Estimation and control for resilience in large-scale network systems

Ujjwal Pratap

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : EEATS - Electronique, Electrotechnique, Automatique, Traitement du Signal (EEATS)
Spécialité : Automatique - Productique
Unité de recherche : Grenoble Images Parole Signal Automatique

Estimation et commande pour la résilience dans les systèmes en réseaux à grande échelle

Estimation and control for resilience in large-scale network systems

Présentée par :

Ujjwal PRATAP

Direction de thèse :

Carlos CANUDAS DE WIT Directeur de Recherche, Université Grenoble Alpes	Directeur de thèse
Federica GARIN INRIA	Co-encadrante de thèse
Henrik SANDBERG KTH Royal Institute of Technology	Co-encadrant de thèse

Rapporteurs :

Ming CAO
PROFESSEUR, Hanzehogeschool Groningen
Constantin MORARESCU
PROFESSEUR DES UNIVERSITES, UNIVERSITE DE LORRAINE

Thèse soutenue publiquement le **7 septembre 2022**, devant le jury composé de :

Olivier SENAME PROFESSEUR DES UNIVERSITES, GRENOBLE INP	Président
Ming CAO PROFESSEUR, Hanzehogeschool Groningen	Rapporteur
Constantin MORARESCU PROFESSEUR DES UNIVERSITES, UNIVERSITE DE LORRAINE	Rapporteur
Alain RAPAPORT DIRECTEUR DE RECHERCHE, CNRS DELEGATION OCCITANIE EST	Examinateur
Julie DUGDALE Maître de conférences HDR, UNIVERSITE GRENOBLE ALPES	Examinatrice
Silvia SIRI PROFESSEUR ASSOCIE, Università Degli Studi di Genova	Examinatrice

Invités :

Federica Garin
CHARGE DE RECHERCHE, Inria
Henrik Sandberg
PROFESSEUR, KTH Royal Institute of Technology

UNIVERSITÉ DE GRENOBLE ALPES

ÉCOLE DOCTORALE EEATS

Eléctronique, Eléctrotechnique, Automatique, Traitement du Signal

T H È S E

pour obtenir le titre de

docteur en sciences

de l'Université de Grenoble

Mention : AUTOMATIQUE

Présentée et soutenue par

Ujjwal PRATAP

Estimation and control for resilience in large-scale network systems

Thèse dirigée par Carlos CANUDAS-DE-WIT et co-encadrée par Federica GARIN et Henrik SANDBERG

préparée au laboratoire Grenoble Images Parole Signal Automatique (GIPSA-Lab)

soutenue le 07/09/2022

Jury :

Rapporteurs :	Ming CAO	-	University of Groningen
	Constantine MORARESCU	-	Universite de Lorraine
Président :	Olivier SENAME	-	Université Grenoble Alpes
Examinateur/Examinatrice :	Alain RAPAPORT	-	INRAE
	Julie DUGDALE	-	Université Grenoble Alpes
	Silvia SIRI	-	University of Genova

Abstract — This research has been done in the context of European Research Council's (ERC) Advance Grant project Scale-Freeback. The aim of Scale-Freeback project is to develop a holistic scale-free control approach to complex systems, and to set new foundations for a theory dealing with complex physical networks with arbitrary dimension.

The contributions of this work are divided into two parts:
Part 1: Outlier detection and state estimation in network systems
Part 2: Modeling and control of mobility for epidemic mitigation in large scale urban networks

In part 1, we consider the problem of average state estimation in network systems where only some of the states are measured. There is an outlier among the unmeasured states, which is so different from the remaining states that it significantly affects the average value. We develop a methodology to detect the outlier and estimate the average state, excluding the outlier for the cases when the system model might be known or partially unknown. We also illustrate the method in a thermal diffusion system.

In part 2, we consider the problem of modeling and control of human mobility in large-scale urban networks. At first, we provide a supply-demand-based mobility model in discrete time, which preserves the properties of non-negativity, boundedness, and mass conservation. This model captures the daily movement of people between residences and places of interest such as workplaces, schools, hospitals, cinemas, etc. called destinations, using time schedules and gating profiles which also accommodate the possibility of imposing restrictions on mobility. We also built the large-scale mobility network in the metropolitan city of Grenoble by using publicly available information to answer the questions: "where do the people move?" "When do they move?" and "how many people move?" Secondly, since the mobility can be controlled in the model by using a control parameter that tunes the capacities of the destinations, we exploit this control for epidemic mitigation. Considering an integrated model for epidemic and mobility in continuous time, we provide a discretization that preserves the property of the original model. Then, we provide optimal mobility control policies that maximize the economic activity of an urban area by tuning the operating capacities of destinations keeping the total infections within the area bounded with application to Grenoble metropolis. Finally, we also developed an online simulation platform called GTL-Healthmob, which aims at visualizing the population movement, simulating epidemic propagation, and optimizing the mobility restrictions to limit the epidemic spread in the Grenoble metropolis by taking different epidemic and mobility control parameters inputs from users and showing the results in terms of plots and heatmaps overlaid onto the map of the metropolis.

Résumé — Cette recherche s'inscrit principalement dans le cadre du projet Scale-Freeback du Conseil européen de la recherche (ERC). L'objectif du projet Scale-Freeback est de développer une approche holistique du contrôle invariant d'échelle des systèmes complexes, et d'établir de nouvelles bases pour une théorie traitant des réseaux physiques complexes de dimension arbitraire.

Les contributions de cette thèse sont divisées en deux parties:
Partie 1: Détection de valeurs aberrantes et estimation d'état dans les systèmes de réseaux
Partie 2: Modélisation et contrôle de la mobilité pour la réduction des épidémies dans les réseaux urbains à grande échelle

Dans la première partie, nous considérons le problème de l'estimation de l'état moyen dans les systèmes de réseau dans lesquels seuls certains états sont mesurés et il existe une valeur aberrante parmi les états non mesurés, qui est si différente des autres états qu'elle affecte la valeur moyenne de manière significative. Nous développons une méthodologie pour détecter la valeur aberrante et pour estimer l'état moyen en excluant la valeur aberrante pour les cas où les matrices du système peuvent être connues ou partiellement inconnues. Notre méthode a également été illustrée sur un système de diffusion thermique.

Dans la deuxième partie, nous considérons le problème de la modélisation et du contrôle de la mobilité humaine dans les réseaux urbains de grande échelle. Dans un premier temps, nous fournissons un modèle de mobilité en temps discret basé sur l'offre et la demande qui préserve les propriétés de non-négativité, de limitabilité et de conservation de la masse. Ce modèle capture le mouvement quotidien des personnes entre les résidences et les lieux d'intérêt tels que les lieux de travail, les écoles, les hôpitaux, les cinémas, etc. appelés destinations, en utilisant des horaires et des profils d'accès qui permettent également d'imposer des restrictions à la mobilité. Nous avons également construit le réseau de mobilité à grande échelle de la ville métropolitaine de Grenoble en utilisant les informations disponibles publiquement pour répondre aux questions suivantes : "où les gens se déplacent-ils ?" "quand se déplacent-ils?" et "combien de personnes se déplacent ?". Deuxièmement, puisque dans le modèle la mobilité peut être contrôlée en utilisant un paramètre de contrôle qui règle les capacités des destinations, nous exploitons ce contrôle pour l'atténuation d'épidémies. En utilisant un modèle intégré pour l'épidémie et la mobilité en temps discret, nous fournissons des solutions à un problème de contrôle optimal, afin de concevoir des politiques de contrôle de la mobilité qui maximisent l'activité économique d'une zone urbaine en ajustant les capacités de fonctionnement des destinations de différents types en utilisant des entrées de commande et en maintenant les infections totales dans la zone bornées avec une application au réseau de mobilité de la métropole de Grenoble. Enfin, nous avons également développé une plateforme de simulation en ligne appelée GTL-Healthmob, qui vise à visualiser le mouvement de la population, à simuler la propagation des épidémies et à optimiser les restrictions de mobilité pour limiter la propagation de l'épidémie dans la métropole grenobloise en prenant différents paramètres de contrôle de l'épidémie et de la mobilité des utilisateurs et en montrant les résultats en termes de graphiques et de heat map superposées à la carte de la métropole.

To my beloved parents

Acknowledgements

It is my pleasure that I worked on a thesis regarding resilience during a period when the whole human race has shown resilience against a pandemic. These last three and half years of my life in Grenoble have been full of learning, adventure and perseverance.

I express my sincere gratitude to my supervisors at Gipsa lab, Carlos Canudas-de-Wit and Federica Garin, for their support, continuous guidance, motivation, valuable suggestions and feedback throughout the thesis. Their critical analysis and integral view of research have laid a strong foundation for my research career. I am thankful for their patience and for sharing their immense knowledge with me. I would also like to thank my co-supervisor at KTH, Henrik Sandberg, for his valuable input and suggestions whenever we met online or offline.

I would like to thank the jury members: Ming Cao, Constantine Morarescu, Olivier Sename, Alain Rapaport, Julie Dugdale, and Silvia Siri for the insightful discussion, questions and comments. I thank them for their valuable time and effort in evaluating this work. I also thank Paolo Frasca and Alain Kibangou for their valuable input during team meetings.

Working with my teammates Martin, Umar, Denis, Liudmila, Vadim, Leo, Mladen, Stephane and Nicolas at INRIA was a pleasure, and I thank them for making it a friendly environment. I am grateful for all the scientific and non-scientific exchanges we had during lunch, coffee breaks and board game nights. Special thanks to Martin for his support during the writing of this thesis.

I am grateful for the financial support during this PhD work which I received the from the ERC ScaleFreeback project.

This thesis would not have been possible without the support of the close-knit group of friends I had here in Grenoble. I thank Madhav, Vivek, Priyank, Vishwajeet and Purwayan for their constant support, be it mental or emotional. All those crazy cycle trips, walks, hiking, discussions and cooking will always be in my memories, specially the spontaneous cycling trips with Vivek to mountains and heated debates with Priyank and Madhav. I want to thank Satyajeet and Arka for organizing wonderful trips and occasional visits to Grenoble. I am also thankful to my friends back home specially Asit, Sanjeev, Ankit, Chandni, Ramen, Anup and Akshay. Special thanks to Asit for all the support.

It is an excellent opportunity to thank my master's thesis supervisor Nutan Kumar Tomar at IIT Patna, who introduced me to control theory. I would like to thank my constant inspiration Ashwini Assam for the motivation and inspiration to pursue a career in research.

Most importantly, it is my parents and sisters whose unconditional love, support and faith in me make everything possible. I owe them everything. Last but not least, I am thankful to my *ardhangini* Kajal for being there for me always and bearing with me.

Contents

General Introduction **1**

I Average state estimation in network systems with outliers **5**

1 Introduction **7**

2 Outlier detection and trimmed-average estimation **11**

2.1 Problem formulation 11

2.2 Outlier at a known position 14

2.3 Outlier at an unknown position 18

2.4 System model is partially unknown 22

2.5 Outlier detection in a faulty metal plate 30

2.6 Concluding remarks 35

II Modeling and control of mobility for epidemic mitigation in large scale urban networks **37**

3 Introduction **39**

4 Model for human mobility and epidemic spread **45**

4.1 Continuous-time mobility model: where, when and how many? 45

4.2 Continuous-time SIR-mobility model 49

4.3 Discrete-time mobility model with preserving properties 50

4.4 Discrete-time SIR-mobility model 55

4.5 Concluding remarks 58

5 Human mobility model calibration **59**

5.1 Origin-destination network 60
5.2 Time dependent profiles 85
5.3 Number of people 88
5.4 Simulation Example 89
5.5 Concluding remarks 91

6 Optimization of human mobility for epidemic mitigation 93

6.1 Urban human mobility and epidemic model 94
6.2 Optimal capacity control for epidemic mitigation 97
6.3 Category-free policies 100
6.4 Category-dependent policies 106
6.5 Concluding remarks 122

7 GTL-Healthmob simulation platform 123

7.1 Platform Overview 124
7.2 User Interface 125
7.3 Concluding remarks 133

Conclusion 135

Bibliography 147

General Introduction

The word resilience originates from the Latin 'resilire', which means to leap back. In general, resilience can be defined as an ability of an entity to cope and recover after confronting unnatural, alarming, and often unexpected events or complex conditions. It is a property of an entity to recoil and bounce back after facing an adversarial event. Resilience as a process is versatile, having a multidisciplinary approach. It has numerous aspects, such as mental, psychological, physical, and resilience in different complex systems, making it interesting for medical practitioners, policymakers, scientists, and engineers.

In this thesis, we are interested in complex systems. A complex system can either be engineered, such as a power network, or can exist naturally, like a human society. Generally, a complex system needs a set of rules for proper functioning in standard times. However, should any external or internal disruption come, it requires improvements to the existing algorithms to ensure the system's proper functioning. In this regard, this thesis deals with two specific aspects of such resilience. The first is imparting resilience in the process of average state estimation in network systems with outliers, and the other is societal resilience in case of an epidemic outbreak.

Specifically, our contributions in these aspects have been presented in two parts:

- Part 1: Average state-estimation in network systems with outliers.
- Part 2: Modeling and control of mobility for epidemic mitigation in large scale urban networks.

In part 1, we consider the problem of average state estimation in network systems where only some of the states are measured. There is an outlier among the unmeasured states, which is so different from the remaining states that it significantly affects the average value. We develop a methodology to detect the outlier and estimate the average state, excluding the outlier for the cases when the system model might be known or partially unknown. We also illustrate the method in a thermal diffusion system.

In part 2, we consider the problem of modeling and control of human mobility in large-scale urban networks. At first, considering a continuous-time supply-demand-based human mobility model, we provide a model in discrete-time, which preserves the properties of non-negativity, boundedness, and mass conservation. This model captures the daily movement of people between residences and places of interest such as workplaces, schools, hospitals, cinemas, etc called destinations, using time schedules and gating profiles which also accommodate the possibility of imposing restrictions on mobility. We also provide techniques to build a model of the mobility network in a metropolitan city with application to Grenoble metropolis by using publicly available information to answer the questions: "where do the people move?"

"When do they move?" and "how many people move?" Secondly, since the mobility can be controlled in the model by using a control parameter that tunes the capacities of the destinations, we exploit this control for epidemic mitigation. Considering an integrated model for epidemic and mobility in continuous time, we provide a discretization that preserves the property of the original model. Then, we provide solutions to an optimal control problem that maximize the economic activity of an urban area by tuning the operating capacities of destinations while keeping the total infections within the area bounded. We apply these techniques to the large-scale mobility network of Grenoble metropolis. Finally, we developed an online simulation platform called GTL-Healthmob, which aims at visualizing the population movement, simulating epidemic propagation, and optimizing the mobility restrictions to limit the epidemic spread in the Grenoble metropolis by taking different epidemic and mobility control parameters inputs from users and showing the results in terms of plots and heatmaps overlaid onto the map of the metropolis.

Thesis Organization

The first part deals with the problem of average state estimation in a network system in presence of outliers and comprises of Chapter 1 and Chapter 2. In Chapter 1, we provide the literature review and identify our contributions related to the average state estimation. In chapter 2, we provide the problem formulation and a methodology for outlier detection and trimmed-average estimation in a network system with an outlier.

The second part deals with the modeling and control of human mobility for epidemic mitigation and comprises of Chapter 3 to Chapter 7. In Chapter 3, we provide the literature review and identify our contributions. Chapter 4 considers a urban human mobility model integrated with epidemic spread process, and provides a model in discrete-time that preserves the properties of mass-conservation, non-negativity and boundedness. In Chapter 5, we provide methodologies to compute the parameters for the model in Chapter 4 for a metropolitan city with application to Grenoble metropolis. Chapter 6 provides optimal mobility restriction policies for epidemic mitigation with application to Grenoble metropolis. Finally, Chapter 7 presents the GTL-Healthmob simulation platform developed on the basis of work done in Chapters 4, 5 and 6.

Publications

Based on the work done in this thesis, we have made following contributions:

Journals

- U. Pratap, C. Canudas-de-Wit and F. Garin, Outlier detection and trimmed-average estimation in network systems, European Journal of Control, vol. 60, pp. 36-47, Jul.

2021. Preprint: https://hal.archives-ouvertes.fr/hal-03225214

- U. Pratap, C. Canudas-de-Wit, and F. Garin, Where, when and how people move in large-scale urban networks: the Grenoble saga, (working paper). Preprint: https://hal.archives-ouvertes.fr/hal-03554612

- U. Pratap, C. Canudas-de-Wit, and F. Garin, Optimizing mobility to various destinations to save lives and economy during an epidemic outbreak: application to Grenoble metropolis, (working paper).

Conference publications

- U. Pratap, C. Canudas-de-Wit and F. Garin, Average state estimation in presence of outliers, 2020 59th IEEE Conference on Decision and Control(CDC) 2020, pp. 6058-6063. Preprint: https://hal.archives-ouvertes.fr/hal-02925053

- U. Pratap, C. Canudas-de-Wit, and F. Garin, Scalable socio-economic optimal mobility policies for epidemic mitigation under health constraints: application to Grenoble metropolis, (working paper).

- U. Pratap, Leo Senique, C. Canudas-de-Wit, GTL-Healthmob: Simulation platform for urban mobility and epidemic control, 6èmes Journées des Démonstrateurs en Automatique 2022, Jun 2022, Angers, France. Preprint: https://hal.archives-ouvertes.fr/hal-03674156v1

Simulation platform

- GTL-Healthmob[1] - It is an online simulation platform developed within the team[2] based on the work done in this thesis. It is a part of Grenoble Traffic Lab(GTL) which takes Grenoble *métropole* as a case study. Here, an user can visualize the developed Grenoble mobility network, simulate different mobility restriction scenarios to see its effect on epidemic spread in Grenoble metropolis and also visualize optimal mobility restriction scenarios.

[1] http://gtlville.inrialpes.fr/covid-19
[2] https://www.inria.fr/en/dance

Part I

Average state estimation in network systems with outliers

Chapter 1

Introduction

This part addresses the problem of outlier detection and trimmed-average state estimation in an LTI network system. The problem of estimation in network control systems is very well studied due to its vast area of application such as power networks, communication networks, and traffic networks to name a few. The complexity in the problem of state estimation in a network system increases when there is a limited information about the network and limited access to only a few states for estimation and control purposes. This problem of complexity in large-scale network can be tackled by aggregation of the state variables as done in the context of Scale-Freeback[1] project and a natural choice for such an aggregate function is the average of the states. Estimation and control of such an aggregated states is reasonable as it can have many applications such as in building thermal monitoring [Den+14],[NWK20], urban traffic networks [RVWF20],[RHG15], power modules [IBF18] and epidemic spread over large-networks [MFW20].

Therefore, in some networks, instead of reconstructing the entire states, some aggregation of the unmeasured states are estimated using a few available measurements obtained from the dedicated sensors placed at some positions. For instance, [NCK20] investigates the problem of estimation of the average of the unmeasured nodes in a network system and [SII17] and [Nia+20] propose design of some average state observers. This approach is very beneficial in terms of reducing complexity. However, if there is an outlier, among the unmeasured states, which could be a result of an error or anomaly, the average value so estimated may not lead to the true average value. Instead, one might look for an average value that excludes the outlier that is a trimmed-average. For example, in a power distribution network, one might be interested in estimating the average household consumption of an area using dedicated sensors deployed in some households. In case, there is a major consumer (an outlier) in the area, whose consumption is not measured directly, one might not obtain the desired value through average estimation since its high power consumption can affect the average significantly. Therefore, it is natural to ask some questions: what if there is an outlier in the unmeasured section of the network? How to apply an estimation technique such that it filters the outlier and detects it simultaneously?

Outlier detection and analysis is a very well studied problem in statistics and data mining. Some of the classical works are [Haw80]; [RL87]. The former gives a formal accepted definition of the outlier and the latter proposes different regression-based detection methods.

[1] http://scale-freeback.eu/

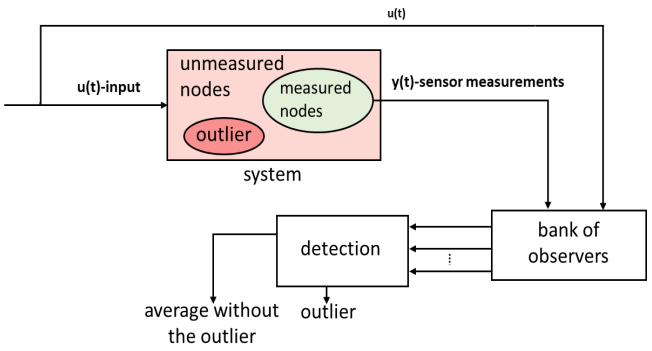


Figure 1.1: Scenario considered in this part: An LTI system with dedicated sensors at some positions and an outlier in the unmeasured part.

Since then, work in this field has flourished. A survey of the state-of-the-art can be found in the book [Agg17]. In particular, some of the methods include depth-based methods [ARR96], distance-based methods [KNT00] and k-nearest neighbour methods [HKF04]. However, these techniques apply to available static data points only. Here, in this part, we consider a network system with dynamics and moreover the aim is to detect the outlier present among the unavailable measurements.

In network systems also, there has been focus on outlier detection in sensor networks with outliers in the sensed data. The paper [ZMH10] is a survey on different techniques used for outlier detection in wireless sensor networks. For example, [Bra+06] addresses unsupervised detection in wireless sensor networks which accommodates different unsupervised techniques, [YYH12] focuses on neighbor based detection methods and [Wan+18] proposes a robust Kalman filter to detect and exclude the outlier from the sensor measurements which is somehow similar to our goal. However, the main difference is that in this literature, outliers are among sensor measurements, arising due to noise, error, sensor limitations, disturbances etc., while in our case the outlier is in the system itself, and is an unmeasured state which is so different from the other states that it affects the average value significantly.

The scenario in this part considers a continuous LTI system with dedicated sensors at some positions, and there exists an outlier in the set of unmeasured nodes. We propose a centralized method to detect the outlier and estimate the average, excluding it simultaneously. Our approach is to run a bank of observers and compare the estimates so obtained in order to detect the outlier, as illustrated in Figure 1.1. We first provide a necessary and sufficient condition under which a bank of scalar and tunable observers can be designed to estimate the average of the unmeasured nodes, excluding an element at every possible position. Then, we define a distance-based dissimilarity criterion to differentiate between the average estimates so obtained.

Using an optimization defined on the distance between the different estimates, we obtain an estimate for the outlier position and the average excluding it(trimmed-average). At first, we investigate the case where the system matrices are known and then the case where the system matrices are only partially known since the outlier results from a fault, and the only available

knowledge is the system model without the fault. For this, we consider a class of localized faults that result in a single outlier, for which we can extend our method. The advantage of this method is that we don't use the fault model information to design the observer. Finally, we illustrate the method on a thermal diffusion system.

Chapter 2

Outlier detection and trimmed-average estimation

Contents

2.1 Problem formulation 11
2.1.1 Motivating example 12
2.1.2 Problem Statement 14
2.2 Outlier at a known position 14
2.3 Outlier at an unknown position 18
2.3.1 Existence condition for the bank of observers 18
2.3.2 Outlier detection 20
2.3.2.1 Dissimilarity criterion 20
2.4 System model is partially unknown 22
2.5 Outlier detection in a faulty metal plate 30
2.5.1 Thermal diffusion system 30
2.5.2 Illustration of the detection method 33
2.6 Concluding remarks 35

When estimating the aggregated states in a network system, the presence of outliers can affect the estimated state significantly. Therefore, this chapter addresses the problem of outlier detection and trimmed-average state estimation in an linear time invariant(LTI) network system. We consider that only some states are measured and there exists an outlier among the unmeasured states, which is so different from the remaining states that it affects the average value significantly. The goal of this paper is both to detect the outlier and to estimate the average state excluding the outlier (trimmed-average). Moreover, we also investigate the case where the system matrices are partially unknown since the outlier results from an unknown localized fault in the system.

2.1 Problem formulation

Consider a network represented by a weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, 3 \dots n\}$ denotes the set of the nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges. We

follow the convention that the edge $(i,j) \in \mathcal{E}$ is represented as $i \leftarrow j$, since this edge will correspond to the influence of state x_j on the dynamics of state x_i. Let $A = [a_{ij}]$ be the associated weighted adjacency matrix, where a_{ij} is the weight of the edge $(i,j) \in \mathcal{E}$.

The dynamics of the network is described by

$$\begin{cases} \dot{\mathbf{x}}(t) = & A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{y}(t) = & C\mathbf{x}(t)\,, \end{cases} \tag{2.1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n, \mathbf{u}(t) \in \mathbb{R}^m$ and $\mathbf{y}(t) \in \mathbb{R}^{n_1}$ are the state vector, the input vector and the output vector respectively. We assume that the input is bounded and the system is BIBS stable. Therefore, we have bounded input and bounded state trajectories.

We assume to have dedicated sensor measurements at n_1 nodes, i.e., the output y contains the values of n_1 states. Without loss of generality, we order the states starting with the measured ones, so that we have the state partition

$$\mathbf{x}(t) = [\mathbf{x}_1^T(t), \mathbf{x}_2^T(t)]^T, \tag{2.2}$$

where the vector $\mathbf{y} = \mathbf{x}_1(t) \in \mathbb{R}^{n_1}$ contains the measured states and $\mathbf{x}_2(t) \in \mathbb{R}^{n_2}$ contains the unmeasured states. We assume that $n_2 > 1$. Denoting by $I_s \in \mathbb{R}^{s\times s}$ the identity matrix of size s, and by $\mathbf{0}_{s,r} \in \mathbb{R}^{s\times r}$ the zero matrix of size $s \times r$, the block structure of the matrices corresponding to the above-mentioned partition is

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},\ B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \text{ and } C = \begin{bmatrix} I_{n_1} & \mathbf{0}_{n_1,n_2} \end{bmatrix}. \tag{2.3}$$

With this partition, the system can be rewritten as

$$\begin{cases} \dot{\mathbf{x}}_1(t) &= A_{11}\mathbf{x}_1(t) + A_{12}\mathbf{x}_2(t) + B_1\mathbf{u}(t) \\ \dot{\mathbf{x}}_2(t) &= A_{21}\mathbf{x}_1(t) + A_{22}\mathbf{x}_2(t) + B_2\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{x}_1(t). \end{cases} \tag{2.4}$$

Having given the system description, we define the outlier in consideration as follows.

Definition 2.1. ***Outliers:*** *A state is called an outlier if its value differs from all the other states by such a large value that the average value changes significantly.*

2.1.1 Motivating example

This section presents an example to motivate the reader toward the problem. First, this example shows how an outlier may affect the average value of the unmeasured states. Then, we use the same example in the subsequent sections to illustrate the results and the method for outlier detection proposed in the chapter.

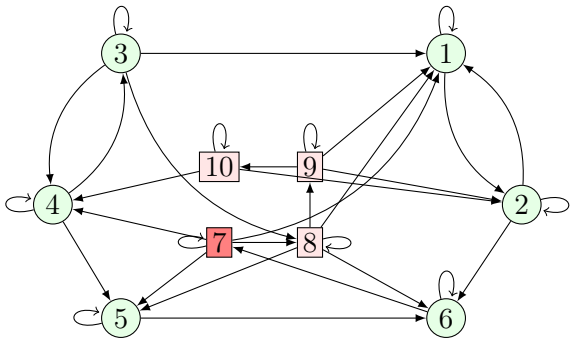


Figure 2.1: A network system with an outlier at node 7. The nodes in green circles are the measured nodes, and the nodes in red squares are the unmeasured nodes with the outlier in a darker red shade.

Example 2.1. *Consider the network depicted in Figure* 2.1. *The dynamics of the network is described as in* (2.1). *The input is given by* $u(t) = 0.2 * \sin(0.1t)$, $\mathbf{x}(0) = \mathbf{1}_{10}$, *where* $\mathbf{1}_s \in \mathbb{R}^s$ *denotes the vector of all ones. The corresponding system matrices* A, B, C *according to the partition in* (2.3) *are*

$$A_{11} = \begin{bmatrix} -3.25 & 0.98 & 0.84 & 0 & 0 & 0 \\ 0.61 & -5.33 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3.53 & 0.48 & 0 & 0 \\ 0 & 0 & 0.25 & -3.05 & 0 & 0 \\ 0 & 0 & 0 & 0.13 & -1.69 & 0 \\ 0 & 0.54 & 0 & 0 & 0.85 & -2.18 \end{bmatrix},$$

$$A_{12} = \begin{bmatrix} 0.35 & 0.35 & 0.35 & 0 \\ 0 & 0 & 0.76 & 0.76 \\ 0 & 0.22 & 0 & 0 \\ 0.23 & 0 & 0 & 0.23 \\ 0.16 & 0.16 & 0 & 0 \\ 0 & 0.35 & 0 & 0 \end{bmatrix}, \quad A_{21} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0.02 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$A_{22} = \begin{bmatrix} -0.03 & 0 & 0 & 0 \\ 0.02 & -1.4 & 0 & 0 \\ 0 & 0.64 & -1.16 & 0 \\ 0 & 0 & 0.4 & -2.0 \end{bmatrix}, \; B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T \text{ and}$$

$C = \begin{bmatrix} I_6 & \mathbf{0}_{6,4} \end{bmatrix}$.

Here, in this example, we have a network of $n = 10$ *nodes depicted in Figure 2.1. The sensor measurements are obtained from the nodes* $\{1, \ldots, 6\}$ *denoted by the circle nodes and the nodes* $\{7, \ldots, 10\}$ *are the unmeasured nodes denoted by the square nodes. Here,* $n_1 = 6$ *and* $n_2 = 4$. *We have an outlier at node* 7 *pointed as the shaded node. It can be seen in Figure* 2.3, *where the averages of the unmeasured states with and without the outlier have been depicted,*

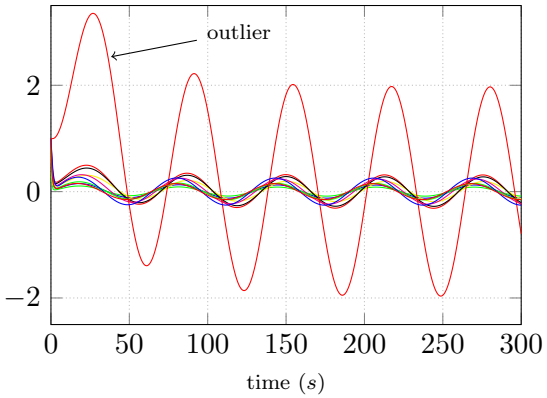


Figure 2.2: State trajectories of the network in Figure 1 in response to $u(t) = 0.2 * \sin(0.1t)$. Here, it can be seen that the outlier state (in red) is behaving very differently than the other states.

that there is a significant difference in the average value because of the outlier. Hence, we must find ways to estimate the average in such a way that it excludes the outlier.

2.1.2 Problem Statement

Consider a network with the dynamics as (2.1) assuming that there exists an outlier in the unmeasured part of the system. Using only the dedicated sensor measurements $\mathbf{y}(t) = \mathbf{x}_1(t)$, how is it possible to reconstruct the average of the unmeasured nodes $\mathbf{x}_2(t)$ without the outlier? In addition, is it possible to design a scalar observer to estimate such an average with an arbitrary rate of convergence? Moreover, if the position of the outlier is unknown, how is it possible to detect it and compute the average excluding it simultaneously? Finally, can the same result be achieved in the case where the system matrices are only partially known, since the outlier results from a fault, and the only available knowledge is the system without the fault?

2.2 Outlier at a known position

In this subsection, we consider the case when the position of the existing outlier is known. Let us define j_o as the true position of the outlier in the set of unmeasured nodes. The position j_o means that the outlier is the node $n_1 + j_o$.

At first, we recall the necessary and sufficient condition to design a scalar and tunable observer to estimate the average of all but one element j in the set of unmeasured nodes. Then, we provide an explicit construction of the observer. In the end, we illustrate the estimation using Example 2.1.

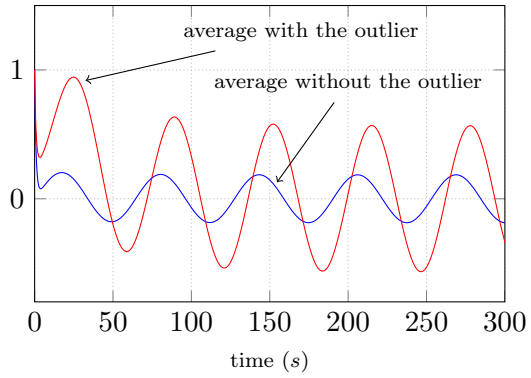


Figure 2.3: Trajectories of the average of the unmeasured states with and without the outlier, demonstrating the effect of an outlier on the average.

Let $c_j \in \mathbb{R}^{n_2}$ be a vector of all ones but zero at the jth position. Let us define $q_j = \frac{1}{n_2-1} c_j$, so that the average state of the unmeasured nodes excluding the element at the jth position be denoted as

$$x_{2,j}^{\text{av}}(t) = q_j^T \mathbf{x}_2(t), \tag{2.5}$$

where $\mathbf{x}_2(t)$ is defined in (2.2).

Definition 2.2. ***Trimmed-average*** *The average of the unmeasured states excluding the outlier denoted as* $x_{2,j_o}^{\text{av}}(t)$.

To reconstruct the average $x_{2,j}^{\text{av}}(t)$, we consider a scalar observer, namely a system of the form

$$\begin{cases} \dot{w}_j(t) & = -\alpha w_j(t) + h_j^T \mathbf{y}(t) + g_j^T \mathbf{u}(t) \\ \hat{x}_{2,j}^{\text{av}}(t) & = w_j(t) + \ell_j^T \mathbf{y}(t), \end{cases} \tag{2.6}$$

where $w_j(t) \in \mathbb{R}$ is the state of the observer, while $\alpha \in \mathbb{R}$, $\ell_j, h_j \in \mathbb{R}^{n_1}$ and $g_j \in \mathbb{R}^m$ will be suitably designed.

Let $\xi_j(t) = x_{2,j}^{\text{av}}(t) - \hat{x}_{2,j}^{\text{av}}(t)$ be the estimation error. We say that (2.6) is an observer if the parameters α, h_j, g_j and ℓ_j can be chosen such that $\xi_j(t) \to 0$ as $t \to \infty$. Moreover, we are interested in designing an observer such that the error $\xi_j(t) \to 0$ as $t \to \infty$ with a desired rate of convergence. The condition under which this is possible has been studied in [Nia+20].

Theorem 2.1. *[Nia+20, Theorem IV.1] A scalar and tunable observer of the form* (2.6) *to estimate* $x_{2,j}^{\text{av}}(t)$ *for a given* j *exists if and only if*

$$\text{rank} \begin{bmatrix} A_{12} \\ q_j^T A_{22} \\ q_j^T \end{bmatrix} = \text{rank}[A_{12}]. \tag{2.7}$$

Proof. For proof see [Nia+20]. □

Now, assuming that the condition (2.7) holds, we give the explicit design of the observer:

Choose an arbitrary $\alpha > 0 \in \mathbb{R}$ and compute the parameters ℓ_j^T, g_j^T and h_j^T as

$$\ell_j^T = q_j^T(A_{22} + \alpha I_{n_2})A_{12}^\dagger, \tag{2.8a}$$
$$h_j^T = -\ell_j^T(\alpha I_{n_1} + A_{11}) + q_j^T A_{21}, \tag{2.8b}$$
$$g_j^T = q_j^T B_2 - \ell_j^T B_1, \tag{2.8c}$$

where $A_{12}^\dagger$ is the Moore-Penrose pseudo-inverse of A_{12}.

In what follows, we show that with the above choice of parameters, the observer (2.6) has the error dynamics $\dot{\xi}_j(t) = -\alpha\xi_j(t)$. For that, let us consider the error dynamics $\dot{\xi}_j(t)$. From (2.4), (2.5) and (2.6) we have

$$\begin{aligned}\dot{\xi}_j(t) = -\alpha\xi_j(t) &+(-\alpha\ell_j^T + q_j^T A_{21} - \ell_j^T A_{11} - h_j^T)\mathbf{x}_1(t)\\ &+(q_j^T A_{22} - \ell_j^T A_{12} + \alpha q_j^T)\mathbf{x}_2(t)\\ &+(q_j^T B_2 - g_j^T - \ell_j^T B_1)\mathbf{u}(t).\end{aligned} \tag{2.9}$$

Assuming that condition (2.7) holds, and with ℓ_j, h_j and g_j as in (2.8), here, we will show that the following conditions are satisfied:

$$q_j^T A_{21} - \alpha\ell_j^T - \ell_j^T A_{11} - h_j^T = 0, \tag{2.10a}$$
$$\alpha q_j^T + q_j^T A_{22} - \ell_j^T A_{12} = 0, \text{ and} \tag{2.10b}$$
$$q_j^T B_2 - g_j^T - \ell_j^T B_1 = 0 \tag{2.10c}$$

and hence the estimation error ξ_j has stable dynamics

$$\dot{\xi}_j(t) = -\alpha\xi_j(t). \tag{2.11}$$

It can be seen that if the condition (2.7) holds, then the row vectors c_j and $c_j^T A_{22}$ lie in the row space of A_{12}. Then, the vector $\alpha q_j^T + q_j^T A_{22}$ also lies in the row space of A_{12} for any $\alpha \in \mathbb{R}$. Therefore, it holds that

$$(\alpha q_j^T + q_j^T A_{22})(I - A_{12}^\dagger A_{12}) = 0.$$

Hence,

$$\alpha q_j^T + q_j^T A_{22} = (\alpha q_j^T + q_j^T A_{22})A_{12}^\dagger A_{12}. \tag{2.12}$$

It can be seen that with the choice of ℓ_j^T in (2.8a), (2.12) is equivalent to (2.10b). Further, it can also be seen that with the choice of g_j and h_j in (2.8b) and (2.8c), the conditions (2.10a) and (2.10c) are satisfied.

We have shown that, under condition (2.7), the observer (2.6) with gains as in (2.8) has error dynamics (2.11), which is stable for arbitrary $\alpha > 0$. However, the rate of convergence α should be tuned taking into account the discretization scheme. For example, with forward Euler method with fixed time-step δt, α must satisfy $\alpha < 2/\delta t$ in order to ensure stability of the discretized error dynamics $\xi_j(t + \delta t) = (1 - \alpha\,\delta t)\xi_j(t)$. Moreover, the rate of convergence

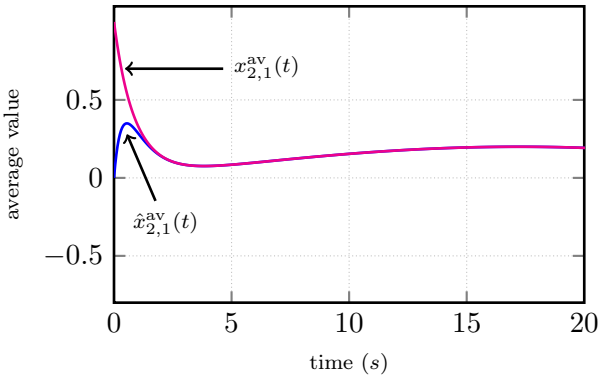


Figure 2.4: For the network system in Example 2.1, where the outlier is node 7 (i.e., $j_o = 1$): estimated ($\hat{x}^{\text{av}}_{2,1}(t)$) and original ($x^{\text{av}}_{2,1}(t)$) average of all unmeasured states except the outlier's.

should be adapted to the system properties, with a rule of thumb that suggests α around twice as fast as the system. In the examples of this chapter, we will take a smaller α for illustrative purposes, to make the transient behaviours more visible.

For initialization of the observer, in the absence of information on the initial state, it is natural to set $\hat{x}^{\text{av}}_{2,j}(0) = 0$, which can be obtained by choosing $w(0) = -\ell^T \mathbf{y}(0)$. The choice $w(0) = 0$, instead, corresponds to $\hat{x}^{\text{av}}_{2,j}(0) = \ell^T \mathbf{y}(0)$, which grows with α (see the definition of ℓ in (2.8a)) and might be used only with small values of α.

Now, we illustrate the estimation using Example 2.1.

Example 2.1 (continued)**.** *Here, we have the knowledge that the outlier is at node 7, which is the first node in the set of unmeasured nodes, i.e., $j_o = 1$. We want to estimate $x^{\text{av}}_{2,j_o}(t)$, i.e., $x^{\text{av}}_{2,1}(t)$. Note that the condition* (2.7) *with $j = 1$ is satisfied, as* $\text{rank} \begin{bmatrix} A_{12} \\ q_1^T A_{22} \\ q_1^T \end{bmatrix} = 4 = \text{rank}[A_{12}]$*, where $q_1^T = \left[0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right]$. Therefore, we can design an observer of the form* (2.6) *to estimate $x^{\text{av}}_{2,1}(t)$ with an arbitrary rate. Let us fix $\alpha = 3$ and use observer* (2.6) *with ℓ_1, g_1 and h_1 computed with* (2.8)*. We initialize the observer with $w_1(0) = -\ell_1^T y(0)$ so as to obtain $\hat{x}^{\text{av}}_{2,1}(0) = 0$.*

It can be seen in Figure 2.4 *that our observer is able to estimate the average $x^{\text{av}}_{2,1}(t)$, i.e., the average of unmeasured states excluding the outlier at node $n_1 + 1 = 7$.*

In the next subsection, we investigate the possibility of estimating the average using an observer of the type (2.6) in presence of an outlier at an arbitrary position, and we also give a method to detect the outlier.

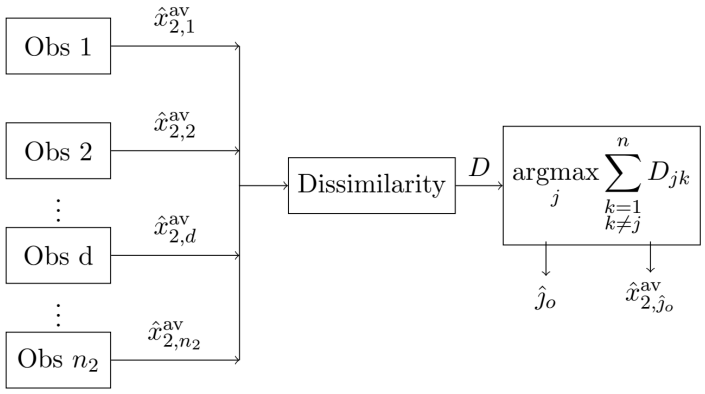


Figure 2.5: Structure of the estimation and detection algorithm presented in Section 2.3 for the detection of the outlier and trimmed-average estimation. Here, Obs stands for the observer and D is the dissimilarity matrix. $\hat{j}_o$ and $\hat{x}^{\text{av}}_{2,\hat{j}_o}$ are the detected position of the outlier and the average estimate excluding the outlier respectively(trimmed-average).

2.3 Outlier at an unknown position

In this section, we consider the case when the position of the outlier i.e., j_o is unknown. We propose a centralised method to estimate the average of the unmeasured nodes excluding the outlier and detecting it simultaneously. Our approach is to estimate the averages $x^{\text{av}}_{2,j}(t)$ for all possible $j \in \{1, \ldots, n_2\}$ using a bank of scalar observers and then compare the estimates in order to detect the outlier. For the comparison, we propose a dissimilarity criterion inspired by the distance-based dissimilarity used in signal processing. Figure 2.5 illustrates the process we use for the detection of the outlier which also gives us the required average. Therefore, at first, we provide a necessary and sufficient condition for the design of observers of the form (2.6) for every possible $j \in \{1, \ldots, n_2\}$. Then we define a dissimilarity criterion to distinguish between the estimates. After that, we define a very general optimization problem to detect the outlier. In the end, we illustrate the method with the help of Example 2.1.

2.3.1 Existence condition for the bank of observers

In order to design observers of the form (2.6) to reconstruct $x^{\text{av}}_{2,j}(t)$ for all $j \in \{1, \ldots, n_2\}$, the condition (2.7) must be satisfied for all $j \in \{1, \ldots, n_2\}$. The resulting condition has been stated in the following theorem.

Theorem 2.2. *A tunable and scalar observer for $x^{\text{av}}_{2,j}(t)$ exists for all $j \in \{1, \ldots, n_2\}$ if and only if*

$$\text{rank}(A_{12}) = n_2. \tag{2.13}$$

Proof. From Theorem 2.1, to estimate $x^{\text{av}}_{2,j}(t)$ for all j, a tunable, scalar observer exists if and only if (2.7) holds for all j.

This is equivalent to

$$\operatorname{rank}\begin{bmatrix} A_{12} \\ q_1^T A_{22} \\ \vdots \\ q_{n_2}^T A_{22} \\ q_1^T \\ \vdots \\ q_{n_2}^T \end{bmatrix} = \operatorname{rank}[A_{12}]. \tag{2.14}$$

Now define $P = (I_{n_2} - \mathbf{1}_{n_2}\mathbf{1}_{n_2}^T)$, and $Q = -\dfrac{1}{n_2 - 1}P = \begin{bmatrix} q_1^T \\ \vdots \\ q_{n_2}^T \end{bmatrix}$, so that the left-hand side of (2.14) is equal to $\begin{bmatrix} A_{12} \\ QA_{22} \\ Q \end{bmatrix}$. We can see that $\operatorname{rank}(Q) = n_2$ by showing that $\operatorname{rank}(P) = n_2$ and we show $\operatorname{rank}(P) = n_2$ by showing that $\det P \neq 0$.
For this, notice that

$$\begin{bmatrix} I_{n_2} & \mathbf{0}_{n_2,1} \\ \mathbf{1}_{n_2}^T & 1 \end{bmatrix}\begin{bmatrix} P & -\mathbf{1}_{n_2} \\ \mathbf{0}_{1,n_2} & 1 \end{bmatrix}\begin{bmatrix} I_{n_2} & \mathbf{0}_{n_2,1} \\ -\mathbf{1}_{n_2}^T & 1 \end{bmatrix} = \begin{bmatrix} I_{n_2} & -\mathbf{1}_{n_2} \\ \mathbf{0}_{1,n_2} & 1 - \mathbf{1}_{n_2}^T\mathbf{1}_{n_2} \end{bmatrix}.$$

Taking determinant of matrices on both the sides, we have $\det(P) = 1 - n_2 \neq 0$.
Therefore, $\operatorname{rank}(P) = n_2$ and hence, $\operatorname{rank}(Q) = n_2$.
Finally, since $\operatorname{rank}(Q) = n_2$, we have $\operatorname{rank}\begin{bmatrix} A_{12} \\ QA_{22} \\ Q \end{bmatrix} = n_2$. Therefore, (2.14) holds if and only if $\operatorname{rank}(A_{12}) = n_2$. □

Now we present some remarks on how restrictive the conditions in Theorem 2.2 are.

Remark 2.1. *It can be seen from* (2.3)*, that* $A_{12} \in \mathbb{R}^{n_1 \times n_2}$ *and the condition* (2.13) *is* $\operatorname{rank}(A_{12}) = n_2$*. It implies* $n_1 \geq n_2$*, that is the number of measured nodes must be greater than or equal to the number of unmeasured nodes.*

Remark 2.2. *From the condition* (2.13)*,* A_{12} *is full column rank so it cannot have an all zero column. Therefore, Theorem* 2.2 *requires that for every unmeasured node, there exists an edge pointing to some measured node, i.e., for every unmeasured node* j *there is an edge* (i, j) *with* i *a measured node (recall that the edge* (i, j) *is depicted as an arrow* $j \to i$*, representing an influence of state* x_j *on the dynamics of state* x_i*).*

Now, we proceed towards the problem of detection of the outlier in the next section. For this, we propose to run a bank of observers and then compare the estimates we obtain.

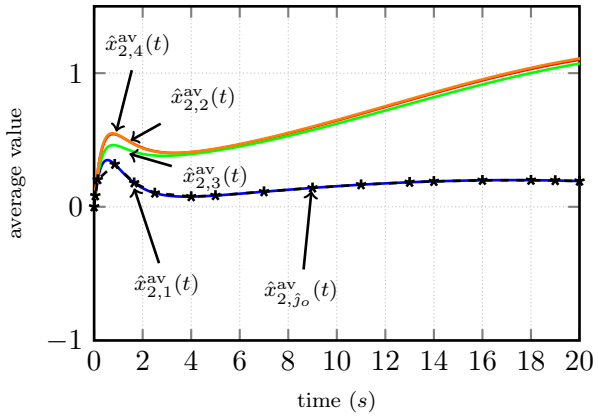


Figure 2.6: Estimated trajectories of the averages of unmeasured nodes in Example 2.1 excluding one node at a time i.e. $\hat{x}^{\text{av}}_{2,j}(t)\ \forall j \in \{1,\dots,n_2\}$, the output of the bank of observers. The dash-starred line is $\hat{x}^{\text{av}}_{2,\hat{j}_o}(t)$, the estimated average without the detected outlier at time t. It can be seen that $\hat{x}^{\text{av}}_{2,\hat{j}_o}(t)$ converges to $\hat{x}^{\text{av}}_{2,1}(t)$ very quickly as the outlier is at $j_o = 1$.

2.3.2 Outlier detection

In this subsection, we define a dissimilarity matrix and an optimization problem in order to detect the outlier.

2.3.2.1 Dissimilarity criterion

Here, we define a dissimilarity criterion in order to distinguish between the estimates obtained in the previous section with a goal in mind to pick the one which is without the outlier. Dissimilarity criteria of this kind are used in signal processing. For instance, [XSB14] defines dissimilarity as pairwise Euclidean distance between two signals.

Consider the estimates $\hat{x}^{\text{av}}_{2,j}(t)\ \forall j \in \{1,\dots,n_2\}$ obtained in the previous section, we define their dissimilarity at time t as

$$D_{jk}(t) = \int_0^t e^{-\beta(t-\tau)} \mid \hat{x}^{\text{av}}_{2,j}(\tau) - \hat{x}^{\text{av}}_{2,k}(\tau) \mid \mathrm{d}\tau \quad \text{for } \beta > 0, \tag{2.15}$$

where $\hat{x}^{\text{av}}_{2,j}(\tau)$ is the average estimate of the unmeasured nodes except the node j at time τ.

This definition seems to require all the average estimates from $\tau = 0$ to the current time $\tau = t$, but this integral can be computed recursively as

$$D_{jk}(t) = e^{-\beta\delta t} D_{jk}(t-\delta t) + \int_{t-\delta t}^{t} e^{-\beta(t-\tau)} \mid \hat{x}^{\text{av}}_{2,j}(\tau) - \hat{x}^{\text{av}}_{2,k}(\tau) \mid \mathrm{d}\tau,$$

which might be simplified with a suitable approximation, e.g.,

$$D_{jk}(t) \simeq e^{-\beta\delta t} D_{jk}(t-\delta t) + \delta t \mid \hat{x}^{\text{av}}_{2,j}(t) - \hat{x}^{\text{av}}_{2,k}(t) \mid .$$

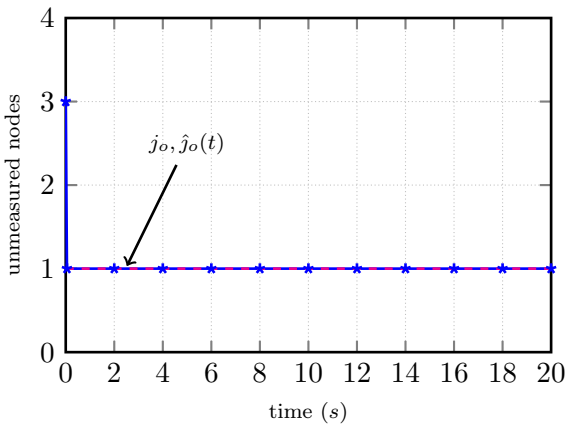


Figure 2.7: Detected position of the outlier in the set of the unmeasured nodes at time t i.e. $\hat{j}_o(t)$ in Example 2.1. $j_o = 1$ is the actual position of the outlier. At first, the method identifies unmeasured node 3 as the outlier but it converges very quickly to j_o.

The matrix $D = [D_{jk}]$ so obtained is called the *Dissimilarity matrix*. Note that D is a non-negative, symmetric matrix with zero diagonal elements. Here, the idea is to measure how far are the estimates from each other.

Rather than instantaneous comparisons, we consider an integral, so as to consider only differences that hold over non-trivial windows of time. The forgetting factor β tunes the weight given to past values, and should be chosen avoiding the two extremes: β near to zero gives too much weight to the initial transient, where the estimates might have large errors, while too large β gives vanishing weight to all past values.

The system is assumed to have an outlier at j_o, i.e., there is a significant difference between the average $x^{\text{av}}_{2,j_o}(t)$ excluding j_o and the average $x^{\text{av}}_{2,k}(t)$ excluding any other node k. Moreover, the outlier is unique. Hence, for any j and k different from j_o, $|x^{\text{av}}_{2,j_o}(t) - x^{\text{av}}_{2,k}(t)|$ is large and $|x^{\text{av}}_{2,j}(t) - x^{\text{av}}_{2,k}(t)|$ is small (at least as an integral over time, as in the dissimilarity matrix). Since each estimate $\hat{x}^{\text{av}}_{2,j}(t)$ converges to the corresponding correct average $x^{\text{av}}_{2,j}(t)$, we also have that $|\hat{x}^{\text{av}}_{2,j_o}(t) - \hat{x}^{\text{av}}_{2,k}(t)|$ is large and $|\hat{x}^{\text{av}}_{2,j}(t) - \hat{x}^{\text{av}}_{2,k}(t)|$ is small, except possibly for an initial transient. For this reason, we can say that $\hat{j}_o(t)$ is the detected position of the outlier at time t if $\hat{j}_o(t)$th row sum of the dissimilarity matrix at time t is the largest:

$$\hat{j}_o(t) = \underset{j}{\text{argmax}} \sum_{\substack{k=1 \\ k \neq j}}^{n} D_{jk}(t). \tag{2.16}$$

The above processes as depicted in Figure 2.5 can be put altogether as follows:

1. Using a bank of n_2 observers of the form (2.6), compute $\hat{x}^{\text{av}}_{2,j}(t)$ for all j from 1 to n_2.

2. Compute the dissimilarity matrix D defined in (2.15).

3. From the dissimilarity matrix, detect the outlier $\hat{\jmath}_o(t)$ at time t given by (2.16). Then choose the corresponding average estimate $\hat{x}^{\text{av}}_{2,\hat{\jmath}_o}(t)$ obtained from the bank of observers, which excludes $x_{n_1+\hat{\jmath}_o}(t)$.

Now, we illustrate the method with Example 2.1. We will see that indeed $\hat{\jmath}_o(t)$ converges to j_o, the actual position of the outlier.

Example 2.1 (continued)**.** *The position of the outlier in Example* 2.1 *is* $j_o = 1$ *but here we assume that this information is unknown. We want to detect the outlier position and obtain the average estimate excluding the outlier. Note that the condition* (2.13) *is satisfied with* $\text{rank}(A_{12}) = 4 = n_2$*. Therefore, we can design a bank of* n_2 *observers of the form* (2.6) *each of which estimates the average of all but one unmeasured node at a time. Here, each observer is designed with* $\alpha = 3$*, and the parameters* ℓ_j^T, g_j^T, h_j^T *computed with* (2.8) *and initialization* $w_j(0) = -\ell_j^T y(0)$*. The estimates obtained by this bank of observers are depicted in Figure 2.6. We compute the dissimilarity matrix with* $\beta = 10$ *in* (2.15) *and then follow the process described above to detect the outlier using* (2.16)*. Figure* 2.7 *shows that indeed the proposed method detects the outlier position* $j_o = 1$*, i.e., node* $n_1 + j_o = 7$ *in the network depicted in Figure 2.1. In Figure 2.7, it can be seen that at first, the method identifies unmeasured node 3 (node 9 in the network) as the outlier but it converges very quickly to the actual position, i.e.,* $j_o = 1$ *(node* $n_1 + j_o = 7$ *in the network). One possible reason for this delay in detection could be the delay in convergence of the observed value to the original value.*

Note that we already have the average estimates $\hat{x}^{\text{av}}_{2,j}(t)$ *for all* $j \in \{1, \ldots, n_2\}$*, obtained from the bank of observers. From them, we obtain* $\hat{x}^{\text{av}}_{2,\hat{\jmath}_o}(t)$*, which is initially equal to* $\hat{x}^{\text{av}}_{2,3}(t)$ *and then equal to* $\hat{x}^{\text{av}}_{2,1}(t)$*, as illustrated by the dash-starred black line in Figure 2.6. This is consistent with the quantity we aim at reconstructing: the average estimate of the unmeasured nodes excluding the outlier, i.e.,* $\hat{x}^{\text{av}}_{2,1}(t)$*.*

2.4 System model is partially unknown

In this section, we consider the case when there exists an outlier in the system but the system matrices in (2.1) are partially unknown. Assuming that this outlier is caused by a fault, we call the initial system which is without the fault, the nominal system. We aim to detect the outlier and to compute the average without the outlier using the detection method proposed in Section 2.3 even when the system with the fault is unknown, and only the nominal system is known. Recall that, in the proposed detection method, we deploy a bank of observers of the form (2.6), designed with the faulty system matrices. However, in this section, since the faulty system is unknown, we propose to design the observers with the nominal system.

Let us define the nominal system as

$$\begin{cases} \dot{\mathbf{x}}(t) = & \tilde{A}\mathbf{x}(t) + \tilde{B}\mathbf{u}(t) \\ \mathbf{y}(t) = & C\mathbf{x}(t), \end{cases} \tag{2.17}$$

which describes the same physical system as system (2.1) but without the fault causing the outlier. The system (2.1) and the system (2.17) are related in such a way that

$$A = \tilde{A} + \Delta \text{ and } B = \tilde{B} + \Psi, \tag{2.18}$$

where $\Delta \in \mathbb{R}^{n\times n}$ and $\Psi \in \mathbb{R}^{n\times m}$ are the matrices which describe the fault responsible for the outlier.

The block structure of the matrices corresponding to the state partition $x(t) = [x_1^T(t), x_2^T(t)]^T$ as in Section 2.1 is

$$\begin{aligned} \tilde{A} &= \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix}, \ \Delta = \begin{bmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{bmatrix}, \\ \tilde{B} &= \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \end{bmatrix}, \ \Psi = \begin{bmatrix} \Psi_1 \\ \Psi_2 \end{bmatrix}, \ C = \begin{bmatrix} I_{n_1} & \mathbf{0}_{n_1,n_2} \end{bmatrix}. \end{aligned} \tag{2.19}$$

Let us consider the following fault model:

$$\text{Fault model: } \begin{cases} \Delta = e_{n_1+j_o} r^T + \rho \tilde{A} e_{n_1+j_o} e_{n_1+j_o}^T, \\ \Psi = e_{n_1+j_o} \psi^T. \end{cases} \tag{2.20}$$

where $e_i \in \mathbb{R}^n$ is the standard ith basis vector, $\rho \in \mathbb{R}$ is a scalar, $r^T \in \mathbb{R}^n$ and $\psi^T \in \mathbb{R}^m$ are row vectors, that can be arbitrary, n_1 is the number of measured nodes and j_o is the true position of the outlier. This means that Δ is a matrix which is non-zero only in its (n_1+j_o)th row and column. Moreover, its (n_1+j_o)th column is proportional to the corresponding column of $\tilde{A}$, while no such restriction is assumed on its (n_1+j_o)th row, which can be arbitrary. The matrix Ψ is a matrix with non-zero entries only in its (n_1+j_o)th row which is equal to the row vector ψ^T.

In terms of the network system, this corresponds to altering only the neighborhood of node n_1+j_o. More precisely, the row vector r^T represents the change in the influence of the in-neighbors of the node (n_1+j_o) on it. It allows for arbitrary changes in the incoming edges and in their weights. The term $\rho\tilde{A}e_{n_1+j_o}e_{n_1+j_o}^T$ in (2.20) represents the change in the influence of node (n_1+j_o) on its out-neighbors. Due to this term, all entries of the (n_1+j_o)th column of A, other than the one on the diagonal, are equal to the corresponding entry of $\tilde{A}$, multiplied by $(1+\rho)$. This means that the outgoing edges from n_1+j_o are unchanged, and their weights are all multiplied by a same scalar factor $(1+\rho)$, which describes a change in the strength of the influence of node n_1+j_o on its out-neighbors. The row vector ψ^T changes the effect of the input, again in a localized way, only on the node n_1+j_0.

Now, we illustrate the nominal system and the fault model with an example.

Example 2.2. *Consider the network depicted in Figure* 2.1*. The input is given by* $u(t) = 0.2 * \sin(0.1t)$ *and* $x(0) = \mathbf{1}_{10}$*. Here, we give a nominal system with no outlier, and a fault as in* (2.20)*, such that the resulting faulty system is the system with an outlier that was presented in Example 2.1. The nominal system matrix* $\tilde{A}$ *has the following blocks (according to the partition in* (2.19)*):*

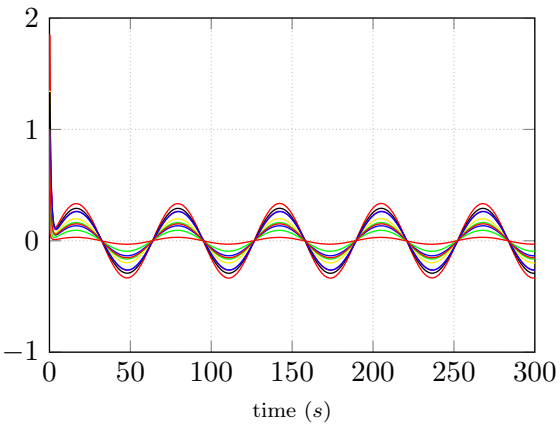


Figure 2.8: State trajectories of the nominal system described in Example 2.

$$\tilde{A}_{11} = \begin{bmatrix} -3.25 & 0.98 & 0.84 & 0 & 0 & 0 \\ 0.61 & -5.33 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3.53 & 0.48 & 0 & 0 \\ 0 & 0 & 0.25 & -3.05 & 0 & 0 \\ 0 & 0 & 0 & 0.13 & -1.69 & 0 \\ 0 & 0.54 & 0 & 0 & 0.85 & -2.18 \end{bmatrix},$$

$$\tilde{A}_{12} = \begin{bmatrix} 17.15 & 0.35 & 0.35 & 0 \\ 0 & 0 & 0.76 & 0.76 \\ 0 & 0.22 & 0 & 0 \\ 11.27 & 0 & 0 & 0.23 \\ 7.84 & 0.16 & 0 & 0 \\ 0 & 0.35 & 0 & 0 \end{bmatrix},$$

$$\tilde{A}_{21} = \begin{bmatrix} 0 & 0.89 & 0 & 0 & 0 & 0.16 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\tilde{A}_{22} = \begin{bmatrix} -11.76 & 0 & 0 & 0 \\ 0.98 & -1.4 & 0 & 0 \\ 0 & 0.64 & -1.16 & 0 \\ 0 & 0 & 0.4 & -2.0 \end{bmatrix}.$$

Matrices B and C are the same as in Example 2.1, i.e., $B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T$ *and* $C = \begin{bmatrix} I_6 & \mathbf{0}_{6,4} \end{bmatrix}$.

It can be seen in Figure 2.8 that there is no outlier in the nominal system. Then we consider a faulty system matrix $A = \tilde{A} + \Delta$ and $B = \tilde{B} + \Psi$, where Δ and Ψ are given by (2.20), with $j_o = 1$, $r^T = [0, -0.89, 0, 0, 0, -0.14, -11.55, 0, 0, 0]$, $\rho = -\frac{48}{49}$ and $\Psi = \mathbf{0}_{n,1}$. This

faulty system is the system considered in Example 2.1, where the state $n_1 + j_o = 7$ is an outlier as shown in Figure 2.2.

Now, we proceed towards the problem of detecting the outlier while designing the observers with the known nominal system instead of the faulty system. Recall that, to estimate $x^{\text{av}}_{2,j}(t)$, the average of the unmeasured nodes excluding the element at the jth position, we use observers of the form (2.6) with observer gains obtained by fixing $\alpha > 0$ and computing:

$$\ell_j^T = q_j^T(\tilde{A}_{22} + \alpha I_{n_2})\tilde{A}_{12}^{\dagger}, \tag{2.21a}$$
$$h_j^T = -\ell_j^T(\alpha I_{n_1} + \tilde{A}_{11}) + q_j^T\tilde{A}_{21}, \tag{2.21b}$$
$$g_j^T = q_j^T\tilde{B}_2 - \ell_j^T\tilde{B}_1, \tag{2.21c}$$

where $\tilde{A}_{12}^{\dagger}$ is the Moore-Penrose pseudo-inverse of $\tilde{A}_{12}$,

Now, we present a lemma to describe an interesting property of the error dynamics when the observer gains are computed as in (2.21).

Lemma 2.1. *With the fault model given by* (2.20) *and* $\operatorname{rank}(\tilde{A}_{12}) = n_2$*, given that the bank of observers are of the form* (2.6) *and the observers gains computed as in* (2.21)*, then the estimation error* $\xi_j(t) = \hat{x}^{\text{av}}_{2,j}(t) - x^{\text{av}}_{2,j}(t)$ *is such that its dynamics is as follows:*

$$\dot{\xi}_j(t) = -\alpha\xi_j(t) + \begin{cases} 0 & j = j_o \\ \frac{1}{n_2-1}\big(r_1^T x_1(t) + (r_2^T - \rho\alpha e_{j_o}^T)x_2(t) + \psi^T\mathbf{u}(t)\big) & j \neq j_o. \end{cases} \tag{2.22}$$

Proof. Let us consider the estimation error $\xi_j(t)$.

$$\begin{aligned}\dot{\xi}_j(t) = &-\alpha\xi_j(t)\\ &+(q_j^T(\tilde{A}_{21} + \Delta_{21}) - \alpha\ell_j^T - \ell_j^T(\tilde{A}_{11} + \Delta_{11}) - h_j^T)\mathbf{x}_1(t)\\ &+(q_j^T(\tilde{A}_{22} + \Delta_{22}) - \ell_j^T(\tilde{A}_{12} + \Delta_{12}) + \alpha q_j^T)\mathbf{x}_2(t)\\ &+(q_j^T(\tilde{B}_2 + \Psi_2) - g_j^T - \ell_j^T(\tilde{B}_1 + \Psi_1))\mathbf{u}(t).\end{aligned}$$

Rearranging the equation by separating the terms with $\Delta_{11}, \Delta_{12}, \Delta_{21}$, Δ_{22}, Ψ_1 and Ψ_2, we have

$$\begin{aligned}\dot{\xi}_j(t) = -\alpha\xi_j(t) &+(q_j^T\tilde{A}_{21} - \alpha\ell_j^T - \ell_j^T\tilde{A}_{11} - h_j^T)\mathbf{x}_1(t)\\ &+(q_j^T\tilde{A}_{22} - \ell_j^T\tilde{A}_{12} + \alpha q_j^T)\mathbf{x}_2(t)\\ &+(q_j^T\tilde{B}_2 - g_j^T - \ell_j^T\tilde{B}_1)\mathbf{u}(t)\\ &+(q_j^T\Delta_{21} - \ell_j^T\Delta_{11})\mathbf{x}_1(t)\\ &+(q_j^T\Delta_{22} - \ell_j^T\Delta_{12})\mathbf{x}_2(t)\\ &+(q_j^T\Psi_2 - g_j^T - \ell_j^T\Psi_1)\mathbf{u}(t).\end{aligned} \tag{2.23}$$

As shown in Section 2.2, the rank condition $\operatorname{rank}(\tilde{A}_{12}) = n_2$ ensures that computing ℓ_j, g_j and h_j as in (2.21), we have

$$-\alpha\ell_j^T + q_j^T\tilde{A}_{21} - \ell_j^T\tilde{A}_{11} - h_j^T = 0, \tag{2.24a}$$
$$+\alpha q_j^T + q_j^T\tilde{A}_{22} - \ell_j^T\tilde{A}_{12} = 0, \text{and} \tag{2.24b}$$
$$q_j^T\tilde{B}_2 - g_j^T - \ell_j^T\tilde{B}_1 = 0. \tag{2.24c}$$

Therefore, from (2.23) and (2.24), we have

$$\begin{aligned}\dot{\xi}_j(t) = -\alpha\xi_j(t) &+(q_j^T\Delta_{21} - \ell_j^T\Delta_{11})\mathbf{x}_1(t)\\ &+(q_j^T\Delta_{22} - \ell_j^T\Delta_{12})\mathbf{x}_2(t)\\ &+(q_j^T\Psi_2 - g_j^T - \ell_j^T\Psi_1)\mathbf{u}(t).\end{aligned} \tag{2.25}$$

From (2.20), using the notation $r^T = [r_1^T, r_2^T]$, we have

$$\begin{aligned}\Delta_{11} &= \mathbf{0}_{n_1,n_1}, & \Psi_1 &= \mathbf{0}_{n_1,1},\\ \Delta_{12} &= \rho\tilde{A}_{12}e_{j_o}e_{j_o}^T, & &\\ \Delta_{21} &= e_{j_o}r_1^T, & \Psi_2 &= e_{j_o}\psi^T,\\ \Delta_{22} &= e_{j_o}r_2^T + \rho\tilde{A}_{22}e_{j_o}e_{j_o}^T. & &\end{aligned}$$

Substituting these values in (2.25), we have

$$\begin{aligned}\dot{\xi}_j(t) = -\alpha\xi_j(t) &+(q_j^Te_{j_o}r_1^T)\mathbf{x}_1(t) + (q_j^Te_{j_o}r_2^T)\mathbf{x}_2(t)\\ &+\rho\left(q_j^T\tilde{A}_{22} - \ell_j^T\tilde{A}_{12}\right)e_{j_o}e_{j_o}^T\mathbf{x}_2(t)\\ &+(q_j^Te_{j_o}\psi^T)\mathbf{u}(t).\end{aligned} \tag{2.26}$$

Now, let us consider the second last term of (2.26). From (2.24b), it can be written as

$$\rho(q_j^T\tilde{A}_{22} - \ell_j^T\tilde{A}_{12})e_{j_o}e_{j_o}^T = -\rho\alpha q_j^Te_{j_o}e_{j_o}^T.$$

Notice that, if $j = j_o$, we have $q_j^Te_{j_o} = 0$, while for all $j \neq j_o$ we have $q_j^Te_{j_o} = \frac{1}{n_2-1}$. Therefore, we conclude that the coefficient of $x_1(t)$ in (2.26) is

$$q_j^Te_{j_o}r_1^T = \begin{cases}\mathbf{0}_{1,n_1} & \text{if } j = j_o\\ \frac{1}{n_2-1}r_1^T & \text{if } j \neq j_o,\end{cases} \tag{2.27}$$

the coefficient of $x_2(t)$ in (2.26) is

$$q_j^Te_{j_o}r_2^T - \rho\alpha q_j^Te_{j_o}e_{j_o}^T = \begin{cases}\mathbf{0}_{1,n_2} & \text{if } j = j_o\\ \frac{1}{n_2-1}(r_2^T - \rho\alpha e_{j_o}^T) & \text{if } j \neq j_o.\end{cases} \tag{2.28}$$

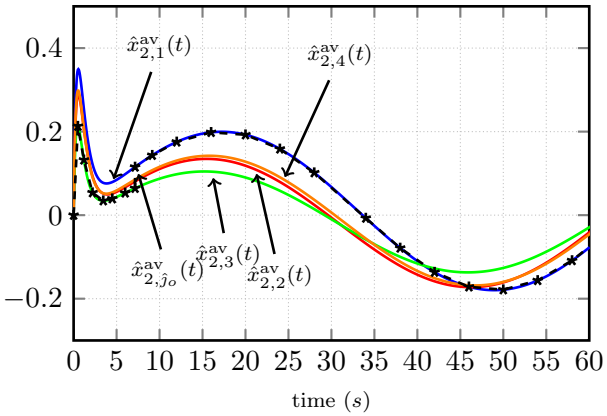


Figure 2.9: Estimated average trajectories $\hat{x}^{\text{av}}_{2,j}(t)\ \forall j \in \{1,\dots,n_2\}$, in Example 2.2. The observer gains were computed using nominal matrices instead of the faulty matrices. The dashed-star line represents $\hat{x}^{\text{av}}_{2,\hat{j}_o}(t)$.

and the coefficient of $u(t)$ in (2.26) is

$$q_j^T e_{j_o} \psi^T = \begin{cases} \mathbf{0}_{1,m} & \text{if } j = j_o \\ \frac{1}{n_2-1}\psi^T & \text{if } j \neq j_o, \end{cases} \tag{2.29}$$

which ends the proof of the lemma. □

Now, we comment on the average estimates in the following theorem.

Theorem 2.3. *With the fault model given by* (2.20) *and* $\text{rank}(\tilde{A}_{12}) = n_2$*, given that the bank of observers are of the form* (2.6) *and the observer gains are computed as* (2.21)*, then*

i.) for $j = j_o$, $\hat{x}^{\text{av}}_{2,j_o}(t) \to x^{\text{av}}_{2,j_o}(t)$ *as* $t \to \infty$.

ii.) for all $j, k \neq j_o$, $(\hat{x}^{\text{av}}_{2,j}(t) - \hat{x}^{\text{av}}_{2,k}(t)) \to (x^{\text{av}}_{2,j}(t) - x^{\text{av}}_{2,k}(t))$ *as* $t \to \infty$.

Proof. i.) It follows immediately from Lemma 2.1, since $\xi_{j_o}(t) = x^{\text{av}}_{2,j_o}(t) - \hat{x}^{\text{av}}_{2,j_o}(t)$ has a stable dynamics $\dot{\xi}_{j_o}(t) = -\alpha\xi_{j_o}(t)$.

ii.) We study the difference $(x^{\text{av}}_{2,j}(t) - x^{\text{av}}_{2,k}(t)) - (\hat{x}^{\text{av}}_{2,j}(t) - \hat{x}^{\text{av}}_{2,k}(t)) = \xi_j(t) - \xi_k(t)$. From (2.22), for $j, k \neq j_o$, such difference has stable dynamics

$$\dot{\xi}_j(t) - \dot{\xi}_k(t) = -\alpha(\xi_j(t) - \xi_k(t)).$$

□

In Section 2.3, the observers were designed using the knowledge of the matrices A and B, and hence it was possible to ensure convergence of the estimates $\hat{x}^{\text{av}}_{2,j}(t)$ to the true averages

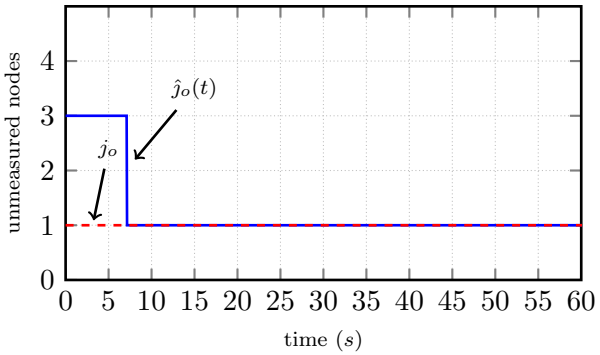


Figure 2.10: Detected position of the outlier in the set of the unmeasured nodes at time t i.e. $\hat{j}_o(t)$ in Example 2.2. $j_o = 1$ is the actual position of the outlier. At first, the method identifies node 3 as the outlier but it converges quickly to j_o.

$x^{\text{av}}_{2,j}(t)$. Here, the observers are designed with the nominal matrices $\tilde{A}$ and $\tilde{B}$, while the true averages are from the faulty system with matrices A and B, so in general convergence of the estimates cannot be ensured. However, Theorem 2.3 gives two important convergence results.

First, for the estimate of the average without the outlier ($\hat{x}^{\text{av}}_{2,j_o}(t)$), convergence to the true average $x^{\text{av}}_{2,j_o}(t)$ is still ensured, despite the use of matrix $\tilde{A}$ instead of the unknown matrix A. Hence, in case the outlier position j_o is known or can be correctly detected, then the trimmed-average $x^{\text{av}}_{2,j_o}(t)$ can be correctly estimated.

Second, for j and k different from j_o, although the estimates $\hat{x}^{\text{av}}_{2,j}(t)$ and $\hat{x}^{\text{av}}_{2,k}(t)$ can be wrong with respect to the true averages $x^{\text{av}}_{2,j}(t)$ and $x^{\text{av}}_{2,k}(t)$, Theorem 2.3 ensures that their difference $(\hat{x}^{\text{av}}_{2,j}(t) - \hat{x}^{\text{av}}_{2,k}(t))$ converges to the true difference $(x^{\text{av}}_{2,j}(t) - x^{\text{av}}_{2,k}(t))$.

Moreover, since $u(t)$ and $x(t)$ are assumed to remain bounded, we can see from (2.22) that the estimates $\hat{x}^{\text{av}}_{2,j}(t)$ and $\hat{x}^{\text{av}}_{2,k}(t)$ remain bounded, which avoids overflow issues in numerical computation.

To see the importance of these two convergence results for our detection method, recall that we use a bank of scalar observers and the dissimilarity matrix defined in (2.15), which involves pairwise differences of estimates $|\hat{x}^{\text{av}}_{2,j}(t) - \hat{x}^{\text{av}}_{2,k}(t)|$. In this chapter, we restrict our attention to systems where a localized fault at an unmeasured node results in the appearence of a single outlier state trajectory, corresponding to such node. As discussed in Sect. 2.3.2.1, this implies that, for any j and k different from j_o, $|x^{\text{av}}_{2,j_o}(t) - x^{\text{av}}_{2,k}(t)|$ is large and $|x^{\text{av}}_{2,j}(t) - x^{\text{av}}_{2,k}(t)|$ is small (at least as an integral over time). Then, our detection method is capable of reconstructing the correct outlier position j_o if these two properties of the true averages are preserved for the estimated averages.

For j and k different from j_o, the second part of Theorem 2.3 ensures that the difference $(\hat{x}^{\text{av}}_{2,j}(t) - \hat{x}^{\text{av}}_{2,k}(t))$ converges to the true difference $(x^{\text{av}}_{2,j}(t) - x^{\text{av}}_{2,k}(t))$. Hence, the property that the true difference $|x^{\text{av}}_{2,j}(t) - x^{\text{av}}_{2,k}(t)|$ is small implies that also the estimated difference $|\hat{x}^{\text{av}}_{2,j}(t) - \hat{x}^{\text{av}}_{2,k}(t)|$ is small.

The case of the differences $|\hat{x}^{\text{av}}_{2,j_o}(t) - \hat{x}^{\text{av}}_{2,k}(t)|$ involving the outlier is more delicate. The first part of Theorem 2.3 ensures that $\hat{x}^{\text{av}}_{2,j_o}(t)$ converges to the true $x^{\text{av}}_{2,j_o}(t)$, but gives no information about $\hat{x}^{\text{av}}_{2,k}(t)$. In most cases, as illustrated in our examples, $\hat{x}^{\text{av}}_{2,k}(t)$ is a wrong estimate of $x^{\text{av}}_{2,k}(t)$, but the difference $|\hat{x}^{\text{av}}_{2,j_o}(t) - \hat{x}^{\text{av}}_{2,j}(t)|$ remains large, thus leading to a correct detection of the outlier position. When this happens, also the correct reconstruction of $x^{\text{av}}_{2,j_o}(t)$ is ensured, since $\hat{x}^{\text{av}}_{2,j_o}(t)$ converges to $x^{\text{av}}_{2,j_o}(t)$. In some particular cases, the wrong estimate $\hat{x}^{\text{av}}_{2,k}(t)$ might happen to be very near to $\hat{x}^{\text{av}}_{2,j_o}(t)$. As discussed above, we also know that all estimates $\hat{x}^{\text{av}}_{2,j}(t)$ with $j \neq j_o$ are near to each other, and hence in this case they are all near to $\hat{x}^{\text{av}}_{2,j_o}(t)$. This might lead to a wrong detection of the outlier position j_o but nevertheless ensures that the trimmed average $x^{\text{av}}_{2,j_o}(t)$ is reconstructed with only a very small error. Now we illustrate this with the help of Example 2.2.

Example 2.2 (continued)**.** *Consider the nominal and the faulty systems described in Example 2.2. The position of the outlier is $j_o = 1$ but this information is assumed to be unknown and moreover, the system is also partially known: only the nominal matrix $\tilde{A}$ is known, not the fault. Our goal is to detect the outlier position and obtain the average estimate excluding the outlier. Note that the condition* (2.13) *is satisfied as* $\operatorname{rank}(\tilde{A}_{12}) = 4 = n_2$*. Therefore, a bank of n_2 observers of the form* (2.6) *can be designed. Here, for all $j \in \{1, \ldots, n_2\}$, we take $\alpha = 3$, $w_j(0) = -\ell_j^T y(0)$ and the parameters ℓ_j^T, g_j^T, h_j^T are computed as in* (2.21)*. The estimates obtained by this bank of observers are depicted in Figure 2.9. We compute the dissimilarity matrix with $\beta = 10$ in* (2.15) *and then follow the proposed method to detect the outlier using* (2.16)*. Figure* 2.10 *shows that indeed the proposed method detects the outlier position $j_o = 1$. In Figure 2.10, it can be seen that at first the method identifies unmeasured node 3 as the outlier but it converges quickly to the true position, i.e., $j_o = 1$ (node $n_1 + j_o = 7$ in the network).*

Remark 2.3. *As a possible variation of our method, the bank of scalar observers for reconstruction of $x^{\text{av}}_{2,j}(t)$ for all j might be replaced by a full-order state observer $\hat{x}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t))$, followed by computation of averages $\hat{x}^{\text{av,full}}_{2,j}(t) := [\mathbf{0}_{1,n_1}, q_j^T]\hat{x}(t)$ for all j. Then, estimates $\hat{x}^{\text{av,full}}_{2,j}(t)$ can be used instead of $\hat{x}^{\text{av}}_{2,j}(t)$ to compute the dissimilarity matrix* (2.15) *and to find the outlier position* (2.16) *and the trimmed average without the outlier.*

Notice that the rank condition (2.13) *that ensures the existence of the bank of tunable scalar observers implies observability of the system* (2.1)*. Hence, in the case where matrices A and B are known (as in Section 2.2), the full-order observer can be designed to have stable error dynamics with any desired rate of convergence.*

However, when the observer is designed using nominal matrices, different from the faulty ones of the system (as in the current section), the estimates obtained from the full-order observer do not share the useful convergence properties of the bank of scalar observers. Theorem 2.3 ensures that $\xi_{j_o}(t) = x^{\text{av}}_{2,j_0}(t) - \hat{x}^{\text{av}}_{2,j_0}(t)$ tends to zero, and the differences $\xi_j(t) - \xi_k(t) = (x^{\text{av}}_{2,j}(t) - x^{\text{av}}_{2,k}(t)) - (\hat{x}^{\text{av}}_{2,j}(t) - \hat{x}^{\text{av}}_{2,k}(t))$ also tend to zero.

The full-order observer designed with nominal matrices is $\dot{\hat{x}} = \tilde{A}\hat{x}(t) + \tilde{B}u(t) + L(y(t) - C\hat{x}(t))$, and has error dynamics $\dot{e}(t) = (\tilde{A} - LC)e(t) + \Delta x(t) + \Psi u(t)$. Because of the

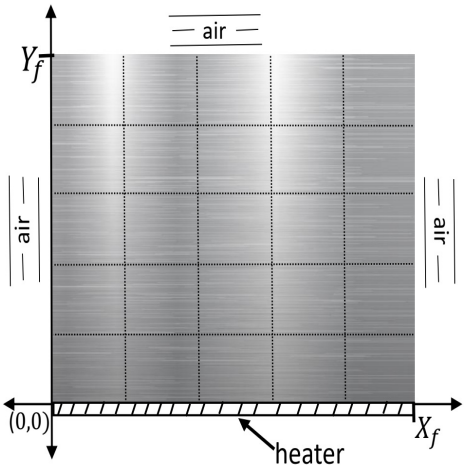


Figure 2.11: A metal plate with a heater attached on one side and surrounded by air on the other sides.

terms $\Delta x(t) + \Psi u(t)$*, the error* $e(t)$ *does not tend to zero, and also when considering* $[\mathbf{0}_{1,n_1}, q_{j_0}^T]e(t) = x_{2,j_0}^{\text{av}}(t) - \hat{x}_{2,j_0}^{\text{av,full}}(t)$ *and* $[\mathbf{0}_{1,n_1}, q_j^T]e(t) - [\mathbf{0}_{1,n_1}, q_k^T]e(t) = (x_{2,j}^{\text{av}}(t) - x_{2,k}^{\text{av}}(t)) - (\hat{x}_{2,j}^{\text{av,full}}(t) - \hat{x}_{2,k}^{\text{av,full}}(t))$ *there is no guarantee that they tend to zero. Therefore, the outlier position and the trimmed average without the outlier might not be obtained correctly.*

2.5 Outlier detection in a faulty metal plate

In this section, we illustrate the outlier detection method using the nominal system on a faulty thermal diffusion system. This system is inspired by [SII17, Section IV]. Here, we deal with a network given by spatial discretization of a thermal diffusion system consisting of a rectangular metal plate attached to a heater on one side and surrounded by air on the other sides as shown in Figure 2.11. Here, we detect the faulty region with only the knowledge of the diffusion properties of the metal plate and the diffusion equations governing the heat diffusion.

2.5.1 Thermal diffusion system

Consider a rectangular metal plate as shown in Figure 2.11. Let $(X, Y) \in \mathcal{D} := [0, X_f] \times [0, Y_f]$ correspond to a point on the metal plate. Let $T(X, Y, t)$ be the temperature of the metal plate at the position (X, Y) at time t. The heat transfer in the metal plate is described by the two-dimensional heat conduction equation

$$\frac{\partial T}{\partial t} = \lambda \frac{\partial^2 T}{\partial X^2} + \gamma \frac{\partial^2 T}{\partial Y^2}, \quad (X, Y) \in int(\mathcal{D}). \tag{2.30}$$

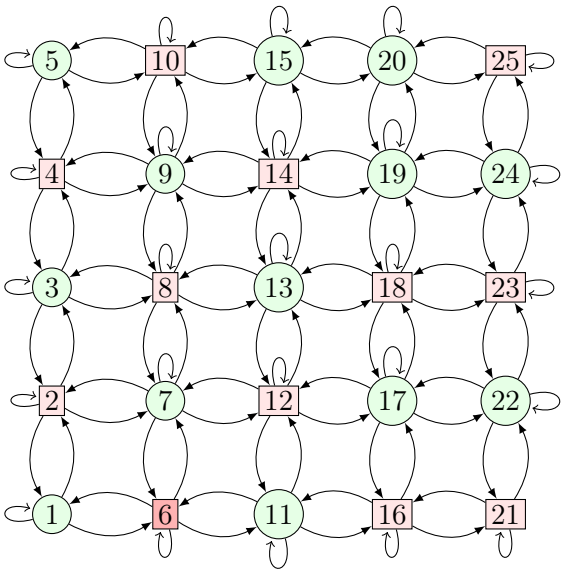


Figure 2.12: Graphical representation of the discretized metal plate. In this graph, the nodes correspond to the cells obtained after discretization and edges show the heat transfer between them. The circle nodes in green are the measured nodes and the red square nodes are the unmeasured nodes. The nodes $\{1, 6, 11, 16, 21\}$ receive input directly and the node 6 in dark red is the outlier.

where $int(\mathcal{D})$ denote the interior region of the plate. The constants λ and γ denote the diffusion coefficient along the X-axis and Y-axis respectively. Moreover, there is an exchange of heat between the plate, the heater and the air which is described by the following boundary conditions. The heat exchange with the air is described with the Robin boundary condition [ZC09]

$$\frac{\partial T}{\partial \nu} = -\eta_a(T - T_a), \quad (X, Y) \in \mathcal{S}_a. \tag{2.31}$$

where S_a is the set of contact points with the air, ν is the outward unit vector normal to $\mathcal{S}_a$, and T_a is the temperature of the air, and η_a is the ratio of coefficient of thermal conductivity and the coefficient of heat transfer between air and the metal plate. For simplicity, we suppose $T_a = 0$ for any t, X and Y. The heat exchange with the heater is described by another Robin boundary condition

$$\frac{\partial T}{\partial Y} = \eta_h(T - u), \quad X \in [0, X_f], Y = 0. \tag{2.32}$$

where u is the temperature of the heater and η_h is the ratio of coefficient of thermal conductivity and the coefficient of heat transfer between heater and the metal plate. The heater is assumed to have a uniform temperature distribution, that is, u is independent of X and Y.

We discretize (2.30) and (2.32) with step size $\delta = \Delta X = \Delta Y$ using central difference quotients. In discretization, let the number of the cells along each axis be n_c so that the total number of cells will be n_c^2. We obtain a network with grid structure with dynamics (2.17), where $\mathbf{x} \in \mathbb{R}^{n_c^2}$ is a vector of spatially discretized temperature T. The numbering of the nodes

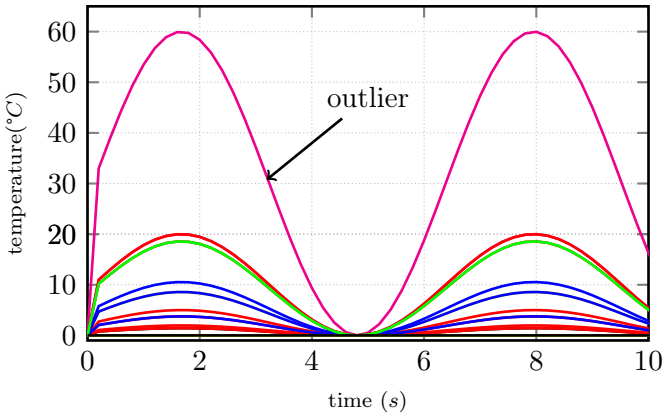


Figure 2.13: State trajectories of the spatially disctretized equations describing the faulty metal plate.

starts from the bottom left, and follows columns, each from bottom to top as in Figure 2.12. To describe the entries $\tilde{a}_{ij}$ of $\tilde{A}$, we need to distinguish different cases, depending on the position of the cell corresponding to vertex i. The non-zero entries of matrix $\tilde{A} = [\tilde{a}_{ij}]$ are given as follows. If i is not a boundary node then

$$\tilde{a}_{ij} = \begin{cases} -2(\lambda+\gamma) & \text{if } j = i \\ \gamma & \text{if } j = i+1, i-1 \\ \lambda & \text{if } j = i+n_c, i-n_c. \end{cases}$$

If i is a boundary node attached to the heater but not a corner node then, from the discretization of (2.30) and (2.32), we have

$$\tilde{a}_{ij} = \begin{cases} -2(\lambda+\gamma+\delta\eta_h\lambda) & \text{if } j = i \\ 2\gamma & \text{if } j = i+1 \\ \lambda & \text{if } j = i+n_c, i-n_c. \end{cases}$$

For the boundaries with air, the discretization of (2.30) and (2.31) is used. If i is on the upper, left or right boundary, except for the corner nodes, we have $\tilde{a}_{ij} = -2(\lambda+\gamma+\delta\eta_a\lambda)$ if $j = i$. If i is on the upper boundary, the non-zero non-diagonal entries are $\tilde{a}_{ij} = 2\gamma$ for $j = i-1$ and $\tilde{a}_{ij} = \lambda$ for $j = i+n_c, i-n_c$. Similarly, if i is on the left border, we have $\tilde{a}_{ij} = \gamma$ for $j = i-1, i+1$ and $\tilde{a}_{ij} = 2\lambda$ for $j = i+n_c$, and then on the right, we have $\tilde{a}_{ij} = \gamma$ for $j = i-1, i+1$ and $\tilde{a}_{ij} = 2\lambda$ for $j = i-n_c$. Finally, if i is an upper-left or upper-right corner node, we have $\tilde{a}_{ij} = -2(\lambda+\gamma+\delta\eta_a\lambda+\delta\eta_a\gamma)$ for $j = i$. If i is on the upper-left corner, we have $\tilde{a}_{ij} = 2\gamma$ for $j = i-1$ and $\tilde{a}_{ij} = 2\lambda$ for $j = i+n_c$. Similarly, if i is on the upper-right corner, we have $\tilde{a}_{ij} = 2\gamma$ for $j = i-1$ and $\tilde{a}_{ij} = 2\lambda$ for $j = i-n_c$.

For a corner node in the bottom, the discretization of (2.30), (2.31) and (2.32) is used. If i is either a bottom-left or bottom-right corner node, we have $\tilde{a}_{ij} = -2(\lambda+\gamma+\delta\eta_a\lambda+\delta\eta_h\gamma)$ for $j = i$. For non-zero non-diagonal entries, we have $\tilde{a}_{ij} = 2\gamma$ for $j = i+1$ and $\tilde{a}_{ij} = 2\lambda$ for

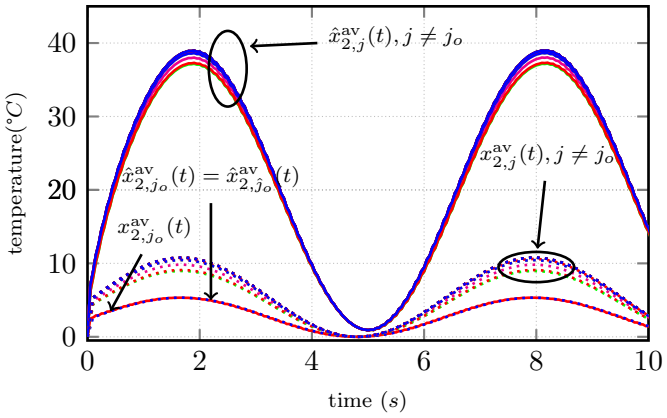


Figure 2.14: In solid lines, estimated average trajectories of the unmeasured nodes excluding one at a time, $\hat{x}^{\text{av}}_{2,j}$. The red trajectory corresponds to the average estimate without the outlier, i.e., $\hat{x}^{\text{av}}_{2,j_o}(t)$. Moreover, in this case $\hat{x}^{\text{av}}_{2,j_o}(t) = \hat{x}^{\text{av}}_{2,\hat{j}_o}(t)$. The dotted trajectories correspond to the true averages $x^{\text{av}}_{2,j}(t)$.

$j = i + n_c$ if i is on the bottom-left corner. Similarly, if i is on the bottom-right corner, we have $\tilde{a}_{ij} = 2\gamma$ for $j = i + 1$ and $\tilde{a}_{ij} = 2\lambda$ for $j = i - n_c$.

The entries of the matrix $\tilde{B} = [\tilde{b}_{ij}] \in \mathbb{R}^{n_c^2 \times 1}$ are

$$\tilde{b}_{i1} = \begin{cases} 2\eta_h \delta\gamma & \text{if } i = 1 \bmod n_c \\ 0 & \text{otherwise.} \end{cases}$$

and the matrix C depends on the choice of the measured nodes.

2.5.2 Illustration of the detection method

In this subsection, we show the effectiveness of the proposed detection method for spatially discretized thermal diffusion network with $\delta = 1$, and $n_c = 5, \lambda = \gamma = 29.1, \eta_h = 1.3{\times}10^4, \eta_a = 10^3$ and input $u(t) = 10 + 10 * \sin(t)$. Initially the plate is kept at $0^\circ C$. On discretization, we get a network with grid structure as shown in Figure 2.12. The nodes $\{1, 6, 11, 16, 21\}$ are attached to the heater. We assume that there is a defect in a region corresponding to the discretization cell number 6. The metal plate corresponding to this region is defective and has different coefficients of diffusion than the rest part of the plate. This defect can be the result of the formation of brown stains, the formation of oxides on the metal plate, casting defects, welding defects, or rolling defects to name a few. In our example, the fault scales the local coefficients of diffusion λ and γ by one-third. This fault is represented as faulty matrices given by $A = \tilde{A} + \Delta$ and $B = \tilde{B} + \Psi$ with Δ satisfying (2.20) with $r^T = -\frac{2}{3}(e_6^T \tilde{A} - (e_6^T \tilde{A} e_6) e_6^T)$, $\rho = -\frac{2}{3}$ and $\Psi = \mathbf{0}_{n,1}$. It can be seen in Figure 2.13 that node 6 is an outlier.

We choose $\{1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 20, 22, 24\}$ as the set of measured nodes, and hence the set of unmeasured nodes is $\{2, 4, 6, 8, 10, 12, 14, 16, 18, 21, 23, 25\}$. Now, we can apply a

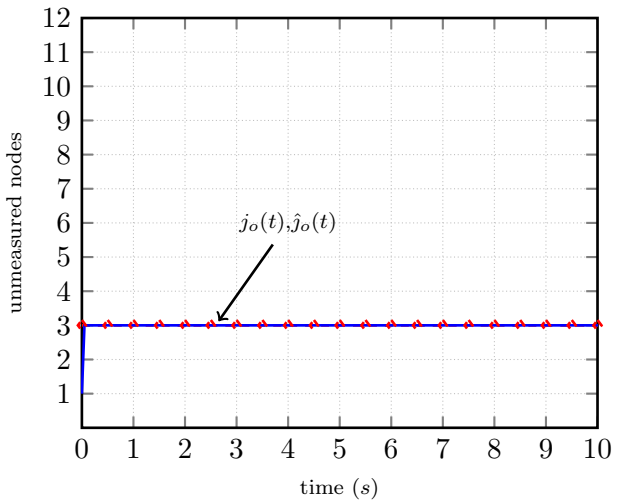


Figure 2.15: Detected position of the outlier $\hat{j}_o$. It can be seen that the detected position converges to the true position of the outlier j_o very quickly. It is found that the outlier is at the third position in the set of unmeasured nodes i.e. $j_o = 3$, which is at node 6 in the network.

suitable permutation to rearrange the nodes such that the first $n_1 = 13$ nodes are measured and the rest n_2 are unmeasured so as to obtain a block structure as in (2.19). Note that, node 6 in the network is the third node in the set of unmeasured nodes, so we have $j_o = 3$. Now, we assume that we neither know the fault nor the position of the outlier. Therefore, for estimation, we use the system matrices obtained by the discretization of the thermal diffusion equation described in the previous subsection. Recall from Theorem 2.2 that in order to design a bank of scalar observers, the sub-matrix $\tilde{A}_{12}$ needs to be of full column rank. We find that indeed $\text{rank}(\tilde{A}_{12}) = 12 = n_2$. Therefore, a bank of scalar observers of the form (2.6) can be designed. Here, we take $\alpha = 3$ and $w_j(0) = -\ell_j^T y(0)$ for all $j \in \{1, \dots 12\}$. Figure 2.14 shows the different average estimates excluding one node at a time $\hat{x}^{\text{av}}_{2,j}(t)$ and the corresponding true averages $x^{\text{av}}_{2,j}(t)$, depicted by solid and dotted lines respectively. The true averages $x^{\text{av}}_{2,j}(t)$ clearly show that j_o is an outlier: for all j except j_o, the averages $x^{\text{av}}_{2,j}(t)$ are near to each other, while $x^{\text{av}}_{2,j_o}(t)$ is more distant. As predicted by Theorem 2.3, the estimate $\hat{x}^{\text{av}}_{2,j_o}(t)$ quickly converges to the true trimmed-average $x^{\text{av}}_{2,j_o}(t)$. For $j \neq j_o$, it can be seen that all the estimates $\hat{x}^{\text{av}}_{2,j}(t)$ for $j \neq j_o$, have a significant error ξ_j from the corresponding true averages, but such errors tend to be the same for all j, consistently with Lemma 2.1. Hence, the fact that the true averages are near to each other results in the estimates also being near to each other, despite their error.

Using $\beta = 10$ in (2.15) and then computing (2.16), we detect the position of the outlier as shown in Figure 2.15.

It can be seen in Figure 2.15 that the detected position of the outlier is $\hat{j} = j_o$. The process identifies the outlier from the beginning as it is evident in Figure 2.14 that the difference between the average estimates $\hat{x}^{\text{av}}_{2,j_o}(t)$ and $\hat{x}^{\text{av}}_{2,j}(t)$ is very large since the beginning.

2.6 Concluding remarks

Average state reconstruction with the help of a few sensor measurements can give unexpected results if there are some outliers among the unmeasured states. A method to estimate the average excluding the outlier has been proposed. For that, a design of a scalar and tunable observer has been given along with the condition under which a bank of these observers can be designed to estimate the average of the unmeasured nodes while excluding an element at every possible position. Moreover, the problem of detection of the existing outlier when the system matrices might be known or partially unknown has also been addressed by using a dissimilarity based matrix inspired from the euclidean distance-based dissimilarity matrix.

Part II

Modeling and control of mobility for epidemic mitigation in large scale urban networks

Chapter 3

Introduction

Human mobility, because of its practical applications in epidemic control, urban planning, traffic planning and land use patterns, has attracted the attention of researchers from transportation [Hua+18], economic geography [RT04], epidemiology [SD95], physics [Sim+12] and many other fields over past decades. Study of urban human mobility in cities, is important and facilitates better planning and policy-making. For example, in the recent events of COVID-19 pandemic, controlled and restricted mobility of people became important and a fundamental issue faced by the policymakers. In such scenarios, both fully operational mobility or a completely restricted mobility could result in negative consequences on the socio-economic status of a country or a region. Therefore, a better understanding of human mobility patterns is necessary on the level of a metropolitan area with the three mobility components, 1. 'Where' do the people go? 2.'When' do they move? and 3.'How many' people move? (WWH). There are several human mobility prediction models which can be categorised mainly in two categories based on information they use: 1. mathematical models using traditional data such as population census or travel survey data 2. models based on sensed data such as mobile phone Call Detail Record (CDR) and GPS data.

At first, we give an overview of the first category which is the mathematical models using traditional datasets. The prediction of human mobility has been studied extensively by statistical physicists over the years. One of the first models proposed to predict population movement was gravity model [ES90]. This model proposes that the flow of people between two locations is proportional to the product of their populations and inversely proportional to the distance between them. Since then, many of its generalised versions have been proposed which have several exponential parameters and functions to be inferred from the empirical data [Gau+09]. This requires previous traffic data to fit the parameters in order to find the exact mathematical formula making it difficult to predict mobility in regions which lack systematic traffic data. Many improvements include the radiation model proposed in [Sim+12] and other models which can be found in the survey [Bar+18]. These models give an estimate of number of people moving between different places in terms of origin-destination matrices which can be useful in predicting migratory flows and large scale mobility between cities. However, they lack spatio-temporal characteristics which are relevant for urban mobility. In other words, they estimate 'how many' and to some extent 'where' but give no estimate of 'when'. Another class of literature such as [Ekm+08], [MM06] model mobility of people between two locations and time spent at such locations. the model is given in form of some power laws. Some mover-stayer type dynamic mobility models have also been proposed in the literature such

as [SD95] and [CMV13] in order to see its effect on a disease spread. These models capture the evolution of mobility over time between different geographical regions. However, they capture averaged mobility between different cities with large time-scales and cannot capture daily mobility patterns and there is no distinction between origins and destinations.

Now, we give a brief overview of the second category which are data-driven models. Recently, with the vast development in the information and communication technology, various data sources have been used to facilitate travel behaviour research. Examples are smartcard records data, GPS data and roadside sensor data, and among these, mobile phone CDR are the most widely applied [WHL18]. The mobile phone CDR can be used to construct the origin-destination matrices and mobility patterns. For example, [Iqb+14] developed OD matrices using mobile phone CDR, [Bac+19] estimated OD matrices by transport mode, [Ale+15] estimated the mobility pattern along the day with OD matrices, to name a few. The main reason of extensive use of this data is ubiquitousness of mobile phones, hence the ability to track the whole population at large span of time and relatively high spatial accuracy. These techniques might cover 'how much' and 'when' to some extent provided the availability of data. However, some researchers have pointed to the biases of these data, caused by the event-triggered nature of mobile phone data [Zha+16]. These data can also be unavailable in case of bad weather or connectivity and moreover amount of mobile phone usage also varies according to the socio-economic conditions of different regions. Another widely used data source is GPS data due to its high geographical precision. It is used to study individuals' movement in order to analyse mobility patterns and give mobility models [LH14]. GPS data have been used to identify stay and destinations [HT04], estimate mobility patterns [ZG10], estimating mobility patterns using taxi GPS data [Tan+15], identifying transport mode [DH18] and updating OD matrices [GF16]. However, these location data come at a risk of privacy of individuals and taxi location data has a very low penetration rate. There is a serious concern over privacy of individuals as removal of personal data does not fully preserve privacy and have risk of re-identification [MYA+13]; [Yin+15] of the individuals. In future, new laws can also be implemented which will restrict the mobile and location data use even more [com17]. Moreover, privacy protection techniques require data aggregation or need to be recorded for short period limiting their usefulness [ZB11].

In this part, we extend, adapt and implement the supply-demand flow mobility model proposed in [Nia+21] to the large-scale network of Grenoble, a metropolitan city in France. In this model, every day, a certain number of people travel from their residences, which are called origins, to locations visited daily such as for work, education, shopping, hospitals etc., which are called destinations, and then return to their home again. The daily mobility patterns are captured by the time-dependent supply and demand gating profiles. The Supply Gating Functions (SGF) of each destination is controlled by its daily destination schedule, which is its opening and closing hours. The Demand Gating Profile (DGP), on the other hand, is defined on each edge of the mobility network and corresponds to the daily mobility window, which is the time interval during which people move between origins and destinations along that edge. The supply function of each destination controlled by the SGF determines the inflow allowed to that destination and depends on its operating capacity which can be tuned using a control input, for instance for epidemic attenuation. The demand function controlled by

the DGP determines the outflow from one location to another. The process of urban human mobility is modeled on the network edges that connect different locations through flows. A significant advantage of this model is that it can capture mobility on a smaller time-scales with a different pattern on a given day of the week thanks to SGF and DGP. Another important advantage with respect to existing models, is that the proposed model can encompass all the three mobility components (WWH). However, [Nia+21] uses a toy example and hence lack a precise procedure for the identification of the functions, gating profiles and parameters needed to implement this model on a real large-scale network. One of the main aim of the work done in this part is to devise methods and procedures for identifying the origin and destination nodes, their O/D relations, setting the parameters and defining the gating functions DGP and SGF for the large-scale mobility network of Grenoble.

In this part, we also build the large-scale network of Grenoble using public information which makes our method free from privacy and legal issues. precisely, we build the three components of mobility: Where, When and How many (WWH). In 'where', We consider the residential areas of the communes as origins and identify different places of interests such as schools, universities, workplaces, hospitals parks etc. as the destinations. The destinations are then classified into categories and subcategories. Each origin and destination is represented by a node in the network. While the origin nodes are placed in the residential areas, the destination nodes are placed at the exact locations using open source map data. Then, we collect the population of the origins from census data and the capacities of the destinations from different sources such as their websites. The capacity of a destination is the maximum number of persons allowed to be there at any instant. If any subcategory has a large number of nodes, then they are aggregated by replacing such nodes in a commune by a single node with its capacity being the sum of their capacities and its location being the barycenter of their locations. Having identified the origins and destinations, we make a bipartite graph corresponding to the destinations of each subcategory to decide which origins and destinations are connected. They are connected based on a form of attraction law proportional to the population and the capacities of origins and destinations and there is a threshold over distance, a person would travel to go to a destination. For this, we compute the real minimum road distance between all possible locations. In the component, 'When', we collect the information about the opening and closing time per destination category and the gating functions are computed using the mobility profiles from travel survey or inspired from the real time popularity trends. Finally, in the component 'How many', we compute the average daily number of people traveling from an origin to a destination if they are connected by using rules of proportions.

Since, the mobility-model proposed in [Nia+21] is in continuous-time, therefore, in order to extend and implement it on the real large -scale network, we redefine the model in discrete-time ensuring mass-preservation and non-negativity. These properties are also satisfied even if we take larger time-steps in simulations. In addition, since the model and the developed setup is modular in nature, it can be used for many applications such as understanding social behavior and urban planning and optimal control of mobility for epidemic mitigation. Indeed human mobility is essential for the economy of a country or a region, however, it can also be responsible for an epidemic spread when infected people mix with the susceptible population in common places. For this reason, the government devises restricted mobility strategies in

events of an epidemic. However, recently during Covid-19 pandemic, it has been seen that unrestricted mobility can lead to health infrastructure overloading and hence to an substantial increase in loss of lives. As an intent to mitigate epidemic spreading and their consequences on the population health, governmental policies tends to reduce human mobility in a conservative manner strongly affecting the social well-being and economy. Therefore, restricted mobility policies should act parsimoniously by maximizing the social well-being and economic activities (minimizing mobility restriction) while respecting the constraints imposed by the available health infrastructure. Some works such as [BM20],[Mol+21] and [DEK21] show some analyses on impact of Covid on mobility during the first wave and first set of hard measures.

In this work, we also extend our focus on integrating the mobility model with an epidemic model and utilize it to design optimal mobility restriction policies to limit the epidemic spread. Recently, the outbreak of Covid-19 motivated several work on optimization for epidemic mitigation but with no network such as [Las+21], [Kö+21], [Mor+21],[MR22]. We are interested in human mobility restrictions for epidemic mitigation in urban network. Most of those works consider mathematical models for the epidemic spreading part, but differs in the way how mobility patterns are captured.

One category of these works uses *data-driven mobility patters* [NBC+21]; [Smo+21]; [Gos+21]; [Dut+21]. For example, the work in [Smo+21] analyses the effect of mobility restrictions put by government bodies during Covid-19 across Italian regions, and [Gos+21] quantifies a trade-off between allowed cross-region mobility and number of infections and gives optimal restriction policies. [Dut+21] proposes an age-structured SEIRD epidemic model in which parameters are tuned using daily mobility patterns and contact traces. The model is given for intra-region mobility, and is used to design optimal mobility policies constrained to health and socio-economic factors. However, one limitation is that it only considers aggregated destinations without specific geographic locations. In this category, contributions rely on the use mobility patterns from telecom or google real-time data which may be difficult to obtain, and could violate privacy of individuals. To overcome privacy issues, [Son+20] proposes reinforced learning methods to find mobility restriction policies to and from high risk regions by using aggregated inter-regional mobility data instead of individual traces.

Another category of works uses *model-driven mobility patters* to find optimal policies using mathematical models based on conservation-like laws integrating both: mobility and epidemic process. Model integrating cross-region mobility and epidemic spread by SIR dynamics has been investigated by [SD95] and more recently by [CMV13]. Multi-cities epidemic models have been extensively studied such as in [AD03], [KR07],[Ari09], [Yin+20], [LS13],[Kel+16], [LABH21], [LXH21] , [KBG04], [Gao20], [CYX14]. Some recent works such as [Hu+21], [Cen+21], [Car+20] exploit cross-region models to find optimal epidemic mitigation policies. However, some limitations of these approaches are that they only capture cross-region mobility patterns and are unable to capture daily mobility patterns in an urban network, necessary for precise assessment of the health and economic impacts.

In this regard, [Nia+21] at first, proposed a dynamic mobility model which incorporates epidemic spread at each geographical location in an urban city with SIR dynamics called SIR-mobility model. This model is used to formulate optimal control problems for epidemic

mitigation while maximizing the economic activity.

Similar to the mobility model, this SIR-mobility model is also given in continuous time, so we propose a numerical mechanism so that the resulting discrete-time SIR-mobility model preserves the original properties while ensuring numerical stability keeping in mind the goal to devise techniques that can be applied to a large-scale network. Then, based on the numerical version of the combined SIR-mobility model, we propose several optimal mobility policies to maximize the socio-economic activity while respecting the ICU-hospital constraints. Since, in large scale networks, solution of an optimal control problem is not possible by using off the shelf solvers, we devise algorithm and techniques which can be implemented to a large-scale network. We apply these techniques to obtain optimal mobility control policies for the Grenoble metropolis. At first, we provide optimal mobility control policies that is category-free. This category-free control approach tunes the operating capacities of all the destination categories. The optimal searching process exploits the monotonicity of the proposed optimization problem to obtain an numerical efficient solver. The algorithm has been deployed in a model predictive control (MPC) fashion with receding time horizon, where both the frequency of updates and the length of the optimization horizon are design parameters. Results show that in this case, it is most effective to have frequent updates and a short optmization horizon, leading to a U-shape control profile where destinations are almost fully open at the beginning and at the end, while they are almost closed near the infection peak. Then, we proceed to the more general problem where we devise control policies that are category-dependent where different control policies are applied to different categories depending on their socio-economic importance. We devise algorithm to solve this general problem which uses outer approximation based methods with several numerical enhancements alongwith an MPC approach provide policies that are category-dependent. Results show that category-dependent policies are better than category-free policies. We provide effective mobility control strategies to be adopted during different phases of epidemic in Grenoble metropolis using the developed mobility network.

Finally, to illustrate our results, we have developed an online demonstrator called GTL-Healthmob interface where an user can simulate different mobility control scenarios and as a case study he/she can visualise the effect on mobility on COVID-19 epidemic spread in Grenoble area.

Chapter 4

Model for human mobility and epidemic spread

Contents

4.1 Continuous-time mobility model: where, when and how many? . . . 45
4.2 Continuous-time SIR-mobility model 49
4.3 Discrete-time mobility model with preserving properties 50
4.4 Discrete-time SIR-mobility model 55
4.5 Concluding remarks 58

Control of human mobility during an epidemic outbreak becomes essential for the well being of a society, therefore, making it an fundamental challenge faced by the policymakers. Such control can only be done efficiently if human mobility is modeled at the scale of a city or a metropolis. Therefore, in this chapter, we consider a continuous-time human-mobility model and a model that integrates the epidemic spread process with mobility. Keeping in mind the goal to extend, adapt and implement this model to a large-scale network which we build in the next chapter, we provide a discretization of these models that preserves the original properties such mass preservation, boundedness and non-negativity as in the continuous-time model.

4.1 Continuous-time mobility model: where, when and how many?

In this section, we give a basic description of the urban-human mobility model proposed in [Nia+21].

Human mobility can be defined as the flow of people between two locations. Consider the human mobility in an urban area between two types of locations: origins and destinations. The origins correspond to the residential areas and the destinations, on the other hand, correspond to locations that people visit daily for some purpose like work, shopping, education or leisure for example, companies, research centers, schools, restaurants, cinemas, etc. Every day a certain number of people visit the destinations during specified time intervals and then return later on the same day.

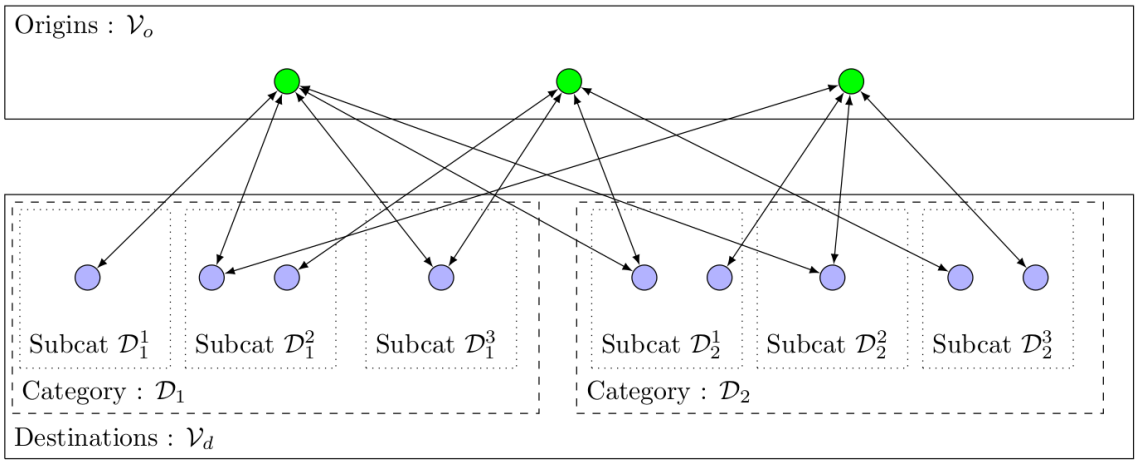


Figure 4.1: An example of urban human mobility network and classification of destinations. Here, Subcat stands for subcategory.

Consider the daily human mobility in the timescale of hours represented by a network $\mathcal{G} = (\mathcal{V}_o, \mathcal{V}_d, \mathcal{E})$ between locations of two types: origins $\mathcal{V}_o$ and destinations $\mathcal{V}_d$. For the sake of simplicity, we assume that the network $\mathcal{G}$ is bipartite, i.e., we assume that there are edges only between origins and destinations, and there is no edge between a pair of origins or a pair of destinations. Since, the mobility is happening instantly between two locations, this assumption doesn't effect the mobility pattern much as the mobility between two destinations can be understood as if a person first goes back to the origin from one destination and then moves from the origin to the other destination. The assumptions on the model are as follows:

- The total population of the city remains constant.
- The mobility occurs only between pairs of origins and destinations, and not among a pair of different origins or a pair of different destinations.
- The number of people who visit destination j from origin i during a day is equal to the number of people who return to i from j on the same day.

The destinations $\mathcal{V}_d$ are partitioned into q categories $\mathcal{D}_1, \ldots, \mathcal{D}_q$, where each category $\mathcal{D}_h$ is further partitioned into subcategories, $\mathcal{D}_h^1, \ldots, \mathcal{D}_h^s$, where s is the number of subcategory in the category $\mathcal{D}_h$. Figure 4.1 shows an example of urban human mobility network which also shows the classification of destination into categories and subcategories. Denote the total population of origin $i \in \mathcal{V}_o$ by P_i which is the total number of people residing in i and the nominal instantaneous capacity of destination $j \in \mathcal{V}_d$ by C_j which is the maximum number of people who can visit j at any given time during normal operation. We introduce the possibility to describe the restrictions of capacity as control inputs, such as those that were imposed by many governments during peaks of Covid-19 pandemic. Such capacity reduction is done per category of destinations, and is described by a coefficient $u_h(t) \in [0, 1]$ for $h \in \{1, \ldots, q\}$ which

determines the allowed operating capacity of the destinations in $\mathcal{D}_h$ in terms of the proportion of nominal capacity at time t. In other words, it can be considered as a policy at time t that limits the operating capacities denoted as $C_j^o(t)$ in the destinations of category h, where

$$C_j^o(t) = C_j u_h(t), \quad \text{for } j \in \mathcal{D}_h. \tag{4.1}$$

In (4.1), if for a destination $j \in \mathcal{D}_h$, $u_h(t) = 0$, then the operating capacity $C_j^o(t) = 0$, which means that no person is allowed to visit j. On the other hand, if $u_h(t) = 1$, then $C_j^o(t) = C_j$ that is the operating capacity of j is equal to its nominal capacity. Therefore, the operating capacity C_j^o of a destination is less than or equal to its nominal capacity C_j and depends on the control input $u_h(t)$ for $j \in \mathcal{D}_h$ which tunes the maximum number of people that can gather at the destinations at any time and hence controls the flow of people between origins and destinations.

Let $N_i(t) > 0$ be the number of people in $i \in \mathcal{V}_o \cup \mathcal{V}_d$ at time t and let $\mathcal{N}_i = \{j : (i,j) \in \mathcal{E}\}$ be the set of i's neighbors. Then, according to the urban human mobility model proposed in [Nia+21], the rate of change of the number of people at any location at time t is equal to the sum of inflows to that location minus the sum of outflows from that location. In other words, if $\phi_{ij}(t, N_i(t), N_j(t))$ denotes the flow from i to j, then the number of people $N_i(t)$ at a location i at time t is given by

$$\dot{N}_i(t) = \sum_{j \in \mathcal{N}_i} (\phi_{ji}(t, N_j(t), N_i(t)) - \phi_{ij}(t, N_i(t), N_j(t))), \tag{4.2}$$

where

$$\phi_{ij}(t, N_i(t), N_j(t)) = \min(\Delta_{ij}(t, N_i(t)), \Psi_j(t, N_j(t))). \tag{4.3}$$

Here, $\Delta_{ij}(t, N_i(t))$ is the demand of i with respect to j which describes the flow of people who would like to travel from i to j and $\Psi_j(t, N_j(t))$ is the supply of j, *i.e.* the inflow that can be allowed to enter j from its neighbors depending on the operating capacity C_j^o, if j is a destination and P_j if j is an origin. For the sake of simplicity, here onward, we will use the notation $Z_j(t)$ for the operating capacity $C_j^o(t)$, if j is a destination or the population P_j if j is an origin unless otherwise stated. Note that, if j is an origin, $Z_j(t)$ will be constant and equal to the population P_j, however, if j is a destination then $Z_j(t)$ will be equal to the operating capacity $C_j^o(t)$ which varies according to the coefficient $u_h(t)$ if $j \in \mathcal{D}_h$ as defined in (4.1). Furthermore, for the sake of simplicity in notations, we will just use $\phi_{ij}(t)$ instead of $\phi_{ij}(t, N_i(t), N_j(t))$ and similarly for $\phi_{ji}(t)$.

Demand The demand of i with respect to j is given by

$$\Delta_{ij}(t, N_i(t)) = \delta_{ij}(t) \cdot f_{ij}(t) \cdot 1_{N_i(t)>0}, \tag{4.4}$$

where $\delta_{ij}(t)$ is the Demand Gating Profile (DGP) such that

$$\begin{cases} \delta_{ij}(t) > 0 & \text{if } t \bmod 24 \in [t_{ij}, t_{ij} + \tau_{ij}) \\ \delta_{ij}(t) = 0 & \text{otherwise} \end{cases} \tag{4.5}$$

satisfying $\int_0^{24} \delta_{ij}(t)dt = 1$. Here, $[t_{ij}, t_{ij} + \tau_{ij}) \subseteq [0, 24)$ is called the *mobility window*, defined as the time interval in which there is mobility from i to j. Here, t_{ij} is the time when mobility from i to j starts and τ_{ij} is time duration for which the mobility from i to j continues to happen. We can have different Demand Gating Profiles and mobility window for different days of the week for different destination subcategories. The function δ_{ij} gives the profile of demand over the day, while the intensity of the demand is described by a factor $f_{ij}(t)$ defined as:

$$f_{ij}(t) = M_{ij}u_h(t). \tag{4.6}$$

Here, M_{ij} is the total number of people that would like to visit j from i, daily and $M_{ij}u_h(t)$ is the number of visitors when capacity is restricted with the coefficient $u_h(t)$. The term $M_{ij}u_h(t)$ reflects the fact that when a government policy regarding capacity reduction is announced, it not only restricts the number of people who can visit a destination but also reduces the number of people willing to go to those destinations. Finally,

$$1_{N_i(t)>0} = \begin{cases} 1 & \text{if } N_i(t) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

is an indicator function.

Supply The supply of j is given by

$$\Psi_j(t, N_j(t)) = \psi_j(t) \cdot F_j \cdot 1_{N_j(t)<Z_j(t),} \tag{4.8}$$

where

$$\psi_j(t) = \begin{cases} 1 & \text{if } t \bmod 24 \in [a_j, b_j) \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

is the Supply Gating Functions (SGF) with $[a_j, b_j) \subseteq [0, 24)$ called the *destination schedule* where a_j is the opening hour and b_j is the closing hour of location j. Similar to DGPs, also SGFs are defined over a day and repeated periodically, possibly with different profiles for different days of the week. For the origins $j \in \mathcal{V}_o$, $[a_j, b_j) = [0, 24)$ because they are always open. Define F_j as the maximum constant flow inflow to j from its neighbors and computed as $F_j = \sum_{i \in \mathcal{N}_j} f_{ij}$ is the maximum, constant inflow to j from its neighbors. Finally,

$$1_{N_j(t)<Z_j(t)} = \begin{cases} 1 & \text{if } N_j(t) < Z_j(t) \\ 0 & \text{otherwise} \end{cases} . \tag{4.10}$$

Notice that N, ϕ, Δ and Ψ, all are implicitly dependent on the capacity control $u(t)$ as it tunes the operating capacity and hence controls the flows. It doesn't appear in notations so as to avoid heavier notations. It will be mentioned when applying control on the mobility in coming chapters.

Similar to the restrictions on capacity, we can also have the possibility to describe the restrictions on closing time of destinations, such as those that were imposed by many governments during peaks of Covid pandemics. Such restriction is decided per category of destinations, and is described as the *effective closing time* denoted by $b_h^e \in [6, 24]$ for $h \in \{1, \ldots, q\}$. Similar restriction on the opening time can also be enforced by another such coefficient to tune the effective opening time. However, if done, it must be enforced that the effective closing time is later or same as the effective opening time.

4.2 Continuous-time SIR-mobility model

The mobility model (4.2) can also be used to see the effect of mobility control on the epidemics spread. In [Nia+21], the mobility model has been incorporated with the SIR epidemic model. The resulting model is called the SIR-mobility model. Here, S, I and R stand for the number of susceptible, infected and recovered people. At each location, the disease spread follows the classical SIR model, but people are also allowed to move from one location to another (irrespective of their status of infection) and such mobility happens according to the mobility model (4.2)-(4.9). This has been depicted in Figure 4.2.

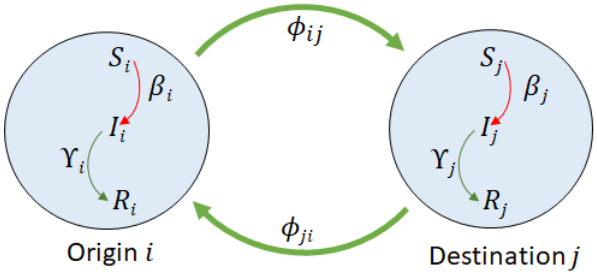


Figure 4.2: Mobility along the edges and the epidemic spread inside the locations.

Let us denote $S_i(t), I_i(t), R_i(t)$ to be the susceptible, infected and recovered population at the location i at time t. Furthermore, as defined in (4.2), let $N_i(t)$ be the number of people at the location i at time t.

The disease transmission at each location i happens according mass action law

$$\beta_i(t) S_i(t) \frac{I_i(t)}{N_i(t)}$$

where

$$\beta_i(t) = \begin{cases} \bar{\beta}_i \frac{N_i(t)}{P_i} & \text{if } i \in \mathcal{V}_o \\ \bar{\beta}_i \frac{N_i(t)}{C_i} & \text{if } i \in \mathcal{V}_d \end{cases}$$

is the infection rate of i at time t with $\bar{\beta}_i$ being the nominal infection rate of i. The nominal infection rate is defined as the product of infection probability and the average number of contacts of a person in location i per hour when the number of people in i is maximum. The infection probability is defined as the probability by which an infected person might infect a susceptible person when in contact. Also notice that, the infection rate β_i is low when the number of people $N_i(t)$ at location i is small and high when $N_i(t)$ is large. The infected people $I_i(t)$ recover with a recovery rate $\gamma \in [0, 1]$, which is a constant that depends on the epidemic and the existing treatments, if any. The recovery rate is defined as the inverse of the average recovery period (in hours) of the infected cases.

Consider the flow $\phi_{ij}(t)$, the number of people moving from the location i to the location j at time t and $\mathcal{N}_i$ to be the neighbours of i, then the resulting SIR-mobility model proposed

in [Nia+21] in continuous-time can be written as

$$\begin{aligned}
\dot{S}_i(t) =& -\beta_i(t)\frac{S_i(t)I_i(t)}{N_i(t)} + \sum_{j\in\mathcal{N}_i}\left(\phi_{ji}(t)\frac{S_j(t)}{N_j(t)} - \phi_{ij}(t)\frac{S_i(t)}{N_i(t)}\right)\\
\dot{I}_i(t) =& \beta_i(t)\frac{S_i(t)I_i(t)}{N_i(t)} - \gamma I_i(t) + \sum_{j\in\mathcal{N}_i(t)}\left(\phi_{ji}(t)\frac{I_j(t)}{N_j(t)} - \phi_{ij(t)}\frac{I_i(t)}{N_i(t)}\right)\\
\dot{R}_i(t) =& \gamma I_i(t) + \sum_{j\in\mathcal{N}_i}\left(\phi_{ji}(t)\frac{R_j(t)}{N_j(t)} - \phi_{ij}(t)\frac{R_i(t)}{N_i(t)}\right)
\end{aligned} \tag{4.11}$$

4.3 Discrete-time mobility model with preserving properties

To implement these models to a large-scale network, one needs a discretization of these models which preserve their original properties. In this section, we propose such a discretization and show that they indeed preserve the properties such as mass conservation and non-negativity.

Notice that in mobility model (4.2)-(4.10), the number of people at any location $N_i(t)$ is always non-negative i.e. $N_i(t) \geq 0$ and it also remains below the population of P_i or the operating capacity $C_i^o(t)$, if it is an origin or a destination respectively i.e. $N_i(t) \leq Z_i$. Recall that $Z_i(t)$ denotes the population of i if i is an origin and the operating capacity C_i^o if it is a destination. Moreover, the total mass is also preserved. In other words, the total number of people in the system remains constant for any time t, we have $\sum_{k\in\mathcal{V}_o\cup\mathcal{V}_d} N_k(t) = \sum_{i\in\mathcal{V}_o} P_i$, which is equal to the sum of all population. To simulate the mobility model, we need to discretize it. The discrete-time version of the continuous-time model (4.2)-(4.9), should preserve the same mathematical properties than its continuous counterpart. Unfortunatelly, trival discretization may lead to loss of such properties. This section analyzes these problems and provides a solution.

Let $N_i(k)$ be the number of people at location i at time step k and Δt be the time-step size, then with forward Euler discretization of (4.2), we have the discrete-time model as

$$N_i(k+1) = N_i(k) + \Delta t \sum_{j\in\mathcal{N}_i}(\phi_{ji}(k) - \phi_{ij}(k)), \tag{4.12}$$

where, the flow $\phi_{ij}(k)$ is the flow from i to j as described in (4.3). For any flow, $\phi_{ij}(k) \geq 0$, the discrete-time model (4.12) preserves the total mass. It can be seen in the following proposition.

Proposition 4.1
Consider system (4.12) *with any flow* $\phi_{ij}(k) \geq 0$, *the total number of people in the system remains constant, i.e., for all time* $k, \sum_{i\in\mathcal{V}_o\cup\mathcal{V}_d} N_i(k) = \sum_{i\in\mathcal{V}_o\cup\mathcal{V}_d} N_i(0)$.

Proof. We prove it recursively, i.e. we show that $\mathbf{1}^T N(k+1) = \mathbf{1}^T N(k)$, where, $\mathbf{1} \in \mathbb{R}^n$ is the vector of all ones and $N(k) = [N_i(k)] \in \mathbb{R}^n$ for $n = |\mathcal{V}_o \cup \mathcal{V}_d|$.

Let us define, $\Phi \in \mathbb{R}^{n\times n}$ as

$$[\Phi]_{ij} = \begin{cases} \phi_{ij}(k) & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases},$$

and rewriting (4.12) in matrix form, we have

$$N(k+1) = N(k) + \Delta t \left(\Phi^T - \Phi\right) \mathbf{1}.$$

By left-multiplication with $\mathbf{1}^T$, we get

$$\mathbf{1}^T N(k+1) = \mathbf{1}^T N(k) + \Delta t(\mathbf{1}^T(\Phi^T - \Phi)\mathbf{1}).$$

We will obtain the desired result by showing that $\mathbf{1}^T(\Phi^T - \Phi)\mathbf{1} = 0$. To do so, we notice that $\mathbf{1}^T\Phi^T\mathbf{1} = (\mathbf{1}^T\Phi\mathbf{1})^T = \mathbf{1}^T\Phi\mathbf{1}$, where the last equality is true since $\mathbf{1}^T\Phi\mathbf{1}$ is a scalar. Therefore, we have

$$\mathbf{1}^T N(k+1) = \mathbf{1}^T N(k).$$

□

Now, we discuss how to define the flows $\phi_{ij}(k)$ to be used in (4.12). For time-step k, one option is to have flows $\phi_{ij}(k)$ which can be computed naively by plugging $N(k)$ in the same equations as in continuous time, namely,

$$\begin{aligned} \phi_{ij}(k) &= \min(\Delta_{ij}(k), \Psi_j(k)) \\ \Delta_{ij}(k) &= \delta_{ij}(k\Delta t) f_{ij}(k\Delta t) 1_{N_i(k)>0} \\ \Psi_j(k) &= \psi_j(k\Delta t) F_j 1_{N_j(k)<Z_j(k\Delta t)} \end{aligned} \tag{4.13}$$

where the functions $\delta_{ij}(k\Delta t), \psi_j(k\Delta t)$ and the $f_{ij}(k\Delta t)$ and F_j are defined in section 4.1. The discrete-time system (4.12) along with the flows defined as in (4.13), could result in two undesired scenarios, which are $N_i(k) < 0$ or $N_i(k) > Z_i(k\Delta t)$, for some i and k . Indeed, in continuous-time, the indicator functions that are present in the definition of Δ_{ij} and Ψ_i are enough to ensure that the demand Δ_{ij} is zero as soon as N_i is zero and that the supply Ψ_i is zero as soon as N_i reaches the capacity (or total population) $Z_i(k\Delta t)$. However, in discrete time, they are not enough to ensure that the total demand over a sampling interval does not exceed $N_i(k)$, nor that the supply over a sampling interval does not exceed the remaining capacity $Z_i(k\Delta t) - N_i(k)$, as we can easily understand by considering the following small example.

Example 4.1. *Let us consider a network of two nodes depicted in Figure 4.3.*

The equations for mobility between node 1 *and node* 2 *are given by*

$$\begin{aligned} \dot{N}_1 &= \phi_{21} - \phi_{12} \\ \dot{N}_2 &= \phi_{12} - \phi_{21} \end{aligned}$$

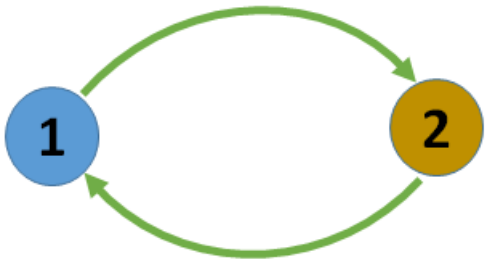


Figure 4.3: A network of two nodes.

Let us consider a simple case where $\phi_{21} = 0$ *and* $\phi_{12} = f_{12}1_{N_1>0}$. *Therefore, we have*

$$\dot{N}_1 = -f_{12}1_{N_1>0}$$
$$\dot{N}_2 = f_{12}1_{N_1>0}$$

This system, starting from some $N_1(t) > 0$, *will have* N_1 *decreasing up to some time* $\bar{t}$ *where* $N_1(\bar{t}) = 0$, *and then remaining equal to zero, as depicted in Figure 4.4a. We consider the non-trivial case where* $\bar{t}$ *is not a multiple of the sampling time* Δt, *i.e.,* $\bar{k}\Delta t < \bar{t} < (\bar{k}+1)\Delta t$ *for some* $\bar{k}$. *Now, since* $N_1(\bar{k}\Delta t) > 0$ *the discrete-time flow computed naively as in* (4.13) *will be* $\phi_{12}(\bar{k}) = \Delta t f_{12}$, *which leads to* $N(\bar{k}+1) < 0$ *as depicted in Figure 4.4a using starred black dots. Therefore, we need to introduce a correction. We saturate the flow so as to ensure that during a sampling time, we do not extract more flow than the available number of people. Hence, we set*

$$\phi_{12}(k) = \min\left(f_{12}, \frac{N_1(k)}{\Delta t}\right) \tag{4.14}$$

so that $\Delta t\phi_{12}(k) \leq N_1(k)$. *Therefore, in this example, we get* $\phi_{12}(\bar{k}) = N_1(\bar{k})$, *and hence* $N_1(\bar{k}+1) = 0$ *as depicted in Figure 4.4a using red circles. This correction is on the flow and not on the* N_1 *itself, which ensures that the total mass in the system is preserved, i.e.,* $N_1 + N_2$ *remains constant. Figure 4.4b depicts the different flows chosen according to flows defined in* (4.13) *and* (4.14). *It can be seen in Figure 4.4b that the corrected flow given by* (4.14) *adjusts the value of* ϕ_{12} *according to the available number of people.*

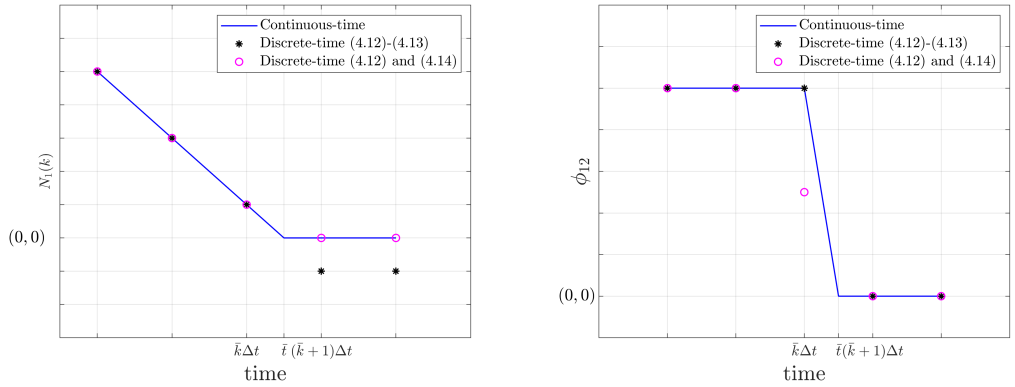
It can also be seen that in a complex network, no matter how small the step size be, this phenomenon can occur. Therefore, following the intuition within the toy example, we define suitable saturated flows, $\phi_{ij}(k)$ that are equal to the continuous-time flows whenever possible but ensures that the resulting model has the following properties

$$\Delta t \sum_{j\in\mathcal{N}_i} \phi_{ij}(k) \leq N_i(k)$$
$$\Delta t \sum_{j\in\mathcal{N}_i} \phi_{ji}(k) \leq Z_i(k\Delta t) - N_i(k)$$

for all k and for all i and hence ensuring $0 \leq N_i(k) \leq Z_i(k\Delta t)$ for all k.

The $\phi_{ij}(k)$ can be defined as

$$\phi_{ij}(k) = \min(\Delta_{ij}(k), \Psi_j(k)), \tag{4.15}$$



(a) (b)

Figure 4.4: (a) Evolution of number of people at node 1 in Example 4.1 and (b) The outflows from 1 to 2 in case of continuous-time model, depicted by blue line, and in discrete time when the flows are defined naively by using (4.13), depicted by black stars, versus when the flow is defined as proposed in (4.14) depicted by red circles.

where, $\Delta_{ij}(k)$ is defined as

$$\Delta_{ij}(k) = \min\left(\delta_{ij}(k\Delta t) f_{ij}(k\Delta t), \frac{N_i(k)\alpha_{ij}(k)}{\Delta t}\right), \tag{4.16}$$

where, $\delta_{ij}(k\Delta t)$ is the Demand Gating Profile(DGP) and $f_{ij}(k\Delta t)$is defined as in (4.6) and the $\psi_j(k\Delta t)$is the Supply Gating Functions defined as in (4.9). Here, $\alpha_{ij}(k)$ in the second term is the proportion in which the number of people will move to its neighbors. It can be computed as

$$\alpha_{ij}(k) = \frac{\delta_{ij}(k\Delta t) f_{ij}(k\Delta t)\psi_j(k\delta t)}{\sum_{q\in\mathcal{N}_i} \delta_{iq}(k\Delta t) f_{iq}(k\Delta t)\psi_q(k\Delta t)}$$

such that

$$\sum_{j\in\mathcal{N}_i} \alpha_{ij}(k) = 1.$$

The supply of j with respect to i $\Psi_{ij}(k)$ is defined as

$$\Psi_{ij}(k) = \psi_j(k\Delta t)\frac{(Z_j(k\Delta t) - N_j(k))\mu_{ij}(k)}{\Delta t} \tag{4.17}$$

Here, $\psi_j(k)$ is defined as in (4.9) and $\mu_{ij}(k)$ is the proportion in which the remaining capacity of the the location j will be divided among its neighbors defined as

$$\mu_{ij}(k) = \frac{\delta_{ij}(k\Delta t) f_{ij}}{\sum_{q\in\mathcal{N}_j} \delta_{qj}(k\Delta t) f_{qj}} \tag{4.18}$$

such that

$$\sum_{i\in\mathcal{N}_j} \mu_{ij}(k) = 1.$$

Notice that the second term in (4.16) ensures that if the number of people N_i at location i is less than the sum of outflows from i, then this sum is taken equal to the number of people available at i. If i has more than one neighbor then, as a natural choice, α_{ij} divides N_i among its neighbors in proportion to what they might have received proportional to their demands. Similarly, if at k, the supply of j is less than the demand of i and number of people at i, then the flow ϕ_{ij} will be equal to the remaining capacity at j divided proportionally by μ_{ij} among its neighbors what they might have received proportional to their demands. The discrete-time model (4.12) with the flows as in (4.15)-(4.18) is same as the continuous time model (4.2) whenever possible and ensures that the number of people at any location i, $N_i(k)$ is non-negative and remains bounded by $Z_i(k\Delta t)$. Now, we propose the following result to support this claim.

Proposition 4.2
Consider the discrete-time system (4.12),(4.15)-(4.18),*with the initial condition* $0 \leq N_i(0) \leq Z_i(0)$ *for all* i. *Then, for all time* k, *we have* $0 \leq N_i(k) \leq Z_i(k\Delta t)$ *for all* i.

Proof. We prove it recursively that is we show that if $0 \leq N_i(k) \leq Z_i(k\Delta t)$, then $0 \leq N_i(k+1) \leq Z_i((k+1)\Delta t)$.

First, notice that from (4.15)-(4.18), $0 \leq N_i(k) \leq Z_i(k\Delta t)$, ensures $\phi_{ij}(k) \geq 0, \forall i, j$ since it can be seen from (4.16) that $\delta_{ij}(k\Delta t) f_{ij}(k\Delta t) \geq 0$ and $\frac{N_i(k)\alpha_{ij}(k)}{\Delta t} \geq 0$.

Now, from the equations (4.15)-(4.17), we have that

$$\phi_{ij}(k) \leq \frac{Z_j(k\Delta t) - N_j(k)}{\Delta t}\mu_{ij},$$
$$\phi_{ij}(k) \leq \frac{N_i(k)\alpha_{ij}(k)}{\Delta t}.$$

Therefore,

$$\Delta t \sum_{i\in\mathcal{N}_j} \phi_{ij}(k) \leq \sum_{i\in\mathcal{N}_j} \mu_{ij}(Z_j(k\Delta t) - N_j(k)),$$
$$\Delta t \sum_{j\in\mathcal{N}_i} \phi_{ij}(k) \leq \sum_{j\in\mathcal{N}_i} \alpha_{ij}(N_i(k)).$$

Since, $\sum_{i\in\mathcal{N}_j} \mu_{ij}(k) = 1$, and $\sum_{j\in\mathcal{N}_i} \alpha_{ij}(k) = 1$, we have

$$\Delta t \sum_{i\in\mathcal{N}_j} \phi_{ij}(k) + N_j(k) \leq Z_j(k\Delta t), \tag{4.19a}$$
$$\Delta t \sum_{j\in\mathcal{N}_i} \phi_{ij}(k) \leq N_i(k). \tag{4.19b}$$

Now from (4.12), we have the discrete model with the corrected

$$N_j(k+1) = N_j(k) + \Delta t \sum_{i\in\mathcal{N}_j} \phi_{ij}(k) - \Delta t \sum_{i\in\mathcal{N}_j} \phi_{ji}(k)). \tag{4.20}$$

From (4.20),(4.19a) and since $\phi_{ji}(k) \geq 0$, we have

$$N_j(k+1) \leq Z_j((k+1)\Delta t), \forall j.$$

Moreover, from (4.20), (4.19b), and since $\phi_{ji}(k) \geq 0$, we have that

$$N_i(k+1) \geq 0, \forall i.$$

Therefore, we have that for all $i \in \mathcal{V}_o \cup \mathcal{V}_d$

$$0 \leq N_i(k+1) \leq Z_i((k+1)\Delta t).$$

□

One of the advantages of the model (4.2) is that it is modular. This mobility model can be used for many applications such as urban planning, studying epidemic spread etc. In the next section, we descritize the SIR-mobility model, which incorporates mobility and the SIR epidemic spread model, to implement it to the large scale mobility network of Grenoble.

4.4 Discrete-time SIR-mobility model

Discretization of (4.11), naively poses the issue that the number of susceptible, infected and recovered people become negative for some time steps. Moreover, losing non-negativity might lead to losing boundedness of the variables, which is ensured in (4.11) and (4.2)-(4.9) where we have $N_i(t) \geq 0$ and $S_i(t) + I_i(t) + R_i(t) = N_i(t)$. First of all, we should use the discretized mobility model that ensures physically meaningful flows, as in Section 4.3. This is a key ingredient to ensure non-negativity in SIR-mobility, but is not enough, we also need a suitable discretization of the SIR dynamics. This can be done using the technique from the paper [SD10]. Inspired from [SD10], we propose a discretization of (4.11) which ensures that all the variables remain non-negative and the total mass is preserved. It can be called non-local as it differs from the forward Euler discretization by taking some variables at time $k+1$ on the right-hand side instead of taking them all at the local time k.

Let Δt be the time step size then the continuous-time SIR-mobility model (4.11) can be discretized as follows.

$$\frac{S_i(k+1) - S_i(k)}{\Delta t} = -\beta_i(k) S_i(k+1) \frac{I_i(k)}{N_i(k)} + \sum_j \left(\phi_{ji}(k) \frac{S_j(k)}{N_j(k)} - \phi_{ij}(k) \frac{S_i(k)}{N_i(k)} \right) \tag{4.21a}$$

$$\frac{I_i(k+1) - I_i(k)}{\Delta t} = \beta_i(k) S_i(k+1) \frac{I_i(k)}{N_i(k)} - \gamma I_i(k+1) + \sum_j \left(\phi_{ji}(k) \frac{I_j(k)}{N_j(k)} - \phi_{ij}(k) \frac{I_i(k)}{N_i(k)} \right) \tag{4.21b}$$

$$\frac{R_i(k+1) - R_i(k)}{\Delta t} = I_i(k+1) + \sum_j \left(\phi_{ji}(k) \frac{R_j(k)}{N_j(k)} - \phi_{ij}(k) \frac{R_i(k)}{N_i(k)} \right). \tag{4.21c}$$

Rearranging the terms, one can obtain the following equations that can be implemented to actually compute $S_i(k+1)$, $I_i(k+1)$ and $R_i(k+1)$.

$$S_i(k+1) = \frac{1}{1+\Delta t\beta_i(k)\frac{I_i(k)}{N_i(k)}}\left[S_i(k) + \Delta t\sum_j\left(\phi_{ji}(k)\frac{S_j(k)}{N_j(k)} - \phi_{ij}(k)\frac{S_i(k)}{N_i}(k)\right)\right] \tag{4.22a}$$

$$I_i(k+1) = \frac{1}{1+\Delta t\gamma}\left[I_i(k) + \Delta t\beta_i(k)S_i(k+1)\frac{I_i(k)}{N_i(k)} + \Delta t\sum_j\left(\phi_{ji}(k)\frac{I_j(k)}{N_j(k)} - \phi_{ij}(k)\frac{I_i(k)}{N_i(k)}\right)\right] \tag{4.22b}$$

$$R_i(k+1) = R_i(k) + \Delta t\gamma I_i(k+1) + \Delta t\sum_j\left(\phi_{ji}(k)\frac{R_j(k)}{N_j(k)} - \phi_{ij}(k)\frac{R_i(k)}{N_i(k)}\right). \tag{4.22c}$$

It is well-known that in standard SIR model, the sum of susceptible, infected and recovered people remains constant, and equal to the total population. Here, in each location, the number of people is not constant because of the mobility, but we can still show that the sum $S_i(k) + I_i(k) + R_i(k)$ is equal to $N_i(k)$ for all i and k, as shown in the following proposition.

Proposition 4.3
Given the discrete model (4.22), (4.12)-(4.18) *with the initial condition* $S_i(0)+I_i(0)+R_i(0) = N_i(0)$*, then for all* i *and* k*, we have* $S_i(k) + I_i(k) + R_i(k) = N_i(k)$.

Proof. We prove it recursively, i.e., we prove that $S_i(k) + I_i(k) + R_i(k) = N_i(k)$, implies $S_i(k+1) + I_i(k+1) + R_i(k+1) = N_i(k+1)$.

Recall that system (4.22) is equivalent to system (4.21); yet another equivalent re-writing is the following:

$$S_i(k+1) = S_i(k) - \Delta t\beta_i(k)S_i(k+1)\frac{I_i(k)}{N_i(k)} + \Delta t\sum_j\left(\phi_{ji}\frac{S_j(k)}{N_j(k)} - \phi_{ij}(k)\frac{S_i(k)}{N_i}(k)\right)$$

$$I_i(k+1) = I_i(k) + \Delta t\beta_i(k)S_i(k+1)\frac{I_i(k)}{N_i(k)} - \Delta t\gamma I_i(k+1) + \Delta t\sum_j\left(\phi_{ji}(k)\frac{I_j(k)}{N_j(k)} - \phi_{ij}(k)\frac{I_i(k)}{N_i}(k)\right)$$

$$R_i(k+1) = R_i(k) + \Delta t\gamma I_i(k+1) + \Delta t\sum_j\left(\phi_{ji}(k)\frac{R_j(k)}{N_j(k)} - \phi_{ij}(k)\frac{R_i(k)}{N_i(k)}\right).$$

We can now sum the three equations above. Then, we use the fact that $S_i(k) + I_i(k) + R_i(k) = N_i(k)$ and $S_j(k) + I_j(k) + R_j(k) = N_j(k)$, and hence also $\frac{S_i(k)+I_i(k)+R_i(k)}{N_i(k)} = 1$ and $\frac{S_j(k)+I_j(k)+R_j(k)}{N_j(k)} = 1$. With this, we obtain

$$\begin{aligned} S_i(k+1) + I_i(k+1) + R_i(k+1) &= N_i(k) + \Delta t\sum_{j\in\mathcal{N}_i}(\phi_{ji}(k) - \phi_{ij}(k)) \\ &= N_i(k+1). \end{aligned}$$

where the last equality follows from (4.12).

□

Now, we show that if the initial number of susceptible, infected and recovered people is non-negative then they remain non-negative at any time-step k.

Proposition 4.4
Given the discrete model (4.22), (4.12)-(4.18) *with the initial condition* $S_i(0) \geq 0, I_i(0) \geq 0, R_i(0) \geq 0, N_i(0) \geq 0$, *then for all* i, k *we have* $S_i(k) \geq 0$, $I_i(k) \geq 0$ *and* $R_i(k) \geq 0$.

Proof. First, recall that by proposition 4.2, $N_i(0) \geq 0$ implies $N_i(k) \geq 0$, for all k. Moreover, recall that the definition of flows in (4.12)-(4.18) together with $N_i \geq 0$ ensures (4.19a)-(4.19b).

With this, we are ready to show recursively, the non-negativity of $S_i(k), I_i(k)$ and $R_i(k)$, namely we will show that $S_i(k) \geq 0$ implies $S_i(k+1) \geq 0$, and then similarly show that $I_i(k) \geq 0$ implies $I_i(k+1) \geq 0$ and $R_i(k) \geq 0$ implies $R_i(k+1) \geq 0$. From (4.22a), we have

$$\begin{aligned} S_i(k+1) &= \frac{1}{1+\Delta t\beta_i(k)\frac{I_i(k)}{N_i(k)}}\left[S_i(k) + \Delta t\sum_j\left(\phi_{ji}(k)\frac{S_j(k)}{N_j(k)} - \phi_{ij}(k)\frac{S_i(k)}{N_i(k)}\right)\right] \\ &= \frac{1}{1+\Delta t\beta_i(k)\frac{I_i(k)}{N_i(k)}}\left[S_i(k) - \Delta t\sum_j\phi_{ij}(k)\frac{S_i(k)}{N_i(k)} + \Delta t\sum_j\phi_{ji}(k)\frac{S_j(k)}{N_j(k)}\right] \\ &= \frac{1}{1+\Delta t\beta_i(k)\frac{I_i(k)}{N_i(k)}}\left[\frac{S_i(k)}{N_i(k)}\left(N_i(k) - \Delta t\sum_j\phi_{ij}(k)\right) + \Delta t\sum_j\phi_{ji}(k)\frac{S_j(k)}{N_j(k)}\right] \end{aligned}$$

from (4.19b), we have $N_i(k) - \Delta t\sum_j \phi_{ij}(k) \geq 0$, and since $S_j(k) \geq 0, N_j(k) \geq 0$ and $\phi_{ji}(k) \geq 0$ for all j, therefore,

$$S_i(k+1) \geq 0.$$

Similarly, from (4.19b), we have $N_i(k) - \Delta t\sum_j \phi_{ij}(k) \geq 0$, which also ensures that $I_i(k) - \Delta t\sum_j \phi_{ij}(k)\frac{I_i(k)}{N_i(k)} \geq 0$, and $R_i(k) - \Delta t\sum_j \phi_{ij}(k)\frac{R_i(k)}{N_i(k)} \geq 0$. Therefore, it can be seen from (4.22b) and (4.22c) that $I_i(k+1) \geq 0$ and $R_i(k+1) \geq 0$ respectively. □

It can be seen that the proposition 4.3 and the proposition 4.4 together ensure that S_i, I_i, and R_i remain bounded, since $0 \leq S_i(k) \leq N_i(k)$, $0 \leq I_i(k) \leq N_i(k)$ and $0 \leq R_i(k) \leq N_i(k)$.

Notice that, in the right hand side of (4.21), the terms $S_i(k+1)$ and $I_i(k+1)$ outside the parenthesis have been taken at $k+1$ instead of k and in proposition 4.4, we show that with this discretization, we have non-negative $S_i(k), I_i(k)$ and $R_i(k)$. Recall that in the SIR dynamics, people are transferred from one compartment to another depending on their infection status. In the discretization (4.21), it is ensured that this transfer is reflected by diminishing the number in one compartment by multiplication with a fraction as can be seen in (4.22). On the other hand, in Euler discretization of SIR-mobility model, this transfer between compartments is done by using a subtraction term as for instance can be seen in (4.23) for the variable S.

$$S_i(k+1) = S_i(k) - \Delta t\beta_i\frac{S_i(k)I_i(k)}{N_i(k)} + \Delta t\sum_{j\in\mathcal{N}_i}\left(\phi_{ji}(k)\frac{S_j(k)}{N_j(k)} - \phi_{ij}(k)\frac{S_i(k)}{N_i(k)}\right) \tag{4.23}$$

In (4.23), the term $-\Delta t\beta_i \frac{S_i(k)I_i(k)}{N_i(k)}$ along with the term $-\Delta t \sum_{j\in\mathcal{N}_i}\left(\phi_{ij}(k)\frac{S_i(k)}{N_i(k)}\right)$ can sometime lead to negative S depending on the time-step size Δt. Instead, in the proposed discretization, the evolution of S is given by (4.22a), where we see that the transfer of people out of this compartment is reflected by diminishing the number through multiplication by a fraction. Therefore, ensuring the non-negativity of S as the non-negativity of the terms in the the square-bracket of (4.22a) is ensured because of the flow defined in the section (4.3).

4.5 Concluding remarks

In this chapter, we consider a supply-demand based mobility model which captures the daily movement of people between residences and places of interests called destinations, using time schedules and gating profiles. This model also accommodates the possibility of imposing restrictions on mobility which when integrated with an epidemic spread model, can be utilized for epidemic mitigation.

These models are given in continuous-time in which the flows are defined such that the model has nice properties of non-negativity, boundedness and mass conservation. For their implementation to large-scale networks, the models needs to be discretised. Therefore, we analysed the problems encountered in the discretization of this model and proposed to redefine the flows in discrete-time which preserves the properties of non-negativity, boundedness and mass conservation. Moreover, we also provide a dicretization of the SIR-mobility model, which integrates the mobility and epidemic spread and also preserves the nice properties of boundedness and non-negativity.

We will calibrate the parameters of the mobility model presented here in Chapter 5. Both the models in discrete-time presented here will be used to devise optimal control policies for epidemic mitigation in Chapter 6.

Chapter 5

Human mobility model calibration

Contents

5.1 Origin-destination network 60
5.1.1 Origins 61
5.1.1.1 Partition of Grenoble *commune* 61
5.1.1.2 Location and population of origin nodes 62
5.1.2 Destinations 62
5.1.2.1 Locating the destination nodes 63
5.1.2.2 Classification of destinations 64
5.1.2.3 Schools 64
5.1.2.4 Hospitals 65
5.1.2.5 Workplaces 69
5.1.2.6 Shopping centers 71
5.1.2.7 Leisure 74
5.1.2.8 Aggregation of nodes 76
5.1.3 Mobility network 78
5.1.3.1 Attraction based rules 78
5.2 Time dependent profiles 85
5.2.1 Destination schedule, mobility window and average time spent 85
5.2.2 Demand Gating Profile (DGP) 86
5.3 Number of people 88
5.3.1 Daily capacity of the destinations 88
5.3.2 Average daily number of people from origins to destinations 89
5.4 Simulation Example 89
5.5 Concluding remarks 91

In this chapter, we provide methodologies to compute the parameters of the mobility model described in Chapter 4 to implement it in the Grenoble metropolis. First, we identify the origins and destinations and then build the mobility network between them before computing the parameters required in the model.

Grenoble is a metropolitan city in France located in the valley of the Alpes mountain range. A *métropole* (French for "metropolis") is an administrative entity in France, in which several

communes cooperate, and which has the right to levy local taxes. Grenoble-Alpes métropole is centered on the city of Grenoble. The area which we consider in this work includes 55 *communes* including all the 49 *communes* in Grenoble-Alpes métropole and 6 *communes* in Gresivaudan, which are of economic importance to the Grenoble *métropole.* The population of this area is about 500000 and has a surface area of about $600km^2$. We refer to this area as Grenoble area here-onwards. In order to implement the mobility model (4.12),(4.15)-(4.18) to a large-scale network of a metropolitan city, we need the following parameters:

1. Map of the *communes* in the area.
2. Origin nodes $\mathcal{V}_o$ and destination nodes $\mathcal{V}_d$.
   (i) Location of all the nodes in $\mathcal{V}_o$ and $\mathcal{V}_d$.
   (ii) Population P_i, $\forall\ i \in \mathcal{V}_o$.
   (iii) Capacities C_j, $\forall\ j \in \mathcal{V}_d$.
3. OD matrix $\mathcal{O}_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$, to decide which origins and destinations are connected to each other.
4. For destination j in each subcategory $\mathcal{D}_c^\alpha$ for each day of the week.
   (i) Time schedules to define DGP and SGF
      – opening times (a_j),
      – closing times (b_j) and
      – average time spent (s_j), which will be described later and used to define DGP for some destination categories.
      – mobility window $[t_{ij}, t_{ij} + \tau_{ij}]$
   (ii) Demand Gating Profiles(DGPs) $\delta_{ic}(t)$ for and Supply Gating Functions(SGFs).
5. For each edge $(i,j) \in \mathcal{E}$, M_{ij}, which is defined the average number of people going from i to j daily.

We will discuss how to retrieve these parameters to build the large-scale mobility network of Grenoble from the public information in detail in the following sections. In particular, we will discuss the parameters of the three components of mobility: 1. Where? 2. When? 3. How many? In 'where', we consider the map, nodes $\mathcal{V}_o$ and $\mathcal{V}_d$ and the OD matrices for each subcategory. In 'when,' we compute the time schedules and the gating profiles, and finally, in 'How many', we compute M_{ij} for the destinations in each subcategory.

5.1 Origin-destination network

At first, we need the map of the communes in the area and need tools to manipulate it. For that purpose, we use an open source software called QGIS. This software can be used to make,

edit/manipulate or display maps and their features. In this work, the main source of data for maps is OpenStreetMap [Opeb]. Openstreetmap is an open source collaborative project to create a free editable map of the world and the geodata underlying the map is considered the primary output of the project. The main source of statistical data is INSEE [INSb]. INSEE is a government agency that conducts a census to collect socio-economic information from the people in France. The other sources, if any, will be indicated in the details corresponding to the different origins or destinations.

The coordinates of the boundaries of all the French *communes* can be downloaded from [RFa] in GeoJSON format. The GeoJSON format is useful to display different information on a map. This GeoJSON file contains name and coordinates for each *commune*. We use GeoJSON files to locate the origin and destination nodes also as it helps us store any kind of information corresponding to any location on the map and can be easily manipulated in 'Python' or simple text editors like 'Notepad++'.

5.1.1 Origins

The residential areas of the communes are considered to be the origins. For mobility, we aim to study the movement of the population of this origins. The population data for the *communes* is retrieved in form of a table from INSEE [INSd]. Note that, in the area of our study, Grenoble is the most populated commune with 160778 residents and the Mont-Saint-Martin is the least populous commune with 79 residents. Moreover, the population of Grenoble is almost five times that of the second most populous commune, therefore, in the implementation, we consider a division of Grenoble, the *commune* which is basically located at the center of the metropolis, into 6 parts which are called *sectors*. Therefore, the number of origins increases from 55 to 60.

5.1.1.1 Partition of Grenoble *commune*

Grenoble *commune* is divided into 6 'sectors', that regroups different neighbourhoods. The sectors defines certain rules/regulations that affect daily life of the residents such as schools, voting etc. This division is done by grouping smaller partitions called IRIS. IRIS is a partition done by the government to identify an area with population greater than 10000 or with population between 5000 to 10000. In Grenoble, there are 70 IRIS. We obtain the coordinates of these sectors by regrouping IRIS whose coordinates can be found on [Opea]. The grouping of IRIS has been done according to the division given on the official webpage of Grenoble [Gma]. In QGIS software, it can be done by taking union of polygons formed by the boundaries of IRIS.

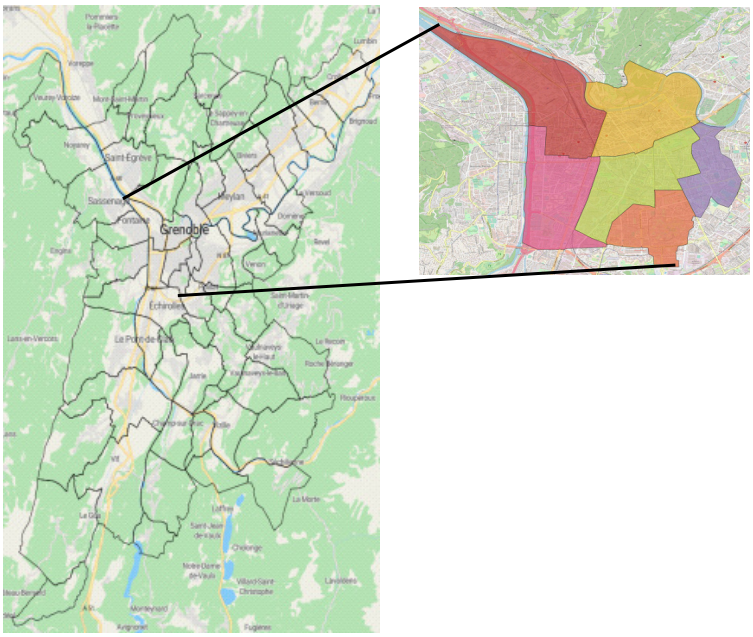


Figure 5.1: Zones considered in our study: 54 *communes* and the 6 sectors in which Grenoble has been partitioned. In the inset, we can see the partition of Grenoble *commune* into six sectors.

Different datasets for these sectors can be retrieved from [Gma]. Now that, Grenoble *commune* has been divided into 6 sectors, from here onwards, the *communes* and these sectors of Grenoble will be referred as zones which are depicted in Figure 5.1.

5.1.1.2 Location and population of origin nodes

For the graphical representation and for implementation of the model, each zone is represented by a node which is positioned in the residential area. This positioning of the nodes has been done manually inspired by the position of nodes on a INSEE [INSe] or by locating the residential areas on the online maps manually. Each origin node has a population P_i of the corresponding zone. The population of the communes has been taken from [INSd] and that for the partition of Grenoble from [Gmc]. Moreover, the population at each origin according to the age groups $0-15, 16-24, 25-64$ can be obtained from [Ins]. For simplicity, we assume that these age groups are uniformly distributed. Therefore, to find the number of children at any origin between the age $5-10$, for instance, we divide the population of age group $0-15$ in that origin by 3. In this way, we obtain the population of different age groups.

5.1.2 Destinations

Destinations comprise of the places where people go for some time during the day or night for a short period of time and then return to their residences. We denote them by a node on the

map. These include working places, schools, hospitals etc. First, in the following subsection, we provide the classification of these destination nodes. Then, in the following subsection, we discuss the method for locating the destination nodes. Finally, when we move to the section of a particular destination category or subcategory, we mention the location method and discuss mainly the strategies or methods for computing the capacities of the destination nodes.

5.1.2.1 Locating the destination nodes

Unlike the origin nodes, the data for location is not always available, so we apply different strategies to locate various classes of destination nodes. We explain these methods in detail in the following:

Method 1: using Opentstreetmap data In general, the objects or entities on Openstreetmap are labeled correctly but many a time some objects are labeled incorrectly since it is open-source. It makes the retrieval of data in categories difficult. In this method, we follow the following steps:

1. Export the map data of a squared region from [Opeb] by using the export option on the website. Since the region of selection is large in our case, we need to use overpass API or planet OSM from the left panel to import the map data. On exporting, it downloads all the data for the chosen region and contains many information like bus stops, shops, trees, parks, fountains, hospitals etc., along with many attributes such as location, opening and closing times, websites etc.

2. Since the downloaded data is for a squared region, we must select only the data concerned with the Grenoble area. For that, we use the 'intersection tool' of QGIS software. It is used to obtain all the common attributes from two given layers if they share some area in common. The output is one layer with all the attributes at the intersection of these two layers. Here, we use this tool with two input layers, one being the map data for a square region downloaded from OSM and the other being the boundaries of our region as depicted in Figure 5.1.

3. Once we have the desired data for our region as an output of the previous step, we need to filter out the nodes of the desired category by using the 'filter tool' in QGIS. The filter tool uses some SQL queries to extract the data matching our queries.

4. Once filtered, we need to remove unwanted attributes and keep only the wanted attributes such as location or names etc. we also need to check for garbage nodes. Sometimes, there are too many garbage values because some contributors do not put the correct labels. If we find any garbage nodes, we need to remove them by applying the filter again. There is also a chance of missing some nodes of the same type because all the nodes are not labeled correctly.

5. Save the file in GeoJSON format using QGIS.

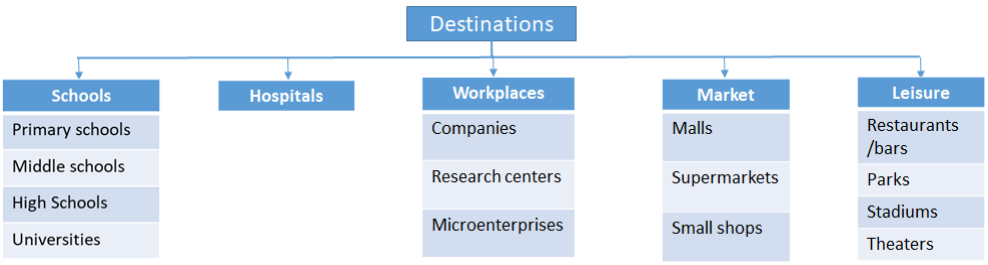


Figure 5.2: Classification of destinations in categories and subcategories.

Method 2: Manually by creating and editing a GeoJSON file directly If the coordinates and other data of the destinations are available from some website, we make a GeoJSON file directly by writing the coordinates, names and capacity manually. Here, the GeoJSON files have been edited using Notepad++. If needed, they can also be edited later directly in Notepad++ or by editing the attributes in QGIS.

Method 3: Locating the destinations manually in QGIS For some destinations, neither the location coordinates nor OSM data is available. In these cases, we locate the nodes manually in QGIS using the node editing tool in a layer. While locating the node in QGIS, the location of the corresponding destination in OSM map can be taken as a reference. Once the nodes are located, we export the layer in GeoJSON format and the other attributes are edited later either in Notepad++ or in QGIS as per convenience.

5.1.2.2 Classification of destinations

The destinations have been classified into five categories and each of them into some subcategories which can be seen in Figure 5.2. It can be seen that we have 5 categories and 15 subcategories in total. Throughout the chapter, we will find the parameters for each subcategory. At first, we will discuss the location and capacities of each destination class. All the figures in this section depicting the destination nodes have been taken from the simulation platform which will be described in Chapter 7. The position of the nodes are their actual locations.

5.1.2.3 Schools

In this category, we have all the educational institutions such as

1. Primary schools
2. Middle schools
3. High schools

4. Universities

Primary Schools In this subcategory, we consider all the primary schools in the region. It includes *école maternelle, école élémentaire and école primaire.* There is at least one primary school in each zone. The nodes in this category have been located manually using method 2 as described in 5.1.2.1. For their capacities, we take the total number of students studying there. The location and capacity for primary schools has been taken from [AGb] by searching for 'école' in each zone. For each zone, this website provides a complete list of such schools with links to their websites. Each school website contains relevant information such as name, address, location coordinates, the total number of students, etc. We manually input these data in the GeoJSON file for primary schools.

Middle and high schools In these two subcategories, we consider all the middle schools*(collège)* and high schools *(Lycée)* in our region. They have been located by method 2 as described in 5.1.2.1. For their capacities, we consider the total number of students studying there. The location and capacity for the colleges and Lycee have been retrieved from the website [AGc] by following the same procedures as that for the primary schools.

Universities Here, we consider the sites of University Grenoble Alpes namely, Campus in Saint Martin d'Hères, Grenoble INP and IUTs in Grenoble downtown. We locate them by method 3 as described in 5.1.2.1. Here, the total capacity is defined as the total number of students and administrative or technical staffs which is 59600 as mentioned on the university website [UGA]. For the capacity of different sites, let C_u = total capacity = 59600. We divide the total capacity to obtain the capacities for different sites as follows:

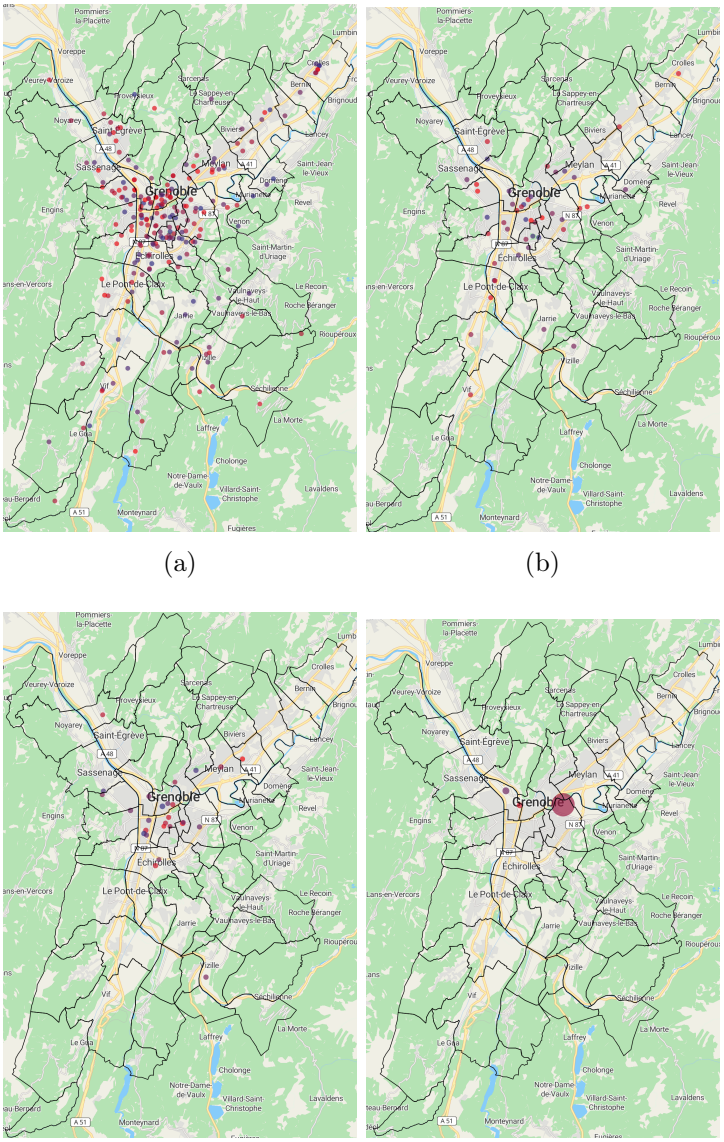
1. Campus- 85% of C_u,
2. Grenoble INP- 10% of C_u,
3. IUTs- 5% of C_u.

This division has been done keeping in mind that most of the students visit the campus daily. Finally, Figure 5.3 depict the nodes in different subcategories of schools.

5.1.2.4 Hospitals

In this category, we have considered the hospitals *Centre hospitalier universitaire (CHU) nord, CHU sud, Clinique belledonne, Clinique des Cèdres* and *Clinique mutualiste.* We locate each of these hospitals by method 3 as described in 5.1.2.1. The capacity of each hospital is considered to be the sum of the following:

1. total number of available beds.



(a) (b)

(c) (d)

Figure 5.3: Nodes for destinations in school category. (a) Primary school nodes. (b) Middle school nodes. (c) High school nodes. (d) University nodes.

2. total number of staffs including the doctors.

3. total number of daily consultations.

4. total number of visitors accompanying the patients. Information about the first three is more or less available in some form, which will be discussed for each hospital. The data regarding the number of visitors need to be estimated based on the first three. So, here we discuss the strategy for this estimation. For the number of visitors to a hospital, we need to consider that there are three kinds of patients who visit the hospital:

   (a) consultations only.

   (b) ambulatoire - they stay at the hospital for half a day to a full day mostly for minor surgeries.

   (c) hospitalization - the patients who need to be hospitalized and stay at the hospital for six days on an average. This information has been retrieved from [CHUa].

   Due to the lack of data regarding the number of visitors, we can follow the following rule of thumb.

   - For consultation and ambulatoire - zero visitors for 80 percent and 1 visitor for 20 percent.
   - For hospitalized patients - one or two visitors per patient. The maximum limit of visitors is 2 per patient at CHU Voiron [CV]. Due to lack of this data for others, we take this rule in general for other hospitals as well. To compute the visitors against hospitalized patients, we assume that 80% of the beds are occupied.
   - Therefore, the general rule for number of visitors in a hospital is as follows:

$$\begin{aligned} &\text{number of visitors} \\ &\quad = 0.2 * (\text{number of consultations} + \text{patients in ambulatoire}) \\ &\quad + 1.5 * 0.8 * (\text{total number of beds.}) \end{aligned} \tag{5.1}$$

Now, we discuss the data for $1, 2$ and 3 in details for each hospital in the following sections.

CHU nord and sud There are two CHUs in the region which are CHU nord located in La Tronche and the other is CHU sud located in Echirolles. These are the main and largest public hospitals in the Grenoble area. Annual data for three CHUs (CHU nord, CHU sud and CHU Voiron) are available collectively on [CHUa]. Here, we consider the latest available annual data which is for 2018. This data is for three hospitals (CHU Voiron, CHU nord and CHU sud) but only CHU nord and CHU sud come under our area. The number of beds in CHU voiron is given on the website [CHUb] so, at first, we divide the data in two parts (one for CHU Voiron and the other for both CHU nord and sud) in proportion to the number of beds. Since, CHU nord is larger than CHU sud, the remaining data is divided such as two-thirds are given to CHU nord and the remaining one-third to CHU sud. The number of beds are given as follows:

Data category	All CHU	CHU nord	CHU sud
Number of beds	2133	1222	611
Number of employees	9000	5156	2568
daily consultations and *ambulatoire*	2677	1534	767
Visitors		1773	886

Table 5.1: Computation of data for CHU nord and CHU sud

- Total number of beds in three CHUs = 2133.
- Total number of beds in CHU Voiron = 300.
- the proportion by which other data needs to be divided in order to obtain them for CHU nord = $\frac{2}{3} * \frac{1833}{2133}$.
- the proportion by which other data needs to be divided in order to obtain them for CHU sud = $\frac{1}{3} * \frac{1833}{2133}$.

Now, we compute all the data for CHU which is depicted in Table 5.1:

Clinique mutualiste Clinique mutualiste de Grenoble is a non-profit Private Health Establishment of Collective Interest (ESPIC) participating in the public hospital service. The data for this hospital has been retrieved from the website [CM]. The number of visitors has been computed using (5.1). Table 5.2 contains the capacities of this hospital.

Clinique Belledonne Clinique Belledonne is a private hospital located in Saint Martin d'Heres. It has 290 beds and around 650 employees [CB]. The data regarding the daily consultations is not available, so it is computed in proportion to the number of doctors by considering the data for CHU as the ground rule. There are 2677 consultations per day at CHU which has 2000 physicians. In this proportion, Clinique Belledonne, which has 150 physicians, will have 201 consultations and ambulatoire patients daily. The number of visitors has been computed using (5.1). The capacity of this hospital is listed in Table 5.2.

Clinique des Cèdres Clinique des Cèdres is a private hospital located in Echirolles. It has 200 beds as mentioned on the website [CC]. No other relevant data other than the number of beds is available to us, so we compute them in proportion to the number of beds taking the data for Clinique Belledonne as ground rule. The number of beds in Clinique Belledonne is 290 and that in Clinique des Cèdres is 200 so the proportion by which the data of Clinique is multiplied in order to get that for Clinique des Cèdres is $\frac{200}{290}$. Please see Table 5.2 for the complete data. Figure 5.4 depict the nodes corresponding to hospitals.

Hospital name	Beds	Consultations and *ambulatoire*	Employees	Visitors	Total
CHU nord	1222	1534	5156	1773	9685
CHU sud	611	767	2568	886	4832
Clinique mutualiste	436	286	1300	581	2603
Clinique Belledonne	290	201	650	348	1489
Clinique des Cèdres	200	136	442	237	1015

Table 5.2: Capacities of hospitals

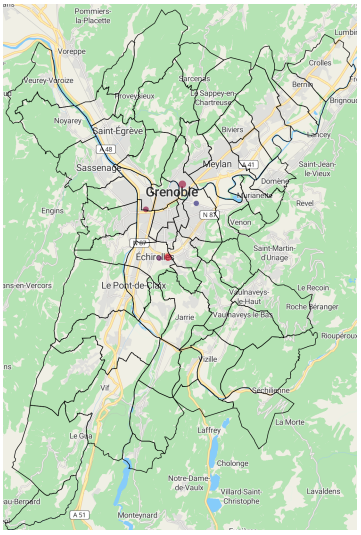


Figure 5.4: Hospital nodes

5.1.2.5 Workplaces

The category workplaces has the following three subcategories:

1. Companies
2. Research centers
3. Microenterprises and others

In France, INSEE [INSa] classifies the 'enterprises' on basis of number of employees as follows:

- Grandes enterprises (GE) - A GE has more than 5000 employees and 25% of the total employees work there.

- Entreprises de taille intermédiaire (ETI)- An ETI has less than 5000 but more than 250 employees and 25% of the total employees work there.
- Petites et moyennes entreprises (PME) - A PME has less than 250 but more than 9 employees and 30% of the total employees work there.
- Microenterprises (MIC) - An MIC has less than 10 employees and 20% of the total employees work there.

We consider GE, ETI and PME in companies subactegory and MIC in Microenterprises and others along with some public offices.

Companies Here, we consider mainly the private companies. Due to lack of information on the list of companies and number of employees working in them, we consider the list of companies provided in [Pre] with employees greater than or equal to 120. The destination nodes corresponding to the companies from this list have been located manually using method 3 as stated in 5.1.2.1. We have also added some other companies located in the same commercial zones as visible in the map on QGIS. Figure 5.5a depicts the companies considered. Here, the capacity is defined to be the total number of employees working in the companies. In this subcategory, we consider the GE, ETI and PMEs which employees around 80% of the total employees. Since, we consider only the companies with more than 120 employees, we assume that we have included only all GE and ETI but only half of the PMEs in our list, that is we have only considered 65% of the total employees. Therefore, we need to inflate this number which is done as follows:

Let T_e = total number of people working in private companies and A_m = sum of employees in companies from the magazine. Since, the magazine gives us 65% of T_e, therefore, T_e can be computed as

$$T_e = \frac{100}{65} * A_m. \tag{5.2}$$

According to INSEE, people working in GE + ETI + PME= 80%T_e. Since, we assume to have only 65% of GE+ETI+PME from the magazine, we need to inflate the capacities of the companies to 80%. Since, 65% of $T_e = A_m$, then 80% of $T_e = \frac{80}{65} * A_m$. Finally, for the individual companies, we have that

$$\begin{aligned} &\text{the inflated capacity of the companies} \\ &\quad = \frac{80}{65} * \text{ original no. of employees from the magazine.} \end{aligned} \tag{5.3}$$

Some Companies like ST microelectronics have more than one sites in the area. We compute the capacity of each site by dividing the total number of employees in that company by the number of sites. To the companies considered apart from the ones from [Pre], an average capacity of 120 has been given.

Research centers In this category, we consider the research labs listed on [GI] and some others which are associated with the University Grenoble Alpes. They have been located manually by method 3 as explained in 5.1.2.1. The capacity of each research center is the sum of the number of doctoral students, researchers and technical staffs working in each center. This information has been taken from the respective websites of each of these research centers.

MIC and others In this subcategory, we include the public offices and microenterpises (MIC). They are represented by a node in each *commune*. The capacity of each node is sum of the employees of MIC and employees in public offices in the corresponding *communes*.

Let $M_e = 20\%$ of T_e = the total number of employees in microenterprises in the region and E_p = the total employees of public offices in the region. Therefore, capacity of each node $= (M_e + E_p) * \frac{\text{population of the zone}}{\text{total population}}$.

We have the knowledge of M_e but we need to find E_p. We observed that there is a huge gap in the actual number of employees in a zone and our estimate of the number of employees taken from the magazine. Therefore, we consider to include the left ones such as public employees as others. We know from INSEE that 33% of the total employees work in public offices. Some of these employees have already been taken into account in hospitals, universities, schools, etc. so that we have already counted one-third of the total employees in them implicitly. We have also counted the employees as the capacity of companies in the previous section that is T_e in (5.2) which we assume to be one-third of the total. So, the remaining one-third is considered among others and hence $E_p = T_e$.

Therefore, we have that the capacity of each node in MIC and others

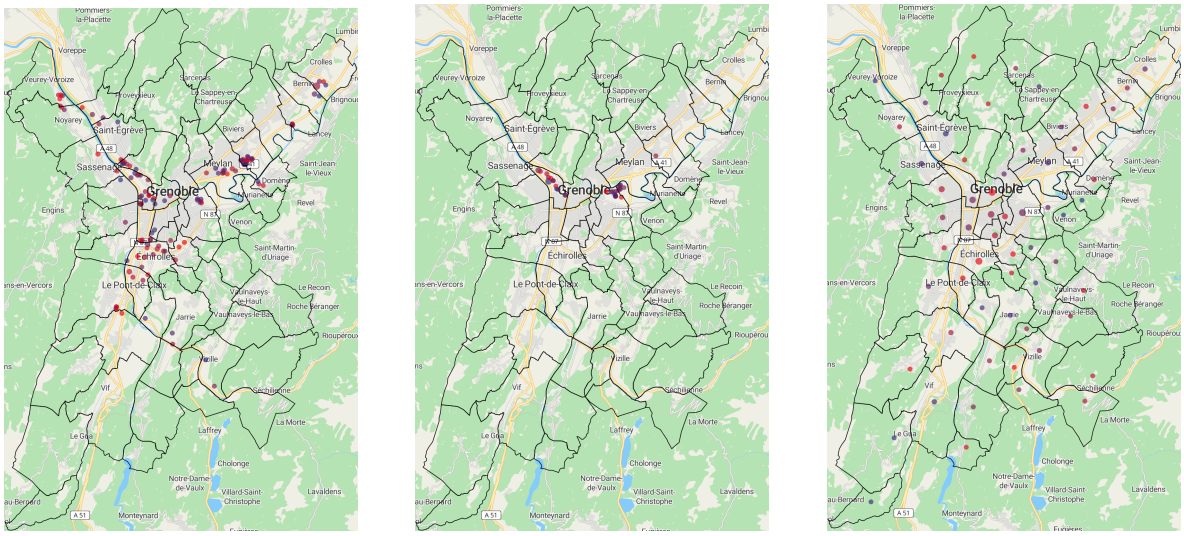
$$\begin{aligned} &= (0.2T_e + T_e)\frac{\text{population of the zone}}{\text{total population}} \\ &= (1.2 * T_e)\frac{\text{population of the zone}}{\text{total population}}. \end{aligned} \tag{5.4}$$

5.1.2.6 Shopping centers

This category is subdivided into three subcategories:

1. Shopping malls
2. Supermarkets and
3. Small shops

The capacity in this category means the maximum number of shoppers allowed in the shop at a given time.



(a) (b) (c)

Figure 5.5: Nodes in the workplaces category. (a) Nodes depicting companies. (b) Nodes depicting research centers. (c) Nodes depicting MIC and others.

Shopping malls In this category, we consider the big shopping malls such as Grand Place, IKEA and Castorama which have more than one floor. The nodes corresponding to these destinations have been located by method 3 as stated in 5.1.2.1.

For the capacity, we follow the national fire regulation [Ter] for maximum number of allowed shoppers in a shop depending on their surface area. From there, we know that on the ground floor, the permissible number of shoppers is 2 person per m^2 in one-third of the surface area, on first floor 1 person per m^2, on second floor 1 person every $2m^2$, in one-third of the surface area. Therefore, the capacity of the shopping mall with three floors available for shopping is given as follows:

$$\text{capacity of shopping malls} = \xi_s * \frac{7}{6} * \text{surface area} \tag{5.5}$$

where ξ_s is a scaling parameter to tune the capacities to a reasonable amount. The parameter ξ_s accounts for the fact that the actual available surface area of a shop also depends on a number of factors such as shopping shelves, walking area, billing area etc. It can also account for the unavailable spaces because of people moving with their shopping trolleys. Here, $\xi_s = \frac{1}{3}$. The surface area for these shopping malls have been taken from their wikipedia pages.

Supermarkets For the supermarkets, we consider the shops with a surface area of more than $300m^2$. Therefore, we consider mainly Carrefour hypermarkets, Carrefour markets, Carrefour Contact, Geant casino, Casino supermarché, Monoprix, Biocoop, Lidl, Intermarché super, Super U and a few others. On the commercial websites of the supermarket groups such as carrefour, they specify how different outlets within their group are classified based on the surface area or location. The classification of supermarkets and their average surface areas

Supermarket name	Average Surface area (m^2)	Capacity ($\xi_s = 1$, fire regulation)	Capacity $\xi_s = 1/3$)
Carrefour Hypermarket	10000	6667	2222
Carrefour Market	2000	1333	444
Carrefour contact	600	400	133
Geant casino	7400	4933	1644
Casino supermarché	1700	1133	378
Monoprix	1800	1200	400
Hyper U	4985	3323	1108
Super U	2016	1344	448
Intermarché Super	2000	1333	444
Lidl	1000	667	222
Biocoop	300	200	67
Supermarkets(national average)	6347	4320	1410

Table 5.3: Surface area and capacities of supermarkets

have been taken from different webpages corresponding to different supermarkets which are Carrefour [Car], Monoprix [Bioa], Intermarché , Lidl [Jou], Biocoop [Biob] and from wikipedia pages.

These supermarkets have been located by method 3 as described in 5.1.2.1. For the capacity, we consider the nation wide rule regarding the permissible number of shoppers according to fire-safety regulations [Ter]. For the sake of simplicity, we assume that the supermarkets and small shops have only one floor for shopping. Since, on the ground floor, the permissible number of shoppers is 2 persons per square meter on one- third of the surface area, the capacities of the supermarkets are given by

$$\text{Capacity of supermarkets or small shops} = \frac{2}{3} * \text{surface area of the shop.} \tag{5.6}$$

similar to the shopping areas, supermarkets and small shops are also never full to this capacity, so, we reduce the capacity by a scaling factor ξ_s. Therefore,

$$\text{Capacity of supermarkets} = \xi_s * \frac{2}{3} * \text{surface area of the shop.} \tag{5.7}$$

Table 5.3 contains the average surface areas and the capacities of different supermarkets. Apart from the shops listed in Table 5.3, we also include Satoriz, Naturalia and Spar markets for which we consider the surface area and capacities equal to that of Carrefour contact as they are of the nature. We also consider some other supermarkets and for that we take the surface area same as the national average from the website [INSc].

Small shops We retrieve their location by filtering the data from openstreetmap using QGIS software as explained in 5.1.2.1. In this category, we consider the shops with average surface

Shop type	Average Surface area (m^2)
Supérettes	216
Commerces d'alimentation générale	57
Commerces de détail de produits surgelés	263
Alimentation spécialisée et artisanat commercial	62
Habillement et chaussures	168
Culture, loisirs, sport	209
Produits pharmaceutiques et articles médicaux et orthopédiques	111
Autres équipements de la personne	79
Équipements de l'information et de la communication	128
Autres commerces de détail	180

Table 5.4: Small shop type and their average surface area as mentioned on [INSc].

area less than $300m^2$. INSEE specifies the shop type and national average of their surface area on [INSc]. These small shops are of the types which have been enlisted in Table 5.4. It can be seen that these shops encompasses almost all types of small shops.

We compute the average of surface area of these shops to obtain a common surface area for all the small shops in Grenoble area and their capacity is computed by (5.7) with $\xi_s = 1/3$. The average surface area is $147m^2$ and hence the average capacity is 33. All the nodes in this subcategory have been given the same capacity.

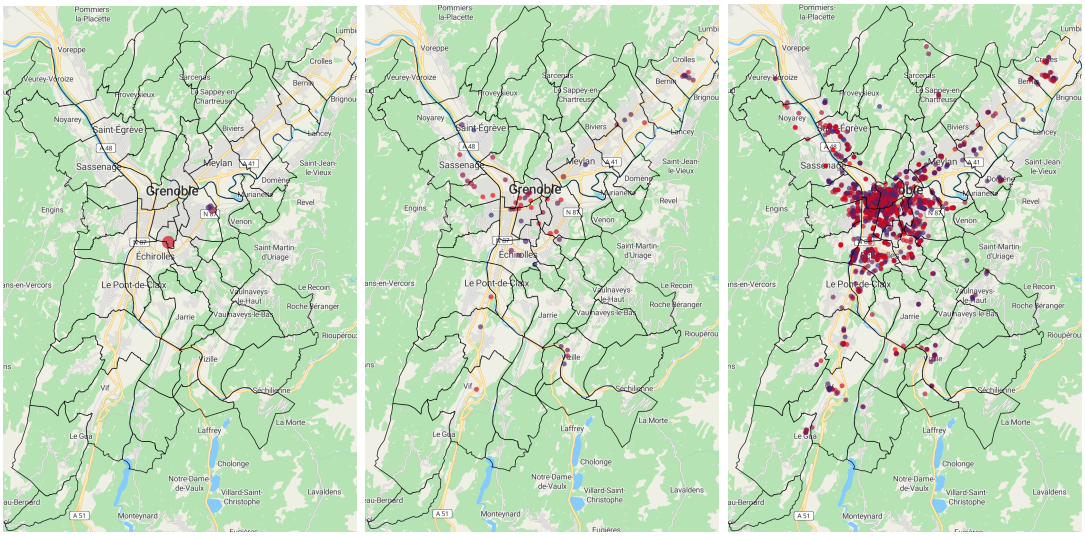
The nodes corresponding to the destinations in this category have been depicted in Figure 5.6.

5.1.2.7 Leisure

The category leisure has four different subcategories which are:

1. Restaurants and bars,
2. Parks,
3. Stadiums and
4. Theaters.

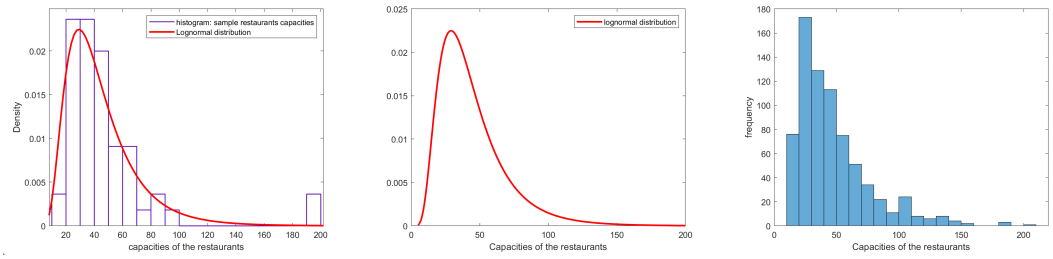
Restaurants and bars: The restaurants and bars have been located by method 1 as described in the 5.1.2.1. We have located around 700 restaurants and bars in the considered zones. Here, the capacity is the total number of seats available in a restaurant. Due to lack of data regarding available seats for all these restaurants, at first we consider the total number of seats in 55 restaurants in Grenoble downtown from the website [Lin]. In our sample of 55



(a) Nodes depicting shopping malls (b) Nodes depicting supermarkets (c) Nodes representing small shops

Figure 5.6: Destination nodes in the shopping category.

restaurants, the mean is 46 and variance is 1320. In order to impute the capacities for the remaining restaurants, at first, we find which probability distribution fits this sample data best using “distributionFitter" app in Matlab as can be seen in Figure 5.7a.



(a) (b) (c)

Figure 5.7: Curve fitting of the 55 sample capacities of restaurants to find the probability distribution which fits them the best. (a) Curve fitting for the total number of seats in the restaurants using ‘distributionFitter’ app in MATLAB. (b) probability density function of the Lognormal distribution which fits our sample data. (c) histogram of the capacities of all the restaurants in our region generated using Lognormal distribution with mean m and variance σ computed using (5.9).

We find that the desired distribution is Lognormal distribution with the following probability density function

$$f_X(x) = \frac{1}{x\chi\sqrt{2\pi}} \exp(-\frac{(\log x - m)^2}{2\chi^2}). \tag{5.8}$$

where m and χ is computed from the mean $mean$ and variance var of the our sample as

follows:

$$m = \log(\frac{mean^2}{\sqrt{var + mean^2}}) \tag{5.9}$$

$$\chi = \sqrt{\log(1 + \frac{var}{mean^2})}. \tag{5.10}$$

For our sample $mean = 46$ and $var = 1320$. The pdf of the distribution which we follow is given in Figure 5.7b. We generate data sets of 800 capacities using Lognormal distribution given by (5.8) and then we distribute these capacities to the restaurants randomly. Figure 5.7c shows the histogram of the capacities generated. We consider the capacities of the restaurants up to 200 only.

Parks: We locate the parks by method 3 as described in 5.1.2.1. We retrieve the surface area of the parks on their respective websites, or the website of Grenoble-Alpes métropole or on the website of Isere tourism. The parks for which we don't have the surface area, we allocate them an average surface area of $50000m^2$. For the capacities of the parks, we consider the following thumb rule:

$$\text{Capacity of a park} = \frac{\text{total surface area of the park}}{15}. \tag{5.11}$$

Here, we assume that in general, in a not very crowded situation, there can be one person in an area of $15m^2$. In case of any event, this thumb rule won't be valid any more because there would be many persons in a small area of the parks. However, in general, this capacity is not filled up and people go there either just to walk or run. Therefore, we scale the above mentioned capacity by ξ_p to have a reasonable number of daily visitors. Here, we take $\xi_p = 0.02$, therefore, we have that

$$\text{Capacity of a park} = 0.02 * \frac{\text{total surface area of the park}}{15}. \tag{5.12}$$

Stadiums and theaters: They nodes in these two subcategories have been located using method 3 as described in 5.1.2.1. The capacities of the stadiums and the theaters are the total number of available seats. This information is taken from their respective websites or from the website of Grenoble métropole [Gmb] or the booking websites for the theaters.

The destination nodes of the leisure category can be seen in Figure 5.8.

5.1.2.8 Aggregation of nodes

In some subcategories such as restaurants and small shops, the number of nodes is very large. So, in order to reduce the complexity in the mobility model to be implemented, we aggregate the destinations nodes in such subcategories. This aggregation is done per zone by replacing all the nodes belonging to a subcategory in a zone by a single node. The location of such

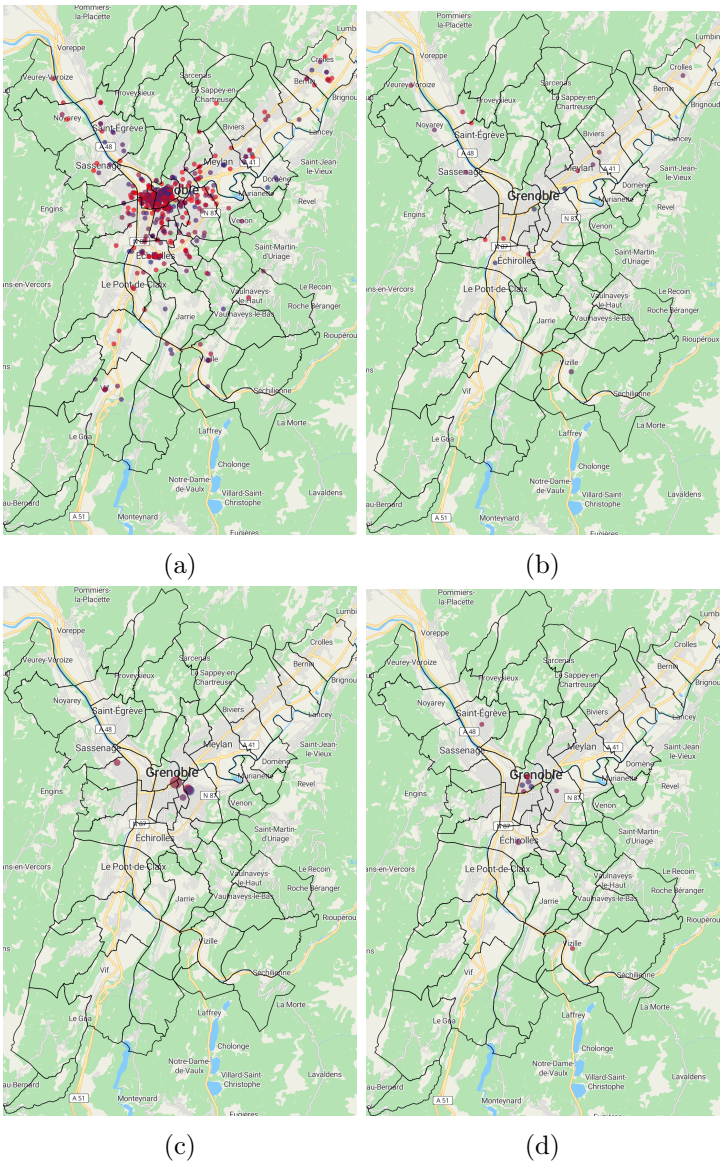


Figure 5.8: Destination nodes in the leisure category. (a) Nodes representing restaurants. (b) Nodes depicting parks. (c) Nodes representing stadiums. (d) Nodes depicting theaters.

aggregated node is the barycenter of the locations of all the nodes of a subcategory in a zone and the capacity of such a node is the sum of the capacities of all the nodes of the same destination category in that zone. The destination subcategories for which this aggregation has been done are primary schools, supermarkets, small shops, companies, research centers and restaurants. Figure 5.9a depicts how the the aggregation of nodes belonging to the same subcategory is done per zone and Figure 5.9b shows all the nodes in the region where as Figure 5.9c depicts the aggregated nodes.

5.1.3 Mobility network

Each origin/destination node i is connected to at least one destination/origin node j. For each subcategory, we have different rules to decide if the pair $(i, j) \in \mathcal{E}$, where $i \in \mathcal{V}_o$ and $j \in \mathcal{V}_d$. These rules have been listed in Table 5.5.

Subcategory	**Rules for connecting origin and destination nodes**
Primary schools	connected to the origin node of the same zone.
Middle schools	connected to the corresponding govt. defined college sector.[DIb]
High schools	govt. defined rules based on one's address [AGa][DIa]
Universities/ Hospitals	connected to every origin node
Workplaces/ Shopping/ Leisure	Attraction based rules to be defined in Section 5.1.3.1.

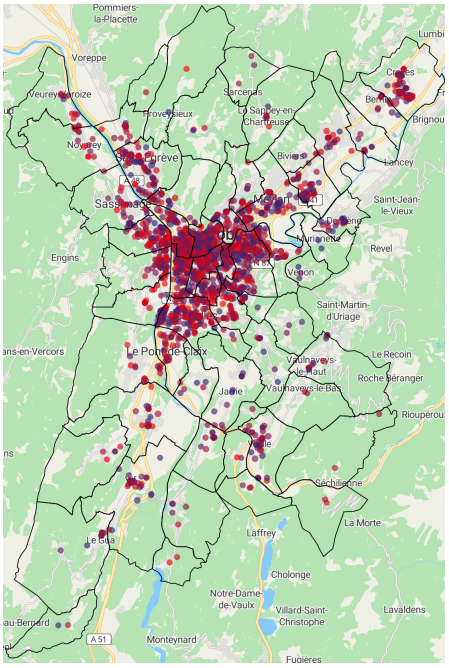
Table 5.5: OD connection rules for different subcategories

5.1.3.1 Attraction based rules

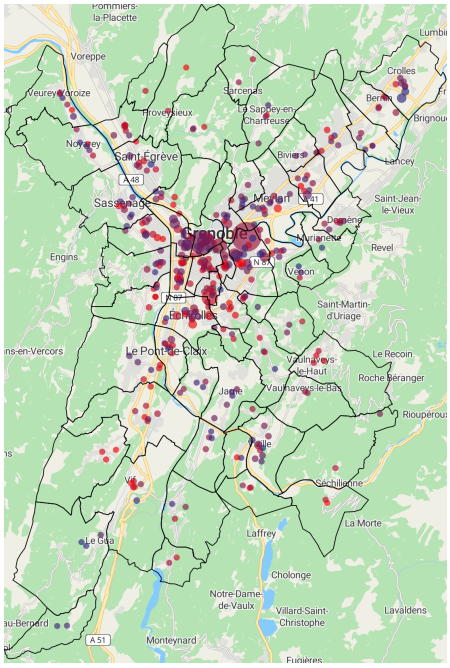
Let d_{ij} represent the road distance between origin i and destination j and σ_j, the maximum distance that ν% of people of i travel 'daily' to visit j. Here, we have assumed $\nu = 95\%$ which means that the rest 5% of the people in i choose to travel a distance more than σ_j to visit the destination j. Given the location of origins and destinations, we compute the minimum real minimum road distance d_{ij} between all possible $i \in \mathcal{V}_o$ and $j \in \mathcal{V}_d$ using OSMnx python library [Boe17]. It has been estimated that the number of individuals per unit area who travel a distance from home is inversely proportional to square of the distance travelled [Sch+21]. Inspired by the gravity models [Bar+18], we compute Q_{ij}, the attraction between i and j as

$$Q_{ij} = P_i C_j e^{-|\ln(1-\nu)|\left(\frac{d_{ij}}{\sigma_j}\right)^2}.$$

(a)



(b)



(c)

Figure 5.9: Aggregation of nodes in the subcategories: primary schools, supermarkets, small shops, companies, research centers and restaurants. The capacities and location of the aggregation nodes are the sum of the capacities of the non-aggregated nodes the barycenter of their locations respectively. (a) Aggregation of nodes belonging to the same subcategory in a zone. (b) Non- aggregated destination nodes in our region. (c) The aggregated destination nodes.

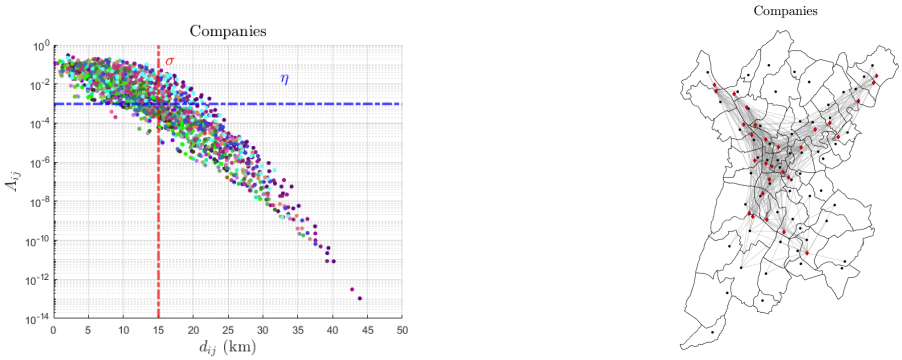
For each destination j, normalize the attraction Q_{ij}

$$A_{ij} = \frac{Q_{ij}}{\sum_h Q_{hj}} = \frac{P_i e^{-|\ln(1-\nu)|\left(\frac{d_{ij}}{\sigma_j}\right)^2}}{\sum_h P_h e^{-|\ln(1-\nu)|\left(\frac{d_{hj}}{\sigma_j}\right)^2}}.$$

where the dependence of C_j cancels out from numerator and denominator. Then, the OD matrix $\mathcal{O}_{ij}$ is computed as

$$\mathcal{O}_{ij} = \begin{cases} 1 & \text{if } d_{ij} \leq \sigma_j \ \& \ A_{ij} \geq \eta_j \\ 0 & \text{otherwise} \end{cases}$$

Note that, here σ_j is a threshold on the maximum distance traveled by the most people 'daily' to go from an origin to destination j and η_j is a threshold on the attraction between i and j. These thresholds are different for destinations in different subcategories and they reflect the fact that a person prefers to go to a nearer destination if available. For example, in general, we visit the shops which are close to our residences. Figure 5.10 depicts how the thresholds σ and η_j are applied to compute the OD connections. In Figure 5.10a, we consider the upper-left quadrant after applying the thresholds.



(a) (b)

Figure 5.10: Computation of OD connection matrix using attraction law. (a) plot of Attraction vs distance for companies. (b) OD connections for companies with $\sigma = 15km$.

Based on the rules for connecting two locations corresponding to each subcategory, according to Table 5.5 and choosing thresholds for each subcategory, we have the connections as depicted in Figures 5.11-5.15.

Finally, we have the mobility network of Grenoble area as depicted in Figure 5.16. It can be seen that the city center has the highest connections which is a fact as most of the destinations are situated in the city center and it attracts a large number of people daily.

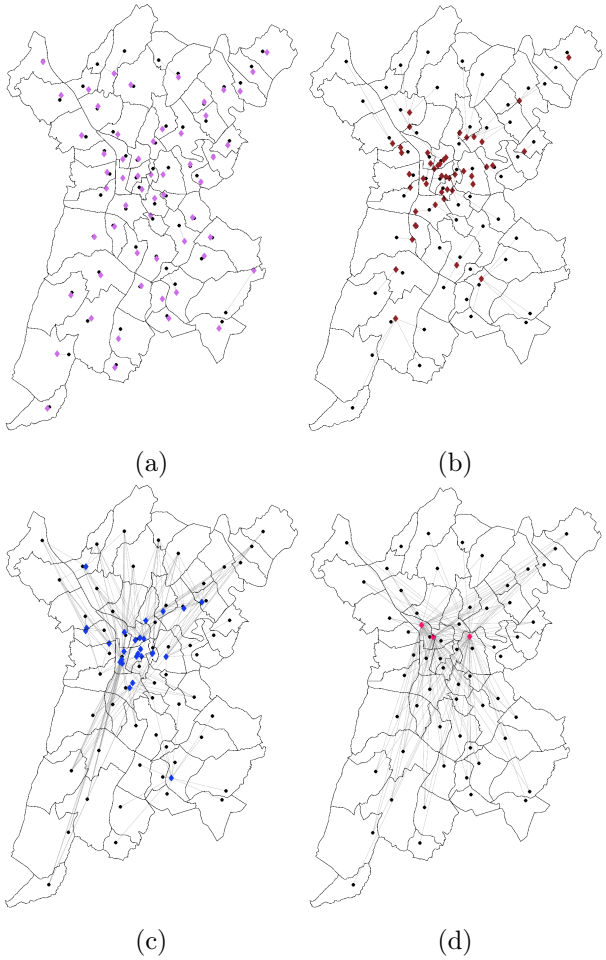
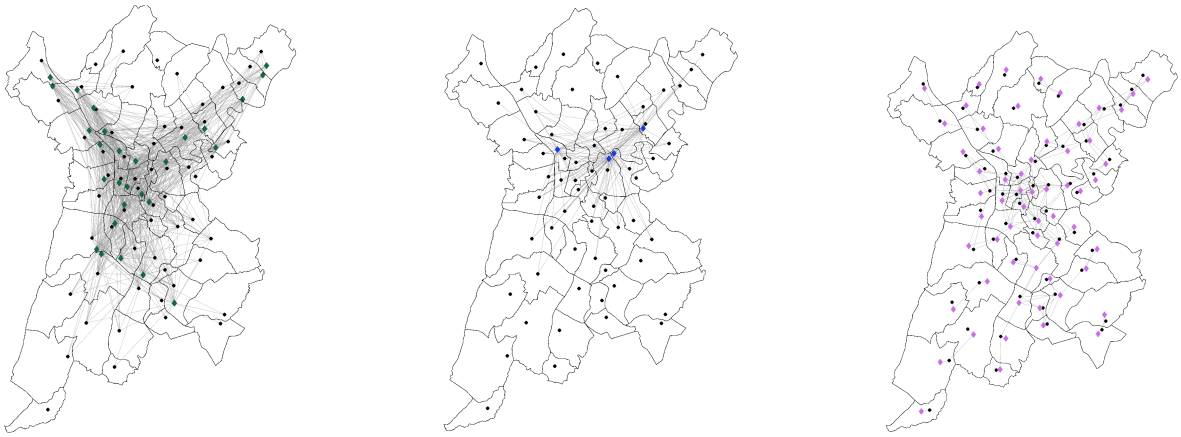
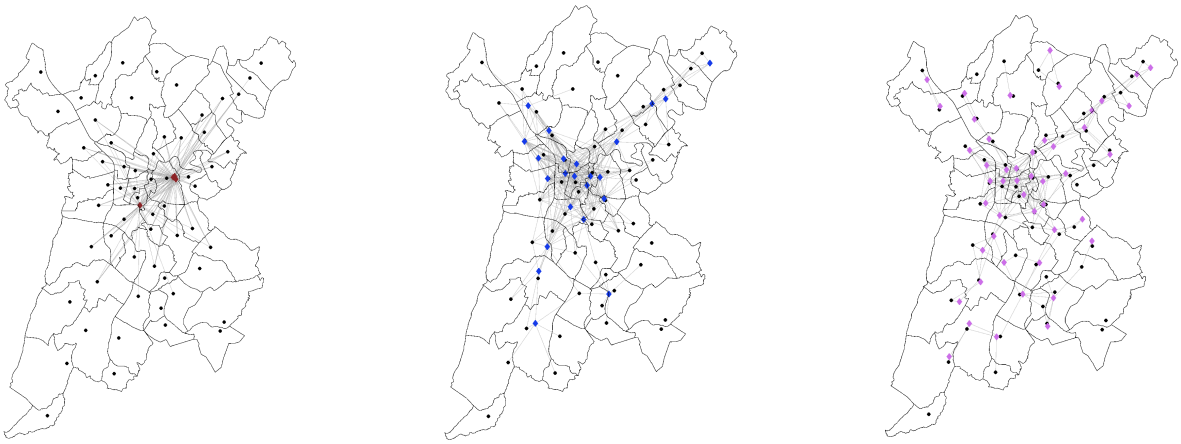


Figure 5.11: Connections for schools based on the government rules. The black square nodes represent the origins and the colored diamond shaped nodes represent the destinations in the corresponding subcategory. (a) OD connections for primary schools. (b) OD connections for middle schools. (c) OD connections for high schools. (d) OD connections for universities.



(a) (b) (c)

Figure 5.12: OD connections for the destinations in workplaces category using the attraction based laws. (a) OD connections for the companies. Here $\eta_j = 0.001$ and $\sigma = 15km$. (b) OD connections for the research centers. Here $\eta_j = 0.001$ and $\sigma = 15km$. (c) OD connections for the MICs. Here $\eta_j = 0.001$ and $\sigma = 5km$.



(a) (b) (c)

Figure 5.13: OD Connections for destinations in shopping category using attraction based laws. The black square nodes represent the origins and the colored diamond shaped nodes represent the destinations in the corresponding subcategory. (a) OD connections for the shopping malls. Here $\eta_j = 0.001$ and $\sigma = 15km$. (b) OD connections for the supermarkets. Here $\eta_j = 0.001$ and $\sigma = 8km$. (c) OD connections for the small shops. Here $\eta_j = 0.001$ and $\sigma = 5km$.

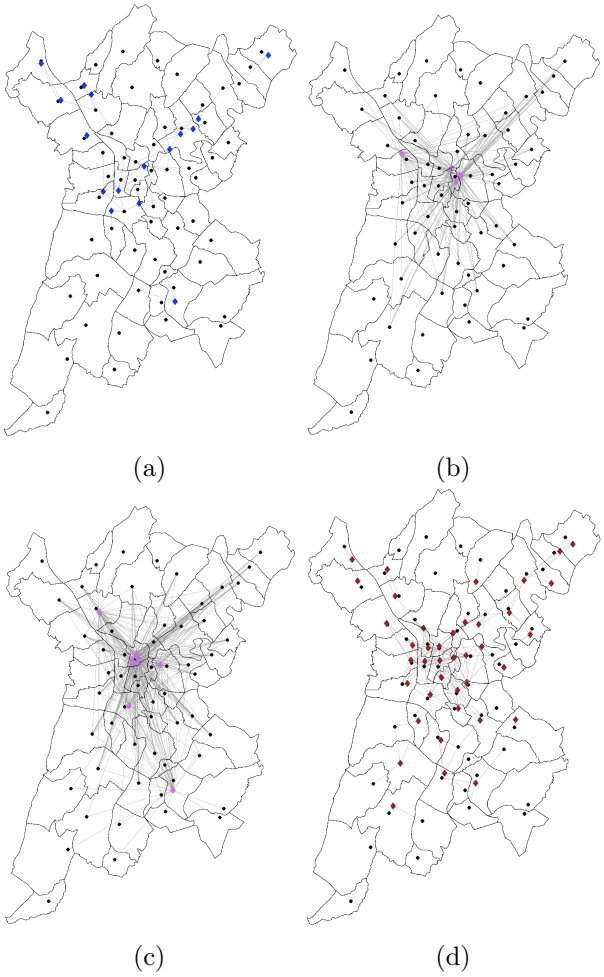


Figure 5.14: Connections for the destinations in the leisure category. The black square nodes represent the origins and the colored diamond shaped nodes represent the destinations in the corresponding subcategory. (a) OD connections for the parks. Here $\eta_j = 0.001$ and $\sigma = 5km$. (b) OD connections for the stadiums. Here $\eta_j = 0.001$ and $\sigma = 20km$. (c) OD connections for the theaters. Here $\eta_j = 0.001$ and $\sigma = 10km$. (d) OD connections for the restaurants. Here $\eta_j = 0.001$ and $\sigma = 5km$.

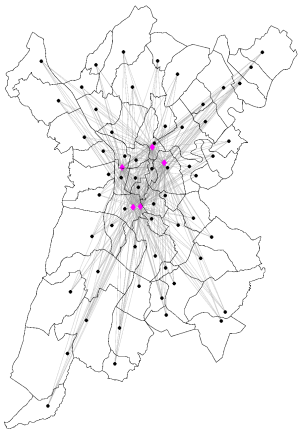


Figure 5.15: Connections for hospitals. The black square nodes represent the origins and the pink diamond shaped nodes represent the hospitals. Hospitals are connected with all the origins.

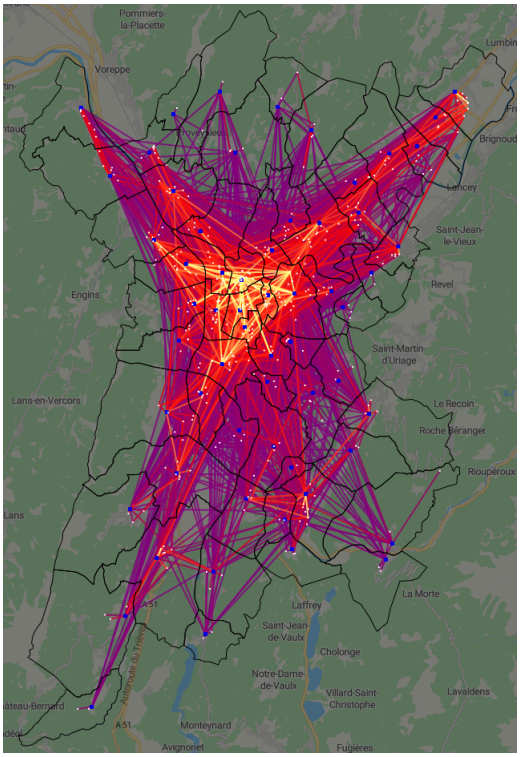


Figure 5.16: Mobility network in Grenoble area. The destinations are denoted by yellow nodes and origins by black. The nodes with higher connections are brighter than the ones which have low connections. This figure has been taken from the demonstrator which will be described in Chapter 7.

5.2 Time dependent profiles

In this section, we find the time schedules for different subcategories. In particular, we find the destination schedule, average time spent, mobility windows and Demand Gating Profile (DGP) for each subcategory.

5.2.1 Destination schedule, mobility window and average time spent

Recall that the destination schedule of a destination $j \in \mathcal{V}_d$ is the daily time interval $[a_j, , b_j)$, where, $0 \leq a_j < b_j \leq 24$ during which j is open. Here, a_j and b_j are the opening time and closing time of the destination $j \in \mathcal{V}_d$ respectively. Mobility window is defined as the time interval $[t_{ij}, t_{ij} + \tau_{ij}]$, during which there is mobility from i to j. Here, t_{ij} is the time when the mobility starts from i to j and τ_{ij} is the duration for which mobility keeps happening between i and j. For the sake of simplicity, here we consider that $\tau_{ij} = \tau_{ji}$. It means that the mobility window for the mobility from an origin i to a destination j is equal to the return mobility window from j to i. Average time spent s_j is the duration for which a person stays in the destination j on an average. We set these parameters for each subcategory. Therefore, we will follow the notations a_g, b_g, s_g for opening time, closing time and the average time spent of the destination subcategory g, where $g \in \{1, \ldots, 15\}$ and the mobility window from the origins $i \in \mathcal{V}_o$ to the destination subcategory g will be denoted by $[t_{ig}, t_{ig} + \tau_{ig}]$. The mobility window for the return from the destinations to the origins, will be denoted as $[t_{gi}, t_{gi} + \tau_{ig}]$. Table 5.6 contains different time-schedules parameters for different subcategories. The opening time, closing time and the average time spent for primary schools, middle schools and high schools has been taken from [RFb], [Sec] and [Lyc] and that for working places from [Off]. For shopping and leisure, we referred to the timings of some destinations on the web or OSM maps and set that for all the other destinations in that subcategory. On the weekends, schools and workplaces are closed. Note that we have different time schedules for each day for the implementation to consider different time schedules for each day. This is important since destinations in some subcategories, such as stadiums, are assumed to open only on the evenings of each Friday and closed otherwise. Also, for example, primary schools have different closing times on specific days of the week, and restaurants and bars are open till very late on weekends. Moreover, it has been mentioned earlier that hospitals have different types of capacities. Hence there are also different purposes for visiting hospitals. These persons can be: employees denoted by E_h, new hospitalized patients H_h, patients going for consultations H_c, and visitors V_h. We also assume a constant population is present in the hospitals during nighttime. This includes the occupied number of beds and the proportion of employees who do night shifts. We compute the occupied number of beds B_h by the following formula:

$$B_h = H_h \times \text{average time of hospitalization}$$

where H_h is the average daily number of newly hospitalized patients. Let us assume that the proportion of employees who do night shifts is 10%, then the constant population present during the night time is given by $B_h + 0.1E_h$. The average time spent at hospitals by a person

Subcategory	$\mathbf{a_g}$	$\mathbf{b_g}$	$\mathbf{s_g}$	$\mathbf{t_{ig}}$	$\tau_{\mathbf{ig}} = \tau_{\mathbf{gi}}$	$\mathbf{t_{gi}}$
Primary Schools	7h30	16h30	$(b_g - a_g)$	7h30	1h	16h
Middle schools		18h				17h
High schools		18h30				17h
Companies	8h	18h	$(b_g - a_g)$	8h	2h	16h30
Research centers						
MIC and others						
Hospitals	7h	20h	using (5.13) with $s_E = 8, s_C = 1$ and $s_V = 1$	7h	$(b_g - a_g)$	$t_{ig} + s_g$
Shopping areas	10h	20h	2h	10h	$(b_g - a_g)$	$t_{ig} + s_g$
Supermarkets	7h 30	23h	$\frac{3}{4}h$	7h 30	$(b_g - a_g)$	$t_{ig} + s_g$
Small shops	9h	19h	$\frac{1}{3}h$	9h	$(b_g - a_g)$	$t_{ig} + s_g$
Restaurants and bars	8h	22h	$2h$	8h	$(b_g - a_g)$	$t_{ig} + s_g$
Parks	0h	24h			$(b_g - a_g)$	$t_{ig} + s_g$
Theaters	14h	23h	3h	14h	$(b_g - a_g)$	$t_{ig} + s_g$
Stadiums						

Table 5.6: Time schedules for different subcategories on a typical week-day. a_g, b_g and s_g are the opening time, closing time and average time spent for destination in subcategory g. t_{ig} is when mobility from i to destination in subcategory g starts. $\tau_{ig} = \tau_{gi}$ is the size of mobility window between i and destination in subcategory g.

is computed as

$$s_h = \frac{E_h s_E + H_c s_C + V_h s_V}{E_h + H_c + V_h} \tag{5.13}$$

where s_E, s_C, s_V denote the employees work duration, average duration of consultation, and average duration of visitors, respectively. For destinations in other categories also, there are people with different purposes but the majority of the persons have a single purpose. For example, in shops there are employees and shoppers but the number of shoppers is very high in comparison to the the number of employees so the average time spent is mainly affected by the shopper so we didn't take the average time spent by the employees into account. On the other hand, all four types of people coming to the hospitals are significant in number. Now that we have the destination schedule we can easily compute the Supply Gating Functions by (4.9). In the next subsection, we will find the Demand Gating Profiles for different destination subcategories.

5.2.2 Demand Gating Profile (DGP)

Recall that DGP denoted by $\delta_{ig}(t)$ is a function such that $\int_0^{24} \delta_{ig}(t)dt = 1$. It captures the mobility pattern from i to g. In general, we have two types of destinations.

1. One in which mobility happens during a window when people come to the destinations

from the origins in the morning and stay there for a long period of time and the mobility in the opposite direction starts at a specified time in the evening. This is the case with all the destinations in school category, workplaces category and stadiums. In this case, the DGP is defined as

$$\delta_{ig}(t) = \begin{cases} \frac{1}{\tau_{ig}} & t \bmod 24 \in [t_{ig}, t_{ig} + \tau_{ig}) \\ 0 & \text{otherwise} \end{cases}$$

and

$$\delta_{gi}(t) = \begin{cases} \frac{1}{\tau_{ig}} & t \bmod 24 \in [t_{gi}, t_{gi} + \tau_{gi}) \\ 0 & \text{otherwise} \end{cases}$$

Notice that, if we have different destination schedules for different days of the week, then we will have different profiles for different days of the week. The destinations in school, workplaces category and stadium subcategory have this profile as shown in Figure 5.17. Recall that the stadiums open only on a friday night, the mobility only on that day and follows the profile as similar to schools and workplaces.

2. The other kind is the one in which mobility keeps happening throughout the day. People keep coming here, stay for some time and then return and it keeps happening till the destination is closed. All the destinations of the categories hospitals, shopping and leisure are of this type. In this case, if we find the demand gating profile $\delta_{ig}(t)$ corresponding to the mobility from origins to the destinations then the profile for return mobility is given by $\delta_{gi}(t) = \delta_{ig}(t - s_g)$, where s_g is the average time spent per category. For the hospitals, we have

$$\delta_{ig}(t) = \begin{cases} \frac{1}{\tau_{ig}} & t \bmod 24 \in [t_{ig}, t_{ig} + \tau_{ig}) \\ 0 & \text{otherwise} \end{cases}$$

For the shopping and leisure category, we inferred this mobility pattern $\delta_{ig}(t)$ from the household travel survey of Grenoble region done by INSEE in 2010 [Emd]. The table 'ED20A' in the file EMD'10 contains the number of people traveling with different motives on a given hour of a day. Since, the data points are given for each hour and in absolute numbers, we obtain the required DGP for destinations in these two categories by interpolation and normalising the values by the integral of the curve. For each of the destinations, we compute these profiles for each day of the week. This information can also be retrieved by observing the popularity trends of destinations in each subcategory on google or OSM. In particular, the profiles on a week day is different from the ones on a weekend reflecting the high popularity of destinations in leisure category. Figure 5.17 depicts the different types of DGPs we have for destinations in different categories on a week day.

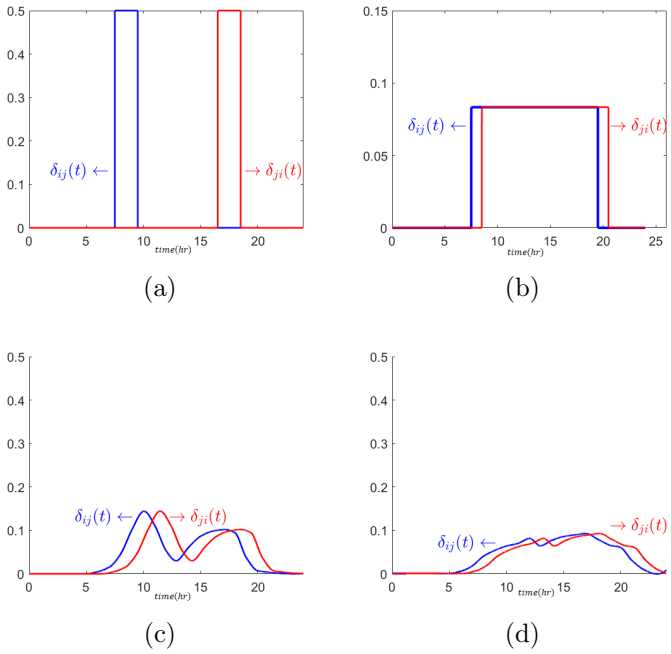


Figure 5.17: DGP on a weekday for different destination categories. (a) DGP for the categories schools and workplaces and stadiums in leisure (b) DGP for hospitals. (c) DGP for shopping. (d) DGP for leisure.

5.3 Number of people

In this section, we will find how many people travel from i to j if they are connected.

5.3.1 Daily capacity of the destinations

In Section 5.1.2, we found the instantaneous capacities of different destinations denoted as C_j. Now, the daily capacity of a destination can be defined as the total number of people who can visit there. Let us denote the daily capacity of j by C_j^d which can be computed as

$$C_j^d = C_j \frac{(b_j - a_j)}{s_j}.$$

Notice that, we have $\frac{(b_j - a_j)}{s_j} \geq 1, \ \forall j \in \mathcal{V}_d$. Therefore, the destinations such as schools where people stay for the whole day, the daily capacity C_j^d is equal to the instantaneous nominal capacity C_j. On the other hand, for destinations such as shops where the mobility is continuously happening, in that case $C_j^d >> C_j$.

5.3.2 Average daily number of people from origins to destinations

If for an origin-destination pair $(i,j) \in \mathcal{E}$, $\mathcal{O}_{ij} = 1$ as computed in Section 5.1.3 for the subcategory $\mathcal{D}_c^a$, then the average daily number of people traveling from i to $j \in \mathcal{D}_c^a$ is denoted by M_{ij} and computed per subcategory $\mathcal{D}_c^a$ as

$$M_{ij} = \min\{X_1, X_2\}, \tag{5.14}$$

where

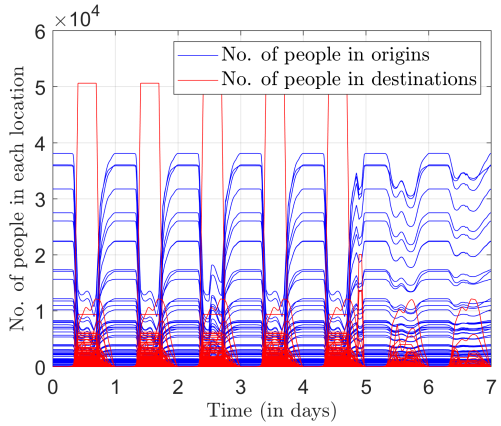
$$X_1 = P_{ic} * f_c^p * \frac{C_j^d}{\sum\limits_{\substack{k \in \mathcal{D}_c^a, \\ k \in \mathcal{N}_i}} C_k^d} \quad \text{and} \quad X_2 = \frac{P_{ic}}{\sum_{h \in \mathcal{N}_j} P_{hc}} * C_j^d * \frac{1}{\delta_{ic}^* \tau_{ic}}.$$

Here, P_{ic} is the population of origin i which is eligible to go to destination subcategory $\mathcal{D}_c^a$, and f_c^p is the fraction of the eligible population of origin i which, in general, goes to $\mathcal{D}_c^a$ (this accounts for the fact that not all the eligible population goes every day to destination c). Also, C_j^d is the daily capacity of destination j, τ_{ic} is size of the mobility window, and δ_{ic}^* is the maximum of the gating function $\delta_{ic}(t)$. This definition of M_{ij} gives daily demands that are consistent with the population sizes and capacities. First, it takes into account the fact that in normal operating situation (i.e., with nominal capacities, without restrictions), the demand from i to j usually does not exceed population size of i and does not exceed capacity of j. Second, it reflects the fact that largest demands are from largest origins towards largest destination (as in the attraction law). The first term of (5.14), X_1, considers the population that is typically going from origin i to category c (this has size $f_c^p P_{ic}$) and then splits it among the destinations j that belong to category c and are neighbors of i, proportionally to their daily capacities (representing their sizes). The second term, X_2, instead, considers the daily capacity of a destination j, weighted by a factor $1/(\tau_{ic}\delta_{ic}^*)$, and splits it among origins i that are neighbors of j, proportionally to their eligible populations. The weighting factor $1/(\tau_{ic}\delta_{ic}^*)$ is equal to 1 when the gating function $\delta_{ic}(t)$ has a rectangular shape, i.e., when $\delta_{ic}(t)$ is constantly equal to $1/\tau_{ic}$ over an interval of size τ_{ic} and zero elsewhere; for other shapes, this correction terms accounts for the fact that the computation of daily capacity assumes that the destination is filled at full capacity over all its opening time, while the demand profile δ is such that the demand reaches full capacity only at its peak δ^*, and is lower at other times.

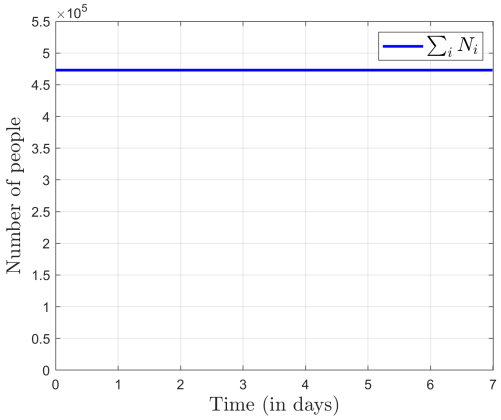
Notice that the model assumes that people travel from origin i to destination j and then returns back from j to i on the same day. This is expressed by defining $M_{ji} = M_{ij}$.

5.4 Simulation Example

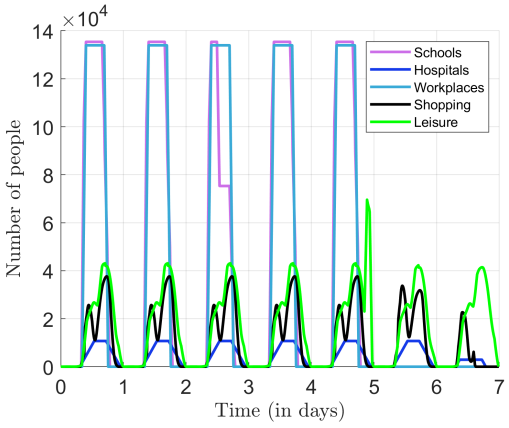
Now that we have defined the model and calibrated the parameters for mobility model, in this section, we show some simulation results for the mobility model (4.12)-(4.18) using the parameters computed in this chapter for the large-scale network of Grenoble. The mobility parameters DGP, SGF, M_{ij}, a_j, b_j for each destination categories are the same as defined in this chapter and the mobility happens along the edges the of the mobility network of Grenoble



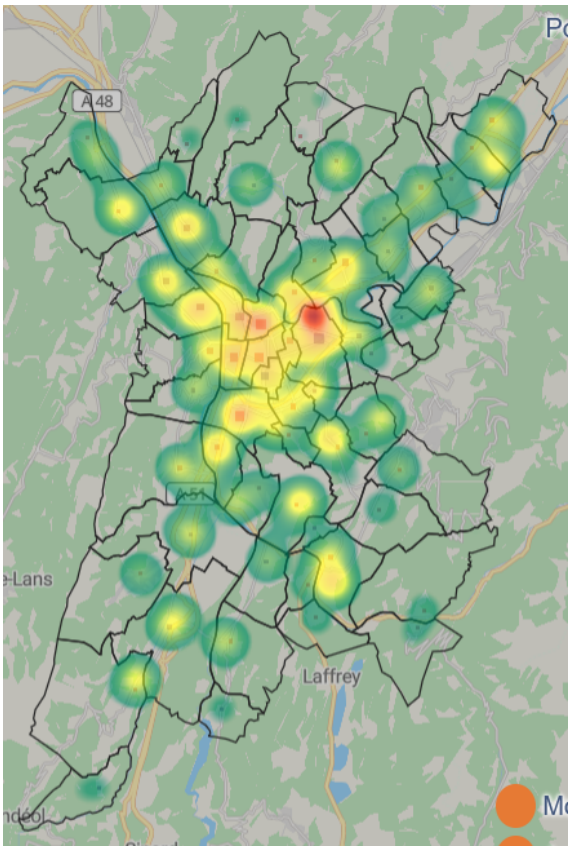
(a)



(b)



(c)



(d)

Figure 5.18: (a) Number of people $N_i(k)$ of people in different origins and destinations in Grenoble area during different times of a week. (b) Total number of people in the network during the week. (c) Number of people in different destination categories during different times of a week. (d) Heatmap showing distribution of people On Monday at 11:30 in the morning. This heatmap is a screenshot from GTL-Healthmob simulation platform described in Chapter 7.

depicted in 5.16. The initial conditions are

$$N_i(0) = \begin{cases} P_i \text{ if } i \in \mathcal{V}_o \\ 0 \text{ if } i \in \mathcal{V}_d \end{cases}$$

and we run the simulation for a week. Figure 5.18a shows the number of people at different locations during the week. It can be seen that when the mobility starts, the people in origins start to move to destinations and stay there for some time depending on the destination category during the day time and return in the evening. Moreover, the total number of people in the network remains constant throughout the week, as shown in Figure 5.18b. It can also be noticed that indeed during the weekend, there are no people in some destinations which remain closed. Figure 5.18c shows the number of people in different categories. Notice that people go mostly to work or for educational purposes, which is natural on a weekday, and there is no mobility during the weekend in these two categories. Also, notice that people go to these destinations and stay at their workplaces or schools, or universities during work hours and return to their residences in the evening. Also, notice that mobility in leisure and shopping destinations keeps happening throughout the day and is higher in the afternoon than in the morning, which is natural as people like to go shopping or for leisure activities after they return from work or school. Moreover, we can see a spike in the number of people visiting destinations in the leisure category on a Friday night as the stadiums open on Fridays. Also notice very less mobility on a Sunday in shopping category as most of the shops are closed. In the hospital also, the mobility keeps happening throughout the day on a weekdat, and there is less mobility on a Sunday as they remain closed for daily consultations. Figure 5.18d is a heatmap depicting the distribution of people on a Monday at $11:30$ am. Notice that at this time, most people are in zones near Grenoble as most of the destinations are located in that and moreover have the most connections as shown in Figure 5.16.

5.5 Concluding remarks

In this chapter, we computed the parameters required for implementing the mobility model (4.12)-(4.18) in the Grenoble area and built a mobility network. In particular, as they are the building blocks of a mobility network, we first identified origins and destinations and classified them into categories and further into subcategories. This classification is necessary for applying control and identifying different parameters common to destinations of the same kind. We collected the data regarding the capacities of destinations. Moreover, we proposed attraction-based laws to decide if two locations are connected based on the minimum road distance between them. We used them to establish network connections for some destinations in some subcategories. In addition to this law, we also use government norms to establish OD connections in some subcategories. Using these techniques, we built the mobility network of the Grenoble area. Moreover, we collected the data to compute the time-dependent profiles needed in the model. We proposed a method to compute the average daily flow between origins and destinations in different subcategories. Finally, we gave a simulation example to showcase the implementation of the model to Grenoble area.

Chapter 6

Optimization of human mobility for epidemic mitigation

Contents

6.1 Urban human mobility and epidemic model 94
6.1.1 Discrete-time mobility model 94
6.1.2 SIR-mobility model 96
6.1.3 Epidemic scenario in Grenoble area 96
6.2 Optimal capacity control for epidemic mitigation 97
6.2.1 Socio-economic activity 98
6.2.2 Optimal capacity control problem 99
6.2.3 Benchmark control policy 100
6.3 Category-free policies 100
6.3.1 Category-free fixed policy 101
6.3.2 Category-free receding-horizon policy 103
6.3.3 Simulations 104
6.4 Category-dependent policies 106
6.4.1 Monotonic optimization 106
6.4.2 Polyblock outer approximation algorithm 108
6.4.3 Implementation issues and Enhancements 113
6.4.3.1 Anti-jamming 113
6.4.3.2 Removal of suboptimal vertices from $\mathcal{T}_{k+1}$ 114
6.4.4 Category-dependent fixed policy 115
6.4.5 Category-dependent receding-horizon policy 119
6.4.6 Simulations 119
6.5 Concluding remarks 122

This chapter addresses the problem of controlling human mobility in order to mitigate an epidemic in a city. We consider the discrete-time SIR-mobility model presented in Chapter 4 For this city-wide model, we provide techniques to compute optimal mobility control policies, that tune the operating capacities of the destinations depending on the current epidemic status, so as to maximize the socio-economic activity while keeping the total infections below a desired threshold. We apply these techniques to the mobility network of Grenoble metropolis developed in Chapter 5.

6.1 Urban human mobility and epidemic model

Consider the human mobility in an urban area between two types of locations: origins and destinations as stated in Chapter 4. Recall that the population of origin $i \in \mathcal{V}_o$ is denoted by P_i and C_i denotes the nominal capacity of the destination i for $i \in \mathcal{V}_d$. Also recall that the model in chapter 4 has the possibility to describe restrictions on capacities of destination, such as those that were imposed by many governments during peaks of Covid-19 pandemic. Such capacity reduction is decided per category of destinations, and is described by a capacity control $u_h(k)$. In order to have a control for capacity restriction which can be updated on a regular basis, we can have a piece-wise constant $u_h(t)$ instead of a control input which is constant for the total time horizon. Let T be the total time horizon of interest and let T_u be the duration after which a policy affecting the capacity of destinations in different categories is updated (the update period). Let $p = \lceil T/T_u \rceil$, where $\lceil \cdot \rceil$ is the ceiling function, then the piece wise constant control $u_h(k)$, can be defined as

$$u_h(k) = \begin{cases} \mu_h^1 & \text{if } \ k\Delta t \in [0, T_u] \\ \mu_h^2 & \text{if } \ k\Delta t \in (T_u, 2T_u] \\ \vdots & \vdots \\ \mu_h^p & \text{if } \ k\Delta t \in ((p-1)T_u, T] \end{cases} \tag{6.1}$$

with $\mu_h^\ell \in [0, 1]$ for every $\ell \in \{1, \ldots, p\}$. $u_h(k)$ determines the allowed operating capacity of the destinations in $\mathcal{D}_h$ at time step k in terms of the proportion of nominal capacity at time $k\Delta t$ where Δt is the time step size. The operating capacity is the allowed capacity of a destination at a given time denoted as C_j^o in the destinations of category h, where

$$C_j^o(k) = C_j u_h(k), \quad \text{for } j \in \mathcal{D}_h. \tag{6.2}$$

In (6.2), if for a destination $j \in \mathcal{D}_h$, $u_h(k) = 0$, then the operating capacity $C_j^o(k) = 0$, which means that no person is allowed to visit j. On the other hand, if $u_h(k) = 1$, then $C_j^o(k) = C_j$ that is the operating capacity of j is equal to its nominal capacity. For the sake of simplicity, here onward, we will use the notation $Z_j(k)$ for the operating capacity $C_j^o(k)$ if j is a destination or the population P_j if j is an origin unless otherwise stated. Note that, if j is an origin, $Z_j(k)$ will be constant and equal to the population P_j, however, if j is a destination then $Z_j(k)$ will be equal to the operating capacity $C_j^o(k)$ which varies according to the coefficient $u_h(k)$ if $h \in \mathcal{D}_h$ as defined in (6.2).

Now, we recall the discrete-time mobility and SIR-mobility model described in Chapter 4 to keep them handy.

6.1.1 Discrete-time mobility model

Recall from Chapter 4 that $N_i(k)$ is the number of people at location $i \in \mathcal{V}_o \cup \mathcal{V}_d$ at time step k and Δt, the time-step size, then, the discrete-time mobility model is

$$N_i(k+1) = N_i(k) + \Delta t \sum_{j \in \mathcal{N}_i} (\phi_{ji}(k) - \phi_{ij}(k)), \tag{6.3}$$

where $\mathcal{N}_i = \{j : (i,j) \in \mathcal{E}\}$ is the set of i's neighbors and the flow $\phi_{ij}(k)$ is the flow from i to j defined as

$$\phi_{ij}(k) = \min(\Delta_{ij}(k), \Psi_j(k)), \tag{6.4}$$

where $\Delta_{ij}(k)$ is the demand of i with respect to j which is defined as

$$\Delta_{ij}(k) = \min\left(\delta_{ij}(k) f_{ij}(k), \frac{N_i(k)\alpha_{ij}(k)}{\Delta t}\right), \tag{6.5}$$

where $f_{ij}(k) = M_{ij}u_h(k)$ with M_{ij} being the total number of people that would like to visit j from i daily and $\delta_{ij}(k)$ is the Demand Gating Profile (DGP), such that

$$\begin{cases} \delta_{ij}(k) > 0 & \text{if } k\Delta t \bmod 24 \in [t_{ij}, t_{ij} + \tau_{ij}) \\ \delta_{ij}(k) = 0 & \text{otherwise} \end{cases}$$

satisfying $\sum_{k\Delta t \in [0,24]} \delta_{ij}(k) = 1$. Here, $[t_{ij}, t_{ij} + \tau_{ij}) \subseteq [0, 24)$ is called the *mobility window*, defined as the time interval in which there is mobility from i to j. Here, α_{ij} in the second term, is the proportion in which the number of people will move to its neighbors. It is computed as

$$\alpha_{ij}(k) = \frac{\delta_{ij}(k) f_{ij}(k) \psi_j(k)}{\sum_{q \in \mathcal{N}_i} \delta_{iq}(k) f_{iq}(k) \psi_q(k)},$$

which ensures that $\sum_{j \in \mathcal{N}_i} \alpha_{ij}(k) = 1$.

The supply of j with respect to i, $\Psi_{ij}(k)$ is defined as

$$\Psi_{ij}(k) = \psi_j(k) \frac{(Z_j(k) - N_j(k))\pi_{ij}(k)}{\Delta t}. \tag{6.6}$$

Here, the $\psi_j(k)$ is the supply gating function defined as

$$\psi_j(k) = \begin{cases} 1 & \text{if } k\Delta t \bmod 24 \in [a_j, b_j) \\ 0 & \text{otherwise} \end{cases} \tag{6.7}$$

is the Supply Gating Function(SGF) with $[a_j, b_j) \subseteq [0, 24)$ called the *destination schedule* where a_j is the opening hour and b_j is the closing hour of location j. Similar to DGPs, also SGFs are defined over a day and repeated periodically, possibly with different profiles for different days of the week. For the origins $j \in \mathcal{V}_o$, $[a_j, b_j) = [0, 24)$ because they are always open. $\pi_{ij}(k)$ is the proportion in which the remaining capacity of the the location j will be divided among its neighbors defined as

$$\pi_{ij}(k) = \frac{\delta_{ij}(k) f_{ij}(k)}{\sum_{q \in \mathcal{N}_j} \delta_{qj}(k) f_{qj}(k)}, \tag{6.8}$$

which ensures that $\sum_{i \in \mathcal{N}_j} \pi_{ij}(k) = 1$.

Recall from Chapter 4, that in the mobility model (6.3)-(6.8), the number of people $N_i(k)$ at any location i is always non-negative, i.e., $N_i(k) \geq 0$, and it also remains below the population of P_i or the operating capacity $C_i^o(k)$, if it is an origin or a destination respectively, i.e., $N_i(k) \leq Z_i(k)$. It is important to notice that the flow $\phi_{ij}(k)$ depends on the control $\mathbf{u}(k)$

as it tunes the operating capacities of the destinations which tunes the demand $\delta_{ij}(k)$ and supply $\Psi_{ij}(k)$ through the terms f_{ij} and $Z_j(k)$ respectively.

Moreover, the total mass is also preserved: the total number of people in the system remains constant, i.e., for any time k, we have $\sum_{j\in\mathcal{V}_o\cup\mathcal{V}_d} N_j(k) = \sum_{j\in\mathcal{V}_o\cup\mathcal{V}_d} N_j(0)$. We will use the notation P to indicate such constant number of people in the system.

6.1.2 SIR-mobility model

Now, we recall, the integrated model for mobility and epidemic in discrete-time from Chapter 4. Here, $S_i(k)$, $I_i(k)$ and $R_i(k)$ denote the number of susceptible, infected and recovered people at location i and time $k\Delta t$. We will also use the notation $I(k)$ to denote the total number of infected people at time $k\Delta t$, i.e., $I(k) = \sum_{i\in\mathcal{V}_o\cup\mathcal{V}_d} I_i(k)$. The SIR-mobility model is given by

$$S_i(k+1) = \frac{1}{1+\Delta t\beta_i(k)\frac{I_i(k)}{N_i(k)}}\left[S_i(k) + \Delta t\sum_j\left(\phi_{ji}(k)\frac{S_j(k)}{N_j(k)} - \phi_{ij}(k)\frac{S_i(k)}{N_i(k)}\right)\right] \tag{6.9a}$$

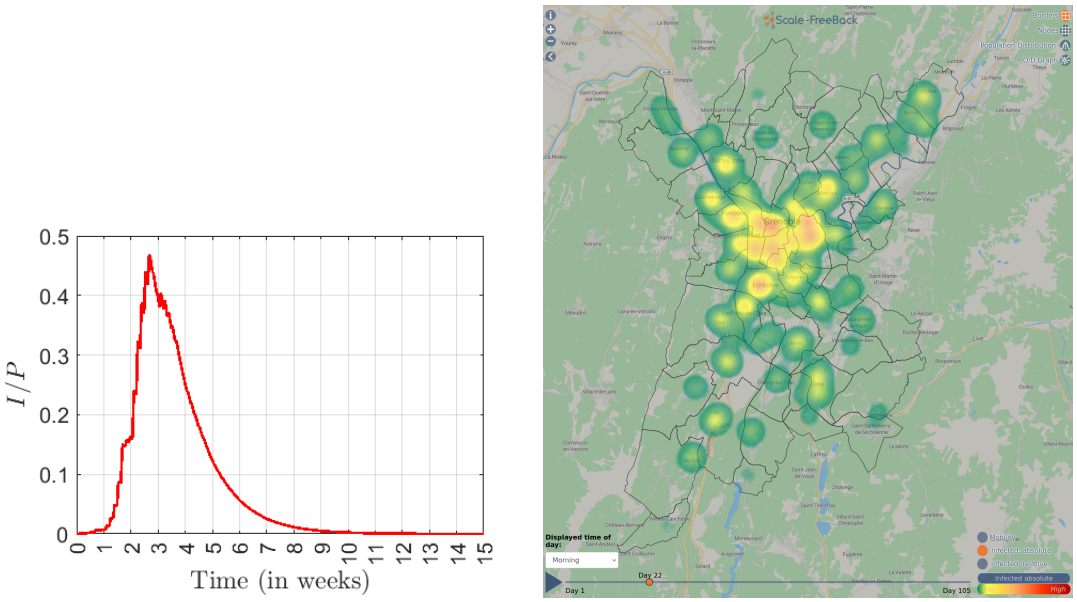
$$I_i(k+1) = \frac{1}{1+\Delta t\gamma}\Bigg[I_i(k) + \Delta t\beta_i(k)S_i(k+1)\frac{I_i(k)}{N_i(k)} + \Delta t\sum_j\left(\phi_{ji}(k)\frac{I_j(k)}{N_j(k)} - \phi_{ij}(k)\frac{I_i(k)}{N_i(k)}\right)\Bigg] \tag{6.9b}$$

$$R_i(k+1) = R_i(k) + \Delta t\gamma I_i(k+1) + \Delta t\sum_j\left(\phi_{ji}(k)\frac{R_j(k)}{N_j(k)} - \phi_{ij}(k)\frac{R_i(k)}{N_i(k)}\right), \tag{6.9c}$$

where the parameters $\beta_i(k) = \bar{\beta}_i\frac{N_i(k)}{X_i}$, $\bar{\beta}_i$ and γ_i are the infection rate, nominal infection rate and recovery rate at i. Here, $X_i = P_i$ if $i \in \mathcal{V}_o$ whereas $X_i = C_i$ if $i \in \mathcal{V}_d$. It is important to note that when $i \in \mathcal{V}_o$, the nominal infection rate $\bar{\beta}_i$ is very low as in the origins, people live in their individual houses and hence the infection spread remains small. Moreover, notice that S_i, I_i, R_i at any location i depends on $\mathbf{u}$ as ϕ_{ij}, N_i, N_j are implicit functions of the control $\mathbf{u}$.

6.1.3 Epidemic scenario in Grenoble area

In this chapter, we consider the large-scale mobility network of Grenoble constructed in Chapter 4 for application and illustration of the results. Recall that there are 60 origins and 374 aggregated destinations grouped in 5 destination categories namely schools, hospitals, workplaces, shopping centers and leisure in this network. Throughout this chapter, destination categories are taken in the order {schools, hospitals, workplaces, shopping centers, leisures} and will be denoted as $\mathcal{D}_h$ for $h \in \{1,2,3,4,5\}$ respectively. The mobility parameters DGP, SGF, M_{ij}, a_j, b_j for each destination categories have been taken as in Chapter 5. For the epidemic, $\gamma_i = 1/8$ per 24 hours and the nominal infection rate $\bar{\beta}_i = 0.25$ per 24 hours for $i \in \mathcal{V}_o$ and that for the destinations in categories schools, workplaces, hospitals, shopping centers, leisure are $2, 2, 4, 2, 4$ per 24 hours respectively. We take time horizon $T = 2520$ hours



(a) (b)

Figure 6.1: (a.)Proportion of infected population I/P in Grenoble area when all the destinations are open up to full capacity. (b.) Heatmap showing infection level in Grenoble area, when all destinations are open to full capacity, at mid-morning on a day near the infection peak. It is a screenshot from the online demonstrator described in Chapter 7

(15 weeks) in all the simulations and sum of total initial infections $\sum_i I_i(0) = 100$ which is distributed randomly only among the origins(residential areas). The initial conditions are

$$\begin{aligned}[N_i(0), I_i(0), S_i(0), R_i(0)] =& [P_i, I_i(0), P_i - I_i(0), 0] \\ & \text{for } i \in \mathcal{V}_o \\ [N_i(0), I_i(0), S_i(0), R_i(0)] =& \mathbf{0} \text{ for } i \in \mathcal{V}_d.\end{aligned} \tag{6.10}$$

With this initialization, the total number of people in the system is $P = \sum_{i \in \mathcal{V}_o} P_i$. All the plots in this chapter refer to the model (6.3)-(6.9) with parameters discussed in this section. All the simulations presented in this chapter has been performed using MATLAB 2020b on a windows PC with Intel(R) Xeon(R) processor running at 2.7GHz using 16MB of RAM. Figure 6.1a depicts the evolution of infection in Grenoble area with no restriction on the capacities of the destinations. It can be seen that more than forty percent of people get infected after some weeks, which can overburden the health infrastructure and cause loss of many lives. This highlights the need for some closure policy, to mitigate the infection level while avoiding an undesirable complete lock-down.

6.2 Optimal capacity control for epidemic mitigation

The SIR-mobility model (6.9) describes the evolution of susceptible, infected and recovered population at each location in the network and also incorporates the mobility of people between

different locations. It can be seen that by applying restrictions in mobility using the control input $u_h(k)$ defined in (6.1), the infection spread in the network can also be controlled. Using this property, in [Nia+21], an optimal capacity control problem has been formulated which aims at maintaining infections below a threshold by mobility control while maximizing the economic activity defined as a function of number of people visiting the destinations. Here, we provide solution techniques for this optimal capacity control problem for a large-scale network.

6.2.1 Socio-economic activity

The socio-economic activity $E(k) \in \mathbb{R}_{\geq 0}$ defined in [Nia+21] at a given time k is

$$E(k) = \sum_{j \in \mathcal{V}_d} \frac{\chi_j}{C_j} N_j(k),$$

where $N_j(k)$ is the number of people at destination j at time k, C_j is the nominal capacity of j and $\chi_j > 0 \in \mathbb{R}$ is the weight assigned to the destination j in the destination category $\mathcal{D}_c$ according to its socio-economic importance in the urban network.

Let us consider a time horizon of interest T, a policy update period T_u, and all the parameters of the model (6.3)-(6.10) be given. Let $\mathbf{u} \in \mathcal{U}$, where $\mathcal{U}$ is the set of vectors of piecewise-constant functions $[u_1(k), ..., u_q(k)]$ as in (6.1), with $p = \lceil T/T_u \rceil$. Recall that in (6.3)-(6.9) there is a dependence on the capacity control $\mathbf{u}$, due to the fact that $\mathbf{u}$ tunes the operating capacities, modifying the flows ϕ_{ij} and hence also all N_i, S_i, I_i and R_i. To highlight this implicit dependence, from now on, we will denote by $N_i(k, \mathbf{u})$ and $I_i(k, \mathbf{u})$ and $E(k, \mathbf{u})$ the number of people and number of infected people at location i and the economic activity, at time $k\Delta t$, when the system is operated with capacity control $\mathbf{u}$.

One of the choices for the weight χ_j is

$$\chi_j = \frac{\chi_h}{\chi_h^{nom}}, \text{ for } j \in \mathcal{D}_h \tag{6.11}$$

where $\chi_h > 0 \in \mathbb{R}$ is a weight given to the destination category h by the policymakers and χ_h^{nom} is the total economic activity done in the destinations of category $\mathcal{D}_h$ and is computed as

$$\chi_h^{nom} = \sum_{k=0}^{T/\Delta T} \sum_{i \in \mathcal{D}_h} \frac{N_i(k, \mathbf{1}_q)}{C_i}.$$

It is the economic activity done in the category h when it is open to full capacity. The weight χ_h corresponds to the socio-economic importance given to the destination category h by policymakers when deciding category-dependent closures and restrictions. The effect of economic weight will be evident when devising the category-dependent control policies in Section 6.4. The weights that will be used in all the simulations in this chapter is given in Table 6.1. Note that these weights are simply meant as an example, to showcase possible effects of extreme weights on the socio-economic cost and control policies devised in later sections, and do not reflect actual governments' policies, nor they are meant as a recommendation.

	Schools	Hospitals	Workplaces	Shopping centers	Leisure
χ_h	0.1	1	0.7	0.8	0.2
χ_h^{nom}	4.2×10^5	1.5×10^4	3.4×10^5	9.7×10^4	3.5×10^5

Table 6.1: Socio-economic weights for different destination categories used in simulations.

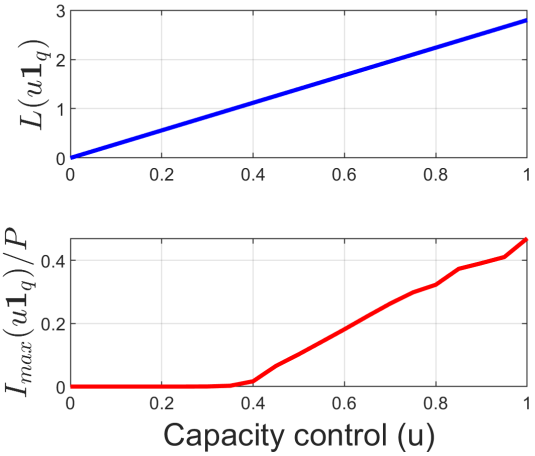


Figure 6.2: Cost function $L(u\mathbf{1}_q)$ and infection peak in proportion to the total population $I_{\max}(u\mathbf{1}_q)/P$ as a functions of the capacity control u, when a fixed control u is applied over the interval $[0, T]$ (i.e., $T_u = T$), with $T = 15$ weeks.

6.2.2 Optimal capacity control problem

Our goal is to keep the economic activity as high as possible, while ensuring that the number of infected people remains below a positive threshold $\bar{I}$ that reflects the capacity of the health-care system, e.g. in relation with the number of ICU beds. More precisely, our goal is to maximize the total economic activity $L(\mathbf{u})$ done in the given time interval of interest $[0, T]$, defined as $L(\mathbf{u}) = \sum_{k=0}^{T/\Delta t} E(k, \mathbf{u})$, i.e,

$$L(\mathbf{u}) = \sum_{k=0}^{T/\Delta t} \sum_{j \in \mathcal{V}_d} \frac{\chi_j}{C_j} N_j(k, \mathbf{u}). \tag{6.12}$$

The constraint that ensures that the capacity of the health-care system is never exceeded can be expressed as

$$\sum_{j \in \mathcal{V}_o \cup \mathcal{V}_d} I_j(k, \mathbf{u}) \leq \bar{I} \; \forall k \text{ s.t. } k\Delta t \in [0, T],$$

or, equivalently, as $I_{\max}(T, \mathbf{u}) \leq \bar{I}$, where

$$I_{\max}(\mathbf{u}) = \max_{k=0,\ldots,T/\Delta t} \sum_{j \in \mathcal{V}_o \cup \mathcal{V}_d} I_j(k, \mathbf{u}) \tag{6.13}$$

is the maximum over the time interval $[0, T]$ of the total number of infected people. In some cases where the interval of interest will be $[T_0, T_f]$ instead of $[0, T]$, then both $L(\mathbf{u})$ and $I_{\max}(\mathbf{u})$

will be computed in the time interval $[T_0, T_f]$ with k in (6.12) and (6.13) ranging from $T_0/\Delta t$ to $T_f/\Delta t$.

With the above notation, the optimal control policy is obtained by solving the following optimization problem:

$$\max_{\mathbf{u}\in\mathcal{U}}\{L(\mathbf{u})\} \tag{6.14a}$$

$$\text{subject to } I_{\max}(\mathbf{u}) \leq \bar{I}, \tag{6.14b}$$

where, $L(\mathbf{u})$ and $I_{\max}(\mathbf{u})$ are obtained with (6.3)-(6.9).

For all the plots in this chapter, we will consider the weights mentioned in the table 6.1 in (6.11) and (6.12) and the threshold on infection $\bar{I} = 0.15P$, where P is the total population.

The optimization problem (6.14) is non-convex since $L(u)$ is a non-convex function as there exist capacity controls $\mathbf{u_1}$ and $\mathbf{u_2}$ such that $L(\lambda\mathbf{u_1} + (1-\lambda)\mathbf{u_2}) > \lambda L(\mathbf{u_1}) + (1-\lambda)L(\mathbf{u_2})$ for $\lambda \in [0,1]$. The problem (6.14) can be easily solved for smaller networks using solvers like *fmincon* in MATLAB as shown in [Nia+21] but for large-scale urban network like that of Grenoble developed in Chapter 5, the solution becomes hard or even impossible to obtain by naive use of off-the-shelf solvers. In this chapter, at first we will start with a simple variation of the problem (6.14) by taking the same control for all the destinations which we call category-free policy. Then, we proceed towards the problem when we have different control policy applied to different destination based on their categories and call it category-dependent policy.

6.2.3 Benchmark control policy

We consider as a benchmark the case where the operating capacity of all the destinations is equal to the nominal capacity that is $u_h(k) = 1$ for all h and k. The economic activity in that case is defined as the nominal economic activity denoted by $\bar{E}(k)$. Figure 6.1a depicts the evolution of infection in Grenoble area in case of benchmark control policy. The aim of the optimization is to control the epidemic and keep the economic activity as close as possible to that of the benchmark case. Therefore, for comparison between different scenarios, we define the *relative economic activity* $R_E(k)$ at time k as

$$R_E(k) = \frac{\sum_{\ell\leq k} E(\ell)}{\sum_{\ell\leq k} \bar{E}(\ell)}. \tag{6.15}$$

Note that $R_E(k) \in [0,1]$, where $R_E(k) = 1$ corresponds to the benchmark scenario where there is no restriction, while $R_E(k) = 0$ corresponds to a complete lockdown.

6.3 Category-free policies

In this section, we start with the simplest variation of the problem that restrict the optimization to the subset of control policies that have the same control for all destination categories,

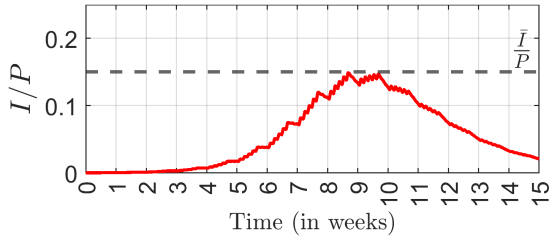


Figure 6.3: Fixed policy: Proportion of infected people I/P when applying the fixed policy $u^*\mathbf{1}_q$, where $u^* = 0.5625$ is obtained by Algorithm 1 with $T_u = T = 15$ weeks.

that is we have $\mathbf{u}(k) = u(k)\mathbf{1}_q$, where $u(k)$ is piece-wise constant as in (6.1) and $\mathbf{1}_q \in \mathbb{R}^q$ is a vector of all ones. This corresponds to considering the following modification of the problem (6.14), where we restrict the search to a subset $\tilde{\mathcal{U}}$ of the set $\mathcal{U}$:

$$\max_{\mathbf{u}\in\tilde{\mathcal{U}}} \{L(\mathbf{u})\} \tag{6.16a}$$

$$\text{subject to } I_{\max}(\mathbf{u}) \leq \bar{I}, \tag{6.16b}$$

where

$$\tilde{\mathcal{U}} = \{u(k)\mathbf{1}_q \text{ s.t. } u(k) \text{ is piece-wise constant as in (6.1)}\},$$

and $L(\mathbf{u})$ and $I_{\max}(\mathbf{u})$ are obtained with (6.3)-(6.10). In Sect. 6.3.1, we will consider a further simplification, choosing $T_u = T$, which corresponds to a constant control over all the time interval $[0, T]$, and we will show how this problem can be solved efficiently, exploiting monotonicity. In Sect. 6.3.2 we will introduce an MPC approach, where the algorithm from Sect. 6.3.1 is applied over a sliding time window, whose length T_m might be equal to T or smaller.

6.3.1 Category-free fixed policy

In this section, we consider problem (6.16) with $T_u = T$, namely the case in which a same fixed value of u is to be applied for all categories and at all times in the interval $[0, T]$. Indeed, when $T_u = T$, piecewise-constant functions as in (6.1) are actually constant functions. We then notice some monotonicity properties of this problem and exploit them in order to find an efficient algorithm. First, we notice that $L(u\mathbf{1}_q)$ is a non-decreasing function of u, as we can see from Figure 6.2. The intuitive explanation is that a larger u allows more visitors to the destinations and hence more economic activity. For this reason, in order to maximize $L(u\mathbf{1}_q)$, we can simply focus on maximising u. Then, we notice that also $I_{\max}(u\mathbf{1}_q)$ is a non-decreasing function of u, except possibly for u very near to zero. This property depends on the parameters of the SIR-mobility model, and corresponds to the scenario of interest, where infection rate β_i is significantly lower at origins than at destinations. The intuitive explanation is that larger u allows more visitors to the destinations, where the infection rate is higher, and hence the number of infected grows more than when some of these persons are forbidden to reach their destinations (except possibly near $u = 0$, where destinations are so empty that the probability

of infection there is small). Figure 6.2 shows $I_{\max}(u\mathbf{1}_q)$ is indeed a non-decreasing function of u in our example (described in Sect. 6.1.3). Provided that $I_{\max}(\mathbf{0}) < \bar{I}$, this property of $I_{\max}$ ensures that the feasible region is simply an interval $[0, \bar{u}] \subseteq [0, 1]$, where $\bar{u}$ is such that $I_{\max}(\bar{u}\mathbf{1}_q) = \bar{I}$. Due to the above-discussed monotonicity of $L(u\mathbf{1}_q)$, $\bar{u}$ is the desired optimal u, that maximizes $L(u\mathbf{1}_q)$ under the constraint that $I_{\max}(\bar{u}\mathbf{1}_q) \leq \bar{I}$. Finding $\bar{u}$ can be done e.g. by bisection method, i.e., by running Algorithm 1 with $[T_0, T_f] = [0, T]$; Algorithm 1 is given on a more general interval $[T_0, T_f]$, because this will be then used within Algorithm 2 in Section 6.3.2.

Algorithm 1: Category-free best fixed policy in $[T_0, T_f]$.

Input: Parameters for (6.3)-(6.9), initial condition $N_i(T_0), I_i(T_0), S_i(T_0), R_i(T_0)\ \forall i$, threshold $\bar{I}$, $I_{\max}(.)$, initial time T_0, final time T_f, maximum number of iterations n_{iter} and tolerance $\epsilon_f > 0$.

Output: $u^* = u_{\min}$

Set $u_{\min} = 0$ and $u_{\max} = 1$
REPEAT
Compute $u = (u_{\min} + u_{\max})/2$,
Run (6.3)-(6.9) with $u(k) = u\mathbf{1}_q$ for $k = T_0/\Delta t, \ldots, T_f/\Delta t$ and compute $I_{\max}(u\mathbf{1}_q)$
IF $I_{\max}(u\mathbf{1}_q) \leq \bar{I}$
    $u_{\min} = u$
ELSE, $u_{\max} = u$
till $(\bar{I} - I_{\max}(u_{\min}\mathbf{1}_q)) \leq \epsilon_f$ or n_{iter} iterations.

Before running Algorithm 1, it is necessary to verify that $I_{\max}(\mathbf{0}) < \bar{I}$, which ensures that there is a feasible solution. Note that the obtained u^* is such that $I_{\max}(u^*\mathbf{1}_q) \leq \bar{I}$ and moreover (unless the algorithm terminates because of reaching n_{iter} iterations) $\bar{I} - I_{\max}(u^*\mathbf{1}_q) < \epsilon_f$.

Under the above-mentioned monotonicity properties, this means that u^* approaches the optimal solution $\bar{u}$ with any desired precision. Also notice that even in the cases where monotonicity does not hold, u^* is ensured to be a feasible solution of the original problem (6.16).

For the large-scale network of Grenoble area, Algorithm 1 with $\epsilon_f = 0.0015P$ and $n_{\text{iter}} = 20$, when applied on $[T_0, T_f] = [0, T]$ with $T = 15$ weeks, gives $u^* = 0.5625$. In Figures 6.5a-6.5c, the plots with $T_u = T = 15$ weeks correspond to the best fixed policy (since $T_u = T$), i.e., to applying the constant policy $u^*\mathbf{1}_q$ with $u^* = 0.5625$. In particular, Figure 6.5c depicts I/P, i.e., the proportion of infected population,

The plot confirms that I/P remains below the threshold $\bar{I}/P = 0.15$, as we expect, since by construction u^* satisfies the constraint $I_{\max}(u^*\mathbf{1}_q) \leq \bar{I}$.

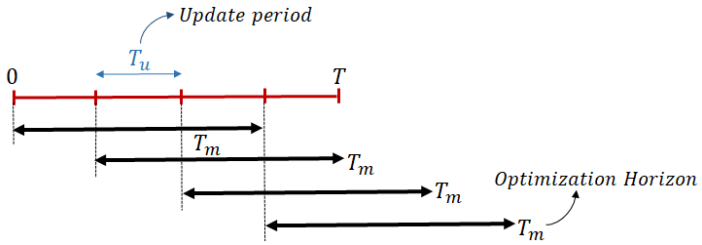


Figure 6.4: Receding horizon approach with update period T_u and the optimization horizon T_m in the time interval $[0, T]$.

6.3.2 Category-free receding-horizon policy

In this section, the considered time interval is $[0, T]$ with the total time horizon T and the policy makers are assumed to update the policy at intervals of length $T_u < T$ but with the same control applied to all the destinations. To find a good, although possibly suboptimal, feasible solution of (6.16) in an efficient way, we adopt a receding horizon approach, where we find a fixed policy within an interval of optimization that is by applying Algorithm 1 over receding time intervals $[\ell T_u, \ell T_u + T_m]$, where $\ell = 0, \dots p$, and the length of the optimization horizon is a parameter $T_m \in [T_u, T]$ to be designed. This approach has been depicted in Figure 6.4 and been summarised in Algorithm 2.

Algorithm 2: Category-free receding-horizon policy, with update period T_u and optimization horizon length T_m

Input: Parameters for (6.3)-(6.9), initial condition $N_i(0), I_i(0), S_i(0), R_i(0)\ \forall i$, threshold $\bar{I}$, $I_{\max}(.), T$, maximum number of iterations n_{iter} and tolerance $\epsilon_f > 0$.

Output: $u^{*1}, \dots, u^{*p}$

1 Initialize $\ell = 1$. Compute $p = \lceil T/T_u \rceil$.
2 **REPEAT**
3 Compute $T_0 = (\ell-1)T_u$ and $T_f = (\ell-1)T_u + T_m$
4 Apply Algorithm 1 over $[T_0, T_f]$ with tolerance ϵ_f and maximum number of iterations n_{iter}, and denote by $u^{*\ell}$ the obtained output
5 Run (6.3)-(6.9) over the interval $[(\ell-1)T_u, \ell T_u]$, with fixed policy $u^{*\ell}\mathbf{1}_q$, so as to obtain $N_i(\ell T_u/\Delta t)$, $S_i(\ell T_u/\Delta t)$, $I_i(\ell T_u/\Delta t)$, $R_i(\ell T_u/\Delta t)$ for all i (to be used as initialization for the next iteration)
6 $\ell = \ell + 1$
7 until $\ell = p$.

Notice that the piecewise constant function $u^*(k)$ defined by $u^*(k) = u^{*\ell}$ for all $k = (\ell-1)T_u/\Delta t, \dots, \ell T_u/\Delta t$ is a feasible solution of (6.16).

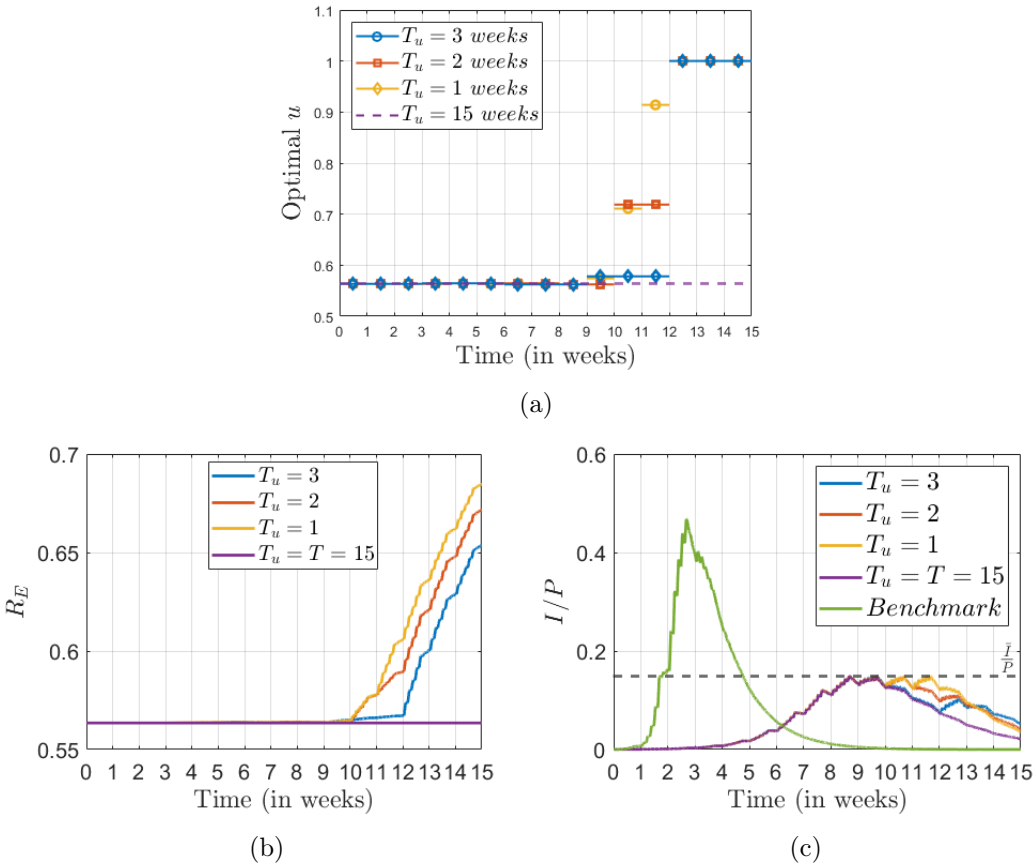


Figure 6.5: Category-free receding-horizon policy with different update period T_u: (a) Optimal control $u^*(k)$, (b) Relative economic activity $R_E(k)$, and (c) proportion of infected people (I/P), when Algorithm 2 is applied with $T_m = 15$ weeks and $T_u = 1, 2, 3, 15$ weeks. (c) also shows the benchmark case ($u(k)$ constantly equal to 1).

6.3.3 Simulations

For the large-scale network of Grenoble area, and with an horizon of interest $T = 15$ weeks, Figure 6.5 illustrates the effect of different update periods T_u, with curves corresponding to $T_u = 1$ week, $T_u = 2$ weeks, $T_u = 3$ weeks and $T_u = T = 15$ weeks (the latter is the fixed policy discussed in Sect. 6.3.1). For all curves, the optimization horizon is $T_m = 15$ weeks. Figure 6.5a shows the optimal control policy, i.e., the piecewise constant function $u^*(k)$ obtained from Algorithm 2, while Fig. 6.5a and Fig. 6.5c show the corresponding relative economic activity and proportion of infected people. We can see that all policies start with the same control u^{*1}, since they all start from the same initial infection status, and they all find the best constant u for an optimization horizon of $T_m = 15$ weeks. They also remain equal until the infection peak (which is smaller than in the benchmark case, and below $\bar{I}$, as enforced by the constraint $I_{\max}(u\mathbf{1}_q) \leq \bar{I}$. Then, after the peak, we can see the advantage of a smaller update period T_u, which allows a faster adaptation to the current epidemic conditions: as infections decrease, the capacity control u is increased and lets the economic activity grow, and this can happen

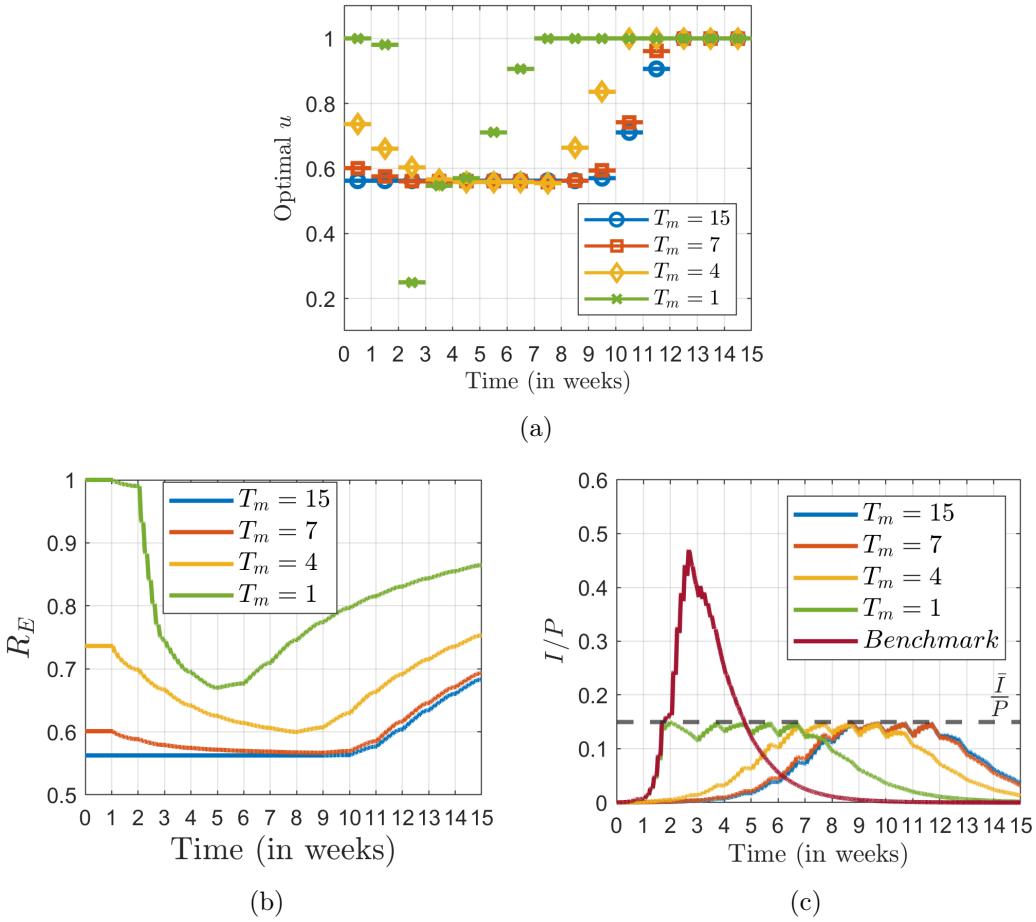


Figure 6.6: Category-free receding-horizon policy with different optimization horizon T_m: (a) Optimal control $u^*(k)$, (b) Relative economic activity $R_E(k)$, and (c) proportion of infected people (I/P), when Algorithm 2 is applied with $T_u = 1$ week and $T_m = 1, 4, 7, 15$ weeks. (c) also shows the benchmark case ($u(k)$ constantly equal to 1).

earlier for smaller values of T_u, leading to the best economic activity in the case of the smallest T_u. This suggests to use the smallest T_u that can be socially acceptable, taking into account the fact that extremely short values of T_u, of the order e.g. of a day, would create confusion and lack of respect of the rules. As a side note, we can see the epidemic curve lasts longer in the optimized cases than in the benchmark case; this is a well-known effect of policies aiming at 'flattening the curve' so as to keep the number of infections below the rate that saturates health infrastructure.

Figure 6.6, instead, illustrates the effect of different optimization horizons T_m: Algorithm 2 is applied with $T_m = 1$, 4, 7 and 15 weeks, while keeping a same small update period $T_u = 1$ week, which is chosen because a small value of T_u is recommended, as discussed above. It can be seen in Figure 6.6b that performance improves when T_u decreases. The best solution is obtained when $T_m = 1$, which corresponds to a U-shape control (see Figure 6.6a): full capacity ($u = 1$) initially, when there are few infections, then strong limitations (small u) near

the infection peak, followed by gradual reopenings, progressively augmenting u while keeping the infection level near the threshold, until full reopening at the end of the infection wave. The fact that the short-sighted policy outperforms the long-sighted ones might seem somewhat counter-intuitive, but it can be clarified by recalling that Algorithm 2 looks for the best *fixed* policy over each receding-horizon interval. Thus, Algorithm 2 with a large T_m enforces an overly-cautious small u^{*1}, since despite the small initial number of infections it chooses u^{*1} sufficiently small to flatten the infections peak that would happen in the coming T_m weeks. With small T_m, instead, Algorithm 2 quickly adapts the control input to current infection level, leading to best performance, both because of a larger initial activity, and because the infection peak happens earlier, allowing for earlier partial reopenings. However, notice that this performance comes at the price of a more aggressive control policy, that uses some lower values of u, although for short time, as it is clearly illustrated in Figure 6.6a. The social acceptability of such restrictions should be taken into account in the policy design.

From these simulations in this section, we can draw the recommendation to design policies using Algorithm 2 with the smallest values of the update period that is $T_u = 1$ and of the optimization horizon $T_m = 1$, within the margin of social acceptability.

6.4 Category-dependent policies

During the COVID-19 outbreak, it was noticed that various governments had different strategies of closures and restrictions for different types of destinations according to their socio-economic importance to save both lives and the economy. Therefore, in this section, we devise policies that apply different capacity controls to destinations in different categories in the optimal control problem (6.14). For this, we implement monotonic optimization techniques.

Throughout this section, for any two vectors, $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, we use the notation $\mathbf{u} \leq \mathbf{v}$(or $\mathbf{u} < \mathbf{v}$) if $u_i \leq v_i$(or $u_i < v_i$) for all $i = 1 \dots n$.

6.4.1 Monotonic optimization

Now, we introduce some mathematical preliminaries of monotonic optimization [Tuy00]; [TAKT05].

Definition 6.1. ***Non-decreasing functions:*** *A function $f : \mathbb{R}^n_+ \to \mathbb{R}$ is an non-decreasing function if $f(\mathbf{x}) \leq f(\mathbf{y})$ when $\mathbf{0} \leq \mathbf{x} \leq \mathbf{y}$, where $\mathbb{R}^n_+$ denotes the set of non-negative real numbers.*

Definition 6.2. ***Box:*** *If $\mathbf{0} \leq \mathbf{b}$, then the set of all $\mathbf{x} \in \mathbb{R}^n$ such that $\mathbf{0} \leq \mathbf{x} \leq \mathbf{b}$ is defined as the box $[\mathbf{0}, \mathbf{b}]$ with the vertex $\mathbf{b}$.*

Definition 6.3. ***Normal set:*** *A set $G \subset \mathbb{R}^n_+$ is normal if for any element $x \in G$, all other elements x' such that $0 \leq x' \leq x$ are also in set G. In other words, the set G is normal if given any element $\mathbf{x} \in G$, then the box $[\mathbf{0}, \mathbf{x}] \subset G$.*

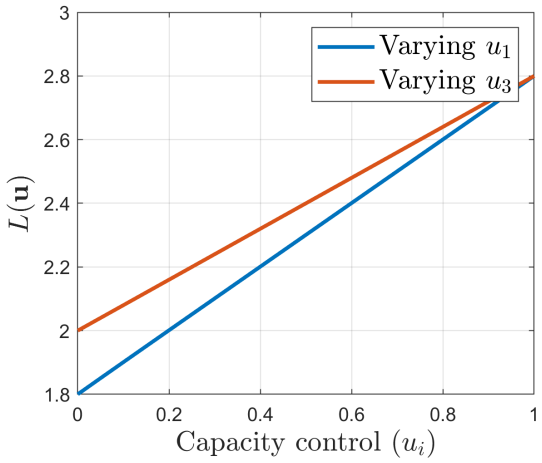


Figure 6.7: Cost function $L(\mathbf{u})$ for $T = 15$ weeks and when we freeze all variables except one. The free variable is increased in $[0, 1]$ and the others are fixed at $\mathbf{1}$. Here, $\mathbf{u} \in [0, 1]^5$ as we have $T_u = T$. The plots are obtained from two separate simulations in which a single variable is kept free namely u_1 and u_3 respectively.

An optimization problem belongs to the class of monotonic optimization[1] if it can be expressed in the following form:

$$\max_{\mathbf{x}} F(\mathbf{x}) \tag{6.17a}$$

$$\text{subject to } \mathbf{x} \in G, \tag{6.17b}$$

where, $F(\mathbf{x}) : \mathbb{R}^n_+ \longrightarrow \mathbb{R}$ is an non-decreasing function and $G \subset [\mathbf{0}, \mathbf{b}] \subset \mathbb{R}^n_+$ is a compact normal set with nonempty interior. In the optimal problem discussed in this chapter, the objective function $F(x)$ is given by $L(x)$ defined in (6.14) and the feasible set G is defined as

$$G = \{\mathbf{u} \in \mathcal{U} | I_{\max}(\mathbf{u}) \leq \bar{I}\}. \tag{6.18}$$

In the range of parameters of interest, where at least a total lockdown is enough to keep the number of infected people below $\bar{I}$, notice that $\mathbf{0} \in \mathbb{R}^n$ is always in G, therefore G always has a non-empty interior. Also notice that for the problem (6.14), G is enclosed in the box $[\mathbf{0}, \mathbf{1}]$, where $\mathbf{1} \in \mathbb{R}^n$. In order to use monotonic optimization techniques to solve (6.14), we need two properties, namely the cost function $L(\mathbf{u})$ be non-decreasing and the feasible set $G = \{\mathbf{u} \in \mathcal{U} | I_{\max}(\mathbf{u}) \leq \bar{I}\}$ be a normal set, where $I_{\max}(\mathbf{u})$ is computed using (6.3)-(6.9). At first, notice that the function $L(\mathbf{u})$ defined in (6.12) is a non-decreasing function of $\mathbf{u}$. The intuitive explanation is that given two control vectors $\mathbf{u}, \mathbf{v} \in \mathcal{U}$ such that $\mathbf{v} \geq \mathbf{u}$, implies that the number of visitors in destinations is higher with the control $\mathbf{v}$ than with the control $\mathbf{u}$. Therefore, more economic activity at the destinations which implies that $L(\mathbf{v}) \geq L(\mathbf{u})$. Figure 6.7 shows an example where when we freeze all variables in $\mathbf{u} \in \mathbb{R}^5$ except one variable,

[1]The general form of monotonic optimization problem from the literature [Tuy00]; [TAKT05] also allows for a further constraint being a so-called co-normal set, but in our case only the normal constraint is involved, and hence we only include the normal set in this section.

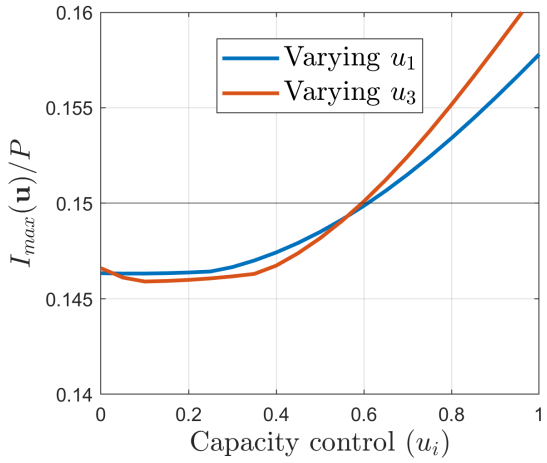


Figure 6.8: Cost function $I_{\max}$ for $T = 15$ weeks and when we freeze all variables at feasible point $0.5625 * \mathbf{1}_5$ except one. The free variable is varied in $[0, 1]$ and the others are fixed at $0.5625 * \mathbf{1}_4$. Here, $\mathbf{u} \in [0, 1]^5$ as we have $T_u = T$. The plots are obtained from two separate simulations in which a single variable is kept free namely u_1 and u_3 respectively.

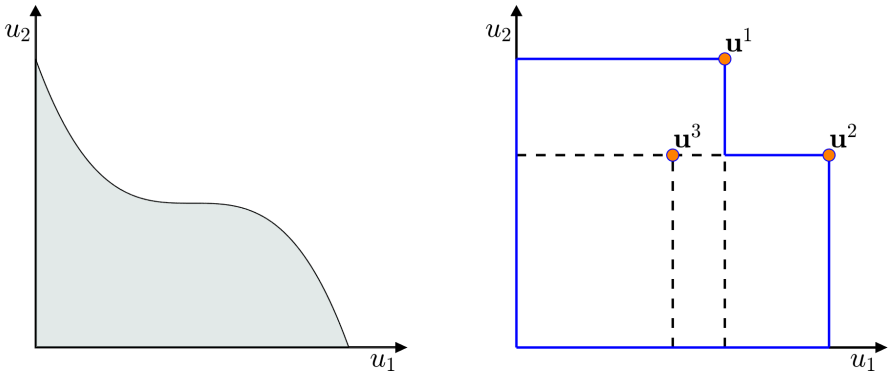
which we vary and we can see that indeed in this case $L(\mathbf{u})$ is non-decreasing which is according to our intuition.

In order to show that G defined in (6.18) is a normal set, we need to show that if control $\mathbf{z} \in G$, then any $\mathbf{v} \leq \mathbf{z}$ satisfies $\mathbf{v} \in G$. Consider $\mathbf{v} \leq \mathbf{z}$, then the number of visitors in each destination is lower with the control $\mathbf{v}$ and higher with the control $\mathbf{z}$. Since the infection rate is higher at destinations, the number of infected people is more in case of more visitors at destinations (except possibly near $\mathbf{u} = \mathbf{0}$, where destinations are so empty that the probability of infection there is small). This certainly depends on the parameters of the SIR-mobility model, and corresponds to the scenario of interest, where infection rate β_i is significantly lower at origins than at destinations. Therefore, if $I_{\max}(\mathbf{z}) \leq \bar{I}$, then we have that also $I_{\max}(\mathbf{v}) \leq \bar{I}$ for $\mathbf{v} \in [\mathbf{0}, \mathbf{z}]$ except possibly for a very low threshold $\bar{I}$, which is outside our interest, since it would correspond to the case where even a total lockdown, i.e. $\mathbf{u} = \mathbf{0}$, would be unfeasible. It can also be seen in Figure 6.8, where we freeze all the variables at a feasible point and vary one variable in $[0, 0.7]$, that once a $\mathbf{z}$ is feasible, it remains feasible for all $\mathbf{u}$ such that $u_i < z_i$ for $i = 1$ while varying u_1 and for $i = 3$ while varying u_3.

6.4.2 Polyblock outer approximation algorithm

Since the objective function (6.12) is non-decreasing, the optimal value lies on the boundary of the feasible set G [Tuy00]; [TAKT05]. However, we do not have the knowledge of the boundary of the feasible set G. Therefore, we aim to have an approximation of the boundary using some outer approximation methods notably Polyblock algorithm[Tuy00]. We start by defining a polyblock and related terms first.

Definition 6.4. ***Polyblocks:*** *A set $\mathcal{P} \subset \mathbb{R}^n_+$ is called a polyblock if it is a union of a finite*



(a) (b)

Figure 6.9: (a) Example of a normal set in $\mathbb{R}^2$ (b) Example of a polyblock with vertices $\mathbf{u}^1$,$\mathbf{u}^2$ and $\mathbf{u}^3$. $\mathbf{u}^1$ and $\mathbf{u}^2$ are proper while $\mathbf{u}^3$ is improper.

number of boxes $[\mathbf{0}, \mathbf{z}]$, *where* $\mathbf{z} \in \mathcal{T}$ *and* $|\mathcal{T}| < +\infty$. *The set* $\mathcal{T}$ *is called the vertex set of the polyblock. A vertex* $v \in \mathcal{T}$ *is said to be proper if there is no* $v' \in \mathcal{T}$ *such that* $v' \neq v$ *and* $v' \geq v$. *The vertices which are not proper are called improper vertices. Improper vertices can be removed from the vertex set* $\mathcal{T}$ *without affecting the shape of the polyblock.*

Definition 6.5. ***Upper boundary:*** *A point* $\bar{x}$ *of a normal closed set* G *is called an upper boundary point of* G *if* $G \cap \{x \in \mathbb{R}^n_+ | x > \bar{x}\} = \emptyset$. *The set of all upper boundary points of* G *is called its upper boundary and denoted by* $\partial^+ G$.

Definition 6.6. ***Projection on the upper boundary:*** *let* $G \subset \mathbb{R}^n_+$ *be a compact normal set with non-empty interior. Then for any point* $\mathbf{z} \in \mathbb{R}^n_+ \backslash G$, *the line segment joining* $\mathbf{0}$ *to* $\mathbf{z}$ *meets the outer boundary* $\partial^+ G$ *at a unique point* $\pi_G(\mathbf{z})$ *defined as*

$$\pi_G(\mathbf{z}) = \lambda \mathbf{z}, \ \lambda = \max\{\alpha > 0 | \alpha \mathbf{z} \in G\}$$

$\pi_G(\mathbf{z})$ *is called the projection of* $\mathbf{z}$ *on the upper boundary of* G.

In Polyblock outer approximation algorithm, the basic idea is to approach the boundary of the feasible set by constructing a sequence of polyblocks. At each iteration, it refines the polyblocks which is an upper bound for the feasible set G. The maximum of an non-decreasing function over a polyblock is attained at one of its proper vertices[Tuy00], therefore, it makes it easier to find the maximum of F over a polyblock. The polyblock algorithm maximizes the non-decreasing objective function on a sequence of polyblocks that enclose the feasible set (or a subset of the feasible set containing the optimal solution). At each iteration k, we find a vertex $\mathbf{z}^k$ that maximizes $F(\mathbf{x})$ *i.e.*

$$\mathbf{z}^k = \underset{\mathbf{x} \in \mathcal{T}^k}{\operatorname{argmax}} F(\mathbf{x}),$$

where $\mathcal{T}_k$ is the vertex set of the polyblcok $\mathcal{P}^k$ enclosing the feasible set G. Therefore, $F(\mathbf{z}^k)$ is an upper bound to the optimal $F(\mathbf{x}^*)$ where

$$\mathbf{x}^* = \underset{\mathbf{x} \in G}{\operatorname{argmax}}(F(\mathbf{x})).$$

Moreover, $\mathbf{z}^k$ might not be feasible, therefore, it is projected on a point $\mathbf{x}^k$ on boundary of the feasible set G. Since, $\mathbf{x}^k$ is on the boundary of G, it is used to cut the cone $\mathcal{K}^+_{\mathbf{x}^k}$, where

$$\mathcal{K}^+_{\mathbf{x}^k} = \{\mathbf{y} \in \mathbb{R}^n_+ | \mathbf{y} > \mathbf{x}^k\}$$

in order to refine the polyblock. Moreover, since $\mathbf{x}^k$ is a feasible point, $F(\mathbf{x}^k)$ is a lower bound to the optimal $F(\mathbf{x}^*)$. More precisely, the algorithm keeps memory of the "current best value (CBV_k)", namely the largest among $F(\mathbf{x}^1), ..., F(\mathbf{x}^k)$. Clearly CBV_k can be updated iteratively, as follows:

$$CBV_k = \max\{CBV_{k-1}, F(\mathbf{x}^k)\}.$$

At each iteration the polyblock is refined, and hence the upper bound $F(\mathbf{z}^k)$ is decreased, while on the other hand the lower bound CBV_k either remains constant or increases. The algorithm terminates when the lower and the upper bounds are within a desired precision threshold, thus certifying optimality, then the algorithm outputs CBV_k, together with the corresponding feasible point. Let us see in details the first iterations of the algorithm. The algorithm starts with the polyblock $P^1 = [0, b]$, which contains the feasible set G and corresponds to $\mathcal{T}^1 = \{b\}$, and hence $z^1 = \mathbf{b}$. Then $\mathbf{z}^1$ is projected on G along the line segment joining $\mathbf{0}$ and $\mathbf{z}^1$ to obtain $\mathbf{x}^1$ on the boundary of G.

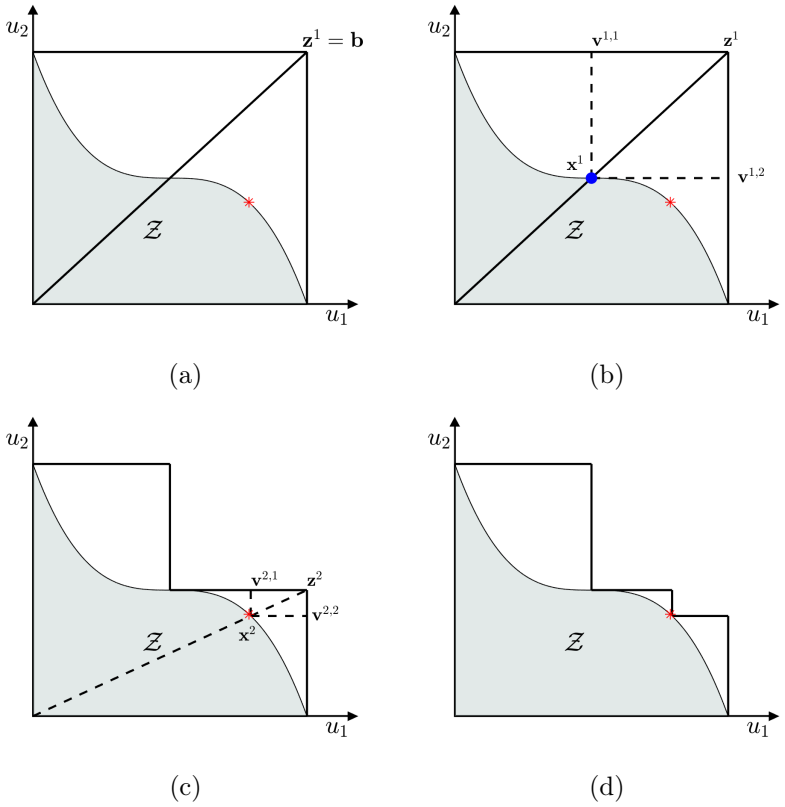


Figure 6.10: Illustration of the polyblock outer approximation algorithm in $\mathbb{R}^2$. The optimal point denoted by a red star is on the boundary of the normal set $\mathcal{Z} \in \mathbb{R}^2$.

Then, a smaller polyblock $\mathcal{P}^2$ enclosing G is constructed based on $\mathcal{P}^1$ by replacing $\mathbf{z}^1$ with

n new vertices $\tilde{\mathcal{T}}_1 = \{\mathbf{v}^{1,1} \dots \mathbf{v}^{1,n}\}$. These new vertices $\mathbf{v}^{1,i}$ is computed as $\mathbf{v}^{1,i} = \mathbf{z}^1 + ((\mathbf{x})_i^1 - (\mathbf{z}^1)_i)e_i$, where $e_i \in \mathbb{R}^n$ is standard basis vector and $\mathbf{x}^1$. Thus, the vertex set of the newly constructed polyblock $\mathcal{P}^2$ is $\mathcal{T}_2 = \mathcal{T}_1 \setminus \{\mathbf{z}^1\} \cup \tilde{\mathcal{T}}_1$. Then, we choose the vertex which maximizes the objective function F of the problem in (6.17), *i.e.*, $\mathbf{z}^2 = \text{argmax}_{\mathbf{z} \in \mathcal{T}_2} F(\mathbf{z})$ for $\mathbf{z} \in \mathcal{T}_2$. This process is repeated successively to construct even smaller polyblocks based on $\mathcal{P}^2$ and so on such that

$$G \subset \dots \mathcal{P}_k \subset \dots \mathcal{P}_2 \subset \mathcal{P}_1.$$

Figure 6.10 illustrates this process for a normal set in $\mathbb{R}^2$. The algorithm terminates when $|F(\mathbf{z}^k) - F(\mathbf{x}^k))| \leq \epsilon$, where $\epsilon > 0$ is the error tolerance specifying the error tolerance of the approximation. Under some mild conditions, [Tuy00], [TAKT05] show that $\mathbf{x}^k$ converges to a global optimal solution of the problem for $k \to \infty$. The basic polyblock outer approximation algorithm has been summarised in the Algorithm 3. In particular, step 4 searches for the vertex $\mathbf{z}^k$ which maximizes the objective function $F(\mathbf{z}^k)$ among the vertices of the polyblock $\mathcal{P}$. Step 5 computes the projection of $\mathbf{z}^k$ on the feasible set G. Step 6,7 and 8 indicate whether the optimal vertex $\mathbf{z}^k$ is already feasible. Otherwise, Step 9 creates a smaller polyblock enclosing G. Finally, step 12 check if the algorithm can be terminated.

There are three steps in the algorithm 3 that we need to comment further:

- **Step 5:** Computing the projection of z_k on the upper boundary of G, $\pi_G(\mathbf{z}^k) = \mathbf{x}^k$.- The projection of the vertex $\mathbf{z}^k$ on the boundary of the feasible set G can be computed using the bisection search algorithm. Clearly, an approximation is used in bisection, therefore $\mathbf{x}^k$ ends up inside the G within the approximation threshold.

- **Step 9(b):** Computing the vertex set $\mathcal{T}_{k+1}$ of the new polyblock. - Once, the projection $\mathbf{x}^k = \pi_G(\mathbf{z}^k)$ is computed, we know that $\mathbf{x}^k \in \partial^+ G$ within the approximation threshold used in bisection search. Since, $L(\mathbf{z})$ is monotonic function, and since G is a normal set, therefore the set of points in

  $$\mathcal{K}^+_{\mathbf{x}^k} = \{\mathbf{y} \in \mathbb{R}^n_+ | \mathbf{y} > \mathbf{x}^k\}$$

  are infeasible. Therefore, the new Polyblock $\mathcal{P}_{k+1}$ is constructed by cutting the cone $\mathcal{K}^+_{\mathbf{x}^k}$ out from $\mathcal{P}_k$. To create the new set of vertices $\mathcal{T}_{k+1}$, at first we compute

  $$\tilde{\mathcal{T}}_k = \{\mathbf{v}^{k,1} \dots \mathbf{v}^{k,n}\}.$$

  These new vertices $\mathbf{v}^{k,i}$ are computed as

  $$\mathbf{v}^{k,i} = \mathbf{z}^k + ((\mathbf{x})_i^k - (\mathbf{z}^k)_i)e_i, \forall i \tag{6.19}$$

  where $\mathbf{x}^k$ is the projection of $\mathbf{z}^1$ on G and $e_i \in \mathbb{R}^n$ is standard basis vector. Notice that $\mathbf{v}^{k,i}$ is obtained by replacing the i^{th} entry of $\mathbf{z}^k$ by the i^{th} entry of $\mathbf{x}^k$. Finally, the set of vertices $\mathcal{T}_{k+1}$ of the polyblock $\mathcal{P}_{k+1}$ is computed as

  $$\mathcal{T}_{k+1} = \mathcal{T}_k \setminus \{\mathbf{z}^k\} \cup \tilde{\mathcal{T}}_k.$$

- **Step 9(c):** Removing the improper vertices from $\mathcal{T}_{k+1}$– The improper vertices in $\mathcal{T}_{k+1}$ are those vertices $\mathbf{v}^{k,i} \in \tilde{\mathcal{T}}_k$ for which there exists another vertex $\mathbf{w} \in \mathcal{T}_{k+1}$ such that $\mathbf{v}^{k,i} \leq \mathbf{w}$ [ZQH13]. The improper vertices can be removed after comparing these vertices with each other.

Algorithm 3: Outer Polyblock Approximation Algorithm

Input: $F(\mathbf{z}), G$, tolerance ϵ

Output: $\epsilon-$optimal solution $\mathbf{x}^*$

Initialize the vertex set $\mathcal{T}_1 = \mathbf{b}$. Set the Current Best Value$(CBV_0) = 0$.

Set $k = 1$.

REPEAT

Select $\mathbf{z}^k = \text{argmax}\{F(\mathbf{z}) | \mathbf{z} \in \mathcal{T}_k\}$,

Compute the projection of $\mathbf{z}^k$, i.e. $\mathbf{x}^k = \pi_G(\mathbf{z}^k)$ on the upper boundary of G.

IF $\mathbf{x}^k = \mathbf{z}^k$,

  Set current best feasible solution $\bar{\mathbf{x}}^k = \mathbf{x}^k$

  $CBV_k = F(\mathbf{z}^k)$

ELSE

(a) IF $F(\mathbf{x}) \geq CBV_{k-1}$,

$\bar{\mathbf{x}}^k = \pi_G(\mathbf{z}^k)$ and $CBV_k = F(\pi_G(\mathbf{z}^k))$

ELSE

$\bar{\mathbf{x}}^k = \bar{\mathbf{x}}^{k-1}$ and $CBV_k = CBV_{k-1}$.

END IF

(b) Compute the new vertices $\mathcal{T}_{k+1}$ of the next polyblock.

i) Create the new vertices from $\mathbf{z}_k$ and $\mathbf{x}^k$ by following the rule: $\mathbf{v}^{k,i} = \mathbf{z}^k + ((\mathbf{x}^k)_i - (\mathbf{z}^k)_i)e_i$ for all $i \in \{1, \ldots, n\}$.

ii) $\mathcal{T}_{k+1} = \mathcal{T}_k \setminus \{\mathbf{z}^k\} \cup \{\mathbf{v}^{k,i}\}$, where $\mathbf{v}^{k,i}$ are the vectors computed in the previous step.

(c) Remove all the improper vertices from $\mathcal{T}_{k+1}$.

End IF

$k = k + 1$

Until $|F(\mathbf{z}_k) - CBV_k| \leq \epsilon$, where $\epsilon > 0$ is a small positive number.

$\mathbf{u}^* = \bar{\mathbf{x}}^k$

6.4.3 Implementation issues and Enhancements

There are some enhancements from the literature [ZQH13]; [CAKA05] that we implemented to Algorithm 3. These enhancements improve its performance significantly to solve the problem (6.14). We discuss them in the following sections.

6.4.3.1 Anti-jamming

Notice that, in Algorithm 3, each iteration is computationally heavy, in particular the projection of $\mathbf{z}^k$, on the feasible set G which requires computing F numerous times. Note that, in our problem it would mean running SIR-mobility model multiple times. Hence, it would be desirable to obtain convergence after a reasonably limited number of iterations. However, in some cases polyblock algorithm is known to have a very large number of iterations, due to a phenomenon known as jamming. When $\mathbf{z}^k$ is such that $\mathbf{z}_i^k$ is close to zero for some i, then in order to converge from $\mathbf{z}^k$ to $(\mathbf{z}_1^k, \ldots, \mathbf{z}_{i-1}^k, 0, \mathbf{z}_{i+1}^k, \ldots, \mathbf{z}_n^k)$, a subsequence of such vertices are created and their difference in euclidean norm being very small. This phenomenon is termed as *jamming*. In particular, *jamming* occurs when the angle between the line from $\mathbf{0}$ to the vertex $\mathbf{z}^k$ and one of the axes is steep. In order to prevent *jamming*, [CAKA05] proposes to find the projection of $\mathbf{z}^k$ on G alternatively.

Consider that $\mathbf{z}^k$ maximizes $\{F(\mathbf{z})|\mathbf{z} \in \mathcal{T}_k\}$ at iteration k. Let d be the minimum non-zero distance between $\mathbf{z}^k$ and $\mathbf{0}$ such that

$$d = \min_{\{i:z_i^k>0\}} (z_i^k). \tag{6.20}$$

Define a new vertex $\mathbf{b'}$ as follows:

$$b'_j = \begin{cases} z_j^k - d & \text{when} \quad z_j^k > 0 \\ 0 & \text{when} \quad z_j^k = 0. \end{cases} \tag{6.21}$$

Now, notice that if $\mathbf{b'} \in G$, then there exists a projection of $\mathbf{z}^k$ on $\partial^+ G$ along the line segment joining $\mathbf{0}$ and $\mathbf{z}^k$. If $\mathbf{b'} \notin G$, then instead, we now find the projection of $\mathbf{b'}$ on $\partial^+ G$ along the line segment joining $\mathbf{0}$ and $\mathbf{b'}$. In both the cases, we can obtain a point $\mathbf{x}^k$ on $\partial^+ G$ in the box $[\mathbf{0}, \mathbf{z}^k]$. Therefore, the next polyblock can be created by removing the region $(\mathbf{x}^k, \mathbf{z}^k]$. The main importance of this technique is that it pushes any vertex close to the axis i.e. vertex with one entry near 0 by distance d to the axis by forcing it to be 0 in a single iteration rather than having recurring iterations in order to become 0. In order to implement this enhancement, step 5, and step 9 in Algorithm 3 need to be modified as follows:

1. Step 5:
   a) Let d be the minimum non-zero distance between $\mathbf{z}^k$ and $\mathbf{0}$ computed in (6.20).
   b) Compute the modified vertex $\mathbf{b'}$ as defined in (6.21).
   c) IF $\mathbf{b'} \in G$

– THEN Compute the projection of $\mathbf{z}^k$ on $\partial^+ G$ i.e. compute $\mathbf{x}^k = \pi_G(\mathbf{z}^k)$.

d) ELSE

– Compute the projection of $\mathbf{b}'$ on $\partial^+ G$ i.e. compute $\mathbf{x}^k = \pi_G(\mathbf{b}')$.

2. Step 9

i) Create the new vertices $\mathbf{v}^{k,i}s$ from $\mathbf{z}_k$ and $\mathbf{x}^k$ by the following rule:

$$\mathbf{v}^{k,i} = \mathbf{z}^k + ((\mathbf{x})_i - (\mathbf{z}^k)_i)e_i$$

for all i such that $\mathbf{z}_i^k > \mathbf{x}_i^k$.

ii) $\mathcal{T}_{k+1} = \mathcal{T}_k \setminus \{\mathbf{z}^k\} \cup \{\mathbf{v}^{k,i}\}$, where $\mathbf{v}^{k,i}$ are the vectors computed in the previous step.

Moreover, we notice that if d is large enough then $\mathbf{b}' \in G$, therefore, we set a threshold $\bar{d}$ on the minimum distance d computed in (6.20) and apply the correction above only if $d < \bar{d}$. This reduces the computation effort in checking if $\mathbf{b}' \in G$ in every iteration since it is checked only when $d < \bar{d}$. Furthermore, to prevent recurring *jamming* due to sufficiently small x_i^k in the previous steps, round off the computed projection $\mathbf{x}^k$ if it is larger than a set tolerance. This threshold should be chosen such that its effect on the objective function is negligible.

6.4.3.2 Removal of suboptimal vertices from $\mathcal{T}_{k+1}$

As the number of iterations grows, so does the size of the vertex set $\mathcal{T}_{k+1}$. This might lead to computational complexity in finding the optimal vertex $\mathbf{z}^k$ in Step 4. Since the Algorithm 3 maximizes the CBV_k or keeps it constant in successive iterations, the vertices with the value of F smaller than the CBV_k cannot be optimal and can be safely discarded [ZQH13]. With this enhancement, Step 9c and Step 12 can be modified as follows:

1. Step 9c:

– Remove all the improper vertices from $\mathcal{T}_{k+1}$ and the vertices w such that

$$\{w | F(\mathbf{w}) \leq CBV_k + \epsilon\}.$$

2. Step 12:

– **Until:** $|F(\mathbf{z}_k) - CBV_k| \leq \epsilon$, OR $\mathcal{T}_k = \emptyset$, where $\epsilon > 0$ is a small positive number.

Moreover, the condition $|F(\mathbf{z}_k) - CBV_k| \leq \epsilon$ can be checked after step 9a, and if this condition is satisfied, then GO TO step 13. This modification will prevent unnecessary computation of $\mathcal{T}_{k+1}$ and therefore save the effort in computing the objective function for the new vertices.

Although polyblock algorithm guarantees optimal solution it is known to be slow in convergence [Cho+19] and like other non-convex optimization solver algorithms, it can be computationally demanding as the dimension increases. Therefore, in addition to enhancements,

as mentioned earlier, we set a maximum number of iterations for the Algorithm 3. Even if the algorithm terminates because the maximum number of iterations has been reached, the obtained solution might or might not be optimal but will always be feasible because of the projection of the vertex on the boundary of the feasible set.

The final Polyblock algorithm after the above mentioned enhancements in Algorithm 3, the modified polyblock algorithm which can be implemented to the problem (6.14) has been presented in Algorithm 4 and the method for projection on the boundary of the feasible set G defined in (6.18) has been presented in Algorithm 5.

Now, we can implement the polyblock algorithm presented in Algorithm 4 to obtain category-dependent capacity control policies applied to destinations in different categories according to their economic importance. At first, we solve the problem 6.14 in case where we have a fixed policy for the total time horizon that is $T_u = T$, namely, for each destination category $\mathcal{D}_h$, we compute a control $u_h(k)$ that remains constant for all k such that $k\Delta t \in [0, T]$. Then, we apply an MPC approach to obtain such policies, which can be updated on a regular interval that is a control $\mathbf{u}$ of the form (6.1). We compare the category-dependent policies with the category-free policies devised in section 6.3.

6.4.4 Category-dependent fixed policy

In this section, we find control $u_h(t)$ for different destination categories that remain constant for the time period $T_u = T$, that is there is no update of policies in the time interval $[0, T]$. Therefore, $p = 1$ in (6.1) and hence now the the searcjh space reduces to $[0, 1]^q$ instead of $[0, 1]^{pq}$. Recall that since the number of categories is 5, hence $q = 5$.

One of the simplest and naive method to find such control by brute force. A suboptimal solution can be found by considering a finite number of points in $[0, 1]^5$ which means exploring a grid of points and running (6.3)-(6.9) for different values of $\mathbf{u}(k) = [u_1(k), \ldots, u_5(k)]$, where $u_h(k) \in [0, 1]$ and finding the $\mathbf{u}^*$ by picking the $\mathbf{u}$ which is feasible and gives the maximum socio-economic cost $L(\mathbf{u})$. In our simulation we consider $u_h(k) \in \{0.4, 0.6, 0.8\}$ for all $h \in \{1, \ldots, 5\}$ hence reducing the computational complexity as we need to run (6.3)-(6.9) 5^3 times only. This particular choice of control inputs not only reduces the total number of iteration to finite number but also prevents aggressive closing or opening of any particular destination category. Figure 6.11 depicts category-dependent fixed control policies computed using brute-force technique mentioned above and using Algorithm 4 with the initial polyblock vertex $\mathbf{1} \in \mathbb{R}^5$, maximum number of iteration $n_{\text{poly}} = 150$ and the threshold $\epsilon = 0.03$(one percent of economic cost in benchmark case), $\bar{d} = 0.01, n_{\text{it}} = 20, \xi = \bar{I}/100$ for $T = 15$weeks. Figures 6.11c and 6.11d compares the relative cost R_E and infection spread for these two cases with that of category-free fixed policy computed in section 6.3.1. First of all, notice in Figure 6.11d that infection is below the desired threshold in all the cases. Moreover, Figure 6.11c shows that category-dependent policies indeed gives a higher socio-economic cost in comparison to the category-free case.

Notice in Figures 6.11a and 6.11b that there is a pattern between capacity control u_h and

Algorithm 4: Category-dependent policy based on polyblock outer approximation method

Input: parameters for (6.3)-(6.9), threshold $\bar{I}$, initial time T_0, final time $T_f, I_{\max}(.)$, initial condition $N_i(T_0), I_i(T_0), S_i(T_0), R_i(T_0)\ \forall i$, tolerance $\epsilon, \bar{d}$, sufficiently large maximum number of iterations n_{poly}, no. of updates p_m in the time interval $[T_0, T_f]$, and projection tolerance ξ and projection maximum iterations n_{it} (for Algorithm 5)

Output: $\epsilon-$optimal solution $\mathbf{u}^*$

Initialize the vertex set $\mathcal{T}_1 = \mathbf{1}_{p_m q}$. Set the Current Best Value$(CBV_0) = 0$.

Set $k = 1$.

REPEAT

Select $\mathbf{z}^k = \text{argmax}\{L(\mathbf{z}) | \mathbf{z} \in \mathcal{T}_k\}$, where $L(\mathbf{z})$ is computed by running (6.3)-(6.9) for time for $k = T_0/\Delta t, \ldots, T_f/\Delta t$.

Compute the minimum distance d as in (6.20)

IF $d < \bar{d}$

Compute $\mathbf{b}'$ as in (6.21)

IF $\mathbf{b}' \in G$

Compute the projection $\mathbf{x}^k = \pi_G(\mathbf{z}^k)$ using Algorithm 5.

ELSE

Compute the projection $\mathbf{x}^k = \pi_G(\mathbf{b}')$ using Algorithm 5.

END IF

ELSE

Compute the projection $\mathbf{x}^k = \pi_G(\mathbf{z}^k)$ using Algorithm 5.

END IF

IF $\mathbf{x}^k = \mathbf{z}^k$,

Set current best feasible solution $\bar{\mathbf{x}}^k = \mathbf{x}^k$

$CBV_k = L(\mathbf{z}^k)$

ELSE

IF $L(x) \geq CBV_{k-1}$

$\bar{\mathbf{x}}^k = \pi_G(\mathbf{z}^k)$ and $CBV_k = L(\pi_G(\mathbf{z}^k))$

ELSE

$\bar{\mathbf{x}}^k = \bar{\mathbf{x}}^{k-1}$ and $CBV_k = CBV_{k-1}$.

END IF

END IF

IF $|L(\mathbf{z}_k) - CBV_k| \leq \epsilon$,

GO TO Step 13 and EXIT

ELSE

– Create the new vertices $\mathbf{v}^j s$ from $\mathbf{z}_k$ and x by following the rule:

$$\mathbf{v}^{k,i} = \mathbf{z}^k + ((\mathbf{x})_i - (\mathbf{z}^k)_i)e_i$$

for all i such that $\mathbf{z}_i^k > \mathbf{x}_i^k$.

– $\mathcal{T}_{k+1} = \mathcal{T}_k \setminus \{\mathbf{z}^k\} \cup \mathbf{v}^{k,i}$, for all i, where $\mathbf{v}^{k,i}$ are the vectors computed in the previous step.

$-$Remove all the improper vertices from $\mathcal{T}_{k+1}$ and the vertices w such that $L(\mathbf{w}) \leq CBV_k + \epsilon$.

End IF

$k = k + 1$

Until: $\mathcal{T}_k = \emptyset$ OR n_{poly} iterations.

$\mathbf{u}^* = \bar{\mathbf{x}}^k$

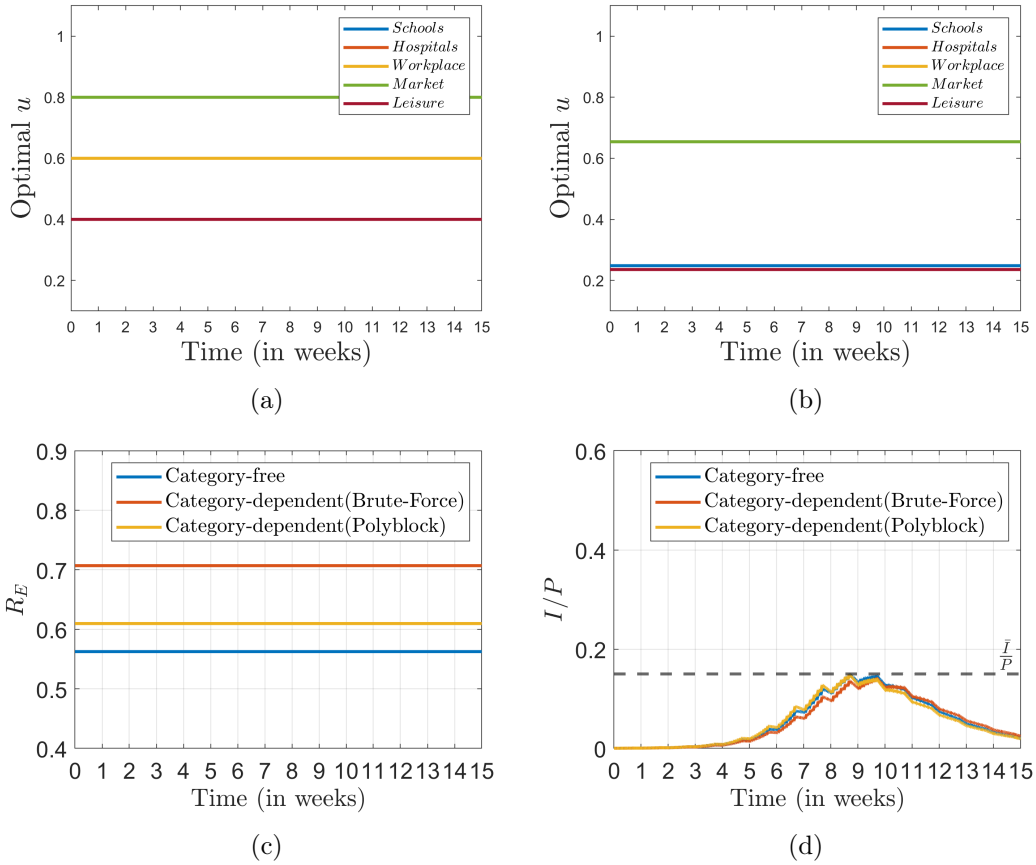


Figure 6.11: Fixed control policy (a) category-dependent optimal control $\mathbf{u}^*(k)$ computed using brute-force technique.(b) category-dependent optimal control $\mathbf{u}^*(k)$ computed using polyblock Algorithm 4. (c) Relative economic activity $R_E(k)$, and (d) proportion of infected people (I/P), when brute-force technique and polyblock Algorithm 4 is applied with $T_u = T = 15$ weeks. (c) and (d) also depicts R_E and (I/P) in case of category-free fixed policy.

Algorithm 5: Bisection search to compute the projection $\mathbf{x} = \pi_G(\mathbf{z})$ in Step 5 of Algorithm 4

Input: $\mathbf{z}$, parameters for (6.3)-(6.9), threshold $\bar{I}$, initial time T_0, final time T_f, initial condition $N_i(T_0), I_i(T_0), S_i(T_0), R_i(T_0)\ \forall i, I_{\max}(.)$, maximum number of iterations n_{it} and tolerance $\xi > 0$.

Output: $\mathbf{x} = a^*\mathbf{z}$ such that $a^* = \text{argmax}\{a^* > 0 | I_{\max}(a^*\mathbf{z}) \leq \bar{I}\}$.

Set $a_{\min} = 0$ and $a_{\max} = 1$.
REPEAT
  Compute $\bar{a} = (a_{\min} + a_{\max})/2$,
  Run (6.3)-(6.9) with $\mathbf{u}(k) = \bar{a}\mathbf{z}$ for $k = T_0/\Delta t, \ldots, T_f/\Delta t$ and compute $I_{\max}(\bar{a}\mathbf{z})$
  IF $I_{\max}(\bar{a}\mathbf{z}) \leq \bar{I}$
    $a_{\min} = \bar{a}$
  ELSE, $a_{\max} = \bar{a}$.
till $(\bar{I} - I_{\max}(\bar{a}\mathbf{z})) \leq \xi$ or n_{it} iterations.
Set $a^* = a_{\min}$
Return $\mathbf{x} = a^*\mathbf{z}$

the weights χ_h (low u_h with low χ_h and high u_h with high χ_h). This is consistent with the intuition that one allows for a stronger closure on low-weight categories, and lighter closure on high-weight categories, thus leading to a higher L, while still ensuring that $I_{\max}$ remains in the desirable range.

Moreover, it seems from Figure 6.11c that brute force technique performs the best among all three but it is important to note that the $\mathbf{u}_h^*$ which we obtain from the Algorithm 4 is not optimal as the algorithm terminated not because of convergence but because of the maximum number of iterations. At the end of $k = 150$ iterations, there was still a significant difference between the upper bound and the lower bound of $L(\mathbf{u}^*)$. Precisely, for $k = 150$, $L(\mathbf{z}^k) - L(\mathbf{x}^k) = 0.8537$ which is very large than the threshold ϵ. Notice that category-free fixed policy retains about 56 percent of the socio-economic cost of that in the benchmark case where as category-dependent policies retain 61 percent and 71 percent using brute-force and Algorithm 4 respectively. Therefore, category-dependent policies are recommended. Moreover, the fact, that brute-force technique computes a policy which is able to retain about 71 percent of the socio-economic cost while keeping the epidemic bounded below the threshold, is an indicator that we can even do better. We noticed in section 6.3 that fixed policies apply tighter restriction from the beginning in order to control the epidemic and hence a policy which is updated regularly was recommended. Therefore, in the next section, we find a category-dependent policy which can be updated on a regular time interval depending on the epidemic spread.

6.4.5 Category-dependent receding-horizon policy

For that, in principle, the problem (6.14) can also be solved directly using Algorithm 4 by taking $p_m = \lceil T/T_u \rceil = p$, and $T_0 = 0, T_f = T$, which corresponds to an optimization problem with pq variables. However, the convergence of Algorithm 4 can be very slow [Cho+19]. Therefore, to find a good, although possibly suboptimal, feasible solution of (6.14) in an efficient way, we adopt a receding horizon approach to obtain policies in which different control is applied on different destination categories, which can be updated on a regular interval such that we obtain a control $\mathbf{u}$ of the form (6.1). We compare the category-dependent policies with the category-free policies devised in section 6.3.

Algorithm 6: Category-dependent receding-horizon policy, with update period T_u and optimization horizon length T_m.

Input: Parameters for (6.3)-(6.9), initialization (6.10), $I_{\max}(.)$, threshold $\bar{I}$, maximum number of iterations $n_{\text{it}}, n_{\text{poly}}$, and tolerance $\epsilon, \bar{d} > 0, T, T_u$.

Output: $\mathbf{u}^* = \{\boldsymbol{\mu}^{*1}, \ldots, \boldsymbol{\mu}^{*p}\}$, where $\boldsymbol{\mu}^{*\ell} = \{\mu_h^\ell\}$ for all $h \in \{1, \ldots, q\}$.

1 Initialize $\ell = 1$. Compute $p = \lceil T/T_u \rceil$.
2 **REPEAT**
3 Compute $T_0 = (\ell-1)T_u$ and $T_f = (\ell-1)T_u + T_m$
4 Apply Algorithm 4 with $p_m = 1$ over $[T_0, T_f]$ and threshold ϵ, bisection tolerance ξ and bisection maximum iterations n_{it}, and denote by $\boldsymbol{\mu}^{*\ell}$ the obtained output.
5 Run (6.3)-(6.9) over the interval $[(\ell-1)T_u, \ell T_u]$, with control $\boldsymbol{\mu}^{*\ell}$, so as to obtain $N_i(\ell T_u/\Delta t)$, $S_i(\ell T_u/\Delta t)$, $I_i(\ell T_u/\Delta t)$, $R_i(\ell T_u/\Delta t)$ for all i (to be used as initialization for the next iteration).
6 $\ell = \ell + 1$
7 until $\ell = p$.

In this receding horizon approach, over each interval of optimization we choose to have a constant policy over time but allow different policies for different destination categories, which corresponds to an optimization problem with q variables. In other words, we apply Algorithm 4 on the receding time intervals $[T_0, T_f]$ with the choice $p_m = 1$, where $T_0 = \ell T_u$ and $T_f = \ell T_u + T_m$ and $\ell = 0, \ldots, p-1$, as described in Algorithm 6 and as depicted in Figure 6.4. Here, the length of the optimization horizon is a parameter $T_m \in [T_u, T]$ to be designed.

6.4.6 Simulations

Algorithm 6 is similar in spirit to Algorithm 2: both have optimization over receding intervals of length T_m, where for simplicity optimization only explores policies which are constant in such time interval, and re-optimize every T_u weeks, while they differ in the fact that Algorithm 6 exploits the opportunity to apply different controls to different categories. In Section 6.3.2, it has been seen that the approach in Algorithm 2 gives the best results when $T_u = 1$ and $T_m = 1$. For this reason, in this section we perform simulations using Algorithm 6

with $T_m = 1$ and $T_u = 1$, to obtain results with this category-dependent policy, and we compare them with the results obtained in Section 6.3 with a category-free policy and same parameters T_u and T_m.

For the large-scale network of Grenoble and with horizon of interest $T = 15$ weeks, Figure 6.12 compares the effect of category-dependent control policy, computed using algorithm 6, with $\epsilon = 0.002$, $n_{\text{poly}} = 150, n_{\text{it}} = 20$, and $\bar{d} = 0.01$ for $T_m = 1$ and $T_u = 1$ with that of the category-free control policy computed using algorithm 2 with $\epsilon_f = \bar{I}/100$ and $n_{\text{iter}} = 20$ in section 6.3.

While running Algorithm 6 in our simulation, in most weeks (i.e., in most iterations of Algorithm 6) Algorithm 4 terminated by having reached a distance between lower and upper bound that is below the threshold ϵ. Only in weeks 4, 5 and 6 Algorithm 4 reached the maximum number of iterations n_{poly} and hence gives a suboptimal result. Recall that even in this case the obtained control policy is feasible. Despite this suboptimality, we can see in Figure 6.12c that the category-dependent policy obtained with Algorithm 6 significantly outperforms the category-free policy, thus confirming the interest of allowing for category-dependent controls. Figure 6.12b and Figure 6.12a show the piecewise-constant function $\mathbf{u}^*(k)$ obtained from Algorithm 6, and piecewise-constant function $u^*(k)$ obtained form Algorithm 2 while Figure 6.12c and Figure 6.12d show the corresponding relative economic activity and proportion of infected people.

We can see that both category-free and category-dependent policies start with the same control $\mathbf{1}_5$, since they are computed considering the same initial epidemics situation with few infections, and hence they both foresee that full opening is possible, without causing too many infections over the short optimization horizon $T_m = 1$. However, from the second week, some partial closure becomes necessary in order to keep the infections below the given threshold. Here, the difference between the category-free and the category-dependent approach becomes apparent: by using a smaller control (i.e., heavier restrictions) on the categories with lower socio-economic weight χ_h, the category-dependent approach allows a higher control (i.e., less restrictions) on the categories with higher socio-economic weight, whose contribution has a major impact on the socio-economic cost, while still ensuring that infections remain below the threshold. For this reason, the category-dependent approach can outperform the category-free one; indeed, Figure 6.12c shows that, after 15 weeks, category-dependent policy is able to retain about 92 percent of the socio-economic of the Benchmark case, in comparison to 85 percent in case of category-free policy.

From these simulations, we can draw the recommendations to design category-dependent policies using Algorithm 6, whenever the weights χ_h (which reflect socio-political priorities) are quite unbalanced.

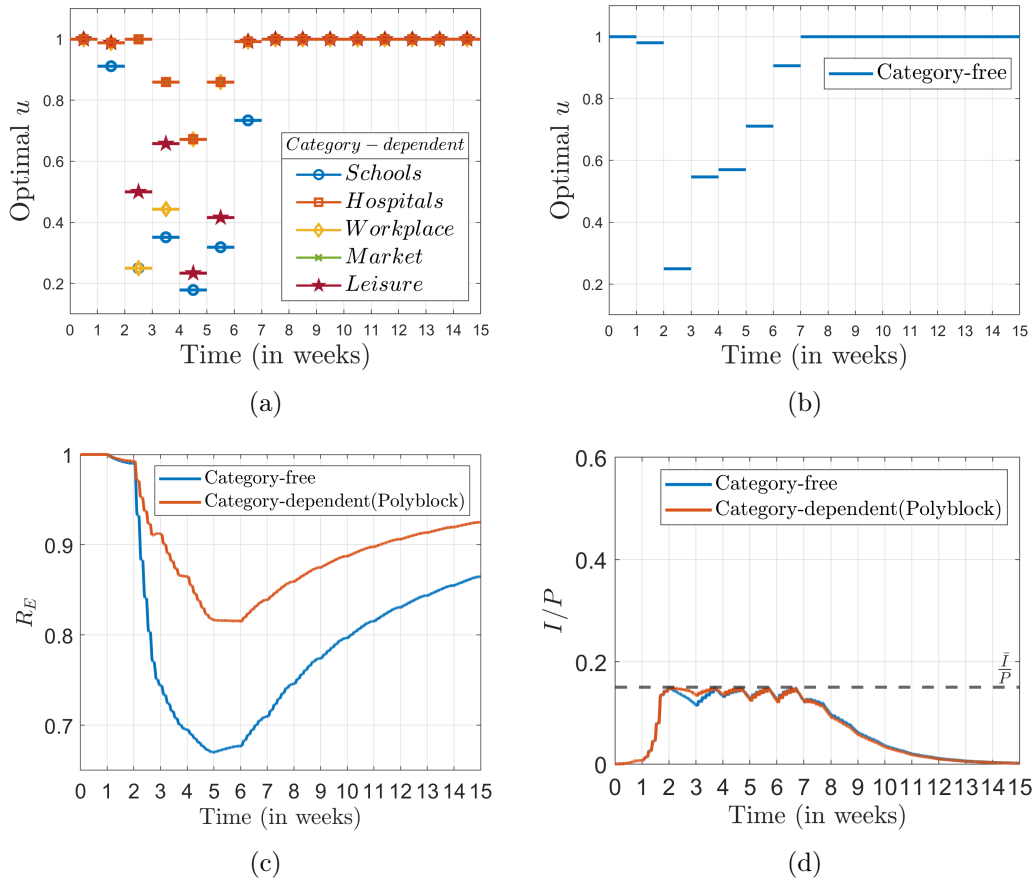


Figure 6.12: Receding-horizon policy with optimization horizon $T_m = 1$ and $T_u = 1$: (a) Category-dependent control $\mathbf{u}^*(k)$ obtained with Algorithm 6, (b) Category-free control $u^*(k)$, obtained with Algorithm 2.(c) Relative economic activity $R_E(k)$, and (d) proportion of infected people (I/P), when Algorithm 6 and Algorithm 2 are applied with $T_u = 1$ week and $T_m = 1$ weeks.

6.5 Concluding remarks

In this chapter, we have designed mobility control policies to mitigate epidemics in a city-wide network.

We have designed policies for capacity control at destinations, so as to maximize the economic activity under the constraint that the number of infections remains below a desired threshold, so as to avoid saturating health-care system.

At first, we proposed scalable algorithms to design such policies, by focusing on the case where the same control is applied to all destination categories, and either a same fixed control is applied over all the time window of interest, or an MPC approach is used, re-optimizing over receding time windows.

Then, we addressed the more general optimization problem, where different controls can be applied to different destination categories and at different times. Our solution technique uses polyblock outer approximation method with some enhancements along with an MPC approach. We implemented these techniques on large-scale network of Grenoble metropolis.

Chapter 7

GTL-Healthmob simulation platform

Contents

7.1 Platform Overview 124

7.2 User Interface 125

7.2.1 Unrestricted mobility 126

7.2.1.1 Result visualization 128

7.2.2 Manual restricted mobility: 129

7.2.2.1 Result visualization: 129

7.2.3 Optimal restricted mobility 130

7.2.3.1 Result Visualization 133

7.3 Concluding remarks 133

In this chapter, we present and describe the GTL-healthmob online simulation platform, available through the link[1]. It has been developed within the DANCE[2] team in the context of the ERC project Scale-FreeBack and the INRIA project Healthy-Mobility, on the basis of work done in Chapters 4,5 and 6. It is a platform where users can

- Simulate different scenarios of epidemic spread,
- Simulate different scenarios of mobility restrictions and visualize its effect on the epidemic spread.
- Simulate the optimal mobility restriction aimed at limiting the epidemic spread and maximizing the economy.

This platform is part of the Grenoble Traffic Lab (GTL), which considers the Grenoble metropolis as a case study.

[1] `http://gtlville.inrialpes.fr/covid-19`
[2] `https://team.inria.fr/dance/`

7.1 Platform Overview

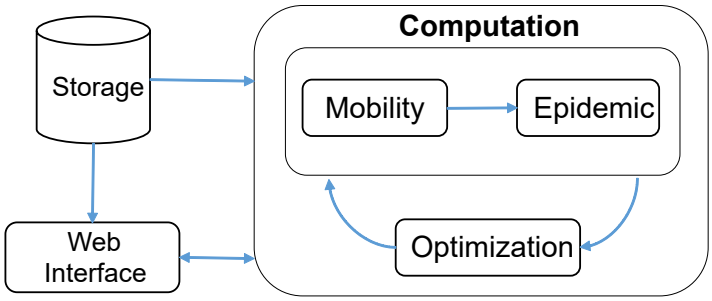


Figure 7.1: Modules and submodules of the GTL-Healthmob

Figure 7.1 shows the basic structure of the GTL-Healthmob platform. It comprises of the following modules:

- The **Storage** modules deals with the database and the transmission of static data to the Computation and the Web Interface modules. The static data mainly comprises of the location of the origin and destinations nodes, Population of communes and capacities of places of interests called destinations, OD matrices and time-dependent functions described in Chapter 5
- The **Web Interface** module allows to visualise the mobility network, origin and destination nodes and also facilitates the users to choose input parameters and visualise the computed results.
- The **Computation** module takes the parameters given as inputs by the users on the Web Interface module and static data from the Storage module to simulate the mobility and epidemic models and compute the evolution of mobility and epidemic spread using the models presented in Chapter 4 using the model parameters described in Chapter 5. The results is then passed on to the Web interface for visualization. It is composed of three submodules:
  - Mobility - Using different control and input parameters provided by the user, it computes the number of people present at each location at different times for each day during the simulation period using the discrete-time mobility model (4.12)-(4.18) in Chapter 4.
  - Epidemic - Using the epidemic parameters provided by the user and the number of people present at any location, given by mobility submodule, it computes the evolution of epidemic spread at each location using the discrete-time SIR-mobility model (4.22) described in Chapter 4.
  - Optimization - Using the results from mobility and epidemic submodules, it computes the optimal mobility restriction that keeps maximum infection below the user defined threshold. It uses the techniques developed in Chapter 6.

7.2 User Interface

In this section, we describe the interfaces and functionality of the web platform available to the general public. An user after clicking on the link `http://gtlville.inrialpes.fr/covid-19` lands on the welcome page of the GTL-Healthmob as shown in Figure 7.2. In the right panel, the map of Grenoble metropolis is shown with different buttons to visualise the network:

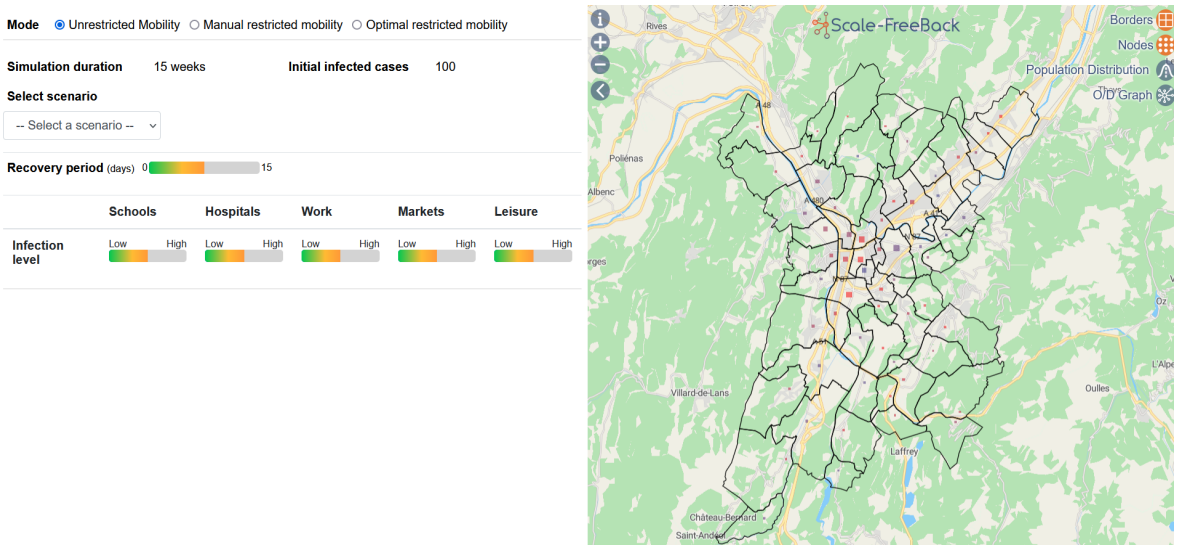


Figure 7.2: Welcome page of the GTL-Healthmob platform. In the left panel, there are different modes and parameters which an user can set to simulate different scenarios. In the right panel, users can visualise mobility-network related information and the heatmap animations for infection and mobility after the simulation.

- **Nodes :** It is used to visualize the origins and destinations and their classification into categories and subcategories. These are the same categories and subcategories of destinations in which we classified them in Section 5.1.2 of Chapter 5. A user can select a category or a subcategory to see the destinations in them. Information like population or capacity of can be seen by hovering over it as in Figure 7.3. The size of the node varies corresponding to its population or capacity. The 'aggregate' button aggregates the nodes in some subcategories as done in Section 5.1.2.8 of Chapter 5. The names of subcategories whose nodes have been aggregated appear in italics. There are around 4000 destination nodes grouped in 5 categories and 15 subcategories but after aggregation, we have 374 destination nodes. For simulations, these aggregated nodes are used.

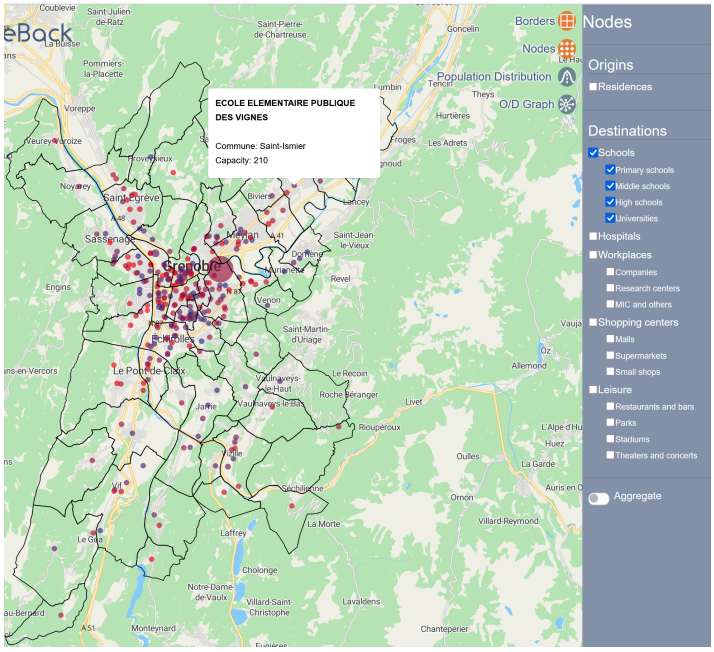


Figure 7.3: Node menu in GTL-Healthmob platform.

- **Population distribution :** Shows the population distribution using a heatmap when origin is selected and distribution of capacities when destination subcategory is selected. It uses a Gaussian like curve to show the gradient. See Figure 7.4a.
- **O/D Graph :** Shows the OD connections between different origins and the aggregated destination nodes which were established in Section 5.1.3. User can visualize the OD connections for different destination categories and subcategories. For example, Figure 7.4b shows the connections for schools.

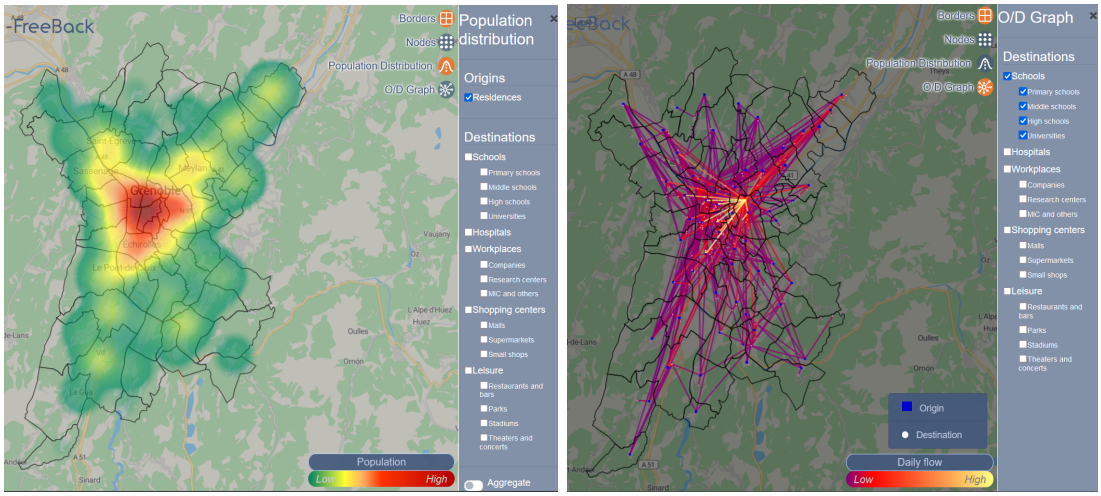
In the left panel of the welcome page, the user has different options meant for simulations. The options are :

1. Unrestricted mobility
2. Manual restricted mobility
3. Optimal restricted mobility

On running the simulations, the user can visualise the results with the help of plots and heatmap animations which will be discussed in upcoming sections.

7.2.1 Unrestricted mobility

In this mode, user can visualise, how an epidemic, with different levels of infection rate and recovery rate, evolves in the Grenoble metropolis. The user has three epidemic scenarios



(a) (b)

Figure 7.4: (a) Population distribution. (b) Mobility network for schools. Origins are denoted in blue color and the destination in white color. The color of the connecting edge ranges from magenta to yellow depending on the amount of average daily flow along them.

to choose from. For all these scenarios, mobility is unrestricted but they differ in terms of infection level in various destination categories. The Parameters in the top left as can be seen in Figure 7.2 are:

- Simulation duration: The number of weeks the simulation runs. In this mode, it is fixed to be 15 weeks.

- Initial infected cases : The number of people initially infected. This total number of infected cases have been distributed among the origins using the following rule: We create a set of intervals for each origin in proportion to their population. For every infected person, we create a random number in the interval $[0, 1]$ and assign this person to the origin corresponding to the interval in which this random number belongs to. The same rule has been used in every mode.

- Select scenario : Choose the epidemic scenario. These scenarios mimic the behavior of different Covid-19 variants.

- Recovery period : This represents the number of days required for a person to recover if he is infected. It depends on the epidemic scenario.

- Infection level : This represents the rate at which an infected person might infect others. It ranges from low to high in different categories depending on the epidemic scenario.

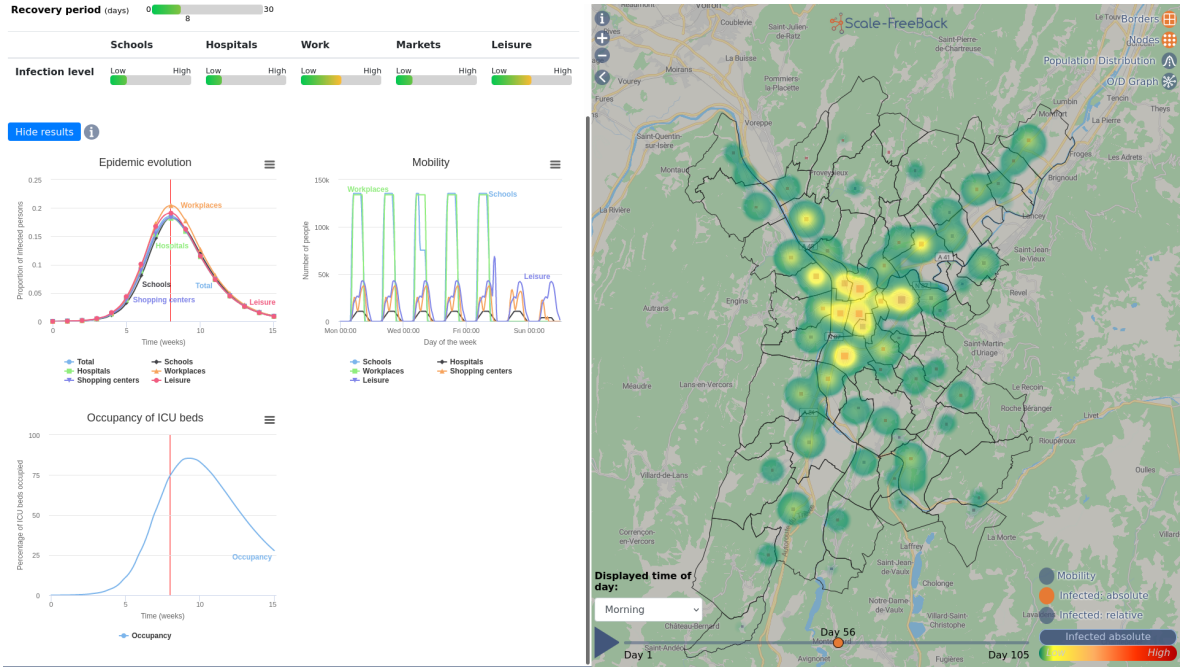


Figure 7.5: Unrestricted mobility: plots showing evolution of epidemic, mobility patterns and ICU occupancy and the right panel shows the heatmap for infection in absolute numbers at different locations on the day of infection peak during morning. Here, the selected scenario is similar to that of Covid19.

7.2.1.1 Result visualization

When the run button is clicked, three plots and heatmaps are shown. The simulation uses the discrete-time SIR-mobility model presented in Chapter 4. In the left panel, user can see the following plots as can be seen in Figure 7.5.

- Epidemic evolution : Weekly average of proportion of total infected people $I(t)$ and proportion of infected persons in different categories.
- Mobility : Number of people in each category during a week.
- ICU bed occupancy : Shows the percentage of ICU beds occupied in the hospitals of Grenoble metropolis. The ICU occupancy is given by following system: Given the number of total infected persons $I(t)$, ICU occupancy $I_{ICU}(t)$ can be characterized as

$$\dot{I}_{ICU}(t) = \frac{1}{\gamma_{ICU}} * p_{ICU} * I(t) - \frac{1}{\mu} I_{ICU}(t),$$

where

  - $I_{ICU}(0) = 0$,
  - γ_{ICU} = number of days after infection to get admitted to ICU,

- p_{ICU} = proportion of infected who get admitted to ICU and
- μ_{ICU} = number of days to recover from ICU.

Apart from the plots, user can choose to see the heatmap animations for mobility and infection evolution. For infections, heatmap is shown in absolute numbers and relative as well. The mobility heatmap shows the movement of people around the area and the infection heatmap shows the evolution of infection in the area where the gradient is shown using Gaussian like curves at each locations.

7.2.2 Manual restricted mobility:

In this mode, the user can choose to simulate mobility and epidemic scenario with parameters and controls of his choice. It simulates the models (4.12)-(4.18) in Chapter 4 and (4.22) with epidemic and control parameters chosen by the user as can be seen in Figure 7.6. The parameters are:

- Simulation duration : The duration in weeks for which to run the simulation.
- Recovery period : Number of days, a person takes to recover after getting infected. It corresponds to the parameter γ_i in (4.22).
- Infection probability : The probability by which an infected person would infect a susceptible person.
- Initial infected cases : Initial number of infected persons.
- Allowed capacity : With its value in the interval $[0, 1]$ for each category, user can impose restriction on mobility by restricting the capacity of destinations in different categories.
- Closing hours : With their value in the range $[6, 24]$ for each destination category, user can simulate partial or full closure of the destinations by forcing them to close before the nominal closing hours.
- Contact rate : Average number of contacts people make in different categories. The product of infection probability and contact rate gives the nominal infection rate $\bar{\beta}_i$ at the destination in model (4.22).

7.2.2.1 Result visualization:

The results are visualised using the same plots and heatmaps described in Section 7.2.1.1. The simulation uses the discrete-time SIR-mobility model presented in Chapter 4. Plots and heatmap can be seen in Figure 7.6 for a simulation done for 15 weeks, when schools are running at half their capacities.

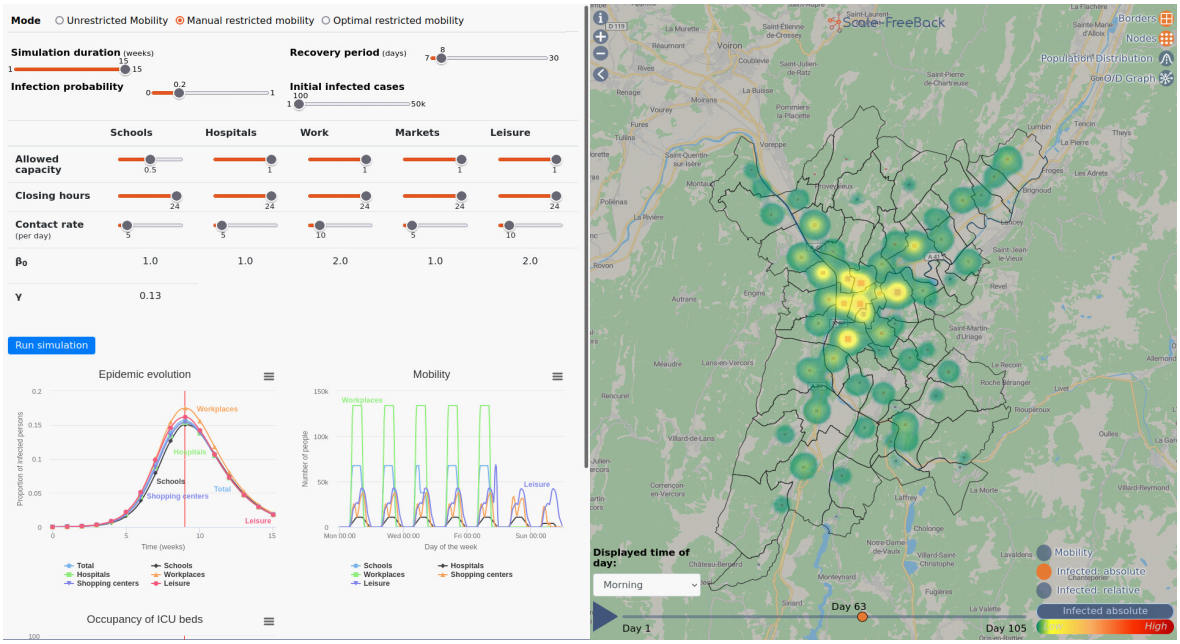


Figure 7.6: Interface with manual restricted mobility mode. Here, we can see different parameters and control which an user can select for a simulation. It shows the results of a simulation when Schools are running at half their capacities.

7.2.3 Optimal restricted mobility

This mode takes into account the implementation of the optimization problem solved in Chapter 6 for the Grenoble area. Here, the user can visualise optimal mobility restriction policies depending on epidemic status during the simulation period. This mode showcases the results obtained in Section 6.3.2 and Section 6.4.5. All the parameters in this mode are fixed and are just shown in the left panel. The parameters are simulation duration, initial infected, recovery period, infection level(nominal infection rate), socio-economic importanceχ_h for all categories h and infection threshold $\bar{I}$ defined in Chapter 6. The rest of the parameters have been explained in previous sections. The right panel showing the map remains the same showing the same set of heatmaps as in previous two modes. In this mode, the user can choose to see the effect of either the category-free optimal policy or category-dependent optimal policy as in Chapter 6.

- Category-free : Here, the same capacity control is applied to all the destinations irrespective of the category while maximising the economy and keeping the infection below a threshold. It is given by the Algorithm 2 in Chapter 6.

- Category-dependent : Here, different capacity controls are applied to different the destinations in destination categories. It is given by Algorithm 6 in Chapter 6.

Figure 7.7: Left panel of the Web Interface, when the category-free in “optimal restricted mobility" is selected.

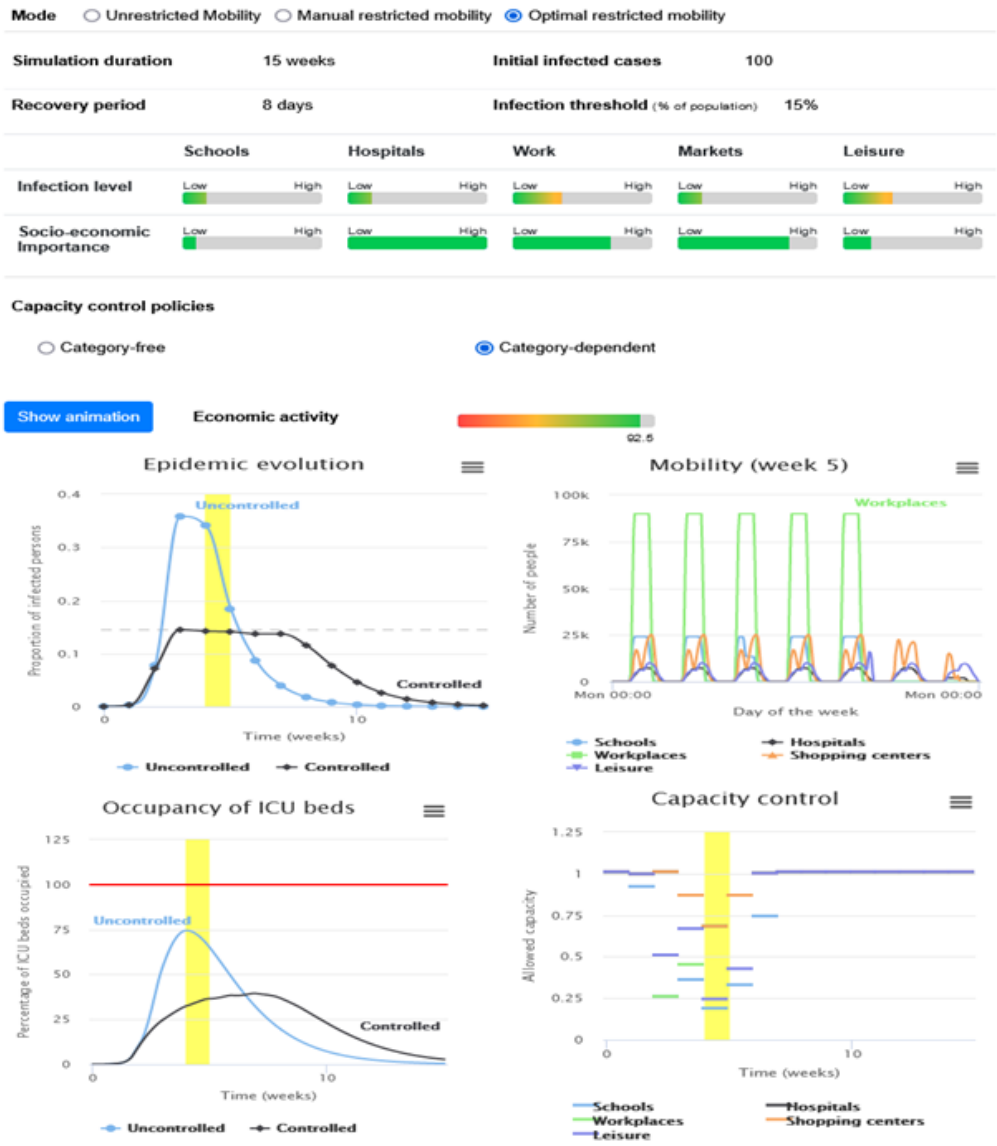


Figure 7.8: Left panel of the Web Interface, when the category dependent option in “optimal restricted mobility" is selected.

7.2.3.1 Result Visualization

The results are visualized through the plots as shown in Figure 7.7:

- Capacity control : It shows the optimal capacity control inputs to be applied during different weeks for simulation to keep the infection in check and to protect the economy.
- Infection evolution : It shows the comparison between infections in uncontrolled case and the case when the optimal capacity control is applied.
- Mobility : It shows the number of people in different categories during the week. It is shown by an animation as the mobility pattern is subject to change with respect change in the capacity control. Since it changes every week, the mobility for a particular week can be seen by selecting that week in capacity control plot.
- Economic activity : It is depicted by a bar, which shows, in terms of percentage, how much economic activity under optimal mobility policy was done with respect to that in case of uncontrolled mobility.
- The animation for heatmap for mobility, absolute infection, and relative infection is shown as in the previous cases.

Figures 7.7 and 7.8 show the left panel of optimal restricted mobility mode when category-free and category-dependent options are selected, respectively. Notice that the Mobility plot indicates the number of people in different categories corresponding to the chosen week in other plots.

7.3 Concluding remarks

In this chapter, we presented the main framework of the GTL-Healthmob simulation platform and functionality of its different components. The main objective of this platform is to facilitate other users to simulate different scenarios of epidemic spread or mobility restrictions to visualise the mobility patterns and its effect on the epidemic spread in Grenoble metropolis. Using it, an user can also get an overview of the current situation during an epidemic. Moreover, the platform uses different visualization techniques to make even a layman user to understand the trends of the mobility patterns and epidemic spread. It can also serve as a testbench for validating upcoming results.

Conclusion

Resilience can have different aspects, and in this thesis, we attempted to deal with two different aspects of resilience in complex network systems. In the first part, we provide resilience in the process of average state estimation in an engineered network system with an outlier. In the second part, we suggest control policies in order to make the society resilient in the events of an epidemic outbreak by providing optimal mobility restriction policies. Now, we summarize the work done in this thesis.

Average state estimation in network systems with outliers

Average state estimation in a network system with a few sensor measurements can give unexpected results if there is an outlier that is unmeasured. Since the outlier is so different from other states, it drives the average very far from the actual average. Instead, an average excluding the outlier will be near the actual average. Therefore, we provided a methodology to detect the outlier and estimate the average excluding the outlier(trimmed-average). This methodology uses a bank of scalar and tunable observers to estimate the average, excluding the states at all possible positions. In this regard, we provided an explicit design of the observer and a necessary and sufficient condition under which a bank of such an observer can be designed. Then the method uses a dissimilarity criterion in order to differentiate between the estimates obtained from the observers. This method has been applied to two different cases, one when the system model is known and the other when the system model is partially unknown when the outlier arises due to a localized fault. The advantage is that when the outlier is a result of the considered localized fault, the design of the observers doesn't require the faulty system model but only the system without the fault.

Future perspectives include the cases of multiple outliers and sequential methods such as group testing for detection. The sequential methods can be useful in reducing the number of observers required. Moreover, it would be interesting to investigate if an outlier creates other outliers and, if yes, how does it propagate in different network structures.

Modeling and control of mobility for epidemic mitigation in large-scale urban networks

Human mobility is an essential component of everyday life. It drives the economy of a city or a country. However, it can also facilitate disease spread if an epidemic outbreak occurs. A full closure or a total lockdown can be detrimental to the economy, whereas unrestricted mobility can harm people's well-being. Therefore, policymakers need to have efficient strategies for controlled mobility. To design such policies, they need a model that captures the daily

movement of people between different locations. Therefore, in Chapter 4, we considered a supply-demand-based mobility model in continuous-time, which captures the daily movement of people between residences and places of interest called destinations in a network, using time schedules and gating profiles. This model also accommodates the possibility of imposing restrictions on mobility which, when integrated with an epidemic spread model, can be utilized for epidemic mitigation. This restriction is applied to the capacities of destinations, which is the maximum number of people that can gather there at one time. Moreover, if the goal of mobility control is epidemic mitigation, we need a model which integrates mobility and epidemic spread. For this, we considered a model which integrates epidemic spread using SIR dynamics with a supply-demand mobility model.

These models are given in continuous-time in which the flows are defined such that the model has nice properties of non-negativity, boundedness, and mass conservation. However, for their implementation in large-scale networks, the models need to be discretized. Therefore, we analyzed the problems encountered in the discretization of these models and redefined the flows in discrete-time, which preserves the properties of non-negativity, boundedness, and mass conservation. We also provide a discretization of the SIR-mobility model, which integrates the mobility and epidemic spread and preserves the nice properties of boundedness and non-negativity.

In Chapter 5, we provide parameters of the discrete-time mobility model by using available public information. The first building block of a mobility network is the location between which mobility happens. So, we provided methodologies to calibrate the model. We do it for Grenoble metropolis, but they can be applied to different cities with some modifications. At first, we collected map data and located different origins and destinations. We classify the destinations in categories and further into subcategories. We provide techniques to obtain the capacities of different destinations. We collect various data for different time-dependent profiles corresponding to destinations in different categories. Then, we provide methodologies to establish connections between an origin and destination based on the minimum road distance between them. After that, We propose methods to compute the daily number of people who move between an origin and destination. Finally, we use these techniques and methodologies to build the large-scale network of the Grenoble metropolis.

Next, in Chapter 6 we designed policies for capacity control at destinations to maximize the economic activity under the constraint that the number of infections remains below a desired threshold to avoid saturating the healthcare system.

At first, we propose scalable algorithms to design such policies by focusing on the case where the same control is applied to all destination categories, and either the same fixed control is applied over all the time window of interest, or an MPC approach is used, re-optimizing over receding time windows. Then, address the more general optimization problem, where different controls can be applied to different destination categories and at different times using polyblock outer approximation with some enhancements along with an MPC approach. We implement these techniques on the large-scale network of Grenoble metropolis.

A simulation platform GTL-Healthmob has also been developed in our team to showcase

this thesis's results, simulate different mobility restriction scenarios, and visualize its effect on an epidemic spread with application to the Grenoble metropolis.

There can be several perspectives on the work done in this thesis. First of all, recall that the mobility model is modular, so it can be used in different applications such as urban planning and traffic planning. Moreover, it can be coupled with various epidemic spread models to model complex disease spread behaviors.

Another possible application of the mobility model is to describe and predict the space-time distribution of the charge of electric vehicles. This can be used to optimize charging station locations and identify the locations where the power networks might be most vulnerable once the use of electric vehicles becomes widespread.

Here, we considered the possibility of mobility restriction only by tuning the capacities of destinations. Another natural possibility is to consider the destination closing times as control and obtain optimal mobility policies for epidemic mitigation.

Finally, another perspective is to analyze the numerical aspect of the optimization done in Chapter 6. Moreover, devising specific algorithms which are faster in convergence can be an extension of the work done in this thesis.

Bibliography

[AD03] Julien Arino and P. van den Driessche. "A multi-city epidemic model." In: *Mathematical Population Studies* 10.3 (2003), pp. 175–193 (cit. on p. 42).

[AGa] Academie-Grenoble. *Eligible high schools according address in Grenoble and Saint-Martin D'Heres*. `https://bv.ac-grenoble.fr/secto/rec`. accessed 2 february 2022 (cit. on p. 78).

[AGb] Academie-Grenoble. *Search engine for list of primary schools in region Grenoblois per commune*. `https://bv.ac-grenoble.fr/carteforpub/ecole`. accessed 6 December 2021 (cit. on p. 65).

[AGc] Academie-Grenoble. *Search engine for list of primary schools in region Grenoblois per commune*. `https://bv.ac-grenoble.fr/carteforpub/etab`. accessed 6 December 2021 (cit. on p. 65).

[Agg17] Charu C. Aggarwal. *Outlier Analysis*. Vol. 2. Springer International Publishing, 2017, pp. xii,466 (cit. on p. 8).

[Ale+15] Lauren Alexander et al. "Origin–destination trips by purpose and time of day inferred from mobile phone data." In: *Transportation Research Part C: Emerging Technologies* 58 (2015). Big Data in Transportation and Traffic Engineering, pp. 240–250 (cit. on p. 40).

[Ari09] Julien Arino. "Diseases in Metapopulations." In: *Modeling and Dynamics of Infectious Diseases*. World Scientific, 2009, pp. 64–122 (cit. on p. 42).

[ARR96] Andreas Arning, Agrawal Rakesh, and Prabhakar Raghavan. "Method for Deviation in Large Databases." In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (1996), 164–169 (cit. on p. 8).

[Bac+19] Danya Bachir et al. "Inferring dynamic origin-destination flows by transport mode using mobile phone data." In: *Transportation Research Part C: Emerging Technologies* 101 (2019), pp. 254–275 (cit. on p. 40).

[Bar+18] Hugo Barbosa et al. "Human mobility: Models and applications." In: *Physics Reports* 734 (2018). Human mobility: Models and applications, pp. 1–74 (cit. on pp. 39, 78).

[Bioa] Biocoop. *Créer mon magasin Biocoop*. `https://ac-franchise.com/annuaire/franchise-monoprix`. accessed 15 April 2021 (cit. on p. 73).

[Biob] Biocoop. *Créer mon magasin Biocoop*. `https://www.biocoop.fr/magasins-bio/creer-mon-magasin-Biocoop`. accessed 8 December 2021 (cit. on p. 73).

[BM20] Hensher DA. Beck MJ. "Insights into the impact of COVID-19 on household travel and activities in Australia - The early days of easing restrictions." In: *Transp Policy (Oxf)* (2020) (cit. on p. 42).

[Boe17] G. Boeing. "OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks." In: *Computers, Environment and Urban Systems* 65 (2017), pp. 126–139 (cit. on p. 78).

[Bra+06] J. Branch et al. "In-Network Outlier Detection in Wireless Sensor Networks." In: *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*. 2006, pp. 51–51 (cit. on p. 8).

[CAKA05] Myun-Seok Cheon, Faiz A. Al-Khayyal, and Shabbir Ahmed. "Global optimization of monotonic programs: applications in polynomial and stochastic programming." In: 2005 (cit. on p. 113).

[Car] Carrefour. *Different format of stores in Carrefour group*. `https://www.carrefour.com/en/group/stores`. accessed 8 December 2021 (cit. on p. 73).

[Car+20] Raffaele Carli et al. "Model predictive control to mitigate the COVID-19 outbreak in a multi-region scenario." In: *Annual Reviews in Control* 50 (2020), pp. 373–393 (cit. on p. 42).

[CB] Clinique-Belledonne. *Clinique belledonne*. `https://www.clinique-belledonne.fr/presentation/`. accessed 6 December 2021 (cit. on p. 68).

[CC] Clinique-Cedres. *Official website of Clinique des Cedres*. `https://www.cliniquedescedres.com/la-clinique`. accessed 6 December 2021 (cit. on p. 68).

[Cen+21] Carlo Cenedese et al. "Optimal policy design to mitigate epidemics on networks using an SIS model." In: *2021 60th IEEE Conference on Decision and Control (CDC)*. 2021, pp. 4266–4271 (cit. on p. 42).

[Cho+19] Hussein Chour et al. "Global Optimal Resource Allocation for Efficient FD-D2D Enabled Cellular Network." In: *IEEE Access* 7 (2019), pp. 59690–59707 (cit. on pp. 114, 119).

[CHUa] CHU. *Collective data for CHU North, South and Voiron*. `https://www.chu-grenoble.fr/content/chiffres-cles-0`. accessed 6 December 2021 (cit. on p. 67).

[CHUb] CHU. *Public data for CHUs in Grenoble*. `https://www.chu-grenoble.fr/content/sites-plans-dacces`. accessed 15 April 2021 (cit. on p. 67).

[CM] Clinique-Mutualiste. *Official website of Clinique Mutualiste*. `https://www.ghm-grenoble.fr/Presentation.11.0.html`. accessed 6 December 2021 (cit. on p. 68).

[CMV13] Poletto C, Tizzoni M, and Colizza V. "Human mobility and time spent at destination: impact on spatial epidemic spreading." In: *J Theor Biol.* (2013) (cit. on pp. 40, 42).

[com17] European commission. "European Commision Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL concerning the respect for private life and the protection of personal data in electronic communications and repealing directive 2002/58/EC (Regulation on privacy and electronic Commu)." In: (2017) (cit. on p. 40).

[CV] CHU-Voiron. *Rules for visitation in CHU Voiron*. `https://www.ch-voiron.fr/patients-visiteurs/Visites.html`. accessed 15 April 2021 (cit. on p. 67).

[CYX14] Yao Chen, Mei Yan, and Zhongyi Xiang. "Transmission Dynamics of a Two-City SIR Epidemic Model with Transport-Related Infections." In: *Journal of Applied Mathematics* 2014 (2014), 764278:1–764278:12 (cit. on p. 42).

[DEK21] Ali Enes Dingil and Domokos Esztergár-Kiss. "The Influence of the Covid-19 Pandemic on Mobility Patterns: The First Wave's Results." In: *Transportation Letters* (2021) (cit. on p. 42).

[Den+14] Kun Deng et al. "Structure-preserving model reduction of nonlinear building thermal models." In: *Automatica* 50.4 (2014), pp. 1188–1195 (cit. on p. 7).

[DH18] Sina Dabiri and Kevin Heaslip. "Inferring transportation modes from GPS trajectories using a convolutional neural network." In: *Transportation Research Part C: Emerging Technologies* 86 (2018), pp. 360–371 (cit. on p. 40).

[DIa] Departement-ISERE. *Eligible high schools according address of a person in Isere*. `cache.media.education.gouv.fr/file/Scolarisation/38/6/Annexe_1_-_sectorisation_lycees-rentree_2021_1405386.pdf`. accessed 2 february 2022 (cit. on p. 78).

[DIb] Department-ISERE. *Sector for different middle schools*. `https://carto.isere.fr/carte-interactive/index.html?_ga=2.205615123.1019776183.1552918602-1334538445.1552918602`. accessed 2 february 2022 (cit. on p. 78).

[Dut+21] Ritabrata Dutta et al. "Using mobility data in the design of optimal lockdown strategies for the COVID-19 pandemic." In: *PLOS Computational Biology* 17 (Aug. 2021) (cit. on p. 42).

[Ekm+08] Frans Ekman et al. "Working Day Movement Model." In: *Proceedings of the 1st ACM SIGMOBILE Workshop on Mobility Models*. Hong Kong, Hong Kong, China: Association for Computing Machinery, 2008, 33–40 (cit. on p. 39).

[Emd] *Enquêtes Ménages Déplacements (EMD)*. `https://www.data.gouv.fr/en/datasets/enquetes-menages-deplacements-emd/`. accessed 6 December 2021. 2010 (cit. on p. 87).

[ES90] S. Erlander and N.F. Stewart. *The Gravity Model in Transportation Analysis: Theory and Extensions*. VSP, 1990 (cit. on p. 39).

[Gao20] Daozhou Gao. "How Does Dispersal Affect the Infection Size?" In: *SIAM Journal on Applied Mathematics* 80.5 (2020), pp. 2144–2169 (cit. on p. 42).

[Gau+09] Krings Gautier et al. "Urban gravity: a model for inter-city telecommunication flows." In: *J. Stat. Mech.* (2009) (cit. on p. 39).

[GF16] Qian Ge and Daisuke Fukuda. "Updating origin–destination matrices with aggregated data of GPS traces." In: *Transportation Research Part C: Emerging Technologies* 69 (2016), pp. 291–312 (cit. on p. 40).

[GI] Grenoble-INP. *Research centers associated with Grenoble INP*. `https://www.grenoble-inp.fr/en/research/laboratories`. accessed 6 December 2021 (cit. on p. 71).

[Gma] Grenoble-metrople. *Official webpage of Grenoble metropole*. https://www.grenoble.fr/1459-les-secteurs-de-grenoble.htm. accessed 6 December 2021 (cit. on pp. 61, 62).

[Gmb] Grenoble-metropole. *Grenoble Alpes metropole Tourism*. https://www.grenoble-tourisme.com/fr/faire/sorties/cinemas/?page=1. accessed 6 December 2021 (cit. on p. 76).

[Gmc] Grenoble-metropole. *Sectors of Grenoble*. https://www.grenoble.fr/1459-les-secteurs-de-grenoble.htm. accessed 6 December 2021 (cit. on p. 62).

[Gos+21] M Gosgens et al. "Trade-offs between mobility restrictions and transmission of SARS-CoV-2." In: *J R Soc Interface* (2021) (cit. on p. 42).

[Haw80] Douglas M Hawkins. *Identification of outliers*. English. Chapman and Hall, 1980 (cit. on p. 7).

[HKF04] V. Hautamaki, I. Karkkainen, and P. Franti. "Outlier detection using k-nearest neighbour graph." In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Vol. 3. 2004, 430–433 Vol.3 (cit. on p. 8).

[HT04] R. Hariharan and K. Toyama. "Project Lachesis:Parsing and Modeling Location Histories." In: *Geographic Information Science* (2004) (cit. on p. 40).

[Hu+21] W Hu et al. "Optimal strategic pandemic control: human mobility and travel restriction." In: *Math Biosci Eng* (2021) (cit. on p. 42).

[Hua+18] Zhiren Huang et al. "Modeling real-time human mobility based on mobile phone and transportation data fusion." In: *Transportation Research Part C: Emerging Technologies* 96 (2018), pp. 251–269 (cit. on p. 39).

[IBF18] 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)*. 2018, pp. 812–817 (cit. on p. 7).

[Ins] *Population of people of different age groups in different communes*. https://statistiques-locales.insee.fr/bbox=577774,5679420,119058,71925&c=indicator&i=rp.pop_3tr_ages&s=2017&view=map1. accessed 6 December 2021 (cit. on p. 62).

[INSa] INSEE. *Classification of enterprises in France*. https://www.insee.fr/fr/statistiques/4277836?sommaire=4318291. accessed 6 December 2021 (cit. on p. 69).

[INSb] INSEE. *INSEE home page*. https://www.insee.fr/en/accueil. accessed 6 December 2021 (cit. on p. 61).

[INSc] INSEE. *National average surface area of supermarkets*. https://www.insee.fr/fr/statistiques/1281004. accessed 6 December 2021 (cit. on pp. 73, 74).

[INSd] INSEE. *Population of the French communes*. https://statistiques-locales.insee.fr/bbox=598577,5676525,91084,55025&c=indicator&i=pop_depuis_1876.pop&s=2017&view=map1. accessed 6 December 2021 (cit. on pp. 61, 62).

[INSe] INSEE. *Statistical details of french communes.* `https://statistiques-locales.insee.fr/bbox=612227,5668725,62775,37924&c=indicator&i=pop_depuis_1876.pop&s=2017&selcodgeo=38045&view=map1`. accessed 6 December 2021 (cit. on p. 62).

[Iqb+14] Md. Shahadat Iqbal et al. "Development of origin–destination matrices using mobile phone call data." In: *Transportation Research Part C: Emerging Technologies* 40 (2014), pp. 63–74 (cit. on p. 40).

[Jou] The business Journals. *What's a Lidl?* `https://www.bizjournals.com/triad/blog/morning-edition/2015/04/whats-a-lidl-heres-the-scoop-on-huge-grocery-chain.html`. accessed 8 December 2021 (cit. on p. 73).

[Kö+21] Johannes Köhler et al. "Robust and optimal predictive control of the COVID-19 outbreak." In: *Annual Reviews in Control* 51 (2021), pp. 525–539 (cit. on p. 42).

[KBG04] Matt J. Keeling, Ottar N. Bjørnstad, and Bryan T. Grenfell. "Metapopulation Dynamics of Infectious Diseases." In: *Ecology, Genetics and Evolution of Metapopulations.* Ed. by Ilkka Hanski and Oscar E. Gaggiotti. Burlington: Academic Press, 2004, pp. 415–445 (cit. on p. 42).

[Kel+16] M.R. Kelly et al. "The impact of spatial arrangements on epidemic disease dynamics and intervention strategies." In: *Journal of biological dynamics* (2016) (cit. on p. 42).

[KNT00] Edwin M. Knorr, Raymond T. Ng, and Vladimir Tucakov. "Distance-based outliers: Algorithms and applications." In: *The VLDB Journal* 3 (2000), pp. 237–253 (cit. on p. 8).

[KR07] Matt J. Keeling and Pejman Rohani. "Modeling infectious disease in Humans and Animals." In: Princeton university Press, 2007 (cit. on p. 42).

[LABH21] A. Lipshtat, R. Alimi, and Y. Ben-Horin. "Commuting in metapopulation epidemic modeling." In: *Scientific reports* (2021) (cit. on p. 42).

[Las+21] Samson Lasaulce et al. "Analysis of the Tradeoff Between Health and Economic Impacts of the Covid-19 Epidemic." In: *Frontiers in Public Health* 9 (2021) (cit. on p. 42).

[LH14] Miao Lin and Wen-Jing Hsu. "Mining GPS data for mobility patterns: A survey." In: *Pervasive and Mobile Computing* 12 (2014), pp. 1–16 (cit. on p. 40).

[Lin] Linternaute. *Website with information on the number of seats in a restaurant in Grenoble.* `https://www.linternaute.com/restaurant/guide/ville-grenoble-38000/`. accessed 6 December 2021 (cit. on p. 74).

[LS13] Xinzhi Liu and Peter Stechlinski. "Transmission dynamics of a switched multicity model with transport-related infections." In: *Nonlinear Analysis: Real World Applications* 14.1 (2013), pp. 264–279 (cit. on p. 42).

[LXH21] Jian Li, Tao Xiang, and Linghui He. "Modeling epidemic spread in transportation networks: A review." In: *Journal of Traffic and Transportation Engineering (English Edition)* 8.2 (2021). Transportation Planning and Operations for COVID-19 Epidemic and Other Emergencies, pp. 139–152 (cit. on p. 42).

[Lyc] *Schedule for Lycee Emmanuel Mounnier*. `https://emmanuel-mounier.ent.auvergnerhonealpes.fr/le-lycee/horaires-des-cours/`. accessed 6 December 2021 (cit. on p. 85).

[MFW20] Nicolas Martin, Paolo Frasca, and Carlos Canudas-de Wit. "Subgraph Detection for Average Detectability of LTI Systems." In: *IEEE Transactions on Network Science and Engineering* 7.4 (2020), pp. 2787–2798 (cit. on p. 7).

[MM06] Mirco Musolesi and Cecilia Mascolo. "A Community Based Mobility Model for Ad Hoc Network Research." In: *Proceedings of the 2nd International Workshop on Multi-Hop Ad Hoc Networks: From Theory to Reality*. REALMAN '06. Florence, Italy: Association for Computing Machinery, 2006, 31–38 (cit. on p. 39).

[Mol+21] Joseph Molloy et al. "Observed impacts of the Covid-19 first wave on travel behaviour in Switzerland based on a large GPS panel." In: *Transport Policy* 104 (2021), pp. 43–51 (cit. on p. 42).

[Mor+21] D. H. Morris et al. "Optimal, near-optimal, and robust epidemic control." In: *Commmunication physics* 4 (2021) (cit. on p. 42).

[MR22] Emilio Molina and Alain Rapaport. *An optimal feedback control that minimizes the epidemic peak in the SIR model under a budget constraint*. 2022 (cit. on p. 42).

[MYA+13] de Montjoye Yves-Alexandre et al. "Unique in the Crowd: The privacy bounds of human mobility." In: *Scientific Reports* (2013) (cit. on p. 40).

[NBC+21] Pierre Nouvellet, Sangeeta Bhatia, Anne Cori, et al. "Reduction in mobility and COVID-19 transmission." In: *Nature Communications* (2021) (cit. on p. 42).

[NCK20] M. U. B. Niazi, C. Canudas-de-Wit, and A. Kibangou. "Average State Estimation in Large-scale Clustered Network Systems." In: *IEEE Transactions on Control of Network Systems* (2020). In press (cit. on p. 7).

[Nia+20] M. U. B. Niazi et al. "Scale-Free Estimation of the Average State in Large-Scale Systems." In: *IEEE Control Systems Letters* 4.1 (2020), pp. 211–216 (cit. on pp. 7, 15).

[Nia+21] Muhammad Umar Niazi et al. "Optimal Control of Urban Human Mobility for Epidemic Mitigation." In: *CDC 2021 - 60th IEEE Conference on Decision and Control, Dec 2021, Austin, United States* (2021) (cit. on pp. 40–42, 45, 47, 49, 50, 98, 100).

[NWK20] Muhammad Umar B. Niazi, Carlos Canudas de Wit, and Alain Y. Kibangou. "Thermal Monitoring of Buildings by Aggregated Temperature Estimation." In: *IFAC-PapersOnLine* 53.2 (2020). 21st IFAC World Congress, pp. 4132–4137 (cit. on p. 7).

[Off] *French official working hours*. `https://businessculture.org/western-europe/business-culture-in-france/work-life-balance-in-france/`. accessed 6 December 2021 (cit. on p. 85).

[Opea] Opendatasoft. *Coordinates of the the IRIS in France*. https://public.opendatasoft.com/explore/dataset/georef-france-iris/export/?disjunctive.reg_name&disjunctive.dep_name&disjunctive.arrdep_name&disjunctive.ze2010_name&disjunctive.bv2012_name&disjunctive.epci_name&disjunctive.ept_name&disjunctive.com_name&disjunctive.com_arm_name&disjunctive.iris_name&sort=year&q=38185&refine.arrdep_name=Grenoble&refine.com_name=Grenoble&location=12,45.19695,5.74791&basemap=jawg.streets. accessed 6 December 2021 (cit. on p. 61).

[Opeb] Openstreetmap. *Grenoble area on OpenStreetMap*. https://www.openstreetmap.org/map=11/45.1635/5.7273. accessed 6 December 2021 (cit. on pp. 61, 63).

[Pre] Presences. *Official website of the magazine Presences: les magazine des enterprises du Sud Isere*. https://www.presences-grenoble.fr/. issue: January, 2021 (cit. on p. 70).

[RFa] Republique-Francaise. *Coordinates of the boundaries of communes*. https://www.data.gouv.fr/fr/datasets/r/07b7c9a2-d1e2-4da6-9f20-01a7b72d4b12. accessed 6 December 2021 (cit. on p. 61).

[RFb] Republique-Francaise. *Schedules for primary schools*. https://www.service-public.fr/particuliers/vosdroits/F24490. accessed 6 December 2021 (cit. on p. 85).

[RHG15] Mohsen Ramezani, Jack Haddad, and Nikolas Geroliminis. "Dynamics of heterogeneity in urban networks: aggregated traffic modeling and hierarchical control." In: *Transportation Research Part B: Methodological* 74 (2015), pp. 1–19 (cit. on p. 7).

[RL87] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. USA: John Wiley & Sons, Inc., 1987 (cit. on p. 7).

[RT04] John R. Roy and Jean-Claude Thill. *Spatial interaction modelling*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 339–361 (cit. on p. 39).

[RVWF20] Martin Rodriguez-Vega, Carlos Canudas de Wit, and Hassen Fourati. "Average density detectability in traffic networks using virtual road divisions." In: *IFAC-PapersOnLine* 53.2 (2020). 21st IFAC World Congress, pp. 17059–17064 (cit. on p. 7).

[Sch+21] Markus Schläpfer et al. "The universal visitation law of human mobility." In: *Nature* 593 (2021) (cit. on p. 78).

[SD10] Agus Suryanto and Isnani Darti. "On the nonstandard numerical discretization of SIR epidemic model with a saturated incidence rate and vaccination." In: *AIMS Mathematics, 6(1): 141–155.* (2010) (cit. on p. 55).

[SD95] Lisa Sattenspiel and Klaus Dietz. "A structured epidemic model incorporating geographic mobility among regions." In: *Mathematical Biosciences* 128.1 (1995), pp. 71–91 (cit. on pp. 39, 40, 42).

[Sec] *Schedule for Fantin Latour middle school*. http://www.ac-grenoble.fr/college/fantin-latour-grenoble/?page_id=71. accessed 6 December 2021 (cit. on p. 85).

[SII17] Tomonori Sadamoto, Takayuki Ishizaki, and Jun Ichi Imura. "Average state observers for large-scale network systems." In: *IEEE Transactions on Control of Network Systems* 4.4 (2017), pp. 761–769 (cit. on pp. 7, 30).

[Sim+12] F. Simini et al. "A universal model for mobility and migration patterns." In: *nature* 484 (2012), pp. 96–100 (cit. on p. 39).

[Smo+21] Alex Smolyak et al. "Effects of mobility restrictions during COVID19 in Italy." In: *Scientific Reports* (2021), pp. 2045–2322 (cit. on p. 42).

[Son+20] Sirui Song et al. *Reinforced Epidemic Control: Saving Both Lives and Economy*. 2020. arXiv: 2008.01257 [cs.AI] (cit. on p. 42).

[TAKT05] Hoang Tuy, Faiz Al-Khayyal, and Phan Thien Thach. "Monotonic Optimization: Branch and Cut Methods." In: *Essays and Surveys in Global Optimization*. Ed. by Charles Audet, Pierre Hansen, and Gilles Savard. Boston, MA: Springer US, 2005, pp. 39–78 (cit. on pp. 106–108, 111).

[Tan+15] Jinjun Tang et al. "Uncovering urban human mobility from large scale taxi GPS data." In: *Physica A: Statistical Mechanics and its Applications* 438 (2015), pp. 140–153 (cit. on p. 40).

[Ter] Territorial.fr. *Safety regulation for the maximum number of persons allowed according to surface area*. https://www.territorial.fr/PAR_TPL_IDENTIFIANT/17172/TPL_CODE/TPL_OVN_CHAPITRE_FICHE/3848-consultation-le-guide-pratique-du-dst.htm. accessed 6 December 2021 (cit. on pp. 72, 73).

[Tuy00] Hoang Tuy. "Monotonic Optimization: Problems and Solution Approaches." In: *SIAM Journal on Optimization* 11.2 (2000), pp. 464–494. eprint: https://doi.org/10.1137/S1052623499359828 (cit. on pp. 106–109, 111).

[UGA] UGA. *Official website of University Grenoble Alpes*. https://www.univ-grenoble-alpes.fr/about/uga-in-numbers. accessed 6 December 2021 (cit. on p. 65).

[Wan+18] Hongwei Wang et al. "Robust Gaussian Kalman Filter with Outlier Detection." In: *IEEE Signal Processing Letters* 25.8 (2018), pp. 1236–1240 (cit. on p. 8).

[WHL18] Zhenzhen Wang, Sylvia Y. He, and Yee Leung. "Applying mobile phone data to travel behaviour research: A literature review." In: *Travel Behaviour and Society* 11 (2018), pp. 141–155 (cit. on p. 40).

[XSB14] Yunwen Xu, Srinivasa M. Salapaka, and Carolyn L. Beck. "Aggregation of graph models and markov chains by deterministic annealing." In: *IEEE Transactions on Automatic Control* 59.10 (2014), pp. 2807–2812 (cit. on p. 20).

[Yin+15] Ling Yin et al. "Re-Identification Risk versus Data Utility for Aggregated Mobility Research Using Mobile Phone Location Data." In: *PLOS ONE* 10 (Oct. 2015), pp. 1–23 (cit. on p. 40).

[Yin+20] Qian Yin et al. "A novel epidemic model considering demographics and intercity commuting on complex dynamical networks." In: *Applied Mathematics and Computation* 386 (2020), p. 125517 (cit. on p. 42).

[YYH12] Sung-Jib Yim and Choi Yoon-Hwa. "Neighbor-Based Malicious Node Detection in Wireless Sensor Networks." In: *Wireless Sensor Networks* 4.September (2012), pp. 361–374 (cit. on p. 8).

[ZB11] Hui Zang and Jean Bolot. "Anonymization of Location Data Does Not Work: A Large-Scale Measurement Study." In: *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*. MobiCom '11. Las Vegas, Nevada, USA: Association for Computing Machinery, 2011, 145–156 (cit. on p. 40).

[ZC09] Dennis Zill and Michael Cullen. *Differential Equations with Boundary-Value Problems*. Brooks/Cole Cengage Learning, Canada, 2009 (cit. on p. 31).

[ZG10] Matteo Zignani and Sabrina Gaito. "Extracting human mobility patterns from GPS-based traces." In: *2010 IFIP Wireless Days*. 2010, pp. 1–5 (cit. on p. 40).

[Zha+16] Ziliang Zhao et al. "Understanding the bias of call detail records in human mobility research." In: *International Journal of Geographical Information Science* 30.9 (2016), pp. 1738–1762 (cit. on p. 40).

[ZMH10] Y. Zhang, N. Meratnia, and P. Havinga. "Outlier Detection Techniques for Wireless Sensor Networks: A Survey." In: *IEEE Communications Surveys Tutorials* 12.2 (2010), pp. 159–170 (cit. on p. 8).

[ZQH13] Ying Jun Zhang, Liping Qian, and Jianwei Huang. "Monotonic Optimization in Communication and Networking Systems." In: *Foundation and trends in networking* 7.1(2012) (2013), pp. 1–75 (cit. on pp. 111, 113, 114).