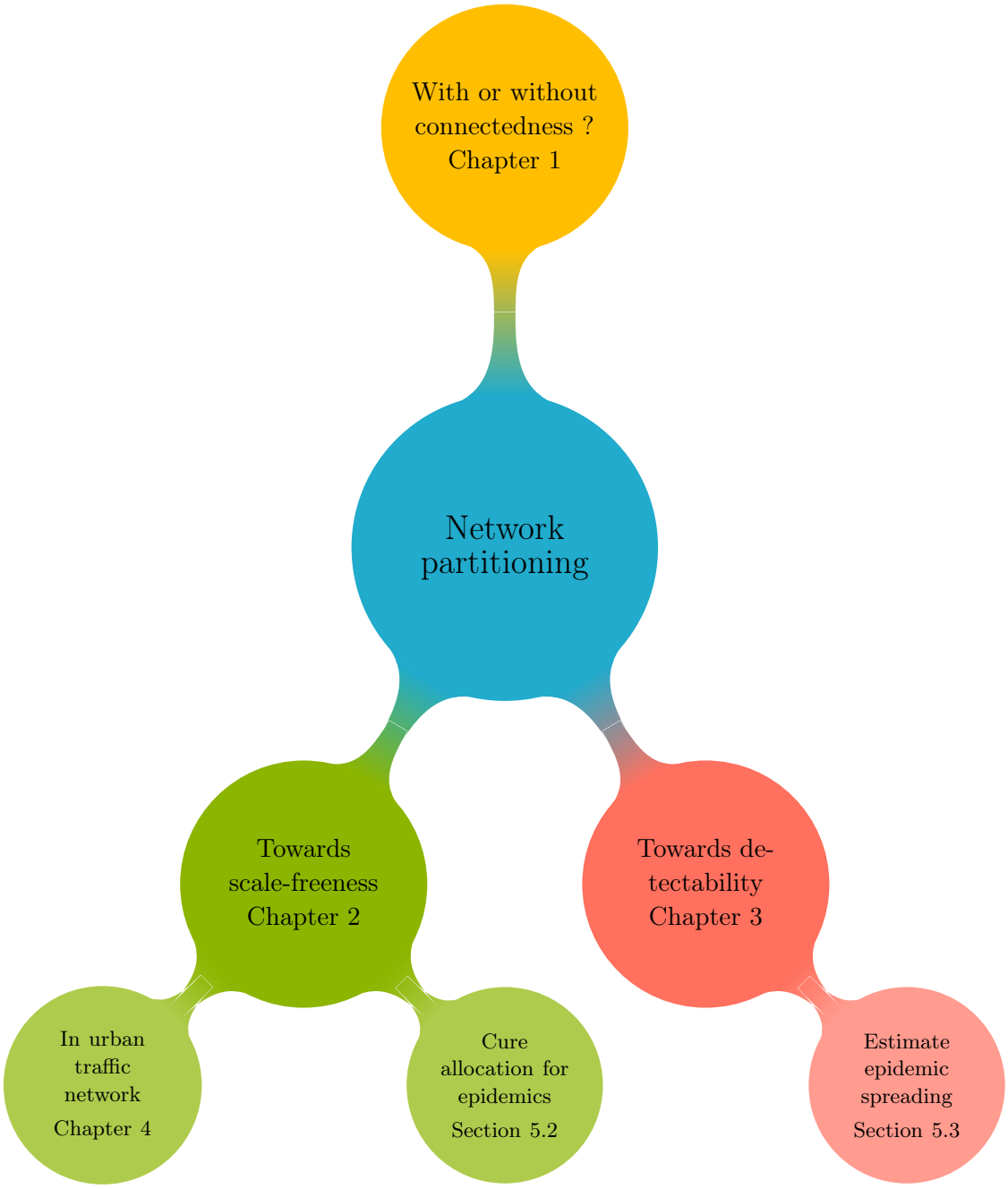


Figure 20: A visualization of the structure of the thesis

Journal

Nicolas Martin, Paolo Frasca, and Carlos Canudas-de Wit. "Large-scale network reduction towards scale-free structure". In: *IEEE Transactions on Network Science and Engineering* (2020)

Chapter 1

The price of connectedness

Tout se paye ici, demande à Pépinot.

Clément Mathieu, Les Choristes

Contents

If the main purpose of the thesis is to investigate partitioning problems, we study in this first chapter a preliminary question. This question concerns the structure of the partition itself and can be formulated as follows: how much adding a connectedness constraint in a partition problem degrades the solution ? Said otherwise, we will compare two classes of problems of optimal network partitioning: in the first class of problem one impose the nodes belonging to a same part of the partition to be connected. In the second class of problem, no constraint of connectedness is imposed. Obviously, this constraint worsens the result of the optimization problem. However, it may be interesting to add this constraint in some applications. Therefore, it is useful to estimate how much this connectedness constraint degrades the optimal solution of the partitioning problem. The estimation of this degradation is the objective of this chapter.
The work presented throughout this chapter has been published in [87] and presented at the European Control Conference 2019. This work has been carried out partially during my stay in Imura Laboratory in Tokyo Technology institute alongside with the team of Pr. Imura and within the JSPS summer program.

Figure 1.1: A class in which pupils have different affinity forms a network. Here the nodes are the pupils and the edges represent the friendship between them which can be one-sided (yes it may be sad, but it is for the sake of the example). To constitute working groups, should the teacher consider this friendship network ? This image is from "Les Choristes" of *Christophe Barratier*.

1.1 The price of connectedness and problem formulation

1.1.1 Introducing problem: a textbook case

We present first a short scenario in order to familiarize the reader with the problem before introducing a more formal framework. We consider a class in which some pupils are friends and some are not as shown in fig. 1.1. For an activity, the class has to be divided in some groups of different sizes: for example, 20 pupils have to be divided in 4 groups of sizes: 4, 7, 6 and 3. The teacher wants to optimize the groups regarding a precise metric: he wants to be the groups to be as homogeneous as possible. Therefore he wants to minimize the average standard deviation of level within the groups. Before addressing the grouping problem, the teacher wonders if he should add a constraint of friendship within the groups. By adding such a constraint, the best grouping he can find would be always worse or equal than in the unconstrained case. Indeed, the unconstrained case includes all the constrained grouping. However, the teacher wonders *how much* this constraint would damage the quality of the solution which is the homogeneity inside each group.
When looking at the network of friendship, adding this constraint means that each group have to form a *weakly connected* subgraph. If the teacher want to please the kids, he has to pay in counterpart a deterioration of the quality of the grouping. He has to pay the *price of connectedness*.

1.1.2 The price of connectedness

Generalizing the previous example, any partitioning problem can be solved with or without the connectedness constraint. Of course, adding a constraint to a problem can only reduce the quality of the optimal solution. But, as discussed later, in some circumstances this constraint is necessary. In this chapter, we consider network partitioning problems which can be seen as optimization problems. A large number of works have treated partitioning problems with different objective function and constraints. For example, [90] aims to preserve stability and synchronization of the system, [23] and [22] preserves the network structure, [97] looks for the best reduced system in order to estimate an aggregated state of the initial network and [60] provides a reduced system with a dynamical behavior close to the initial system while preserving several properties for control purpose. This last work will be presented in this chapter as a motivating example.

A partition respecting the connectedness constrained is named here *connected partition*. In some cases, it is relevant to prefer connected partitions: as in the introducing example, when consider social network one may want to find a partition ensuring that in each part the individuals are related. When studying epidemic spreading through a population, it is interesting to detect communities[1] in order to apply a control or to observe the evolution at the borders [85]. In another register, when considering networks with a geographical nature such as urban traffic networks, transportation networks or power grids one may want to preserve the geographical nature of the system and so imposing each part to be connected. In urban traffic network, if the reduced network is used for estimating the traffic state [78], we want that each part of the partition corresponds to a geographical area. Since we consider directed networks, connectedness may have different definitions as explained in the introduction. We will consider here weak connectedness as it is enough to ensures the preservation of the geographical nature of the network.

When the connectedness constraint is considered, it is clear that the optimal solution is always worse or equal to the optimal solution without the constraint. In this chapter, we investigate how much the connectedness constraint degrades the solution regarding to the structure of the network. The difference between the two optimal solutions is named *the price of connectedness*[2] and denoted by Δ. It is clear that the price of connectedness actually depends mainly on the optimization problem. For example, if the metric to minimize favors connected partitions, adding the constraint will not affect too much the result.

As an example, we formalize a similar problem as in the introducing case. We consider a network where a value is given to every node. We pose now the following problem: Find a partition of size n minimizing the average variance within the parts. If in the system of interest it turns out that adjacent nodes tend to have close values then the price of connectedness will be relatively small. Figure 1.2 illustrates such a case.

The approach proposed here aims to estimate the price of connectedness without solving the two optimization problems. Even more, we actually do not consider any particular partitioning problem and we want to estimate the price of connectedness in a network for any problem.

[1] in a population, communities can be viewed as a connected subgraph

[2] This term has been coined in reference to the concept of *price of anarchy* in game theory. This refers to the degradation due to the selfish behavior of the agent compared to the global optimum [69].

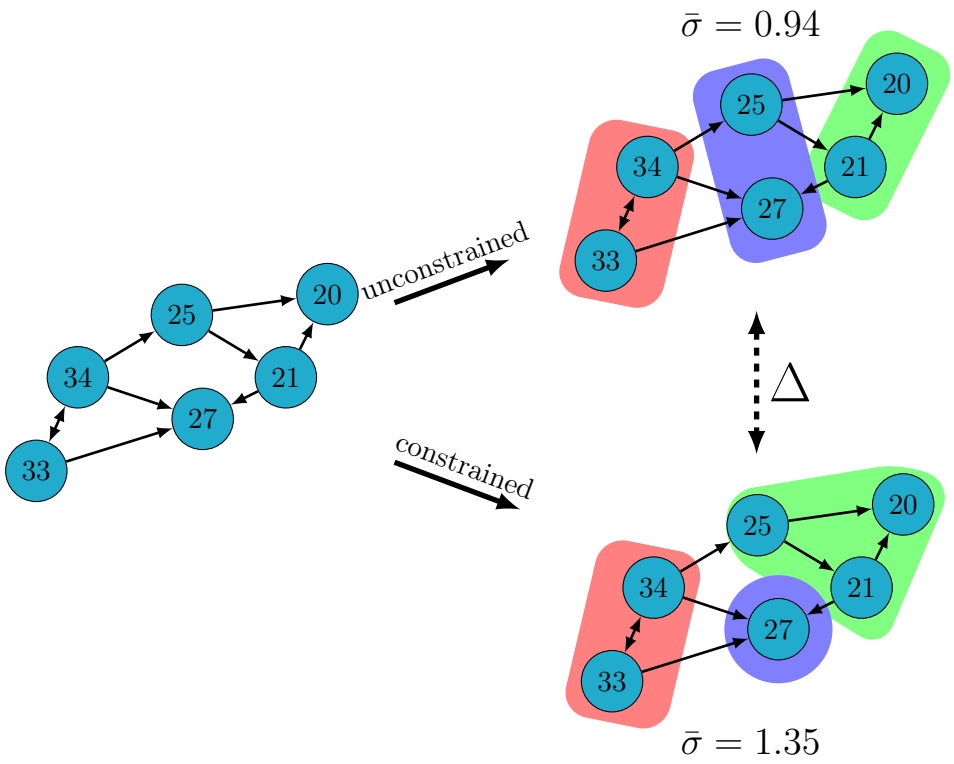


Figure 1.2: Partitioning the network in three parts in order to minimize the standard deviation within the parts. The price of connectedness is the difference between the optimal solutions in the unconstrained and constrained case, in this case $\Delta = 1.35 - 0.94 = 0.41$.

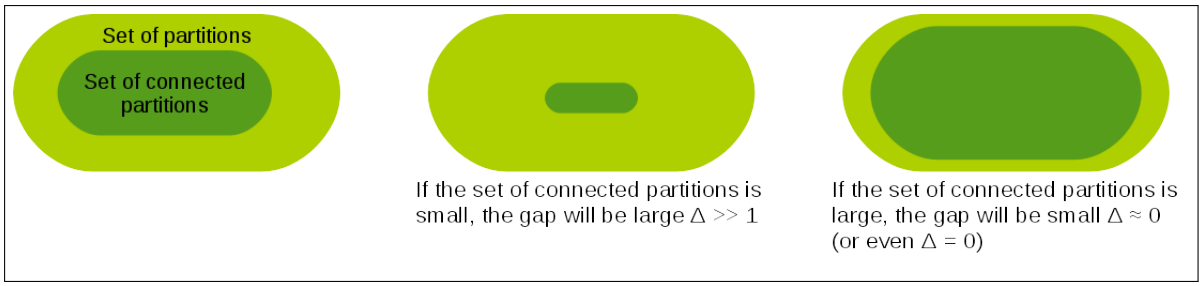


Figure 1.3: In order to estimate the price of connectedness, we compare for a given network the number of partitions and the number of connected partitions. The ratio of the two cardinalities is the *ratio of connectedness*.

As the partitioning problem is an optimization problem, adding the connectedness constraint amounts to reduce the feasible set of the problem. Therefore, we propose to estimate the price of connectedness with the ratio of the cardinalities of the two feasible sets: the set of partitions and the set of connected partitions. It is clear that this approximation is somehow coarse since we do not consider any particular problem. In the other hand, it allows to have an estimation of the price of connectedness of a given network for any partitionning problem. See fig. 1.3 for an illustration of this approach. This ratio is named hereafter *ratio of connectedness* and is denoted by ρ. To motivate this approach we will first present some simulations on a partitioning problem from the literature: the cluster model reduction [58]. We focus then on the estimation of the ratio of connectedness ρ. Note that this ratio can also be viewed as the probability that a given partition is a connected partition. We consider in this article random directed network obtained via the directed Erdős-Rényi model detailed later. We will show then that the value of ρ in an Erdős-Rényi graph is directly linked with $\bar{\rho}$ the probability that an Erdős-Rényi graph is connected. The value of this latter probability has been investigated by a broad literature providing asymptotic estimations. See [63] for a discussion on these results. These asymptotic estimations are valid for very large networks. We propose here an upper bound on $\bar{\rho}$ valid for networks of any size. From this result we derive an upper bound on the ratio of connectedness ρ. Let us reconsider the introducing problem to explain approach. For any problem of pupils partition for which the sizes of the part are known, we propose to estimate the deterioration of the optimal solution caused by the friendship constraint by comparing the number of ways to partition the pupils with and without the constraint. Intuitively, if almost all pupils are friends, almost every partition will be a *friendly partition* and so the price of connectedness is low. At the contrary, if the number of friendship is low, there are few friendly partitions compared to the number of partition, and so the price of connectedness is high.
The main contributions of this chapter are:

(i) Highlighting the relation between the price of connectedness Δ and the ratio of connectedness ρ via a numerical example

(ii) The link between ρ in an Erdős-Rényi graph and the probability $\bar{\rho}$ that an Erdős-Rényi graph is connected; and finally

(iii) the derivation of a tight upper bound on $\bar{\rho}$ leading to a tight upper bound on ρ, both for Erdős-Rényi graphs

The chapter is composed as follows: in Section 1.2 an example illustrates the price of connectedness in a concrete problem and the link between Δ and ρ, Section 1.3 contains the main results, namely the estimation of Δ. Section 1.4 presents some simulations showing the validity and the tightness of the estimation. A last section concludes the chapter.

1.2 Illustrative example: The clustered model reduction

In this section, we will present an example of partitioning problem from the literature. Within this example we will emphasize the price of connectedness which is the gap between the solutions of the constrained and unconstrained problems.

1.2.1 The clustered model reduction problem

Consider the following discrete-time linear system:

$$\Sigma : x(t+1) = Ax(t) + Bu(t) \tag{1.1}$$

where $x(t) \in \mathbb{R}^n$ is the state of the system. Given $\hat{n}$ the desired size of the reduced system. We define the reduced system as follows:

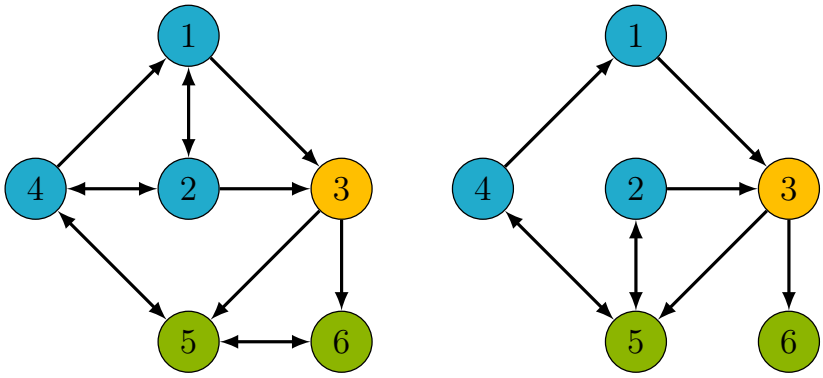
$$\hat{\Sigma}_P : \begin{cases} \xi(t+1) & = PAP^\top \xi(t) + PBu(t) \\ \hat{x}(t) & = P^\top \xi(t) \end{cases} \tag{1.2}$$

where $P \in \mathbb{R}^{\hat{n} \times n}$ is the reduction matrix. The problem is to find the matrix P making the dynamics of $\hat{\Sigma}_P$ the *closest* to the dynamics of Σ. More precisely, we want to minimize $\|g - \hat{g}_P\|_{\mathcal{H}_2}$, where g is the transfer function from control u to state x and $\hat{g}_P$ is the transfer function from control u to state $\hat{x}$ and where $\|\cdot\|_{\mathcal{H}_2}$ which is the $\mathcal{H}_2$ norm is the root-mean-square of the impulse responses of the system defined as follows:

$$\|g\|_{\mathcal{H}_2} = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} Tr\left(g(e^{j\omega})^\top g(e^{j\omega})\right) d\omega} \tag{1.3}$$

The $\mathcal{H}_2$ norm measures the energy of the impulse response and imposing $\|g - \hat{g}_P\|_{\mathcal{H}_2}$ to be small implies that the responses of system Σ and $\hat{\Sigma}_P$ are close. Thus, the system $\hat{\Sigma}_P$ can be used as an abstraction of the system Σ for control purposes since they have a similar behavior.
A continuous-time version of the problem was first presented in [59] followed by different extensions in [58, 60] and [61]. The unconstrained problem can be written as follows:

$$\min_{P \in \mathbb{R}^{\hat{n} \times n}} \|g - \hat{g}_P\|_{\mathcal{H}_2}, \tag{1.4}$$



(a) With this network the partition is connected: $I(P) \in \mathbb{I}_c(G)$.

(b) As $G_{\{1,2,4\}}$ and $G_{\{5,6\}}$ do not form weakly connected subgraphs, the partition is not connected: $I(P) \notin \mathbb{I}_c(G)$

Figure 1.4: Connected and disconnected partitions. The partition is the one defined in (1.6) applied to two different networks.

To formulate the constrained problem associated to (1.4) we introduce $I(P)$ the partition associated to the reduction matrix P. We have $I(P) := \{I_1, I_2, \dots, I_{\hat{n}}\}$ where I_l is a *part* of the partition defined as:

$$I_l := \{j \in \{1, \dots, n\}, P_{l,j} \neq 0\} \tag{1.5}$$

As an example, let P be the following partition matrix:

$$P = \begin{pmatrix} \star & \star & 0 & \star & 0 & 0 \\ 0 & 0 & \star & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \star & \star \end{pmatrix} \tag{1.6}$$

where $\star$ represents any non-zero real number, then $I(P) = \{\{1, 2, 4\}, \{3\}, \{5, 6\}\}$.

Definition 1.1 (Connected partition)
A partition I of a network G is a connected partition if and only if every subgraph G_{I_l} induced by a part I_l is weakly connected.
$\mathbb{I}_c(G)$ represents the set of connected partitions of a network G.

See fig. 1.4 for an example of connected and disconnected partitions. Using this definition, the constrained problem can be written as follows:

$$\min_{\substack{P \in \mathbb{R}^{\hat{n} \times n} \\ I(P) \in \mathbb{I}_c(G)}} \|g - \hat{g}_P\|_{\mathcal{H}_2} \tag{1.7}$$

In [58] the authors propose an algorithm to solve the problem defined in (1.4). In appendix A, we present this algorithm and its adaptation allowing to solve the constrained problem (1.7). In the next section, some simulations illustrate the price of connectedness in this case.

1.2.2 From the price of connectedness to the ratio of connectedness

We propose here to show the price of connectedness in the clustered model reduction via numerical simulations. Then we present an estimator of interest: the ratio of connectedness.

Simulation 1.1. Let us first introduce properly the directed Erdős-Rényi model that we will use.

The directed Erdős-Rényi model The model first introduced in [40] allows to generate random directed networks as follows:
We consider n vertices and each of the n^2 potential edges (self-loop are allowed) exists with a probability p. The expected value of number of edges is then pn^2 and the mean degree is pn. The degree distribution is known to be a Poisson law.

We consider a collection of 500 Erdős-Rényi graphs with $p = 0.2$ and $n = 100$ denoted $G_1, \dots, G_{500}$. Thanks to the algorithms presented in Appendix A, we solve the partitioning problems (1.4) and (1.7) for each network G_i and for different size of the reduced network $\hat{n}$. We denote by ϵ_u the mean error in the unconstrained problem and by ϵ_c the mean error in the constrained problem which is:

$$\epsilon_u = \frac{1}{500} \sum_{i=1}^{500} \min_{P \in \mathbb{R}^{\hat{n} \times n}} \|g - \hat{g}_P\|_{\mathcal{H}_2} \tag{1.8}$$

$$\epsilon_c = \frac{1}{500} \sum_{i=1}^{500} \min_{\substack{P \in \mathbb{R}^{\hat{n} \times n} \\ I(P) \in \mathbb{I}_c(G_i)}} \|g - \hat{g}_P\|_{\mathcal{H}_2} \tag{1.9}$$

Figure 1.5 shows the value of ϵ_u and ϵ_c as a function of the reduction factor $\frac{n-\hat{n}}{n}$.

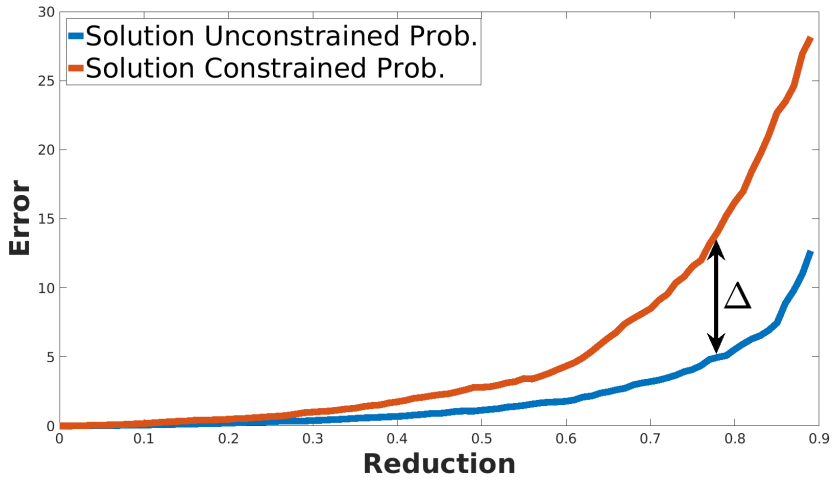


Figure 1.5: Estimation error in the unconstrained problem and in the constrained problem. The difference between the two results is the price of connectedness.

The relative difference between the two errors represented by an arrow in fig. 1.5 is *the*

price of connectedness denoted by Δ and defined as:

$$\Delta := \frac{\epsilon_c - \epsilon_u}{\epsilon_u} \tag{1.10}$$

To estimate the value of this difference by only knowing the structure of the network, we propose to use the ratio of connectedness presented hereafter.

The ratio of connectedness: a counting-based estimation

Beyond this example, the loss due to the connectedness constraint always exists in any partitioning problem. We propose here to tackle this question at a high level: we propose to estimate the price of connectedness for any couple of problems which can be formulated as an optimization problem.

$$\min_{I \in \mathbb{I}(G_0, M)} J_{G_0}(I) \qquad \text{and} \qquad \min_{I \in \mathbb{I}_c(G_0, M)} J_{G_0}(I) \tag{1.11}$$

where J_{G_0} is any cost function of the following form:

$$J_{G_0} \,:\, \mathbb{I}(G_0, M) \longrightarrow \mathbb{R} \tag{1.12}$$

and where $\mathbb{I}(G_0, M)$ is the set of partition of G_0 such that the part i has a size m_i; and $\mathbb{I}_c(G_0, M) \subset \mathbb{I}(G_0, M)$ is the set of connected partition. $M = (m_1, m_2, \ldots, m_{|I|})$ is called the *size vector* and verifies $\sum m_i = |\mathcal{V}_0|$
The two problems of (1.11) are respectively the Unconstrained Problem and the Constrained Problem. The approach presented here consists in comparing the cardinalities of $\mathbb{I}$ and $\mathbb{I}_c$ which are respectively the set of partition and the set of connected partition of G_0.
More precisely, given a network G having n nodes, and a vector M we want to estimate the ratio of connectedness defined as follows:

$$\rho(G, M) := \frac{|\mathbb{I}_c(G, M)|}{|\mathbb{I}(G, M)|} \tag{1.13}$$

This serves as a proxy to estimate the price of connectedness. Note that $\mathbb{I}(G, M)$ actually does not depend on G as the connections between nodes do not matter[3]. Before going deeper into the estimation of ρ, we will present in the next section some evidences on the link between the ratio of connectedness ρ and the price of connectedness Δ.

1.2.3 Relation between ρ and Δ in the clustered model reduction

A way to emphasize and make more precise the relation between ρ and Δ is to observe simulations on the clustered model reduction presented in the previous section. Precisely, we want to observe how the price of connectedness Δ evolves when simulations are done with

[3]Cardinality of $\mathbb{I}(G, M)$ is a known result [93]. However we will not use this result since we will derive an upper bound of ρ and not its exact value.

networks having a different ratio of connectedness ρ. For this purpose we consider 100 Erdős-Rényi graphs with $n = 100$ vertices and an edge probability p varying between 0 and 1[4]. Figure 1.6 shows the relation between ρ and Δ.

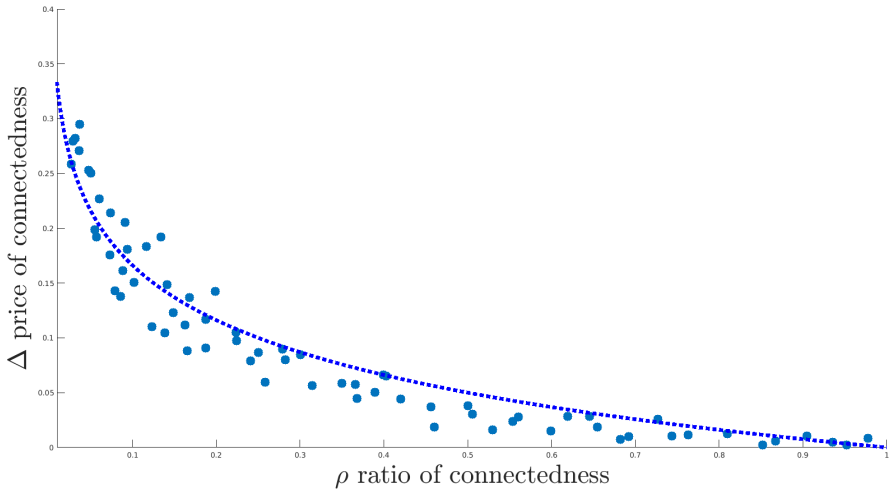


Figure 1.6: Relation between the price of connectedness and the ratio of connectedness. Each point correspond to an Erdős-Rényi graph with $n = 100$ and $p \in [0; 1]$. The ratio of connectedness, $\rho(G, M)$ is estimated by counting the number of connected partitions over the 500 partitions randomly generated (we fix arbitrarily $M = [5, 5, \ldots, 5]$). The price of connectedness $\Delta(G)$ is computed using (1.10) and Appendix A. In dotted blue the fitting function in (1.14).

By inspection it seems that, in this case, Δ can be approximately related to ρ via the following equation:

$$\Delta(\rho) = -\gamma \log(\rho) \tag{1.14}$$

with $\gamma \approx 0.17$. Moreover, this function verifies $\Delta(\rho) \longrightarrow \infty$ as $\rho \to 0$ and $\Delta(1) = 0$ which is what we expect from the relation[5] between Δ and ρ. While the relation (1.14) is intrinsically linked to the particular couple of problems (1.4)- (1.7) considered here, it is clear that Δ and ρ are generally related. This relation is intuitive when formulated as follows: the smaller the feasible set of the constrained problem, the larger the gap between the solutions of the constrained and unconstrained problems. Motivated by this example, we propose, in the next section, a tight upper bound on ρ in a (n, p)-Erdős-Rényi graph for a given partition size vector M.

[4]It is clear that the ratio of connectedness ρ grows with p: increasing p gives a network with more edges and so the number of connected partitions is higher. At the limit, if $p = 1$ (complete graph) then $\rho = 1$ and if $p = 0$ (null graph) then $\rho = 0$.

[5]When ρ goes to 0 the number of connected partitions goes to 0 and the optimal solution of the constrained problem worsens which is $\Delta \to \infty$. On the other hand, when $\rho = 1$, every partition is a connected partition (the graph is complete) and the solutions of the constrained and unconstrained problems are the same which is $\Delta = 0$.

1.3 Value of the ratio of connectedness in a (n,p)-Erdős-Rényi graph

In this section we develop the calculations leading to an estimation the ratio of connectedness in a directed (n,p)-Erdős-Rényi graph for a given partition size vector M.

1.3.1 Factorization formula

As we consider Erdős-Rényi graph, we denote by $\rho(n,p,M)$ the expected value of the ratio of connectedness of a (n,p) Erdős-Rényi graph. Noticing that drawing randomly m nodes in a (n,p) Erdős-Rényi graph is equivalent to generate a (m,p) Erdős-Rényi graph, we can decompose the price of connectedness as follows:

$$\rho(n,p,M) = \prod_{i=1}^{\hat{n}} \bar{\rho}(m_i,p) \tag{1.15}$$

where $\bar{\rho}(m_i,p)$ is the probability that a (m,p) Erdős-Rényi graph is weakly connected. We focus now on $\bar{\rho}(m,p)$ the probability of connectedness of a (m,p)-Erdős-Rényi graph. This question has been investigated in several works. Already the seminal paper of Erdős and Rényi [40] described a phase transition behavior for the value of this probability in the case of undirected graph. Several works [16, 47] investigate the phase transition behavior in Erdős-Rényi directed graph. However only results for strong connectedness are provided while we look here for weak connectedness. In particular, if we denote by $G_{n,p}$ a random (n,p)-Erdős-Rényi graph and $\mathcal{S}$ the set of strongly connected networks it is shown that:

$$\begin{cases} \mathbb{P}(G_{n,p} \in \mathcal{S}) \underset{n\to\infty}{\longrightarrow} 1 & \text{if } p > \frac{\ln n}{n} \\ \mathbb{P}(G_{n,p} \in \mathcal{S}) \underset{n\to\infty}{\longrightarrow} 0 & \text{if } p < \frac{\ln n}{n} \end{cases} \tag{1.16}$$

Where $\mathbb{P}(\chi)$ is the probability of χ. From our knowledge there is not such result for weak connectedness. However, it is clear that if such a threshold exists it should be smaller or equal to $\frac{\ln n}{n}$, because weak connectedness is a milder property than strong connectedness, and we presume that it is equal. Phase transitions results are pertinent for large-scale networks. However in the case of network partitioning the size of the parts are relatively small (which is m_i in (1.15) is small) and so these results can not be used. In the next section we propose an upper bound on $\bar{\rho}$: the probability of weak connectedness for a directed Erdős-Rényi graph.

1.3.2 Upper-bound on $\bar{\rho}$

To estimate the probability $\bar{\rho}(m,p)$, we propose to consider an upper bounding probability $\bar{\rho}_0(m,p)$ which is the probability that in a (m,p)-Erdős-Rényi graph there is no isolated node (node with zero indegree and outdegree). The set of connected networks is included in the set of networks without isolated nodes, which implies that

$$\bar{\rho}_0(m,p) > \bar{\rho}(m,p) \tag{1.17}$$

Some works have emphasized a phase transition behavior for the probability of presence of isolated nodes [44, 56]. These results are only for undirected networks and are valid for large-scale networks. The following proposition gives an exact expression for $\bar{\rho}_0$:

Proposition 1.1
Let $\bar{\rho}_0(m,p)$ be the probability that a directed (m,p)-Erdős-Rényi graph has no isolated nodes. We have

$$\bar{\rho}_0(m,p) = 1 - \sum_{k=0}^{m^2-2} \theta(m,k) p^k (1-p)^{m^2-k} \tag{1.18}$$

where $\theta(m,k)$ is the number of directed networks with m nodes and k edges and with (at least) one isolated node.

Proof of Proposition 1.1 Let us denote Γ_m the set of networks with m nodes having isolated nodes and $\bar{\Gamma}_m$ its complement which is the set of network with m nodes and no isolated nodes. Let $G_{m,p}$ be a random Erdős-Rényi graph

$$\bar{\rho}_0(m,p) = \mathbb{P}(G_{m,p} \in \bar{\Gamma}_m) \tag{1.19}$$
$$= 1 - \mathbb{P}(G_{m,p} \in \Gamma_m) \tag{1.20}$$

Now if we denote Γ_m^k the set of networks of size m with isolated nodes and k edges we have:

$$\bar{\rho}_0(m,p) = 1 - \sum_{k=1}^{m^2} \mathbb{P}(G_{m,p} \in \Gamma_m^k) \tag{1.21}$$

Every network in Γ_m^k has the same probability to appear which is the probability that a (m,p)-Erdős-Rényi graph has k edges. Hence we have $\mathbb{P}(G_{m,p} \in \Gamma_m^k) = |\Gamma_m^k| p^k (1-p)^{m^2-k}$. Let us denote $\theta(m,k) := |\Gamma_m^k|$. Noting that $\theta(m,m^2) = \theta(m,m^2-1) = 0$ leads to the result. □

Therefore, we obtain a value for $\bar{\rho}_0$ depending on the value of $\theta(m,k)$. By reusing the factorization formula in (1.15) we deduce ρ_0 an upper bound on ρ which is:

$$\rho_0(n,p,M) = \prod_{i=1}^{\hat{n}} \bar{\rho}_0(m_i,p) \tag{1.22}$$

To complete the calculations, the next result specifies the value of $\theta(m,k)$

Proposition 1.2
Let $\theta(m,k)$ be the number of networks with m nodes, k edges and (at least) one isolated node. We have

$$\theta(m,k) = \begin{cases} \binom{m^2}{k} & \text{if } k < \dfrac{m}{2} \\ \displaystyle\sum_{s=1}^{\lfloor m-\sqrt{k} \rfloor} \binom{m}{s} \left[\binom{(m-s)^2}{k} - \theta(m-s,k) \right] & \text{if } k \geq \dfrac{m}{2} \end{cases} \tag{1.23}$$

Proof of Proposition 1.2 When k the number of edges of a network is strictly smaller than $\frac{m}{2}$ it is impossible that the k edges cover the m nodes. Thus, any network with m nodes and $k < \frac{m}{2}$ edges contains (at least) one isolated node. The number of such network is equal to the number of way to arrange k edges among the m^2 possible edges. This number is equal to $\binom{m^2}{k}$.
Let's denote $\theta_s(n,k)$ the number of networks with n nodes, k edges and exactly s isolated nodes. We have:

$$\theta(m,k) = \sum_{s=1}^{m} \theta_s(m,k) \tag{1.24}$$

Let us remark that $s > m - \sqrt{k} \implies \theta_s(m,k) = 0$ because it is not possible to have k edges in less than $\sqrt{k}$ nodes. Hence we can rewrite the previous equation as:

$$\theta(m,k) = \sum_{s=1}^{\lfloor m-\sqrt{k} \rfloor} \theta_s(m,k) \tag{1.25}$$

Moreover, θ_s can be rewritten as follows:

$$\theta_s(m,k) = \binom{m}{s} \sigma(m-s,k), \tag{1.26}$$

where $\sigma(m-s,k)$ is the number of networks with $m-s$ nodes, k edges and no isolated node. This value is multiplied by the number of ways to choose the s isolated nodes within the m nodes, which is $\binom{m}{s}$. We have then:

$$\theta(m,k) = \sum_{s=1}^{\lfloor m-\sqrt{k} \rfloor} \binom{m}{s} \sigma(m-s,k) \tag{1.27}$$

Finally, σ the number of networks without isolated nodes is equal to the total number of networks minus the number of networks with at least one isolated node, and so: $\sigma(m-s,k) = \binom{(m-s)^2}{k} - \theta(m-s,k)$. Leading to the result. □

The same reasoning can be used to solve the problem for a directed network without self-loops or for an undirected network with or without self-loops. In these cases the m^2-2 in the limit of the sum in (1.18) should be replaced respectively with $m(m-1)-2$, $m(m+1)/2-1$ and $m(m-1)/2-1$ and $\binom{m^2}{k}$ the total number of network with m nodes and k edges should be replaced respectively with $\binom{m(m-1)}{k}$, $\binom{m(m+1)/2}{k}$ and $\binom{m(m-1)/2}{k}$.
Considering that there are *relatively* few networks which are not connected while having no isolated nodes, the two probabilities $\bar{\rho}(m,p)$ and $\bar{\rho}_0(m,p)$ appears to be really close. Actually, the inital work from Erdős and Rényi [39] implies that the two probabilities tend towards each other for large m:

Proposition 1.3
Let $\bar{\rho}(m,p)$ be the probability that a random (m,p)-Erdős-Rényi graph is connected and $\bar{\rho}_0(m,p)$ be the probability that a random (m,p)-Erdős-Rényi graph has no isolated nodes, then:

$$\lim_{m\to\infty} \frac{\bar{\rho}(m,p)}{\bar{\rho}_0(m,p)} = 1 \tag{1.28}$$

This result straightforwardly leads to this second result:

$$\lim_{n\to\infty} \frac{\rho(n,p,M)}{\rho_0(n,p,M)} = 1 \tag{1.29}$$

In the next section we present some simulations to validate the closeness of the upper bound proposed. Before this, fig. 1.7 sums up the approach developed throughout this chapter.

1.4 Validation of the upper bound

To illustrate the relevance of the upper bound used for the probability of weak connectedness in an Erdős-Rényi graph, we compare an experimental value of $\bar{\rho}(m,p)$ and the theoretical value of $\bar{\rho}_0(m,p)$ given in (1.18). To find the experimental value we generate 1000 (m,p)-Erdős-Rényi graphs for each value of of $m = \{6,8,10,12\}$ and $p = \{0, 0.02, 0.04, \ldots, 1\}$. An approximation of $\bar{\rho}(m,p)$ consists in counting the proportion of the 1000 networks that are connected. The results are presented in fig. 1.8. We notice that i) the inequation $\bar{\rho}_0 \geq \bar{\rho}$ is well verified and ii) the upper bound $\bar{\rho}_0$ is relatively close to $\bar{\rho}$ and the difference seems to vanish when m grows as predicted by Proposition 1.3. Let us note that the formula given in Proposition 1.2 does not allow to compute $\theta(m,k)$ for large value of m. Indeed the recursive call to the function θ is computationally heavy and the value of the binomial coefficient grows quickly.
Now that we have seen that $\bar{\rho}_0$ is tightly upper bounding $\bar{\rho}$ we want to come one step back and verifying that the upper bound ρ_0 derived from $\bar{\rho}_0$ by way of (1.22) is also tightly upper bounding ρ. To compute an experimental value of $\rho(n,p,M)$, as before, we generate 1000 Erdős-Rényi with $n = 100$ nodes for each value of p varying between 0 and 1. For each network we generate a random partition with the part sizes $m_1 = m_2 = \ldots = m_{20} = 5$. An approximation of ρ consists in counting the proportion of the 1000 networks whose partition is connected. The results are presented in fig. 1.9. We verify that the upper bound is indeed very close to the real value. Though for few value of p it seems that $\rho(n,p,M) > \rho_0(n,p,M)$, it is only due to statistical noise.

1.5 Conclusion

When looking at network partitioning problems, it is clear that imposing a connectedness constraint deteriorates the quality of the solution. However, this constraint is essential for some applications (in particular when the network has an underlying geographical nature). We presented here an approach based on probability and counting allowing to estimate *the price of connectedness* which is the degradation due to the connectedness constraint. This is a high-level approach in the sense that it does not focus on the particular optimization problem, only on the cardinalities of the feasible sets in the constrained and unconstrained cases. The drawback of this high-level approach is that, the results are not precise when considering a particular problem. Then an extension of this work would be to investigate to what extent,

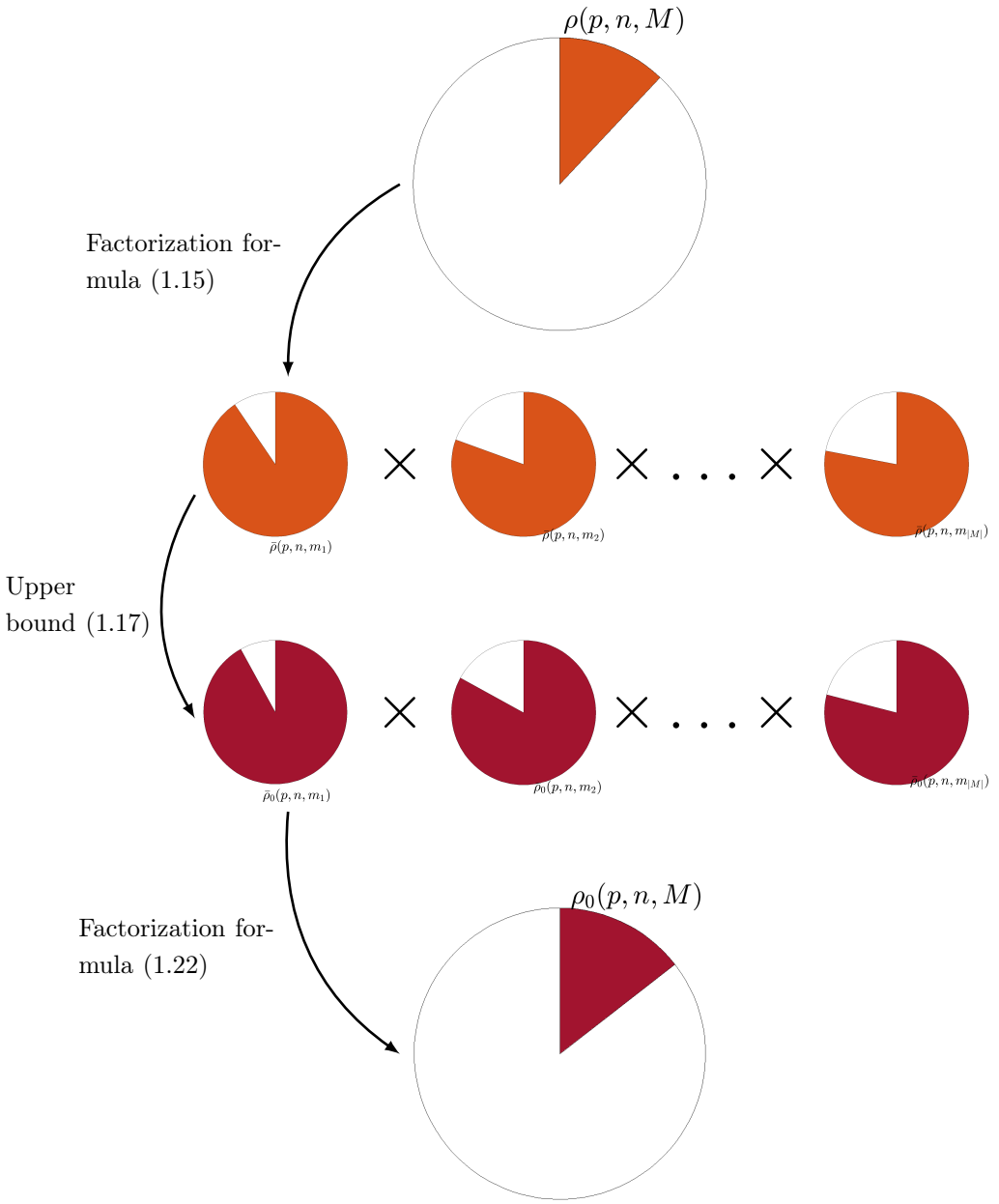


Figure 1.7: This diagram illustrates the approach proposed in this chapter. In order to estimate the price of connectedness we estimate the ratio of connectedness $\rho(p,n,M)$: line (1). Based on (1.15), we factorize the problem: line (1) to line (2). Then the upper bound (1.17) allows the passage from line (2) to line (3). The rest of the chapter present the calculations leading to Propositions 1.1 and 1.2 giving the value of $\bar{\rho}_0$. Finally the factorization formula (1.22) lead to the final upper bound of the ratio of connectedness: line (3) to line (4).

the ratio of connectedness influences the price of connectedness. In other words, the study of the function $\Delta(\rho)$ which depends on the particular problem considered. Another extension of

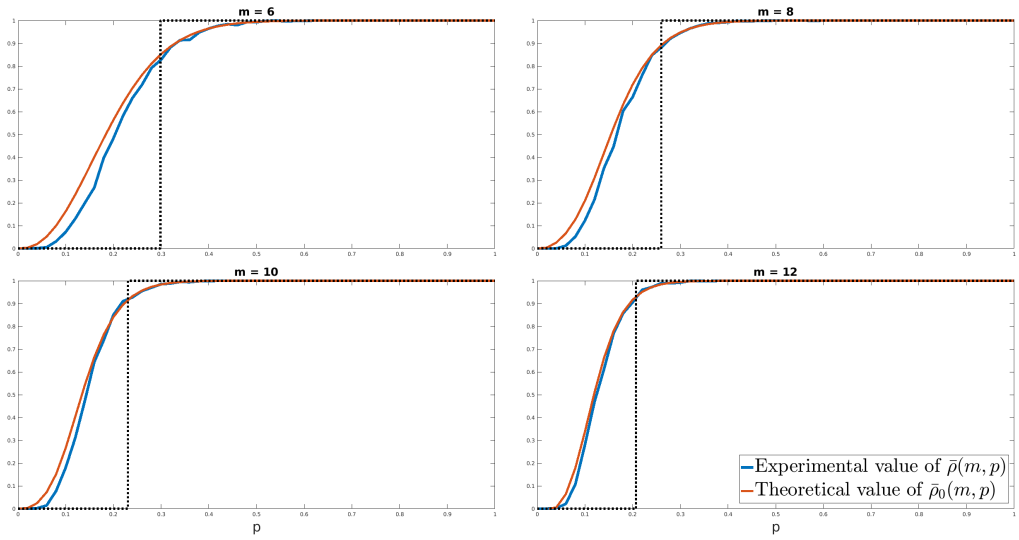


Figure 1.8: Comparison between the experimental value of $\bar{\rho}(m,p)$ (the probability that a given (m,p)-Erdős-Rényi graph is connected) and the theoretical value of the upper bound $\bar{\rho}_0(m,p)$ (the probability that a given (m,p)-Erdős-Rényi graph has no isolated node) for $p \in [0,1]$ and for different values of m. The phase transition behavior detailed in (1.16) is represented in dotted black: when m grows $\bar{\rho}(m,p)$ gets closer to the transition.

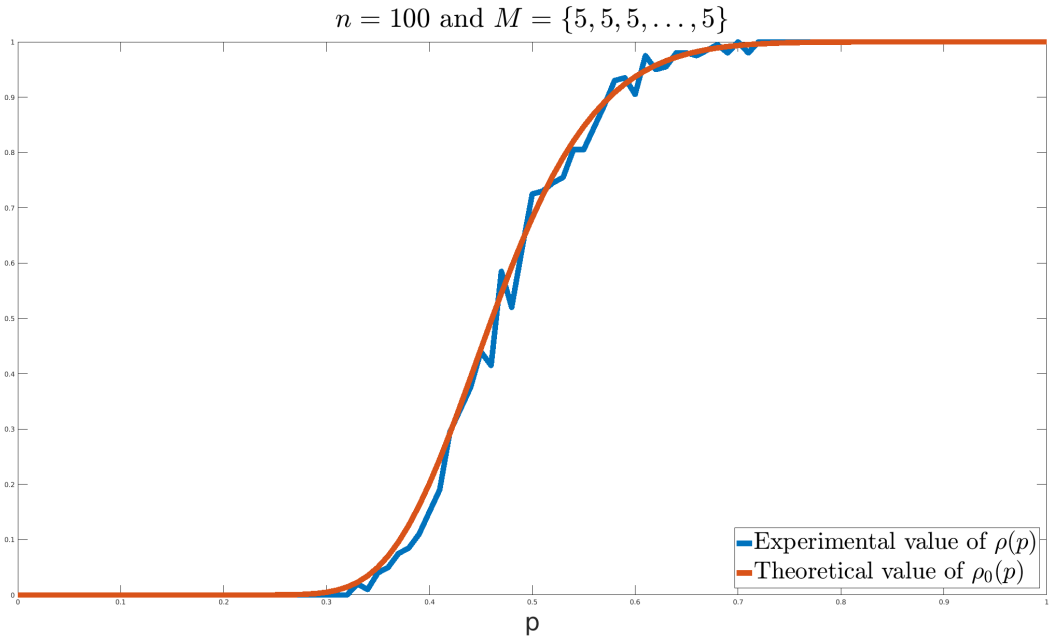


Figure 1.9: Comparison between the experimental value of $\rho(100,p,\{5,\ldots,5\})$, and the theoretical value of the upper bound $\rho_0(100,p,\{5,\ldots,5\})$ for $p \in 0:0.005:1$.

this work would be the simplification of the recursive formula given in Proposition 1.2 allowing to make it computable for large-scale networks.
This chapter and in particular the last results shows us the effect of the connectedness con-

straint with respect to the the topology. The price of connectedness is negatively correlated to the density of the network. In the scenario considered in fig. 1.9 if $p < 0.3$ (which means that each nodes has less than 30 connections in average), the ratio of connectedness is almost null which means that the number of connected partitions is negligible compared to the number of partitions and the difference between the constrained and unconstrained problems will be considerable. At the contrary, when $p > 0.7$ (which means that each nodes has more than 70 connections in average), the ratio of connectedness is almost equal to one which means that almost any partition is a connected partition and the difference between the constrained and unconstrained problems is negligible.
Throughout the thesis we will apply results about partitioning to urban traffic networks and epidemic networks. As discussed in the introduction, both of these applications gain to be partitioned with the connectedness constraint. This preliminary analysis enlighten us regarding the loss due to this physical constraint.

Chapter 2

Network partitioning algorithm towards scale-free structure

Contents

As seen in Introduction, scale-free networks endow interesting properties which are helpful in different contexts. Moreover, if a network does not have this particular structure, we claim that it is interesting to consider a scale-free network abstracting the initial one. To this end, this chapter presents the results we obtained regarding the problem of network partitioning inducing scale-freeness. Further in this thesis, Section 5.2 and Chapter 4 present practical implementations of this approach are proposed. A primary version of the result introduced in this chapter has been presented at the 2018 European Control Conference in Limassol, Cyprus [83] and a more complete version has been published in Transaction on Network Science and Engineering [84]. A video popularizing this approach has also been produced[1].

[1]The video is available here: `https://www.youtube.com/watch?v=UXc76Z5Ek3M&t=56s`

2.1 Introduction

The approach proposed in this chapter combines several objectives: starting with a large-scale initial network, we aim to reduce it into a scale-free network while preserving characteristics of the initial network. As seen in the introduction, interesting properties emerge naturally from the definition of scale-free networks [95]. We recall here some of those properties: small distances between nodes [27], robustness to random failures, easiness to disconnect, hyperbolic space embedding [14]. These properties allow to gain efficiency in some applications such as network navigation, vaccination in epidemiology, or control design as developed in Appendix 0.3.4. In order to reap these advantages, we propose in this chapter to find a network abstracting an arbitrary large-scale network and exhibiting a scale-free structure. However, we do not investigate here to what extent the scale-freeness of the reduced network may be meaningful for the initial network. We focus only on the design of a reduction method whose output is a scale-free network. In Section 5.2 though, we will present an application where the scale-freeness of the abstracting network is useful.
The main contributions of this chapter are as follows:

(i) The introduction of a framework to treat a class of network partitioning problem along with a meta-algorithm to solve such problems.

(ii) The analysis of two particular problems falling in the scope of the framework. In particular, the second problem requires the analysis of several properties of network.

(iii) The analysis of the performance of the algorithm and the tuning of the parameters studied on synthetic networks.

The chapter is organized as follows: Section 2.2 introduces a class of problem and gives a general approach to solve them, Sections 2.3 and 2.4 presents two particular problems of this class: the first one is rather simple and has a pedagogic and illustrative purpose. The second one implies the analysis of the preservation of different network-related properties. Section 2.5 gives some results on the algorithm proposed in 2.4 and a simulation of this algorithm applied on an academic case. Finally, Section 2.6 concludes the chapter.

2.2 Framework for a class of problems

This section is devoted to the introduction of a class problems of network partitioning inducing scale-freeness while preserving features of the initial network. This type of problem can be seen as an optimization problem with some constraints. We propose then a general method to solve this type of problem. The problem formulation along with the solving method form a framework that we will use to treat two particular problems in the two following sections.

2.2.1 Targeting scale-freeness as an optimization problem

The problem treated in this chapter is to find a partition $\mathcal{S}$ of an initial large network G_0, such that the network G_1 coming out of the partition $\mathcal{S}$ has a degree distribution *close* to a given scale-free distribution. We also want G_1 to preserve some physical properties of G_0 and guarantee a certain *similarity* between G_0 and G_1. Therefore, this problem can be formally stated as follows:

Problem 2.1
Given an initial network $G_0 \in \Psi$, and a desired scale-free coefficient α, find a network G_1, solution of the following minimization problem.

$$\min_{G} \ \mathrm{J}_{\mathrm{SF}_\alpha}(G), \quad \textit{subject to} \quad \begin{array}{l} G_0 \succ G \\ G \in \Psi \\ G_0 \equiv G \end{array} \tag{2.1}$$

where:

- $\mathrm{J}_{\mathrm{SF}_\alpha}$ *is a α-scale-free cost function ($\mathrm{J}_{\mathrm{SF}_\alpha}(G) = 0$ meaning that G is perfectly scale-free with a coefficient equal to α)*
- Ψ *is the set of networks respecting the physical properties imposed.*
- $\equiv$ *is a binary relation translating a certain similarity between the initial and final network.*

The physical constraint ($G \in \Psi$) and the similarity constraint ($G_0 \equiv G$) does not have any mathematical difference and one could have gather them. We separate them in the formulation to emphasize that the underlying motivations for both is different.
This formulation includes a large number of partitioning problems depending on Ψ the properties to preserve and $\equiv$ the binary relation considered. Note also that any scale-free cost function can be used to steer the reduce network towards this type of network. We present hereafter a simple scale-free cost function that we will use.

2.2.2 Scale-free cost function

We present here an intuitive measure of scale-freeness that we will use throughout this chapter. We first define what is the ideal scale-free distribution and then we explain how we measure the distance to this ideal case.

Definition 2.1 (Scale-free target distribution)
The α-scale-free target distribution of range n, denoted $\Pi^{SF}_{\alpha,n}$ corresponds to the most scale-free

distribution that a network of size n could have. It is defined as follows:

$$\Pi^{SF}_{\alpha,n} = \frac{1}{\sum_{i=1}^{k_{cut}} i^{-\alpha}} \begin{pmatrix} 1^{-\alpha} \\ 2^{-\alpha} \\ \vdots \\ k_{cut}^{-\alpha} \end{pmatrix} \tag{2.2}$$

where k_{cut} is a cut-off calculated as the highest degree for which the number of nodes having this degree is higher than 1 in a α-scale-free network of range n:

$$k_{cut} = \underset{k}{\operatorname{argmax}} \left\{ \frac{k^{-\alpha}}{\sum_{i=1}^{k} i^{-\alpha}} \geq \frac{1}{n} \right\} \tag{2.3}$$

The normalization term $\frac{1}{\sum_{i=1}^{k_{cut}} i^{-\alpha}}$ allows us to compare degree distributions of any size with the target distribution. The cutoff k_{cut} is needed more for technical reasons: it avoids to penalize a distribution for negligible value in the tail of the target distribution and it fixes a limit for the practical computation of the distribution. We define now the scale-free cost function:

Definition 2.2 (Scale-free cost function)
For any network G the scale-free cost function is the $\mathcal{L}_2$ relative distance to the target distribution.

$$\mathrm{J}_{\mathrm{SF}_\alpha}(G) = \frac{\left\| \Pi_G - \Pi^{SF}_{\alpha,|G|} \right\|_2}{\|\Pi_G\|_2}, \tag{2.4}$$

where $\Pi^{SF}_{\alpha,|G|}$ is the scale-free target distribution defined as in (2.2) and $|G|$ is the number of nodes in G.

Let us note that Π_G and $\Pi^{SF}_{\alpha,|G|}$ are not necessarily of the same size. In this case, zeros are added at the end of the smallest vector so that the sizes match. Figure 2.1 illustrates this cost function.

It is also possible to impose a scale-free distribution both on the indegree and the outdegree with two different coefficients α_{in} and α_{out}. In this case the scale-free cost function is:

$$\mathrm{J}_{\mathrm{SF}_\alpha}(G) = \mu_{in} \frac{\left\| \Pi_{in} - \Pi^{SF}_{\alpha_{in},|G|} \right\|_2}{\|\Pi_{in}\|_2} + \mu_{out} \frac{\left\| \Pi_{out} - \Pi^{SF}_{\alpha_{out},|G|} \right\|_2}{\|\Pi_{out}\|_2} \tag{2.5}$$

where $\mu_{in}, \mu_{out} > 0$ are two coefficients allowing to adjust the relative importance to steer one distribution with respect to the other one. Having two different scale-free coefficients for indegree and outdegree can be useful: if the network considered is endowed with a linear dynamical system as in (19), the work in [75] suggests that different scale-free coefficients for indegree and outdegree distribution can be useful to improve controllability.
Note that the mathematical results presented further are independent of the scale-free cost function chosen. Hence, any definition of the cost function could be used without questioning the validity of the results. However, as we are going to see, this definition gives sufficiently good results.

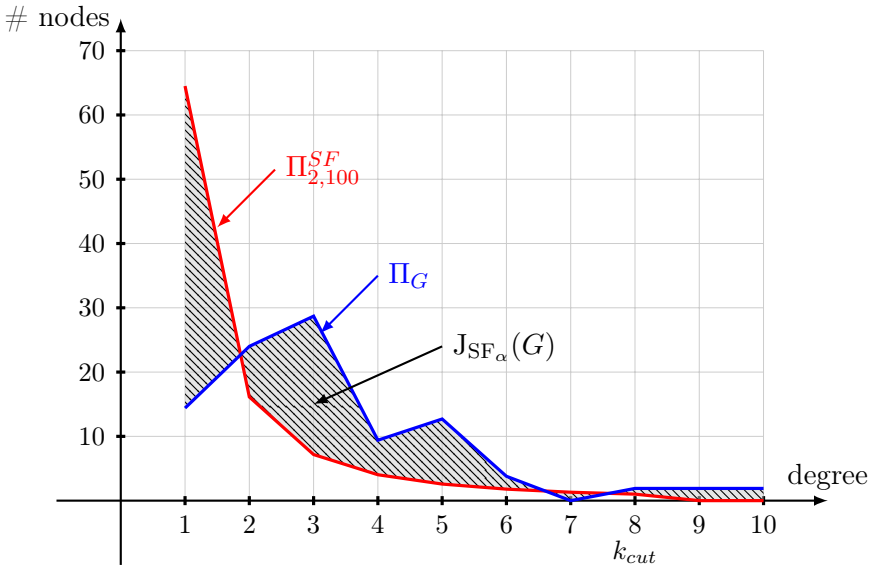


Figure 2.1: The scale-free cost function measures the difference between the degree distribution Π_G and the target scale-free distribution which is here $\Pi^{SF}_{2,100}$. By minimizing the dashed zone the reduce networks tends to be scale-free.

2.2.3 A meta-algorithm to solve the optimization problem

In this section, we present a general method providing a sub-optimal solution to problems of the class presented in the previous section. This is a meta-algorithm in the sense that it gives the skeleton of the method, and for each specific problem, some specifications have to be provided. A way to solve exactly this problem is to explore all the partition of a given network and select the one which minimizes the scale-free cost function and respects the constraints. However, as mentioned in Chapter 1 the number of partitions of a network is very large (even for connected partitions) and the exploration of all of them is computationally unfeasible. We propose hence an iterative algorithm in which at each step we look for the merging $\mathcal{S}_{v,w}$ minimizing the scale-free cost function[2]. As presented in introduction, a merging is a particular type of partition where only two nodes are gathered together. Therefore, instead of exploring once the set of all partitions we explore iteratively the set of merging.
By doing this, the number of computations is hugely reduced. Here is an insight of this economy of computation: the number of partitions of a network with n nodes in k parts is called the *Stirling number of the second kind*, and its order of magnitude is k^{n-k} [107]. In contrast, the number of merging of a network with n nodes is n^2 and to reach k parts, the research has to be iterated $n-k$ times. Therefore, via this approximation, the number of candidates to test goes down from about k^{n-k} to $(n-k)n^2$, which is a huge improvement when considering large-scale network. As an example, to partition a network of $n = 100$ nodes in $k = 20$ parts, the exploration of all partitions would require around 10^{104} test and an iterative exploration of merging would require around 8.10^5 tests. In the particular case of connected partitions,

[2]This is why we just need the result of Theorem 2.2 for merging.

the computations would be more complicated (the estimation found in Chapter 1 could be used) but the gain would be of the same magnitude.
By using this approximation, the Scale-Free cost-function is only minimized at each iteration. This is a greedy algorithm, and it will not find the best solution of Problem 2.1. We will see, though, that the networks obtained with this method are sufficiently close to a scale-free structure. Furthermore, even with this approximation, when dealing with large-scale networks, looking for the best merging $\mathcal{S}_{v,w}$ within all pair of nodes (v, w) still requires heavy computations. As pointed above the number of tests is cubic with respect to the number of nodes. For a network with ten thousand nodes, it would require around 10^{12} tests, which is unreachable in reasonable time. Therefore, to cut this complexity, the best merging $\mathcal{S}_{v,w}$ is searched among a relatively small random subset of pair of nodes (v, w). The effect of the approximation due to this random selection will be discussed later.
Based on this, we present now the meta-algorithm providing a sub-optimal solution to Problem 2.1. Algorithm 1 describes our approach; therein G_k is the network at step k and is represented by the triple (A_k, E_k, V_k). The inputs are the initial network $G_0 \in \Psi$, a scale-free coefficient $\alpha_{SF} > 0$ and an integer $n_{\text{rand}} \in \mathbb{N}$ corresponding to the the size of the random subset of pair of nodes. As explained previously, the algorithm consists in a loop (line 2)

Algorithm 1 Meta algorithm - Merge To Scale-Free

Input: G_0, α_{SF}, n_{rand}
Output: G_{end}
1: $k = 0$
2: **while** $\neg stop$ **do**
3: $\quad \Omega \longleftarrow n_{\text{rand}}$ random elements of $V_k \times V_k$ verifying constraints of Problem 2.1
4: $\quad$ **for** $(v, w) \in \Omega$ **do**
5: $\quad\quad n_{SF}(\mathcal{S}_{v,w}) = \mathrm{J}_{\mathrm{SF}_\alpha}(\mathrm{Merge}(G_k, \mathcal{S}_{v,w}))$
6: $\quad$ **end for**
7: $\quad \mathcal{S}_{best} = \text{argmin } n_{SF}(\mathcal{S}_{v,w})$
8: $\quad G_{k+1} = \mathrm{Merge}(G_k, \mathcal{S}_{best})$
9: $\quad k = k + 1$
10: **end while**

decomposed as follows: first a random subset Ω of n_{rand} pair of nodes is generated (line 3). Then for each pair of nodes (v, w) in Ω, the scale-freeness of the network obtained by merging $\mathcal{S}_{v,w}$ in the network G_k is computed (line 5). Finally, the best merging under the constraints of the problem is selected (line 7) and the new network is obtained by operating this merging (line 8).
The function Merge associates to a network G and a merging $\mathcal{S}$, a new network obtained G_1 verifying $G_0 \succ^{\mathcal{S}} G_1$. Function Merge may include the computation of the weights of the new network in order to respect the constraints of the problem. The stopping criterion *stop* is not discussed here, it may be defined as the step where it is no more possible to find a merging that decreases the scale-free cost function, or as a fixed number of iterations if the size of the reduced network is given.
Let us note that if we want to obtain a connected partition it is sufficient to consider only

pair of connected nodes at each step: the line 3 is thus replaced by

$$\Omega \longleftarrow n_{\text{rand}} \text{ random elements of } E_k. \tag{2.6}$$

The loss due to this modification has been discussed in Chapter 1. For now on, we will mainly focus on the connected partition case, as the problem posed in this chapter is more addressed to traffic networks with a geographical nature.
The class of problem 2.1 and the Algorithm 1 form the framework in which the two following sections take place. This framework will also be used in an application to epidemic developed in 5.2 where we take advantage of the scale-freeness of the abstracting network to design a vaccination strategy.

2.3 A first instance of the problem fostering node similarity

In this section, we present a first problem fitting the class of problem 2.1. We consider here a network, in which a value is attributed to each node. Besides the scale-free objective, we want to obtain a partition in which the value of nodes within a same part are close. This section does not present any major contribution but has a didactic purpose to manipulate the framework presented before. It will also allow to illustrate the algorithm with an online animation.

2.3.1 Specific problem formulation

A way to ensure a coherency in the reduced network G_1 is to gather only nodes of G_0 having a close value. For example, in a social network we may want to merge only individuals of the same age, or in a road traffic network to merge only roads with a similar density. We denote $\{x_1, \ldots, x_n\}$ the value attributed to each node of G_0 and $\mathcal{S}$ the partition of G_0 defined as:

$$\mathcal{S} = \{S_1, S_2, \ldots, S_p\} \quad \text{where } S_k = \{v_{i_1}, \ldots, v_{i_{p_k}}\} \subset V_0 \tag{2.7}$$

In order to define the constraint we introduce the following notion.

Definition 2.3 (Divergence)
Considering a network G_0 and a partition $\mathcal{S}$, we define $\Delta(G_0, \mathcal{S})$ the divergence of G_0 in $\mathcal{S}$ the following quantity:

$$\Delta(G_0, \mathcal{S}) = \max_{S_k \in \mathcal{S}} \max_{i,j \in S_k} |x_i - x_j| \tag{2.8}$$

The divergence of G in $\mathcal{S}$ is the maximal difference between two nodes in a same part. We want to ensure that two nodes in a same part have at maximum a distance of ϵ. This leads to the following formulation of the problem:

Problem 2.2
Given an initial network G_0, a desired scale-free coefficient α, and a threshold ϵ, find a network

G_1, solution of the following minimization problem.

$$\min_{G} \ \mathrm{J}_{\mathrm{SF}_\alpha}(G), \quad \textit{subject to} \quad \begin{array}{l} G_0 \succ G \\ \Delta(G_0, S) < \epsilon \end{array} \tag{2.9}$$

Therefore, $\Delta(G_0, S) < \epsilon$ is the binary relation translating the similarity we want to preserve. Let us note that unlike the formulation of the general Problem 2.1 the similarity constraint here concerns directly the initial network G_0 and the partition $\mathcal{S}$. Actually, this constraint concerns indirectly the graph G_1 as this one depends only on G_0 and S. This difference is negligible and an equivalent formulation of Problem 2.2 could make appear G_0 and G_1.

2.3.2 Specific algorithm

Based on the skeleton given in Algorithm 1, we propose here an algorithm providing a sub-optimal solution to Problem 2.2. To ensure that two nodes in a same part can not respect the distance ϵ, at each step the lower and higher value of the nodes composing a part must be known. Therefore, we associate to the nodes v of the current network G_k, the couples (m_v, M_v) respectively the minimum and maximum value of the nodes merged in v. This algorithm is obtained by adding a block *if..then..else* to test the difference between the nodes of the tested edge.

Algorithm 2 Merge To Scale-Free with node similarity

Input: G_0, α_{SF}, n_{rand}, ϵ
Output: G_{end}

```
k = 0
while ¬stop do
   Ω ⟵ n_rand random elements of E_k
   for (v, w) ∈ Ω do
      if |M_v − m_w| < ε and |m_v − M_w| < ε then
         n_SF(S_{v,w}) = J_{SF_α}(Merge(G_k, S_{v,w}))
      else
         n_SF(S_{v,w}) = +∞
      end if
   end for
   S_best = argmin n_SF(S_{v,w})
   G_{k+1} = Merge(G_k, S_best)
   k = k + 1
end while
```

Where, the function $\mathrm{Merge}(G_k, \mathcal{S}_{best})$ is as follows:

$$\begin{array}{rl} \mathrm{Merge}(G, \mathcal{S}_{best}) \ : \ \Gamma_n \times \mathcal{E}_k & \longrightarrow \Gamma_{n-1} \\ G, (v, w) & \longmapsto G' = (\,\cdot\,, \mathcal{V}', \mathcal{E}') \end{array} \tag{2.10}$$

where

$$\begin{aligned} \mathcal{V}' &= (\mathcal{V}\backslash\{v,w\}) \cup vw \\ \mathcal{E}' &= (\mathcal{E}\backslash\{v,w\}) \cup \{(vw, \mathcal{N}_{out}(v) \cup \mathcal{N}_{out}(w))\} \cup \{(\mathcal{N}_{in}(v) \cup \mathcal{N}_{in}(w)), vw\} \\ m_{vw} &= \min(m_v, m_w) \\ M_{vw} &= \max(M_v, M_w) \end{aligned} \tag{2.11}$$

An animation illustrating an execution of Algorithm 2 is available online[3]. In this video, the color of the nodes indicates their value between 0 and 1 and the parameters are fixed as follows $\epsilon = 0.2$, $n_{rand} = 10$ and $\alpha_{SF} = 2.3$. The convergence of the degree-distribution (plotted in a log-log representation) towards a scale-free distribution appears clearly. However, we can also see that at the very, when the reduced network becomes too small the degree distribution starts to get apart of this objective. A way to explain is to imagine that at the limit the reduced network is reduced to an only node. At this point, the network is not scale-free anymore. Therefore, after passing an optimum the network will get further to a scale-free structure. Figure 2.2 illustrate this evolution of the scale-free cost function.

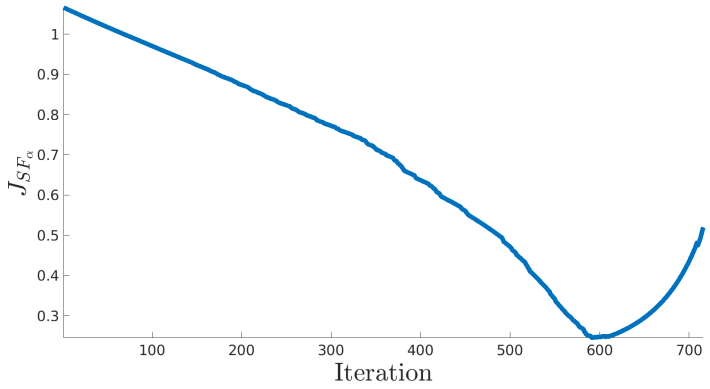


Figure 2.2: Evolution of the scale-free cost function through the algorithm. The evolution is averaged over 100 simulation on a same network of size 1000 with $n_{rand} = 50$.

2.4 Property preserving problem

In this section we present a less trivial problem, implying constraints on the dynamics of the network. In order to design an algorithm respecting these constraints we will first present several intermediate results. This problem and its solving represent the main contribution of this chapter. Sections 2.5 and 2.1 will present additional results and simulations about this problem.

[3]see `https://www.youtube.com/watch?v=h3NUfM5weP4`

2.4.1 Preliminaries

We first present a couple of concepts that will allow us to specify the merge to scale-free problem.

Eigenvector distance

In order to compare the behavior of two networks, we will consider the eigenvector centrality defined hereafter.

Definition 2.4 (Eigenvector centrality)
The vector of eigenvector centrality is a vector, denoted $x^\star$, associating to each node of the network its relative importance. It is the only vector[4] *verifying:*

$$x^\star = P^\top x^\star \quad \textit{and} \quad \|x^\star\|_1 = 1 \tag{2.12}$$

where P is is the adjacency matrix normalized by rows which is:

$$P_{i,j} = \frac{A_{i,j}}{\sum\limits_k A_{i,k}}, \tag{2.13}$$

This centrality can be viewed as the ratio of time spent by a random walker[5] on each node of the network. This centrality is also the steady state of the following discrete dynamical equation:

$$\begin{cases} x(k+1) = P^\top x(k) \\ x(0) = x_0 \end{cases} \tag{2.14}$$

for any value of x_0.

We denote Φ the operator associating a network in with its eigenvector centrality:

$$\Phi : G \longmapsto x^\star, \quad \text{s.t.} \quad x^\star = P^\top x^\star \text{ and } \|x^\star\|_1 = 1 \tag{2.15}$$

Let us note that strong connectivity is a necessary condition for computing the eigenvector centrality and hence to develop the rest of our analysis. Therefore, for now on, we consider, in this section, that the initial network G_0 is strongly connected. Let us note that traffic networks, one of our applications of interest, are always strongly connected (one can picture that it is always possible to reach any point from any point via roads). To compare a network G_0 with a network G_1 issued from the partition $\mathcal{S}$ of G_0 based on their eigenvector centrality, we need an operator of projection σ_S. The projection corresponds to the sum of the components within each part of the partition. We give here a precise definition:

[4] The existence and uniqueness is ensured by the Perron-Frobenius theorem [103]

[5] When he is on a node, the walker chooses randomly his next step among the neighbors of the node with a probability proportionally to the weights on the edges.

Definition 2.5 (Projection operator)
Let $x \in \mathbb{R}^n$ and S a partition of the set $\{1, .., n\}$, we define the projection operator σ_S as:

$$\begin{aligned} \sigma_S : \mathbb{R}^n &\longrightarrow \mathbb{R}^{|S|} \\ x &\longmapsto y \ : \ \forall i, \ y_i = \sum_{j \in S_i} x_j \end{aligned} \tag{2.16}$$

This operator can be written as a matrix operation: $\sigma_S(x) = K_S x$ where:

$$(K_S)_{i,j} = \begin{cases} 1 & \text{if } j \in S_i \\ 0 & \text{else} \end{cases} \tag{2.17}$$

We can compare two networks G_0 and G_1 verifying $G_0 \overset{S}{\succ} G_1$ by looking at the vector Δ_{G_0,G_1} defined as:

$$\Delta_{G_0,G_1} = \Phi(G_1) - \sigma_S(\Phi(G_0)) \tag{2.18}$$

The $i-th$ entry of Δ_{G_0,G_1} represents how close the centrality of the nodes i in G_1 is to the sum of the centralities in the subset S_i in G_0. Δ_{G_0,G_1} will be called *eigenvector distance*[6] between the networks G_0 and G_1.

Figure 2.3 illustrates the computation of the projection and the comparison between G_0 and G_1.

The eigenvector distance defined by (2.18) quantifies the loss of information through the partitioning regarding the steady-state of the initial network. The network G_1 is aimed to abstract the dynamical behavior of G_0 but the dynamics inside each parts is lost and only the dynamics between the parts is taking into account. Therefore, the eigenvector distance measures how much the deletion of the internal dynamics impacts the estimation of the aggregated eigenvector centrality. Imposing Δ_{G_0,G_1} to be small implies that the weights of G_1 have to be computed in such a way that the dynamics of G_1 can mimic the dynamics of G_0 without taking into account the internal structure of the parts. We will see in the next section that it is actually always possible to compute the weights of the reduced network in order to cancel out the eigenvector distance. Said otherwise, for any partition of G_0, the weights of G_1 can be computed such that the eigenvector centrality of each node of G_1 corresponds to the sum of the eigenvector centralities in the corresponding subset of G_0.
The eigenvector centrality is particularly interesting to preserve for some applications. As an example, consider a network where some items (cars, people, goods,...) travel from one node to another following equation (**??**). By canceling the eigenvector distance, the distribution of the items at the steady-state is preserved: the number of the items in a part of the partition equals the number of items given by the abstracting network. Therefore, the abstracting network can be used to estimate and analyze the aggregated steady state of a large-scale network.

[6]Note that the term *distance* does not correspond used here to the mathematical acceptance of this term which is a scalar verifying a certain number of properties.

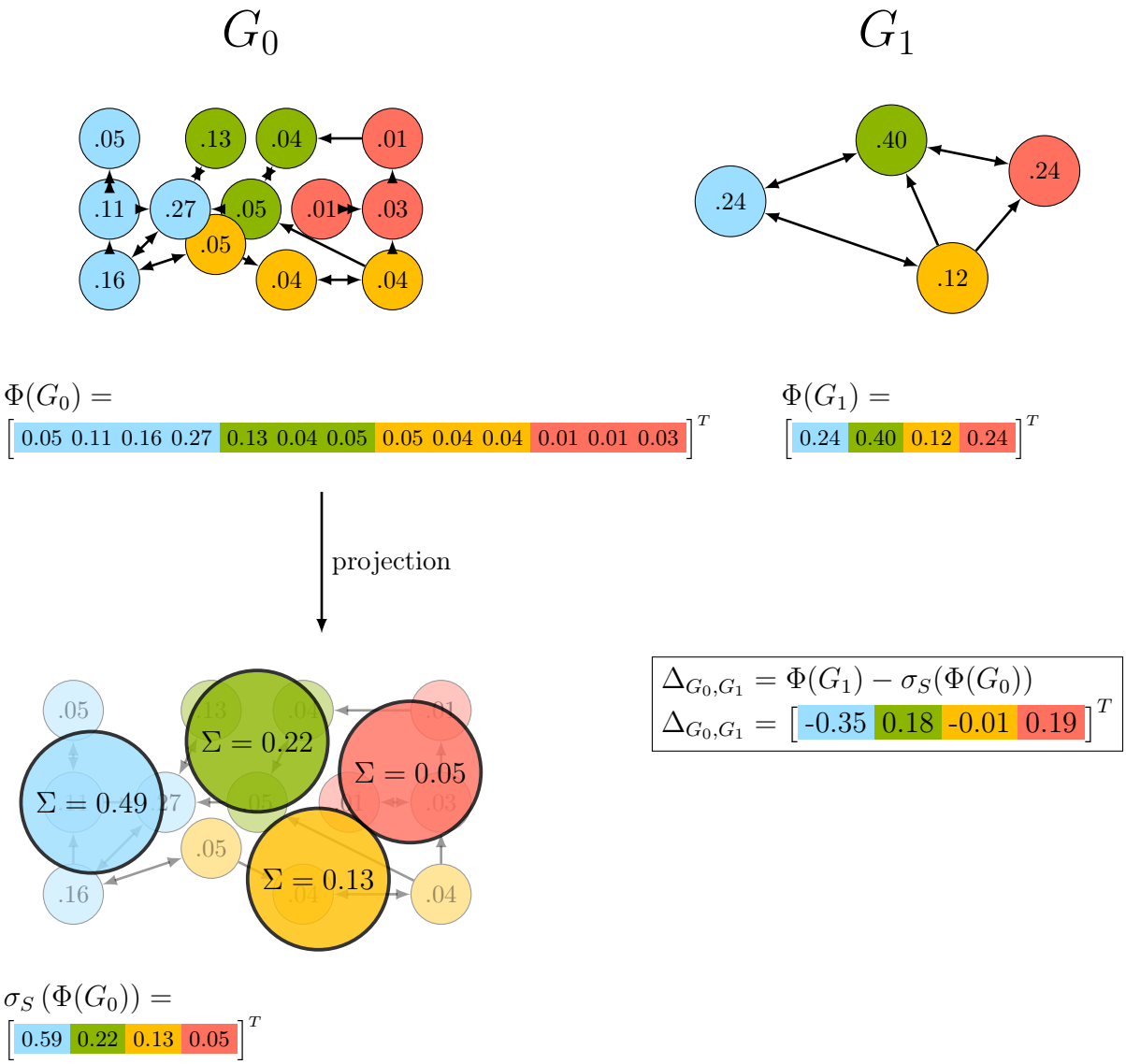


Figure 2.3: This figure illustrates the projection operator (2.16) and the computation of the eigenvector distance (2.18). To compare the dynamical behavior of G_0 and G_1 (with $G_0 \succ G_1$) we first compute the eigenvector centrality for each using (2.12). The computation of this centrality depends on the weights assigned on the edges, here we considered that all weights are equal to 1. As the two vectors are not of the same dimension we sum the component of $\Phi(G_0)$ within each part of the partition. In this way we obtain $\sigma_S(\Phi(G_0))$ which can be compared with $\Phi(G_1)$ leading to the vector Δ_{G_0,G_1}. Here we notice that the reconstruction of the yellow part is good, while the reconstruction of the other part are not.

Mass conservation

In the the problem we treat, we assume that the initial network is a flow network. As seen in the introduction, this means that for each nodes, the amount of weight going in equals the amount of weight going out. We aim to preserve this property through the reduction. The preservation of this property has a strong physical meaning because some networks as electrical networks, water supply networks or generally every network representing transportation are flow networks by their nature. For instance, in electrical network this property corresponds to the famous Kirchoff's circuit law. Thus, by preserving this property we ensure that the reduction method does not violate an intrinsic physical property of the system. We give here a definition of the set of networks having this property:

$$\Psi_{flow} = \left\{ G = (A, V, E),\ \forall k,\ \sum_i A_{ik} = \sum_j A_{kj} \right\} \tag{2.19}$$

We also call flow matrix an adjacency matrix of a flow network which is a matrix A such that $\forall k,\ \sum_i A_{ik} = \sum_j A_{kj}$.

Total mass

A third property of interest, when considering system implying a flow, is the total mass. The total mass is the sum of all the weights in the network. When these weights have a physical meaning (capacity, resistance,...) it may be useful to preserve their sum. The total mass of a network G will be denoted by $\|G\|_0$. Equivalently, the sum of all the values of its adjacency matrix A is denoted by $\|A\|_0$.

2.4.2 Specific problem formulation

In the previous section, we presented three notions allowing us to give specifications to Problem 2.1. Thus, we look for a reduced network that comes out of a partition of the initial network with the following three features: it minimizes the scale-free cost function (2.4), it is a flow network as defined in (2.19), it cancels out the eigenvector distance (2.18) and it preserves the total mass of the initial network. This leads to the following formulation:

Problem 2.3
Given a strongly connected initial network $G_0 \in \Psi_{flow}$ and a desired scale-free coefficient α, find a network G_1, solution of the following minimization problem.

$$\begin{aligned} \min_G \ \mathrm{J}_{\mathrm{SF}_\alpha}(G), \quad \textit{subject to} \quad & G_0 \succ G \\ & G \in \Psi_{flow} \\ & \Delta_{G_0,G} = \boldsymbol{0} \\ & \|G_0\|_0 = \|G\|_0 \end{aligned} \tag{2.20}$$

Where Ψ_{flow} is given in (2.19) and $\mathbf{0}$ is a vector of zeros of the *corresponding* dimension. To compare with the general formulation (2.1), here the physical property to preserve is the mass conservation $\Psi = \Psi_{flow}$ and the similarity is decomposed in two components: $G_0 \equiv G \iff \Delta_{G_0,G} = \mathbf{0}$ and $\|G_0\|_0 = \|G\|_0$.

2.4.3 Result on the optimization problem

In this section, we present preliminary results on the constraints of Problem 2.3 leading to the design of the specific algorithm. In particular, we will see how the weights of the reduced network can be chosen such that: i) the eigenvector distance is null, ii) the reduced network remains a flow network and iii) the sum of all weights in the network is preserved.
These results refers to the *hard* constraint of the problem.

Canceling eigenvector distance

Here, we are going to see that, with a certain choice of the weights of the reduced network, we can ensure a perfect consistency, in terms of eigenvector centrality, between the two networks.

Theorem 2.1
Let $G_0 = (A_0, V_0, E_0) \in \Gamma_n$. Let G_1 be the network coming out of the merging $\mathcal{S}_{v,w}$ for any edges $(v, w) \in E_0$. There is a choice of the weights of G_1 such that the eigenvector distance between G_0 and G_1 is null, which is $\Delta_{G_0,G_1} = 0$.
To do so, it is sufficient to take P_1, the normalized adjacency matrix of G_1 as:

$$P_1 = F P_0 H^\top, \tag{2.21}$$

where $F, H \in \mathbb{R}^{n-1 \times n}$ are defined by:

$$F_{i,j} = \begin{cases} 1 & \text{if } i < n-1 \text{ and } S_i = \{j\} \\ \beta_v & \text{if } i = n-1 \text{ and } j = v \\ \beta_w & \text{if } i = n-1 \text{ and } j = w \\ 0 & \text{else} \end{cases} \tag{2.22}$$

$$H_{i,j} = \begin{cases} 1 & \text{if } j \in S_i \\ 0 & \text{else} \end{cases} \tag{2.23}$$

and $\beta_v = \frac{x_0^\star(v)}{x_0^\star(v)+x_0^\star(w)}$, $\beta_w = \frac{x_0^\star(w)}{x_0^\star(v)+x_0^\star(w)}$ where $x_0^\star$ is the eigenvector centrality of G_0.

Proof of Theorem 2.1 In order to prove Theorem 2.1 we have to show three points which are: the matrix P_1 defined in (2.21)

i) has a structure compatible with a network coming out of the merging $\mathcal{S}_{v,w}$ of the network G_0

ii) is normalized

iii) has an eigenvector centrality equal to the projection of the eigenvector centrality of P_0

This three points will be proved one after the other:

i) By definition of a network coming out of a partition (see (7)), there is an edge $i \longrightarrow j$ in the reduced network if and only if there exists an edge $l \longrightarrow k$ in G_0 such that $l \in S_i$ and $k \in S_j$. We want to show that the position of the non-zeros values in P_1 respects this. By definition of P_1 (2.21), we have:

$$P_{1i,j} = \sum_{k=1}^{n} \sum_{l=1}^{n} F_{i,l} P_{0l,k} H_{j,k}$$

and, from the definition (2.23), we know that $F_{i,l} \neq 0 \Leftrightarrow l \in S_i$ and that $H_{j,k} \neq 0 \Leftrightarrow k \in S_j$. Thus:

$$P_{1i,j} = \sum_{k \in S_j} \sum_{l \in S_i} F_{i,l} P_{0l,k} H_{j,k}$$

It comes out that:

$$P_{1i,j} \neq 0 \Longleftrightarrow \exists (l,k) \in S_i \times S_j \quad \text{s.t.} \quad P_{0l,k} \neq 0 \tag{2.24}$$

Therefore, P_1 is indeed the adjacency matrix of a network coming out the merging $\mathcal{S}_{v,w}$ of the network G_0.

ii) We want to show that the matrix P_1 is normalized in the sense that $\forall i \in [1, \ldots, n-1]$ we have $\sum_j P_{1i,j} = 1$. By definition of P_1 (2.21), we have:

$$\sum_{j=1}^{n-1} P_{1i,j} = \sum_{j=1}^{n-1} \sum_{k \in S_j} \sum_{l \in S_i} F_{i,l} P_{0l,k} H_{j,k}$$

We decompose $\sum_{j=1}^{n-1} P_{1i,j}$ in $\sum_{j=1}^{n-2} P_{1i,j} + P_{1i,n-1}$, and so

$$\sum_{j=1}^{n-1} P_{1i,j} = \sum_{k \in S_{n-1}} \sum_{l \in S_i} F_{i,l} P_{0l,k} H_{n-1,k} + \sum_{j=1}^{n-2} \sum_{k \in S_j} \sum_{l \in S_i} F_{i,l} P_{0l,k} H_{j,k}$$

We recall that a merging $\mathcal{S}_{v,w}$ has the following form:

$$\mathcal{S}_{v,w} = \{\underbrace{\{1\}}_{S_1}, \ldots, \underbrace{\{v-1\}}_{S_{v-1}}, \underbrace{\{v+1\}}_{S_v}, \ldots, \underbrace{\{w-1\}}_{S_{w-2}}, \underbrace{\{w+1\}}_{S_{w-1}}, \ldots, \underbrace{\{n\}}_{S_{n-2}}, \underbrace{\{v,w\}}_{S_{n-1}}\} \tag{2.25}$$

We note that for all $j < n-1$, there exists a unique $k \in S_j$ and so, with the definition of H (2.23), a unique k such that $H_{j,k} = 1$. Denoting it by k_j, we have:

$$\sum_{j=1}^{n-1} P_{1i,j} = \sum_{k\in\{v,w\}} \sum_{l\in S_i} F_{i,l} P_{0l,k} + \sum_{j=1}^{n-2} \sum_{l\in S_i} F_{i,l} P_{0l,k_j}$$

Moreover, when j covers the set $\{1, ..., n-2\}$ then k_j covers the set $\{1, .., n\}\backslash\{v, w\}$ and then:

$$\sum_{j=1}^{n-1} P_{1i,j} = \sum_{k\in\{v,w\}} \sum_{l\in S_i} F_{i,l} P_{0l,k} + \sum_{k\in\{1,..,n\}\backslash\{v,w\}} \sum_{l\in S_i} F_{i,l} P_{0l,k}$$

Finally, merging the two parts of the sum, we have:

$$\begin{aligned}\sum_{j=1}^{n-1} P_{1i,j} &= \sum_{k=1}^{n} \sum_{l\in S_i} F_{i,l} P_{0l,k} \\ &= \sum_{l\in S_i} F_{i,l} \sum_{k=1}^{n} P_{0l,k}\end{aligned}$$

Yet, P_0 is normalized, so $\sum_{k=1}^{n} P_{0l,k} = 1$. It follows:

$$\sum_{j=1}^{n-1} P_{1i,j} = \sum_{l\in S_i} F_{i,l}$$

Now, either $i < n-1$ and, with the definition of F (2.23) we have directly $\sum_{j=1}^{n-1} P_{1i,j} = 1$, or $i = n-1$ and then

$$\begin{aligned}&\sum_{j=1}^{n-1} P_{1n-1,j} = \sum_{l\in S_{n-1}} F_{n-1,l} \\ &\sum_{j=1}^{n-1} P_{1n-1,j} = \beta_v + \beta_w = \frac{x_0^\star(v)}{x_0^\star(v) + x_0^\star(w)} + \frac{x_0^\star(w)}{x_0^\star(v) + x_0^\star(w)} = 1\end{aligned}$$

Therefore, based on the proof of i) and i)), P_1 is the normalized adjacency matrix of a reduced network G_1 coming out of the merging $\mathcal{S}_{v,w}$ of G_0.

iii) Let $x_1^\star$ be the eigenvector centrality of G_1. We want to show that $\sigma(x_0^\star) = x_1^\star$. It is sufficient to show that $\sigma(x_0^\star)$ fulfill the definition of the eigenvector centrality of G_1 which are $\sigma(x_0^\star) = P_1^\top \sigma(x_0^\star)$ and the sum of $\sigma(x_0^\star)$ is 1. By uniqueness of the eigenvector centrality, this will proof iii). Before that, we need some preliminary results:
At first, we remark that the definition of H is the same as the definition of K_S in (2.17) so we have $\sigma(x) = Hx$ for all x. Moreover, the matrix product $H^\top F$ has the following form:

$$H^\top F = \begin{pmatrix} 1 & & & & & & & & & \\ & \ddots & & & & & & & & \\ & & 1 & & & & & & & \\ & & & \beta_v & & & \beta_w & & & \\ & & & & 1 & & & & & \\ & & & & & \ddots & & & & \\ & & & & & & 1 & & & \\ & & & \beta_v & & & \beta_w & & & \\ & & & & & & & 1 & & \\ & & & & & & & & \ddots & \\ & & & & & & & & & 1 \end{pmatrix} \tag{2.26}$$

which can also be written as follows:

$$H^\top F = I_{n-1} + (\beta_v - 1)e^{n-1}_{v,v} + \beta_v e^{n-1}_{w,v} + (\beta_w - 1)e^{n-1}_{w,w} + \beta_w e^{n-1}_{v,w}$$

where $e^n_{i,j}$ is the square matrix of size n whose only non-zero entry is a 1 at the (i,j)th position, and I_n is the identity matrix of size n. It follows that

$$\begin{aligned}
\forall i \notin \{v,w\}, \quad & ({x^\star_0}^\top H^\top F)_i = x^\star_0(i) \\
\text{and if } i = v, \quad & ({x^\star_0}^\top H^\top F)_v = x^\star_0(v)\beta_v + x^\star_0(w)\beta_v \\
& ({x^\star_0}^\top H^\top F)_v = (x^\star_0(v) + x^\star_0(w))\frac{x^\star_0(v)}{x^\star_0(v) + x^\star_0(v)} = x^\star_0(v) \\
\text{and if } i = w, \quad & ({x^\star_0}^\top H^\top F)_w = x^\star_0(v)\beta_w + x^\star_0(w)\beta_w \\
& ({x^\star_0}^\top H^\top F)_w = (x^\star_0(v) + x^\star_0(w))\frac{x^\star_0(w)}{x^\star_0(v) + x^\star_0(v)} = x^\star_0(w)
\end{aligned}$$

Hence ${x^\star_0}^\top = {x^\star_0}^\top H^\top F$, and then:

$${x^\star_0}^\top P_0 H^\top = {x^\star_0}^\top H^\top F P_0 H^\top$$

and by definition ${x^\star_0}^\top P_0 = {x^\star_0}^\top$ and $F P_0 H^\top = P_1$

$$\begin{aligned}
{x^\star_0}^\top H^\top &= {x^\star_0}^\top H^\top P_1 \\
H x^\star_0 &= P_1^\top H x^\star_0 \\
\sigma(x^\star_0) &= P_1^\top \sigma(x^\star_0)
\end{aligned}$$

Moreover, it is clear that $\sum \sigma(x^\star_0) = \sum x^\star_0 = 1$. Thus, $\sigma(x^\star_0)$ fulfill the definition of the eigenvector centrality of G_1. And by uniqueness of the eigenvector centrality $\sigma(x^\star_0) = x^\star_1$.

□

This result concerns only merging, which is a particular partition, but it can be extended to any partition. For simplicity, and since it is sufficient for the following, we have only shown the case of the merging. Note that the preservation of the eigenvector centrality influences only the normalized adjacency matrix of G_1. Therefore, it remains many degrees of freedom to choose the adjacency matrix and so the weights of G_1.

Preservation of the mass conservation

The mass conservation property defined in (2.19) is another physical property that we aim to preserve through the reduction. We see here that if the initial network is a flow network we can ensure that the reduced network is a flow network too.

Theorem 2.2
Let P be a normalized matrix as defined in (2.13) *then it exists a diagonal matrix X such that XP is a flow matrix.*
To do so, it is sufficient to take X as:

$$X = \kappa \,\mathrm{Diag}(x^\star) \tag{2.27}$$

= where $\kappa \neq 0$ and $x^\star$ is the eigenvector centrality associated with P.

Proof of Theorem 2.2

Let $x^\star$ be the eigenvector centrality of P, which is $x^\star P = x^\star$ and $X = \kappa\,\mathrm{Diag}(x^\star)$ for any $\kappa \neq 0$. We have that $\kappa x^\star P = \kappa x^\star$ and consequently $\mathbf{1}^\top XP = \mathbf{1}^\top X = (X\mathbf{1})^\top = (XP\mathbf{1})^\top$. Thus, the vector whose entries are the sum of the column of XP is equal to the vector which entries are the sum of the row of XP. Hence, XP is a flow matrix. □

As the preservation of the mass conservation does not influence the normalized adjacency matrix it is compatible with the canceling of eigenvector distance. Moreover, with κ we still have one degree of freedom to choose the weights of G_1.

Preservation of total mass

From Theorems 2.1 and 2.2 we have the following corollary:

Corollary 2.1
Let G_0 be a network and P_0 its normalized adjacency matrix. Let S be any merging and F and H the merging matrices associated to S. Consider the network G_1 defined by the matrix A_1 as:

$$A_1 = \kappa \,\mathrm{Diag}(x_1^\star) F P_0 H^\top, \tag{2.28}$$

where $x_1^\star$ is the eigenvector centrality of $P_1 := FP_0H^\top$ and $\kappa \neq 0$. Then G_1 is a flow network and $\Delta_{G_0,G_1} = 0$.

Let us note that the network G_1 is not unique as its adjacency matrix is defined up to a multiplicative constant κ. In the following we fix κ such that the sum of all weights in G_1 is equal to the sum of all weights in G_0, which is:

$$\kappa = \frac{|A_0|_0}{|\operatorname{Diag}(x_1^\star) F P_0 H^\top|_0} \tag{2.29}$$

where $|\cdot|_0$ is defined as: $|A|_0 = \sum_{i,j} A_{i,j}$ for all matrices A.
In this way, the reduced network G_1 is uniquely defined and verifies the constraints of Problem 2.3.
Thus given a network G_0 and a merging $\mathcal{S}$, we denote $G_0^{\mathcal{S}}$, the network coming out the partition of S and fulfilling all the constraints of Problem 2.3. The adjacency matrix of this graph is given by:

$$A_0^{\mathcal{S}} = \frac{|A_0|_0}{|\operatorname{Diag}(x_1^\star) F P_0 H^\top|_0} \operatorname{Diag}(x_1^\star) F P_0 H^\top \tag{2.30}$$

Reformulation of the optimization problem

The results of this section has shown that for any partition we can choose the weights of the resulting network such that the constraints of Problem (2.3) are respected. Therefore, finding the partition minimizing the scale-free cost function is enough to solve Problem (2.3). Thus, we formulate this new problem:

Problem 2.4
Given $G_0 = (A, V, E) \in \Psi$, find the partition $\mathcal{S}$ minimizing the scale-free cost function:

$$\min_{\mathcal{S}} \quad \mathrm{J}_{\mathrm{SF}_\alpha}(G_0^{\mathcal{S}}) \tag{2.31}$$

2.4.4 Specific algorithm

Based on the result of the Section 2.4.3, a description of the algorithm is presented in Algorithm 3. Therein,, the network G_k is represented by (A_k, E_k, V_k). First, we compute the eigenvector centrality $x_0^\star$ of the initial network (line 1), the normalized adjacency matrix P_0 (line 2) and the coefficient κ (line 3). A random subset Ω of n_{rand} edges is drawn (line 6). For each edge S, we compute the reduced network G^S coming from the merging $\mathcal{S}$ in G_k (line 8) using κ, $x_k^\star$ and P_k. The scale-free cost function of G^S is computed (line 9). Finally, the best edge to merge is chosen (line 11) and the new network is computed (line 12). After recomputing $x_k^\star$ as a projection of the previous value (line 13) and P_k (line 14), the loop restarts with the new network.

Algorithm 3 Merge to scale-free with property preservation

Input: G_0, α_{SF}, n_{rand}
Output: G_{end}

$x_0^\star = \Phi(G_0)$
P_0 from (2.13)
κ from (2.29)
$k = 0$
while $\neg stop$ **do**
  $\Omega \longleftarrow n_{\text{rand}}$ random elements of E_k
  for $(v, w) \in \Omega$ **do**
    Compute $G_k^{\mathcal{S}_{v,w}}$ from (2.30)
    $n_{SF}(\mathcal{S}_{v,w}) = \mathrm{J}_{\mathrm{SF}_\alpha}(G_k^{\mathcal{S}_{v,w}})$
  end for
  $\mathcal{S}_{best} = \text{argmin} n_{SF}(\mathcal{S}_{v,w})$
  $G_{k+1} = G_k^{\mathcal{S}_{best}}$ from (2.30)
  $x_{k+1}^\star = \sigma_{S_{best}}(x_k^\star)$
  P_{k+1} from (2.13)
  $k{+}{+}$
end while

2.5 Some results about the algorithm

2.5.1 Algorithm complexity

In this section, we discuss the complexity of the algorithm in terms of number of operations. Several definitions and notations are used to quantify an order of magnitude. Here, we use the notation $\Theta(\,\cdot\,)$ defined as follows:

$$f(x) = \Theta(g(x)) \iff \exists\, c_1,\, c_2 > 0,\; c_1 g(x) < f(x) < c_2 g(x) \tag{2.32}$$

Proposition 2.1
Consider Algorithm 3 on an initial network G_0 with N nodes, and with n_{rand} the size of the random subset. The algorithm can be divided in two phases:

- *An initial phase in which, in particular, the eigenvector centrality is computed. The complexity of this phase is $\Theta(N^3)$ in the worst case, but can be lowered to $\Theta(N^2)$ in some cases.*
- *The reduction phase which consists in at most N iterations having each a complexity of $\Theta(N n_{rand})$*

Overall the worst-case complexity is $\Theta(N^3)$.

Proof of Proposition **??** For the first phase, the computation of an eigenvector needs $\Theta(N^3)$ operations. However in [117], it is shown that if the maximum degree of the network is bounded, the computation can be done in $\Theta(N^2)$.
For the second phase, we know that at each step of the algorithm, the number of nodes decreases by 1 and the total number of iterations can not be larger than N. Then, we consider that the number of steps is $\Theta(N)$. Let C be the total number of operations in each iteration, we have

$$C = C_{\text{rand}} + n_{\text{rand}} \left(C_{\text{Merge}} + C_{\text{J}_{\text{SF}_\alpha}} \right) + C_{\text{argmin}} + C_\sigma$$

The different functions can be detailed as follows:

- rand consists in picking n_{rand} values, so $C_{\text{rand}} = \Theta(n_{\text{rand}})$.
- Merge is the computation of the new network (2.30), which can be computed as a combination of the columns and rows of the previous adjacency matrix, so $C_{\text{Merge}} = \Theta(N)$.
- $\text{J}_{\text{SF}_\alpha}$ can be decomposed as follow:
  - Compute the degree distribution of the adjacency matrix. The number of operation needed is the number of different degrees in the network. Thus, the number of operations is smaller than N.
  - Computation of the scale-free cost function: it is the norm of a difference of two vectors whose sizes are always smaller than N. Thus, the number of operations is also smaller than N.

  Finally $C_{\text{J}_{\text{SF}_\alpha}} = \Theta(N)$.
- argmin require $\Theta(n_{\text{rand}})$ operations.
- σ consists in the combination of the coordinates of $x_k^\star$. Hence $C_\sigma = \Theta(N)$

Finally we have:

$$\begin{aligned} C &= \Theta(n_{\text{rand}} + n_{\text{rand}}(N + N) + n_{\text{rand}} + N) \\ C &= \Theta(N n_{\text{rand}}) \end{aligned}$$

Thus, the complexity of the second phase is $\Theta(N^2 n_{rand})$. □

This complexity is polynomial with respect to the size of the initial network, whereas the naive way to find a partition of a network by testing all possibilities have an exponential complexity. Even by improving the partitioning algorithm, the complexity is lower-bounded by the complexity of the eigenvector centrality which can not be lowered. However, this algorithm is not supposed to be run in real time, but once to find a reduced network which can then be used as an abstracting model. Thus a relatively high complexity is not crippling for the application.

The complexity of the reduction phase is linear with respect to n_{rand}. Therefore, it is possible to tune this parameter to reduce the computation time. We will investigate in the next section the influence of this parameter on the performance of the algorithm.

2.5.2 Influence of the size of the random subset

In this section, some experiments are presented to emphasize the influence of n_{rand} in Algorithm 3. In particular, we focus on two points:

1. The speed of convergence towards a scale-free network: Through the iterations, the current network gets closer and closer to the scale-free structure. We will wonder to what extent testing more candidates at each iteration improves this convergence.

2. The reproducibility of the the algorithm: As the algorithm implies a random process (the generation of the subset of pair of nodes), even with similar initial conditions, two executions of the algorithm will produce different output. We will wonder if these different output still have similar properties. Beyond the similarity of the properties, we will also wonder if the reduced networks come from similar partitions and so if some areas of the network of the initial network tend to always cluster together.

For these experimentations we will consider a synthetic family of networks: the Manhattan-like grids introduced hereafter

The Manhattan-like grid Manhattan grids, or simply grids, is a family of directed network inspired by the topology of the urban network of Manhattan [112]. It consists simply of a grid of size $N \times N$ in which each of the N^2 intersections is a node. Therefore $G = (A, \mathcal{V}, \mathcal{E})$ is a $N \times N$ Manhattan grid if:

$$\forall\, i,j \in [1,\dots,N^2],\ A_{i,j} = A_{j,i} = \begin{cases} 1 & \text{if } i-j = 1 \text{ and } i \not\equiv 1 \mod n \\ 1 & \text{if } i-j = N \\ 0 & \text{else} \end{cases} \tag{2.33}$$

In our case, in order to get heterogeneous networks we add some random irregularities:

- Nodes are removed with probability β:

$$\forall\, v \in \mathcal{V}, \quad \mathcal{V} = \mathcal{V}\backslash\{v\} \text{ with probability } \beta \tag{2.34}$$

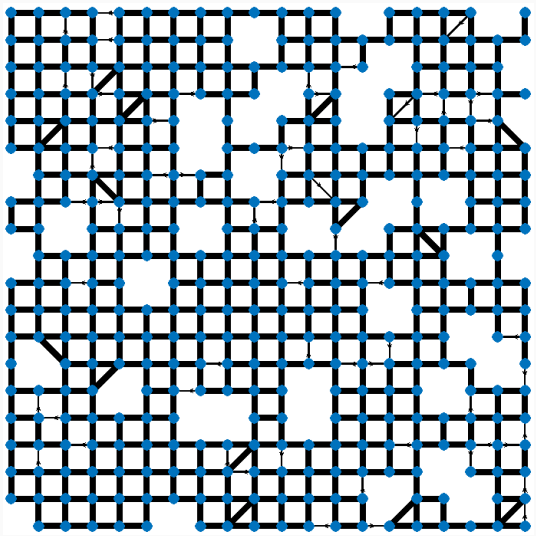
- Diagonal shortcuts are created with probability γ

$$\begin{aligned} \forall\, i,j \in [1,\dots,N^2], \text{ s.t. } & i-j = N+1 \text{ and } i \not\equiv 1 \mod n \\ \text{or } & i-j = N-1 \text{ and } i \not\equiv 0 \mod n \\ \mathcal{E} = \mathcal{E} \cup \{(i,j)\} & \text{ with probability } \gamma \end{aligned} \tag{2.35}$$

- Edges are restricted to a unique direction with probability δ

$$\begin{aligned} \forall\, (i,j) \in \mathcal{E}, \quad & \mathcal{E} = \mathcal{E}\backslash\{(i,j)\} \text{ with probability } \delta/2 \\ \text{and } & \mathcal{E} = \mathcal{E}\backslash\{(j,i)\} \text{ with probability } \delta/2 \end{aligned} \tag{2.36}$$

Such a network is a Manhattan grid with perturbations, hence the name Manhattan-like grid. The weights on the edges are generated randomly while ensuring that the grid is a flow network (see Appendix B). The advantages of this type of network are multiple: i) the degree distribution is far from a scale-free distribution, so it can shows the ability of the algorithm to steer the network towards a scale-free structure, ii) it is easy to build this type of network, even with an arbitrarily large size, making the results presented easily reproducible and iii) it is a good representation of some physical networks [116] such as urban traffic networks or brain networks [121]. Hereafter an example from this family of network where thick lines represent double-way edges.



On the convergence of the algorithm

In fig. 2.4 we observe the evolution of the scale-free cost function through the iterations of the algorithm for different values of n_{rand}. When $n_{\text{rand}} = 1$, the edge merged is randomly chosen. That is why it is labeled 'Random edge' in the legend. Let first remark that for every value of n_{rand} the error decreases initially, even in the case $n_{\text{rand}} = 1$, where the edge to merge is randomly selected. This shows that networks (at least this type of grids) naturally tends towards a scale-free structure when edges are iteratively merged. The figure shows that the benefit from increasing n_{rand} is logarithmic. Thus, increasing n_{rand} does not increase substantially the performance while it increases linearly the computation time as seen in Proposition 2.1. In this case, $n_{\text{rand}} = 10$ produces network almost as scale-free as in the exhaustive case while being around 100 times faster (in the exhaustive case, there is about 1000 edges to test).

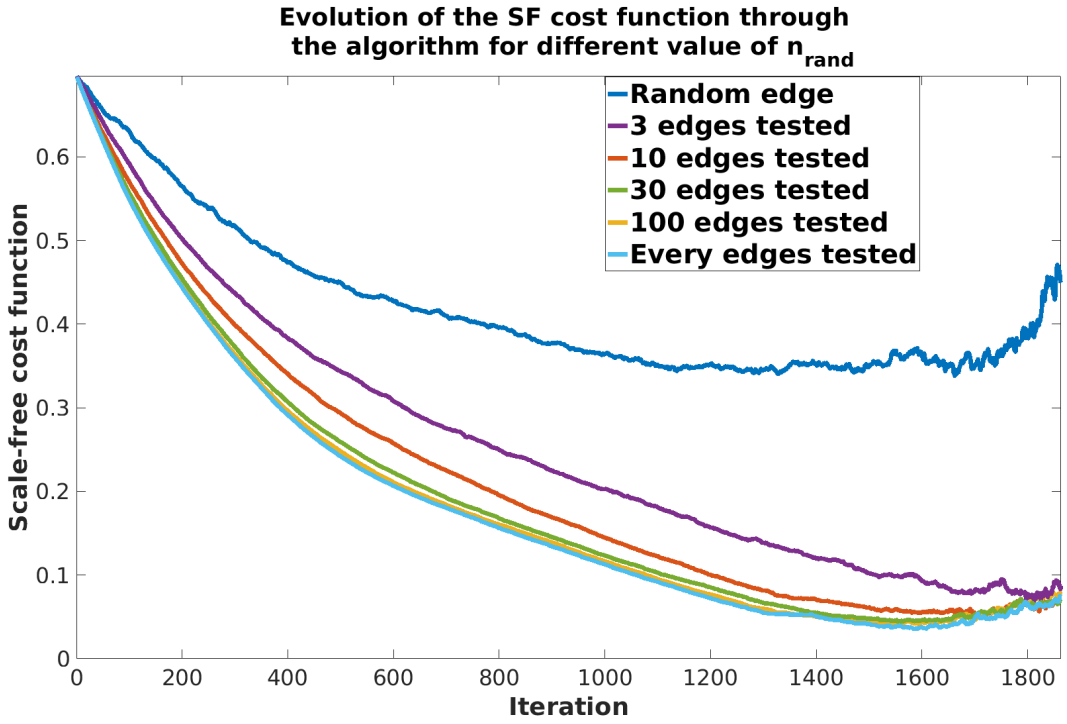


Figure 2.4: Comparison of the evolution of the scale-free cost function through the algorithm for different values of n_{rand}. The initial network is a Manhattan-like grid of size 45×45 and with 1962 nodes. The simulation is stopped when the size of the current network is equal to 5% of the size of the initial network.

On the reproducibility of the algorithm

Now we examine, how close the networks obtained via several executions of the algorithm are and how the value of n_{rand} influences this consistency. To answer this question, we have

executed Algorithm 3 several times with the same initial network and the same parameters. Then, we compare topological properties of the networks obtained. Precisely, we consider an initial Manhattan-like grid 25×25 and we run the algorithm until there is no more merging that increases the scale-freeness. We have executed the algorithm 50 times with $n_{\text{rand}} = 3$, 50 times with $n_{\text{rand}} = 30$ and once with $n_{\text{rand}} = +\infty$ which is the exhaustive case (as the output is deterministic one execution is enough). Then, we compute four properties for each output network: number of edges, number of nodes, radius (minimum eccentricity of any node), and scale-free cost function. Let us note that, to have a fair comparison, the radius is divided by the number of nodes in the network. In fig. 2.5 the results are presented in the form of histograms.

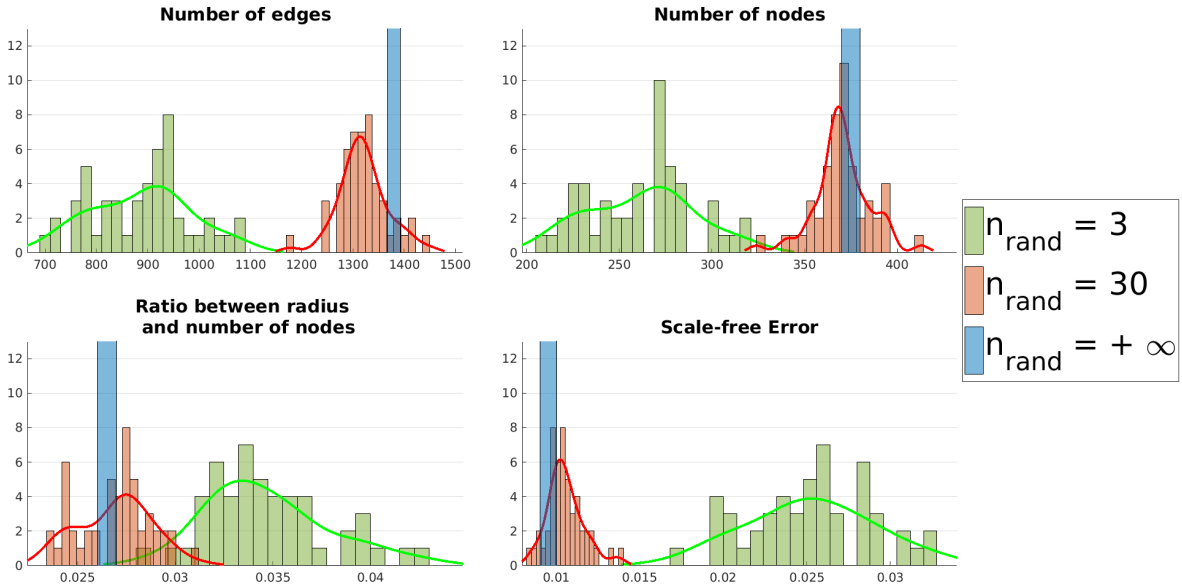


Figure 2.5: Histograms of the properties of the different networks obtained with different instances of the algorithm and with the same initial network. For each property, the x-axis represents the values and the y-axis, the frequency of apparition of each value. The green bars shows the values obtained when $n_{\text{rand}} = 3$, the red bars when $n_{\text{rand}} = 30$ and the blue bar when $n_{\text{rand}} = +\infty$. The superimposed lines fits the histograms.

We remark that the values are arranged around an expected value in an almost bell-shaped distribution. When n_{rand} is higher, this Gaussian behavior is more marked with a lower mean deviation is lower and an expected value closer to the case $n_{\text{rand}} = +\infty$ (which can be considered as a reference value). Therefore, as expected, a large value of n_{rand} reduces randomness.

In addition to the question of the consistency of the properties, we wonder if the partitions obtained from different execution of the algorithm are close. For this purpose, we compare the partitions obtained when $n_{\text{rand}} = 3$ and $n_{\text{rand}} = 30$ with the partition of reference obtained with $n_{\text{rand}} = +\infty$. A common way to compare two partitions is to use the normalized mutual information [32, 42]. The definition is quickly recalled hereafter. It is rather technical, thus the reader can pursue the reading just knowing that the normalized mutual information gives

a value between 0 and 1 on the similarity between two partitions.

The normalized mutual information Let us consider two partitions $\mathcal{S} = \{S_1, ..., S_m\}$ and $\mathcal{T} = \{T_1, ..., T_p\}$ over a network[a] of size n. The partial distributions of the part S_i is $P(S_i) = \frac{1}{n}|S_i|$. It measures the proportion of nodes which are in the part S_i. The joint distribution of S_i and T_j is $P(S_i, T_j) = \frac{1}{n}|S_i \cap T_j|$. It measures the proportion of nodes which are both in S_i and T_j. The entropy of the partition $\mathcal{S}$ is defined as $H(\mathcal{S}) = -\sum_i P(S_i)\log(P(S_i))$ and translates how much the set $\{1, .., n\}$ is fragmented in $\mathcal{S}$:
if $\mathcal{S} = \{\{1, ..., n\}\}$, then $H(\mathcal{S}) = -P(\{1, ..., n\})\log(P(\{1, ..., n\})) = 0 = -1 \times log(1)$.
On contrary if $\mathcal{S} = \{\{1\}, ..., \{n\}\}$, then the entropy is maximal.
The joint entropy is defined as $H(\mathcal{S}, \mathcal{T}) = -\sum_{i,j} P(S_i, T_j)\log(P(S_i, T_j))$ and translates how much the set $\{1, .., n\}$ is fragmented in the *intersection partition* $\{S_1 \cap T_1, S_2 \cap T_1, ..., S_m \cap T_p\}$.
Finally, the mutual information is defined as $\tilde{I}(\mathcal{S}, \mathcal{T}) = H(\mathcal{S}) + H(\mathcal{T}) - H(\mathcal{S}, \mathcal{T})$. In a word, it translates how much the *intersection* of $\mathcal{S}$ and $\mathcal{T}$ is more fragmented than $\mathcal{S}$ and $\mathcal{T}$ are. To calculate the normalized mutual information, we add a normalization factor: $I(\mathcal{S}, \mathcal{T}) = \frac{2\tilde{I}(\mathcal{S},\mathcal{T})}{H(\mathcal{S})+H(\mathcal{T})}$.
Thanks to this tool coming from information theory, we are able to attribute a value $I(\mathcal{S}, \mathcal{T}) \in [0; 1]$ to measure the similarity between two partitions $\mathcal{S}$ and $\mathcal{T}$. In particular, if $\mathcal{S} = \mathcal{T}$ then $I(\mathcal{S}, \mathcal{T}) = 1$.

[a] Here, we consider partitions over a network, but the definition remains the same for partitions over any set.

We consider a Manhattan-like grid of size 35×35. To test the consistency of the partitions, we compare each of the 50 partitions obtained with $n_{\text{rand}} = 3$ with the reference partition (obtained when $n_{\text{rand}} = +\infty$). We also make the comparison between each of the 50 partitions obtained with $n_{\text{rand}} = 30$ and the reference partition. Figure 2.6 shows the obtained results. Once again, we observe a bell-shaped distribution of the values. We observe that the mean value is closer to 1 when n_{rand} is larger. It means that, as expected, when the random effect is reduced, the partitions obtained are closer to the reference partition.
In conclusion, in Sections 2.5.1 and 2.5.2 we have seen that when n_{rand} increases: the run time increases linearly, the scale-free error decreases logarithmically and the variability of the outputs decreases. However, the consistency of the output is not necessarily a property of interest. According to the numerical results, it appears that a relatively small value of the number of edges tested at each step ($n_{\text{rand}} \approx 20$) is a good balance.

2.5.3 Modification of topological properties

The main effect of the algorithm is to modify the degree distribution of an arbitrary network. However, one can wonder how the other structural properties of the initial network are modified. We would want to compare the properties of the initial and final networks, but the direct comparison could be biased by the difference of their sizes. Therefore, to compare the properties fairly, we generate a small-scale version of the initial network. This network is build

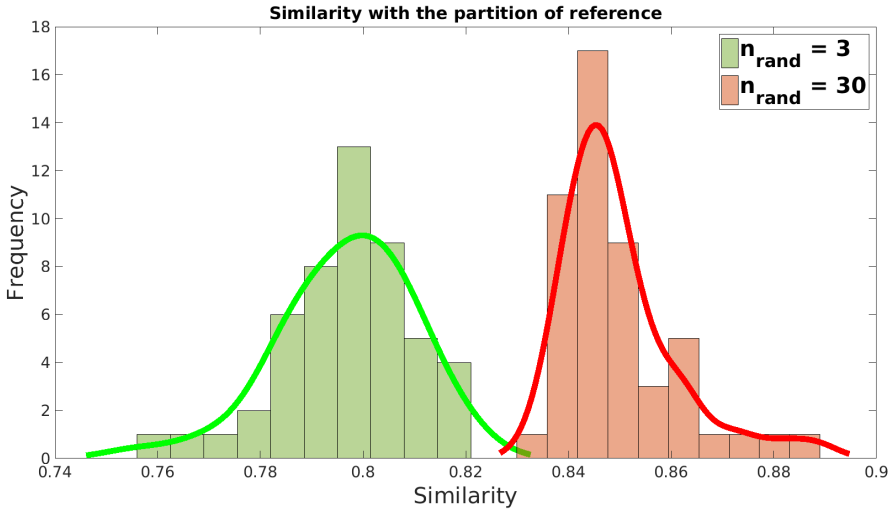


Figure 2.6: Histograms of the normalized mutual information between the partition of reference and the 50 partitions obtained with $n_{\text{rand}} = 3$ (in blue) and the 50 partitions obtained with $n_{\text{rand}} = 30$ (in red).

the same way as the initial network but with a size similar to the final network.
In Table 2.1 we show the structural properties of the initial network, of the small-scale version of the initial network named *reference network* and the averaged values for 50 networks resulting of the algorithm. The structural properties which are compared are: Number of nodes, number of edges, radius[7], diameter [7], number of hub (we defined a hub as a node connected with at least 5% of all nodes), clustering coefficient [7], maximum indegree and mean indegree.

Table 2.1: Modification of the properties of the network through the reduction

	Initial network	Reference network	Reduced networks	Evolution
Num. of nodes	1165	508	516.42	$\simeq$
Num. of edges	4426	1911	2236.80	$\simeq$
Radius	29	20	9.16	$\searrow$
Diameter	57	37	17.22	$\searrow$
Num. of hub	0	0	35.14	$\nearrow$
Clust. coeff.	0.1	0.1	0.43	$\nearrow$
Max indegree	6	6	23.3	$\nearrow$
Mean indegree	3.80	4.33	3.76	$\simeq$

The comparison is rather fair since the reduced networks and the reference network have a similar size (number of nodes and edges) and since the initial network and the reference network are consistent (no hubs, same clustering coefficient, same max and mean degree).
One can remark that the characteristic distances (radius and diameter) of the reduced network are significantly lower than the reference network, while the clustering coefficient is significantly higher, which are two features commonly observed in scale-free models. The

[7] These properties are defined in Section 0.2.2

other topological modifications, presence of hubs and higher maximum indegree, is directly explained by the power-law degree distribution of scale-free networks.

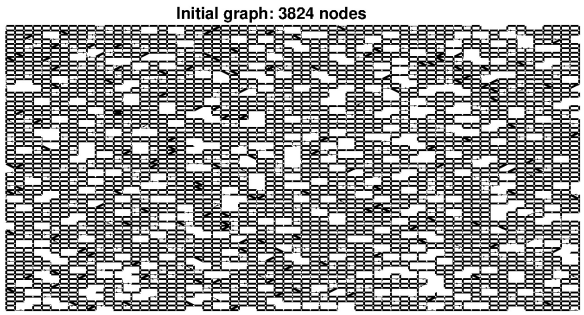
Simulation 2.1. In this section, Algorithm 3 is applied to an academic case. We consider an initial large-scale Manhattan-like grid and we apply the reduction algorithm to it. The tuning of the different parameters is presented in Table 2.2 and the output of the simulation is presented in fig. 2.7.

Table 2.2: Parameters of the simulation on the Manhattan-like grid

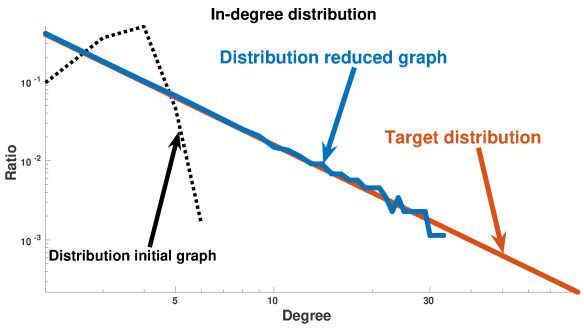
Size	$\|G\|$	α_{SF}	n_{rand}	Degree
65×65	3824	2	10	in

2.6 Conclusion

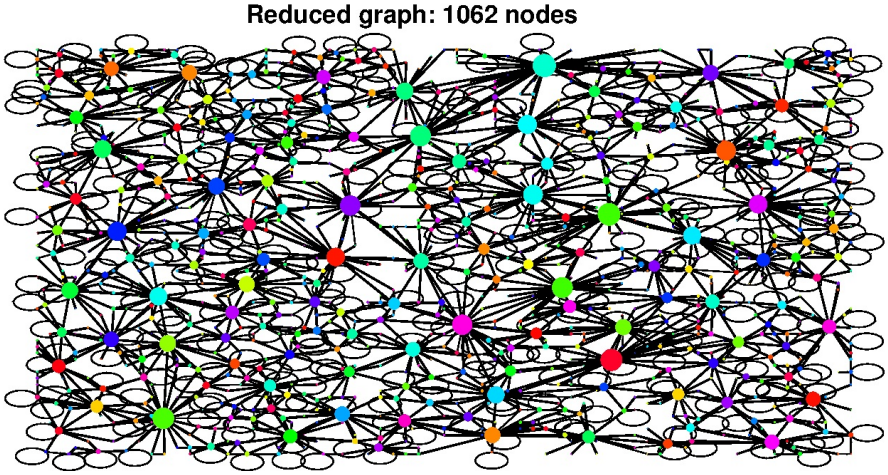
In this chapter, we have introduced a class of network partitioning problem inducing a scale-free distribution. These problems are formulated as optimization problems: we want to optimize the scale-freeness of the graph under constraints on similarity and properties preservation. In particular, while treating a particular problem, we have shown that it is possible to optimize the scale-freeness under three constraints: the eigenvector centrality is preserved up to a projection, the total mass remains the same and the mass conservation property is preserved. Then, we define a partition algorithm which takes advantage of these results and allows to find a sub-optimal solution. Experimental results brought some insights on the functioning of the algorithm and strong clues on the choice of a free-parameter of the algorithm. In particular, we show that we can speed up the execution with almost the same efficiency.
The multi-level presentation of the problem and the algorithm allows to easily generalize our results to other contexts. Moreover, the algorithm could be used to drive a network towards any desired structure and still verify the similarity constraints. Chapter 4 presents an implementation of this algorithm on a large-scale traffic network taking advantage of the preservation of the properties. In the other, Section 5.2 presents an application in epidemiology taking adavantage of the scale-freeness of the reduced network.



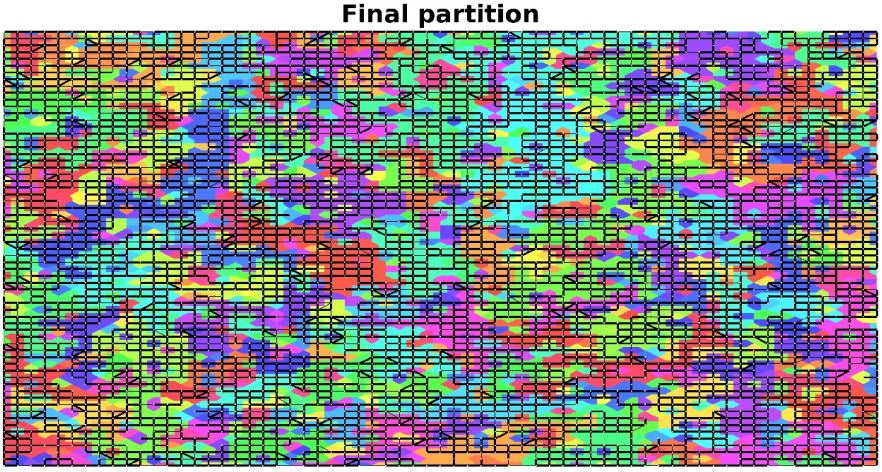
(a) The initial network. Double-way links are thicker than one-way ones.



(b) Indegree distribution of the initial network, of the reduced network and the target distribution in log-log representation.



(c) The reduced network issued from the partition. The size of the nodes represents their degree and the color matches with the color of the partition in (d).



(d) Final partition of the initial network. The color matches with the colors of the nodes in the reduced network (c).

Figure 2.7: Result of the Manhattan-like grid simulation. (a) the initial network, (b) the indegree distributions, (c) the final network and (d) the final partition. Sub-figure (b) emphasizes well the efficiency of the network to drive the degree distribution very close to the target distribution.

Chapter 3

Network partitioning algorithm towards average detectability

Contents

One of the main concern of the Scale-Freeback project is to develop tools for control and observation of large-scale network. In particular, concerning observability, one question echoed our investigation of network partitioning problems: how to partition a network between measured and unmeasured nodes in order to ensure a good reconstruction of the network state ? Precisely, the recent work of Niazi et al. [96] proposed the novel notions of average detectability and average observability referring to systems in which it is possible to reconstruct the average state of the unmeasured nodes (respectively in open and close loop). We will focus here on average detectability that we will introduce more precisely later. Through this chapter we present tools to partition a network in order to ensure this property or a relaxed version of it. The last algorithm proposed in this chapter will be applied to network epidemiology in Chapter 5.

3.1 Introduction

In network theory, observation problems aim to reconstruct the state of the whole system by knowing only the state of few nodes [123]. This topic has been widely studied and has raised the question, among others, of the choice of the measured nodes to improve the reconstruction [46, 62, 124]. In observability and controllability theories, most of the approaches aim to reconstruct or to control the whole state of the systems [65, 76, 99, 106]. To solve the problem in large-scale networks two issues emerge: the complexity of the computation and the limited number of sensors (and/or their price). Taking this into account, it is often difficult to reconstruct the exact state of a large-scale network. Yet, in many instances, there is no need to reconstruct the state of each node, but only an aggregation of these states. For example, in a large-scale traffic network, the number of measurements and the computational effort required to reconstruct the density of vehicles in each street would be disproportionate since the knowledge of the average density in different areas is quite enough for traffic management purposes. Recently, Niazi et al. [96] took advantage of this observation and proposed the notions of average observability and average detectability. These notions refer to the capacity to reconstruct the average state of the unmeasured nodes respectively in closed-loop and in open-loop. Precisely, in an average detectable system, it is possible to design an observer estimating the average of the unmeasured nodes such that the error converges to zero. In this chapter, we focus on this latter notion. In [96] the authors propose a sufficient condition for average detectability allowing to test if a given network and a given subset of measured nodes form an average detectable system. To go further, if it is possible to choose the placement of the measured nodes, a question emerges: given a network what is the smallest subset of nodes to measure in order to fulfill the average detectability condition ? Said otherwise, what is the smallest subset of nodes to measure in order to be able to reconstruct the average of the unmeasured nodes ? This is the question we address throughout this chapter.
In order to transform the condition for average detectability into a structural condition, we will restrict ourselves to a particular type of system: negative uniform network. We will show that with such systems, if the subgraph of unmeasured nodes forms a regular network[1], then the system is average detectable. Therefore, our problem becomes detecting the largest regular induced subgraph (RIS) out of a given network.
The question of finding induced subgraphs with particular properties has been addressed in several works. We can cite the maximum clique problem [18] which has implications, in particular, in social networks; frequent subtree mining [24] which is applied to data analysis; induced subgraph isomorphism problem [113] or its variant as snake-in-the-box problem or the maximum independent set problem. In most cases, these problems are either oriented to data analysis or are graph problems with no direct application. To our knowledge, the present work is the first to use an induced subgraph problem for a reconstruction concern. The problem posed here -the detection of regular induced subgraph- has been studied in different contexts and we will propose a brief review of the works in this domain in Section 3.2.
However we will see that the detection of regular induced subgraph to reach average detectability raises some difficulties. Therefore, in Section 3.3, we relax the problem by introducing the

[1]Here, we call regular a network in which all the nodes have the same *outdegree*. The term *out-regular* would be more accurate, but we prefer *regular* for readability concerns.

notion of quasi-regularity, which qualifies a network which is *close* to be regular. We derive then a result linking the error of regularity and the error of reconstruction: the more regular is the unmeasured subgraph , the better is the reconstruction. On these grounds, we treat the problem of quasi-regular induced subgraph detection (qRIS). Finally, we also extend the approach to multiple quasi-regular induced subgraph (mqRIS) which allows to estimate averages of several subsets of nodes.
The main contributions of this chapter are:

- The emphasis of link between the reconstruction of the average and the regularity of the unmeasured nodes.
- The design of algorithms identifying the nodes to measure such that the average of the unmeasured nodes can be efficiently estimated.

The chapter is organized as follows: after some preliminaries, Section 3.2 presents the exact problem of reconstruction and the algorithm associated and Section 3.3 presents the relaxed problem, the extension to the multiple subgraphs case and the algorithms associated to both problems.

3.1.1 Preliminaries

In this chapter we associate to the network a dynamical equation as introduced in the introduction:

$$\Sigma : \left\{ \begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{array} \right. \tag{3.1}$$

where A is the weighted adjacency matrix of the network, $x(t) = [\, x_1(t), \ldots, x_n(t) \,]^T$ is the state vector associating a value to each nodes of the network and $u(t) = [\, u_1(t), \ldots, u_p(t) \,]^T$ is the input vector.
We will consider that the output vector y contains a sample of k components of the state vector x. This means that $y = [\, x_1(t), \ldots, x_k(t) \,]^T$ and so $C = \left[\begin{array}{cc} I_k & 0 \end{array} \right]$. The nodes $\mathcal{V}_1 := \{v_1, \ldots, v_k\}$ are called the measured nodes while $\mathcal{V}_2 := \{v_{k+1}, \ldots, v_n\}$ are the unmeasured nodes. We denote m the number of unmeasured nodes: $m = n - k$. We also denote $\mathbf{x_2}(t) = [\, x_{k+1}(t), \ldots, x_n(t) \,]^T$, the state of unmeasured nodes and x_2^{av} the average value of the unmeasured nodes:

$$x_2^{av} = \frac{1}{m} \mathbf{1}^T \mathbf{x_2} \tag{3.2}$$

Correspondingly, we decompose the matrices A and B as follows:

$$A = \left[\begin{array}{cc} A_{11} & A_{12} \\ A_{21} & A_{22} \end{array} \right], \; B = \left[\begin{array}{c} B_1 \\ B_2 \end{array} \right]. \tag{3.3}$$

With $A_{11} \in \mathbb{R}^{k\times k}$, $A_{22} \in \mathbb{R}^{m\times m}$ and all the other block matrices of corresponding dimensions. We denote by σ the *deviation vector* defined as:

$$\sigma = \mathbf{x}_2 - \mathbf{1}x_2^{av} = \begin{pmatrix} x_{k+1} - x_2^{av} \\ x_{k+2} - x_2^{av} \\ \vdots \\ x_n - x_2^{av} \end{pmatrix} \tag{3.4}$$

This vector contains the difference between the value of each unmeasured nodes and the average value of the unmeasured nodes. We have $\mathbf{1}^T\sigma \equiv 0$. The evolution of x_2^{av} is described by the following equation:

$$\dot{x}_2^{av} = \alpha x_2^{av} + gy(t) + bu(t) + \eta\sigma(t) \tag{3.5}$$

with $\alpha = \frac{1}{m}\mathbf{1}^T A_{22}\mathbf{1}$, $g = \frac{1}{m}\mathbf{1}^T A_{21}$, $b = \frac{1}{m}\mathbf{1}^T B_2$ and $\eta = \frac{1}{m}\mathbf{1}^T A_{22}$.
Let $\hat{x}_2^{av}$ be an open-loop observer for x_2^{av} described as:

$$\dot{\hat{x}}_2^{av} = \alpha\hat{x}_2^{av} + gy(t) + bu(t) \tag{3.6}$$

We will see in the next section the condition on the system to ensure that the observer $\hat{x}_2^{av}$ converges to x_2^{av}.

3.2 Regular Induced Subgraph (RIS) detection for exact average detectability

In this section, we explore the problem of network partition between measured and unmeasured nodes to ensure the average detectability property that we will precisely define. We will see that under hypothesis, this partitioning problem can be solved by detecting a regular induced subgraph. We will see then how to detect such subgraphs. Finally, we will enlighten some limitations of this approach leading to the relaxed problem presented in the next section.

3.2.1 Average detectable system

Definition 3.1 (Average Detectability: AD)
A system Σ as defined in (3.1)*, with a subset $\mathcal{V}_1$ of measured nodes is said to be average detectable if zero output and zero input implies that the average value of $\mathcal{V}_2$ converges to zero:*

$$\forall\, x_2^{av}(0) \in \mathbb{R}^m, \quad \exists T \in \mathbb{R},\ \forall t > T,\ y(t) = \mathbf{0}_m,\ u(t) = \mathbf{0}_p \implies \lim_{t\to\infty} x_2^{av}(t) = 0 \tag{3.7}$$

where $\mathbf{0}_n$ is a vector of zeros of length n.

This means that, in an average detectable system, if nothing *goes in* or *comes out* the unmeasured subgraph then the average value of the unmeasured nodes will vanish: No internal

dynamics can live only within the unmeasured subgraph. The average detectability of system Σ is in fact equivalent [96] to the convergence of this observer

$$\Sigma \text{ AD} \iff \lim_{t\to\infty} |\hat{x}_2^{av}(t) - x_2^{av}(t)| = 0 \tag{3.8}$$

where $\hat{x}_2^{av}$ is the estimator defined in (3.6).
We exhibit hereafter a sufficient condition for average detectability. In order to introduce the condition for average detectability we first define a network-based notion.

Definition 3.2 (Outflow balanced network)
Let G be a directed weighted network represented by its adjacency matrix A. G is said to be outflow balanced[2] *if the sum of the weights of the outgoing edges is the same for every nodes, which is :*

$$\exists \gamma \in \mathbb{R}, \qquad \boldsymbol{1}^T A = \gamma \boldsymbol{1}^T \tag{3.9}$$

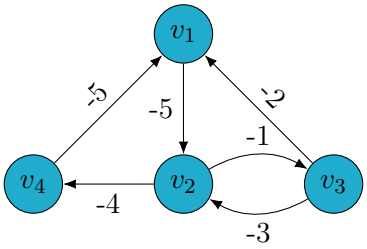
Moreover, if $\gamma < 0$, then the network is said to be negatively outflow balanced.

Example 3.1 illustrates this property.

Example 3.1. Consider the network G represented in the figure hereafter. Its adjacency matrix A is as follows:

$$A = \begin{pmatrix} 0 & 0 & -2 & -5 \\ -5 & 0 & -3 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & -4 & 0 & 0 \end{pmatrix} \tag{3.10}$$

We verify that $\mathbf{1}^T A = -5 \times \mathbf{1}^T$. Hence the network G is negatively outflow balanced.



We can now give the condition for average detectability.

Proposition 3.1 (Sufficient condition for average detectability - Theorem 3 in [96])
Consider a system Σ associated to the network G and denote by $G_{\mathcal{V}_2}$ the subgraph of G induced by the subset of unmeasured nodes. The system Σ is average detectable if $G_{\mathcal{V}_2}$ is negatively outflow balanced.

[2] This term should not be mistaken with the notion of *balanced network* existing in network theory in a different context.

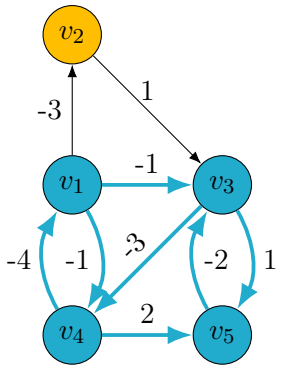


Figure 3.1: By measuring the node v_2, the unmeasured nodes $\mathcal{V}_2 = \{v_1, v_3, v_4, v_5\}$ induces a negatively outflow balanced $G_{\mathcal{V}_2}$ with $\gamma = -2$. Thus, the system is average detectable.

This result implies that, if the set of nodes can be partitioned into two subsets such that one induced subgraph is negatively outflow balanced, then measuring the nodes outside this subgraph makes the system average detectable. Figure 3.1 gives an example of such a system.

3.2.2 Regular Induced Subgraph detection

Motivated by Proposition 3.1, we shall consider the following problem.

Problem 3.1 (Negatively outflow balanced induced subgraph detection)
In a given network, find the largest induced subgraph which is negatively outflow balanced.

Based on Proposition 3.1, by measuring the nodes outside the subgraph found the system is average detectable. However, in an arbitrary network with independent weights there is, *almost surely*, no outflow balanced subgraph. That being said, two solutions can be considered: either we restrict ourselves to particular systems for which we know that negatively outflow balanced subgraphs can be found, or we relax the notion of average detectability and hence the notion of regularity. The latter solution will be explored in Section 3.3 where we introduce the notion of quasi-regularity. In the remainder of this section, we focus on the first solution: we treat the problem for a particular type of systems: the negative uniform networks defined hereafter.

Definition 3.3 (Negative uniform network: NUN)
A network G, represented by the adjacency matrix A, is said to be negative uniform if all its non-zero weights are equal and negative, which is

$$\forall\, i, j,\ A_{ij} \in \{a; 0\} \tag{3.11}$$

with $a < 0$.

Given this definition, the following fact is immediate

Proposition 3.2 (Sufficient condition for average detectability of negative uniform network)
Let Σ be a system associated to a negative uniform network G. Then, Σ is average detectable if $G_{\mathcal{V}_2}$ is regular.

Proof of Proposition 3.2 In a negative uniform network we remark that

$$\mathbf{1}^T A = a \times \deg_{\text{out}}(G), \tag{3.12}$$

Therefore, with (3.9) and (3.12), a negative uniform network is negatively outflow balanced if there exists $\gamma < 0$:

$$\begin{aligned} a \times \deg_{\text{out}}(G) &= \gamma \mathbf{1}^T \\ \deg_{\text{out}}(G) &= \frac{\gamma}{a} \mathbf{1}^T \end{aligned}$$

This mean that all the outdegree of G must be the same, which is G has to be regular. □

Finally we have that finding a regular induced subgraph in a negative uniform network implies average detectability which is equivalent to the convergence of the estimator:

$$\text{RIS \& NUN} \implies \text{AD} \iff \hat{x}_2^{av}(t) - x_2^{av}(t) \underset{t\to\infty}{\longrightarrow} 0 \tag{3.13}$$

Remark 3.1. Although negative uniform network is a quite restrictive case, we can also consider positive uniform network with a same negative self-loop η. Indeed in this case, we have $\mathbf{1}^T A = a \times \deg_{\text{out}}(A) - \eta \mathbf{1}$, and even with $a > 0$ the right side can be negative if the self-loop η is large enough. Proposition 3.2 remains true if the system is associated with such a network. This type of networks includes for example some heat systems with high dissipation [57]. The model of epidemic used in the application in Chapter 5 falls also in this scope. In the rest of the section, we only consider negative uniform network for the simplicity of the development.

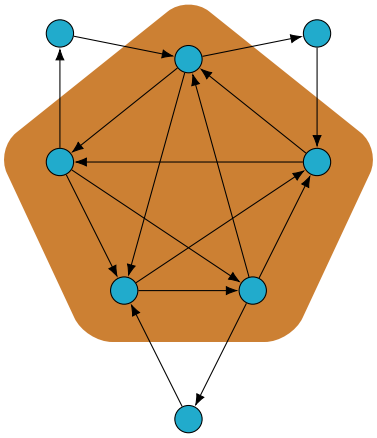
We can now formulate the problem arising from Problem 3.1 restricted to the negative uniform networks case.

Problem 3.2 (Regular induced subgraph detection)
Let G be a negative uniform network. We look for the largest regular induced subgraph (RIS) of G, which is:

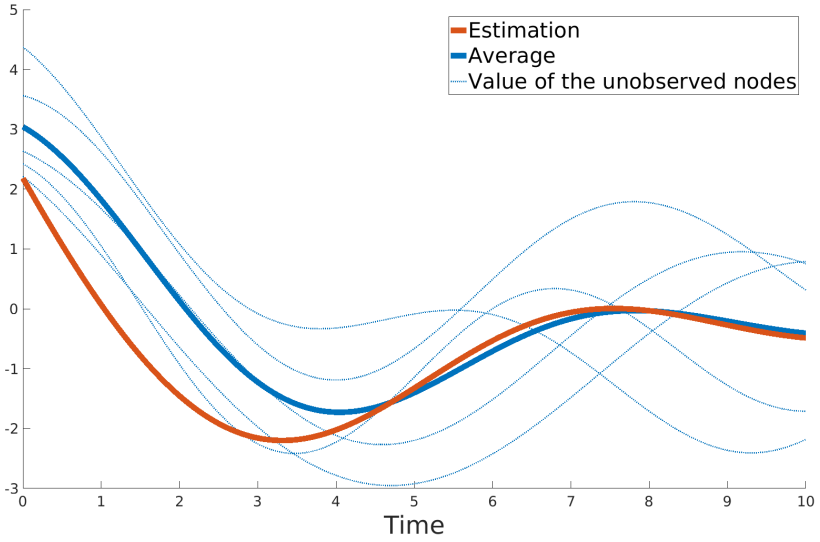
$$\begin{aligned} &\max_{I \subset \mathcal{V}} |I|, \\ &\textit{s.t. } G_I \textit{ is regular.} \end{aligned} \tag{3.14}$$

where G_I is the subgraph of G induced by the subset of nodes I.

Based on Proposition 3.2, by measuring the nodes outside the subgraph solution of Problem 3.2, we obtain an average detectable system. Figure 3.2 illustrates this approach.



(a) We detect a regular induced subgraph. Here the five nodes in the center have a same out-degree equal to 2.



(b) The system is excited with an arbitrary sinusoidal input in some nodes. The dotted blue lines represent the evolution of the unmeasured node and the solid blue line their average. The solid red line represents the evolution of the observer described in (3.6) which converges towards the averaged value.

Figure 3.2: Representation of the approach proposed in this section: (a) from an initial negative uniform network, a regular subgraph is detected; (b) as the system is average detectable, the observer (3.6) converges towards the averaged value of the unmeasured nodes.

Solving Problem 3.2 can benefit from the extended literature on the $k-$regular induced subgraph (or k-RIS) problem, which differs from Problem 3.2 in fixing the desired degree k. The works in this domain can be classified as follows:

- **Complexity of the k-RIS problem**: A first work [79] showed that the problem in the case $k = 0$ is NP-hard. Then several works [4, 20] generalize the result for any k and different type of networks. Some studies [68, 80] exhibit a polynomial complexity for some particular types of network.

- **Algorithms to detect the largest k-RIS**: Despite the complexity of the problem, some algorithms have been proposed to solve it. In the case $k = 0$, [41, 109] propose *fast-exponential* algorithms (which is in $\mathcal{O}(c^n)$ with $c \leq 2$). For any k, a fast-exponential algorithm based on a branch-and-bound approach is proposed in [51] and a polynomial algorithm for a particular type of networks in [1]. However these results consider undirected network and standard regularity (not *out-regularity* as us). Still, the branch-and-bound approach proposed in [51] can be extended to our case and will be presented later.

- **Upper-bound on the size of the largest k-RIS**: Facing the complexity of the problem it is interesting to obtain an upper bound on the size of the optimal solution. An intuitive upper bound on the size of the largest k-RIS is the size of the k-Core of the network. The k-Core is a subgraph obtained by removing iteratively nodes with a degree smaller than k. More elaborate upper bounds have been proposed for any k [20, 81] or for particular values of k [79, 82]. As above, these results are valid only for undirected network and, from our knowledge, only the k-Core approach can be extended to our case.

- **Approximating algorithm to find a sub-optimal solution**: While it would be interesting to find an approximate solution of our problem, it is shown that this problem is hard to approximate [3].

These insights will be the basis for our approach to RIS, which we elaborate next.

3.2.3 An algorithm for the RIS problem

As discussed in the previous section, the literature proposes only methods to find the largest k-regular induced subgraph but nothing to solve the problem for every k. A simple approach consists in solving the problem for each k and then keeping the best solution: We denote RIS(G) and k-RIS(G) respectively, the largest regular and k-regular induced subgraphs of G. We have then:

$$|\,\mathrm{RIS}(G)| = \max_{k\in\mathbb{N}} |\,\text{k-RIS}(G)| \tag{3.15}$$

Some tricks can be used to optimize the approach:

- As seen in the previous section, there are some methods allowing to find an upper bound on the size of k-RIS(G). We denote this upper bound by $\theta_k(G)$. While testing the value of k-RIS(G) for every k, if $\theta_{k_0}(G)$ is smaller than the largest k-RIS(G) so far, it is useless to compute k_0-RIS(G). Noticing that the computation of this upper bound is faster by far than the computation of the k-RIS(G), this helps to compute the regular induced subgraph.
- The cases with $k = 0$ and $k = 1$ are particular: they are known as the maximum independent set and the maximum matching. Specific algorithm can be applied for these case.
- Using an approximate algorithm for the k-RIS would imply a sub-optimal solution for the RIS which can be interesting if the approximation is good.

Based on these remarks, we propose Algorithm 4 to detect the largest regular induced subgraph within a given network.

Algorithm 4 Regular induced subgraph detection

Input: G
1: RIS = []
2: **for** $k = 0 : \max(\deg_{\text{out}}(G))$ **do**
3: $\quad\theta_k$ = UpperBound_k-RIS(k, G)
4: $\quad$**if** $\theta_k >$ | RIS | **then**
5: $\quad\quad$k-RIS = Find_k-RIS(k, G)
6: $\quad\quad$RIS = max(RIS, k-RIS)
7: $\quad$**end if**
8: **end for**
Output: RIS

This is actually a *meta-algorithm* as we gave here only the skeleton of the method and not the *sub-algorithms* UpperBound_k-RIS and Find_k-RIS. The choice of these sub-algorithm is discussed hereafter.

Remark 3.2. The sub-algorithms Find_k-RIS(k, G) and UpperBound_k-RIS(k, G) can be different algorithms for the particular cases $k = 0$ and $k = 1$. However, for directed networks, it seems that there is no specific algorithms providing a clear benefit.

Sub-algorithm for the upper bound

We first discuss the implementation of the UpperBound_k-RIS(k, G) algorithm. This algorithm is detailed in Algorithm 5 and is based on the computation of the k-Core. As said before, the k-Core of a network is the subgraph obtained after removing repetitively the nodes with a degree smaller than k. It is clear that

$$\text{k-RIS}(G) \subset \text{k-Core}(G) \tag{3.16}$$

and so, the size of the k-Core is an upper bound on the size of k-RIS.

Algorithm 5 k-Core

Input: G, k

1: $I \leftarrow$ Nodes in G with outdegree $< k$
2: **while** I is not empty **do**
3: Remove I from G
4: $I \leftarrow$ Nodes in G with outdegree $< k$
5: **end while**
6: $\theta_k \leftarrow$ number of nodes in G

Output: θ_k an upper bound on the size of the k-RIS

Sub-algorithm to find the k-RIS

The implementation of the second sub-algorithm Find_k-RIS(k, G) is described in Algorithm 6. It is an extension of the branch-and-bound approach proposed in [51]. It is a recursive algorithm designed as follows: Given a network G and a degree k, we first compute the k-Core of G (line 1). If the network obtained is $k-$regular then the k-RIS is found (line 3) and the algorithm terminates. Otherwise there is at least one node, denoted r, with a degree larger than k (line 5). It is clear that either r is not in the k-RIS either (at least) one of its successors[3] is not (line 6). Thus, we consider the subgraphs obtained by removing r or a successor of r (line 8). Finally we compute the k-RIS for each of them (line 9) and select the largest one (line 11).

Algorithm 6 k-RIS

Input: G, k

1: $G \leftarrow$ k-Core(G, k)
2: **if** G is $k-$regular **then**
3: k-RIS $\leftarrow G$
4: **else**
5: $r \leftarrow$ a node of G with degree $> k$
6: $I \leftarrow \mathcal{N}_{out}(r) \cup r$
7: **for** $i \in I$ **do**
8: $G_{tmp} \leftarrow$ remove i in G
9: k-RIS$_{tmp}\{i\}$ = k-RIS(G_{tmp}, k)
10: **end for**
11: k-RIS = max(k-RIS$_{tmp}$)
12: **end if**

Output: k-RIS

Remark 3.3. It is possible to use a technical trick to speed-up the algorithm. The size of the current best solution can be stored and passed to the recursive call to k-RIS (line 9). If the size of the current subgraph is smaller than the size of the best solution, the recursion stops. This shortcut is not in algorithm 6 for the sake of readability.

[3] $\mathcal{N}_{out}(r)$ is the set of successors of r

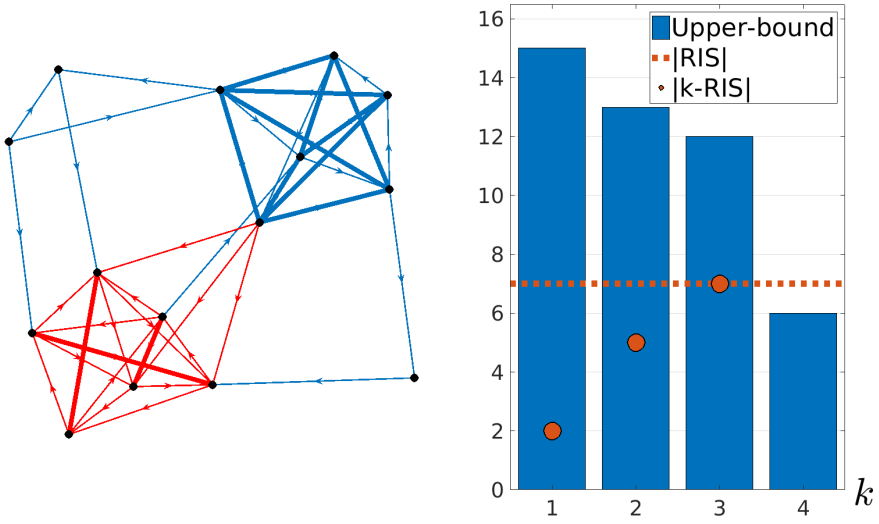


Figure 3.3: Left: The largest regular induced subgraph found is highlighted in red. The double-way edges are thicker. All the nodes in the subgraph have an outdegree equal to 3. Right: The upper bound (k-Core) is displayed in blue for $k \in \{1;2;3;4\}$. The real size of k-RIS are in dot red for $k \in \{1;2;3\}$. The 4-RIS is not computed since $\sigma(4) <$ |3-RIS|. The red dotted line represents the size of the RIS.

Simulation 3.1. Figure 3.3 presents a result obtained with Algorithm 4 (and Algorithms 5-6). Considering that the network is negative uniform, the system is average detectable by measuring the 8 nodes outside of the red subgraph.

3.2.4 Discussions

The method proposed above suffers some limitations which we summarize below:

- Negative uniform systems, which motivate our search for RIS, are rare: It seems that there is no practical example in the literature. Even if we consider positive uniform systems with large negative self-loop, as mentioned in Remark 3.1, the applications are scarce.
- RIS is a fragile notion: For example, a grid network which is very close to regular (only the nodes on the border have a smaller degree) does not fulfill the condition of the theorem. Moreover, the largest RIS in a network is generally relatively small.
- The RIS problem is hard to solve: there is no specific method for the RIS problem and the k-RIS problem has to be solved several times. Moreover, as said before, the k-RIS problem is a NP-hard problem and there is no *good* approximation algorithm to solve it.
- The largest RIS in a network is often very small, and suboptimal heuristics return even smaller subgraphs.

Based on these statements, the RIS detection approach introduced in this section is difficult to apply to real-world problems. We will see that it is possible to accept some errors on the regularity will preserving the possibility to reconstruct efficiently the average. We introduce, in the next section, the notion of quasi-regularity leadong to more flexibility.

3.3 Approximate average detectability

3.3.1 Link between error of regularity and error of reconstruction

In Proposition 3.1, the sufficient condition for average detectability was $\mathbf{1}^T A_{22} = \gamma \mathbf{1}^T$ with $\gamma < 0$. To relax the problem we consider now that this equality is no more exactly verified and we introduce a perturbation vector s defined as:

$$\gamma = \underset{\gamma' \in \mathbb{R}}{\operatorname{argmin}} \, \mathbf{1}^T A_{22} - \gamma' \mathbf{1}^T, \qquad s = \mathbf{1}^T A_{22} - \gamma \mathbf{1}^T \tag{3.17}$$

and a regularity error ϵ defined as:

$$\epsilon = \frac{\|s\|_1}{m|\gamma|} \tag{3.18}$$

We denote $\tilde{x}_2^{av} := x_2^{av} - \hat{x}_2^{av}$ the reconstruction error. The following proposition gives the link between the regularity error the reconstruction error, defined as:

$$e_{ss} = \limsup_{t \to \infty} |\tilde{x}_2^{av}| \tag{3.19}$$

Proposition 3.3 (Relation between regularity error and reconstruction error)
Assume that

$$\frac{1}{m} \boldsymbol{1}^T A_{22} \boldsymbol{1} < 0 \tag{3.20}$$

Then, the reconstruction and regularity errors satisfiy

$$e_{ss} \leq \max_i \bar{\sigma}_i \frac{\epsilon}{1 - \epsilon}, \tag{3.21}$$

where $\bar{\sigma} = \limsup_{t \to \infty} |\sigma(t)|$.

Proof of Proposition 3.3 From equations (3.5) and (3.6) we have then:

$$\dot{\tilde{x}}_2^{av}(t) = \alpha \tilde{x}_2^{av}(t) + \eta \sigma(t) \tag{3.22}$$

where σ defined in (3.4) is the deviation from average,

$$\alpha = \frac{1}{m} \mathbf{1}^T A_{22} \mathbf{1} = \frac{1}{m} (\gamma \mathbf{1}^T + s^T) \mathbf{1} = \gamma + \frac{1}{m} s^T \mathbf{1} \tag{3.23}$$

aand since $\mathbf{1}^T \sigma(t) = 0$,

$$\eta \sigma(t) = \frac{1}{m} \mathbf{1}^T A_{22} \sigma(t) = \frac{1}{m} (\gamma \mathbf{1}^T + s^T) \sigma(t) = \frac{1}{m} s^T \sigma(t) \tag{3.24}$$

Therefore, (3.22) becomes

$$\dot{\tilde{x}}_2^{av}(t) = (\gamma + \frac{1}{m}s^T\mathbf{1})\tilde{x}_2^{av}(t) + \frac{1}{m}s^T\sigma(t) \tag{3.25}$$

which is a stable dynamics by virtue of (3.20). Since A is stable, $\sigma(t)$ is bounded in magnitude and

$$\left|\limsup_{t\to\infty} \tilde{x}_2^{av}(t)\right| \leq -\frac{\frac{1}{m}|s^T\bar{\sigma}|}{\gamma + \frac{1}{m}s^T\mathbf{1}} \tag{3.26}$$

Finally,

$$-\frac{\frac{1}{m}|s^T\bar{\sigma}|}{\gamma + \frac{1}{m}s^T\mathbf{1}} = \frac{|s^T\bar{\sigma}|}{m|\gamma| - s^T\mathbf{1}} \leq \frac{|s^T\bar{\sigma}|}{m|\gamma| - |s^T\mathbf{1}|} \leq \frac{\|s\|_1 \max_i \bar{\sigma}_i}{m|\gamma| - \|s\|_1} \tag{3.27}$$

and (3.21) follows from with the definiton of ϵ in (3.18) □

The constant $\bar{\sigma}$ depends on the physical system of interest and corresponds to the maximal difference between the state of an unmeasured node and the average state of the unmeasured nodes.
According to Proposition 3.3, in order to minimize the reconstruction error, it is interesting to find a subgraph having a reasonable regularity error ϵ. Relation (3.21) is emphasized through simulations in the following section.

Remark 3.4. The reconstruction error e_{ss} grows with the regularity error ϵ which means that the reconstruction will be better if the subgraph of unmeasured nodes is *large*, *close to be regular* and with a *large degree of regularity.*

Before investigating the qRIS detection problem, we present some simulations enlightening the relation described in Proposition 3.3. To this end, we first introduce a family of networks for which we can control the regularity.

Definition 3.4 ($p-$reg network)
Given a network G, we denote $\mathcal{N}_{out}(i) = \{j, (j,i) \in \mathcal{E}\}$, the set of successors of i.
A network is said $p-$reg if it verifies:

$$\begin{cases} |\mathcal{N}_{out}(i)| = 1 & \text{if } i \text{ is odd} \\ |\mathcal{N}_{out}(i)| = p & \text{if } i \text{ is even} \end{cases} \tag{3.28}$$

Graphs of this family have the particularity to have one half of their nodes with outdegree 1 and the other half with outdegree p. In particular if $p = 1$, the network is 1-regular (it is a cycle). By increasing the value of p, the regularity worsens as shown in fig. 3.4. In the following numerical simulations, we consider a series of networks composed of a p-reg network and one additional measured node as in fig. 3.5. Therefore, with such network, by tuning the value of p we can modify the regularity of $G_{\mathcal{V}_2}$ while preserving the shape of the system. The network used in the experiment contains 101 nodes (100 nodes in the p-reg network plus one extra node to measure). We add some random inputs to the system and we compare the error e_{ss} for different value of p. The results are displayed in fig. 3.6. We notice that as predicted by

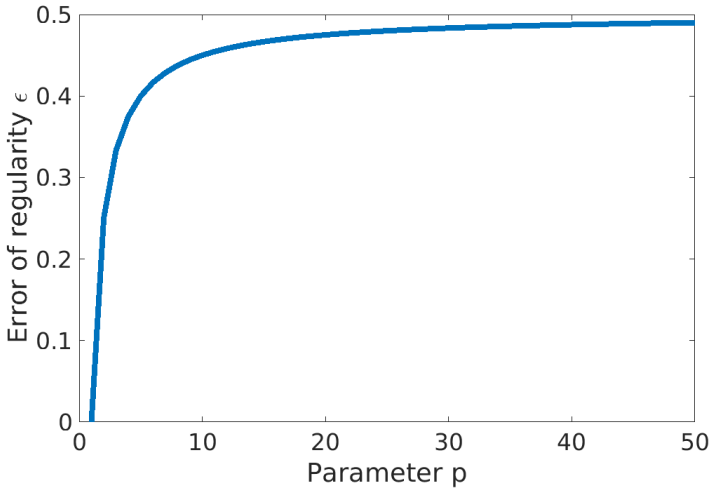


Figure 3.4: The family of $p-$reg networks allows to control the regularity of a network.

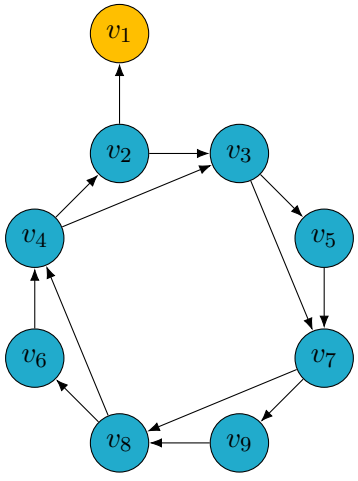


Figure 3.5: A 2-reg network as defined in Definition 3.4 with an additional node to measure. For the experiments, a similar network is used with 100 nodes and p varying from 1 to 100.

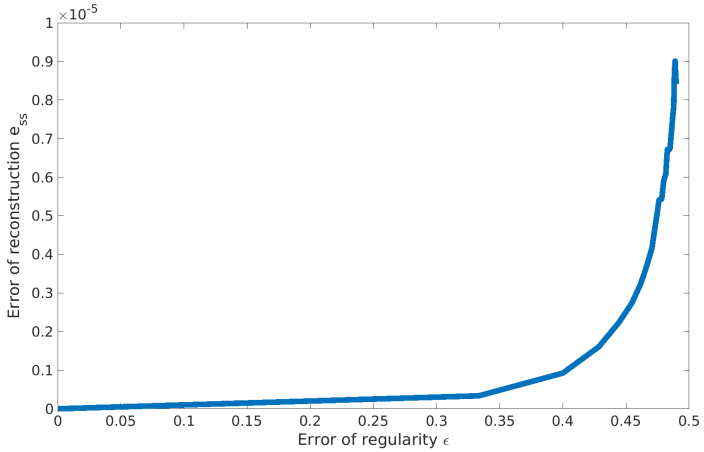


Figure 3.6: Error of reconstruction as a function of the error regularity for the family of p-reg network. For each $p \in [1, ..., 100]$, 25 p-reg network are generated with different inputs. The error of reconstruction is then computed as the bias between the signal and the reconstruction at $t = 1000$ (which is a good approximation of e_{ss}). This emphasizes the link between the regularity of a subgraph and the ability to reconstruct its average.

Proposition 3.3 the error of reconstruction grows with the error of regularity. This simulation enlightens the fact that by minimizing the error of regularity ϵ we can control the error of reconstruction e_{ss}.

3.3.2 Quasi-Regular Induced Subgraph detection

While detecting quasi-regular subgraph, if a small error is imposed on the regularity, the subgraph found might be small (and so the number of nodes to measure would be high). In the other hand, Proposition 3.3 ensures that the reconstruction would be better. Consequently, a compromise between the number of measure and the quality of the reconstruction has to be found. An interesting way to implement this compromise is by fixing a threshold for the error of regularity and then find the largest subgraph whose regularity error is lower than this threshold. This leads to the qRIS detection problem defined hereafter.

Problem 3.3 (Quasi-regular induced subgraph detection)
Let G be a network and $\epsilon_0 > 0$ a threshold for the quasi-regularity. We look for the largest quasi-regular subgraph of G, which is:

$$\begin{aligned} &\max_{I \subset \mathcal{V}} |I|, \\ &s.t.\ \epsilon(G_I) < \epsilon_0 \end{aligned} \tag{3.29}$$

where $\epsilon(G)$ is the regularity error associated to G as defined in (3.18).

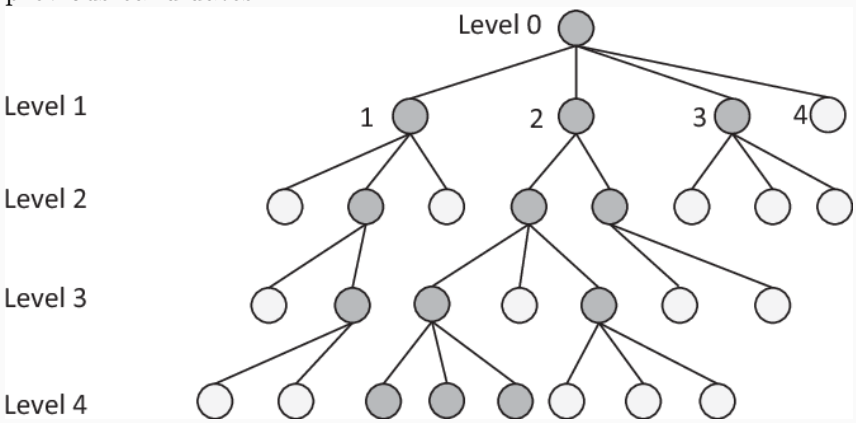
Therefore, by measuring the nodes outside the subgraph solution of Problem 3.3 we can reconstruct the average value of the subgraph with an error of the order of $\bar{\sigma}\frac{\epsilon_0}{1-\epsilon_0}$. Note that

this problem may have no feasible solution if ϵ_0 is chosen to be too small. After this caveat, we present a beam search algorithm providing a sub-optimal solution to Problem 3.3.

3.3.3 An algorithm for qRIS detection

Like in the RIS case, a combinatorial algorithm exploring every possible subgraph would find the exact solution but it is uncomputable in practice even for networks with a relatively small size. We present here a beam-search algorithm providing a sub-optimal solution to Problem 3.3.

Principle of beam search algorithms The principle of beam-search algorithms is as follows: A set of candidate solutions is considered as a seed. Then, a set of solution deriving from these candidates is considered and the β most *promising* are memorized (β is called the beam width) and form the new set of candidates. The algorithm stops when a candidate is satisfying enough or when the new candidates are no more satisfying. In this latter case, the final solution is chosen among all the previous candidates.



For the quasi-RIS detection the algorithm is described in Algorithm 7: We initialize the set of candidates with the singletons of each node (line 1). Then we iterate while one of the candidate has a regularity error smaller than the minimum accepted ϵ_0 (line 2) (see Remark 3.5 for a discussion on this point). At each iteration, new candidates are derived from the current candidates (line 4-5). These new candidates J_c are all the subsets composed by one current candidate c and any other nodes $\mathcal{S}_i$ of the network. Finally all these new candidates are united (line 7), the β best form the new candidates (line 8) and the best one is stored as $\text{quasi}-\text{RIS}$ (line 9). By repeating this operation several times, the size of the candidates grows until none of the candidate have a regularity error small enough. At the end, the candidate with the smallest regularity error is chosen. Figure 3.7 illustrates this algorithm. Beam-search algorithms, like this one, are a type of greedy algorithm and hence do not provide an optimal solution. However the computations are relatively fast and the solutions found are rather good, as we will see.

Remark 3.5. In Algorithm 7 the *while* loop will stop if none of the candidate have a regularity

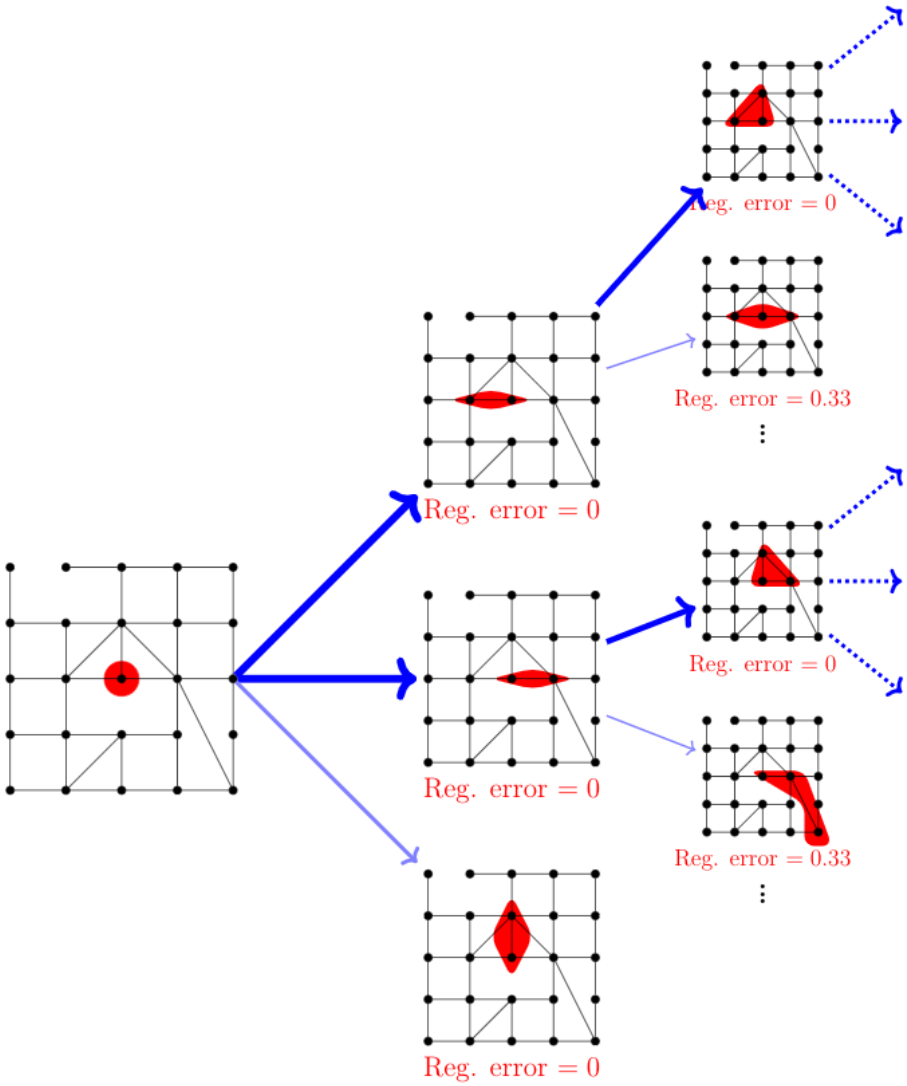


Figure 3.7: Illustration of Algorithm 7. Here the beam width $\beta = 2$, so at each step the best two candidates are kept (solid line). Two small modifications are brought to make the example more readable: the seed is a single candidate while in the algorithm the seed is composed of the singletons of each node; the new candidates are the subsets composed of a previous candidate and a node *neighboring* this candidate, while the latter node can be arbitrary chosen in the algorithm.

Algorithm 7 quasi-RIS detection

Input: G: network with n nodes, ϵ_0 maximum acceptable error, β beam width

1: Cand $\leftarrow \{\{1\};\{2\};\dots;\{n\}\}$
2: **while** $\min_{I\in\text{Cand}} \epsilon(G_I) < \epsilon_0$ **do**
3: **for** $c \in$ Cand **do**
4: $\mathcal{S} \leftarrow [1,\dots,n]\backslash c$
5: $J_c = \{c\cup\mathcal{S}_1; c\cup\mathcal{S}_2;\dots; c\cup\mathcal{S}_{end}\}$
6: **end for**
7: $\Omega \leftarrow \bigcup_c J_c$
8: Cand $\leftarrow \beta$ smallest $\epsilon(G_c)$ for $c \in \Omega$
9: quasi-RIS $\leftarrow$ smallest $\epsilon(G_c)$ for $c \in \Omega$
10: **end while**

Output: quasi$-$ RIS

error small enough. However, this condition is relatively strict as it is possible that it is not verified at some iteration but it will be verified in the future. Thus, a relaxed condition would be to stop if the the condition is disrespected several iterations in a row. Then, the solution would be chosen among the last candidate verifying the condition.

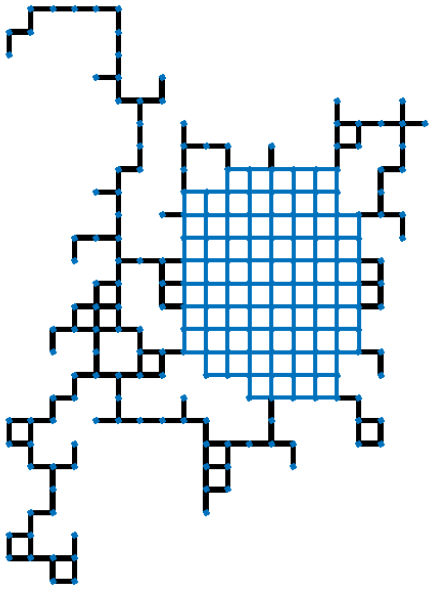
Remark 3.6. The new candidates are chosen in a way that may induce a disconnected subgraph. This allows to explore more potential solutions. However, as discussed in Chapter 1, for some applications the connectedness of the unobserved subgraph may be required or wished. For example when the network has a geographical nature (as urban traffic network or electrical grid), it is interesting to estimate the average of a geographical area. In this case, the algorithm can be adapted by changing line 4 with $\mathcal{S} \leftarrow \mathcal{N}_{in}(c) \cup \mathcal{N}_{out}(c)$. Figure 3.7 implements this solution.

Simulation 3.2. For this simulation, we designed a network including a particularly regular subgraph to emphasize the ability of the algorithm to detect it. In this simulation the parameters are $\epsilon_0 = 0.1$ and $\beta = 300$ which means that at each step we conserve the 300 best candidates. The subgraph obtained is presented in fig. 3.8(a). We compute then the reconstruction of the average in this subgraph. Figure 3.8(b) shows the actual average and the estimation made by measuring the nodes outside the subgraph.

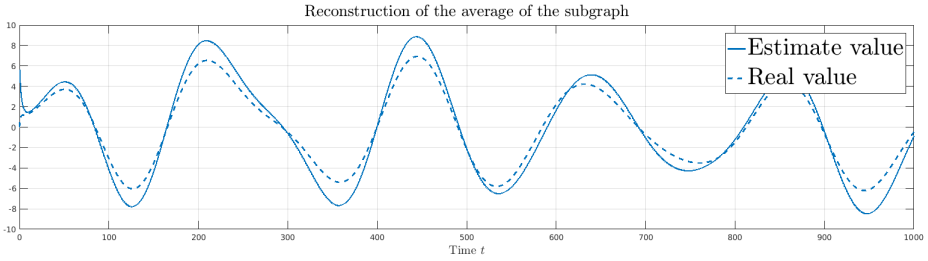
We have seen how to detect a regular or quasi-regular subgraph in order to estimate their average. We propose, in the next section, to generalize this approach, to detect several subgraphs and estimate different averages.

3.3.4 Extension to Multiple quasi-Regular Induced Subgraphs (mqRIS)

In the previous problems we tried to find one regular or quasi-regular subgraph in order to estimate its average. We wonder, now, to which extent it is possible to detect several regular or quasi-regular subgraphs and estimate their respective average. Considering the limitations posed by the regularity case evoked in Section 3.2.4, we focus only on the quasi-regular problem. However a similar generalization can be led for the regularity case. In



(a) In blue, the quasi-regular subgraph detected with algorithm 7.



(b) Reconstruction of the average of the quasi-regular subgraph obtained

Figure 3.8: Illustration of the qRIS approach. Our algorithm is able to detect a subgraph (a) which is enough regular to make the reconstruction of its average quite good (b).

the RIS and qRIS problems we wanted to have the minimum number of nodes to measure leading to the minimization problems 3.2 and 3.3. Here again we have the same objective to minimize the number of measured nodes. Thus, we want to find disjoint quasi-regular induced subgraphs $G_{I_1}, \ldots, G_{I_m}$ maximizing the cardinality of the union of the subgraphs. Moreover, these subgraphs can not share any successors which is nodes outside the subgraph and pointed by a node of the subgraph. This is because the successors of a subgraph are measured to estimate the average value of the subgraph. If a node is pointed by two nodes belonging to two different subgraphs, the condition for the reconstruction does not hold (see [96]). We denote $\mathcal{I} = [I_1, \ldots, I_m]$ the set of the subsets of nodes. The problem tackled in this section is formulated as follows:

Problem 3.4 (Multi quasi-regular induced subgraph detection)
Let G be a network and $\epsilon_0 > 0$ a threshold for quasi-regularity. We look for a set $\mathcal{I}$ of quasi-regular subsets minimizing the number of nodes to measure, which is:

$$\begin{aligned} &\max_{\mathcal{I}=[I_1,\ldots,I_m]} \left|\bigcup I_j\right|, \\ s.t.\ &\forall\, i,\ \epsilon(G_{I_i}) < \epsilon_0 \\ &\forall\, i, j,\ (I_i \cup \mathcal{N}_{out}(I_i)) \cap (I_j \cup \mathcal{N}_{out}(I_j)) = \emptyset \end{aligned} \tag{3.30}$$

The second constraint translates the non-overlapping of the subgraphs and their successors. The quasi-RIS detection algorithm 7 presented in the previous section can be extended almost straightforwardly to the multiple subgraphs case as follows: a first quasi-regular subgraph is detected, the subgraph and its neighborhood (which are the nodes to measure) are removed from the network and the process is repeated with the new network. To limit the number of nodes to measure it is interesting to limit the number of neighbors of the subgraph selected at each iteration. To this end, the new candidate at each iteration are the subgraphs maximizing $|I|/|\mathcal{N}_{out}(I)|$ which is the ratio between the size of the subgraph and the size of its neighborhood instead of the subgraph minimizing $\epsilon(G_I)$ in Algorithm 7.

The algorithm for multi quasi-RIS detection is described in Algorithm 8 where quasi-RIS* refers to the algorithm 7 where "smallest $\epsilon(G_c)$" in lines (8) ans (9) is replaced by

$$\underset{c\in\Omega,\,\epsilon(G_c)<\epsilon_0}{\operatorname{argmax}} \frac{|c|}{|\mathcal{N}_{out}(c)|} \tag{3.31}$$

Simulation 3.3. We propose here a simulation of Algorithm 8 for the mqRIS detection. To this end, we consider an initial network (fig. 3.9) designed with five zones more regular than the average. This aims to test the capacity of the algorithm to detect regular subgraphs. The result of the simulation is displayed in fig. 3.10. In this case, the quasi-RIS detection algorithm is applied five times before no more satisfying subgraphs can be found. At each step, we can see that the subgraph and its out-neighborhood found at the previous step is removed and the algorithm is applied with the new network obtained.

Algorithm 8 Multi quasi-RIS detection

Input: G: network with n nodes, ϵ_0 maximum acceptable error, β beam width, i_{max} maximum number of subgraph detected

$\mathcal{I} = []$
$I = \text{quasi-RIS}^\star(G, \epsilon_0, \beta)$
while I is not empty **do**
  $\mathcal{I} = \mathcal{I} \cup I$
  $G = \text{remove } I \cup \mathcal{N}_{out}(I) \text{ from } G$
  $I = \text{quasi-RIS}^\star(G, \epsilon_0, \beta)$
end while

Output: $\mathcal{I}$ set of subsets inducing multiple quasi-RIS

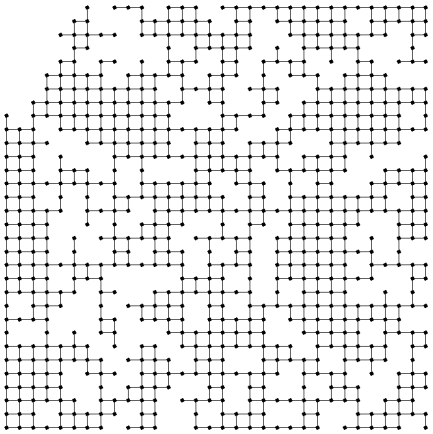
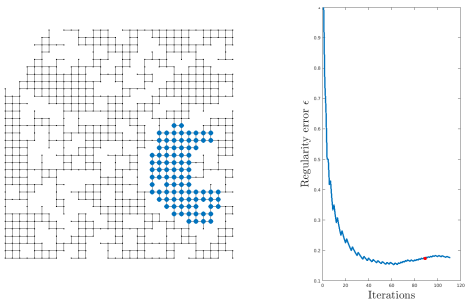


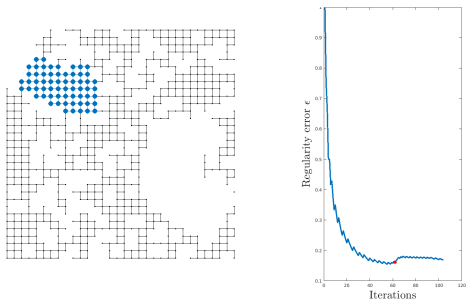
Figure 3.9: The initial network is a grid designed such that certain zones are more regular. The algorithms aims to find these zones. These zones are not easily identifiable at the naked eyes and the reader is invited to try it before looking at the result of the algorithm.

3.4 Conclusion

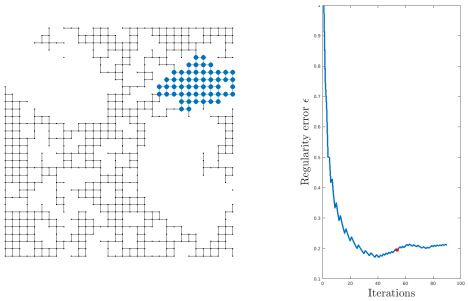
Based on the novel notion of average detectability, we proposed here three algorithms identifying measured node placement in order to estimate the average of the unmeasured subgraph. Considering a particular type of system, the first algorithm finds regular induced subgraph to reach exact average detectability which is the estimation of the average is asymptotically unbiased. Due to the limitation of this first problem, we proposed a relaxation: we focused on the detection of *quasi-regular* induced subgraph which results in an estimation of the average with a bias which depends on the quasi-regularity. The second algorithm achieves this task. The third algorithm allows to detect several quasi-regular induced subgraphs to estimate the averages of different subsets of the system. In Chapter 5 we will apply the third algorithm to estimate the evolution of a disease spreading in an interaction network over French territory.



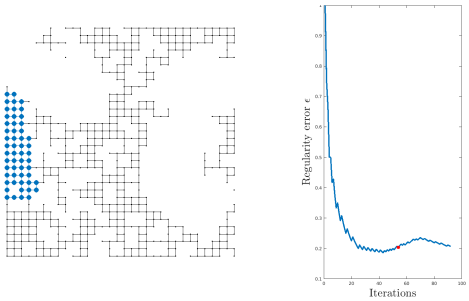
(a) A first subgraph is found (in blue). On the right the evolution of the regularity error of the best candidate through the iterations. The red dot corresponds to the iteration where the subgraph maximizing (3.31) is found.



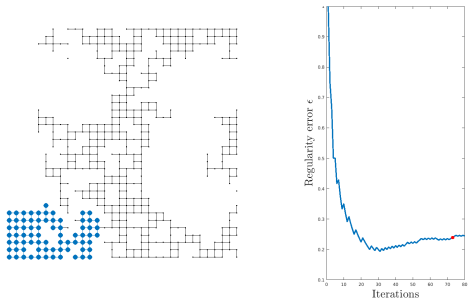
(b) The first subgraph found and its successors are removed and the algorithm is applied again to detect a second subgraph (in blue).



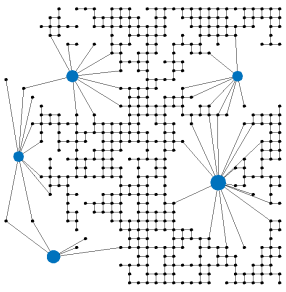
(c) The second subgraph found and its successors are removed and the algorithm is applied again to detect a third subgraph (in blue).



(d) The third subgraph found and its successors are removed and the algorithm is applied again to detect a fourth subgraph (in blue).



(e) The fourth subgraph found and its successors are removed and the algorithm is applied again to detect a fifth subgraph (in blue). The algorithm stops here as there is no other subgraph which are enough regular.



(f) The network obtained when merging the subgraphs obtained at each steps. By measuring the remaining nodes, the average of the subgraphs detected can be estimated.

Figure 3.10: **(a-e)** Left: in blue the detected subgraph within the current network; Right: the evolution of the regularity error of the best candidate is displayed through the iteration. A red dot shows the iteration of the selected subgraph. **(d)** represents the network obtained by merging the different subgraphs. This reduced network offers an estimation of the initial network, and thus can be seen as an aggregation of it.

Chapter 4

Application to traffic

> The question of whether machines can think is about as relevant as the question of whether submarines can swim
>
> Edsger Djikstra

Contents

This chapter is dedicated to present a concrete implementation of the partitioning algorithm introduced in Chapter 2. We consider here the urban traffic network of Grenoble that we will present in a first section. Then, a partition of this algorithm is obtained thanks to the MergeToScaleFree algorithm. Finally, due to the properties preserved by the algorithm, the reduced network is used to simulate the dynamics of the initial system.

4.1 Traffic network of Grenoble

4.1.1 Geographical network

Situated in the south-east of France, close to the Alps, Grenoble is the 16th largest city of France. The situation of the city, surrounded by three ranges of mountains, constraints the urban traffic network making Grenoble the fifth most congested city in France [115]. See fig. 4.1 for a satellite picture of the city. In the framework of the ScaleFree-Back project[1] the collection of traffic data and the monitoring of the traffic condition over the whole city is

[1] For further information see `http://scale-freeback.eu/`.

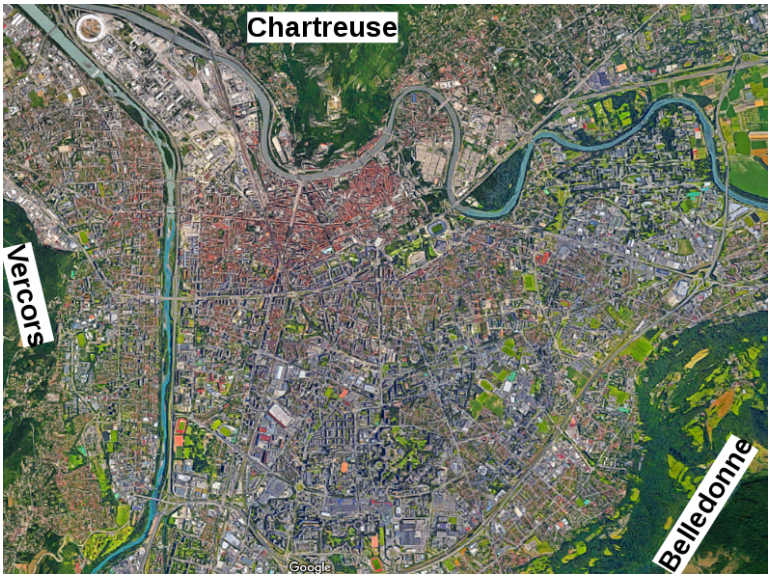


Figure 4.1: Satellite picture of Grenoble. We can see the Vercors range in the west, the Chartreuse range in the north and the Belledonne range in the south-east constraining the development of the urban traffic network.

studied and is a part of the *GTL-ville* experimentation[2] (following the *GTL* experimentation started in 2009 [122]). As large networks are hard to control, we wonder how an abstracted version of the network could help to design a control strategy and how the scale-freeness of this abstracting network can be an advantage. It is in this framework that the study of the reduction of the urban traffic of Grenoble takes place. The network, provided by TomTom, is a representation of the city center of Grenoble as it contains all roads and intersections within an area of about 5 km per 5 km. Figure 4.2 represents the area considered within the Grenoble Metropolis. In this network, the nodes correspond to the roads and there exists an edge between two nodes n_i and n_j if there is an intersection linking road n_i to road n_j. The weight on the edges is an estimation of the flow of vehicles going from one road to another in average. The computation of this estimation is explained hereafter.The network possesses 13099 nodes and 31570 edges.

4.1.2 Estimation of the flow

In the light of the properties preserved by the MergeToScale-Free algorithm, and in particular the eigenvector centrality, one of the objective of this reduction could be to estimate the distribution of vehicle in the city at the steady state. To this end, the weights assigned to each edge of the network represents the average flow of vehicle going from one road to another. To estimate this flow we consider two different pieces of information: the number of lanes and the speed limit of the downstream road. Indeed, the more important (in terms of flow) a road

[2]The GTL-ville aims to monitor the traffic condition inside the whole city of Grenoble. See *http://gtlville.inrialpes.fr/*.

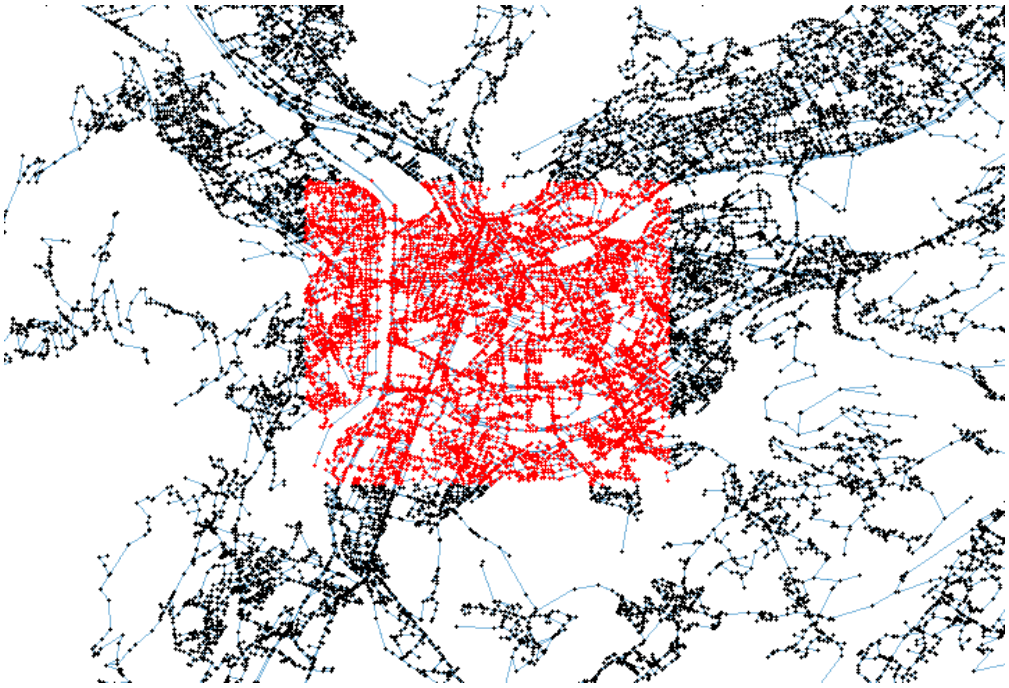


Figure 4.2: The traffic network of Grenoble metropolis and in red the city center that we will consider here.

is, the higher is its speed limit and the larger is the road. Based on this assumption the weight w_{ij} from road n_i to road n_j is computed as follows:

$$w_{ij} = l_j v_j^{max} \tag{4.1}$$

where l_j is the number of lanes in road j and v_j^{max} is the speed limit on road j. These data are provided by TomTom.
However, with this computation, nothing ensures that the network is a flow network[3]. To ensure this we project the network into the flow network space. This computation is explained in Appendix B. Figure 4.3 illustrates the network model.

4.1.3 Physical properties of traffic networks

In a traffic network, if the weights represent the average flow of cars from a street to another, the flow coming in one node is equal to the flow coming out the node[4]. Therefore, the mass conservation is an intrinsic property of the system, and the preservation of this property through the reduction is essential to ensure the physical consistency. The cancellation of the eigenvector distance ensures that the dynamics of the initial network and the abstracting network are consistent. In particular, the number of vehicles in each zone of the network at

[3] as defined in Section 0.2.2 a flow network is a network with the mass conservation property
[4] In this model we neglect the cars parking along a street which would violate the mass conservation property

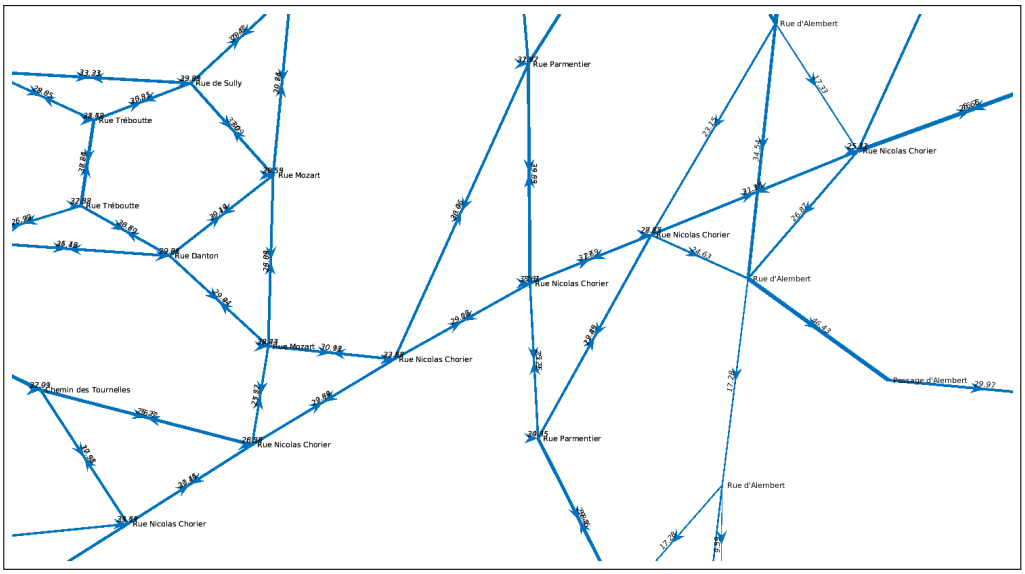


Figure 4.3: Sample of the network: the nodes represent the streets, and the weights on edges are estimated flow from one node to another. The weights are computed such that the network is a flow network.

the steady state will be the same in the two systems. The preservation of the total sum of all weights guarantees that the number of vehicles is the same in the initial and the reduced networks.
Therefore, Algorithm 3, which ensures the preservation of these three properties, is particularly relevant for traffic network. In the next section we present the results obtained with the algorithm and we compare the dynamics of the two networks.

4.2 Reduction of the Grenoble traffic network

Within this section, we apply the method presented in Chapter 2 in order to obtain a scale-free network abstracting the Grenoble traffic network and preserving a certain consistency. We will see then, how this reduced network can be used to simulate the dynamics of the system compared to the initial network.

4.2.1 The reduced system

We apply the reduction Algorithm 3 on this network and we impose a scale-freeness on the indegree distribution. The choice of the different parameters of the algorithm is presented in Table 4.1. As a reminder, $|G|$ is the number of nodes in the initial network, α_{SF} is the imposed scale-free coefficient for indegree, n_{rand} is the number of edges tested at each iteration and n_{iter} gives the number of iterations. Hence, the number of nodes in the reduced network is 2099 which correspond to a reduction of 83.9%.

Table 4.1: Parameters of the simulation on the Grenoble urban traffic network

$\|G\|$	α_{SF}	n_{rand}	n_{iter}
13099	2	50	11000

Figure 4.4 presents the initial network and the partition found by the algorithm. Figure 4.5 presents the reduced network coming out of this partition and the different degree distributions. The ability of the algorithm to get close to the desired distribution is clear as the reduced network exhibits a power law degree distribution. Few hubs are visible on the reduced network. However, as seen in Chapter 2, when the reduced network becomes too small it tends to loose its scale-free structure. Therefore, a compromise must be found between the scale-freeness of the reduced network and its smallness.

4.2.2 Simulations on the reduced network

Model

We will now see that the reduced network can be used to analyze the dynamics of the system. We consider a linear time-invariant discrete-time system:

$$\Sigma : x(k+1) = Px(k) + B\phi_{in}(k) \tag{4.2}$$

where x represents the number of vehicles in each road, P is the normalized adjacency matrix issued from the adjacency matrix A as defined in (2.13), B points the road having an in-flow, and ϕ_{in} represents the dynamics of the inflows. The computation of the adjacency matrix A in order to estimate the real flows and to ensure the mass conservation property has been explained in the previous section and in Appendix B. In this model, the distribution of vehicle at time $k+1$ is given by the state at time k and the matrix P describing the transitions from one road to another. Therefore, the time from a vehicle to go from one node to another does not depend on the length of the street, which is a drawback of the model. However, we can roughly estimate that this time is, in average, around thirty seconds. Therefore in the simulation, we consider that the time past between two states is thirty seconds.
Let us note that a linear system like this one is not realistic for traffic system where a lot of non-linear phenomena occur. However, in the case of free-flow (which is without any congestion on the roads) a linear system is acceptable.
The results obtained in Chapter 2 state that the algorithm preserves the steady-state in the reduced system if the evolution equation is:

$$\dot{x}(t) = Px(t) \tag{4.3}$$

This equation does not include any input. This is why, in the scenario we will consider, the in-flows stop after a while ($\phi_{in}(t) = 0,\ t > T$) to show that after that the distributions converges towards each other. The advantage of considering some in-flows at the beginning of the simulation is to emphasize the capacity of the abstracting system to emulate the dynamics of the initial system even in the range where we do not have theoretical results.

Scenario

We consider that, at $t = 0$, there is no car inside the network which is $x(0) = 0$. This can be seen as an ideal case of the network during the night, where there is almost no traffic. Then, in the morning, at $t > 0$, some cars enter into the network. The entry points of the network, pointed by matrix B, are the roads at the border of the network[5]. As shown in fig. 4.6, we divide the entry points in three categories: the ones from the north (in blue), the ones from the east (in red) and the ones from the south (in yellow). The west border remains unexcited. Different dynamics govern the inflow in these different entry points. After a while, when $t = 400$ (which corresponds to more than 3 hours) the in-flows are stopped and we observe the evolution of the vehicles distribution converging towards an equilibrium.

Simulations

Figure 4.7 shows the results of the simulation after 2000 time steps[6]. The color displayed here represents the relative density at each node. Precisely the color of nodes v_i at time t is proportional to:

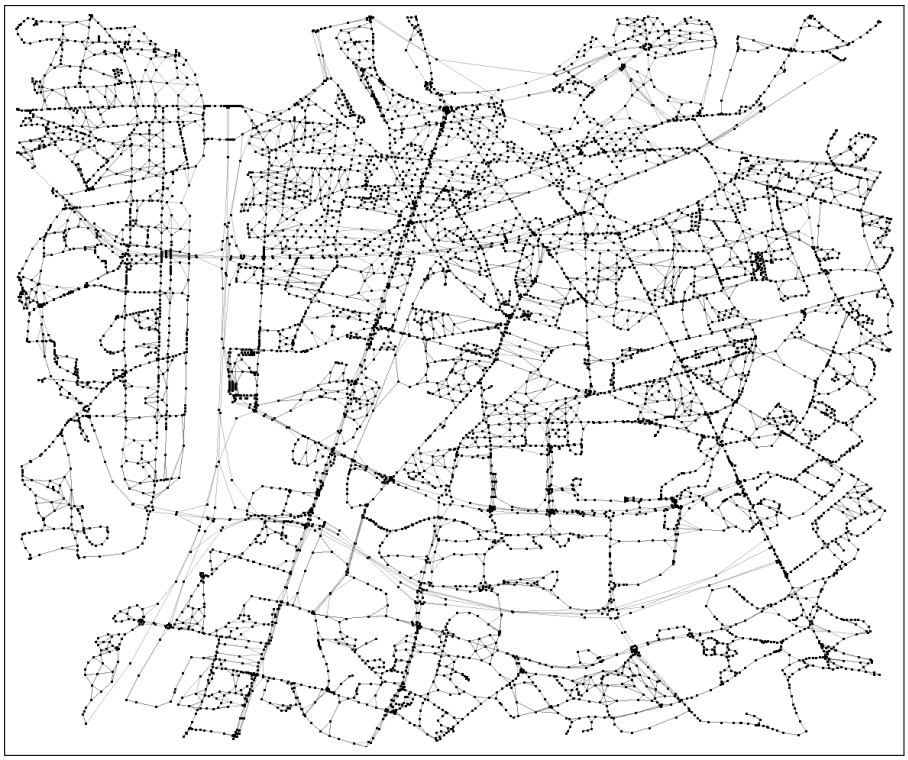
$$c_i(t) = \frac{x_i(t)}{\max_j x_j(t)} \tag{4.4}$$

and the color on the edges corresponds to the average color of its two extremities.

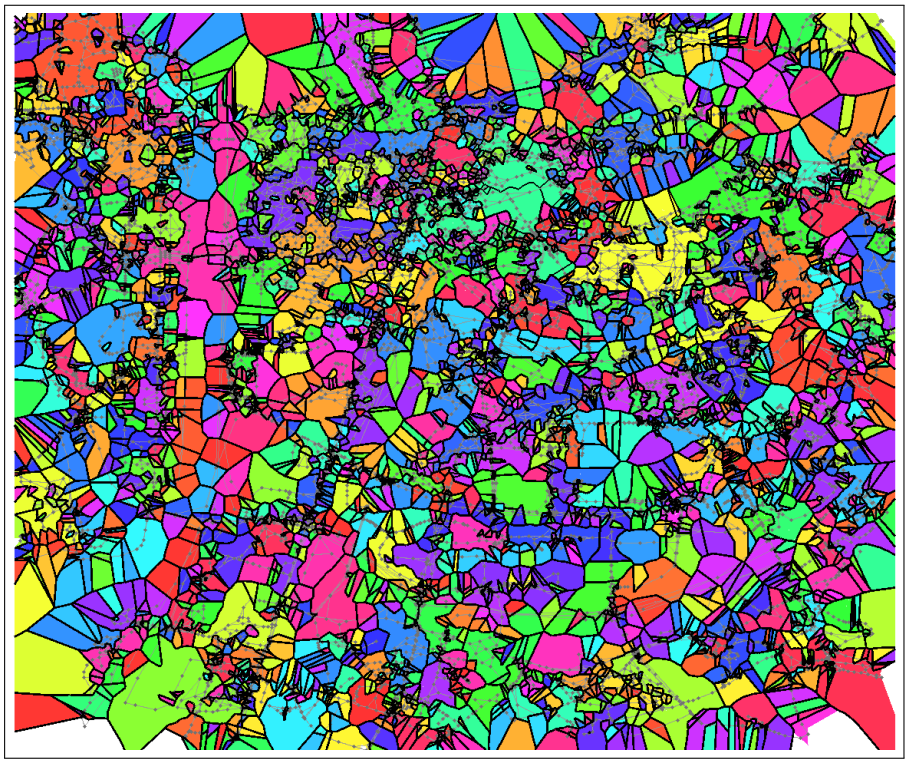
This simulation enlightens the capacity of the abstracting network to mimic the dynamical behavior of the initial network and in particular its equilibrium point. We note however that the nodes of the reduced network do not necessarily correspond to *sensitive* zones of the initial network that we would want to watch over. As an example, the large road crossing the north of Grenoble from west to east (more or less red in the initial network) is not well captured by the reduced system. That being said, one could add a constraint or an additional cost function to the problem in order to force the partition to capture these tactical points of the traffic network.
The simulation on the reduced network is about 5 times faster. Therefore, using the abstracting network to estimate the traffic condition in real-time or to compare different strategies of traffic control offers a good performance both with regard to the run-time and the accuracy.

[5]Let us note that, in reality, a lot of cars are parked in the city and starts from inside the network. However, we choose to have inflows only from the outside of the network to observe the progressive diffusion of the cars in the network

[6]An animation of the complete simulation is available here: `https://youtu.be/6wSF_kqFVL8`

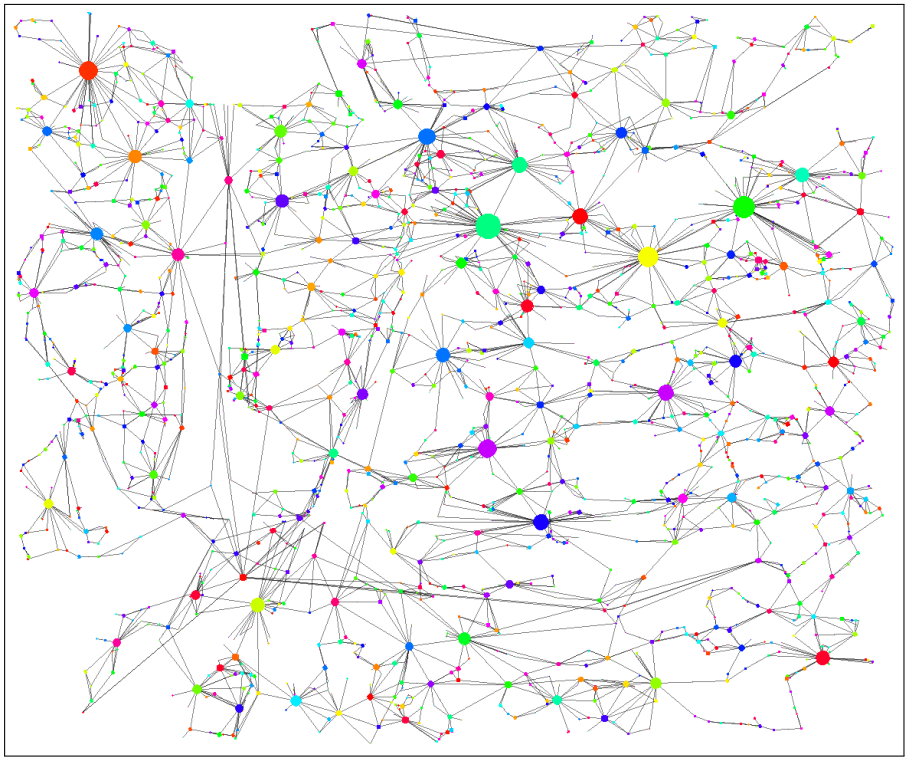


(a) Initial network

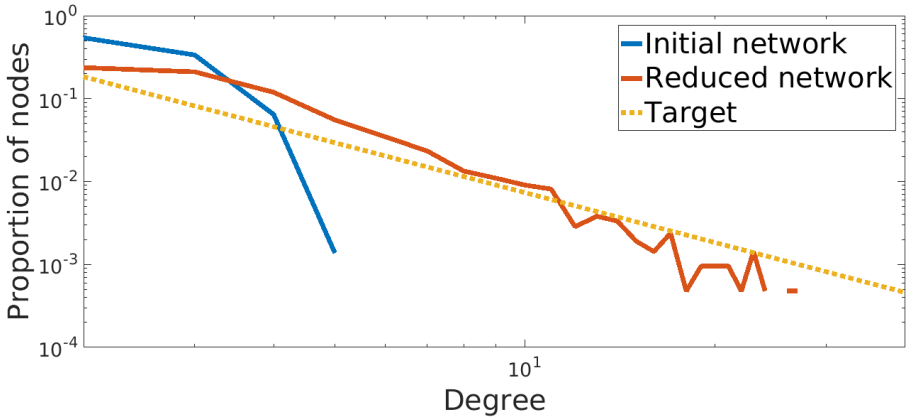


(b) Partition obtained through the algorithm. The color of each part matches the color of the nodes of the reduced network in fig. 4.5

Figure 4.4



(a) Reduced network: the size of the nodes represents their indegree and their color matches the color of the partition in fig 4.4.



(b) Log-log plot of the indegree distributions of the initial network, the reduced network and the scale-free target.

Figure 4.5

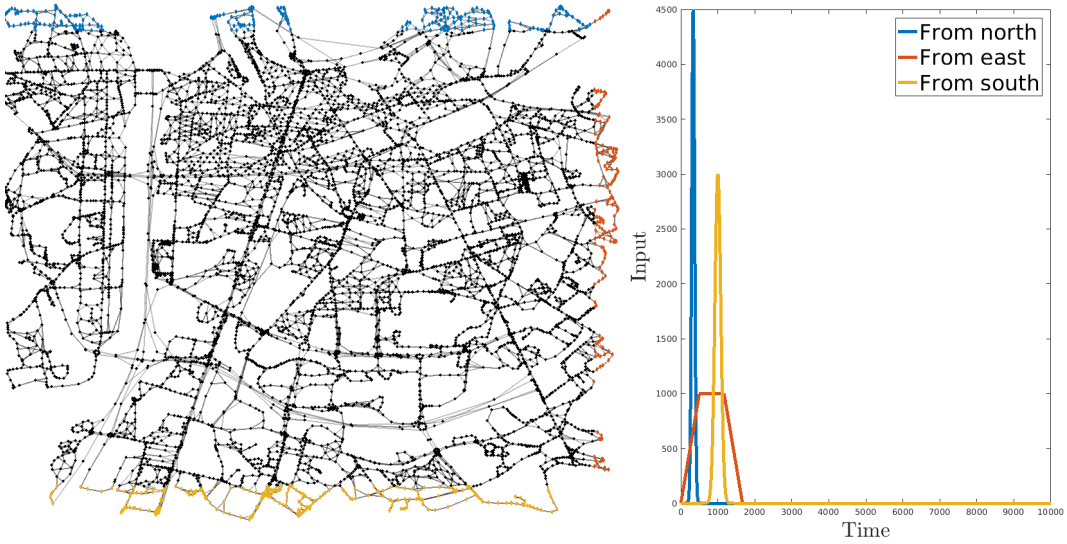
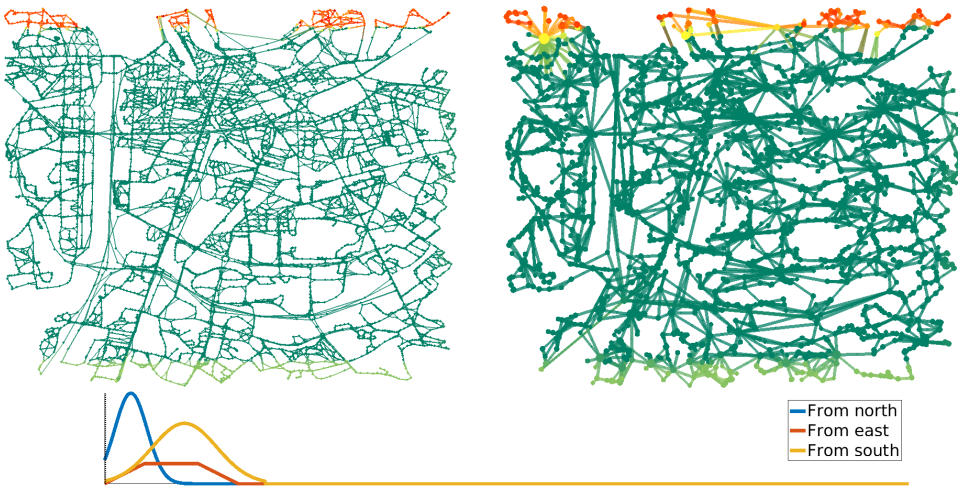
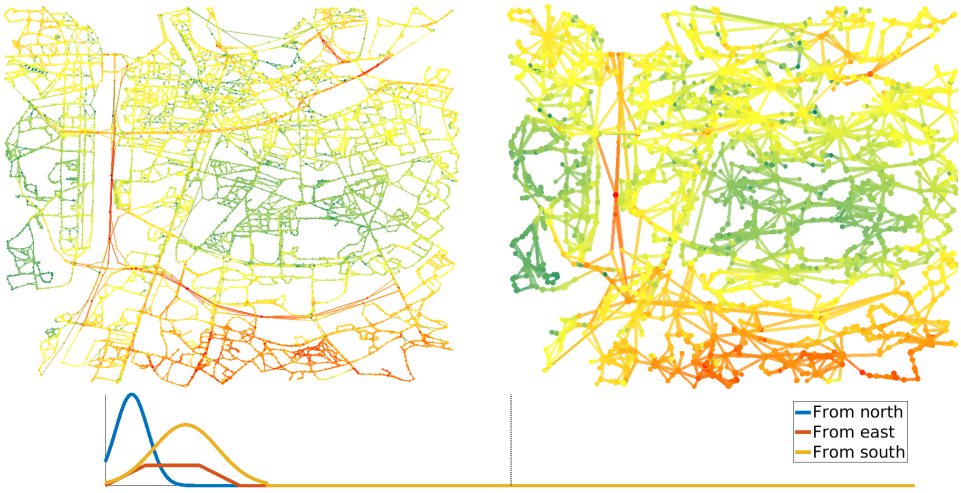


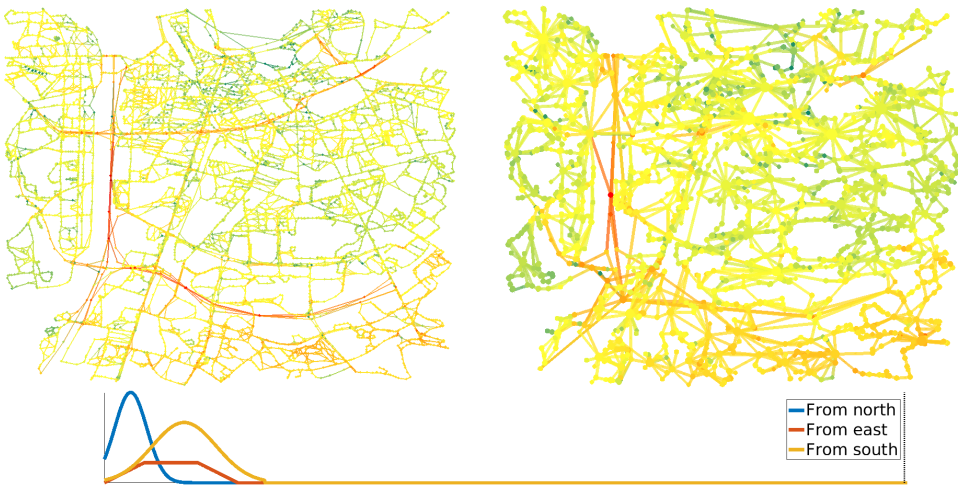
Figure 4.6: The Grenoble traffic network with the inflows coming from the borders



(a) $t = 1$ (0h)



(b) $t = 1000$ (8h20)



(c) $t = 2000$ (16h40)

Figure 4.7: Representation of the state of the initial (left) and reduced (right) networks at different time. On the bottom the dynamic of the inflows and the current time (the dotted line) are plotted. The initial networks possesses 13099 nodes while the reduced nodes possesses 2099 nodes. At $t = 2000$, we can consider that the steady-state is reached.

Chapter 5

Application to network epidemiology

> First, the terror. And then a moment of hunger. This is how the sense of taste disappears from our world. They don't even have time to give the disease a name.
>
> Susan, Perfect Sense

Contents

The theoretical development of Chapters 2 and 3 appears rather academic and disconnected from real-world problems. Within this chapter, we consider the study of diseases spreading over a population to present two applications of these results. In the first section, some preliminaries about network epidemiology are given which allow to introduce the models used in the applications. Section 5.2 present a cure-allocation strategy using a scale-free abstraction as presented in Chapter 2. This work has been presented at the NecSys conference 2018 conference in Groningen and published in [85]. Finally, Section 5.3 uses the result of Chapter 3 to reconstruct from few measurements an aggregated state of the evolution of an epidemic.

5.1 Introduction to network epidemiology

5.1.1 Overview of the models

History of network epidemiology

The mathematical modeling of epidemic spreading aims to evaluate the progression of an epidemic through a population and to design tools to control the propagation. The first work using mathematics to understand disease is due to Daniel Bernouilli in 1760 [11] when he aimed to study the reliability of the smallpox inoculation[1]. This work is considered as the first attempt of mathematical modeling of life and as a precursor of biomathematics. However it was not until 1897 that a first model of disease spreading was proposed by Ronald Ross in his study of malaria [110]. He obtained results on the processus of evolution of the epidemic and promoted the study of this new field. In particular, Gray McKendrick encouraged by Ross published from 1927 together with William Kermack three articles [**kermack1927contributions**] introducing a new class of models: the compartmental models. In these class of models, the population is divided into compartments having the same characteristics and the evolution of the size of the compartments is described. As we will see, the most common compartments used are Susceptible (S), Infected (I) and Recovered (R) leading, among others, to the two models: SIS and SIR. All along the twentieth century, works in this field increased and focused, in particular, on this compartmental models. A more complete review of the history and the models of mathematical epidemiology can be found in [34].
At the same time, the second part of the twentieth century saw the appearance of the network theory explaining a wide range of natural phenomenon. Limited until then by the hypothesis of homogeneity of the population, compartmental models benefited in the early 2000s from the development of network theory. Indeed, networks allowed to take into account the interactions among the population. This leads to the network epidemiology models that we will use in the two applications presented in this chapter. A review of the use of networks in epidemiology can be found in [100].

Compartmental models

In 1927, the groundbreaking theory of Kermack–McKendrick [67] introduced the notion of compartments in epidemiology. Their model divides the population into three compartments:

Susceptible (S): healthy individuals who can be infected

Infected (I): individuals suffering the disease

Recovered (R): individual immunized to the disease who can not be infected anymore

[1] also called *variolation*, it is somehow the ancestor of the smallpox vaccine.

A susceptible individual can pass into the infected compartment if he gets the disease and then in the recovered state if he is cured. Thus the evolution of a patient follows this scheme $S \longrightarrow I \longrightarrow R$. The proportion of the population represented by each compartment n_S, n_I and n_R (such that $n_S + n_I + n_R = 1$) is governed by a differential equation which depends on the parameters of the disease. Thus, in the Kermack–McKendrick model the evolution of individuals does not take into account their neighborhood and is restricted to the evolution of their compartment.
Yet, one can imagine that an individual which is in contact with a lot of people is more vulnerable to the disease than an isolated individual, which can play a role in the spreading of the epidemic. With the introduction of networks into compartment models, the evolution of the individuals depends also on the state of their neighborhood. Thus, for example, a susceptible individual will be more likely to be infected if he is surrounded by infected individuals. Figure 5.1 illustrates the difference between this two types of models.

Originally, the compartmental models included only the three compartments mentioned above. The models which use these compartments are commonly referred as SIR models. After that, ither compartments have been introduced, for example:

Exposed: individuals infected but not yet infectious

Carrier: individuals not suffering the disease but who can infect other

Quarantine: individuals infected but put in quarantine so they can not infect others

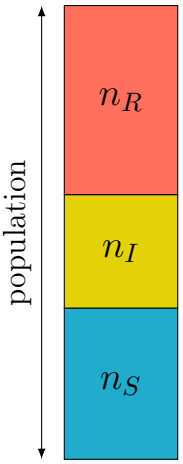
Considering these additional compartments allows to refine the accuracy of the model and is sometimes necessary to model certain epidemics. On the contrary, another commonly used model, the SIS model, considers only the Susceptible and Infected compartments. This model, used in the two applications presented in this chapter, is described in the next section.

5.1.2 The SIS model

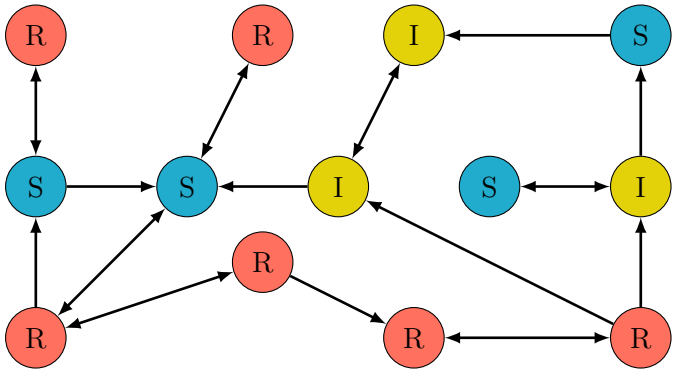
Classical model

As the name suggests, in the SIS model, a susceptible individual can pass into the infected compartment if he gets the disease and then comes back in the susceptible compartment if he is cured as illustrated in fig. 5.2. It differs from the SIR model in the state of the cured individual: in the SIS model a cured individual can be infected again, while in the SIR model a cured individual can not be infected again. This model is particularly relevant for some infections, as common cold or influenza, for which there is no immunization upon recovery. The transition from one compartment to the other is governed by two values: the infection rate β and the recovery rate δ (with $0 \leq \beta, \delta \leq 1$).

We denote $X_i(k)$ the state of individual i at time k: $X_i(k) = 0$ means that the individual i is susceptible and $X_i(k) = 1$ means that the individual i is infected. The exact equation of



(a) The model proposed by Kermack–McKendrick is the first one to propose a compartment system. In this model, the population is divided into three different compartments (susceptible, infected, recovered) whose size is governed by a differential equation. Other compartment models exist with different compartments and different dynamics.



(b) With the arrival of network theory, compartmental models can take into account the interaction between individuals. The equation of evolution depends on the model chosen and is not precised here.

Figure 5.1: Illustration of the compartments model in epidemiology

evolution that we consider here is as follows:

$$\begin{aligned} &\text{if } X_i(k) = 0, \quad X_i(k+1) = 1 \text{ with probability } \beta \frac{1}{|\mathcal{N}_{in}(i)|} \sum_{j \in \mathcal{N}_{in}(i)} X_j(k) \\ &\text{if } X_i(k) = 1, \quad X_i(k+1) = 0 \text{ with probability } \delta \end{aligned} \tag{5.1}$$

This can be read as follows: if an individual is susceptible at time k, the probability that he gets infected at time $k+1$ is the proportion of infected people within its neighbors scaled

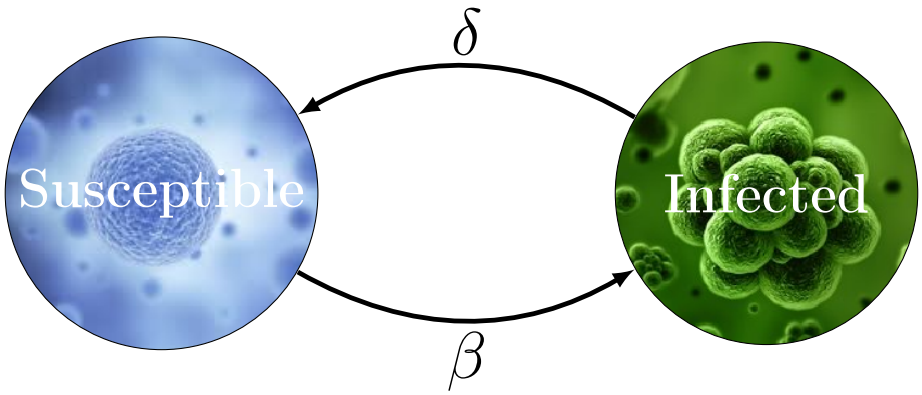


Figure 5.2: Sketch of the SIS model

by the infection rate β. In the other hand if an individual is infected at time k, the probability that he recovers at the time $k+1$ is the recovery rate δ. Thus, a disease with high β and low δ is particularly violent, while a disease with low β and high δ is benign.
After several time steps, the proportion of infected people converges towards a steady state called the prevalence [100]. If this convergence is well defined in model without network, or in deterministic network model, in our case, due to the stochastic process, the proportion of infected people fluctuate around the prevalence, denoted ρ. In the experiments presented in 5.2 this prevalence is computed as the average proportion of infected people over a window of time after the convergence:

$$\rho = \frac{1}{100} \sum_{k=1}^{100} \bar{X}(1000 + k) \tag{5.2}$$

where $\bar{X}(k)$ is the average value of the vector X at time k. This formulation of the prevalence has been chosen empirically to efficiently capture the prevalence observed. Let us note that the prevalence only depends on the structure of the network G and the spreading rate $\lambda = \frac{\beta}{\delta}$, but not on the initial subset of infected nodes (as long as this subset is non-empty) [102]. We will write the prevalence as $\rho(G, \lambda)$.
For the first application, in Section 5.2, we will consider Model (5.1) which is commonly used in network epidemiology [33, 101]. For the second application, in Section 5.3, we consider a slightly different version described in the next section.

Linearized mean-field approximation

We consider here a mean-field approximation of the SIS model introduced in [118]. In this case, the nodes of the network do not represent individuals but groups of people. The nodes are not in a fixed state S or I but have a proportion p of people infected. The dynamics of p is then:

$$\dot{p}(t) = (AB - \delta I)p(t) - PABp(t), \quad p(0) = p_0 \tag{5.3}$$

where $p = [p_1, \ldots, p_n]$ are the proportions of infected people in each group (and $P = \text{diag}(p_1, \ldots, p_n)$), I is the identity matrix, $B = \beta I$ and p_0 is the initial proportion of infected people in each group. In order to use the approach developed in Chapter 3, we consider a linearization around zero [100] of (5.3):

$$\dot{p}(t) = \underbrace{(AB - \delta I)}_{\mathcal{A}} p(t), \quad p(0) = p_0 \tag{5.4}$$

In order to include sources of infection that are external to the population, we define a matrix $\mathcal{B} \in \{0;1\}^{n \times b}$ to identify nodes that are in contact with sources of infection, and the function $u(t) \in \mathbb{R}^b$ to represent the temporal evolution of these sources of infection. By including these inputs u, the dynamics becomes

$$\dot{p}(t) = \mathcal{A}p(t) + \mathcal{B}u(t), \quad p(0) = p_0 \tag{5.5}$$

This leads to the matrices with the following form:

$$\mathcal{A} = \begin{pmatrix} -\delta & \beta & 0 & \beta & 0 & 0 & 0 \\ 0 & -\delta & 0 & 0 & \beta & \beta & 0 \\ \beta & 0 & -\delta & 0 & 0 & 0 & 0 \\ \beta & 0 & \beta & -\delta & 0 & \beta & 0 \\ 0 & 0 & 0 & 0 & -\delta & 0 & \beta \\ 0 & 0 & 0 & \beta & \beta & -\delta & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta & -\delta \end{pmatrix} \quad \text{and} \quad \mathcal{B} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{5.6}$$

These matrices correspond to the system illustrated in fig. 5.3. System (5.5) falls in the scope of positively uniform system with a large negative self-loop as discussed in Remark 3.1. Therefore, the Multiple Quasi-Regular Induced Subgraph detection can be applied.

Remark 5.1. Although we use in this article the terminology of epidemics, let us note that these models are also used in other contexts such as: spreading of computer viruses over web networks [7] or information spreading over social networks [72].

5.2 MergeToCure, scale-free abstraction for cure allocation strategy

5.2.1 Introduction

In network epidemiology, a common problem is to choose individuals to cure in order to reduce the spreading. As cured individuals can not be infected or infect other individuals they can be considered as removed from the network. Therefore, if the number of cures is limited, identifying individuals which removal will reduce the most the spreading is an interesting problem known as cure allocation[2]. Intuitively, the best choice is to remove nodes such that

[2]instead of the term *cure*, the terms *vaccine*, *treatment* or *quarantine* are also used. Indeed, all refer, in a certain way, to the removal of the node from the network.

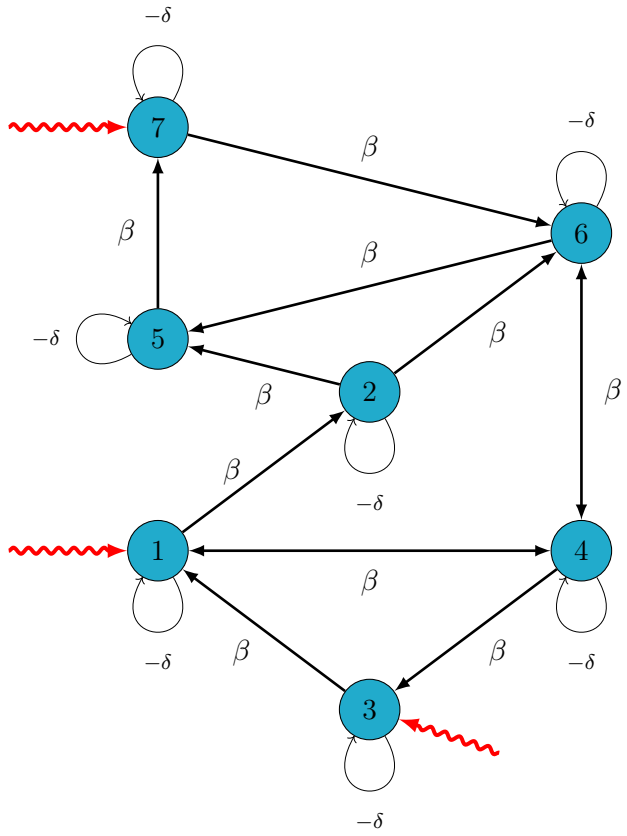


Figure 5.3: Example of a network corresponding to the SIS model in (5.6). The *snake* lines correspond to the inputs.

the network becomes less connected and so the epidemic can not spread easily. However, finding the best nodes to remove in order to decrease the connectedness of the network is a NP-complete problem [119]. Therefore, several papers propose heuristics to approximately solve this problem. Essentially it is proposed to remove the most *important* nodes: [55] proposes to remove the nodes with the highest betweenness centrality and [88] the nodes with the highest PageRank [88][3]. In [33], the problem is addressed in the context of scale-free networks. As discussed in Appendix 0.3.4, it is shown that for this type of networks allocating cures to the most connected nodes reduces efficiently the spreading. This strategy will be called here the *hub-removal strategy*. In homogeneous networks such as Manhattan-like grids, the absence of hubs reduces the efficiency of this strategy. In [108] the question of allocating cure in a grid network is addressed. However, in this work the proposed strategy requires to know the location of the seed[4] of the epidemic, which is unrealistic.
The main contributions of this section is the introduction of a strategy to allocate cure in Manhattan-like grids. This strategy uses a scale-free network abstracting the initial Manhattan-like grid. We show numerically that this strategy is better than other strategies

[3]These notions has been presented in Section 0.2.2
[4]patient zero, the first infected individual.

for a large range of the parameters.

Allocating cures in scale-free networks

Appendix 0.3.4 presents cure-allocation strategy in scale-free networks. Let us introduce some additionnal results. It is known [2] that in a grid or a random network there exists an epidemic threshold λ_c under which the prevalence is always zero:

$$\forall \lambda < \lambda_c, \quad \rho(G_{hom}, \lambda) = 0 \tag{5.7}$$

where G_{hom} is any *homogeneous* network (grid or random).
At the contrary, in scale-free network there is no such epidemic threshold [33]. Thus, if a population forms a scale-free network, even a minor infection will always persist. However, as discussed in Appendix 0.3.4 removing the hubs in a scale-free network allows to reduce drastically the epidemic. In particular, [33] shows that it is possible to restore such an epidemic threshold by removing the biggest hubs, and this threshold can even be made larger than in a homogeneous network by removing enough hubs. Precisely, if we consider a scale-free network build with the Barabási-Albert model, we have:

$$\lambda_c = \frac{k_0 - m}{k_0 m} \ln^{-1}\left(\frac{k_0}{m}\right) \tag{5.8}$$

where m is the number of edges added at each step in the Barabási-Albert model[5] and k_0 is the degree above which every node is removed.

However this strategy is not efficient if the degree distribution of the network is homogeneous. Similarly, the strategies recommending to remove nodes with the highest PageRank or the highest betweenness centrality lose their interest in a network where all the nodes are more or less similar. Therefore, in the next section we address the following problem: how to find an efficient strategy of cure allocation for homogeneous networks ? To this end, we present an approach inspired by the hub-removal strategy: by finding a scale-free abstraction of a homogeneous network we can highlight some zone-hubs which are more interesting to cure. To find this scale-free abstraction we use the algorithm MergeToScaleFree, presented in Chapter 2.

5.2.2 The strategy Merge To Cure

Description of the strategy

The idea of the strategy is to identify zones which play the role of hubs in the network by using the algorithm MergeToScaleFree, and then to focus on curing these zones.

The strategy can be detailed as follows:

[5] see Section 0.3.2

1. Extract a scale-free abstraction out of the original homogeneous network.
2. Identify the hubs in the scale-free abstraction, corresponding to zones-hubs in the initial network.
3. Assign a cure to the nodes *at the border* of the zone-hubs.

The interest of the hub removal is to cut the main routes of infection. Then, in the case of zones-hubs, it is sufficient to only remove the border of these zones. Figure 5.4 gives an illustration of the strategy.

Remark 5.2. An interesting link can be made with the so-called meta-population models. In these models, introduced in [5], nodes do not represent individuals but groups such as cities. In our case, if the initial network is a network of individuals, our strategy consists in finding how to regroup these individuals at a coarser scale and this way extract a meta-population model out of the initial model. As the meta-population network is designed to be scale-free we just have to identify the most connected groups of individuals to cure and therefore reduce the spreading of the epidemic.

Experimental results

In this section, we present the experimental setup to validate our strategy. Through this section we consider Manhattan-like grids as defined in 2.5.2. To test the efficiency of our method we consider a Manhattan-like grid G and we numerically compute its prevalence using (5.1). Then we remove nodes according to different strategies:

1. MergeToCure. We denote n_{cure} the number of nodes removed.
2. By curing randomly n_{cure} nodes
3. By curing the n_{cure} nodes with the highest degree: hub-removal strategy.
4. By curing the n_{cure} nodes with the highest betweenness centrality.
5. By curing the n_{cure} nodes with the highest PageRank.

Then the prevalence is computed for the networks obtained with each of these strategies. The prevalences are compared with the null case where no cure is allocated in the network. For each strategy σ we compute the benefit B_σ as:

$$B_\sigma(n_{cure}, \lambda) = \frac{\rho(G, \lambda) - \rho(G_\sigma, \lambda)}{\rho(G, \lambda)} \tag{5.9}$$

where G_σ is the network obtained with the strategy σ. This benefit measures how much the strategy σ reduce the initial prevalence of the network. Thus, the higher is B_σ the better is the strategy σ.
In the simulation presented in fig. 5.5, a tenth of the population is initially infected. In the

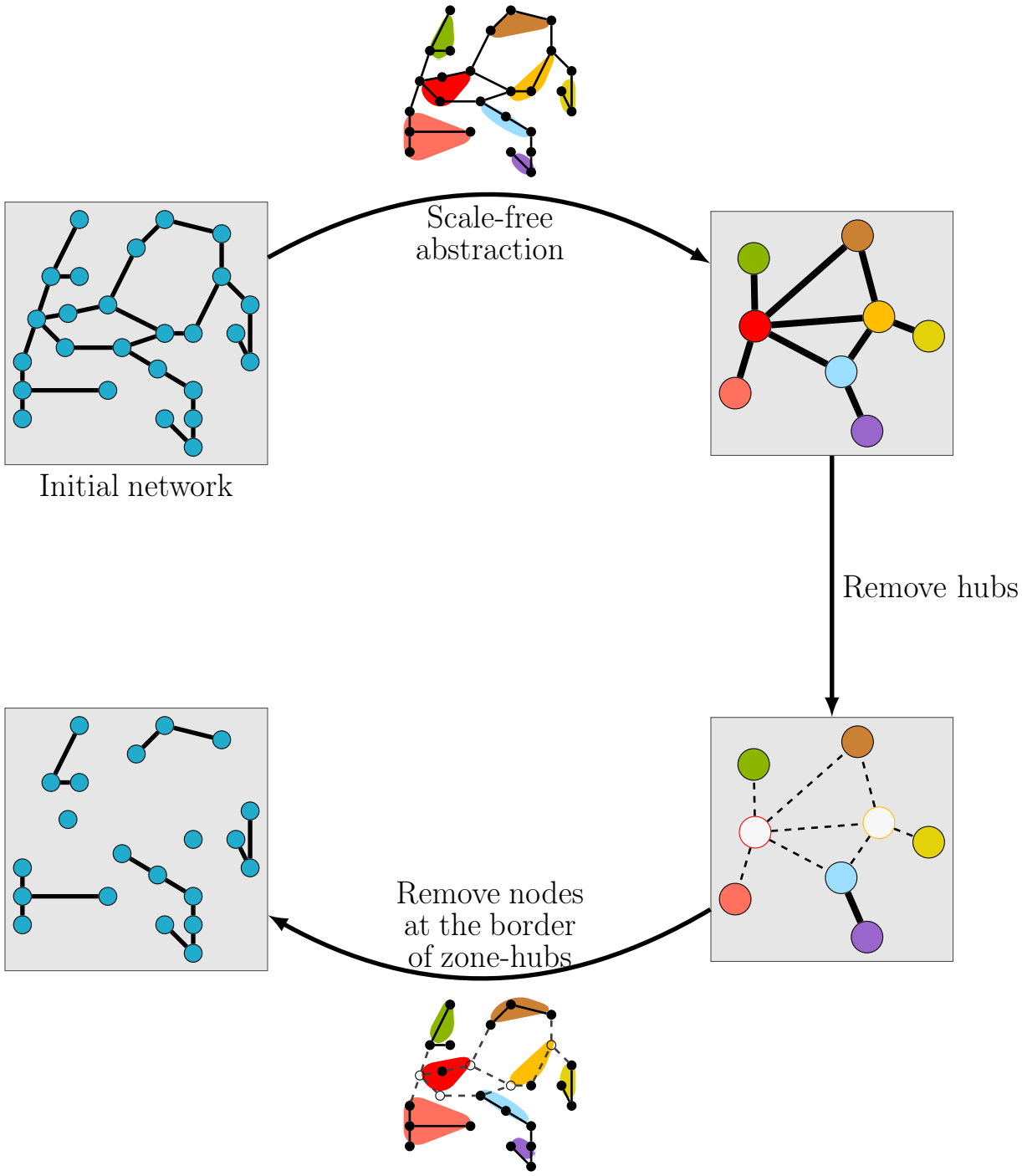


Figure 5.4: Illustration of the strategy MergeToCure to find a subset of nodes to cure in a Manhattan-like grid. For simplicity, we present here an undirected network, but the same process can be applied to a directed network.

four simulations we have removed, respectively, 1, 2, 3 and 4 zone-hubs in the MergeToCure strategy. For the other strategies we removed the same number of nodes as removed in the MergeToCure strategy. The percentage of nodes hence removed is specified for each result. The abstracting scale-free is generated with a scale-free coefficient $\alpha_{SF} = 2.8$. The value of the infection rate λ is varying between 0.4 and 4 with a step of 0.1. The results are averaged over 50 realizations with a different set of initially infected nodes. The considered network is a 60×60 Manhattan-like grid which contains 3287 nodes.

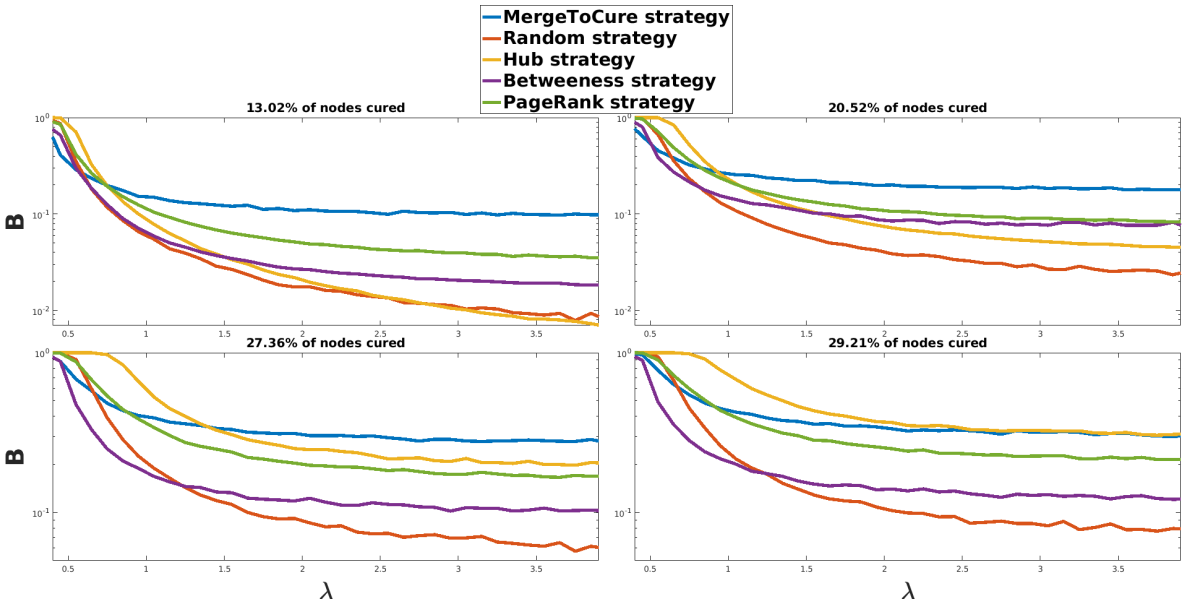


Figure 5.5: Averaged benefit brought by each strategy for different value of λ and N_{hub} in a 60×60 grid network. When $B = 1$ it means that $\rho(G_\sigma, \lambda) = 0$ which means that the strategy allows to completely eradicate the disease.

We observe that the MergeToCure strategy has a larger benefit than all other strategies when $\lambda \gtrsim 1$ and $n_{cure} \lesssim 25\%$. For a large proportion of removed nodes (larger than 30% of the population) the hub strategy becomes more efficient than the MergeToCure strategy. We suggest then, that, in a Manhattan-like grid, the MergeToCure strategy is the most efficient to reduce the prevalence for parameters: $(\lambda, n_{cure}) \in [1; +\infty) \times (0, 0.3N]$, where N is the number of nodes in the network. This means that our strategy is the best when the infection is particularly virulent and the number of available cure is very limited. To get an idea of the value of λ for real diseases, [38], for example, estimates this value for sexually transmissible diseases between $[0.76, 1.52]$.

In order to improve the efficiency of the MergeToCure strategy we also wonder how to choose the scale-free coefficient in the MergeToScalefree algorithm. Thus, we consider the benefit $B(n_{cure}, \lambda, \alpha_{SF})$ with 25% of node reduced and varying λ and α_{SF}. Figure 5.6 shows the results. The benefit tends to be higher for high value of α_{SF}, which is when the hubs are fewer but more connected.

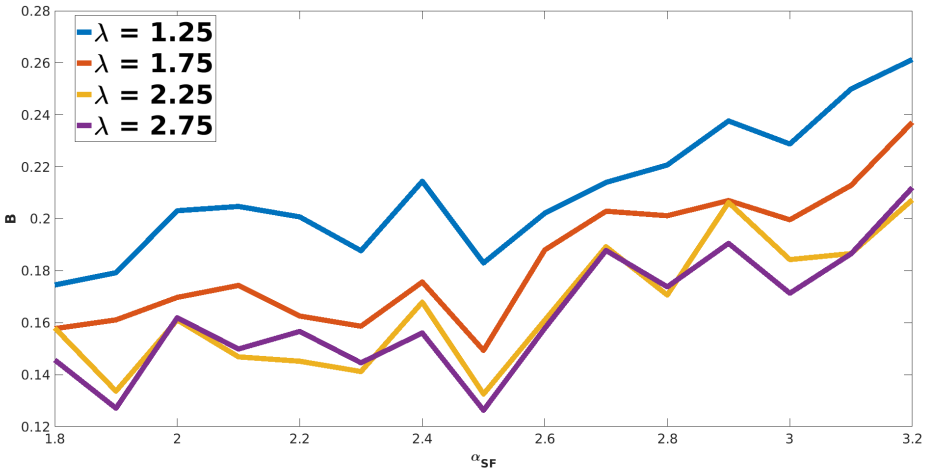


Figure 5.6: Benefit brought by the strategy MergeToCure as a function of the scale-free coefficient of the abstracting network.

5.3 Average detectability of an epidemic spreading

In this section we apply the mqRIS algorithm presented in Section **??** to a real-world case: the spreading of a disease over an interaction network of the main cities in France. By reconstructing the average state of different subgraphs, we aim to estimate the evolution of the proportion of infected people in different areas.

5.3.1 The estimation problem

It is clear how important it is to estimate the evolution of a disease, for instance in order to take appropriate sanitary measures or to study the efficiency of a treatment (fig. 5.7). As it is very costly to determine the state of each individual, one needs methods to reconstruct the spreading of the epidemic per areas and with relatively few measurements. We propose here to use our approach to estimate the evolution of the proportion of infected people in different areas, by measuring the state of few groups. In order to construct a concrete example, we shall consider the above linearized SIS dynamics over a contact network beween the main French cities.

5.3.2 Topology of the network

We consider a network of interactions between groups of individuals in the main cities of France. The network is structured at two different scales: one level within the cities and one level between the cities. It is known that at the level of a city, individuals are strongly interconnected and tend to form clusters [17, 114]. Here we use the Watts-Strogatz model [120] which is known to well capture the features of social networks. At the level of the country,

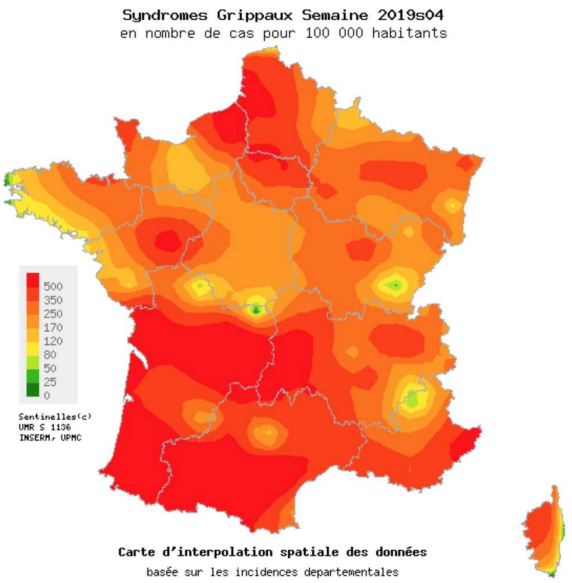


Figure 5.7: Interpolating map of the number of individual infected with flu for 100000 inhabitants in January 2019 in France. Map available on www.sentiweb.fr. ©Inserm

there are fewer connections between different cities and the number of connections between two cities is proportional to the numbers of their inhabitants and inversely correlated to their distance. The network is generated by considering twenty two of the most populated cities in France. The details of the topology of the network are given in table 5.1. Figure 5.8 shows the obtained network.

5.3.3 Simulations

Figure 5.9 presents the partition found with Algorithm **??**. The network has been divided in 11 subgraphs containing 1112 nodes in total. Thus, only 292 nodes remains to measure which represents only 20.80% of the nodes. Some parts fit cities while others include a whole region. The figure gives also the regularity error ϵ, the number of nodes $|\mathcal{V}|$ and the mean degree k for each subgraph.

Next, since we have the partition in which each induced subgraph is quasi-regular it is possible to estimate the value of the average inside each subgraph.
The parameters of the model are fixed as follows: $\beta = 0.05$ and $\delta = 0.98$. We randomly add 323 sources of infection randomly distributed in the territory. To be close to the reality we use as initial conditions the situation presented in fig. 5.7 which is available on *www.sentiweb.fr*. Figure 5.10 shows the evolution of the proportion of infected individuals inside each subgraph and the estimation made with the open-loop observer $\hat{x}_2^{av}$ described in (3.6). The solid lines are the actual averages while the dotted lines are the estimated averages. Figure 5.11 shows the evolution of the absolute error for each subgraph. We observe that, as expected, the estimation errors decrease quickly and remain relatively small. However, as we did not find

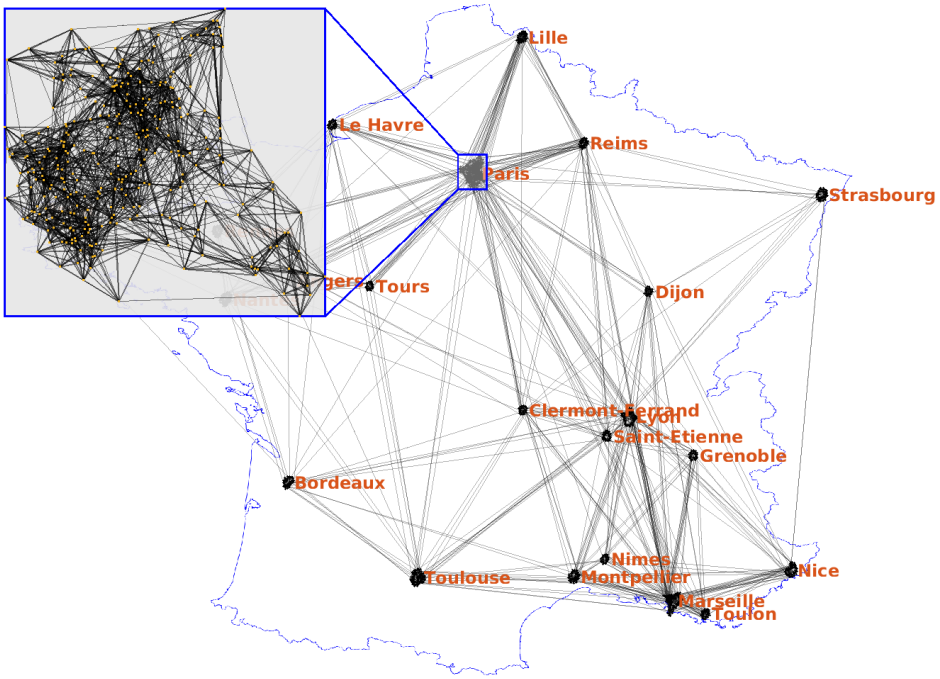


Figure 5.8: The network of interactions over the France territory. It is composed of 1404 nodes, each representing a population of 5000 individuals. The subgraphs within the cities are based on the Watts-Strogatz model while the network between the cities is a random network where the probability of connection between two nodes decreases exponentially with the distance.

Population	
Number of groups (nodes)	1404
Pop. per groups	5000
Number of cities	22
Network	
Model within cities	Watts-Strogatz
Mean degree K	10
Prob. rewire β	0.1
Prob. connection inter-cities	$\frac{e^{d^2/10}}{850}$
Number of inputs	323
SIS model	
Infection rate β	0.05
Recovery rate δ	0.98

Table 5.1: Parameters for the network of interactions and the SIS model

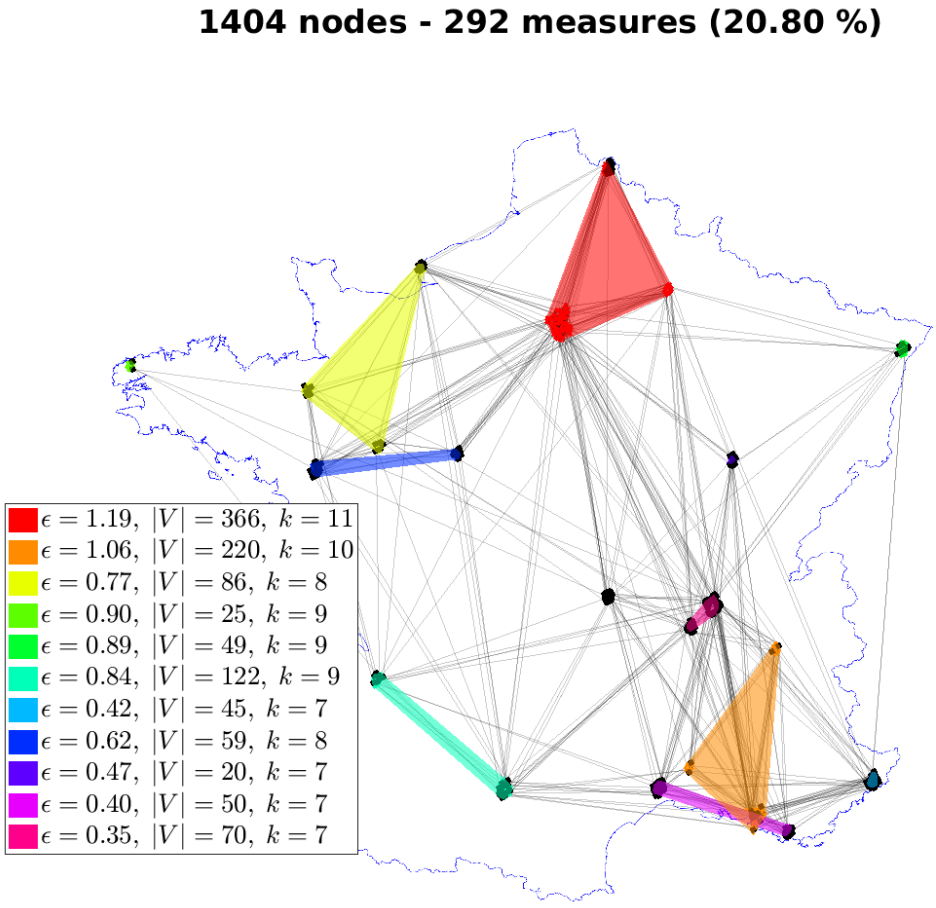


Figure 5.9: Partition obtained via the mqRIS algorithm. The legend gives the error of regularity, the number of nodes and the mean degree for each subgraph detected.

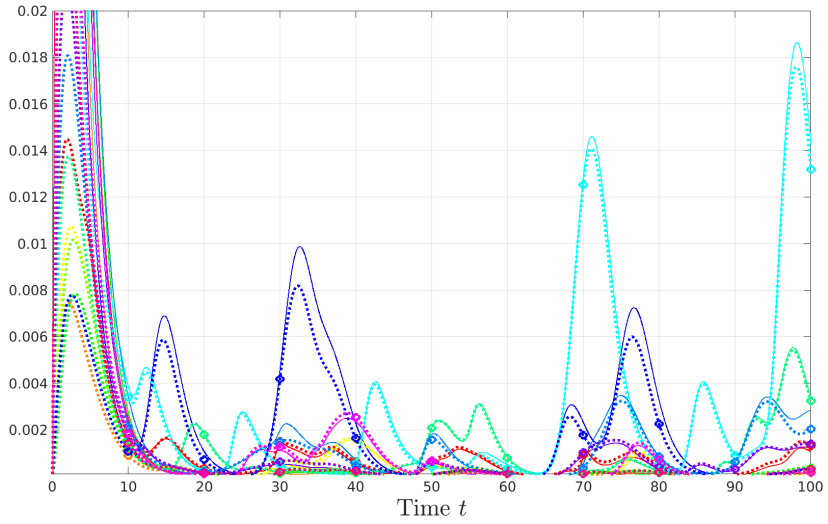


Figure 5.10: Proportion of infected individuals within each subgraph. The solid lines are the ground-truth values and the dotted lines are the estimated values.

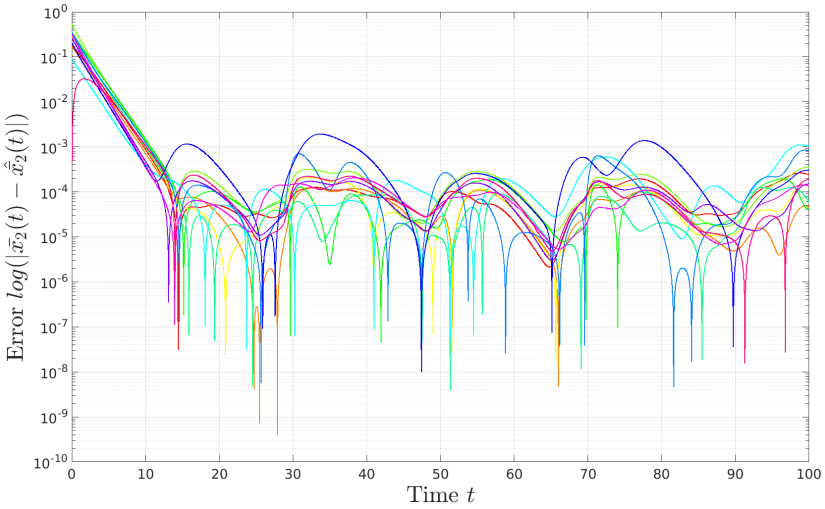


Figure 5.11: Semi-log representation of absolute errors between the ground-truth value and the estimated value for each subgraph. The errors are decreasing but they do not converge to zero.

exact regular subgraphs, the system is not detectable and the errors do not converge to zero.

5.4 Conclusion

In this chapter we applied two methods developed throughout this thesis to network epidemiology: the MergeToScaleFree algorithm and the Multiple Quasi-Regular Induced Subgraph detection.
In the first application, we explored the problem of allocating a limited number of cure in a network in order to reduce the spreading of an epidemic. It is known that in scale-free networks an efficient strategy consists in curing the hubs. Thus, thanks to the MergeToScaleFree algorithm presented in Chapter 2, we derived a scale-free abstracting network to identify some *zone-hubs* in a homogeneous network. We showed numerically that this strategy is better than all other strategies when the epidemic is particularly violent and the number of vaccine is very restricted.
In the second section, we explored the problem of estimating the evolution of an epidemic over a population with few measures. As shown in Chapter 3, it is possible to estimate the average of unmeasured nodes if the subgraphs which they form are close to regular. Based on this observation, we use the algorithm developed in Chapter 3 to identify such quasi-regular subgraphs. We identify thus the individuals to diagnose in order to know the average state of the unmeasured communities in the population.

Conclusion

> All models are wrong,
> but some are useful
>
> Georges Box

In this conclusion we will first draw up a list of the contributions presented throughout the thesis. Based on the limitations of these contributions we will present some possible extensions and opened questions that this thesis sparked.

Review of the contributions

Through this thesis we have raised and investigated several questions concerning network partitioning algorithms. We first wondered how a partitioning can be done regarding to the connectedness of the parts. Precisely, we estimated the loss due to the connectedness constraint. Following this theoretical question, we proposed two problems of network partitioning and we exhibited algorithms bringing solutions to these problems. In the first problem, we wanted the reduced network to have a scale-free structure while preserving somehow the characteristics of the initial network. In the second problem, we aimed to partition the network with a reconstruction purpose: we separated the measured nodes and the unmeasured nodes such that the average of the unmeasured nodes can be efficiently estimated. Following this three first chapters presenting theoretical results, we proposed applications divided into two chapters corresponding to the field of application: traffic and epidemiology. A detailed review of the contributions per chapter is given hereafter:

The price of connectedness: For any problem of optimal network partitioning, one may consider that the part have to be connected or not. Depending on the application, this constraint of connectedness may be necessary or only desirable. Therefore, a question emerges: how much this constraint of connectedness will damage the quality of the optimization. This damaging is what we call *the price of connectedness* and we aimed to estimate its value throughout the chapter. We first emphasized the link between this price of connectedness with another quantity: the ratio of connectedness which is the ratio between the number of connected partitions and the total number of partitions of a given network. We proposed then a tight upper bound on the ratio of connectedness which is the ratio between the number of partitions without isolated nodes and the total number of partitions. Finally, we exhibit an exact formula for this value and we show that it is indeed a good upper bound for the ratio of connectedness.

Network partitioning algorithm towards scale-free structure: In the introduction we have exhibited clues on the potential interest of a scale-free abstracting network. Thus,

this chapter was dedicated to the elaboration of a partitioning algorithm inducing a scale-free reduced network. We first introduced a general framework composed of a class of partitioning problems inducing scale-free networks and a meta-algorithm to solve such problems. We treated then two problems corresponding to this class of problem. The first problem had more a didactic purpose whereas the second one required an important mathematical development. Indeed, in this second problem, we want to preserve three properties through the partition: the eigenvector centrality[6] of the system up to a projection, the mass conservation[7] and the total mass[8]. We show then how to recompute the weights of the reduced network such that these properties are preserved for the particular type of partition used in the algorithm. Finally, we proposed some results on the algorithm and a simulation. Note that the framework developed in this chapter is reused in the first application to epidemic which take advantage of the scale-freeness of the reduced network.

Network partitioning algorithm towards average detectability: Noticing that it is often useless and difficult to reconstruct exactly each state of a large-scale network, it seems relevant to reconstruct only an aggregated state. Yet, precisely, the recently introduced notion *average detectability* refers to systems in which it is possible to reconstruct the average state of the unobserved nodes (in open loop). In this chapter, we proposed a method to determine which nodes have to be measured in order to make the system average detectable. Precisely, we show that for a certain type of systems, the *regularity* of the unmeasured subgraph implies the average detectability of the system. We reuse then the existing literature on regular subgraph detection to build an algorithm inducing average detectable system. As the hypothesis on the initial system were restrictive and the results quite poor, we relaxed the strict notion of average detectability to allow errors in the regularity of the unmeasured subgraph. We show that, in this case, the error of reconstruction, is upper bounded by a function of the error of regularity. Therefore, we proposed an algorithm detecting *quasi-regular* subgraph. Finally, we generalized the method to consider several different subgraphs for which we estimate separately the average. This approach is reused in the last chapter to estimate the evolution of an epidemic.

Application to traffic: The second algorithm developed in Chapter 2 is particularly interesting for traffic network. Thus, we applied this algorithm to the traffic network of the city-center of Grenoble. First, we introduced the model considered: the geographical network and the estimation of the traffic flow. Then, a scale-free network abstracting this network is obtained through the algorithm. Finally, we simulate a scenario over the two networks and show that the reduced network can efficiently be used to analyze the evolution of the traffic conditions.

Application to network epidemiology: The last chapter is divided into two different ap-

[6] this measure corresponds to the steady state of the network excited by a linear and stable discrete-time system.

[7] if a network have this property, the sum of the weights coming in a node equals the sum of the weights coming out of this node.

[8] the sum of all the weights in the network

plications using the same models of network epidemiology. Therefore, in a first part, we introduced network epidemiology theory and the different models used. Then the two applications are presented:

MergeToCure, scale-free abstraction for cure allocation strategy We proposed in this section a strategy for cure allocation[9] for networks with a homogeneous degree distribution: we partition the initial network and we use the scale-free abstraction to identify some *zone-hubs*. Then, the individual at the border of theses zone are cured. We numerically compared this method with others cure-allocation strategies, using a SIS model. We showed that our method is the most efficient to reduce the propagation of the disease when the number of cure is very limited and when the epidemics is particularly virulent.

Average detectability of an epidemic spreading: When considering the spreading of a disease, it is not possible to know precisely the state of each individual. In this section, we applied the method proposed in Chapter 3 to determine the individuals which state have to be measured in order to reconstruct an aggregation of the states. By measuring few nodes, one can estimate the average state in different parts of the population. This approach has been applied on a large-scale network that we have generated: an interaction network over the french territory.

Possible extensions

The results presented in this thesis are at some point limited by the hypothesis or the models considered. However, since these limitations are not locked doors but opportunities to extend our work, we will highlight some questions raised and potential future work for each chapter:

The price of connectedness: The formula (1.23) for the upper bound on the ratio of connectedness is a recursive formula and can not be used for large value of the parameters. Thus, it could be very useful, and surely feasible, to rewrite this formula in another way making it more easily computable.
While the tightness of the upper bound has been enlightened through simulations, this work would benefit from a theoretical result quantifying somehow this tightness and/or showing that the two quantities converge towards each other.
Finally, if the estimation of the ratio of connectedness is a quite solid result, the link with the price of connectedness is weaker. A better understanding of the link between these two quantities, possibly regarding to the partitioning problem considered, would enhance this work.

Network partitioning algorithm towards scale-free structure: The framework developed is quite efficient to obtain a reduced network with a scale-free structure. However, we can wonder if the scale-free cost function proposed is relevant. Maybe a cost-function

[9] In network epidemiology, cure allocation strategies aim to find the nodes to remove from the network (by curing them) in order to reduce the propagation of a disease.

focusing more on the right part of the distribution could be better if only the distribution of hubs matter.

The framework proposed includes a large variety of problems. An investigation could be lead to understand in which contexts and to what extent the scale-freeness of the reduced network can be used, and the constraints that have to be added to the problem in each cases.

As discussed in the chapter, the computation of the weights satisfying the constraints can be generalized to any partition (and not only merging). Therefore, an efficient way to find the best partition inducing a scale-free network would be more efficient than looking iteratively for the best merging.

Network partitioning algorithm towards average detectability: Several limitations for the strict case (RIS) have been presented within the chapter leading to the relaxed case (qRIS and mqRIS). These limitations was not only about the solutions but also about the problem itself so that we do not expect any extension for the RIS case. In contrast, for the relaxed problem, it seems that the link between the error of regularity and the error of reconstruction could be better understood. Indeed, the relation that we proposed includes a constant of the system making it less quantifiable. Throughout the chapter, we proposed to identify the nodes to measure in order to reach average detectability. A same approach could be used with the notion of average observability also introduced in [96], or in order to identify nodes to control to steer the average state as proposed in [98].

Application to traffic: The model used for traffic is quite unrealistic as it considers a linear dynamic. While the theoretical results are valid only for such a linear dynamic, one can wonder if the estimations made by the reduced network are still relevant with a more realistic traffic model implying non-linearity.

The value of the flow on the network are estimated from partial data on the roads. With real flow information the simulation would be more interesting. In the frame of the ERC Scale-Freeback project, the collect of traffic data is precisely one of the objectives and will be integrated to the GTL-ville platform[10].

Finally, one can wonder to what extent the scale-freeness of the reduced network is useful in the context of traffic modeling. It seems that a first answer could be brought by the boundary control theory as shown in [98], where the use of scale-free network is favored.

Application to network epidemiology: As for the application to traffic, the application here suffers from the limitation of the models. In this case, one of the main limitation of the network model of epidemiology is that it is impossible to know with precision the exact interconnection within a population. Therefore, it could be interesting to investigate, how the results of this section can be generalized to other models. More precisely for each of the application:

MergeToCure, scale-free abstraction for cure allocation strategy In this sec-

[10]Following the GTL project [122] who aimed to monitor the traffic condition on a single road in the south of Grenoble, the GTL-ville aims to monitor the traffic condition inside the whole city of Grenoble. See *http://gtlville.inrialpes.fr/*.

tion, we proposed a cure assignation strategy for networks with a homogeneous degree distribution. However, one can wonder if some interaction networks (or computer networks) are actually homogeneous. At least [73] shows that for sexual transmitting diseases, transmission networks have a scale-free structure.
As said before, the misunderstood of the exact structure of the network prevents from identifying the zone-hubs and, even more, the individuals at the border of these zones. However, one could investigate, how to identify individuals who probably play this role with partial information on the network, and how much this inaccuracy would impact the efficiency of the strategy.
Finally, it is obvious that choosing the individuals to cure implies some ethical issues. Most of the time, the resources allow to cure every individuals without distinction, making this question only academic. However, in some cases, such as for computer network, this strategy may be applicable.

Average detectability of an epidemic spreading: Above the limitations due to the model, the fact that the subgraphs detected do not necessarily correspond to city or small area may be an inconvenient. However, a simple way to prevent this issue would be to add a cost function in the algorithm penalizing the subgraphs scattered.
Moreover, this work would be more relevant if it would have been compared to other estimation methods specially with regard to the number of measurement, the quality of the reconstruction, and the running time of the algorithm.

Hopefully, the possible extensions presented in this section may reach someone and the work presented throughout this thesis will be carried a little bit further until it reaches any practical usage. At least, all this work will benefit to myself. In a minor part, for all the knowledge I discovered through these years. And in a major part, for all the personal skills I had to develop to achieve such a piece of work. Resilience being probably the most important of them.

Appendix A

Algorithms for clustered model reduction

We present here the algorithms used to solve problems (1.4) and (1.7). A deep description of the algorithm would require the introduction of several definition and notions which are out of the scope of this thesis, we present only some insights to understand its principle and especially how it has been adapted to solve the constrained problem (1.7). The detailed algorithm can be found in [58]. The algorithm is an iterative greedy algorithm which means that at each step the *best* pair of nodes is selected and merged. The metric used to quantify the value of a pair of node, denoted by δ in the algorithm, of node is not given here. At each step only the number of nodes in the network is reduced by one, and then the same procedure is applied $(n-\hat{n})$ times where n is the size of the initial graph and $\hat{n}$ is the desired size of the final graph. By merging iteratively nodes, we end up with a partition of the initial network. In the original algorithm, any pair of nodes may be merged at each iterations. Thus, the final partition may be disconnected. This is described in Algorithm 9 hereafter and is taken from [58]. In the adaption we proposed, described in Algorithm 10, only the connected nodes can be merged at each step. Thus, the final partition is necessarily connected. In the algorithms, $\mathcal{V}_k$ and $\mathcal{E}_k$ correspond respectively to the set of nodes and the set of edges of the network after k iterations.

Algorithm 9 Original Cluster Model Reduction

Input: Network G and desired final size $\hat{n}$
Output: Final partition I

1: Initialize the partition as $I \leftarrow \{I_1, \dots, I_N\} = \{\{1\}, \dots, \{N\}\}$
2: **for** $k = 1 : (n - \hat{n})$ **do**
3: **for** all pair $(i, j) \in \mathcal{V}_k \times \mathcal{V}_k$ **do**
4: Assign a reducibility value to the pair $(I_i, I_j) := \delta_{i,j}$
5: **end for**
6: Merge the parts minimizing the reducibility value, which is

$$(I_a, I_b) := \operatorname{argmin} \delta(I_i, I_j), \quad I_a \leftarrow I_a \cup I_b, \quad I \leftarrow I \backslash I_b \tag{A.1}$$

7: **end for**

Other computations are done after this algorithm to deduce the weight in P but there are identical for the two problems. For our concern we are only interested in this part of the reduction method. Note that these two algorithms are suboptimal.

Algorithm 10 Cluster Model Reduction inducing connected partition

Input: Network G and desired final size $\hat{n}$

Output: Final partition I

1: Initialize the partition as $I \leftarrow \{I_1, \ldots, I_N\} = \{\{1\}, \ldots, \{N\}\}$
2: **for** $k = 1 : (n - \hat{n})$ **do**
3: **for** all pair $(i, j) \in \mathcal{E}_k$ **do**
4: Assign a reducibility value to the pair $(I_i, I_j) := \delta_{i,j}$
5: **end for**
6: Merge the parts minimizing the reducibility value, which is

$$(I_a, I_b) := \operatorname{argmin} \delta(I_i, I_j), \quad I_a \leftarrow I_a \cup I_b, \quad I \leftarrow I \backslash I_b \tag{A.2}$$

7: **end for**

Appendix B

Weight projection to ensure the mass conservation property

Given a network $G = (A, \mathcal{V}, \mathcal{E})$, the edge adjacency matrix, denoted by A^e, is a matrix of size $n \times m$, where n is the number of nodes and m is the number of edges in the matrix and define as follows:

$$A^e_{ij} = \begin{cases} 1 & \text{if } \mathcal{E}_j = (\,\cdot\,, i) \\ -1 & \text{if } \mathcal{E}_j = (i, \,\cdot\,) \\ 0 & \text{else} \end{cases} \tag{B.1}$$

The vector of weights $W \in m$ is defined as:

$$W = (a_{i_1,j_1}, a_{i_2,j_2}, \ldots, a_{i_m,j_m})^\top, \quad \text{where } \mathcal{E} = ((i_1, j_1), (i_2, j_2), (i_m, j_m)) \tag{B.2}$$

If the network is a flow network then A^e and W verify

$$A^e W = \mathbf{0}_n \tag{B.3}$$

Therefore, we want the vector of weights to belong to the null space of A^e. Thus, considering a vector W_0, we obtain a vector W_1 which ensure the mass conservation property by projecting W_0 on $\mathrm{Ker}(A^e)$. The projection is orthogonal in order to minimize the value of $\|W_1 - W_0\|_2$.

Example B.1. Consider the network depicted in fig. B.1.

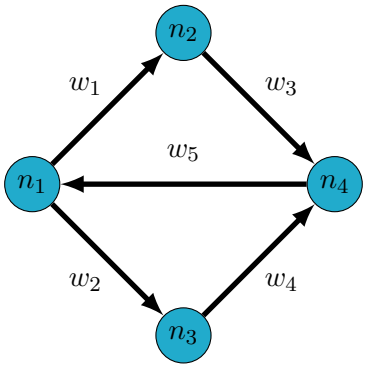
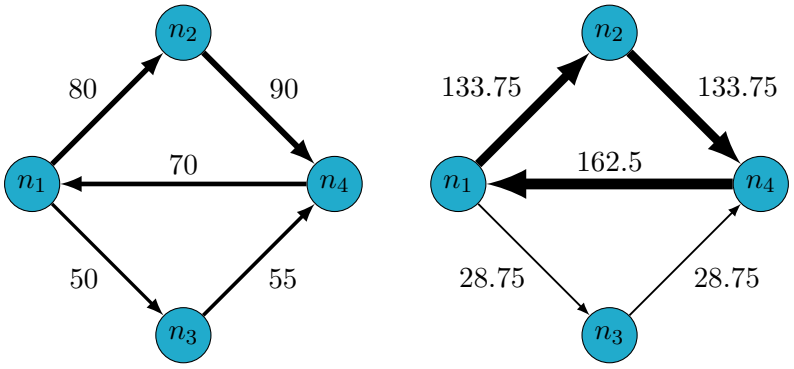


Figure B.1: Example of network corresponding to the matrices in (B.4)



(a) The initial estimation of the flow. The thickness of the edges represents the flow.

(b) The flow after projection on the space ensuring a flow network. The thickness of the edges represents the flow.

Figure B.2: The projection of the set of weight on the null space of the edge adjacency matrix ensures the network to be a flow network and to be as close as possible from the initial weights.

The edge adjacency and the adjacency matrices are as follows:

$$A = \begin{pmatrix} 0 & w_1 & w_2 & 0 \\ 0 & 0 & 0 & w_3 \\ 0 & 0 & 0 & w_4 \\ w_5 & 0 & 0 & 0 \end{pmatrix}, \qquad A^e \begin{pmatrix} -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 1 & -1 \end{pmatrix} \tag{B.4}$$

To ensure the network to be a flow network we want:

$$\begin{cases} w_5 - w_1 - w_2 &= 0 \\ w_1 - w_3 &= 0 \\ w_2 - w_4 &= 0 \\ w_3 + w_4 - w_5 &= 0 \end{cases} \tag{B.5}$$

Which corresponds well to $A^e[w_1, w_2, w_3, w_4, w_5]^\top = \mathbf{0}_5$.
In a first estimate we consider that the weights are $W_0 = [80, 50, 90, 55, 70]^\top$. This set of weights does not induce a flow network. By computing the orthogonal projection of W_0 on $\mathrm{Ker}(A^e)$ we find, a vector W_1 of weights inducing a flow network and minimizing the difference $\|W_0 - W_1\|_2$. In this case, the vector is as follows $W_1 = [133.75, 28.75, 133.75, 28.75, 162.5]^\top$. Figure B.2 illustrates this transformation of the weights.

Bibliography

[1] Agostinho Agra et al. "The k-regular induced subgraph problem". In: *Discrete Applied Mathematics* 222 (2017), pp. 14–30 (cit. on p. 91).

[2] Roy M Anderson, Robert M May, and B Anderson. *Infectious diseases of humans: dynamics and control*. Vol. 28. Wiley Online Library, 1992 (cit. on p. 124).

[3] Yuichi Asahiro, Hiroshi Eto, and Eiji Miyano. "Inapproximability of Maximum r-Regular Induced Connected Subgraph Problems". In: *IEICE Transactions on Information and Systems* E96.D.3 (2013), pp. 443–449 (cit. on p. 91).

[4] Yuichi Asahiro et al. "Complexity of finding maximum regular induced subgraphs with prescribed degree". In: *Theoretical Computer Science* 550 (2014), pp. 21–35 (cit. on p. 91).

[5] Norman TJ Bailey. "Macro-modelling and prediction of epidemic spread at community level". In: *Mathematical Modelling* 7.5-8 (1986), pp. 689–717 (cit. on p. 125).

[6] Norman TJ Bailey et al. *The mathematical theory of infectious diseases and its applications*. Charles Griffin & Company Ltd, 5a Crendon Street, High Wycombe, Bucks HP13 6LE., 1975 (cit. on p. 23).

[7] Justin Balthrop et al. "Technological networks and the spread of computer viruses". In: *Science* 304.5670 (2004), pp. 527–529 (cit. on p. 122).

[8] AL Barabási. "Love is all you need: Clauset's fruitless search for scale-free networks". In: *Blog post available at https://www. barabasilab. com/post/love-is-all-you-need* (2018) (cit. on p. 20).

[9] Albert-László Barabási and Réka Albert. "Emergence of scaling in random networks". In: *science* 286.5439 (1999), pp. 509–512 (cit. on p. 19).

[10] Albert-László Barabási and Eric Bonabeau. "Scale-free networks". In: *Scientific american* 288.5 (2003), pp. 60–69 (cit. on p. 20).

[11] Daniel Bernoulli. "Essai d'une nouvelle analyse de la mortalité causée par la petite vérole, et des avantages de l'inoculation pour la prévenir". In: *Histoire de l'Acad., Roy. Sci.(Paris) avec Mem* (1760), pp. 1–45 (cit. on p. 118).

[12] Marián Boguná and Dmitri Krioukov. "Navigating ultrasmall worlds in ultrashort time". In: *Physical review letters* 102.5 (2009), p. 058701 (cit. on p. 23).

[13] Marian Boguna, Dmitri Krioukov, and Kimberly C Claffy. "Navigability of complex networks". In: *Nature Physics* 5.1 (2009), p. 74 (cit. on p. 23).

[14] Marián Boguná, Fragkiskos Papadopoulos, and Dmitri Krioukov. "Sustaining the internet with hyperbolic mapping". In: *Nature communications* 1 (2010), p. 62 (cit. on p. 54).

[15] Marián Boguná, Romualdo Pastor-Satorras, and Alessandro Vespignani. "Absence of epidemic threshold in scale-free networks with degree correlations". In: *Physical review letters* 90.2 (2003), p. 028701 (cit. on p. 23).

[16] Marián Boguná and M Ángeles Serrano. "Generalized percolation in random directed networks". In: *Physical Review E* 72.1 (2005), p. 016106 (cit. on p. 45).

[17] Marián Boguná et al. "Models of social networks based on social distance attachment". In: *Physical review E* 70.5 (2004), p. 056122 (cit. on p. 128).

[18] Immanuel M Bomze et al. "The maximum clique problem". In: *Handbook of combinatorial optimization*. Springer, 1999, pp. 1–74 (cit. on p. 84).

[19] Anna D Broido and Aaron Clauset. "Scale-free networks are rare". In: *Nature communications* 10.1 (2019), p. 1017 (cit. on p. 19).

[20] Domingos M Cardoso, Marcin Kamiński, and Vadim Lozin. "Maximum k-regular induced subgraphs". In: *Journal of Combinatorial Optimization* 14.4 (2007), pp. 455–463 (cit. on p. 91).

[21] Giacomo Casadei, Carlos Canudas-de Wit, and Sandro Zampieri. "Scale Free Controllability of Large-Scale Networks: an Output Controllability Approach". In: 2018 (cit. on pp. 29, 30).

[22] Xiaodong Cheng, Yu Kawano, and Jacquelien MA Scherpen. "Model Reduction of Multiagent Systems Using Dissimilarity-Based Clustering". In: *IEEE Transactions on Automatic Control* 64.4 (2018), pp. 1663–1670 (cit. on p. 37).

[23] Xiaodong Cheng, Yu Kawano, and Jacquelien MA Scherpen. "Reduction of second-order network systems with structure preservation". In: *IEEE Transactions on Automatic Control* 62.10 (2017), pp. 5026–5038 (cit. on p. 37).

[24] Yun Chi et al. "Frequent subtree mining–an overview". In: *Fundamenta Informaticae* 66.1-2 (2005), pp. 161–198 (cit. on p. 84).

[25] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. "Power-law distributions in empirical data". In: *SIAM review* 51.4 (2009), pp. 661–703 (cit. on p. 19).

[26] Reuven Cohen and Shlomo Havlin. "Scale-free networks are ultrasmall". In: *Physical review letters* 90.5 (2003), p. 058701 (cit. on p. 22).

[27] Reuven Cohen, Shlomo Havlin, and Daniel Ben-Avraham. "Structural properties of scale-free networks". In: *Handbook of graphs and networks* (2003), p. 85 (cit. on p. 54).

[28] Reuven Cohen et al. "Breakdown of the Internet under Intentional Attack". In: *Phys. Rev. Lett.* 86 (16 2001), pp. 3682–3685 (cit. on p. 23).

[29] Luciano da Fontoura Costa et al. "Analyzing and modeling real-world phenomena with complex networks: a survey of applications". In: *Advances in Physics* 60.3 (2011), pp. 329–412 (cit. on p. 3).

[30] Noah J Cowan et al. "Nodal dynamics, not degree distributions, determine the structural controllability of complex networks". In: *PloS one* 7.6 (2012), e38398 (cit. on p. 29).

[31] Chavdar Dangalchev. "Generation models for scale-free networks". In: *Physica A: Statistical Mechanics and its Applications* 338.3-4 (2004), pp. 659–671 (cit. on p. 20).

[32] Leon Danon et al. "Comparing community structure identification". In: *Journal of Statistical Mechanics: Theory and Experiment* 2005.09 (2005), P09008 (cit. on p. 77).

[33] Zoltán Dezső and Albert-László Barabási. "Halting viruses in scale-free networks". In: *Physical Review E* 65.5 (2002), p. 055103 (cit. on pp. 121, 123, 124).

[34] Odo Diekmann, Hans Heesterbeek, and Tom Britton. *Mathematical tools for understanding infectious disease dynamics*. Vol. 7. Princeton University Press, 2012 (cit. on p. 118).

[35] Edsger W Dijkstra. "A note on two problems in connexion with graphs". In: *Numerische mathematik* 1.1 (1959), pp. 269–271 (cit. on p. 23).

[36] Mohammadreza Doostmohammadian and Usman A Khan. "On the controllability of clustered Scale-Free networks". In: *arXiv preprint arXiv:1905.01630* (2019) (cit. on p. 29).

[37] Sergey N Dorogovtsev, José Fernando F Mendes, and Alexander N Samukhin. "Structure of growing networks with preferential linking". In: *Physical review letters* 85.21 (2000), p. 4633 (cit. on p. 20).

[38] Ken TD Eames and Matt J Keeling. "Modeling dynamic and network heterogeneities in the spread of sexually transmitted diseases". In: *Proceedings of the National Academy of Sciences* 99.20 (2002), pp. 13330–13335 (cit. on p. 127).

[39] Paul Erdős and Alfréd Rényi. "On the evolution of random graphs". In: *Publ. Math. Inst. Hung. Acad. Sci* 5.1 (1960), pp. 17–60 (cit. on pp. 16, 47).

[40] Paul Erdos and Alfréd Rényi. "On the evolution of random graphs". In: *Publ. Math. Inst. Hung. Acad. Sci* 5.1 (1960), pp. 17–60 (cit. on pp. 42, 45).

[41] Fedor V. Fomin, Fabrizio Grandoni, and Dieter Kratsch. "Measure and conquer: a simple O(20.288n) independent set algorithm". In: *SODA*. 2006 (cit. on p. 91).

[42] Santo Fortunato. "Community detection in graphs". In: *Physics reports* 486.3 (2010), pp. 75–174 (cit. on p. 77).

[43] Alan Gabel and Sidney Redner. "Sublinear but never superlinear preferential attachment by local network growth". In: *Journal of Statistical Mechanics: Theory and Experiment* 2013.02 (2013), P02043 (cit. on p. 19).

[44] Nicola Galli and Klaus Simon. "On the distribution of the number of isolated vertices in random undirected graphs". In: *Institute for Theoretical Computer Science, ETH Zurich* 11 () (cit. on p. 46).

[45] K Glover and L Silverman. "Characterization of structural controllability". In: *IEEE Transactions on Automatic control* 21.4 (1976), pp. 534–537 (cit. on p. 25).

[46] Bei Gou and Ali Abur. "An improved measurement placement algorithm for network observability". In: *IEEE Transactions on Power Systems* 16.4 (2001), pp. 819–824 (cit. on p. 84).

[47] Alasdair J Graham and David A Pike. "A note on thresholds and connectivity in random directed graphs". In: *Atl. Electron. J. Math* 3.1 (2008), pp. 1–5 (cit. on p. 45).

[48] Martin Grandjean. *Connected World: Untangling the Air Traffic Network*. `http://www.martingrandjean.ch/connected-world-air-traffic-network/`. [Online; accessed 27-August-2019]. 2016 (cit. on p. 18).

[49] Adrien Guille et al. "Information diffusion in online social networks: A survey". In: *ACM Sigmod Record* 42.2 (2013), pp. 17–28 (cit. on p. 23).

[50] Roger Guimera et al. "The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles". In: *Proceedings of the National Academy of Sciences* 102.22 (2005), pp. 7794–7799 (cit. on p. 22).

[51] Sushmita Gupta, Venkatesh Raman, and Saket Saurabh. "Fast exponential algorithms for maximum r-regular induced subgraph problems". In: *International Conference on Foundations of Software Technology and Theoretical Computer Science*. Springer. 2006, pp. 139–151 (cit. on pp. 91, 93).

[52] Md Kamrul Hassan, Liana Islam, and Syed Arefinul Haque. "Degree distribution, rank-size distribution, and leadership persistence in mediation-driven attachment networks". In: *Physica A: Statistical Mechanics and its Applications* 469 (2017), pp. 23–30 (cit. on p. 20).

[53] Joao P Hespanha. *Linear systems theory*. Princeton university press, 2018 (cit. on pp. 14, 15).

[54] Petter Holme. "Rare and everywhere: Perspectives on scale-free networks". In: *Nature communications* 10.1 (2019), p. 1016 (cit. on p. 20).

[55] Petter Holme et al. "Attack vulnerability of complex networks". In: *Physical review E* 65.5 (2002), p. 056109 (cit. on p. 123).

[56] John Hopcroft. *Advanced Design and Analysis of Algorithms - Lecture 7: More on Graph Connectivity*. 2008 (cit. on p. 46).

[57] Sakhraoui Imane, Trajin Baptiste, and Rotella Frédéric. "Discrete Linear Functional Observer for the Thermal Estimation in Power Modules". In: *2018 IEEE 18th International Power Electronics and Motion Control Conference (PEMC)*. IEEE. 2018, pp. 812–817 (cit. on p. 89).

[58] Takayuki Ishizaki et al. "Clustered model reduction of positive directed networks". In: *Automatica* 59 (2015), pp. 238–247 (cit. on pp. 39–41, 139).

[59] Takayuki Ishizaki et al. "Clustering-based $\mathcal{H}_2$-state aggregation of positive networks and its application to reduction of chemical master equations". In: *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE. 2012, pp. 4175–4180 (cit. on p. 40).

[60] Takayuki Ishizaki et al. "Model Reduction and Clusterization of Large-Scale Bidirectional Networks." In: *IEEE Trans. Automat. Contr.* 59.1 (2014), pp. 48–63 (cit. on pp. 37, 40).

[61] Takayuki Ishizaki et al. "Model reduction of multi-input dynamical networks based on clusterwise controllability". In: *American Control Conference (ACC), 2012*. IEEE. 2012, pp. 2301–2306 (cit. on p. 40).

[62] Francesco Lo Iudice, Francesco Sorrentino, and Franco Garofalo. "On node controllability and observability in complex dynamical networks". In: *IEEE Control Systems Letters* (2019) (cit. on p. 84).

[63] Matthew O Jackson. *Social and economic networks*. Princeton university press, 2010, pp. 89–97 (cit. on p. 39).

[64] Hawoong Jeong et al. "The large-scale organization of metabolic networks". In: *Nature* 407.6804 (2000), pp. 651–654 (cit. on p. 19).

[65] Meng Ji and Magnus Egerstedt. "Observability and estimation in distributed sensor networks". In: *2007 46th IEEE Conference on Decision and Control*. IEEE. 2007, pp. 4221–4226 (cit. on p. 84).

[66] James Holland Jones and Mark S Handcock. "Social networks (communication arising): Sexual contacts and epidemic thresholds". In: *Nature* 423.6940 (2003), p. 605 (cit. on p. 19).

[67] William Ogilvy Kermack and Anderson G McKendrick. "A contribution to the mathematical theory of epidemics". In: *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character* 115.772 (1927), pp. 700–721 (cit. on p. 118).

[68] Daniel Kobler and Udi Rotics. "Finding Maximum Induced Matchings in Subclasses of Claw-Free andP 5-Free Graphs, and in Graphs with Matching and Induced Matchingof Equal Maximum Size". In: *Algorithmica* 37.4 (2003), pp. 327–346 (cit. on p. 91).

[69] Elias Koutsoupias and Christos Papadimitriou. "Worst-case equilibria". In: *Annual Symposium on Theoretical Aspects of Computer Science*. Springer. 1999, pp. 404–413 (cit. on p. 37).

[70] Dmitri Krioukov et al. "Hyperbolic geometry of complex networks". In: *Physical Review E* 82.3 (2010), p. 036106 (cit. on p. 20).

[71] Jure Leskovec and Andrej Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. `http://snap.stanford.edu/data`. June 2014 (cit. on p. 3).

[72] Jure Leskovec et al. "Patterns of cascading behavior in large blog graphs". In: *Proceedings of the 2007 SIAM international conference on data mining*. SIAM. 2007, pp. 551–556 (cit. on p. 122).

[73] Fredrik Liljeros et al. "The web of human sexual contacts". In: *Nature* 411.6840 (2001), pp. 907–908 (cit. on pp. 19, 137).

[74] Ching-Tai Lin. "Structural controllability". In: *IEEE Transactions on Automatic Control* 19.3 (1974), pp. 201–208 (cit. on p. 25).

[75] Gustav Lindmark and Claudio Altafini. "A driver node selection strategy for minimizing the control energy in complex networks". In: *IFAC-PapersOnLine* 50.1 (2017), pp. 8309–8314 (cit. on p. 56).

[76] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási. "Controllability of complex networks". In: *nature* 473.7346 (2011), p. 167 (cit. on pp. 29, 84).

[77] Yike Liu et al. "Graph Summarization Methods and Applications: A Survey". In: *ACM Computing Surveys (CSUR)* 51.3 (2018), p. 62 (cit. on p. 3).

[78] Andres Ladino Lopez. "Traffic state estimation and prediction in freeways and urban networks". PhD thesis. 2018 (cit. on p. 37).

[79] László Lovász. "On the Shannon capacity of a graph". In: *IEEE Transactions on Information theory* 25.1 (1979), pp. 1–7 (cit. on p. 91).

[80] Vadim V Lozin and Raffaele Mosca. "Maximum regular induced subgraphs in 2P3-free graphs". In: *Theoretical Computer Science* 460 (2012), pp. 26–33 (cit. on p. 91).

[81] Carlos Luz. "Improving an upper bound on the size of k-regular induced subgraphs". In: *J. Comb. Optim.* 22 (Nov. 2011), pp. 882–894 (cit. on p. 91).

[82] Carlos J. Luz. "An upper bound on the independence number of a graph computable in polynomial-time". In: *Operations Research Letters* 18.3 (1995), pp. 139 –145 (cit. on p. 91).

[83] Nicolas Martin, Paolo Frasca, and Carlos Canudas-de Wit. "A network reduction method inducing scale-free degree distribution". In: *2018 European Control Conference (ECC)*. IEEE. 2018, pp. 2236–2241 (cit. on pp. 32, 53).

[84] Nicolas Martin, Paolo Frasca, and Carlos Canudas-de Wit. "Large-scale network reduction towards scale-free structure". In: *IEEE Transactions on Network Science and Engineering* (2020) (cit. on pp. 34, 53).

[85] Nicolas Martin, Paolo Frasca, and Carlos Canudas-de Wit. "MergeToCure: a New Strategy to Allocate Cure in an Epidemic over a Grid-like network Using a Scale-Free Abstraction". In: *IFAC-PapersOnLine* 51.23 (2018), pp. 34–39 (cit. on pp. 32, 37, 117).

[86] Nicolas Martin, Paolo Frasca, and Carlos Canudas-de Wit. "Network reduction towards a scale-free structure preserving physical properties". In: *Complex Networks 2017-The 6th International Conference on Complex Networks and Their Applications*. 2017, pp. 294–296 (cit. on p. 32).

[87] Nicolas Martin et al. "The price of connectedness in graph partitioning problems". In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 2313–2318 (cit. on pp. 32, 35).

[88] Joel C Miller and James M Hyman. "Effective vaccination strategies for realistic social networks". In: *Physica A: Statistical Mechanics and its Applications* 386.2 (2007), pp. 780–785 (cit. on p. 123).

[89] F Molnár et al. "Minimum dominating sets in scale-free network ensembles". In: *Scientific reports* 3 (2013), p. 1736 (cit. on p. 29).

[90] Nima Monshizadeh, Harry L Trentelman, and M Kanat Camlibel. "Stability and synchronization preserving model reduction of multi-agent systems". In: *Systems & Control Letters* 62.1 (2013), pp. 1–10 (cit. on p. 37).

[91] Franz-Josef Müller and Andreas Schuppert. "Few inputs can reprogram biological networks". In: *Nature* 478.7369 (2011), E4 (cit. on p. 29).

[92] PC Müller and HI Weber. "Analysis and optimization of certain qualities of controllability and observability for linear dynamical systems". In: *Automatica* 8.3 (1972), pp. 237–246 (cit. on p. 28).

[93] Steve Mussmann. *Counting Partitions or Subsets*. `https://www.projectrhea.org/rhea/index.php/Counting_subsets_of_sets#Partitions_Of_Sets_into_Fixed_Sizes`. [Online; accessed October-2018] (cit. on p. 43).

[94] Jose C Nacher and Tatsuya Akutsu. "Dominating scale-free networks with variable scaling exponent: heterogeneous networks are not difficult to control". In: *New Journal of Physics* 14.7 (2012), p. 073005 (cit. on p. 29).

[95] Mark EJ Newman. "The structure and function of complex networks". In: *SIAM review* 45.2 (2003), pp. 167–256 (cit. on p. 54).

[96] Muhammad Umar B. Niazi, Carlos Canudas-de Wit, and Alain Kibangou. "Average observability of large-scale network systems". In: *ECC 2019 - European Control Conference*. Naples, Italy, June 2019, pp. 1–6 (cit. on pp. 83, 84, 87, 103, 136).

[97] Muhammad Umar B Niazi et al. "Scale-free estimation of the average state in large-scale systems". In: *IEEE Control Systems Letters* 4.1 (2019), pp. 211–216 (cit. on p. 37).

[98] Denis Nikitin, Carlos Canudas-de Wit, and Paolo Frasca. "Boundary Control for Output Regulation in Scale-Free Positive Networks". In: 2019 (cit. on p. 136).

[99] Giuseppe Notarstefano and Gianfranco Parlangeli. "Controllability and observability of grid graphs via reduction and symmetries". In: *IEEE Transactions on Automatic Control* 58.7 (2013), pp. 1719–1731 (cit. on p. 84).

[100] Cameron Nowzari, Victor M Preciado, and George J Pappas. "Analysis and control of epidemics: A survey of spreading processes on complex networks". In: *IEEE Control Systems Magazine* 36.1 (2016), pp. 26–46 (cit. on pp. 118, 121, 122).

[101] Romualdo Pastor-Satorras and Alessandro Vespignani. "Epidemic spreading in scale-free networks". In: *Physical review letters* 86.14 (2001), p. 3200 (cit. on p. 121).

[102] Romualdo Pastor-Satorras and Alessandro Vespignani. "Immunization of complex networks". In: *Physical review E* 65.3 (2002), p. 036104 (cit. on pp. 25, 121).

[103] S Unnikrishna Pillai, Torsten Suel, and Seunghun Cha. "The Perron-Frobenius theorem: some of its applications". In: *IEEE Signal Processing Magazine* 22.2 (2005), pp. 62–75 (cit. on p. 62).

[104] José Roberto C Piqueira and Vanessa O Araujo. "A modified epidemiological model for computer viruses". In: *Applied Mathematics and Computation* 213.2 (2009), pp. 355–360 (cit. on p. 23).

[105] Derek de Solla Price. "Statistical studies of networks of scientific papers". In: *Statistical Association Methods for Mechanized Documentation: Symposium Proceedings*. Vol. 269. US Government Printing Office. 1965, p. 187 (cit. on p. 18).

[106] Amirreza Rahmani et al. "Controllability of multi-agent systems from a graph-theoretic perspective". In: *SIAM Journal on Control and Optimization* 48.1 (2009), pp. 162–186 (cit. on p. 84).

[107] Basil Cameron Rennie and Annette Jane Dobson. "On Stirling numbers of the second kind". In: *Journal of Combinatorial Theory* 7.2 (1969), pp. 116–121 (cit. on p. 57).

[108] CJ Rhodes and Roy M Anderson. "Epidemic thresholds and vaccination in a lattice model of disease spread". In: *Theoretical Population Biology* 52.2 (1997), pp. 101–118 (cit. on p. 123).

[109] John Michael Robson. "Algorithms for maximum independent sets". In: *Journal of Algorithms* 7.3 (1986), pp. 425–440 (cit. on p. 91).

[110] Ronald Ross. "On some peculiar pigmented cells found in two mosquitos fed on malarial blood". In: *British medical journal* 2.1929 (1897), p. 1786 (cit. on p. 118).

[111] Robert Shields and J Pearson. "Structural controllability of multiinput linear systems". In: *IEEE Transactions on Automatic control* 21.2 (1976), pp. 203–212 (cit. on p. 25).

[112] Dan Stanislawski. "The origin and spread of the grid-pattern town". In: *Geographical Review* 36.1 (1946), pp. 105–120 (cit. on p. 75).

[113] Maciej M SysŁ et al. "The subgraph isomorphism problem for outerplanar graphs". In: *Theoretical Computer Science* 17.1 (1982), pp. 91–97 (cit. on p. 84).

[114] Riitta Toivonen et al. "A model for social networks". In: *Physica A: Statistical Mechanics and its Applications* 371.2 (2006), pp. 851–860 (cit. on p. 128).

[115] *TomTom Traffic Index*. 2018. URL: `https://www.tomtom.com/trafficindex/` (cit. on p. 107).

[116] Franco Travostino, Joe Mambretti, and Gigi Karmous-Edwards. *Grid networks: enabling grids with advanced communication technology*. John Wiley & Sons, 2006 (cit. on p. 75).

[117] Eimutis Valakevicius and Mindaugas Snipas. "A recursive algorithm for computing steady state probabilities". In: *IIIS* (2010) (cit. on p. 73).

[118] Piet Van Mieghem. "The N-intertwined SIS epidemic network model". In: *Computing* 93.2-4 (2011), pp. 147–169 (cit. on p. 121).

[119] Piet Van Mieghem et al. "Decreasing the spectral radius of a graph by link removals". In: *Physical Review E* 84.1 (2011), p. 016101 (cit. on p. 123).

[120] Duncan J Watts and Steven H Strogatz. "Collective dynamics of small-world networks". In: *Nature* 393.6684 (1998), p. 440 (cit. on p. 128).

[121] Van J Wedeen et al. "The geometric structure of the brain fiber pathways". In: *Science* 335.6076 (2012), pp. 1628–1634 (cit. on p. 75).

[122] Carlos Canudas-de Wit et al. "Grenoble traffic lab: An experimental platform for advanced traffic monitoring and forecasting". In: *IEEE Control Systems* 35.3 (2015), pp. 23–39 (cit. on pp. 108, 136).

[123] Felix F Wu and A Monticelli. "Network observability: theory". In: *IEEE Transactions on Power Apparatus and Systems* 5 (1985), pp. 1042–1048 (cit. on p. 84).

[124] Shi Henghua1 He Jingsha2 Xu Xin. "Algorithm research for selecting measurement nodes in measuring network traffic based on adjacency matrix". In: *Journal of Southeast University (Natural Science Edition)* (2008), S1 (cit. on p. 84).

[125] Soon-Hyung Yook, Hawoong Jeong, and Albert-László Barabási. "Modeling the Internet's large-scale topology". In: *Proceedings of the National Academy of Sciences* 99.21 (2002), pp. 13382–13386 (cit. on p. 19).

[126] Haifeng Zhang et al. "Hub nodes inhibit the outbreak of epidemic under voluntary vaccination". In: *New Journal of Physics* 12.2 (2010), p. 023015 (cit. on p. 25).

Résumé — En raison de la complexité inhérente à l'analyse de réseau de très grande taille, l'élaboration d'algorithmes de partitionnement et diverses problématiques connexes sont traitées au long de cette thèse. Dans un premier temps, une question préliminaire est traitée: puisque les noeuds au sein d'une partie ne sont pas nécessairement connexes, comment quantifier l'impact d'une contrainte de connexité ? Nous proposons ensuite un algorithme de partitionnement assurant que le réseau réduit soit *scale-free*. Ceci permet de tirer profit des propriétés intrinsèques de ce type de réseaux. Nous nous intéressons également aux propriétés à préserver pour respecter la nature physique et dynamique du réseau initial. Dans une troisième partie, nous proposons une méthode pour identifier les noeuds à mesurer dans un réseau pour garantir une reconstruction efficace de la valeur moyenne des autre noeuds. Finalement, nous proposons trois applications: la première concerne le trafic routier et nous montrons que notre premier algorithme de partitionnement permet d'obtenir un réseau réduit émulant efficacement le réseau initial. Les deux autres applications concernent les réseaux d'épidémiologie. Dans la première nous montrons qu'un réseau réduit *scale-free* permet de construire une stratégie efficace d'attribution de soin au sein d'une population. Dans la dernière application, nous tirons profit des résultats sur la reconstruction de moyenne pour estimer l'évolution d'une épidémie dans un réseau de grande taille.

Mots clés : Algorithme de partitionnement, réseau scale-free, réduction de réseau

Abstract — In light of the complexity induced by large-scale networks, the design of network partitioning algorithms and related problematics are at the heart of this thesis. First, we raise a preliminary question on the structure of the partition itself: as the parts may includes disconnected nodes, we want to quantify the drawbacks to impose the nodes inside each part to be connected. Then we study the design of a partitioning algorithm inducing a reduced scale-free network. This allows to take advantage of the inherent features of this type of network. We also focus on the properties to preserve to respect the physical and dynamical profile of the initial network. We investigate then how to partition a network between measured and unmeasured nodes ensuring that the average of the unmeasured nodes can be efficiently reconstructed. In particular we show that, under hypothesis, this problem can be reduced to a problem of detection of subgraph with particular properties. Methods to achieve this detection are proposed. Finally, three applications are presented: first we apply the partitioning algorithm inducing scale-freeness to a large-scale urban traffic network. We show then that, thanks to the properties preserved through the partition, the reduced network can be used as an abstraction of the initial network. The second and third applications deal with network epidemics. First, we show that the scale-freeness of the abstracting network can be used to build a cure-assignation strategy. In the last application, we take advantage of the result on average reconstruction to estimate the evolution of a disease on a large-scale network.

Keywords: Partitionning algorithm, scale-free network, network reduction

GIPSA-Lab, 11 Rue des Mathématiques
38400 Saint-Martin-d'Hères