POLITECNICO DI TORINO

Master's degree in
Mathematical Engineering

Master's degree thesis

Design, validation, and analysis of a game-theoretic network epidemic model for COVID-19 spreading in Italy

Supervisor
prof. Alessandro Rizzo
prof. Ming Cao
dott. Lorenzo Zino
supervisors' signature

. . . . . .
. . . . . .

Candidate
Luca Ambrosino

candidate's signature

. . . . . .

Academic year 2021-2022

Ringraziamenti

Ringrazio in primo luogo il professor Alessandro Rizzo per avermi dato fiducia fin da subito, e avermi concesso la possibilità di lavorare insieme a questo progetto di ricerca. Grazie per i consigli e la disponibilità sempre mostrata.

Ringrazio anche il professor Ming Cao dell'università di Groningen, per avermi ospitato durante il mio soggiorno all'estero, permettendomi di vivere la vita universitaria in modo completamente diverso da come ero abituato in Italia. Conserverò con cura questa fantastica esperienza, anche grazie ai tanti ricercatori e ricercatrici provenienti da tutto il mondo che ho avuto il piacere di conoscere lì.

Un super grazie al dottor Lorenzo Zino, con cui ho avuto il piacere di collaborare durante tutta la durata del mio lavoro di tesi, prima a Groningen e poi a Torino. Grazie per i tanti preziosi consigli che mi hai dato letteralmente dal primo all'ultimo giorno, e soprattutto a qualsiasi ora e in qualsiasi luogo. La stima che hai mostrato nei miei confronti è stato un eccellente carburante per permettermi di concludere questo lavoro al meglio, ponendo le basi per delle collaborazioni future che sarei felice di portare a termine. Ovviamente, questa stima è ricambiata da parte mia, ti auguro di realizzare in futuro tutti i tuoi obiettivi professionali e non.

Questa tesi non è soltanto un elaborato finale, ma è il coronamento di un percorso cominciato per me diversi anni fa, ed è per questo che vorrei ringraziare tutte le persone che mi sono state vicine durante questo viaggio:

Tutto questo non sarebbe certamente stato possibile senza l'aiuto e la spinta costante dei miei genitori Sonia e Ignazio, che fin da quando ero piccolo hanno investito il loro tempo e le loro energie nella mia istruzione, senza farmi mancare mai nulla. Vi chiedo scusa se a volte, per qualsiasi motivo, sono risultato scontroso, e magari non ho apprezzato alcuni vostri gesti come il venirmi a salutare prima di andare a dormire, aprendo la porta della mia stanza mentre ero stanco e lavoravo al PC, chiedendomi semplicemente se andasse tutto bene. Col senno di poi, lo apprezzo moltissimo. Mi sono sempre impegnato al massimo in tutto quello che facevo, e in particolare nello studio. Spero che adesso possiate essere fieri dei risultati che ho ottenuto e di quello che sono.
Vi ringrazio davvero. Vi voglio tanto bene.

Ringrazio i miei fratelli e tutta la mia famiglia per avermi sempre supportato e sopportato ogni giorno, perchè mi rendo conto che spesso non sono stato presente quanto e come avrei voluto, preso da tante altre cose, magari rispondendovi male o non dando il giusto peso alle vostre attenzioni. Grazie per avermi sempre dato una mano quando ne avevo più bisogno, ognuno a modo suo. Auguro a ciascuno di voi di poter vivere in futuro altri mille giorni in cui essere felici tanto quanto io lo sono oggi.

Ai miei quattro meravigliosi nonni. Mentre sto scrivendo mi tornano in mente dei ricordi che vorrei non scomparissero mai.
Mi ricordo di quando nonno Giuseppe mi faceva trovare i pacchetti di figurine di nascosto nella tasca del mio giubbotto, o di come gli si illuminavano gli occhi appena pescava il settebello mentre giocavamo a scopa.
Mi ricordo delle bellissime torte che nonna Ida mi preparava ad ogni compleanno, così come mi ricordo, quando ancora ero più basso di lei, le sue mani accarezzarmi la testa come se fossi la cosa più preziosa del mondo. Avrei davvero voluto avere più tempo con voi. Mi mancate molto.
Mi ricordo gli occhi pieni di gioia di mio nonno Filippo ad ogni pranzo di famiglia, in ogni occasione di festa, e mi ricordo con quanto impegno cercavi di spiegarci tutti i sistemi che giochi al Superenalotto. Stiamo tutti ancora aspettando che tu faccia sei, sono convinto che un giorno lo farai.
Mi ricordo ogni volta in cui sottovoce nonna Maria mi lasciava 20 euro "per la benzina", e mi ricordo di ogni singolo abbraccio che ci siamo dati dopo la paura del COVID-19. Con le tue parole mi hai spinto ogni giorno a dare il massimo per concludere al meglio questa tesi, e sono sicuro che ora tu sia tanto felice quanto lo sarebbe stato chi non ha la fortuna di potermi abbracciare oggi. Tutto questo è anche per voi.

Grazie a Marco e Lorenzo, gli amici e compagni più presenti in questo lungo viaggio, anche quando forse quello assente ero io. Vi ringrazio per ogni pomeriggio (perchè di mattina almeno uno di noi era sempre in ritardo) passato insieme a studiare, a chiacchierare, a girare la ruota e ad apprendere per osmosi i metodi di studio del buon Marco, a cui devo davvero molto. Grazie anche per ogni pre esame in cui ci davamo la carica a vicenda, prendendoci amichevolmente a pugni per gasarci. Ne abbiamo passate davvero tante insieme, se ora le nostre strade professionali si separeranno spero di non perderci mai.

A Emanuele, Martina, e tutti gli altri colleghi e amici che hanno condiviso con me delle tappe fondamentali per il nostro percorso universitario. Sempre pronti a darci una mano a vicenda, un consiglio, un abbraccio, o qualsiasi cosa di cui avessimo bisogno in quelle lunghissime sessioni esami durante le quali prendevamo residenza insieme in aula studio. Sono davvero felice di condividere questo traguardo oggi con voi, grazie, e congratulazioni!

Ora che sto concludendo questo percorso non può non venirmi in mente da dove invece è cominciato, da quell'aula 3 in cui ho conosciuto Marco e Arianna appena abbiamo messo piede all'università. Vi ringrazio per quel primo anno e per tutti i successivi passati insieme, anche se a volte distanti. Non dimenticherò ogni singolo caffè preso insieme, i pranzi in mensa e, soprattutto, tutte le serate degne di un cinque altissimo. Ci siamo

sempre stati e sono sicuro che ci saremo sempre. Non vedo l'ora di condividere con voi altri racconti davanti a un buon bicchiere di vino e alla cena preparata con amore dalla nostra grande piccola cuoca.

Se sono riuscito a superare tutte le difficoltà e le sfide che l'università mi ha riservato, è anche grazie a tutte le persone che mi hanno voluto bene al di fuori di questa:

Grazie a Maurizio, una persona dal cuore d'oro e da cui ho imparato molto sul cosa voglia dire voler bene a qualcuno. Lui e tutta la sua famiglia mi hanno accolto a braccia aperte quando neanche mi conoscevano, donandomi un affetto e una fiducia per niente scontata. Ogni momento passato insieme, in qualsiasi regione italiana, mi ha sempre aiutato a sentirmi bene, specialmente quando avevo tanto bisogno di staccare. Per tutti i momenti passati insieme, e per non avermi mai fatto pesare invece i momenti non passati insieme, ti ringrazio amico mio.

Grazie a Fabio, Daniele, Fabio, Martina e Stefania, gli amici di Candiolo e di sempre, citati in puro ordine cronologico. Mi dispiace per avervi detto di no innumerevoli volte perchè ero concentrato nel perseguire i miei obiettivi universitari o altro, ma sono felice che nonostante le nostre strade professionali seguano 6 binari tutti diversi, riusciamo a non perdere l'occasione di ritrovarci per una cena, una sessione di palestra (chi più, chi meno...), un calcetto o una serata giochi tutti insieme. Mi auguro che tutto questo continuerà anche quando un giorno, magari, non vivremo più tutti vicini.

Grazie a Federica e Aurora per avermi dimostrato negli anni di tenere a me quanto io tengo a loro. Vi auguro davvero tante gioie nella vita e, perchè no, anche al fantacalcio. Grazie agli amici di Perle, ai miei compagni di squadra nei Citmabun e a tutte quelle persone che in momenti diversi della mia vita, magari anche senza pensarci, mi sono state vicine e hanno reso speciale ogni ora trascorsa insieme.

Grazie anche a quelle persone che nella mia vita ci sono state, ma ora non più.
Ognuna di loro ha sicuramente costruito un pezzo più o meno grande della persona che sono adesso, nel bene e nel male. Il mondo ha di recente dovuto affrontare una pandemia e una guerra, perciò mi auguro di riuscire in futuro a dare il giusto peso alle gioie, ma soprattutto alle delusioni.

Infine, vorrei ringraziare me stesso, per l'impegno e la determinazione che ho messo ogni giorno per perseguire i miei obiettivi. Ora, finalmente, posso godermeli, prima di fissarne di nuovi.

Grazie.

Summary

In the recent past, the world population has faced a very difficult period due to a globally widespread disease, known to all as COVID-19. This virus originated in China but then spread rapidly throughout the rest of the world, arriving in Italy at the end of February 2020 which was the beginning of some terrible months: COVID-19 positive individuals increased dramatically and hospitals, with their medical staff, had to endure unprecedented stress, not to forget also the huge number of deaths this epidemic caused.
The government that was in charge at that time tried to limit the damage caused by this epidemic with a series of interventions and restrictive measures which were absolutely necessary to contain the spread of the virus, but on the other hand inevitably caused a difficult economic crisis causing discontent among a large part of the population: this is certainly a trouble, because the population is notoriously heterogeneous and it is not always possible to predict everyone's behavior and reactions. At the state of the art were studied many epidemic models already before the spread of COVID-19, but very few of these deal with the aspect of people's behavior as a dynamic that influences the spread of the epidemic.
This master's thesis aims to design two new epidemic models that also consider the behavior of people, with the design of specific payoff functions for a decision-making process based on game theory, in the first model to evaluate the choice of adopting self-protective behaviors like social distancing or wearing protective mask, while in the second model a decision on vaccination will also be added, which has been a delicate matter and discussed among people for a long time.
Once this model has been created, it will be validated with a calibration of parameters and a validation based on the real data of infected and dead COVID-19 collected in Italy in recent years. After the validation of the model we will try to assess through a control function new intervention policies by the government called NPI, or non-pharmaceutical interventions, with the aim of obviously reducing the number of infections and deaths to a minimum, being useful in the case of a future new pandemic that we all wish it would never happen again.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In several moments of history, humanity had to deal with epidemics. Just think at the typhus that has decimated numerous armies over the years, including that of Napoleon; the Spanish flu that killed about 50 millions people between 1918 and 1920; AIDS which today has about 35 million victims. COVID-19 has recently joined this list; COVID-19 is a disease caused by the coronavirus SARS-CoV-2.
This virus appeared for the first time in December 2019 in China and then spread around the world [31]: even though, the exact origin of COVID-19 is not known, as well as it's not known who is the so-called "patient zero", i.e. the first individual infected ever.
On 31 December 2019, Chinese health authorities reported a cluster of pneumonia cases of unknown etiology in the city of Wuhan (China) [10]. Many of the initial cases reported a history of exposure to Wuhan's South China Seafood City market, therefore a transmission from live animals was suspected. People with this pneumonia often suffer from flu-like symptoms such as dermatitis, fever, dry cough, fatigue, difficulty breathing. In the most serious cases, often found in subjects already burdened by previous pathologies, pneumonia develops, acute respiratory failure up to even death. Up to today, more than 6.5 million people have died due to COVID-19-related health problems worldwide [8], and such a death toll is still growing.

The first signs of the spread of this virus in Italy [32] came on 30 January 2020, when two Chinese tourists tested positive for SARS-CoV-2 in Rome. In the following weeks, COVID-19 inevitably spread throughout the Italian peninsula, initially affecting the areas of northern Italy the most, as it's shown in Figure 1.2, because they are denser in population and with a climate perhaps more favorable to the spread of the virus than in the areas of southern Italy.
In response of such health threat, on 4 March 2020 the Italian government ordered the complete closure of all schools and universities of all levels at the national level, having registered 100 deaths in the country due to the disease. On the evening of 9 March 2020, during a live nationally televised press conference, the Italian prime minister Giuseppe Conte [28] extended the quarantine ordering severe confinement measures for the population, and the interruption of numerous productive, commercial and professional activities,

Figure 1.1. COVID-19 epidemic spread in China up to 31/01/2020, taken from [5]. We can see in black the city of Wuhan, where COVID-19 is born at the end of 2019.

with the exception of the agri-food, health, essential services and other businesses supplying basic necessities.

In Italy, as well as in the rest of the world, those months were difficult for the population who necessarily has had to change their lifestyle and daily habits: just think of the huge increase in workers confined to smart working at their homes, an activity that until then had never been an option for most companies, or think about how quickly the use of masks and hand sanitizing gels has spread, in fact almost every person did not leave the house (when they could) without taking these items with them.
For many months there was a lockdown and most people were confined to their homes to limit as much as possible contacts and consequentially the spread of infections. These measures, unfortunately, made a large slice of the population unhappy, due to their social, psychological and economical consequences. Such an impact on the population may also reduce the effectiveness of such measures, since unhappy people may not comply with them.
This is an important matter when dealing with a long lasting disease. In particular, if we want to create a model able to capture all the dynamics involved in an epidemic spread, we should not take only a purely epidemic point of view, but we should include in our analysis the population's behavior and mental health as a crucial component of the model.

The current literature on mathematical modeling of epidemic disease is rich in epidemic pattern studies though adopting oversimplified behavioral response [19], limiting their real-world applicability when long-lasting epidemic are concerned.
This research master thesis aims to fill in this gap by building on the work developed by Ye

Figure 1.2. Covid-19 epidemic spread in Italy up to 13/03/2020.

et al. in their paper named *"Game-theoretic modeling of collective decision making during epidemics"* [39], which combines a game-theoretical behavioral decision process with an epidemic analysis of a simple SIS model, which we are going to introduce in section 2.4.2. The game-theoretical behavioral dynamic concerns the possibility for each individual to adopt or not to adopt security measures, and it is linked to a specific payoff function (see (3.5)) which groups several elements that influence people's behavior, such as the non-pharmaceutical interventions (NPIs) that the national authorities can apply in order to reduce the spreading of the epidemics, such as lockdowns or other duties; this NPIs are represented in [39] by the control function $u(t)$.
Here, we expand on this work in several directions. First, we are going to extend their basic model to capture COVID-19. Second, we will use the real data collected in [7] for the COVID-19 pandemic in Italy starting from 4 February 2020 (therefore concerning the first epidemic wave in Italy) to calibrate and validate the model. Third, we perform an extensive analysis of the effects of NPIs by performing campaign of Monte Carlo simulations to highlight the impact of the control function $u(t)$ on the epidemic spreading.

Then, further extensions are performed. In chapter 4 we will move to a more recent time period, when the vaccine against COVID-19 was available and the vaccination campaign started (i.e., we are talking about the first months of 2021 in Italy). We capture this new element in our model by introducing a new decision-making process concerning

the possibility for each individual to get vaccinated or not. To this aim, we implement a new payoff function (4.1) with another control input concerning the vaccination campaign, named $u_v(t)$.
A new model validation has been performed on the model with vaccine, similar to those done for the first wave model.
Finally, the impact of the two (possibly related (4.7)) control functions $u(t)$ and $u_v(t)$ is extensively investigated, toward optimally design them.

In order to calibrate, validate and analyze our models we needed to perform several MATLAB simulations, which can be found in the appendix A, at the end of this thesis. We had to simulate numerous different trajectories because the epidemic dynamics that our models describe are stochastic, therefore we calculated the average among many different Monte Carlo simulations [30], in order to exclude anomalies due to noise and therefore judge the results obtained as reliable.

The results of our studies could be of interest for researches, as well as for public health authorities to help design effective intervention policies. In fact, our model calibrations suggest that an epidemic model that explicitly incorporate human behavior is key to accurately capture the evolution of an epidemic outbreak. Moreover, the promising results obtained through the analysis of alternative of NPIs suggest that our framework can be a useful supporting toolbox for public administrations for their decision-making.

Chapter 2

Preliminaries

First of all, it may certainly be useful to start by introducing the basic concepts of game theory, game over networks and epidemic models, necessary to better understand the foundations on which our model is developed.

2.1 Basic Elements of Game Theory

We can start by considering a simple game [3] in which we have a finite set of players $\mathcal{V}$ and a finite set of actions $\mathcal{A}$. It is useful to introduce the notion of *configuration vector* $\mathbf{x} \in \mathcal{A}^{\mathcal{V}}$ which stores the actions chosen by each player; clearly, the vector $\mathbf{x}$ belong to the *configuration space* $\mathcal{X} = \mathcal{A}^{\mathcal{V}}$.
Each player $i \in \mathcal{V}$ is equipped with a **utility** function u_i (also called *reward* or *payoff* function)

$$u_i : \mathcal{X} \to \mathbb{R}$$

that associates every configuration vector $\mathbf{x}$ in $\mathcal{X}$ with the utility $u_i(x)$ that player i gets when each player j is playing action $x_j \in \mathcal{A}$ (where x_j is the j-th element of the vector $\mathbf{x}$); the utility function is a kind of mathematical quantification of a player's happiness in choosing action x_i, knowing all the actions chosen by the other players.
In general, the image of function $u_i(x)$ can be the whole set of real numbers, but very often we will find as image a more precise subset $\mathcal{S} \subset \mathbb{R}$, for example $\mathcal{S} = [0,1]$ or $\mathcal{S} = [-1,1]$.
We can also indicate as $\mathbf{x}_{-i}$ the vector obtained from the configuration x by removing its i-th entry (clearly, $x_i \cup \mathbf{x}_{-i} = \mathbf{x}$). Given this, we can also rewrite

$$u_i(\mathbf{x}) = u(x_i, \mathbf{x}_{-i})$$

as the utility received by player i when she chooses to play action x_i, and the rest of the players choose to play $\mathbf{x}_{-i}$.
The triple

$$(\mathcal{V}, \mathcal{A}, \{u_i\}_{i \in \mathcal{V}})$$

will be referred to as a (strategic form) **game** [3].

Every player i is to be interpreted as a rational agent choosing her action x_i from the action set $\mathcal{A}$ so as to maximize her own utility $u_i(x_i, \mathbf{x}_{-i})$. In consideration of the fact that this utility depends not only on player i's action x_i but also on the actions of the rest of the players' actions $\mathbf{x}_{-i}$, it is natural to introduce the **best response ($\mathcal{BR}$)** function

$$\mathcal{BR}_i(\mathbf{x}_{-i}) = \underset{x_i \in \mathcal{A}}{\operatorname{argmax}} \quad u_i(x_i, \mathbf{x}_{-i}).$$

Assuming that player i knows what the rest of the players' actions are and that these are not changing, choosing an action in $\mathcal{BR}_i(\mathbf{x}_{-i})$ is for her the rational choice as it makes her utility as large as possible.
In game theory we often deal with the concept of **Nash equilibrium**: a (pure strategy) Nash equilibrium (NE) for the game $(\mathcal{V}, \mathcal{A}, \{u_i\}_{i \in \mathcal{V}})$ is an action configuration $x^* \in \mathcal{X}$ such that

$$x_i^* \in \mathcal{BR}_i(\mathbf{x}_{-i}), \quad \forall i \in \mathcal{V}.$$

The Nash equilibrium x^* is called strict if $|\mathcal{BR}_i(\mathbf{x}_{-i})| = 1$ for every i.
The interpretation of a Nash equilibrium is the following: it is a configuration such that no player has any incentive to *unilaterally* deviate from her current action, as the utility she is getting with the current action is the best possible given the current actions chosen by the other players. Note the emphasis on "unilaterally": it is not at all guaranteed that coordinated deviations of more than one player from their actions in a Nash equilibrium could not lead to a higher utility for these players. As we shall see, there are games with multiple Nash equilibria and games which instead have none. We will denote by $\mathcal{N}_{eq}$ the set of NE of a game.

Instead of using the standard best response functions, some models use *smoothed* best response functions [26], in which the function does not "jump" from one pure strategy to another but the decision changes more continuously; In standard best response correspondences, even the slightest benefit to one action will result in the individual playing that action with probability 1, while with the smoothed best response approach the decision is more stochastic. There are many functions that represent smoothed best response functions, most of them are of the form of:

$$\frac{e^{\frac{u_i(0,\mathbf{x}_{-i})}{\gamma}}}{e^{\frac{u_i(0,\mathbf{x}_{-i})}{\gamma}} + e^{\frac{u_i(1,\mathbf{x}_{-i})}{\gamma}}} \tag{2.1}$$

where again $u_i(x_i, \mathbf{x}_{-i})$ is the utility that player i has by choosing action x_i (i.e. 0 or 1 in this case) while all the other players are choosing $\mathbf{x}_{-i}$, and γ is a scaling parameter.

2.1.1 Two-player games

The simplest examples of games are those with just two players $\mathcal{V} = \{1, 2\}$, that are referred to as *two-player games*. A two-player game [3] is simply characterized by two utility functions $u_i(a_i, a_j)$, for $i = 1, 2$, with the understanding that a_i is the action played

by player i and a_j the action played by his opponent $j \neq i$.
An important special case is when

$$u_1(a_1, a_2) = u_2(a_2, a_1) = \varphi(a_1, a_2) = \varphi(a_2, a_1), \quad a_1, a_2 \in \mathcal{A}, \tag{2.2}$$

which amounts to say that the role of the two players is exchangeable. A two-player game satisfying (2.2) is referred to as *symmetric*; symmetric two-player games are indeed characterized by a single utility function $\varphi(a_i, a_j)$.
When the action set $\mathcal{A}$ is finite, the payoffs (or utilities) of a two-player game are usually represented by a table called *payoff matrix*, whose rows and columns correspond to the actions of player 1 and player 2 respectively.
Table 2.1 shows an example of a payoff matrix, when we have a symmetric two-player game with (finite) action set $\mathcal{A} = \{0,1\}$.

	0	1
0	a,a	d,c
1	c,d	b,b

Table 2.1. Payoff matrix of a symmetric 2-player game with (finite) action set $\mathcal{A} = \{0,1\}$.

2.1.2 The prisoner's dilemma

The prisoner's dilemma is a standard example of a (symmetric) game analyzed in game theory that shows why two completely rational agents might not cooperate, even if it appears that it is in their best interests to do so.
This dilemma [3] concerns two members of a criminal gang who are arrested and imprisoned; each prisoner is in solitary confinement, not able to speak to or exchange messages with the other. The police do not have enough evidence to convict the pair on the principal charge but have evidence to convict them to b years in prison on a lesser charge. Simultaneously, the police offer each prisoner an interesting bargain. Each prisoner is given the opportunity either to: confess the other by testifying that she committed the main crime (action 1), or to remain silent (action 0).
If we rename the two prisoners as A and B, the possible outcomes are [33]:

- If A and B each betray the other, each of them serves two years in prison.
- If A betrays B but B remains silent, A will be set free and B will serve three years in prison.
- If A remains silent but B betrays A, A will serve three years in prison and B will be set free.
- If A and B both remain silent, both of them will serve one year in prison (on the lesser charge).

The prisoner's dilemma is therefore a symmetric 2×2-game with payoff matrix as in 2.1 whose entries satisfy the relation $c > a > b > d$.

Standard prisoner's dilemma payoff matrix

A \ B	B stays silent	B betrays
A stays silent	-1 / -1	0 / -3
A betrays	-3 / 0	-2 / -2

Figure 2.1. Payoff matrix of the prisoner's dilemma [33].

It is not hard to verify that the prisoner dilemma admits a unique (pure strategy) Nash equilibrium (the configuration $(1, 1)$, where the prisoners betray each other) and it's also clear that action 1 (betraying) dominates action 0 (remaining silent): no matter what the other prisoner does betraying always guarantees a better payoff.
We can also observe that, on the other hand, if the prisoners could coordinate and remain both silent then they would both get a better payoff than the one they get at the previous Nash equilibrium $(1, 1)$. The prisoner's dilemma illustrates indeed that the decisions made under collective rationality may not necessarily be the same as that made under individual rationality.

2.1.3 Coordination game

A coordination game is a symmetric 2×2-game with payoff matrix as in 2.1 with $a > c$ and $b > d$. These two simple inequalities imply that the best response for each player is to copy the action of the other player. Suppose we have the action set $\mathcal{A} = \{0,1\}$, we will have in formulas:

$$\mathcal{BR}_1(0) = \mathcal{BR}_2(0) = 0, \qquad \mathcal{BR}_1(1) = \mathcal{BR}_2(1) = 1.$$

It is very simple to verify that a 2×2-coordination game admits two Nash equilibria: the two "consensus" configuration (0,0) and (1,1).

A classical example of a coordination game [27] is choosing the sides of the road on which to drive: assume that two drivers meet on a narrow dirt road and obviously both have to swerve in order to avoid a collision. If both execute the same swerving maneuver (i.e. both go right or both go left) they will manage to pass each other safely, but if they

choose different maneuvers they will collide.
In this case there are two pure Nash equilibria: either both swerve to the left, or both swerve to the right.

2.2 Basic elements of graph theory

Graphs are mathematical structures used to model pairwise relations between agents (which can be objects, people, etc); they are made up of *vertices* (also called nodes) connected by *edges* (also called links).
More formally [3], a graph $\mathcal{G}$ is triple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ where:

- $\mathcal{V}$ is the set of vertices. We usually define the number of vertices $n = |\mathcal{V}|$ as the *order* of the graph.
- $\mathcal{E} \subseteq \{(x, y)|x, y \in \mathcal{V}, x \neq y\}$ is the set of edges. Each edge is usually an ordered pair of nodes, with the second one (i.e. y in the case of the edge (x, y)) called the end point of the edge.
- $\mathcal{W} \in \mathbf{R}_{+}^{\mathcal{V}\times\mathcal{V}}$ is the *weight matrix*, which has $\mathcal{W}_{x,y} > 0$ if and only if (x, y) is a link of the graph.

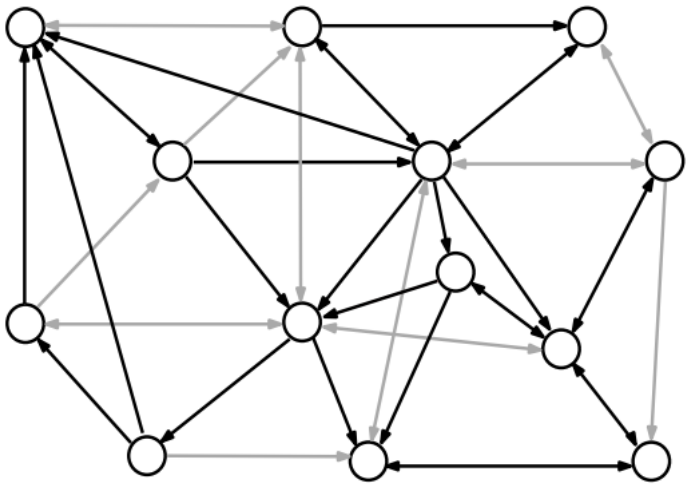

Figure 2.2. An example of a weighted and directed graph where edges of weight 1 are drawn in grey and edges of weight 2 are drawn in black. [3]

Depending on its characteristics a graph can also be:

- **Unweighted** if

$$\mathcal{W}_{x,y} = \begin{cases} 1, & \text{if} \quad (x, y) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases}$$

In this case we can describe the graph only as the pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and $\mathcal{W}$ is called *adjacency matrix*.

- **Undirected** if $\mathcal{W} = \mathcal{W}'$, i.e. if the weight matrix is symmetric.
- **Simple** if it is both unweighted and undirected, and furthermore the weight matrix (or adjacency matrix) $\mathcal{W}$ is zero-diagonal, meaning that the graph has no self-loops.

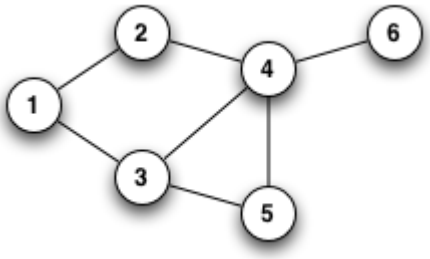

Figure 2.3. An example of a simple graph (unweighted and undirected) [29].

Another useful definition is the **neighborhood** $\mathcal{N}_i$ of a node i as the set of nodes directly linked to i with an edge. In a directed graph, we use to differentiate between *in-neighborhood* and *out-neighborhood*, depending on the position of the node i (end-point and starting-point of the edge respectively).
Assuming to have an unweighted graph, if we count the number of link starting or ending in each node, we obtain respectively the *out-degree* w_i^{out} and the *in-degree* w_i^{in}:

$$w_i^{out} = \sum_{j \in \mathcal{V}} \mathcal{W}_{ij}, \quad w_i^{in} = \sum_{j \in \mathcal{V}} \mathcal{W}_{ji}. \tag{2.3}$$

It is quite easy to verify that for an undirected graph the two quantities in 2.3 are the same (i.e. the graph is *balanced*) and we can define in a natural way the **degree** of a node i as $w_i = w_i^{out} = w_i^{in}$.
For example, the vertices of the unweighted and undirected graph in 2.3 has:

$$w_1 = 2, \quad w_2 = 2, \quad w_3 = 3, \quad w_4 = 4, \quad w_5 = 2, \quad w_6 = 1.$$

2.2.1 Some examples: cycle and ring lattice

A simple graph with n vertices and whose edges form a polygon is called **cycle**(n).

In a cycle every node has the same degree, equal to 2 (i.e. $|\mathcal{N}_i| = 2$, where $\mathcal{N}_i$ is the neighborhood of a node i in an undirected graph), because the edges link each vertex to its nearest vertices, one from the left and the other one from the right. But what happens if we want to link each node not only yo the first nearest vertex, but also to the second

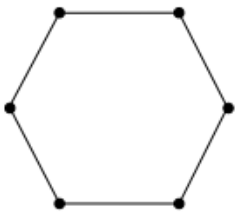

Figure 2.4. A cycle with $n = 6$ nodes.

one, to the third one, and so on?
In that case, we obtain a **ring lattice**. This type of graph is similar to the cycle but each node is linked with an edge to its k (constant for every node) nearest vertices, such that $\frac{k}{2}$ are on its left and the other $\frac{k}{2}$ are on its right. Furthermore, also in this graph each node has the same degree, equal to k.

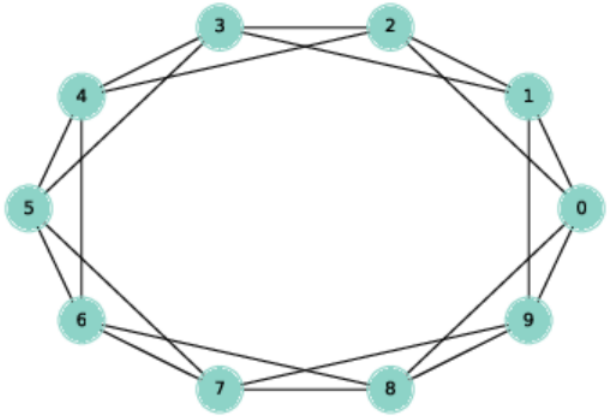

Figure 2.5. A ring lattice with $n = 10$ and $k = 4$ nodes [20].

2.2.2 Watts-Strogatz graph

The Watts–Strogatz model [38] is a random graph generation model [36], meaning that we have a probability distribution over a graph, or a random process used to generate the graph.
The Watts-Strogatz graph depends on 3 parameters:

- The number of nodes n;

- The mean degree k (more often an even integer) satisfying $n \gg k \gg ln(n) \gg 1$;
- A parameter β satisfying $\beta \in [0,1]$.

The model constructs an undirected graph with n nodes and $\frac{nk}{2}$ edges, in two steps:

1. Construct a regular ring lattice with n nodes, each one connected to the nearest k neighbors, $\frac{k}{2}$ on each side.
2. Once the first step is done, we take every edge and we change it with probability β, making it connect two different nodes than before.

This type of graph is particularly useful when we want to represent a social network community, in fact it is quite clustered [38], that is each node (person) is linked with the majority of his neighbor nodes in the ring lattice, as well as a person is linked with her family or friends. Clearly, there is always some "unusual" link, maybe some far friend known during summer, or at the university, or anything else.

2.2.3 Time-varying network

A temporal network [37], also known as a **time-varying network**, is a network whose links are active only at certain points in time and each link carries information on when it is active; Time-varying networks are of particular relevance to spreading processes, like the spread of information and disease, since each link is a contact opportunity that an individual has at a certain time instant t.
One of the common representations for time-varying network data is denoted as "*contact sequences*": if the duration of interactions are negligible, the network can be represented as a set C of contacts (i, j, t) where i and j are the nodes and t the time instant when the interaction occurs; Alternatively, it can be also represented as an edge list E where each edge e is a pair of nodes and has a set of active times $t_{\text{edge}} = \{t_1, t_2, \ldots, t_T\}$, where T is the time horizon of interest, i.e. the very last time step in which the network is analyzed. A social network can be seen in some way as a time-varying complete network (i.e. a network induced by a complete graph) where each link e between two individuals switches on only in the time instants stored in t_{edge}, but if we change for a second our point of view, we may also imagine it as a graph with a fixed set of nodes $\mathcal{V}$ (i.e. individuals) and a set of edges $\mathcal{E}$ (i.e. the physical or sociological interactions between pairs) varying at every time instant t. We can see an example in figure 2.6

In a time-varying network some edges (but not everyone) may of course be always activated or always not, meaning that they never change in time: this will be the case of the *contact layer* [39], which we are going to introduce in the next chapter, which describes the physical contacts happening between pair of individuals among all the population at each time instant t; in our contact layer some links will be fixed, describing for example the interactions one has with her family, colleagues or close friends every day, while other links will be changing at every time instant t in order to capture the occasional contacts one may have during the day, for example in the elevator or at the supermarket.

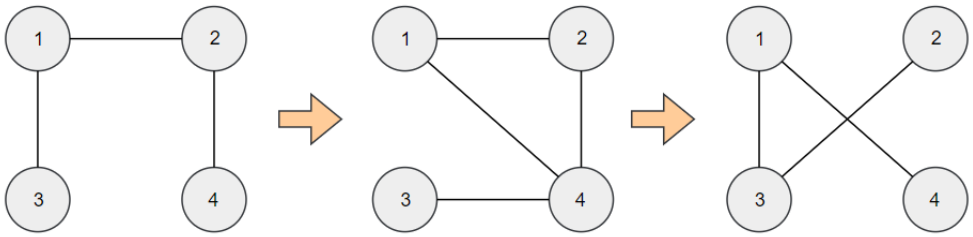

Figure 2.6. An example of time varying network for $T = 3$ with a fixed set of nodes $\mathcal{V} = \{1,2,3,4\}$ and a varying set of edges.

2.3 Network games

In these type of games players are represented as vertices of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and their utility functions depend only on their own and their out-neighbors' actions (or simpler neighbors, if we suppose to have an undirected graph).
Formally [3], a $\mathcal{G}$-game (i.e. a network game over the graph $\mathcal{G}$) is any triple $(\mathcal{V}, \mathcal{A}, \{u_i\}_{i\in\mathcal{V}})$ whose utilities satisfy the following property: for any player $i \in \mathcal{V}$ and any configurations $\mathbf{x}, \mathbf{y} \in \mathcal{A}^{\mathcal{V}}$ such that $x_j = y_j, \forall j \in \mathcal{N}_i \cup i$, it holds that:

$$u_i(\mathbf{x}) = u_i(\mathbf{y})$$

We can easily define the *network game* $(\mathcal{V}, \mathcal{A}, \{u_i\}_{i\in\mathcal{V}})$ over $\mathcal{G}$ by setting the utility of every player $i \in \mathcal{V}$ as the weighted sum of the utilities of the various two-player games that i is playing with his neighbors:

$$u_i(\mathbf{x}) := \sum_j W_{ij}\varphi^{(i,j)}(x_i, x_j) \tag{2.4}$$

If we have an unweighted graph, the formula (2.4) changes easily in

$$u_i(\mathbf{x}) := \sum_j \varphi^{(i,j)}(x_i, x_j).$$

2.3.1 Network games with binary actions: network coordination game

A simple case of network game is when the action set is $\mathcal{A} = \{0,1\}$, for example "to buy or not to buy", "to do or not to do", or something similar.
For the sake of simplicity, we can assume that the graph over which we have a network game is undirected and unweighted (it is quite easy to extend this type of game over a more complicate graph), and that the interaction utility function is the same on every edge:

$$\varphi^{(i,j)} = \varphi, \forall (i,j) \in \mathcal{E}.$$

A very popular example is the so called **network coordination game** (or *majority game*) [3]: consider an unweighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ and the interaction utility function:

$$\varphi(i,j) = \begin{cases} 1, & \text{if} \quad x_i = x_j \\ 0, & \text{if} \quad x_i \neq x_j \end{cases}$$

while the utility function for the player i (i.e. for the i-th node) in the configuration action x is simply given by the number of his neighbors playing his same action:

$$u_i(\mathbf{x}) = \sum_{j \in \mathcal{N}_i} \varphi(i,j).$$

where $\mathcal{N}_i$ is the set of neighbors of node i.
It is clear that in this type of game one individual (i.e. one node of the graph) aims to assume the same choice as the majority of her neighbors; for example, her neighbors could be her family or her friends, and we can think about the choice as a political thought, religion, or the simpler choice about where to go on holidays.
Given these information we can easily notice that the "consensus" configurations are Nash equilibria, such as the coordination game, but they are not the only, as it's shown in 2.7, where the 4 nodes of a cycle can be red (action 0) or blue (action 1).

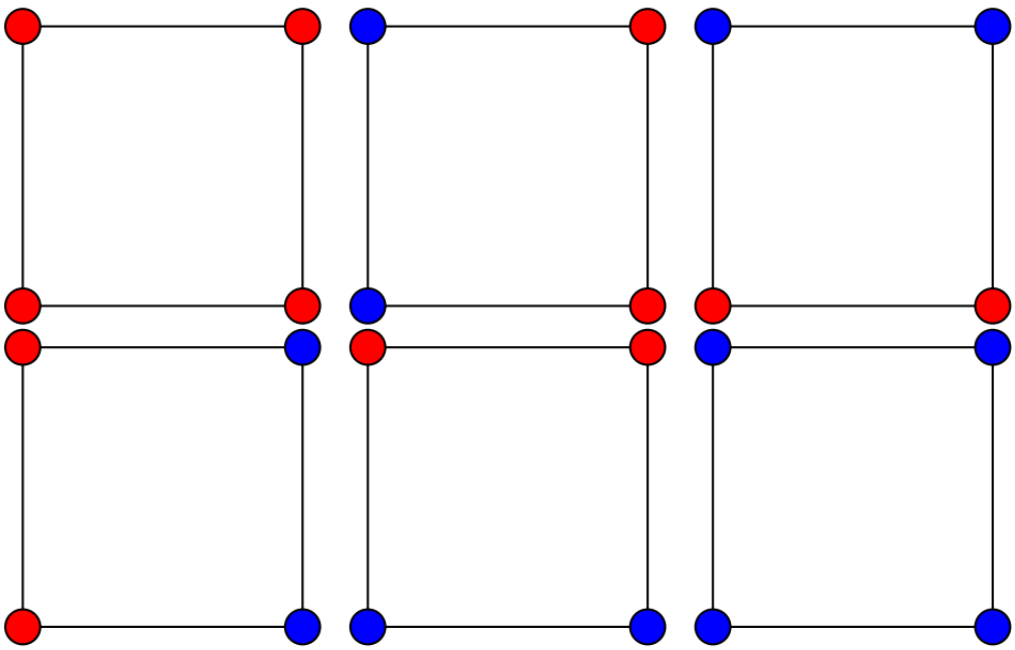

Figure 2.7. The majority coordination game on a cycle graph with $n = 4$ nodes has 6 Nash equilibria [3].

2.4 Epidemic models

2.4.1 Discrete-time Markov chains

Let us start with a motivating example. Imagine to be a gambler who wants to bet 1 euro on the roll of a regular 6-sided dice: if you win, you gain 1 euro, while at the same time if you lose you lose 1 euro. We are aware that this is not an advantageous game for the gambler, because he will win one single dice roll with probability $p = \frac{1}{6}$ and lose with probability $q = 1 - p = \frac{5}{6}$.
Suppose the gambler starts the game with a budget of k euro, and he left the game only when he reaches the amount of n euro (obviously $n > k$) or when he losees everything, remaining with 0 euro. We can visualize this game as a *random walk* over a directed graph, exactly like in 2.8.

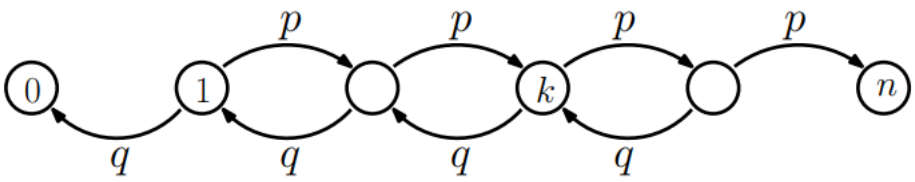

Figure 2.8. The gambler's capital can be represented as a random walk on the directed graph above [3].

The gambler game is a very simple example of a **discrete-time Markov chain**, that is, once we have a state space properly defined, a discrete-time process over a directed graph (also called *chain*) having the *Markov property*, which we will discuss in a while.

Given a weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, imagine you are able to move from one of its nodes to another: each time step t you randomly choose the next node to move to among the out-neighbors of the current node with probability proportional to the weight of the edge pointing from the current node to next one.
Formally [3], what we are describing is a discrete-time stochastic process $X(t), t = 0, 1, \ldots,$ with state space $\mathcal{X}$ coinciding with the vertices set $\mathcal{V}$ of the graph $\mathcal{G}$.
The state space is such that, for any states i and j in $\mathcal{X}$:

$$\mathbf{P}\left(X(t+1) = j | X(0) = i_0, X(1) = i_1, \ldots, X(t-1) = i_{t-1}, X(t) = i_t\right) = P_{i_t,j} \tag{2.5}$$

where $P = D^{-1}W$ is the normalized weight matrix of the graph $\mathcal{G}$.
Equation (2.5) states that the future state $X(t+1) = j$ is completely independent from the past history of the stochastic process, when conditioned to the present state $X(t) = i_t$.
The property we have just stated, i.e. the fact that the dependence of the future state is

limited to the present state is known as the **Markov property**, and the stochastic process $X(t)$ is usually referred to as a *discrete-time Markov chain* with *transition probability matrix* P.
In fact, a discrete-time Markov chain $X(t)$ with finite state space $\mathcal{X}$ can be associated with a stochastic matrix P, i.e. a non-negative square matrix whose entries are labeled by the corresponding elements of $\mathcal{X}$ and whose rows all sum up to 1, and an initial probability distribution $\pi(0)$, i.e. a probability vector whose entries correspond to the elements of $\mathcal{X}$ such that

$$\mathbf{P}(X_0 = i) = \pi_i(0), \quad \forall i \in \mathcal{X}.$$

On the other hand, we can also notice that a stochastic matrix P can always be interpreted as the normalized weight matrix of some (in general, weighted and directed) graph. To do that, it is sufficient to consider the graph $\mathcal{G}_P = (\mathcal{V}, \mathcal{E}, P)$ whose node set $\mathcal{V} = \mathcal{X}$ coincides with the state space, whose link set $\mathcal{E}$ consists of the pairs of nodes $(i, j) \in \mathcal{V} \times \mathcal{V}$ for which $P_{ij} > 0$, and whose weight matrix coincides with matrix P. The graph $\mathcal{G}_P$ is usually referred to as the *normal graph* associated to the matrix P.

2.4.2 SI and SIS models

In this thesis we will consider epidemic models in a discrete time implementation, meaning that at every discrete time instant t one individual has a specific probability to move from one (health) state to another, because they are predisposed to be studied through simulations.

The **SI model** is the simplest example of a pairwise interacting network system, meaning that his dynamic and the epidemic diffusion spreads due to the interaction (say contact) between pairs of individuals. In the SI epidemic model the state space for each individual is $\mathcal{X} = \{S, I\}$, i.e. each individual can be Susceptible or Infected, meaning that the global state space is $\mathcal{X}^{\mathcal{V}} = \{S, I\}^{\mathcal{V}}$; supposing that there is no spontaneous mutation, a susceptible can change her state only by having a contact with an infected. At every time step, each individual susceptible, for every contact with an infected, has a probability $\lambda \in (0,1]$, to become infected herself. With this type of model, it's very easy to notice that the whole population is going to become infected, soon or later, that is:

$$\lim_{t \to +\infty} S(t) = 0, \quad \lim_{t \to +\infty} I(t) = n.$$

where we call $S(t)$ and $I(t)$ respectively the number of people susceptible and infected, and n the size of the population.

If we introduce the possibility for an individual in state I to recover, and thus come back spontaneously in state S with a probability $\mu \in (0,1]$ every time step, we obtain the **SIS model** like figure 2.9. Since we are in a discrete time Markov chain setting, we have to say that both λ and μ are independent from the story past and also independent from one individual to another. Furthermore, we must differentiate the two probabilities: λ is a per-contact probability, that is a susceptible individual who interacts with an infected one has probability λ to become infected, while μ is the probability for each infected to

re-become susceptible at every time step.
One of the modeling limitations of the SIS model is that we don't have "immune" people, meaning that one can get infected infinitely many times. This trouble make us thinking about introducing a new (*absorbing*, i.e. a state from which nobody can move, no matter when) state R for the recovered people.

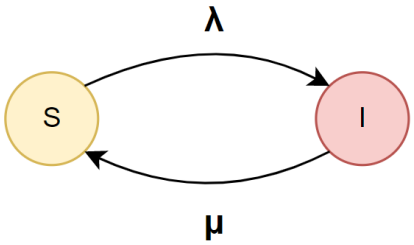

Figure 2.9. The state transition graph a single individual in the SIS model.

2.4.3 SIR model

In the **SIR model** (Susceptible-Infected-Recovered) epidemics, nodes can be in one of three possible states: susceptible (S), infected (I), or recovered (R), so that $\mathcal{X} = \{S, I, R\}$ (sometimes the notation $\mathcal{X} = \{0,1,2\}$ is used [3]). Exactly as in the SI and in the SIS epidemic models, also in the SIR epidemics infected nodes j may infect their susceptible neighbors i every time step independently with probability λ. On the other hand, infected nodes spontaneously recover independently every time step with probability μ, similarly to the SIS epidemics, with the crucial difference that recovered nodes are not susceptible anymore but rather remain in state R ever after; λ and μ are again supposed to be independent from the story past and from one individual to another.
In general we will refer to the health status of an individual i as $X_i(t) \in \{S, I, R\}$, while $S(t)$, $I(t)$ and $R(t)$ are the total number of people susceptible, infected or recovered respectively. The dynamics described above induce Markov chains for the stochastic processes $S(t)$, $I(t)$ and $R(t)$, in which individual transitions are represented in figure **??**.

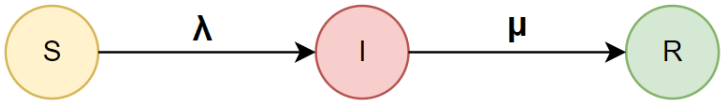

Figure 2.10. The state transition graph for a single individual in a simple SIR model.

Again, just like the SI model and provided that $\lambda, \mu > 0$, if we try to calculate the limit for t that goes to infinity we will find that all the population n is grouped in a single state R, that is the absorbent one:

$$\lim_{t \to +\infty} S(t) = 0, \quad \lim_{t \to +\infty} I(t) = 0, \quad \lim_{t \to +\infty} R(t) = n.$$

where $R(t)$ is the number of people recovered at time t.
More details on epidemic models can be found in the following survey papers [40].

Chapter 3

First wave model

The research community has been studying the spread of an epidemic for a long time, and never before has interest in this area been so high as in recent years, due to the outbreak of the COVID-19 pandemic; the years during the COVID-19 have been really hard for most people, and health authorities have always tried by every means to intensify advertising campaigns in order to sensitize the population in the adoption of prudent behaviors to limit the spread of infections, such as for example wear a protective mask, disinfect your hands often and avoid unnecessary physical interactions with strangers as much as possible. However, classical mathematical epidemic models often consider oversimplified behavioral response [19]; to fill in this gap, awareness-based models have been proposed in the last decade [12, 21, 24, 23], in which the epidemic process co-evolves with the spread of the awareness of the outbreak. While these models have demonstrated effectiveness in capturing the early stage of the pandemic they are limited because they assume fully rational and purely instantaneous decision making in the population. Such models therefore fail to capture the very range of factors that affect real-world behavioral responses over the whole course of an epidemic, such as social influence from parents and neighbors, perceived infection risk, accumulating frustration and socioeconomic costs, and last but not least the impact of government interventions; what can we do in order to capture also this very important aspect?

The model introduced by Ye *et al.* in their paper "*Game-theoretic modeling of collective decision making during epidemics*" [39] aims to capture the individual responses and time-varying contagion patterns, whose co-evolution collectively shapes the epidemic outbreak; their study takes into account most salient factors that each individual trades off when deciding their time-varying behavioral response to an ongoing epidemic.
The approach used in [39], like the title suggests, is to apply game theory for modeling human behavior, i.e. the decision whether to be prudent wearing the mask and isolating themselves or not. In [39], the authors proposed their game-theoretic framework coupled with a simplistic SIS model; here, we extend such a framework toward capturing real-world epidemics. Specifically, we will adapt it to the COVID-19 disease, fitting the model with the real data captured in Italy during such pandemic.

3.1 The original behavioral-epidemic model

Assume to have a population $\mathcal{V}$ of size n, where each individual $i \in \mathcal{V}$ is characterized by a two-dimensional variable $(X_i(t), Y_i(t))$ [39], which models their *social behavior* and *health state* at the discrete time $t \in \mathbb{N}^+$, respectively.
The **social behavior** of individual $i \in \mathcal{V}$ is captured by the binary variable $X_i(t) \in \{0, 1\}$, which describes whether individual i adopts self-protective behaviors ($X_i(t) = 1$), or not ($X_i(t) = 0$).
The **health state** $Y_i(t)$ also takes values in a discrete set $\mathcal{Y}$, for example we can have $\mathcal{Y} = \{S, I\}$ in the SIS epidemic process introduced section 2.4.2. Furthermore, a global observable $Z(t)$ quantifies the spread of the epidemic at time t, in formula

$$Z(t) := \frac{1}{n}|\{i : Y_i(t) = I\}|,$$

counting the fraction of the whole population being infected at time t.
The decision making and disease spreading in the population coevolve, mutually influencing each other on a two-layered network $\mathcal{G} = (\mathcal{V}, \mathcal{E}_I, \mathcal{E}_C(t))$, as schematized in 3.1.

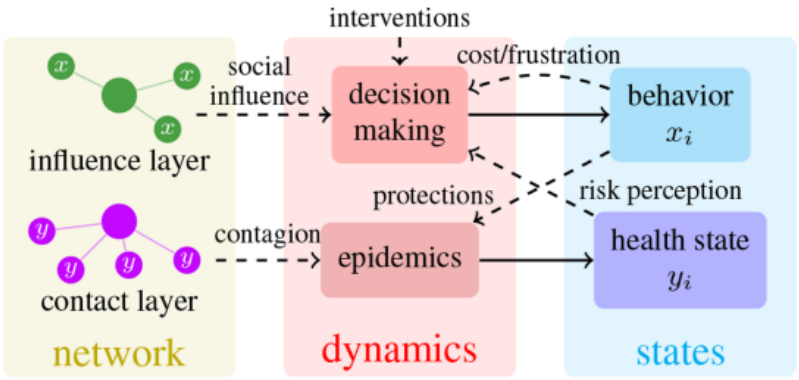

Figure 3.1. Schematic of the co-evolutionary paradigm, taken from [39].

The set of undirected links $\mathcal{E}_I$ defines the static **influence layer**, composed for example by the people in their life that have strong social influence on their decision-making processes, like parents, other relatives or close friends. The **contact layer** $\mathcal{E}_C$ is defined instead through a time-varying set of undirected links, which represent the people with whom one individual has physical contacts at a given time step t: this layer is very important because the contacts between pairs of people are the avenues for the transmission of the disease. In [39], the contact layer is created completely at random, while in this thesis it is thought as an extension of the influence layer: it sounds natural indeed that most of the physical contact an individual has during the day are the ones with her relatives, especially during a lockdown due to the epidemic.

3.1.1 Behavioral decision-making process

At each discrete time step t, every individual $i \in \mathcal{V}$ enacts a decision-making process on the adoption of self-protective behaviors and measures, according to an evolutionary mechanism based on game theory. Two payoff functions (or utility functions) were defined, $\pi_i^0(t)$ and π_i^1 which represent a good mix of sociological, psychological, economic, and personal benefits received by individual i for enacting behaviors $X_i(t) = 0$ or $X_i(t) = 1$ at time t, respectively. In [39] these payoff functions are defined as:

$$\pi_i^0(t) := \frac{1}{d_i} \sum_{j:(i,j)\in\mathcal{E}_I} [1 - X_j(t)] - u(t), \tag{3.1}$$

$$\pi_i^1(t) := \frac{1}{d_i} \sum_{j:(i,j)\in\mathcal{E}_I} X_j(t) + r(Z(t)) - f_i(t), \tag{3.2}$$

where $d_i = |\{j : (i,j) \in \mathcal{E}_I|$ is the degree of node i on the undirected graph representing the influence layer. The two functions in (3.1) and (3.2) contain some interesting terms:

- The first term in (3.1) and (3.2) (i.e. the summation weighted by the node degree d_i) is the so-called **social influence** inspired by network coordination games [3, 27] discussed in section 2.3.1; this term captures the social influence that arises from the neighbors of individual i, giving her the desire to coordinate her behavior with them. Since $X_j(t)$ can assume only the binary values 1 or 0 (i.e. to behave prudent or not), this mathematical formulation allow us to say that as more of the neighbors of individual i adopt self-protection or do not adopt self-protection, then individual i also has more incentive to adopt or not.

- The **risk perception** function $r(Z(t)) : [0,1] \to \mathbb{R}^+$ is a monotonically non-decreasing function of the *detectable prevalence* $Z(t)$ (i.e. the percentage of the population infected at time t). Here, we assume that the risk perception function is linear in the prevalence, that is $r(Z(t)) = kZ$, with a scaling factor $k > 0$; the larger k, the faster is the population to react to an outbreak.

- The **frustration function** $f_i(t)$ quantifies in some way the cost of long term self-protective behaviors for individual i: adopting self-protections could indeed be source of frustration if it is done for a long time, introducing the risk that people refuse to continue behaving in a prudent way, for example by wearing mask or by maintaining social distances. The frustration function in formula is:

$$f_i(t) = c + \sum_{s=1}^{t} \gamma^s c X_i(t-s), \tag{3.3}$$

  where $c \geq 0$ quantifies the social, economic and psychological immediate cost for individual i in adopting self-protective measures, due to the inability to socialize, work from the office, or anything else; another parameter $\gamma \in [0,1]$ is the accumulation factor: the more γ increases, the more impact on the payoff of all past decisions increases, reflecting the accumulating nature of fatigue, stress, and economic losses

in each individual.
In general both c and γ could be heterogeneous, say different from one individual to another, but for simplicity we assume from now on that the whole population has the same immediate cost c and accumulation factor γ.

- The last term in (3.1) is the quantification of **policy interventions**, that is all the laws and restrictions (see lockdowns or advertising campaigns for the awareness of the population in wearing the protective face mask) the government applies in order to reduce physical interactions between people during the epidemic spread and discouraging dangerous behaviors. The function $u(t)$, also called *non-pharmaceutical interventions* (NPIs), is a time-varying function reasonably thought as piecewise constant, and it will be a crucial element in the next pages of this thesis, because it is properly an element thanks to which we can apply *control* over the epidemic model. Later, we will try to optimize this function in order to reduce the epidemic spread and people deaths.

The decision process is stochastic rather than deterministic, therefore individual i adopts self-protective behaviors (i.e. $X_i(t+1) = 1$) with a probability equal to

$$\mathbb{P}(X_i(t+1) = 1) = \frac{e^{\beta \pi_i^1(t)}}{e^{\beta \pi_i^0(t)} + e^{\beta \pi_i^1(t)}} \tag{3.4}$$

where the parameter $\beta \in [0, +\infty]$ measures individual's rationality in the decision-making process. In [39] is assumed that β is homogeneous among all individuals, but (3.4) could be easily generalized using heterogeneous β_i. Anyway, in this thesis has been set a common level of rationality $\beta = 6$ for the whole population, which captures a moderate level of rationality so that individuals tend to maximize their payoff, but still having a small but non-negligible probability of choosing the action with the lower payoff. Finally we can notice that if $\beta = 0$, then individuals make decisions uniformly at random, while on the other hand if $\beta \to +\infty$ individuals always chose the behavior with highest payoff, no matter how much is the lower payoff.

In order to simplify the model, we implemented a new **payoff function** $\Pi_i(t)$ for individual i as the difference between $\pi_i^1(t)$ and $\pi_i^0(t)$ (in (3.2) and (3.1)), so that at every time step t we have the following payoff function for the game-based decision on the adoption of self-protective behavior for individual i:

$$\begin{aligned}
\Pi_i(t) &= \pi_i^1(t) - \pi_i^0(t) = \\
&= \frac{1}{d_i} \sum_{j:(i,j)\in\mathcal{E}_I} X_j(t) + r(Z(t)) - f_i(t) - \left(\frac{1}{d_i} \sum_{j:(i,j)\in\mathcal{E}_I} [1 - X_j(t)] - u(t) \right) = \\
&= \frac{\#_i^1 - \#_i^0}{d_i} + r(Z(t)) - f_i(t) + u(t) = \\
&= \frac{\#_i^1 - \#_i^0}{d_i} + kZ(t) - c\left(1 + \sum_{s=1}^{t} \gamma^s x_i(t-s)\right) + u(t), \quad \forall i = 1,2,\dots,n.
\end{aligned} \tag{3.5}$$

where $\#_i^1$ and $\#_i^0$ are respectively the number of node i's neighbors adopting or not adopting self-protective measures, and d_i is the degree of node i in the influence layer.
We could have had a deterministic and simpler decision if we had assumed that individual i at time t would choose action 1, hence having $X_i(t) = 1$, if $\Pi_i(t) \geq 0$ (i.e. if $\pi_i^1(t) \geq \pi_i^0(t)$) and would choose action 0 otherwise. Anyway, this decision process is too simple and doesn't reflect the real behavior of people, who very often don't act in a perfectly rational and schematic way; it is surely better to implement a probabilistic decision like in [39], but since we have only one payoff function, the stochastic decision becomes more easily:

$$\mathbb{P}(X_i(t+1) = 1) = \frac{e^{\beta \Pi_i(t)}}{1 + e^{\beta \Pi_i(t)}} \tag{3.6}$$

3.1.2 Epidemic dynamics

Concurrently with the behavioral decision, at each time step t, every individual $i \in \mathcal{V}$ that does not adopt self-protections and is susceptible (i.e., $X_i(t) = 0$ and $Y_i(t) = S = 0$) and gets in contact with an infected individual j ($Y_j(t) = I = 1$), has a per-contact infection probability $\lambda \in [0, 1]$. Obviously the more physical contacts (denoted by $n_i(t)$) individual i has at time t, the higher is her total infection probability $\lambda_i(t)$.
Individuals who adopt self-protective behaviors have a decreased probability to be infected; we capture this by introducing a parameter $\sigma \in [0, 1]$, representing the effectiveness of self-protective behaviors in preventing contagion, assuming on the other hand that the adoption of self-protection does not affect the individual's probability of transmitting the disease. Hence, the contagion probability for a susceptible individual $i \in \mathcal{V}$ as function of time is:

$$\mathbb{P}(Y_i(t+1) = I | Y_i(t) = S) = (1 - \sigma X_i(t))(1 - (1 - \lambda)^{n_i(t)}) \tag{3.7}$$

where $n_i(t) = |\{j \in \mathcal{V} : (i, j) \in \mathcal{E}_C(t), Y_j(t) = I\}|$ is the number of infectious physical contacts of node i at time t.
Exactly like the SIS model, at every time step t each infected individual i recovers with probability $\mu \in (0, 1]$, becoming susceptible again to the disease, that is:

$$\mathbb{P}(Y_i(t+1) = S | Y_i(t) = I) = \mu.$$

In this thesis, we've introduced also the Death state D (an obviously absorbing state, corresponding to a kind of Recover state in a SIR model), since during COVID-19 one of the main focus was on the mortality of this virus, especially in weak people. The death probability for an individual i infected at every time step t is δ, i.e.:

$$\mathbb{P}(Y_i(t+1) = D | Y_i(t) = I) = \delta.$$

3.2 Model calibration and validation

The first goal of this thesis is to properly extend the modeling framework in [39] adapting it to the COVID-19 epidemic in Italy during the months of the first wave, that is, from the end of February 2020. In order to pursue this, we collected the real data of officially

reported cases and fatalities stored in the GitHub database [7] owning to the *Presidency of the Council of Ministers - Department of Civil Protection.*
The database collects data on COVID-19 in Italy from February 24, 2020, and it stores several useful data like the total number of positive cases, the number of daily swabs, how many hospitalized people, the number of daily deaths, and many other; what is interesting for our analysis is obviously the cumulative number of deaths and the daily total number of people positive to covid-19.
Since the database collects daily data, we will discretize time by setting a time step equal to 1 day to easily simulate the epidemic dynamics with our model. Furthermore, it's not difficult to understand that for a common portable computer it would be an infeasible effort to simulate a model by considering a population of about 60 million people like Italy is; for this reason, we have scaled down the population to $n = 1000$.

The first thing we need is an undirected graph called influence layer $\mathcal{E}_I$ modeling the strong connections between very close people: family, partner, close friends, schoolmates or colleague. The influence layer $\mathcal{E}_I$ is assumed to be fixed during the entire the time horizon, and this is a fairy assumption because in those days of early pandemic people used to be close to the same few people.
Trying to represent the real world more faithfully as possible we have created the influence layer by using a very useful function [16] for creating a Watts-Strogatz graph, introduced in 2.2.2, with $n = 1000$ nodes, $k_{\mathcal{E}_I} = 5$ edges on each side of the nodes (meaning that, before applying a shuffling of the arcs thanks to the parameter β, every node has $2k_{\mathcal{E}_I} = 10$ neighbors, corresponding to their family components or close friends) and the parameter $\beta = 0.15$ which is the probability for every link to be changed and replaced with another one completely at random. The influence layer $\mathcal{E}_I$ is fixed at the beginning and never change in time, reasonably corresponding to reality, just think of the lockdown periods in which the people we interacted with were practically always the same.

We need to put attention also to the creation of the *contact* layer $\mathcal{E}_C$. According to the study in [17], almost half of the contacts one individual has in a day happen at home, at school or at work; the other half is composed by temporary and unusual contacts, for example at the supermarket, in the elevator or while we are on the public transport.
Consistently, we create a contact layer where the first half is exactly the influence layer $\mathcal{E}_I$, and the second half is chosen completely at random, with no restrictions other than that of having no self-loops; this means that, even if the probability of this event is quite low, some of the random contacts created in the second half of the contact layer may be the same of the influence layer $\mathcal{E}_I$, so that they are already existing in the first half of the contact layer $\mathcal{E}_C$. This is not a problem for us: we could have avoided this possibility by inserting specific constraints, but thinking more deeply it's quite reasonable that a particular interaction between two people (for example boyfriend and girlfriend) is stronger than others had during the same day, or it is also possible to have contact twice with the same friend, in fact this kind of contact layer is such that those stronger links are doubled and have the double contagion probability for the two individuals included. With this kind of contact layer, each individual has 20 physical contacts during one single day (if instead some links are doubled, i.e. made stronger, the degree of that node decreases,

but this happens quite rarely), and this is in perfect agreement with the study conducted in [17], where an heterogeneous sample of Italian people results to have on average 19.77 contacts per day.
While the influence layer (and so the first half of the contact layer) is fixed, the second half the contact layer $\mathcal{E}_C$ is instead varying in time since it represents the physical contacts one individual has within 24 hours, hence we simulate it at every time t; the contact layer should indeed be indicated as $\mathcal{E}_C(t)$, because it is a time-varying network (2.2.3) different from one time instant t to another.

3.2.1 Parameters setting

In order to validate our model, we need to think about some parameters to ensure that the model reflects the true trend of the epidemic; we have three kind of parameters: *setting* parameters, *epidemic* parameters, *behavioral* parameters.
Let's start with the setting parameters, useful to bring the model closer to reality, making it comparable in terms of proportions and dates:

- We set the **size of population** to $n = 1000$, which is a good tradeoff between sample size and computational effort, making the simulation times quite bearable. For the calculation we have considered an approximation of 60 million as the population size of Italy, even if it isn't obviously the exact number, since this is varying day after day.

- The **time horizon** T has been set equal to 100 days: this is of course a choice dictated by the need to obtain simulations in a reasonable computational time, but not only, because if we count one hundred days from February 24, 2020, we arrive until the first days of June 2020, just before the beginning of summer. For those who remember that summer, it was the first after months of lockdown and the vast majority of people (and locals) were fed up with the restrictions, and tried to fully enjoy the hottest months of the year on the beach with friends: the infections during that period were really much lower than the previous spring (and above all also compared to the autumn 2020 that was about to arrive), both because with the warmer climate the virus struggled to spread, and because during the summer many people did not they did swabs despite developing COVID-19 symptoms, to avoid having to do a quarantine that deprived them of going out. For this reason, the data collected relating to positive cases during the summer could be underestimated, or in any case altered, and we therefore decided to set a time horizon of $T = 100$ days. In further applications described in the following of this thesis (chapter 4), we will consider longer time horizon.

- The **detection rate** DR indicates the fraction of positive people founded thanks to the pharmaceutical swabs: during the first wave of pandemic, the swabs were available in very poor quantities, and this is why most of the infected weren't detected and didn't contribute to the stored data in [7]. The study in [2] suggests that the detection rate in Italy during the first wave was about 0.07 (i.e. 7%), that is a very little part of the effective infected were detected; in those months, during the

first epidemic wave in Italy, the swabs availability as well as their price was a big problem, this is why very few people had a swab when they felt sick and manifested symptoms.

- Even the **detection delay** (DD) is related to the difficulty in the detection of the positive cases of COVID-19: it is not a percentage, but an absolute number that indicates how many days passed on average from the positivity of a certain infected individual, with the consequent possibility of infecting his contacts, to the actual ascertainment of his positivity through the result of a swab. In [2] we can read that the DD in Italy during the very first months of pandemic was about 2 weeks: this is consistent if we think that in those months, when an individual became infected, perhaps a few days passed before the first symptoms appeared, then it was necessary to call the pharmacy to book a tampon that was carried out in the following days based on availability, and also to to receive the result of the swab (and therefore the notification of the actual positivity) it was necessary to wait further at least another 2-3 days. In line with this data, we have set a detection delay $DD = 14$ days.

Let's have a look at the epidemic parameters:

- The per-contact **infection probability** λ means that one individual susceptible (i.e. in state S) has probability λ to become infected (i.e. to move in the state I) *for each* physical contact with another infected individual, which we can see thanks to the contact layer.

- The **external infection probability** $\bar{\lambda}$ represents the probability of an infection due to something that is not directly contact, such as when we enter an empty elevator but in which dozens of people have just gone who may have left traces of the virus, or, if we think of our population as a small village of $n = 1000$ people, the fact that an individual leaves the village and comes into contact with someone from outside or from a foreigner; this probability, in addition to finding confirmation in what happens in the real world, is also fundamental from a purely modeling point of view because it guarantees us that the epidemic never dies, especially in the very first moments of time where the "*patient zero*", or the first individual to be infected (which is our initial condition), she could recover or die before having infected another person, thus immediately stopping the epidemic and making our analysis useless. For this reason, we set this non-zero probability to a reasonable value of $\bar{\lambda} = \frac{1}{n}$.

- The **recovery probability** μ is the probability for an infected individual to recover, i.e. to move out from state I at every time instant t, exactly like in the SIS model and SIR model (sections 2.4.2 and 2.4.3 respectively). In practice, this probability can't be evaluated directly, but we can though estimate it by considering the inverse of the mean number of days that one individual remains infected ($\frac{1}{\mu}$) [18].

- The **death probability** δ is the probability for an infected individual to die at every time instant t, as the name clearly suggests. For obvious reasons this probability assumes a key role in the lethality of a disease, and unluckily it was very high during

the first wave of covid-19, especially because it was not yet well known how to cure this disease and also because of the inadequacy of health facilities not yet ready to accommodate huge numbers of patients in intensive care. The death probability δ is applied not directly in the state I, but instead when an individual infected recovers: when individual i move out from stat I, meaning that she is no more infected, she can recover or die, and this split (see fig 3.2) is governed by the death probability δ.

- In some extensions of SIR model, there is the possibility for an individual recovered (i.e. in state R) to become again susceptible: this is the **unimmunization probability** φ. During the first $T = 100$ days of pandemic in Italy, almost nobody got infected twice, hence we can set $\varphi = 0$ for our first model, but we will need φ in chapter 4.

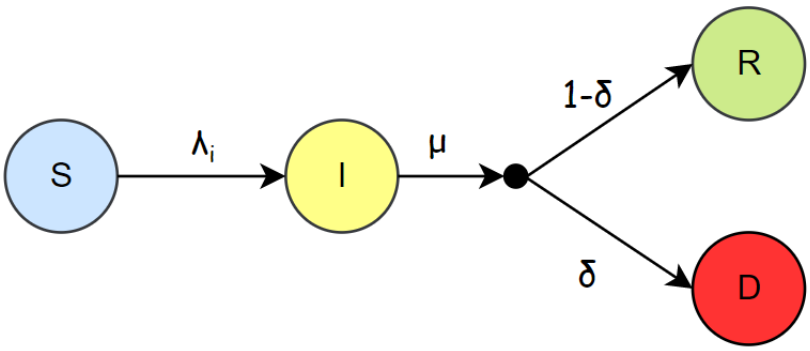

Figure 3.2. Discrete-time Markov chain for individual i's health at every time step t.
S: susceptible, I: infected, R: recovered, D: dead.

At the end we must introduce the behavioral parameters, most of which we are going to optimally calibrate in order to validate the model:

- The **effectiveness of self-protective behavior** σ is a crucial parameter; in fact, not all the people wear protective masks, or not correctly, furthermore maybe when one individual who thinks to behave prudent meets her friend can't resist to hug her and in doing so, social distancing is lost, this is why we think that a middle value of $\sigma = 0.5$ may be appropriate for the model.

- The people **rationality** in the decision-making process has been introduced as $\beta \in [0, +\infty]$ in equation (3.4), and according to [39] we can set a quite rational value of $\beta = 6$ for every individual i.

- The social, psychological and economic **immediate cost** $c \geq 0$ for individual i in adopting self-protective measures. This parameters appears twice in equation (3.3)

and finds a direct bond with reality: just think of when we are annoyed at always having to wear a mask with us when we leave the house, or the annoyance of having to cancel participation in a social event, or more concretely also the cost that a shopkeeper has in having to close his shop due to of the lockdown.

- The **accumulation factor** $\gamma \in [0,1]$, which we assume equal for everyone, is the impact of all our past decisions, reflecting the stress and fatigue protracted over time, such as economic cost (if we think about a restaurant), due to the continuous use of self-protective behavior.

- The **scaling factor** k in the risk perception $r(Z(t)) = kZ$ (3.5)has been thought for a pure mathematical reason: it's clear to everyone that people adopt self-protective behavior when the disease spreads too much, but how quickly are these prudent measures taken by the population? We will be able to answer this question only after a specific tuning of the parameters.

3.2.2 Simulation setting

The first natural goal of this work is to create an epidemic model that possesses a decision-making process based on game theory which is capable of faithfully reproducing real human behavior during the spread of the epidemic (in this specific case, of COVID-19 in Italy). In order to do this, we obviously need to plot some curves that describe the real trend of the epidemic in Italy, starting from 24 February 2020 (which refers to the first line of the dataset in [7]) for the following $T = 100$ days, and compare it with our simulated ones.

The two curves on which we will focus are the one of $Z(t)$ describing the spread of the disease in term of fraction of the population (this is needed in order to compare properly the Italian population size of 60 million people to our model's population size of 1000 people), and the one of $D(t)$ describing the fraction of population dead.
However, we must take into account the detection rate (DR) and detection delay (DD) introduced before, since they influence the shape of the curve and its values: this is another difference between our model and reality, because in our model simulations the count of positive cases (and then also the fraction of population infected) is done at the moment, while instead if we focus on the number of infected in real data at a certain date, this refers to DD days before that date; furthermore, the number of positive case at a certain date is not the effective number of infected, since the detection rate was very low in those months ($DR = 0.07$). Therefore if we name $\tilde{Z}(t)$ the real number of (current at time t) infected cases at time t directly taken from the Italian database [7], the plotted function of the real $Z_r(t)$ will be:

$$Z_r(t) = \frac{\tilde{Z}(t + DD)}{6 \times 10^7} \frac{1}{DR}, \quad \forall t \in [1, T]. \tag{3.8}$$

For what regards the fraction of dead $D_r(t)$ we should not scale it by using the detection rate because every dead individual is detected, and there is also no delay (DD), hence we

have $D_r(t) = \tilde{D}(t)$, where $\tilde{D}(t)$ is the fraction of people dead given by real data, similar to $\tilde{Z}(t)$.

Our basic goal will be to fit the real trend of $Z_r(t)$ and $D_r(t)$ by evaluating both in a quantitative way, that is by computing the SSE (sum of square errors) between the simulated curves and the real ones, and in a qualitative way, that is by looking at the plots: for our purposes, we will fit both $Z_r(t)$ and $D_r(t)$ but the main focus will only be on the infected trend, since the death trend will be more "random" because the very low death probability and population size make the death curve $D(t)$ to be more nervous even with only one more single death (or one less).
For what regards the initial condition, on the first day the real fraction of infected was $Z_r(1) = \frac{221}{6\times10^7} \approx 0.0000037$ and the real fraction of death was $D_r(1) = \frac{7}{6\times10^7} \approx 0.00000012$, but our model can't reach a so low precision, because at minimum we can have a fraction of $\frac{1}{n} = 0.001$; for this reason, we decided to set an initial state of:

$$Z(1) = \frac{1}{n} = 0.001, \quad D(1) = 0. \tag{3.9}$$

It is quite reasonable for us to set these values because our model should start with 0 dead and with the "patient zero" as the first infected individual among all the population (in our model we will set the first individual as the patient zero).
We may also assume that at the first day of pandemic nobody was wearing protective masks or adopting other type of self-protective behaviors, therefore we set $X_i(1) = 0$, $\forall i = 1,2,\ldots,n$.

3.2.3 Parameter tuning

We performed many simulations in order to evaluate how the model's population behave (i.e. adopt self-protective measures or not) and how is the COVID-19 spreading in our sample of $n = 1000$ individuals. This quite small number of people is the reason why the epidemic parameters in the model turns out to be very small.
We performed a grid optimization by simulating $SIM = 200$ different trajectories for every parameters configuration, at first with a coarse-grain approach in order to find a good range of reasonable values for each parameter, later by studying a denser grid of values, thus carrying out a more precise search in that good range of values that we had previously found. More in detail:

- The per-contact infection probability λ has been tested with several values between 10^{-3} and 10^{-1}: the optimal value found is $\lambda = 0.007$.

- The recovery probability μ has been tested with several values between 0.01 and 0.5: the optimal value found is $\mu = 0.14$, and this is also coherent with [18] where one can read that the mean infection time is about 5 days (i.e. μ should be around 0.2).

- The death probability δ has been tested with several values between 10^{-3} and 10^{-1}: the optimal value found is $\delta = 0.003$.

- The immediate cost of adopting self-protective behavior c has been tested with several values between 0 and 3 (it was useless to test values greater than 3 because nobody would choose to adopt self-protective measures with a such high immediate cost): the optimal value found is $c = 0.5$.
- The accumulation parameter γ has been tested in all its admissible range [0,1]: the optimal value found is $\gamma = 0.25$; we should remember this value because in the next chapter it will change for an interesting reason.
- The scaling factor k has been tested with several values between 0 and 20 (it was useless to test values greater than 20 because almost everyone would choose to adopt self-protective measures with a such high reaction to the epidemic spread): the optimal value found is $k = 12$.
- The government control function $u(t)$ has been fixed to $u(t) = 0.5, \forall t \in [0, T]$ and all the other parameters have been trained as a consequence of this value. This control will be changed later (3.3), when we will try to implement some different intervention strategies in order to reduce as much as possible the spread of the disease.

Parameter	Symbol	Tuning range	Optimal value
Per-contact infection probability	λ	$[10^{-3},10^{-1}]$	0.007
Recovery probability	μ	[0.01,0.5]	0.14
Death probability	δ	$[10^{-3},10^{-1}]$	0.003
Immediate cost	c	[0,3]	0.5
Accumulation factor	γ	[0,1]	0.25
Scaling factor	k	[0,20]	12

Table 3.1. Tuning parameter and optimal value summary for the first model without vaccine.

Each simulation took about 1 minute to be run, resulting in more than 3 hours for all the $SIM = 200$ trajectories; given all the trajectories, we calculated the average among the 200 realizations at every time instant t, obtaining the final curve for our model which we may call $Z(t)$; the same method applies for the fraction of death $D(t)$. We also calculated the *standard deviation* at every time t in order to define a proper **confidence interval** of the type:

$$\left[Z(t) - \frac{\text{std}(Z(t)) \times q_{0.95}}{\sqrt{SIM}}, \quad Z(t) + \frac{\text{std}(Z(t)) \times q_{0.95}}{\sqrt{SIM}}\right] \tag{3.10}$$

where $Z(t)$ is the average of the fraction of infected individuals among all the trajectories at time t, "std" stands for standard deviation and $q_{0.95}$ is the 95% probability quantile [35] for a standard normal distribution (more details on the standard normal distribution and confidence intervals can be found in [13]).

For the purpose of evaluating which parameter configuration is the best one, we computed

the SSE (*sum of square errors*) both for $Z(t)$ and $D(t)$; actually we did not considerate the latter one because in our model with $n = 1000$ people just one more death is enough to bust the curve changing a lot in the fraction of dead $D(t)$ (and one more death can be realized quite easily with our epidemic dynamic). Together with the SSE, a very good evaluation element is plotting the curve of $Z(t)$ and $D(t)$ near, respectively, $\tilde{Z}(t)$ and $\tilde{D}(t)$: this is clearly a qualitative approach but it must not be underestimated because the results are very surprising and near to the real data (see 3.3), meaning that the covid-19 epidemic spread of the first wave in Italy is very well fitted by our model.

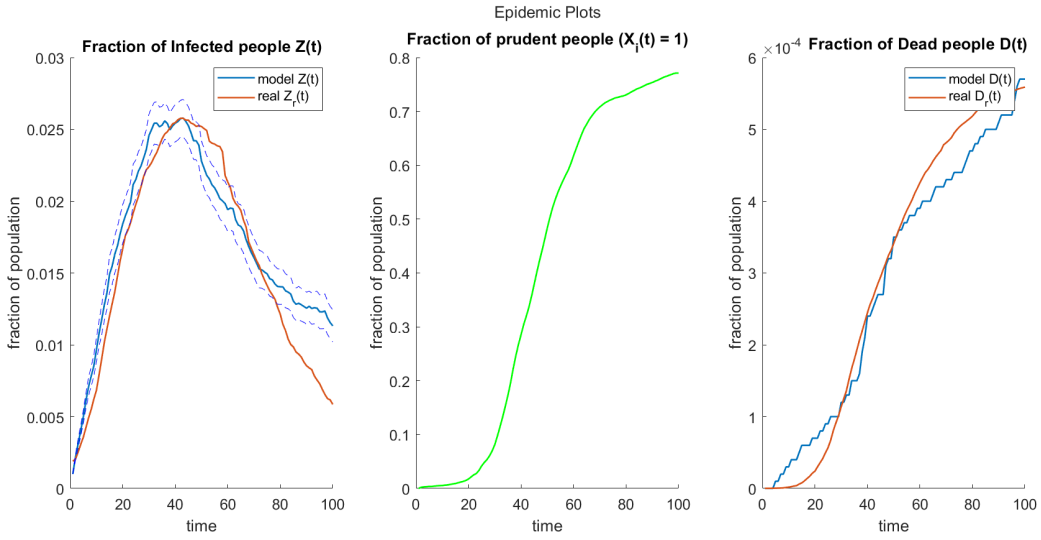

Figure 3.3. Validation of the model without vaccine, related to the first epidemic wave in Italy starting from 24^{th} February 2020.
The first picture on the left shows the trend of our model's $Z(t)$ in blue, with its 95% confidence interval (the dotted lines), compared to real $Z_r(t)$ taken from [7]; the green curve in the center describes the fraction of people adopting self-protective behavior at time t; on the right we can see the very accurate trend of deaths $D(t)$.

3.3 Optimization and control

Once we have validated the model, we can start investigating the role of the control function $u(t)$, representing the non pharmaceutical interventions (NPIs) that the government could apply in order to reduce the spread of COVID-19, such as complete or partial lockdown, protective mask wearing duty or other specific rules.
We could reasonably think that a single value for $u(t)$ during all the time period would not be an optimal solution, therefore we will try to find an optimal *changing point* $\tilde{t}$, that

is the time instant when we should change the value of the control function $u(t)$, hopefully by decreasing it.

3.3.1 The role of the control function $u(t)$

After having shown that our model is successfully able to reproduce the real world temporal evolution of the epidemic outbreak, we may try to increase, decrease or modify the function $u(t)$ with the reasonable aim of reducing infections and deaths in order to investigate a range of what if scenarios. A naive option could be to increase $u(t)$ a lot, for example by implementing very severe lockdowns as happened in China, but this would be a very desperate case and would have a dramatic impact on economy [14], because it would mean to close everything for an indefinite time period; in our model we are working from a purely epidemic point of view and we are not considering the economic loss that a national lockdown causes (and this is a very tough matter for those who are running the government and making decisions), therefore we will not consider this extreme option.

Anyway, it is clear that if we want to decrease the spread of the disease, fixed all the other parameters, we need to increase the strength of the NPIs (non pharmaceutical interventions by the government). We evaluated the different configurations by simulating $SIM = 50$ trajectories each and computing the average among the 50 trajectories; for each value of $u(t)$, in addition to a qualitative analysis of the contagion curve trend $Z(t)$, we saved three values of interest:

- The average fraction of people infected along the whole time period, which reflects the trend of the disease and number of infected.
- The average of the maximum values in each of the 50 simulation, which is a very good evaluation of the stress endured by the hospitals, which in those months had the intensive care wards always full and many positive patients were forced to other less equipped wards or even to stay at home, greatly increasing their probability of dying.
  Notice that the value obtained in this way may be different from the maximum value we can see in the results graph, but they both have the same interpretation.
- The fraction of dead people (and then the total number) at the end of the time period of interest.

The first approach aims to reduce the maximum value of the epidemic spread, trying to relieve the hospitals and the pressure that even the health personnel who were reduced to exhaustion in that period had to bear. The simulation results are shown in figure 3.4 and in table 3.2.

As expected, increasing $u(t)$ decreases the fraction of infected individuals, both in terms of maximum value and in average; the same reasoning can be applied also for $D(t)$, but we already said that this value is a bit nervous due to the small population size, hence its

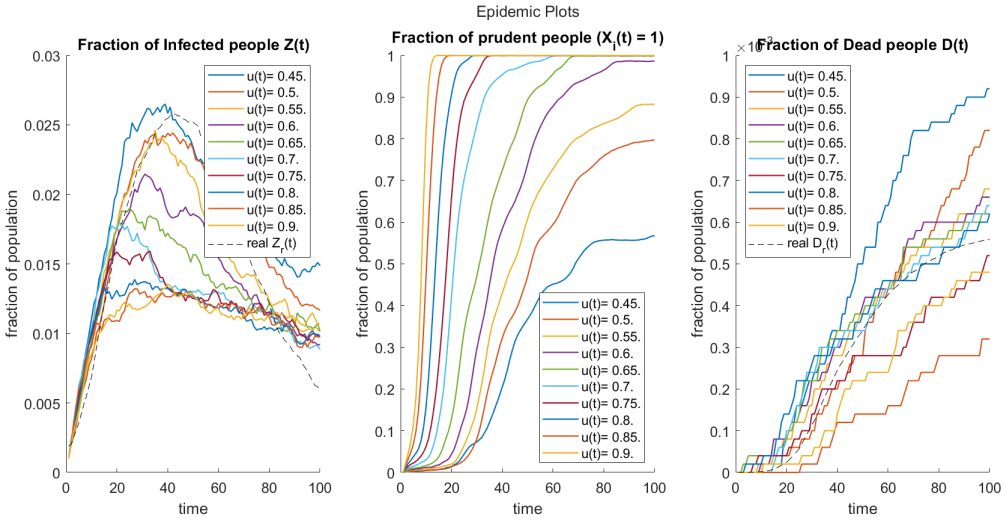

Figure 3.4. Testing several values for alternative $u(t)$ during the first epidemic wave in Italy.

$u(t)$	Avg prudent $X_i(t) = 1$	Avg $Z(t)$	Max $Z(t)$	Avg $D(100)$
0.45	0.3059	0.0186	0.0388	9.2×10^{-4}
0.50	0.4182	0.0169	0.0368	8.2×10^{-4}
0.55	0.4951	0.0155	0.0337	6.8×10^{-4}
0.60	0.5973	0.0141	0.0309	6.6×10^{-4}
0.65	0.6750	0.0132	0.0283	6.2×10^{-4}
0.70	0.7762	0.0123	0.0247	6.4×10^{-4}
0.75	0.8215	0.0119	0.0229	5.2×10^{-4}
0.80	0.8651	0.0112	0.0212	6.2×10^{-4}
0.85	0.9035	0.0110	0.0209	3.2×10^{-4}
0.90	0.9250	0.0109	0.0203	4.8×10^{-4}

Table 3.2. Several values for $u(t)$ were tested during the first epidemic wave in Italy. The first column stores the value of $u(t)$ applied to the model; the second column stores the average fraction of population adopting self-protective behavior (i.e. with $X_i(t) = 1$); the third column stores the average fraction of population infected $Z(t)$; the fourth column stores the maximum value of $Z(t)$, directly linked to the hospital stress; the fifth and last column stores the average fraction of population dead at time $T = 100$ among the 50 trajectories.
Each of the $n_{\text{row}} \times \text{SIM} = 10 \times 50 = 500$ simulations took around 40 seconds to be run, with a total time of around 5 hours and a half.

values could be affected by noise much more than $Z(t)$. If we look closely at the graph and read it from the top to the bottom lines we can see that the improvement seems sharp at the beginning and then instead fades towards the end: it seems to increase $u(t)$

over 0.75 (i.e. increasing the intensity of NPIs by 50%, passing from 0.5 to 0.75) does not bring such a marked improvement. So, we consider $u(t) = 0.75$ an excellent compromise between containing the epidemic and avoiding a complete lockdown, which would greatly damage the economy of the country.

3.3.2 Finding an optimal changing point

Even if we let the government avoid to apply too strict restrictions, a value of $u(t) = 0.75$ is however quite high and many people may think that maintaining such a high $u(t)$ for a long time could became a trouble both for the population happiness/mental health and for the long term national economy; furthermore, it may not provide beneficial effects. To solve this problem, like it really happens in reality, we should try to decrease the strength of NPIs after a certain date: we may extend the interest time from $T = 100$ to $T = 200$ in order to evaluate a longer impact of this policy, and we may arbitrarily set $\tilde{t} = 100$ as the **changing point** in which we modify the value of $u(t)$, clearly by decreasing it, such that the control function $u(t)$ becomes piecewise constant:

$$u(t) = \begin{cases} 0.75, & t \leq 100 \\ u, & t > 100 \end{cases} \tag{3.11}$$

We simulated 50 trajectories (computing the average, similar to as before) for each value of $u = 0.70, 0.65, 0.60, 0.55, \ldots$, in this order. We observed a great improvement at the beginning, while the results were quite similar for values smaller than $u(t) = 0.3$; noticed that this value was quite good, we then explore more deeply some other values around it, obtaining what is shown in Table 3.3 and in Figure 3.5.

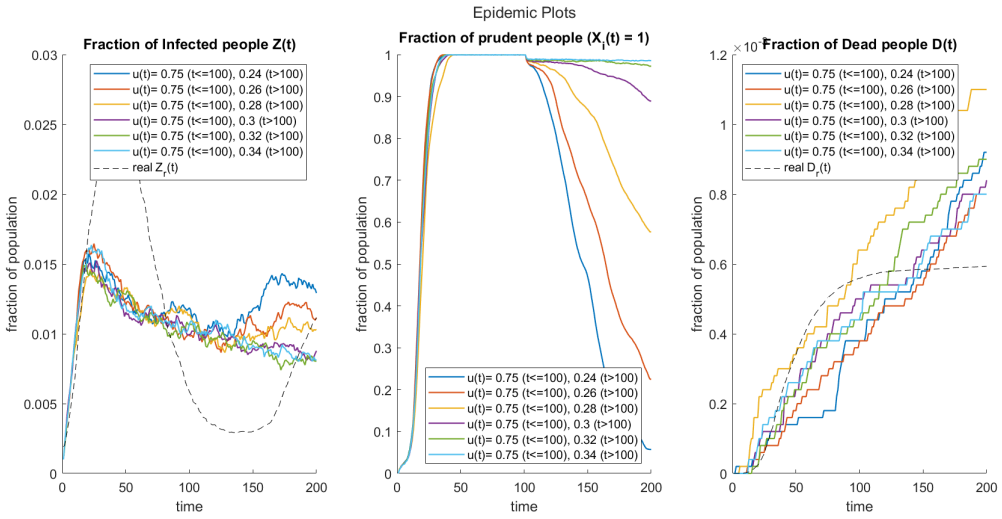

Figure 3.5. Testing several values for u when $t > 100$, fixed $u(t) = 0.75$ for $t \leq 100$.

$u(t), t > 100$	Avg prudent $X_i(t) = 1$	Avg $Z(t)$	Max $Z(t)$	Avg $D(200)$
0.24	0.6468	0.0120	0.0262	9.2×10^{-4}
0.26	0.7265	0.0112	0.0254	8.4×10^{-4}
0.28	0.8117	0.0108	0.0232	1.1×10^{-3}
0.30	0.8847	0.0103	0.0226	8.4×10^{-4}
0.32	0.8990	0.0101	0.0223	9.0×10^{-4}
0.34	0.9001	0.0105	0.0235	8.0×10^{-4}

Table 3.3. Testing several values for $u(t)$ when $t > 100$, fixed $u(t) = 0.75$ for $t \leq 100$. The first column stores the value of $u(t)$ for $t > 100$, fixed $u(t) = 0.75$ for $t \leq 100$; the second column stores the average fraction of population adopting self-protective behavior (i.e. with $X_i(t) = 1$), calculated over the time horizon $T = 200$; the third column stores the average fraction of population infected $Z(t)$, calculated over the time horizon $T = 200$; the fourth column stores the maximum value of $Z(t)$ over all the time horizon $T = 200$, anyway this data does not affect the decision, because the maximum value of $Z(t)$ usually happens when $t \leq 100$, in fact the values in this column are not in an increasing or decreasing order; the fifth and last column stores the average among the 50 simulations of the fraction of population dead at time $T = 200$.
Also here, each of the $n_{\text{row}} \times \text{SIM} = 6 \times 50 = 300$ simulations took around 70 seconds to be run, with a total time of almost 6 hours.

The first important thing we may notice in Figure 3.5 is that the black dotted line, which shows the real trend that the covid epidemic has had in Italy, has a period after $\tilde{t} = 100$ in which it is very low for about 60 days: this should not surprise us, in fact that period of decline in infections corresponds approximately to the months of July and August (from day 128 to day 190), in which it seemed that the contagiousness of the virus had dropped due to the warmer temperature, and also many people maybe did not take a swab even if they had symptoms, to avoid ruining themselves the holidays (this assumption is confirmed by the data stored in [4], showing that the daily number of swabs during summer was quite lower on average with respect to the past spring and the following autumn); for these reasons, the real black curve is lower in that period of our simulations, and then goes to grow towards the end of August when we returned from summer holidays and unfortunately we were preparing for a new autumn wave.
If we look at the first picture on the left in Figure 3.5 plotting the infected $Z(t)$, a too low value of u, say 0.24, 0.26, 0.28, makes the spread of the epidemics increase a lot more than 0.3, 0.32, 0.34, where actually the difference is very few. Furthermore, if we look at the central picture describing the fraction of people adopting self-protective measures, a low value of u causes the whole population to stop adopting these measures very quickly (in fact the curve drops steeply), while a value higher than 0.3 causes a large majority of the population to continue to adopt them, which is certainly fine in intense periods of the epidemic, but the hope is always that of being able to stop having to apply these safety measures as soon as possible: according to this reasoning, we can say that the value $u = 0.3$, for $t > 100$, is an excellent value for easing government restrictions.

If we focus a while on this value, it means a reduction of 40% from the original value of 0.5, which is quite similar to what happened in the 2020 summer in Italy, when many restrictions were abolished but others were maintained.
We noticed that piece-wise policies may be beneficial, whereby the intensity of NPIs is reduced after 100 days; at this point, we could wander whether is $\tilde{t} = 100$ the optimal changing point? Can we do better if we change the value of $u(t)$ in another time instant?

First, we observe that we could try to set a changing point in some $\tilde{t} > 100$, and this obviously would decrease the average number of infected and the fraction of death, since it would let the whole population to behave prudent for a longer time, but as we said before we should avoid this possibility because there would be serious damage in the national economy, other than the discontent of the population.
Therefore we should find some $\tilde{t} < 100$ as a new changing point, admitted that this can have a better impact on the epidemic spread than our baseline scenario with $\tilde{t} = 100$. In Figure 3.6 are shown the trends obtained (by calculating the average among 50 different trajectories with the same configuration) by using several changing point $\tilde{t}$.

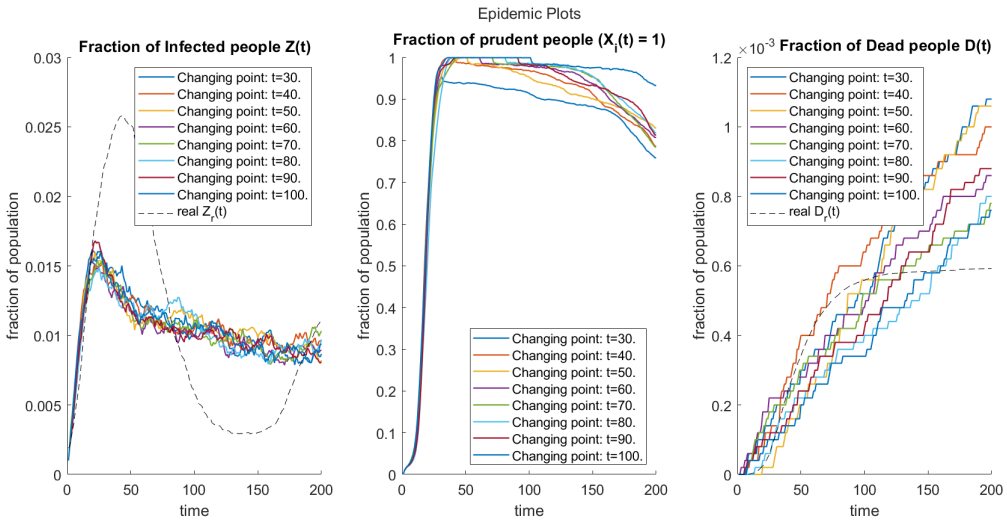

Figure 3.6. Testing several changing points $\tilde{t}$ lower than $\tilde{t} = 100$.

The most surprisingly result is that the COVID-19 spread wouldn't change a lot if we choose a $\tilde{t} < 100$, and this is clear not only from the trend of $D(t)$ in the third picture in Figure 3.6, but also in the first picture describing the epidemic spread in terms of $Z(t)$. Furthermore, if we have a look at the central picture, we can see that no matter when the restrictions change from 0.75 to 0.3, the population starts to behave at the same way.
All these arguments allow us to say that it would be useless to maintain $u(t) = 0.75$ until $t = 100$, because we wouldn't have any improvement over having an earlier changing

Changing point	Avg prudent $X_i(t) = 1$	Avg $Z(t)$	Max $Z(t)$	Avg $D(200)$
0	0.0515	0.0164	0.0399	1.2×10^{-3}
10	0.0358	0.0164	0.0392	1.3×10^{-3}
20	0.5058	0.0130	0.0306	1.0×10^{-3}
30	0.8176	0.0107	0.0237	1.1×10^{-3}
40	0.8540	0.0106	0.0227	1.0×10^{-3}
50	0.8501	0.0107	0.0239	1.1×10^{-3}
60	0.8649	0.0102	0.0224	8.6×10^{-4}
70	0.8659	0.0104	0.0230	7.8×10^{-4}
80	0.8654	0.0104	0.0227	8.0×10^{-4}
90	0.8697	0.0104	0.0233	8.8×10^{-4}
100	0.8978	0.0105	0.0228	7.6×10^{-4}

Table 3.4. Testing several values for a changing point in which we should decrease $u(t)$ from 0.75 to 0.3.
The first column stores the time instant chosen as the changing point; the second column stores the average fraction of population adopting self-protective behavior (i.e. with $X_i(t) = 1$), calculated over the time horizon $T = 200$; the third column stores the average fraction of population infected $Z(t)$, calculated over the time horizon $T = 200$; the fourth column stores the maximum value of $Z(t)$ over all the time horizon $T = 200$; the fifth and last column stores the average fraction of population dead at time $T = 200$, anyway this values are not too significant, as we explain earlier.
Each of the $n_{\text{row}} \times \text{SIM} = 11 \times 50 = 550$ simulations took around 75 seconds to be run, with a total time of more than 11 hours.

point; this choice is also very helpful for everyone because a too long lockdown would have put the country's economy even more in crisis. We should now investigate the optimal first-lockdown duration, by choosing the best changing point: for this goal we may try to decrease again the duration of the first lockdown, as we can see in Figure 3.7, where it appears very clear that choosing a changing point of $\tilde{t} = 50$, $\tilde{t} = 40$ or $\tilde{t} = 30$ it makes no difference, because the curves are substantially the same both as regards the spread of infections $Z(t)$ and as regards the deaths $D(t)$; a changing point of $\tilde{t} = 20$ instead worsens the situation a lot, increasing both the dead and the number of infected by about 50%: we can find an explanation for this fact in the central plot in Figure 3.7, where we see that with a first lockdown that is too short it fails to raise awareness among the whole population in adopting self-protective measures, and this unfortunately affects the progress of the epidemic. A one-month lockdown (i.e. $\tilde{t} = 30$), on the other hand, manages to ensure that the entire population adopts these measures, and once the value of $u(t)$ is decreased, the majority of the population will continue to behave in the same way thanks to a sort of "*civic sense*", maybe induced by the social pressure, precisely as we remember that it happened in reality.

Of course, also the possibility of very short lockdown (that is decrease the changing point to less than a month) has been explored, but the results were not satisfactory, in fact

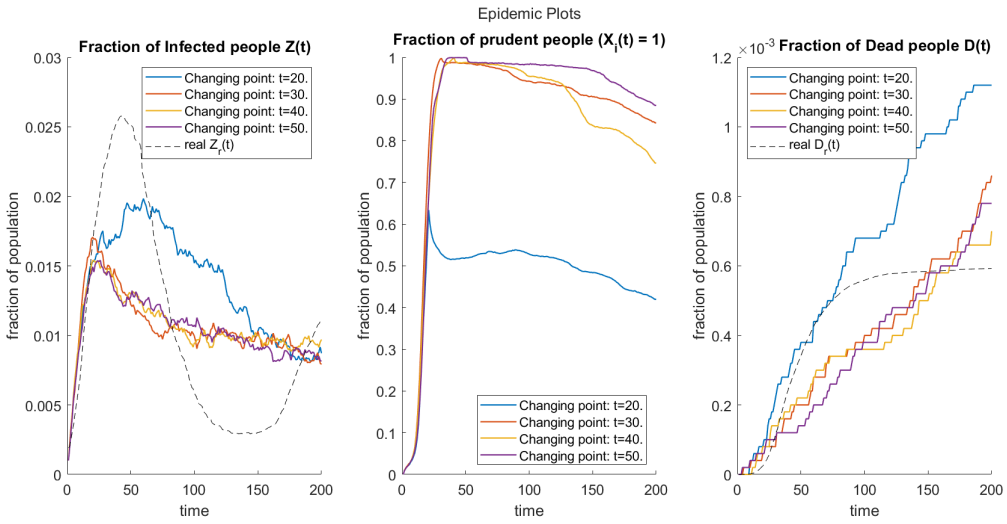

Figure 3.7. Finding an optimal changing point as the duration of the first lockdown.

the spread of infections in our simulations really struggled to contain itself, thus going to draw scenarios unfortunately unsustainable and that we all would like never to live again, as we can see in Figure 3.8.
The results obtained by testing every changing point $\tilde{t}$ are summarized in Table 3.4.

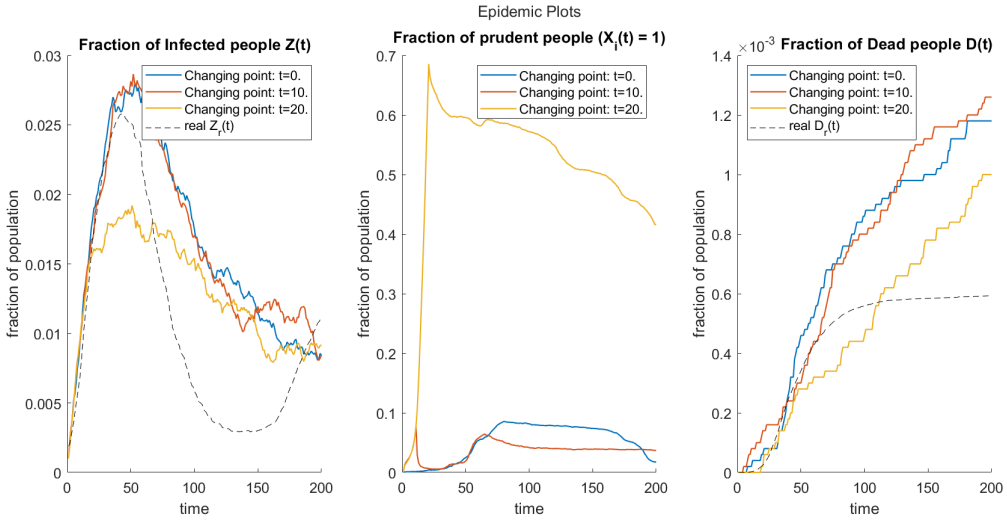

Figure 3.8. Testing very low values for the first lockdown duration.

Finally, one could manage to search a second changing point in which we could change $u(t)$ from 0.3 to another value maybe lower: we explored this possibility, but the results were not helpful and they have not been useful, and indeed have worsened the solution found previously. A possible explanation for this fact could be that, given that in Figure 3.6 it is clear that the number of infections was stabilizing on a low value, lowering $u(t)$ further compared to 0.3, which in any case is already a low enough value, would have led the whole population to no longer protect themselves, inevitably going to greatly increase the number of infections and therefore the trend of $Z(t)$.

We conclude the first part of our analysis by summarizing our main findings. According to our extensive campaign of numerical Monte Carlo simulations, the best choice for the control function $u(t)$ turned out to be:

$$u(t) = \begin{cases} 0.75, & t \leq 30 \\ 0.3, & t > 30 \end{cases} \tag{3.12}$$

This means that, to the best of our modeling insight, public administrations should impose stronger restrictive measures ($u(t) = 0.75$) at the inception of an epidemic outbreak for a shorter period of time, to then decrease them decisively ($u(t) = 0.3$), but never drop them off totally; if we set this optimal value for the control function $u(t)$ together with the optimal parameters configuration found in the previous section, we obtain the epidemic trend shown in blue in Figure 3.9.

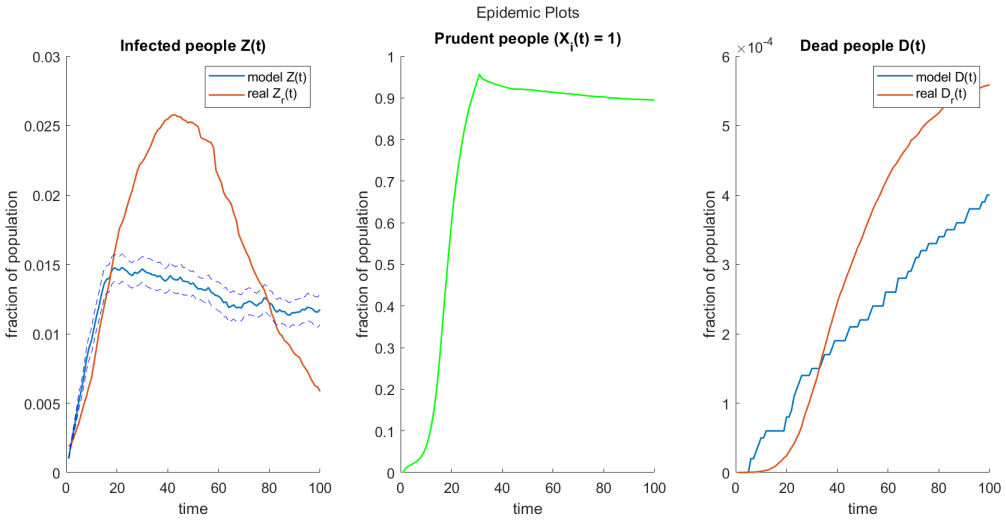

Figure 3.9. Epidemic dynamics simulated by our model by applying the optimal parameters configuration found in Table 3.1 combined with the NPIs in (3.12).

Chapter 4

Model with vaccine

In chapter 3, we presented, validated and optimized an extension of the model in [39] tailored to the first wave of COVID-19 in Italy starting from February 2020. At that time, a vaccine that prevented infection and death due to COVID-19 had not yet been developed. This was introduced at the end of 2020 and the very first doses in Italy [25] were injected on December 27, 2020 (precisely 308 days after the first COVID-19 outbreak in Italy, dated on 24 February 2020). At that moment, Italy had already gone through two major waves of COVID-19 and the arrival of the vaccine seemed to give everyone excellent hope of being able to get out of the pandemic very soon. Unfortunately today we are still struggling with this disease, even if the current prevention measures are very low (for example there is no longer the obligation to wear a protective mask).

After having developed the model of COVID-19 in absence of vaccination, the goal of this thesis is to build a new model by adding a new decision-making process, again based on game theory, regarding the will to vaccinate or not. Clearly there is a substantial difference between this decision and the previous one regarding the use of self-protective measures (3.6). In fact, if the latter decision can change at any moment of time, the one concerning vaccination will instead be definitive (obviously, once vaccinated there is no way to go back).
The state of an individual in this new model is described by the triple $(Y_i(t), X_i(t), V_i(t))$, where the three elements are, in order:

- $Y_i(t)$ is the individual i's health state at time t, that can be 0 if i is susceptible (S), 1 if i is infected (I), 2 if i is recovered (R), or 3 if i is dead (D).
- $X_i(t)$ is the first decision about whether to adopt or not self-protective behavior, which can assume the binary values 1 if i decides to adopt self-protective measures or 0 if i discard this opportunity.
- $V_i(t)$ regards the second decision about whether to get vaccinated or not: it can be 0 if individual i is not vaccinated, 1 if i is vaccinated but the vaccine has not yet taken effect and therefore the individual will have to wait some time to become protected, 2 if i is vaccinated and effectively protected.

The distinction between $V_i(t) = 1$ and $V_i(t) = 2$ arise from the fact that in reality individual i is protected by the vaccine only some days after the first dose, or alternatively if she takes the 2-nd dose, and only after that individual i is fully protected by the vaccine. To avoid complicating the model too much, this time frame was modeled by inserting the transient state "1", in which an individual is not yet protected (has not yet taken the second dose) but has already been vaccinated (so she has already chosen to vaccinate and take the first dose). Each individual is in this transient state in fact for 14 days on average [9], to better imitate reality, because he will have $\frac{1}{14}$ probability of moving towards state "2" at every instant of time t. Formally, we have:

$$\mathbb{P}(V_i(t+1) = 2 | V_i(t) = 1) = \frac{1}{14}, \quad \mathbb{P}(V_i(t+1) = 1 | V_i(t) = 2) = 0$$

Since we want to introduce a new decisional process for the vaccination, we also need a new **payoff function** that shapes people's utilities of getting vaccinated or not. The new payoff function Π_i^v for the vaccination decision process is inspired by the one used in the vaccine-free model (3.5), but obviously brings with it some differences; like we did in the previous model, we create one single payoff function rather than 2 different functions (one for the decision "get vaccinated" and one for the opposite decision "don't get vaccinated"), therefore at time t the payoff for individual i in general form is:

$$\Pi_i^v(t) = c_1 Z(t) + c_2 \frac{\#_i^1 - \#_i^0}{d_i} + u_v(t) + f_i(t) - (1 - V(t)) e^{-c_3 V(t)} - c_4 \tag{4.1}$$

The higher the payoff, the higher is the probability for an individual with $V_i(t) = 0$ (i.e. not yet vaccinated) to decide to get vaccinated, according each time instant t to the following stochastic rule very similar to (3.6):

$$\mathbb{P}(V_i(t+1) = 1 | V_i(t) = 0) = \frac{e^{\beta \Pi_i^v(t)}}{1 + e^{\beta \Pi_i^v(t)}} \tag{4.2}$$

- $c_1 Z(t)$ is the corresponding of the reaction function in (3.5), modeling the population's reaction to the spread of the disease; the coefficient c_1 can effectively be assumed equal to the previous scaling factor k, but for the moment we keep the notation more general. One could think to add another reaction function, in which the population react to the fraction of death instead of the fraction of infected; however, this reaction can be directly included in $c_1 Z(t)$, because the number of deaths is clearly positive correlated with the fraction of infected, and for this reason we decided not to complicate the model adding another parameter to be calibrated.
- $\frac{\#_i^1 - \#_i^0}{d_i}$ is the same fraction we found in the payoff for the self-protective behavior decision (3.5) in chapter 3, with the difference that in this case $\#_i^1$ and $\#_i^0$ are respectively the number of node i's neighbors vaccinated and not vaccinated, divided by the degree d_i of node i in the influence layer; this fraction is multiplied by a constant c_2 which we expected to be slightly greater than 1, like it was instead in the other model: in fact we may think that our neighbors influence on the vaccination decision could be stronger than the influence on the self-protective decision, since the so-called *no vax* people (i.e. people who don't want to get vaccinated) are usually clustered in the population.

- $u_v(t)$ represents the incentives by the government linked to the vaccine: restrictions for people not vaccinated, green pass, or other measures like these; as we did in section 3.3 for $u(t)$, we will apply control also on this variable in order to find an alternative policy, again with the reasonable aim of reducing the spread of the pandemic.

- $f_i(t)$ is exactly the same frustration function we found in (3.2), but here it shows up with a positive sign in front, because if one individual is frustrated with always wearing a mask or taking protective measures, then he may have more incentive to get vaccinated in order to alleviate those measures a little.

- The term $(1-V(t))e^{-c_3V(t)}$ represents the "fear of the vaccine": at the very beginning of the vaccine injection phase, many people were doubtful about the vaccine because this was created in a hurry and therefore there was the fear that there might be some mistake in it or it might be ineffective; however, once so many people are vaccinated this fear fades considerably; the fear can decrease more or less quickly as the population is vaccinated depending on the parameter c_3.

- c_4 is the immediate cost in getting vaccinated, for example linked to the need to lose a few days of work due to the side effects of the vaccine; this cost has the same conceptual meaning of c in (3.1), but their values do not necessarily have to coincide. In particular, the immediate cost c_4 is the only heterogeneous parameter in our model, which tries to capture the *no-vax* people, which are approximately the 10% of the whole population size [15]; therefore the parameter c_4 will be a vector assuming 2 different values, a lower value for people (the first 90%, for simplicity) and a higher value for *no-vax* people (the last 10%).

The new payoff function in (4.1) is not the only novelty brought to the decision based on game theory, because the addition of the vaccine has introduced a fundamental and much more powerful component against covid infection. For this reason many people vaccinated (as indeed happened in the real world) choose to adopt self-protection measures much more rarely, feeling sufficiently protected by the effectiveness of the vaccine. This fact suggests a modification of the payoff function also for the first decision concerning the adoption of self-protective behaviors, which becomes:

$$\Pi_i(t) = \frac{\#_i^1 - \#_i^0}{d_i} + kZ(t) - c\left(1 + \sum_{s=1}^{t} \gamma^s X_i(t-s)\right) + u(t) - gV_i(t), \tag{4.3}$$

for a certain individual i at time t. The probability for individual i to adopt self-protective measures follows the same law as in (3.4) but using the new payoff function in (4.3).
The last component in (4.3) introduces the **security parameter** g, which captures the behavior of an individual already vaccinated who feels more safe about getting infected and decides to adopt self-protective behavior more rarely; $gV_i(t)$ can assume only three values: 0 if individual i is not yet vaccinated at time t (i.e. $V_i(t) = 0$); g if individual i has just been vaccinated (i.e. $V_i(t) = 1$) but she is not yet protected; $2g$ if individual i has been vaccinated and is protected (i.e. $V_i(t) = 2$). Notice that in this latter case, the

value of the last term in (4.3) is doubled with respect to when i is vaccinated but not yet immune: this is reasonable because a person who just got vaccinated (i.e. $V_i(t) = 1$) may keep adopting self-protective measures for some days in order to wait for the effective protection (i.e. $V_i(t) = 2$).

An individual i who is not vaccinated has an infection probability equal to (3.7) at every time instant t, which is the same for an individual vaccinated but not yet protected by the vaccine (i.e. $V_i(t) = 1$). On the other hand, when the protection due to the vaccine begins to take effect, the probability of getting infected is significantly lowered through the application of a multiplication factor $1 - \alpha_1$, where $\alpha_1 \in [0,1]$ is the effectiveness of the vaccine in preventing contagion:

$$\mathbb{P}(Y_i(t+1) = I | Y_i(t) = S, V_i(t) = 2) = \underbrace{(1-\alpha_1)}_{\text{vax protection}} \times \underbrace{(1-\sigma X_i(t))}_{\text{self-behavior}} (1-(1-\lambda)^{N_i(t)}) \quad (4.4)$$

At the same time, a vaccinated individual i who is infected and in which vaccine protection is already active has a lower probability of dying than the one in vaccine-free model, in which this probability was dictated by δ at each time instant $t \in [0, T-1]$, thanks to a multiplication factor $(1 - \alpha_2)$ (with $\alpha_2 \in [0,1]$ not necessarily equal to α_1 in general), so that:

$$\mathbb{P}(Y_i(t+1) = D | Y_i(t) = I, V_i(t) = 2) = \underbrace{(1-\alpha_2)}_{\text{vax protection}} \times \delta, \qquad \forall i = 1,2,\ldots,n. \quad (4.5)$$

The remaining two epidemic probabilities, that is the recovery probability μ and the probability φ to re-become susceptible after recovering, are not affected by the vaccine efficacy; nevertheless, the per-contact infection probability λ and the death probability δ are not necessarily the same as the previous model, since during the vaccination period (i.e. early 2021) COVID-19 had developed a new variant with a different rate of contagiousness and lethality, so these epidemic parameters will certainly need to be re-calibrated to ensure that our model properly fits the real contagion curve $Z_r(t)$. At the same time, new protocols have been developed in hospitals to reduce severe consequences.

4.1 Model calibration and validation

Following the approach implemented in section 3.2, we want to calibrate and validate our model by using the real Italian data taken from [7], considering as first day the first day in which the vaccine administration has started, i.e. on the 27$^{\text{th}}$ December 2020 (308 days after 24$^{\text{th}}$ February 2020), and setting a time horizon $T = 200$ days, until the beginning of summer 2021.
In order to find a good configuration including all the tuning parameters, we performed $SIM = 50$ different simulations for each configuration, generating 50 trajectories and computing the average at every time instant t among all these trajectories, similar to what we did for the previous model in section 3.2.2, with the only difference in the number of simulations. We have been forced to decrease it for a computational timing reason: in

this part of the work we must simulate 2 decision processes and we want to study the epidemic trend over a longer (precisely, a doubled) time horizon $T = 200$. Hence, the computational time for each run of the simulation almost tripled.

4.1.1 Simulation setting

In December 2020 Italian healthcare was much more advanced and prepared in dealing with COVID-19, and for this reason both the detection rate DR and the detection delay DD as regards positive cases had changed compared to the previous ones. In particular, the detection rate has increased [22] and the detection delay has decreased thanks to the easier swabs availability [7], becoming

$$DR = \underbrace{0.07}_{\text{first wave model}} \rightarrow \underbrace{0.4}_{\text{vax model}}, \qquad DD = \underbrace{14}_{\text{first wave model}} \rightarrow \underbrace{10}_{\text{vax model}}.$$

The real $Z_r(t)$ and $D_r(t)$ are obtained exactly like in (3.8), taking though the 309^{th} line of the Italian dataset [7] as the first day of our new time period (i.e. $t = 1$); noticed this, the real values of the epidemic trend in Italy was $Z_r(1) = \frac{581760}{6\times10^7} \approx 0.0096$, meaning that we should have as initial condition around 10 individuals infected over the $n = 1000$ of our population size, but we should also divide this value for the detection rate (DR), because not all the real infected were detected in real life. In formula, we obtain:

$$Z(1) = \frac{1}{n}\frac{0.0096 \times n}{DR} = 0.024, \tag{4.6}$$

meaning that we should set 24 people out of 1000 to be infected at day-1, and we choose them randomly.
For what regards the deaths at day-1 we set $D(1) = 0$ because we would like to evaluate fatalities starting from the first day of our simulation period. Anyway, we are not so much interested in this value anymore since the number of deaths has fortunately after the first wave, therefore we will not plot the death curve $D(t)$ anymore, focusing our attention on the two population choices, and on the fraction of infected individuals $Z(t)$.
The fraction of prudent people at the beginning of the simulation is another initial condition we have to take into account, since in the previous model it was simply assumed to be equal to 0. On the 27 December 2020, there were several restrictions in Italy hoping to limit the COVID-19 spread during the winter holidays. Hence the majority of the people, if not the entire population, was adopting self-protective measures. Based on this observation, we decided to set the initial fraction of prudent people to 1.
The fraction of people vaccinated at day-1 is trivially set equal to 0, because obviously nobody was vaccinated before the 27 December 2020 [25].

4.1.2 Parameter tuning

Following the same procedure of chapter 3, we performed several simulations testing each parameter to be tuned in a reasonable range, and fixing all the other parameters to certain appropriate values, that are:

- The rationality parameter appearing both in (3.4) and (4.2) has been fixed again to $\beta = 6$, following [39].

- The scaling factor k in (4.3) has been fixed to the optimal value found in Table 3.1, i.e. $k = 12$.

- The immediate cost c for the self-protection decision, appearing both in (3.1) and in (4.3), has been fixed to the optimal value found in Table 3.1, i.e. $c = 0.5$.

- The external infection probability $\bar{\lambda}$ is reasonably the same as in chapter 3, i.e. $\bar{\lambda} = \frac{1}{n}$.

- We assumed that the vaccine protection doesn't affect the recovery probability μ, maintaining it to $\mu = 0.14$.

- The probability to become again susceptible φ has been fixed to $\varphi = 0.1$, meaning that after recovery, one individual is immune for 10 days on average.

- The effectiveness of self-protective behavior σ has been fixed to $\sigma = 0.5$ like in the first wave model in section 3.2.1.

- The vaccine efficacy α_1 for preventing infection, introduced in (4.4), has been fixed to a reasonable value like $\alpha_1 = 0.7$ [6].

- The vaccine efficacy α_2 for preventing death, introduced in (4.5), has been fixed to a value lower than α_1, i.e. $\alpha_2 = 0.5$, because in general vaccine is slightly more effective in preventing infection rather than death, even if it is also helpful in the latter case.

- The control functions $u(t)$ and $u_v(t)$ are both set to 0.5: the first one is the value used for the model validation in section 3.2 (see Figure 3.3), while $u_v(t)$ is thought to be similar to it, so we may keep them equal before testing other values when we apply control also in this second model.

All the others parameters, as it is reported in Table 4.1, have instead been tested at first with a coarse-grain approach in order to find a reasonable range of values for each parameter, and afterwards by focusing our attention on a denser grid of values for a finer search. More precisely:

- The **per-contact infection probability** λ has changed with respect to the first wave period, because COVID-19 developed some variants of the disease which had different transmissibility: we investigated several values between 10^{-4} and 10^{-2}, finding an optimal value of $\lambda = 0.008$.

- For the **death probability** δ the reasoning is the same as the one for λ because different COVID-19 variants have different mortality rates: we tested several values between 10^{-3} and 10^{-2}, finding an optimal value of $\delta = 0.002$.

- The **accumulation factor** γ was 0.25 in the optimal solution for the first wave model (see Table 3.1), but in this second model, the population had already stand almost 1 year of pandemic and restrictions, hence their patience was weaker: many values higher than the previous γ (i.e. in the interval [0.25,1]) have been tested, and the optimal value found it's $\gamma = 0.51$, meaning that the patience of people has more than halved, which is consistent with empirical observations [34].

- The **security parameter** g introduced in (4.3) is tested in several values between 10^{-4} and 10^{-2}, and the optimal value found for it is $g = 0.0018$.

- The **reaction coefficient** c_1 is very similar to the scaling factor k, therefore we tested values around the optimal $k = 12$ found in Table 3.1, but we didn't obtain any improvement by changing it, so we set it again to $c_1 = 12$.

- c_2 in (4.3) is the **influence coefficient**. In (3.5) this was implicitly set to 1. Here, we conjecture that the influence that our neighbors have on our decisions could be stronger if we are talking about getting vaccinated or not, since this is a permanent decision, while instead wearing the protective mask or adopting self-protective behaviors in general could be a fickler decision. Hence, we tested values greater than 1. In fact the optimal value for this coefficient is found at $c_2 = 1.1$.

- For the **fear coefficient** c_3 we tested only small integer numbers, which is a coherent choice because this parameter is a multiplicative exponent of an exponential function: the optimal value found is $c_3 = 2$.

- The **immediate cost** c_4 for the vaccination decision is very similar to the immediate cost c for the decision whether to adopt self-protective behavior or not, in fact the optimal value found by the tuning is again $c_4 = 0.5$, as we found in Table 3.1.

Parameter	Symbol	Tuning range	Optimal value
Per-contact infection probability	λ	$[10^{-4},10^{-2}]$	0.008
Death probability	δ	$[10^{-3},10^{-2}]$	0.002
Accumulation factor	γ	[0.25,1]	0.51
Security parameter	g	$[10^{-4},10^{-2}]$	0.0018
Reaction coefficient	c_1	[1,20]	12
Influence coefficient	c_2	[1,2]	1.1
Fear coefficient	c_3	[0,5]	2
Immediate cost	c_4	[0,10]	0.5

Table 4.1. Tuning parameter and optimal value summary for the model with vaccine.

Each simulation took about 3 minutes to be run, resulting in more than 2 hours for all the $SIM = 50$ trajectories; given all the trajectories, we calculated the average among the 50 realizations at every time instant t like we did for the first wave model in section 3.2.2, obtaining the final curve for this new model which we may call again $Z(t)$; the same

method applies for the fraction of population adopting self-protective measures ($X_i(t) = 1$) and the fraction of population already vaccinated but not necessarily immune ($V_i(t) > 0$) at time t. We also calculated the *standard deviation* at every time t in order to define a proper 95% **confidence interval** [13] for the fraction of infected of the same type like in (3.10).
For the purpose of evaluating which parameter configuration is the best one, we computed for $Z(t)$ the sum of square errors (SSE) compared to $Z_r(t)$; this quantitative approach has a qualitative counterpart, that is a reasonable analysis of the obtained plots of the infected trend $Z(t)$, the average fraction of prudent people and vaccinated people.
This qualitative approach is also an important element we should apply in order to select the best parameter configuration, since the SSE doesn't allow us to know *where* our model is going bad, if it has a high value. Hence, both approaches should be used to determine the optimal calibration of the model.
We can see in Figure 4.1 the result obtained by using the best parameter configuration for fitting the real infected trend $Z_r(t)$. The most surprisingly result is the shape of the model line in blue, which is able to faithfully reproduce the second peak of $Z_r(t)$; furthermore we can see that the model is able to replicate very well the two decreases after the two peaks.
In the plot on the right in Figure 4.1 we can see the fraction of prudent people and vaccinated people evolving in time: while one trend increases (vaccinated people, i.e. the yellow line) the other one decreases (prudent people, i.e. the green line) almost simultaneously; in general, the greater the parameter g and the more pending is the green line of the prudent people, since g models the rejection of people in adopting self-protective behaviors if they are vaccinated (see (4.3)).

4.2 Optimization and control

In the above, we have verified that our extended model is able to accurately reproduce the evolution of the pandemic even during the vaccination campaign. Here, similar to what we did for the first wave model in chapter 3, we utilize our model calibrated on real data to seek for an alternative strategy for the government, by investigating the impact of changing the value of the control function.
In particular, since we have introduced the vaccination decision and payoff $\Pi_i^v(t)$ in (4.1), we are dealing with 2 control functions: $u(u)$ modeling the government interventions to sensitize people to take self-protective measures, such as restrictions and lockdowns, and $u_v(t)$ modeling the government interventions to sensitize people to get vaccinated, such as vaccination campaign or green pass requirement at the restaurant.

4.2.1 The role of the control functions $u(t)$ and $u_v(t)$

It is quite obvious that we may increase these control functions obtaining a very huge reduction of the number of infections, but exactly as we said for $u(t)$, also for $u_v(t)$ it turns to be unrealistic to increase it too much by applying severe restrictions and furthermore

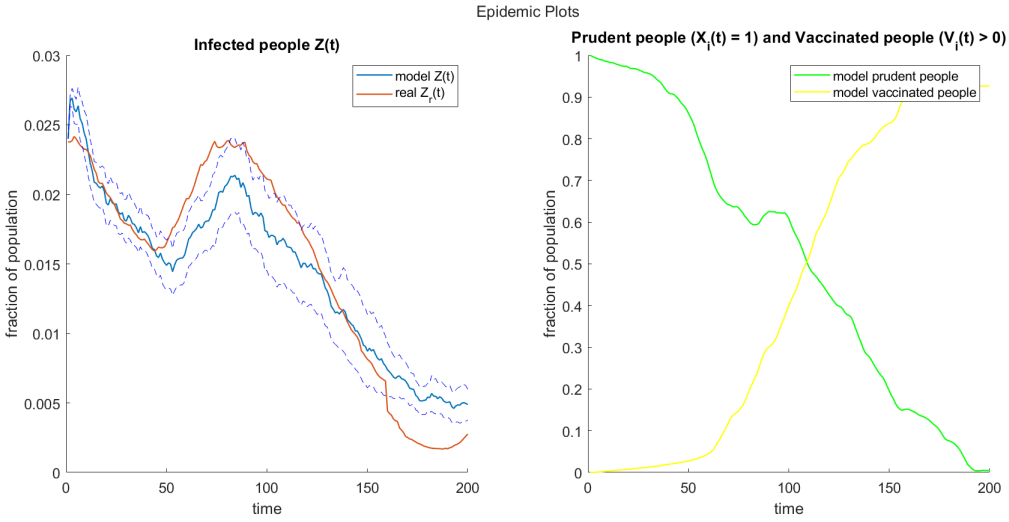

Figure 4.1. Validation of the model with vaccine, related to the the period starting from 27^{th} December 2020 in Italy. The picture on the left shows the trend of our model's $Z(t)$ in blue, with its 95% confidence interval (the dotted light blue lines), compared to real $Z_r(t)$ taken from [7]; the green curve on the right describes the fraction of people adopting self-protective behavior at time t, while the yellow line represents the fraction of vaccinated people evolving in time.

the government in that months did a lot of vaccination campaign therefore a large increase in $u_v(t)$ would have a difficult interpretation.
Nevertheless, we can think that the government economic and time resources are limited, meaning that it could be almost impossible to keep high values for both control functions simultaneously, because it would require a too big effort. For this reason, we could set a *budget constraint* which allow us to reasonably maintain the government efforts limited. Consistently with our calibrated model, where $u(t) = u_v(t) = 0.5$, we set:

$$u(t) + u_v(t) \leq 1, \quad \forall t = 1,2,\dots,T. \tag{4.7}$$

As we already have seen in chapter 3, the control function $u(t)$ has the effect of decreasing the epidemic spread when it increases, and the same stands for the control function $u_v(t)$ concerning the vaccination decision. For this reason we expect that in our model, which doesn't consider deeply the economic effects of the pandemic, the optimal choice will be on the boundary of the feasible set, that is, $u(t) + u_v(t) = 1$.
Furthermore we have to say that the choice to set an upper bound equal to 1 is completely arbitrary: it seems clever because in the best parameter configuration used to obtain the curve fitting in Figure 4.1 the control functions were both constant $u(t) = u_v(t) = 0.5$ for every time instant t in the whole period of study $T = 200$, hence their sum was equal to 1. We should furthermore remark that $u(t)$ and $u_v(t)$ are not bounded a priori, but their optimal values allow us to think that 1 could be a reasonable upper bound for the sum of the two.

Our first goal is to avoid the fraction of infected people $Z(t)$ to have a peak after around 80 days, trying then to stabilize the epidemic spread (or even better to decrease it) rather than having another outbreak. We tried to change the values of the two control functions, paying attention to satisfy the budget constraint in (4.7): the results of this study are shown in Figure 4.2 and in Table 4.2.
The first clear result is that we can't increase $u_v(t)$ (and then decrease $u(t)$) too early because the immunization requires some weeks, and lowering the restrictions modeled by $u(t)$ would have the result of making an explosion of the disease, returning to exert unsustainable pressure and stress for hospital staff (something that we hope to avoid), before decreasing deeply luckily thanks to the immunization brought by vaccinations.
In the first plot in Figure 4.2, the best curve seems to be the purple one, where $u(t) = 0.6$ and $u_v(t) = 0.4$: the meaning of this fact is that we should keep an high level of attention or self-protective behavior while the population is gradually getting vaccinated, otherwise we would face another outbreak of the epidemic.
Anyway, we should not increase $u(t)$ up to 0.7 because in the first part (let's say up to around $t = 120$) of the left plot in Figure 4.2 the green curve is very similar to the purple one, while in the final part the purple line goes very low because the immunization due to the vaccine is having positive effects; in fact, if we focus on the right plot showing the fraction of vaccinated individuals, we can see that the green line stays quite constant to an almost-zero value, while the purple line related to $u(t) = 0.6$ and $u_v(t) = 0.4$ is increasing fast after an initial low phase. To sum up, it seems that there is an optimal setting of the two control functions to avoid resurgent outbreaks without delaying too much the vaccination campaign.

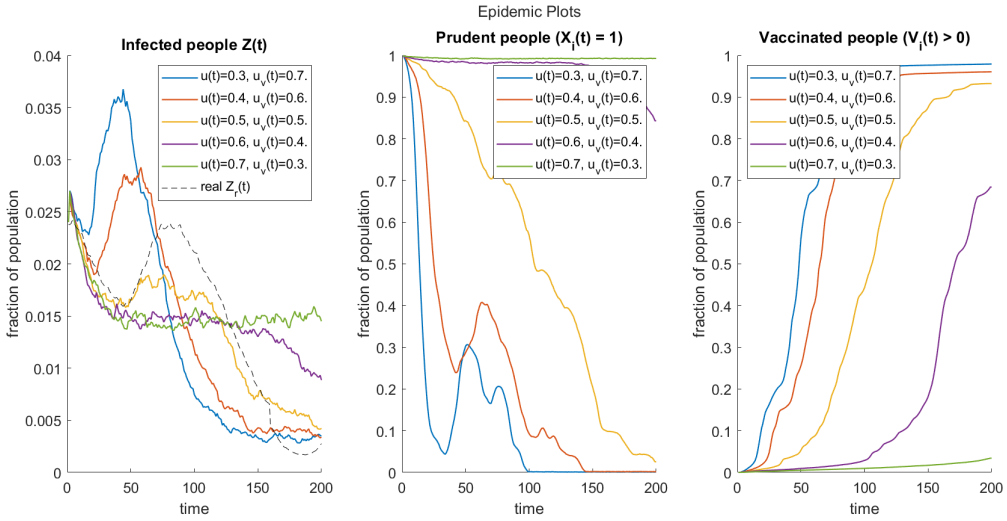

Figure 4.2. Testing different combinations for $u(t)$ and $u_v(t)$, related to the the period starting from 27^{th} December 2020 in Italy.

$u(t)$	$u_v(t)$	Avg $X_i(t) = 1$	Avg $V_i(t) > 0$	Avg $Z(t)$	Max $Z(t)$	Avg $D(200)$
0.30	0.70	0.1267	0.7394	0.0141	0.0535	6.2×10^{-4}
0.40	0.60	0.2326	0.6541	0.0139	0.0472	9.8×10^{-4}
0.50	0.50	0.5304	0.4651	0.0138	0.0364	5.8×10^{-4}
0.60	0.40	0.9673	0.1544	0.0146	0.0310	7.6×10^{-4}
0.70	0.30	0.9929	0.0118	0.0154	0.0303	9.2×10^{-4}

Table 4.2. Testing several combinations for $u(t)$ and $u_v(t)$.
The first and second columns store the value of $u(t)$ and $u_v(t)$ respectively; the third and fourth columns store respectively the average fraction of population adopting self-protective behavior (i.e. with $X_i(t) = 1$) and the average fraction of population vaccinated (i.e. $V_i(t) > 0$), calculated over the time horizon $T = 200$; the fifth column stores the average fraction of population infected $Z(t)$, calculated over the time horizon $T = 200$; the sixth column stores the maximum value of $Z(t)$ over all the time horizon $T = 200$, which describes in some way the strain borne by hospitals and intensive care units; the seventh and last column stores the average among the $SIM = 50$ simulations of the population dead at time $T = 200$.
Each of the $n_{\text{row}} \times \text{SIM} = 5 \times 50 = 250$ simulations took around 200 seconds to be run, with a huge total time of more than 17 hours.

4.2.2 Finding an optimal changing point

The results obtained in Table 4.2 allow us to consider as a very good policy the adoption of $u(t) = 0.6$ and $u_v(t) = 0.4$ for the NPIs, at least for a first time period in order to avoid resurgent waves. However, keeping such a high level of NPIs may be sub-optimal, the risk of a resurgent wave has passed. For this reason, we could be interested in modifying the values of the control functions by increasing the vaccination campaign effort from a certain time instant $\tilde{t}$ on, because we noticed in Figure 4.2 how powerful is the vaccine in preventing contagion.
Similar to what we did for the first wave model in section 3.3.2, we fix $\tilde{t} = 100$ as a first **changing point**, making the control functions to became piecewise constant, i.e.:

$$u(t) = \begin{cases} 0.6, & t \leq 100 \\ u, & t > 100 \end{cases}, \quad u_v(t) = \begin{cases} 0.4, & t \leq 100 \\ 1 - u, & t > 100 \end{cases} \tag{4.8}$$

In Figure 4.3 we show the simulations obtained by setting $u = \{0.6,0.5,0.4,0.3,0.2\}$ and generating trajectories with the control functions in (4.8); the numerical results are reported in Table 4.3, but just by looking at the plot in Figure 4.3 it is clear that setting $u = \{0.2,0.3,0.4\}$ is a bad choice, because we would have a huge peak of the positive cases of COVID-19, almost higher than the fraction of infected at the beginning of the studied time period. On the other hand, keeping $u(t) = 0.6$ and $u_v(t) = 0.4$ even for $t > \tilde{t} = 100$ may become the best choice for what concerns the fraction of infected (the left plot in Figure 4.3), but this configuration would let the vaccination process proceeding very slowly,

as we can see in the third picture of Figure 4.3.
For these reasons, we think it should be better to re-calibrate the control to $u(t) = 0.5$ and $u_v(t) = 0.5$ after the changing point, i.e. for $t > \tilde{t} = 100$: this choice can be an excellent compromise between the need to vaccinate a large part of the population quickly (therefore not beyond our time horizon T) and the goal of not to make the $Z(t)$ curve which describes the trend of infections day by day rise too high.

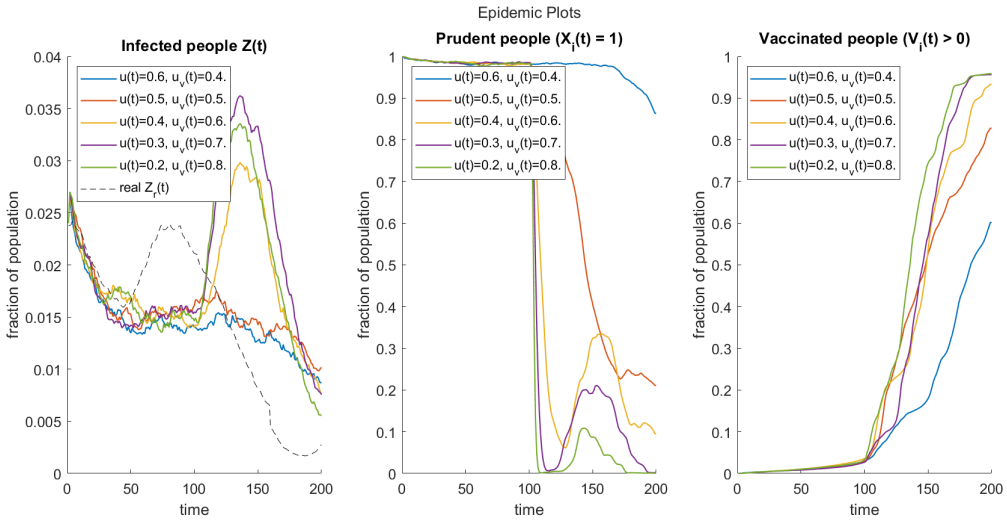

Figure 4.3. Testing different combinations for $u(t)$ and $u_v(t)$ for $t > 100$, having fixed $u(t) = 0.6$ and $u_v(t) = 0.4$ for $t \leq 100$.

Following the same reasoning of section 3.3.2, we could be interested in finding a good changing point $\tilde{t}$ in which decrease the control function $u(t)$ from 0.6 to 0.5, while increasing simultaneously the vaccination control function $u_v(t)$ from 0.4 to 0.5, such that the vaccination campaign grows faster.

Remember that our first goal was to avoid the peak in the $Z(t)$ trend around $t = 100$, which is exactly the changing point we set as default in Figure 4.3, since a too lower value would not have avoided to have a peak that is difficult to manage for healthcare workers. We simulated the epidemic dynamic for several choices of changing point, even testing some $\tilde{t} < 100$. As we can see in Figure 4.4, to have a changing point $\tilde{t} = 100$ (the red line in the plot on the left) is better than have $\tilde{t} = 80$ (the blue line), because the latter one dominates almost always the former one from above in the trend of the fraction of infected $Z(t)$.
For what regards values larger than $\tilde{t} = 100$, we can see in the left plot of Figure 4.4 that both the yellow and purple line (i.e. respectively $\tilde{t} = 120$ and $\tilde{t} = 140$) have a terminal value $Z(T)$ much higher with respect to the red line describing the trend of infected with

$u(t)$	$u_v(t)$	Avg $X_i(t) = 1$	Avg $V_i(t) > 0$	Avg $Z(t)$	Max $Z(t)$	$D(200)$
0.60	0.40	0.9739	0.1343	0.0144	0.0300	7×10^{-4}
0.50	0.50	0.7478	0.2454	0.0156	0.0355	6×10^{-4}
0.40	0.60	0.6160	0.2654	0.0182	0.0453	7.8×10^{-4}
0.30	0.70	0.5610	0.2741	0.0200	0.0530	9.8×10^{-4}
0.20	0.80	0.5214	0.3183	0.0190	0.0502	7×10^{-4}

Table 4.3. Testing several combinations for $u(t)$ and $u_v(t)$ when $t > 100$, having fixed $u(t) = 0.6$ and $u_v(t) = 0.4$ for $t \leq 100$.
The first and second columns store the value of $u(t)$ and $u_v(t)$ respectively (both are meant for $t > 100$); the third and fourth columns store respectively the average fraction of population adopting self-protective behavior (i.e. with $X_i(t) = 1$) and the average fraction of population vaccinated (i.e. $V_i(t) > 0$), calculated over the time horizon $T = 200$; the fifth column stores the average fraction of population infected $Z(t)$, calculated over the time horizon $T = 200$; the sixth column stores the maximum value of $Z(t)$ over all the time horizon $T = 200$, which describes in some way the strain borne by hospitals and intensive care units; the seventh and last column stores the average among the $SIM = 50$ simulations of the population dead at time $T = 200$.
Each of the $n_{\text{row}} \times \text{SIM} = 5 \times 50 = 250$ simulations took around 200 seconds to be run, with a huge total time of almost 14 hours.

$\tilde{t} = 100$, and nevertheless their fraction of vaccinated at the end of the time period are (quite obviously) lower than the red one.
Furthermore, we should not to forget that in each time instant t before the changing point $\tilde{t}$ the NPIs about the self-protective behavior is $u(t) = 0.6$, that is 20% higher than the "default" value for the measures adopted in real life at that time; given that, we would like to have a lockdown as short as possible, therefore we may keep $\tilde{t} = 100$ as the optimal changing point because it is a very good compromise.

Finally, we may say that, according to our analysis, we identified as an optimal choice for the control functions $u(t)$ and $u_v(t)$ the following policy:

$$u(t) = \begin{cases} 0.6, & t \leq 100 \\ 0.5, & t > 100 \end{cases}, \quad u_v(t) = \begin{cases} 0.4, & t \leq 100 \\ 0.5, & t > 100 \end{cases} \tag{4.9}$$

which combined with the best parameters configuration found in Table 4.1 gives as result the epidemic trend $Z(t)$ shown in Figure 4.5.
In the very last days before $T = 200$ the epidemic spread $Z(t)$ is decreasing steadily, even after a peak much less dangerous than the real one (look at the red line in Figure 4.5), surely thanks to the vaccination campaign which led almost the whole population to get vaccinated, with the exception of a 10% of people representing who doesn't want to get vaccinated (the so-called *no vax*).

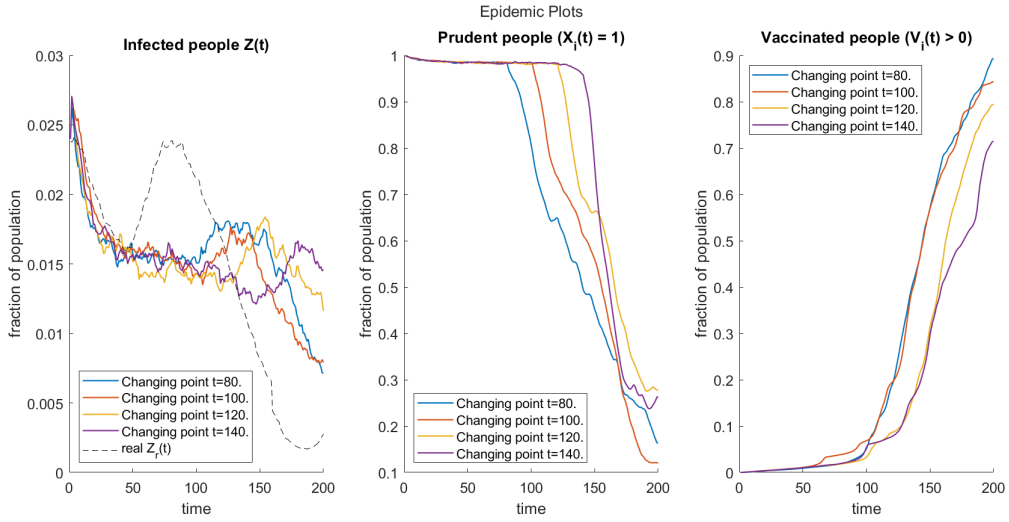

Figure 4.4. Searching for an optimal changing point $\tilde{t}$, having fixed $u(t) = 0.6$ and $u_v(t) = 0.4$ for $t \leq \tilde{t}$, and $u(t) = 0.5$ and $u_v(t) = 0.5$ for $t > \tilde{t}$.

$\tilde{t}$	Avg $X_i(t) = 1$	Avg $V_i(t) > 0$	Avg $Z(t)$	Max $Z(t)$	$D(200)$
80	0.7093	0.2648	0.0157	0.0373	6.6×10^{-4}
100	0.7455	0.2640	0.0152	0.0348	7.8×10^{-4}
120	0.8144	0.1907	0.0156	0.0343	9.4×10^{-4}
140	0.8253	0.1650	0.0156	0.0351	1.1×10^{-3}

Table 4.4. Searching for an optimal changing point $\tilde{t}$ for $u(t)$ and $u_v(t)$, set $u(t) = 0.6$ and $u_v(t) = 0.4$ for $t \leq \tilde{t}$ and $u(t) = 0.5$ and $u_v(t) = 0.5$ for $t > \tilde{t}$.
The first column stores the time instant of the changing point tested $\tilde{t}$; the second and third columns store respectively the average fraction of population adopting self-protective behavior (i.e. with $X_i(t) = 1$) and the average fraction of population vaccinated (i.e. $V_i(t) > 0$), calculated over the time horizon $T = 200$; the fourth column stores the average fraction of population infected $Z(t)$, calculated over the time horizon $T = 200$; the fifth column stores the maximum value of $Z(t)$ over all the time horizon $T = 200$, which describes in some way the hospitals and intensive care units pressure; the last column stores the average among the $SIM = 50$ simulations of the population dead at time $T = 200$.
Each of the $n_{\text{row}} \times \text{SIM} = 4 \times 50 = 200$ simulations took around 220 seconds to be run, with a total time of more than 12 hours.

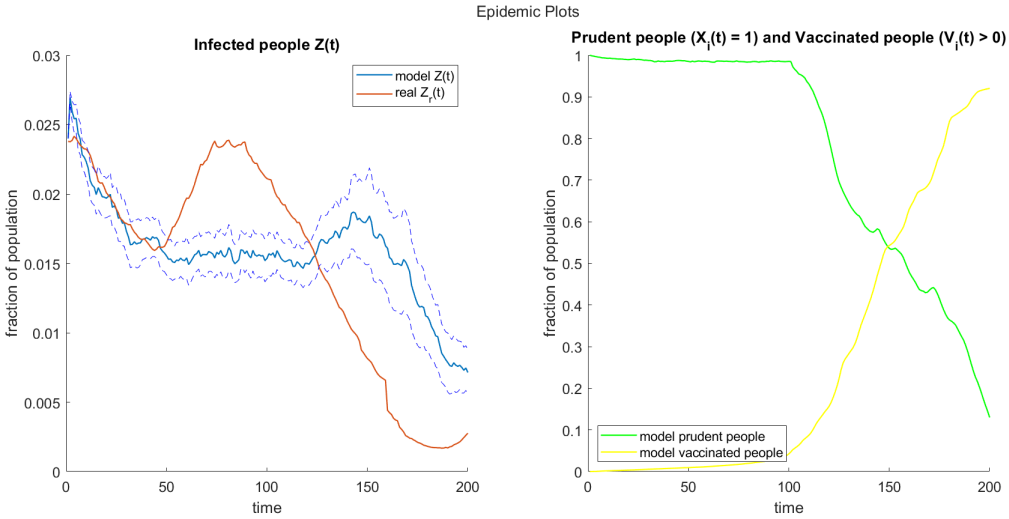

Figure 4.5. Epidemic trend simulated by our model with vaccine (blue line) compared to the real trend (red line) applying the optimal parameters configuration (see Table 4.1) and control functions like in (4.9).

Chapter 5

Conclusion and further research

The need to tackle a global pandemic is certainly a tough challenge for every country, especially nowadays where every part of the world is highly connected and easily accessible thanks to the hundreds of planes that travel every day. It is perhaps even more problematic for Italy, due to its population's lifestyle, which is always very friendly and which bases a good part of the day on interpersonal relationships. Developing mathematical models for epidemics that capture the dynamics that the population follows is therefore fundamental to be able to forecast the evolution of an epidemic and mitigate its spread, in particular if it concerns behaviors that directly concern the adoption of protective measures or vaccination.

In this thesis, we combined the study of people's behavior, using an approach based on game theory and the introduction of special payoff functions, with the analysis of the spread of the COVID-19 epidemic in Italy, both in the period before the administration of vaccines and after it, thus having two different parameter configurations depending on the time period studied. The behavior of people is greatly influenced not only by the fear of epidemics and what happens around them and their loved ones, but also by the regulations imposed by the government of the country in which they live: in our case study, we analyzed the impact of the restrictions applied by the Italian government during the emergency of the COVID-19 epidemic in Italy on the behavior of people and consequently on the spread of the epidemic.
The analysis we have carried out on the control of the function $u(t)$ in chapter 3, and also $u_v(t)$ in the second model with the vaccine in chapter 4, is certainly simplified and leaves a lot of space for possible future investigations, as well as being limited by the focus purely sociological and epidemic and almost never considering the economic aspect, which is instead crucial in the decisions that the government takes for the good of the nation, but already thanks to the results obtained in this thesis we can suggest that, in the future, an alternative policy with respect to restrictions that have been applied during the COVID-19 period could help in limiting not only the number of infections

(and consequently deaths), but also the very heavy situation of pressure and stress that all hospital departments, nurses, doctors and all health staff had to endure in those very difficult months.

Specifically, our framework shows that when the vaccine against the spreading disease is not yet available, applying stronger restrictions (i.e. our control function $u(t)$, see (3.12)) for an initial short period of time, and subsequently partially uplifting these measures to a lower value (but without removing them completely), may be more advantageous than keeping the restrictive measures constant at a medium value over the entire period of time, as was done during the first wave of COVID-19 in Italy with $u(t) = 0.5$.
On the other hand, when we are at the beginning of the vaccination campaign, we should not forget the use of self-protective measures, incurring in the risk for the population to stop adopting those safety measures because they feel protected enough by the vaccine. As we all know, it takes long time to vaccinate most of the population, due to limited logistic and budget capacity (4.7). It seems that it would be better for reducing the epidemic spread to keep a quite high value for the NPIs related to the self-protective measures (i.e. our first control function $u(t)$) at the beginning of the vaccination campaign, and subsequently reduce the restrictions while pushing on the vaccination (i.e. by increasing our second control function $u_v(t)$) with the final goal of immunizing mosto of the population. The results obtained by our analysis are shown in (4.9), where both the control functions are reasonable *piecewise constant* functions, with an optimal changing point $\tilde{t}$ identified in section 4.2.2.

5.1 Further research

The work of this master's thesis has been carried out with care and diligence, but it would be foolish to think that it can be exhaustive for the purpose of a complete learning on the combined dynamics between population decision-making and the spread of a disease. In the future, we could certainly expand this work and deepen it by developing other interesting research. For example, the following are questions that are worth of being investigated:

- We could think of increasing the population size n to evaluate the scalability of the model: due to limitations in the computational power, we had to limit ourselves to consider $n = 1000$, but obviously increasing the number of individuals in our sample would have helped to bring the model closer to reality, given that, as we know, about 60 million people live in Italy.

- Not only increasing the population size could be interesting, but also increasing the time horizon T. In fact, we focused our attention on a short-term analysis, setting $T = 100$ for the first wave model and $T = 200$ for the model with the vaccine instead. Our goal was to improve the epidemic situation in the short-term when considering the first COVID-19 wave in Italy, because who lived that months knows how much

Figure 5.1. Crowd in the center of Rome during COVID-19 pandemic [11] on 30 January 2021, a bit more than one month after the "vax day" [25].

they were hard but we can imagine that they could have been better if the number of infected people had not been so high, as we can see in the Figure 3.9; increasing the time horizon T can be very useful to be able to model the second wave of COVID-19, also trying to apply control to find solutions that stabilize the spread of the disease at an acceptable and therefore preferably low level.

- Another limitation of our model is the assumption of a homogeneous population, except for the immediate cost c in (4.1). Actually this is not the reality, because each individual is different from the others. Just think, for example, of the very first weeks of vaccination, in which the people who were vaccinated were the oldest and paradoxically those who had less contact during the day because they did not go to work like the rest of the adult population. Another equally valid example is the fact that different people have different jobs, therefore the impact of the restrictive measures has a different weight on each individual, thus leading us to think that the accumulation factor γ, modeling people's long-term frustration, may be heterogeneous.
  In a future work, it can therefore be very interesting to try to make many parameters of our model heterogeneous, such as for example the immediate cost c, the accumulation factor γ or also some epidemic parameter (λ, μ, or δ) in order to differentiate between the population the so-called *high-risk* people from the *low-risk* ones, that is, with the lowest immune defenses or with other previous diseases.

- Another interesting research can be conducted towards artificial intelligence which nowadays is increasingly pervasive in many areas, thanks to the rapid development of related technologies: one could therefore think of automating the control over the functions $u(t)$ and $u_v(t)$, by implementing self-learning algorithms in the model to make the model itself to select the optimal policy on its own, instead of having to go and study different configurations as we have done in this thesis.

In conclusion, the research work of this thesis helped investigating the interactions between the decision-making processes that each individual carries out, perhaps even unconsciously, and the spread of the COVID-19 epidemic.
We have used our modeling approach to highlight how government intervention with specific laws or restrictive measures has an impact on people's behavior, and consequently inevitably also on the spread of the disease. Our analysis therefore could be useful to create preparedness in the future in the unfortunate case in which new waves of COVID-19 or a new pandemic ever breaks out.

Bibliography

[1] Ashish Sadanandan, MathWorks. Csvimport. available at https://www.mathworks.com/matlabcentral/fileexchange/23573-csvimport., 2022.

[2] Martí Català, David Pino, Miquel Marchena, Pablo Palacios, Tomás Urdiales, Pere-Joan Cardona, Sergio Alonso, David López-Codina, Clara Prats, and Enrique Alvarez-Lacalle. Robust estimation of diagnostic rate and real incidence of covid-19 for european policymakers. *PLOS ONE*, 16(1):1–26, 01 2021.

[3] G. Como and F. Fagnani. *Lecture notes on network dynamics*. 2020.

[4] Statistiche Coronavirus. Statistiche coronavirus italia. available at https://statistichecoronavirus.it/coronavirus-italia/.

[5] Cristina Da Rold, Valigia Blu. La disinformazione sul coronavirus e cosa sappiamo finora. available at https://www.valigiablu.it/coronavirus-disinformazione/.

[6] Brechje de Gier, Stijn Andeweg, Jantien A Backer, Susan JM Hahné, Susan van den Hof, Hester E de Melker, and Mirjam J Knol and. Vaccine effectiveness against SARS-CoV-2 transmission to household contacts during dominance of delta variant (b.1.617.2), the netherlands, august to september 2021. *Eurosurveillance*, 26(44), November 2021.

[7] Presidenza del Consiglio dei Ministri Dipartimento della Protezione Civile. Covid-19 italia. available at https://github.com/pcm-dpc/COVID-19.

[8] Ministero della Salute. Covid-19 - situazione nel mondo. available at https://salute.gov.it/portale/nuovocoronavirus/dettaglioContenutiNuovoCoronavirus.jsp?area=nuovoCoronavirus&id=5338&lingua=italiano&menu=vuoto.

[9] Istituto Superiore di Sanità. available at https://www.iss.it/covid19-faq/-/asset_publisher/yJS4xO2fauqM/content/quanto-devo-aspettare-dopo-la-vaccinazione-per-essere-protetto-.

[10] Istituto Superiore di Sanità. International outbreak of novel sars-cov-2 coronavirus infection. available at https://www.epicentro.iss.it/en/coronavirus/sars-cov-2-international-outbreak.

[11] Fabrizio Caccia, Corriere della Sera. Covid, shopping e folla nei centri storici: le città non aspettano la zona gialla. available at https://www.corriere.it/cronache/21_gennaio_30/shopping-folla-centri-storici-citta-non-aspettano-zona-gialla-23a5c298-6336-11eb-abca-1766700006e6.shtml.

[12] Sebastian Funk, Marcel Salathé, and Vincent A. A. Jansen. Modelling the influence of human behaviour on the spread of infectious diseases: a review. *Journal of The Royal Society Interface*, 7(50):1247–1256, May 2010.

[13] M. Gasparini. *Modelli probabilistici e statistici*. CLUT, 2014.

[14] Marta Casadei, Il Sole 24Ore. Cina, stime riviste al ribasso: i lockdown rallentano i consumi. available at https://www.ilsole24ore.com/art/cina-stime-riviste-ribasso-lockdown-rallentano-consumi-AEkUwZFC.

[15] Edouard Mathieu, Hannah Ritchie, Esteban Ortiz-Ospina, Max Roser, Joe Hasell, Cameron Appel, Charlie Giattino, and Lucas Rodés-Guirao. A global database of COVID-19 vaccinations. *Nature Human Behaviour*, 5(7):947–953, May 2021.

[16] MathWorks. Build watts-strogatz small world graph model. available at https://it.mathworks.com/help/matlab/math/build-watts-strogatz-small-world-graph-model.html.

[17] Joël Mossong, Niel Hens, Mark Jit, Philippe Beutels, Kari Auranen, Rafael Mikolajczyk, Marco Massari, Stefania Salmaso, Gianpaolo Scalia Tomba, Jacco Wallinga, Janneke Heijne, Malgorzata Sadkowska-Todys, Magdalena Rosinska, and W. John Edmunds. Social contacts and mixing patterns relevant to the spread of infectious diseases. *PLoS Medicine*, 5(3):e74, March 2008.

[18] Francesco Parino, Lorenzo Zino, Maurizio Porfiri, and Alessandro Rizzo. Modelling and predicting the effect of social distancing and travel restrictions on COVID-19 spreading. *Journal of The Royal Society Interface*, 18(175), February 2021.

[19] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. Epidemic processes in complex networks. *Rev. Mod. Phys.*, 87:925–979, Aug 2015.

[20] Janice L. Pearce. On complexity. available at https://runestone.academy/ns/books/published/complex/index.html.

[21] Nicola Perra, Duygu Balcan, Bruno Gonçalves, and Alessandro Vespignani. Towards a characterization of behavior-disease models. *PLoS ONE*, 6(8):e23084, August 2011.

[22] Giulia Pullano, Laura Di Domenico, Chiara E. Sabbatini, Eugenio Valdano, Clément Turbelin, Marion Debin, Caroline Guerrisi, Charly Kengne-Kuetche, Cécile Souty, Thomas Hanslik, Thierry Blanchon, Pierre-Yves Boëlle, Julie Figoni, Sophie Vaux, Christine Campèse, Sibylle Bernard-Stoecklin, and Vittoria Colizza. Underdetection of cases of COVID-19 in france threatens epidemic control. *Nature*, 590(7844):134–139, December 2020.

[23] Alessandro Rizzo, Mattia Frasca, and Maurizio Porfiri. Effect of individual behavior on epidemic spreading in activity-driven networks. *Physical Review E*, 90(4), October 2014.

[24] Faryad Darabi Sahneh, Fahmida N. Chowdhury, and Caterina M. Scoglio. On the existence of a threshold for preventive behavioral responses to suppress epidemic spreading. *Scientific Reports*, 2(1), September 2012.

[25] SkyTG24. Covid, è il v-day: primo giorno di somministrazione del vaccino. available at https://tg24.sky.it/cronaca/2020/12/27/covid-vaccino-primo-giorno.

[26] Wikipedia. Best response. available at https://en.wikipedia.org/wiki/Best_response.

[27] Wikipedia. Coordination game. available at https://en.wikipedia.org/wiki/Coordination_game.

[28] Wikipedia. Giuseppe conte. available at https://it.wikipedia.org/wiki/Giuseppe_Conte.

[29] Wikipedia. Graph theory. available at https://en.wikipedia.org/wiki/Graph_theory.

[30] Wikipedia. Monte carlo method. available at https://en.wikipedia.org/wiki/Monte_Carlo_method.

[31] Wikipedia. Pandemia. available at https://it.wikipedia.org/wiki/Pandemia.

[32] Wikipedia. Pandemia di covid-19. available at https://it.wikipedia.org/wiki/Pandemia_di_COVID-19#Italia.

[33] Wikipedia. Prisoner's dilemma. available at https://en.wikipedia.org/wiki/Prisoners_dilemma.

[34] Wikipedia. Proteste contro le politiche covid-19 in italia. available at https://it.wikipedia.org/wiki/Proteste_contro_le_politiche_COVID-19_in_Italia.

[35] Wikipedia. Quantile. available at https://it.wikipedia.org/wiki/Quantile.

[36] Wikipedia. Random graph. available at https://en.wikipedia.org/wiki/Random_graph.

[37] Wikipedia. Temporal network. available at https://en.wikipedia.org/wiki/Temporal_network.

[38] Wikipedia. Watts–strogatz model. available at https://en.wikipedia.org/wiki/Watts-Strogatz_model.

[39] Mengbin Ye, Lorenzo Zino, Alessandro Rizzo, and Ming Cao. Game-theoretic modeling of collective decision making during epidemics. *Physical Review E*, 104(2), August 2021.

[40] Lorenzo Zino and Ming Cao. Analysis, prediction, and control of epidemics: A survey from scalar to dynamic network models. *IEEE Circuits and Systems Magazine*, 21(4):4–23, 2021.

Appendix A

MATLAB code

A.1 Watts-Strogatz function

Function to create a Watts-Strogatz graph, introduced in section 2.2.2, taken from [16]. We used a Watts-Strogatz graph to create the influence layer for both models with and without the introduction of the vaccination decision.

```
1  % Copyright 2015 The MathWorks, Inc.
2
3  function h = WattsStrogatz(N,K,beta)
4  % H = WattsStrogatz(N,K,beta) returns a Watts-Strogatz model graph with N
5  % nodes, N*K edges, mean node degree 2*K, and rewiring probability beta.
6  % beta = 0 is a ring lattice, and beta = 1 is a random graph.
7
8  % Connect each node to its K next and previous neighbors. This constructs
9  % indices for a ring lattice.
10 s = repelem((1:N)',1,K);
11 t = s + repmat(1:K,N,1);
12 t = mod(t-1,N)+1;
13
14 % Rewire the target node of each edge with probability beta
15 for source=1:N
16     switchEdge = rand(K, 1) < beta;
17
18     newTargets = rand(N, 1);
19     newTargets(source) = 0;
20     newTargets(s(t==source)) = 0;
21     newTargets(t(source, ~switchEdge)) = 0;
22
23     [~, ind] = sort(newTargets, 'descend');
24     t(source, switchEdge) = ind(1:nnz(switchEdge));
25 end
26
27 h = graph(s,t);
28 end
```

A.2 Payoff function for the model without vaccine

The following is our implementation of the payoff function (3.5) for the self-protective behavior decision-process (3.6), introduced in section 3.1.1 for the first model without vaccine, related to the first wave of COVID-19 spreading in Italy.

```
%Payoff function for the self-protective decision process

function Pi = Payoff(i,t)
%we compute tha payoff for individual i at time instant t

global N
global K
global state
global influence_layer
global u
global c
global accumulation
global k
global degree

one = 0;
three = 0;
count = 0; %counter for breaking the for cycle when count = degree(i)
for j = 1:K*N
    if influence_layer(j,1) == i
        one = one + (1-state(1,influence_layer(j,2),1));
        three = three + state(1,influence_layer(j,2),1);
        count = count+1;
    elseif influence_layer(j,2) == i
        one = one + (1-state(1,influence_layer(j,1),1));
        three = three + state(1,influence_layer(j,1),1);
        count = count+1;
    end
    if count == degree(i)
        break
    end
end

one = one/degree(i); %first term of Pi0
two = u(t); %second term of Pi0
three = three/degree(i); %first term of Pi1
four = k*sum(state(2,:,1)==1)/N; %r(t), second term of Pi1
five = c + accumulation(i); %third term of Pi1

Pi = (three+four-five) - (one-two); % Pi = Pi1 - Pi0

end
```

A.3 Payoff function for the model with vaccine

The following is our implementation of a single payoff function which includes both the payoff for the self-protective behavior decision-process (4.3) and the one for the vaccination decision (4.1), designed for the model with vaccine in chapter 4. Which decision to take is indicated by inserting a binary flag in the values that the function takes as input, together with the individual i and the time instant t.

```
%Payoff function for the vaccine model

function Pi = Payoff_vax(i,t,flag)
%compute the payoff for individual i at time instant t
%the flag is a binary value which determines whether we need to compute
%the payoff for the self-protective behavior decision (flag=0) or
%the payoff for the vaccination decision (flag=1)

global N
global K
global state
global influence_layer
global u
global u_v
global c
global accumulation
global k
global degree
global g
global c1
global c2
global c3
global c4

if flag == 0 %payoff for the self-protective behaviour

one = 0;
three = 0;
count = 0; %counter for breaking the for cycle when count = degree(i)
for j = 1:K*N
    if influence_layer(j,1) == i
        one = one + (1-state(1,influence_layer(j,2),1));
        three = three + state(1,influence_layer(j,2),1);
        count = count+1;
    elseif influence_layer(j,2) == i
        one = one + (1-state(1,influence_layer(j,1),1));
        three = three + state(1,influence_layer(j,1),1);
        count = count+1;
    end
    if count == degree(i)
        break
```

```
42      end
43  end
44
45  one = one/(sum(sum(influence_layer==i))); %first term of Pi0
46  two = u(t); %second term of Pi0
47  three = three/(sum(sum(influence_layer==i))); %first term of Pi1
48  four = k*sum(state(2,:,1)==1)/N; %r(t), second term of Pi1
49  five = c + accumulation(i); %third term of Pi1
50
51  % Payoff(i,t,1) (Pi = Pi1 - Pi0)
52  Pi = (three+four-five) - (one-two) - g*state(3,i,1);
53  %g is the parameter introduced in this new version of the payoff function
54
55  else %payoff for the vaccination decision
56
57  %reaction to the epidemic diffusion
58  one_v = c1*sum(state(2,:,1)==1)/N;
59
60  three_v = 0;
61  four_v = 0;
62
63  %influence (positive or negative) from the influence layer
64  count = 0; %counter for breaking the for cycle when count = degree(i)
65  for j = 1:K*N
66      if influence_layer(j,1) == i
67          three_v = three_v + (1-min(1,state(3,influence_layer(j,2),1)));
68          four_v = four_v + min(1,state(3,influence_layer(j,2),1));
69          count = count+1;
70      elseif influence_layer(j,2) == i
71          three_v = three_v + (1-min(1,state(3,influence_layer(j,1),1)));
72          four_v = four_v + min(1,state(3,influence_layer(j,1),1));
73          %i have to compute min(1,_) because I need 0 or 1, not 2.
74          count = count+1;
75      end
76      if count == degree(i)
77          break
78      end
79  end
80
81  three_v = three_v/(sum(sum(influence_layer==i)));
82  four_v = four_v/(sum(sum(influence_layer==i)));
83
84  %fear of collateral effects of the vaccine
85  five_v = (1-(sum(state(3,:,1)>0)/N)).*exp(-c3*(sum(state(3,:,1)>0)/N));
86
87  %frustration function
88  six_v = accumulation(i);
89
90  %immediate heterogeneous cost of vaccination
```

```
91  seven_v = c4(i);
92
93  %non-pharmaceutical intervention (control function)
94  eight_v = u_v(t);
95
96  %Payoff for the vaccination decision
97  Pi = one_v + two_v + c2*(four_v-three_v) - five_v + six_v - seven_v + eight_v;
98
99  end
100
101 end
```

A.4 Model without vaccine: calibration and validation

In this section we report the MATLAB code used to calibrate and validate the first wave model of chapter 3, applying a grid optimization for the parameters λ, μ, δ, c, γ, k. We obtained the results in Table 3.1 by running several times the following code, testing different combinations for the model's parameters. Depending on how much the grid of values we want to investigate is dense, the matrix `sse_matrix` can be bigger or smaller. At line 74 and 78 we use the function `csvimport` [1] to import the real Italian data from the dataset in [7].

```
1  clc
2  clear all
3  close all
4
5  %%% In this model we have several health states:
6  %%% 0 = S  Susceptible
7  %%% 1 = I  Infected
8  %%% 2 = R  "Removed" (after being recovered, you are immune for some days)
9  %%% 3 = D  Dead
10
11 tic %evaluate time
12
13 global N
14 global K
15 global T
16 global beta
17 global state
18 global influence_layer
19 global degree
20
21 N = 1000; %population size
22 T = 200; %time horizon T
23 K = 5; %number of neighboors connected to each node
24 beta = 0.15; %probability for changing a link in the WattStrogatz graph
```

```
25
26 state = zeros(2,N,2); %state of the individual: decision and health state
27 %we can collect only 2 time steps and update it at every step, instead of
28 %having a big matrix of dimensions (2,N,T).
29 state(2,1,1) = 1; %we start with the only individual number 1 infected
30 %the decision is 0 (risky) or 1 (safe). the health state could be 0
31 %(Supsectible), 1 (Infected), 2 (Removed) or 3 (Death)
32
33 initial_graph = WattsStrogatz(N,K,beta); %initial influence graph
34 influence_layer = table2array(initial_graph.Edges); %influence layer
35 degree = zeros(1,N); %degree of node i in the influence_layer
36 for i = 1:N
37     degree(i) = sum(sum(influence_layer==i)); %calculate the degree
38 end
39
40 % The contact_layer is composed by half of the influence layer,
41 %which never change. The other half is varying random at each
42 % time step. Then the number of total contacts is 2*K*N.
43 contact_layer = zeros(2*K*N,2); %the contact layer is varying in time
44 contact_layer(1:K*N,:) = influence_layer(:,:); %first half
45
46 %global variables, used in the other functions
47 global u
48 global c
49 global gamma
50 global accumulation
51 global k
52
53 %vector that calculates the diffusion of the infected
54 z_evaluated = zeros(1,T);
55 %vector that calculates the fraction of death
56 death_rate = zeros(1,T);
57 u = @(t) 0.75*(t<=30) + 0.3*(t>30); %nonpharmaceutical interventions (NPIs)
58 accumulation = zeros(1,N); %accumulation block for frustration function
59 %c = 0.5; %immediate cost for function f. c>=0
60 %gamma = 0.5; %accumulation factor for function f. 0<=gamma<=1
61 %lambda = 0.012; %per-contact infection probability. 0<=lambda<=1
62 %mu = 0.2; %recover probability. 0<mu<=1
63 %delta = 0.01; %death probability for an infected individual
64 phi = 0; %probability for an R to re-become S (0 for first wave)
65 sigma = 0.5; %effectiveness of self-protective behaviour. 0<=sigma<=1
66 ratio = 6; %rationality: in the paper it was beta.
67 detection_rate = 0.07; %detection rate: not every positive has been founded
68 DD = 14; %detection_delay taken from the paper: in Italy we go from 9 to 14
69
70
71 %Import the real data from Italy. The vector Z contains the real number of
72 %positive people (infected I) day by day, while the vector D contains the
73 %Death people in Italy day by day since 24/02/2020 until 02/06/2022.
```

```
74 [Z,D] = csvimport('Andamento_Nazionale_Italia.csv', ...
75 'columns',{'totale_positivi', 'deceduti'});
76 Z = Z/(60000000); %italian population is 60mln
77 D = D/60000000; %italian population is 60mln
78 [Guariti,totale_casi] = csvimport('Andamento_Nazionale_Italia.csv', ...
79 'columns', {'dimessi_guariti', 'totale_casi'});
80
81
82 %%% Simulation
83
84 sse_matrix = zeros(1*1*1*1,9); %matrix to be populated with square errors
85 %the 9 columns of sse_matrix are:
86 % lambda, mu, delta, c, gamma, k, time, sse_infected, sse_dead
87
88 % Try different configurations of the parameters using neasted for cycle
89 % (lambda, mu, delta, c, gamma, r1, delta, ratio)
90 for index1 = 1:1
91     delta = 0.003; %delta
92     lambda = 0.007;
93     mu = 0.14; %mu
94     for index2 = 1:1
95         c = 0.5; %c immediate cost
96         for index3 = 1:1
97             gamma = 0.25;%accumulation factor gamma
98             for index4 = 1:1
99                 k = 12; %scaling factor k
100
101    %create a matrix for saving the errors in different
102    %simulations using the same configuration
103    SIM = 1;
104    montecarlo = zeros(3,SIM); %sse_I and sse_D
105    montecarlo_I = zeros(2,SIM,T); %fraction and sse_I for every sim
106    montecarlo_D = zeros(2,SIM,T); %fraction and sse_D for every sim
107    montecarlo_avgI = zeros(SIM,T); %for storing the average of simulations
108    montecarlo_avgD = zeros(SIM,T); %for storing the average of simulations
109    average_sse = zeros(1,2); %for calculating next the sse of the average path
110    %(in the first cell the sse_I, in the second the sse_D)
111    prudent_people = zeros(SIM,T); %fraction of people adopting decision 1
112
113                hbar=waitbar(0,'','Name','Iterazioni');
114                for sim = 1:SIM
115                    waitbar(sim/SIM,hbar,sprintf('sim = %d / %d',sim,SIM));
116
117    tic %evaluate the simulation time
118
119    %need to reinitialize the state matrix for each parameters configuration
120    state = zeros(2,N,2); %state of the individual: decision and health state
121    state(2,1,1) = 1; %we always start with the only individual number 1 infected
122
```

```
for t = 1:T-1 %in t=1 there is the initial state
    %fill montecarlo matrices: in 1 the fraction of spread, in 2 the sse
    montecarlo_I(1,sim,t) = sum(state(2,:,1)==1)/N;
    montecarlo_D(1,sim,t) = sum(state(2,:,1)==3)/N;
    montecarlo_I(2,sim,t) = ...
    ((Z(t+DD)/detection_rate)-montecarlo_I(1,sim,t))^2;
    montecarlo_D(2,sim,t) = (D(t+DD)-montecarlo_D(1,sim,t))^2;
    prudent_people(sim,t) = sum(state(1,:,1)==1); %count the prudent people
    %update matrix for calculating the average: this is useful until last
    %sim, because in the last we must enter the If in the simulation that
    %calculates the sse_I and sse_D of the average path.
    montecarlo_avgI(sim,t) = montecarlo_I(1,sim,t);
    montecarlo_avgD(sim,t) = montecarlo_D(1,sim,t);
    %calculate the sse of the mean path
    if sim == SIM
        average_sse(1) = average_sse(1) + ...
        ((Z(t+DD)/detection_rate)-mean(montecarlo_avgI(:,t)))^2;
        average_sse(2) = average_sse(2) + ...
        (D(t+DD)-mean(montecarlo_avgD(:,t)))^2;
    end

    %fill in the random contacts in the second half of the contact_layer
    for j = K*N+1:2*K*N
        %take two random integers between 1 and N
        contact_layer(j,:) = randperm(N,2);
    end

    for i = 1:N %for each individual

        %update recursively the accumulation for the frustration function
        accumulation(i) = gamma*(accumulation(i) + c*state(1,i,1));
        %stochastic decision for adopting protective measures
        %Payoff_matrix(i,t) = Payoff(i,t);
        if rand(1) < (exp(ratio*Payoff(i,t))/(exp(ratio*Payoff(i,t))+1))
            state(1,i,2) = 1; %behave prudent
        else
            state(1,i,2) = 0; %behave not prudent
        end

        %compute the number of infected contacts for current individual i
        N_i = 0; %initialization
        for j = 1:2*K*N
    if (contact_layer(j,1)==i && state(2,contact_layer(j,2),1)==1) || ...
            (contact_layer(j,2)==i && state(2,contact_layer(j,1),1)==1)
        N_i = N_i + 1;
    end
        end

        %%%epidemic dynamics:
```

```
172         %infection
173         if state(2,i,1)==0
174     if (rand(1)<(1-sigma*state(1,i,2))*(1-(1-lambda)^(N_i)) || rand(1)<1/N)
175         state(2,i,2) = 1; %infected (by contact or by external reasons)
176     end
177         %recovery
178         elseif state(2,i,1)==1 && t>1 && rand(1)<mu
179             state(2,i,2) = 2; %recover
180             %death
181             if rand(1)<delta
182                 state(2,i,2) = 3; %dead
183             end
184         %an R returns to be Supsectible (this is useful for the second model)
185         elseif state(2,i,1)==2 && rand(1)<phi
186             state(2,i,2) = 0; %again Supsectible
187         else
188             state(2,i,2) = state(2,i,1);
189         end
190     end
191     state(:,:,1) = state(:,:,2); %update state for the next time step
192 end %end t
193
194 %Calculate Z,D and sse for the montecarlo in the last time step T
195 montecarlo_I(1,sim,T) = sum(state(2,:,1)==1)/N;
196 montecarlo_D(1,sim,T) = sum(state(2,:,1)==3)/N;
197 montecarlo_I(2,sim,T) = ((Z(T+DD)/detection_rate)-montecarlo_I(1,sim,T))^2;
198 montecarlo_D(2,sim,T) = (D(T+DD)-montecarlo_D(1,sim,T))^2;
199 montecarlo_avgI(sim,T) = montecarlo_I(1,sim,T);
200 montecarlo_avgD(sim,T) = montecarlo_D(1,sim,T);
201 % for the last step T, we copy the prudent people value at step T-1. This
202 % method for missing data is called Last Observation Carried Forward (LOCF)
203 prudent_people(sim,T) = prudent_people(sim,T-1);
204 %calculate the sse of the average path at step T
205 if sim == SIM
206         average_sse(1) = average_sse(1) + ...
207         ((Z(T+DD)/detection_rate)-mean(montecarlo_avgI(:,T)))^2;
208         average_sse(2) = average_sse(2) + ...
209         (D(T+DD)-mean(montecarlo_avgD(:,T)))^2;
210 end
211
212 montecarlo(1,sim) = toc;
213 montecarlo(2,sim) = sum(montecarlo_I(2,sim,:));
214 montecarlo(3,sim) = sum(montecarlo_D(2,sim,:));
215                 end %end for the SIM
216                 close(hbar)
217 sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,1) = lambda;
218 sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,2) = mu;
219 sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,3) = delta;
220 sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,4) = c;
```

```
sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,5) ...
= gamma;
sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,6) = k;
sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,7) ...
= mean(montecarlo(1,:));
sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,8) ...
= average_sse(1);
sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,9) ...
= average_sse(2);
%we saved the average of the simulations and we calculated the sse
%of the average path (not the average of the sse).

%calculate the mean along the first dimension
z_evaluated(1,:) = mean(montecarlo_avgI(:,:),1);
death_rate(1,:) = mean(montecarlo_avgD(:,:),1);

standard_devI = std(montecarlo_avgI,0,1); %standard deviation
range_I = zeros(2,T);
%compute the confidence intervals
range_I(1,:) = z_evaluated(1,:) + (standard_devI*norminv(0.025))/sqrt(SIM);
range_I(2,:) = z_evaluated(1,:) + (standard_devI*norminv(0.975))/sqrt(SIM);

average_prudent_people(1,:) = mean(prudent_people(:,:),1);
            end
        end
    end
end


%% PLOTS
figure
gr = tiledlayout(1,3);
title(gr,"Epidemic Plots")
x = 1:1:T;

%first plot: fraction of infected Z(t)
nexttile
hold on
title("Fraction of Infected people Z(t)")
plot(x,z_evaluated(1:T),'LineWidth',1.3)
xlabel("time")
ylabel("fraction of population")
plot(x,Z(1+DD:T+DD)/detection_rate,'LineWidth',1.3)
xlabel("time")
ylabel("fraction of population")
plot(x,range_I(1,:),'b--')
xlabel("time")
ylabel("fraction of population")
plot(x,range_I(2,:),'b--')
```

```
xlabel("time")
ylabel("fraction of population")
legend("model Z(t)","real Z_r(t)")
hold off

%second plot: fraction of prudent
nexttile
hold on
title("Fraction of prudent people (X_i(t) = 1)")
plot(x,(average_prudent_people(1,:))/N, "g",'LineWidth',1.3)
xlabel("time")
ylabel("fraction of population")
hold off

%third plot: fraction of deaths D(t)
nexttile
hold on
title("Fraction of Dead people D(t)")
plot(x,death_rate(1:T),'LineWidth',1.3)
xlabel("time")
ylabel("fraction of population")
plot(x,D(1:T),'LineWidth',1.3)
xlabel("time")
ylabel("fraction of population")
legend("model D(t)","real D_r(t)")
hold off

toc %evaluate total time
```

A.5 Model without vaccine: control

In section 3.3 we investigated the role of the control function $u(t)$, representing the non-pharmaceutical interventions that government may impose during everyday life. The following script has been used to evaluate the impact of alternative policies, obtaining the results shown in Figure 3.4, Figure 3.5, Figure 3.6, Figure 3.7 and Figure 3.8.
Furthermore, the matrix `find_policy` in the following code is exactly what is shown in Table 3.2 and Table 3.3, by testing appropriate values for $u(t)$.

```
clc
clear all
close all

%%% In this model we have several health states:
%%% 0 = S  Susceptible
%%% 1 = I  Infected
%%% 2 = R  "Recovered" (after being recovered, you are immune for some days)
%%% 3 = D  Dead

```

```
11 tic %evaluate time
12
13 global N
14 global K
15 global T
16 global beta
17 global state
18 global influence_layer
19 global degree
20
21 N = 1000; %population size
22 T = 200; %time horizon T
23 K = 5; %number of neighboors connected to each node
24 beta = 0.15; %probability for changing a link in the WattStrogatz graph
25
26 state = zeros(2,N,2); %state of the individual: decision and health state
27 %we can collect only 2 time steps and update it at every step, instead of
28 %having a big matrix of dimensions (2,N,T).
29 state(2,1,1) = 1; %we start with the only individual number 1 infected
30 %the decision is 0 (risky) or 1 (safe). the health state could be 0
31 %(Supsectible), 1 (Infected), 2 (Removed) or 3 (Death)
32
33 initial_graph = WattsStrogatz(N,K,beta); %initial influence graph
34 influence_layer = table2array(initial_graph.Edges); %influence layer
35 degree = zeros(1,N); %degree of node i in the influence_layer
36 for i = 1:N
37     degree(i) = sum(sum(influence_layer==i)); %calculate the degree
38 end
39
40 % The contact_layer is composed by half of the influence  layer,
41 % which never change. The other half is varying random at each
42 % time step. Then the number of total contacts is 2*K*N.
43 contact_layer = zeros(2*K*N,2); %the contact layer is varying in time
44 contact_layer(1:K*N,:) = influence_layer(:,:); %first half of contact_layer
45
46 %global varialbes
47 global u
48 global c
49 global gamma
50 global accumulation
51 global k
52
53 z_evaluated = zeros(1,T); %vector for the diffusion of the infected
54 death_rate = zeros(1,T); %vector for the fraction of death
55 accumulation = zeros(1,N); %accumulation factor for the frustration function
56 %we put phi = 0 for the first wave
57 phi = 0; %probability for an R to re-become S
58 sigma = 0.5; %effectiveness of self-protective behaviour. 0<=sigma<=1
59 ratio = 6; %rationality: in the paper it was beta.
```

```
60  detection_rate = 0.07; %detection rate: not every positive has been founded
61  DD = 14; %detection_delay taken from the paper: in Italy we go from 9 to 14
62
63  % Import the real data from Italy. The vector Z contains the real number of
64  % positive people (infected I) day by day, while the vector D contains the
65  % Death people in Italy day by day since 24/02/2020 until 02/06/2022.
66  [Z,D] = csvimport('Andamento_Nazionale_Italia.csv', 'columns',...
67  {'totale_positivi', 'deceduti'});
68  Z = Z/(60000000); %italian population is 60mln
69  D = D/60000000; %italian population is 60mln
70  [Guariti,totale_casi] = csvimport('Andamento_Nazionale_Italia.csv',...
71  'columns', {'dimessi_guariti', 'totale_casi'});
72
73  %%
74  %%% SIMULATION
75
76  cases = 4; %how many u(t) to evaluate
77
78  plot_matrix_I = zeros(cases,T); %matrix for store all the simulations
79  plot_matrix_D = zeros(cases,T); %matrix for store all the simulations
80  plot_matrix_yes = zeros(cases,T); %matrix for store all the simulations
81  %(in each row there is the average among all sim of the same configuration)
82
83  find_policy = zeros(cases,6); %matrix that aims to find the optimal policy
84  %evaluating different values of u(t)
85
86  % we fix a configuration and test several u(t)
87  %fixed parameters (the best configuration found for the first model)
88  delta = 0.003; %delta
89  lambda = 0.007; %lambda
90  mu = 0.14; %mu
91  c = 0.5; %c
92  gamma = 0.25; %gamma
93  k = 12; %k
94
95  for index = 1:cases %index for u(t)
96
97      %non pharmaceutical interventions (NPI)
98      u = @(t) 0.75*(t<=(10+index*10)) + 0.3*(t>(10+index*10));
99      %u = @(t) 0.4+0.05*index;
100     save_u = (10+index*10); %u to save in the matrix later
101
102     SIM = 50; %simulations for every u(t)
103
104     model_I = zeros(SIM,T);
105     model_D = zeros(SIM,T);
106     model_yes = zeros(SIM,T);
107     simtime = 0; %variable for calculate the time of simulations
108
```

```
109                 for sim = 1:SIM
110     tic %evaluate the simulation time
111
112     %need to reinitialize the state matrix for each configuration
113     state = zeros(2,N,2); %state of individual: decision and health state
114     state(2,1,1) = 1; %start with the individual number 1 infected
115
116 for t = 1:T-1 %in t=1 there is the initial state
117
118     model_I(sim,t) = sum(state(2,:,1)==1)/N; %count the infected people
119     model_D(sim,t) = sum(state(2,:,1)==3)/N; %count the deaths
120     model_yes(sim,t) = sum(state(1,:,1)==1)/N; %count the prudent people
121
122     for j = K*N+1:2*K*N %random contacts in the 2nd half of contact_layer
123         %take two random integers between 1 and N
124         contact_layer(j,:) = randperm(N,2);
125     end
126
127     for i = 1:N %for each individual
128
129         %compute the number of infected contacts for current individual i
130         N_i = 0; %initialization
131         for j = 1:2*K*N
132     if (contact_layer(j,1)==i && state(2,contact_layer(j,2),1)==1) || ...
133             (contact_layer(j,2)==i && state(2,contact_layer(j,1),1)==1)
134         N_i = N_i + 1;
135     end
136         end
137
138         %update the accumulation for the frustration function recursively
139         accumulation(i) = gamma*(accumulation(i) + c*state(1,i,1));
140         %stochastic decision for adopting self-protective measures
141         if rand(1) < (exp(ratio*Payoff(i,t))/(exp(ratio*Payoff(i,t))+1))
142             state(1,i,2) = 1; %behave prudent
143         else
144             state(1,i,2) = 0; %behave not prudent
145         end
146
147         %epidemic dynamics:
148         %infection
149         if state(2,i,1)==0
150     if (rand(1)<(1-sigma*state(1,i,2))*(1-(1-lambda)^(N_i)) || rand(1)<1/N)
151         state(2,i,2) = 1; %infected (by contact or by external people)
152     end
153         %recovery
154         elseif state(2,i,1)==1 && t>1 && rand(1)<mu
155             state(2,i,2) = 2; %recover
156             %death
157             if rand(1)<delta
```

```
158                 state(2,i,2) = 3; %dead
159             end
160         %an R returns to be Supsectible (this is for the second model)
161         elseif state(2,i,1)==2 && rand(1)<phi
162             state(2,i,2) = 0; %again Supsectible
163         else
164             state(2,i,2) = state(2,i,1);
165         end
166     end
167     state(:,:,1) = state(:,:,2); %update state for the next time step
168 end %end t
169
170 model_I(sim,T) = sum(state(2,:,1)==1)/N;
171 model_D(sim,T) = sum(state(2,:,1)==3)/N;
172 % for the last step T, we copy the prudent people value at step T-1.
173 model_yes(sim,T) = model_yes(sim,T-1);
174
175 simtime_single = toc;
176 simtime = simtime + simtime_single; %evaluate time
177                 end %end for the SIM
178
179 find_policy(index,1) = save_u; %store the value of u
180 find_policy(index,2) = simtime/SIM; %average time for 1 simulation
181 find_policy(index,3) = mean(mean(model_yes(:,:))); %average prudent people
182 find_policy(index,4) = mean(mean(model_I(:,:))); %average infected people
183 find_policy(index,5) = mean(max(model_I(:,:),[],2)); %mean of max infected
184 find_policy(index,6) = mean(model_D(:,T)); %average death people at T.
185
186 plot_matrix_I(index,:) = mean(model_I(:,:),1);
187 plot_matrix_D(index,:) = mean(model_D(:,:),1);
188 plot_matrix_yes(index,:) = mean(model_yes(:,:),1);
189
190 end
191
192 toc %evaluate total time
193
194 %% PLOTS
195 figure
196 gr = tiledlayout(1,3);
197 title(gr,"Epidemic Plots")
198 x = 1:1:T;
199
200 %plot of infected Z(t)
201 nexttile
202 hold on
203 title("Fraction of Infected people Z(t)")
204 xlabel("time")
205 ylabel("fraction of population")
206 legend
```

```
for p = 1:cases
    %plot(x,plot_matrix_I(p,1:T),'DisplayName',...
    %['u(t)= {0.75 (t<=100), ',num2str(0.22+p*0.02), ' (t>100)}'],...
    %'LineWidth',1.1)
    plot(x,plot_matrix_I(p,1:T),'DisplayName',...
    ['Changing point: t=',num2str(10+p*10), '.'],'LineWidth',1.1)
    %plot(x,plot_matrix_I(p,1:T),'DisplayName',...
    %['u(t)= ',num2str(0.4+0.05*p),'.'],'LineWidth',1.1)
end
plot(x,Z(1+DD:T+DD)/detection_rate,"k--",'DisplayName','real Z_r(t)')
xlabel("time")
ylabel("fraction of population")
hold off

%plot of people adopting self-protective behavior
nexttile
hold on
title("Fraction of prudent people (X_i(t) = 1)")
legend('Location','southeast')
for p = 1:cases
    %plot(x,plot_matrix_yes(p,:),'DisplayName',...
    %['u(t)= {0.75 (t<=100), ',num2str(0.22+p*0.02), ' (t>100)}'],...
    %'LineWidth',1.1)
    plot(x,plot_matrix_yes(p,1:T),'DisplayName',...
    ['Changing point: t=',num2str(10+p*10), '.'],'LineWidth',1.1)
    %plot(x,plot_matrix_yes(p,:),'DisplayName',...
    %['u(t)= ',num2str(0.4+0.05*p),'.'],'LineWidth',1.1)
    xlabel("time")
    ylabel("fraction of population")
end
%legend("model prudent people")
hold off

%plot of the fraction of deaths
nexttile
hold on
title("Fraction of Dead people D(t)")
xlabel("time")
ylabel("fraction of population")
legend('Location','northwest')
for p = 1:cases
    %plot(x,plot_matrix_D(p,1:T),'DisplayName',...
    %['u(t)= {0.75 (t<=100), ',num2str(0.22+p*0.02), ' (t>100)}'],...
    %'LineWidth',1.1)
    plot(x,plot_matrix_D(p,1:T),'DisplayName',...
    ['Changing point: t=',num2str(10+p*10), '.'],'LineWidth',1.1)
    %plot(x,plot_matrix_D(p,1:T),'DisplayName',...
    %['u(t)= ',num2str(0.4+0.05*p),'.'],'LineWidth',1.1)
end
```

```
256 plot(x,D(1:T),"k--",'DisplayName','real D_r(t)')
257 hold off
```

A.6 Model with vaccine: calibration and validation

In this section we report the MATLAB code used to calibrate and validate the model with the vaccination dynamic of chapter 4, applying a grid optimization for all the parameters that we need to calibrate.
We obtained the results in Table 4.1 by running several times the following code, testing different combinations for the epidemic and model's parameters. Depending on how much the grid of values we want to investigate is dense, the matrix `sse_matrix` can be bigger or smaller.
At line 80 we use the function `csvimport` [1] to import the real Italian data from the dataset in [7], exactly as we did for the model without vaccine in section A.4.

```
1  clc
2  clear all
3  close all
4
5  %%% In this model we have several health states:
6  %%% 0 = S  Susceptible
7  %%% 1 = I  Infected
8  %%% 2 = R  "Recovered" (after being recovered, you are immune for some days)
9  %%% 3 = D  Dead
10
11 %%% We have two decision processes: one decide wether to adopt or not to
12 %%% adopt safety measures, and one to decide wether to get vaccinated or not.
13
14 tic %evaluate time
15
16 %global variables useful for the functions
17 global N
18 global K
19 global T
20 global beta
21 global state
22 global degree
23 global influence_layer
24 global u
25 global u_v
26 global c
27 global gamma
28 global accumulation
29 global g
30 global k
31
32 % NETWORK PARAMETERS
```

```
N = 1000; %population size n
T = 200; %time horizon T
K = 5; %number of neighboors connected to each node in influence layer
beta = 0.15; %probability for changing a link in the WattStrogatz graph

% STATE MATRIX & NETWORK LAYERS:
% 1-first line decision x_i. The decision is 0 (risky) or 1 (safe).
% 2-second line health state S,I,(IV),R,D. The health state could be:
%   0 (Supsectible), 1 (Infected), 2 (Recovered) or 3 (Death).
% 3-third line vaccination decision v_i
state = zeros(3,N,2); %state of the individual: decision and health state
%we add a third line in the state matrix: it is for the vaxination. It will
%be 0 if novax, 1 if sivax. If it is 1 it can't return to 0, but if it's 1
%it will surely go to 2
%we can collect only 2 time steps and update it at every step, instead of
%having a big matrix of dimensions (3,N,T).

%Influence Layer: we use a WattsStrogatz graph
initial_graph = WattsStrogatz(N,K,beta); %initial influence graph
influence_layer = table2array(initial_graph.Edges); %influence layer
degree = zeros(1,N); %degree of node i in the influence_layer
for i = 1:N
    degree(i) = sum(sum(influence_layer==i)); %calculate the degree
end
%The average node degree is 2K.

% Contact_layer is composed by half of the influence layer,
% The other half is varying random at each time step.
% The number of total contacts (rows of the contact_layer) is indeed 2KN.
contact_layer = zeros(2*K*N,2); %the contact layer is varying in time
%the first half of the contact_layer is the influence layer
contact_layer(1:K*N,:) = influence_layer(:,:);

z_evaluated = zeros(1,T); %vector for the diffusion of the infected
death_rate = zeros(1,T); %vector for the fraction of death

% MODEL FUNCTIONS AND PARAMETERS
u = @(t) 0.6*(t<=100) + 0.5*(t>100); %nonpharmaceutical interventions (NPIs)
u_v = @(t) 1-u(t); %green pass and other measures
accumulation = zeros(1,N); %accumulation factor for the frustration function
ratio = 6; %rationality: in the paper it was beta.

% EPIDEMIC PARAMETERS
% Import the real data from Italy. The vector Z contains the real number of
% positive people (infected I) day by day, while the vector D contains the
% Death people in Italy day by day since 24/02/2020 until 02/06/2022.
[Z,D,Guariti,totale_casi] = ...
csvimport('Andamento_Nazionale_Italia.csv', 'columns',...
    {'totale_positivi', 'deceduti', 'dimessi_guariti', 'totale_casi'});
```

```
82  Z = Z/(60000000); %italian population is 60mln
83  D = D/60000000; %italian population is 60mln
84  detection_rate = 0.4; %detection rate: not every positive has been founded
85  DD = 10; %detection_delay taken from the paper: in Italy we go from 9 to 14
86  mu = 0.14; %recovery probability. We can fix mu=0.14, i.e. 5 days.
87  phi = 0.1; %probability for an R to re-become S
88  sigma = 0.5; %effectiveness of self-protective behaviour. 0<=sigma<=1
89  alpha = 0.7; %effectiveness of the vaccine. 0<=alpha<=1
90  alpha_2 = 0.5; %protection to death for a vaccinated person
91
92  % PARAMETERS FOR THE VACCINATION PAYOFF FUNCTION
93  global c1
94  global c2
95  global c3
96  global c4
97
98  c1 = 12;
99  c2 = 1.1;
100 c3 = 2; %the bigger, the more fear of the vaccine
101 c4 = [0.5*ones(1,N*0.9),1.5*ones(1,N*0.1)]; %eterogeneous immediate cost
102 g = 0.0018;
103
104 % Loading the vectors containing the number of vaccinations
105 load('V.mat');
106 load('V_tot.mat');
107
108 %%
109 %%% SIMULATION
110
111 sse_matrix = zeros(1*1*1*1,10); %matrix to be populated with square errors
112 %the 10 columns of sse_matrix are:
113 % lambda, delta, g, c, gamma, k,
114 % time, sse_infected, sse_dead, sse_vaccinated
115
116 % Try different configurations of the parameters
117 for index1 = 1:1
118     g = 0.0018;
119     delta = 0.002; %delta
120     lambda = 0.008; %lambda
121     for index2 = 1:1
122         c = 0.5; %c immediate cost for self-protective behavior
123         for index3 = 1:1
124             gamma = 0.51; %gamma accumulation factor
125             for index4 = 1:1
126                 k = 12; %k scaling factor
127
128     %create a matrix for saving the errors in different
129     %simulations using the same configuration
130     SIM = 100; %number of simulations
```

```
131     montecarlo = zeros(3,SIM); %sse_I and sse_D
132     montecarlo_I = zeros(2,SIM,T); %fraction and sse_I for every sim
133     montecarlo_D = zeros(2,SIM,T); %fraction and sse_D for every sim
134     montecarlo_V = zeros(2,SIM,T); %fraction and sse_V for every sim
135     montecarlo_avgI = zeros(SIM,T); %for the average of simulations
136     montecarlo_avgD = zeros(SIM,T); %for the average of simulations
137     montecarlo_avgV = zeros(SIM,T); %for the average of simulations
138     average_sse = zeros(1,3); %for the sse of the average path
139     %(in the first cell the sse_I, in second the sse_D, in third the sse_V)
140     prudent_people = zeros(SIM,T); %fraction of people adopting decision 1
141     vaccinated_people = zeros(SIM,T); %fraction of people vaccinated
142
143                 hbar=waitbar(0,'','Name','Iterazioni');
144                 for sim = 1:SIM
145                     waitbar(sim/SIM,hbar,sprintf('sim = %d / %d',sim,SIM));
146     tic %evaluate the simulation time
147
148     %need to reinitialize the state matrix for each parameters configuration
149     %state of the individual: decision prudent, health state, vax state
150     state = zeros(3,N,2);
151     %On the 27/12/2020 the population infected in Italy was about the 1%
152     %of the total, that means 10 people every 1000 individuals (our n).
153     %But we also have to adjust the number dividing by the detection_rate,
154     %so we need 10/detection rate = 24 .
155     state(2,randperm(N,24),1) = 1; %initial health state on 27/12/2020
156     state(1,randperm(N,N),1) = 1; %initial behaviour on 27/12/2020
157
158     Payoff_matrix = zeros(N,T);
159     Payoff_matrix_vax = zeros(N,T);
160
161 for t = 1:T-1 %in t=1 there is the initial state
162
163     %fill montecarlo matrices: in 1 the fraction of spread, in 2 the sse
164     montecarlo_I(1,sim,t) = sum(state(2,:,1)==1)/N; %infected
165     montecarlo_D(1,sim,t) = sum(state(2,:,1)==3)/N; %deaths
166     montecarlo_V(1,sim,t) = sum(state(3,:,1)>0)/N; %vaccinated
167     prudent_people(sim,t) = sum(state(1,:,1)==1); %count the prudent people
168     vaccinated_people(sim,t) = sum(state(3,:,1)>0); %count the vaccinated
169     %update matrix for calculating the average: this is useful until last
170     %sim, because in the last we must enter the If in the simulation that
171     %calculates the sse_I, sse_D, sse_V of the average path.
172     montecarlo_avgI(sim,t) = montecarlo_I(1,sim,t);
173     montecarlo_avgD(sim,t) = montecarlo_D(1,sim,t);
174     montecarlo_avgV(sim,t) = montecarlo_V(1,sim,t);
175
176     for j = K*N+1:2*K*N %random contacts in the 2nd half of contact_layer
177         %take two random integers between 1 and N
178         contact_layer(j,:) = randperm(N,2);
179     end
```

```
180
181     for i = 1:N %for each individual
182
183         %compute the number of infected contacts for current individual i
184         N_i = 0; %initialization
185         for j = 1:2*K*N
186             if (contact_layer(j,1)==i && state(2,contact_layer(j,2),1)==1) || ...
187                     (contact_layer(j,2)==i && state(2,contact_layer(j,1),1)==1)
188                 N_i = N_i + 1;
189             end
190         end
191
192         %update the accumulation for the frustration using recursive formula
193         accumulation(i) = gamma*(accumulation(i) + c*state(1,i,1));
194
195         % DECISIONS BASED ON GAME THEORY
196         Payoff_matrix(i,t) = Payoff_vax(i,t,0);
197         %stochastic daily decision for adopting self-protective measures
198         if rand(1) < ...
199         (exp(ratio*Payoff_matrix(i,t))/(exp(ratio*Payoff_matrix(i,t))+1))
200             state(1,i,2) = 1; %behave prudent
201         else
202             state(1,i,2) = 0; %behave not prudent
203         end
204
205         Payoff_matrix_vax(i,t) = Payoff_vax(i,t,1);
206         %stochastic decision (if v(i) = 0) for getting vaccinated
207         if state(3,i,1) == 0 && rand(1) < ...
208         (exp(ratio*Payoff_matrix_vax(i,t))/(exp(ratio*Payoff_matrix_vax(i,t))+1))
209             state(3,i,2) = 1; %get vaccinated,
210             % but one has to wait some days in order to receive the
211             %immunity, i.e. becoming 2 instead of 1.
212         elseif state(3,i,1) == 1 && rand(1) < 1/14 %external infected prob
213             %thanks to the elseif there is no the possibility to be immune
214             %the same day of the vaccination.
215             state(3,i,2) = 2; %become immune
216         else
217             state(3,i,2) = state(3,i,1); %remains unchanged
218         end
219
220         %EPIDEMIC DYNAMICS
221         %infection without vaccine protection
222         if state(2,i,1)==0 && state(3,i,1)<2
223             if...
224     rand(1)<(1-sigma*state(1,i,1))*(1-(1-lambda)^(N_i)) || rand(1)<1/N
225                 state(2,i,2) = 1; %infected
226             end
227         %infection with vaccine protection
228         elseif state(2,i,1)==0 && state(3,i,1)==2
```

```
229             if rand(1)<...
230     (1-alpha)*(1-sigma*state(1,i,1))*(1-(1-lambda)^(N_i)) || ...
231     rand(1)<(1-alpha)*1/N
232                 state(2,i,2) = 1; %infected
233             end
234         %recovery
235         elseif state(2,i,1)==1 && t>1 && rand(1)<mu
236             state(2,i,2) = 2; %recover
237             %death without vaccine
238             if state(3,i,1)<2 && rand(1)<delta
239                 state(2,i,2) = 3; %dead
240             end
241             %death with vaccine
242             if state(3,i,1)==2 && rand(1)<(1-alpha_2)*delta
243                 state(2,i,2) = 3; %dead
244             end
245         %an R returns to be Supsectible
246         elseif state(2,i,1)==2 && rand(1)<phi
247             state(2,i,2) = 0; %again Supsectible
248         else
249             state(2,i,2) = state(2,i,1);
250         end
251     end
252     %update the state matrix
253     state(:,:,1) = state(:,:,2); %update state for the next time step
254 end %end t
255
256 %Calculate Z,D and sse for the montecarlo in the last time step T
257 montecarlo_I(1,sim,T) = sum(state(2,:,1)==1)/N;
258 montecarlo_D(1,sim,T) = sum(state(2,:,1)==3)/N;
259 montecarlo_V(1,sim,T) = sum(state(3,:,1)>0)/N;
260 montecarlo_avgI(sim,T) = montecarlo_I(1,sim,T);
261 montecarlo_avgD(sim,T) = montecarlo_D(1,sim,T);
262 montecarlo_avgV(sim,T) = montecarlo_V(1,sim,T);
263
264 prudent_people(sim,T) = sum(state(1,:,1)==1); %count the prudent people
265 vaccinated_people(sim,T) = sum(state(3,:,1)>0); %count the vaccinated people
266 %calculate the sse of the mean path at step T
267
268 montecarlo(1,sim) = toc;
269 montecarlo(2,sim) = sum(montecarlo_I(2,sim,:));
270 montecarlo(3,sim) = sum(montecarlo_D(2,sim,:));
271                 end %end for the SIM
272                 close(hbar)
273 sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,1) ...
274 = lambda;
275 sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,2) ...
276 = delta;
277 sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,3) ...
```

```
278 = g;
279 sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,4) ...
280 = c;
281 sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,5) ...
282 = gamma;
283 sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,6) ...
284 = k;
285 sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,7) ...
286 = mean(montecarlo(1,:));
287 sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,8) ...
288 = average_sse(1);
289 sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,9) ...
290 = average_sse(2);
291 sse_matrix((index1-1)*(1*1*1)+(index2-1)*(1*1)+(index3-1)*(1)+index4,10) ...
292 = average_sse(3);
293 %we saved the average of the simulations and we calculated the sse
294 %of the average path (not the average of the sse).
295
296 %calculate the mean along the first dimension to do the plots
297 z_evaluated(1,:) = mean(montecarlo_avgI(:,:),1);
298 death_rate(1,:) = mean(montecarlo_avgD(:,:),1);
299 average_prudent_people(1,:) = mean(prudent_people(:,:),1);
300 average_vaccinated_people(1,:) = mean(vaccinated_people(:,:),1);
301
302 standard_devI = std(montecarlo_avgI,0,1); %standard deviation
303 range_I = zeros(2,T);
304 %compute confidence intervals
305 range_I(1,:) = z_evaluated(1,:) + (standard_devI*norminv(0.025))/sqrt(SIM);
306 range_I(2,:) = z_evaluated(1,:) + (standard_devI*norminv(0.975))/sqrt(SIM);
307
308             end
309         end
310     end
311 end
312
313 %% PLOTS
314 figure
315 gr = tiledlayout(1,2);
316 title(gr,"Epidemic Plots")
317 x = 1:1:T;
318
319 nexttile
320 hold on
321 title("Infected people Z(t)")
322 plot(x,z_evaluated(1:T),'LineWidth',1.1)
323 xlabel("time")
324 ylabel("fraction of population")
325 plot(x,Z(1+308+DD:T+308+DD)/detection_rate,'LineWidth',1.1)
326 xlabel("time")
```

```
ylabel("fraction of population")
plot(x,range_I(1,:),'b--')
xlabel("time")
ylabel("fraction of population")
plot(x,range_I(2,:),'b--')
xlabel("time")
ylabel("fraction of population")
legend("model Z(t)","real Z_r(t)")
hold off

nexttile
hold on
title("Prudent people (X_i(t) = 1) and Vaccinated people (V_i(t) > 0)")
plot(x,(average_prudent_people(1,:))/N, "g",'LineWidth',1.1)
xlabel("time")
ylabel("fraction of population")
plot(x,(average_vaccinated_people(1,:))/N, "y",'LineWidth',1.1)
xlabel("time")
ylabel("fraction of population")
legend("model prudent people","model vaccinated people",...
'Location','southwest')
hold off

toc %evaluate total time
```

A.7 Model with vaccine: control

In section 4.2 we investigated the role of the control functions $u(t)$ and $u_v(t)$, representing respectively the non-pharmaceutical restrictions that government may impose during everyday life in order to limit the epidemic spread and those regarding the vaccination campaign.
The following script has been used to evaluate the impact of alternative combinations of those control functions, always satisfying the budget constraint introduced in (4.7). The results obtained are shown in Figure 4.2, Figure 4.3 and Figure 4.4.
Furthermore, the matrix `find_policy` in the following code is exactly what is shown in Table 4.2, Table 4.3 and Table 4.4, by testing appropriate values for the combination of $u(t)$ and $u_v(t)$.

```
clc
clear all
close all

%%% In this model we have several health states:
%%% 0 = S  Susceptible
%%% 1 = I  Infected
%%% 2 = R  "Recovered" (after being recovered, you are immune for some days)
%%% 3 = D  Dead
```

```
10
11 % We also have two decision processes: one decide wether to adopt or not to
12 % adopt safety measures, and one to decide wether to get or no vaxinated.
13
14 tic %evaluate time
15
16 global N
17 global K
18 global T
19 global beta
20 global state
21 global degree
22 global influence_layer
23 global u
24 global u_v
25 global c
26 global gamma
27 global accumulation
28 global g
29 global k
30
31 % NETWORK PARAMETERS
32 N = 1000; %population size
33 T = 200; %time horizon T
34 K = 5; %number of neighboors connected to each node
35 beta = 0.15; %probability for changing a link in the WattStrogatz graph
36
37 % STATE MATRIX & NETWORK LAYERS:
38 % 1-first line decision x_i. The decision is 0 (risky) or 1 (safe).
39 % 2-second line health state S,I,(IV,)R,D. The health state could be:
40 %   0 (Supsectible), 1 (Infected), 2 (Recovered) or 3 (Death).
41 % 3-third line vaxine decision v_i
42 state = zeros(3,N,2); %state of the individual: decision and health state
43 %we add a third line in the state matrix for the vaccination. It will
44 %be 0 if novax, 1 if sivax. If it is 1 it can't return to 0, but if it's 1
45 %it will surely go to 2
46 %we may collect only 2 time steps and update it at every step, instead of
47 %having a big matrix of dimensions (3,N,T).
48
49 %Initial state: we need to compute it with real italian data
50 state(2,1,1) = 1; %we start with the only individual number 1 infected
51
52 %Influence Layer: we use a WattsStrogatz graph
53 initial_graph = WattsStrogatz(N,K,beta); %initial influence graph
54 influence_layer = table2array(initial_graph.Edges); %influence layer
55 degree = zeros(1,N); %degree of node i in the influence_layer
56 for i = 1:N
57     degree(i) = sum(sum(influence_layer==i)); %calculate the degree
58 end
```

```
%The average node degree is 2K.

% Contact_layer is half made up of influence layer, which never change.
% The other half is varying random at each time step.
% The number of total contacts (rows of the contact_layer) is indeed 2KN.
contact_layer = zeros(2*K*N,2); %the contact layer is varying in time
contact_layer(1:K*N,:) = influence_layer(:,:); %first half of contact_layer

z_evaluated = zeros(1,T); %vector for the diffusion of the infected
death_rate = zeros(1,T); %vector for the fraction of deaths

% MODEL FUNCTIONS AND PARAMETERS
%u = @(t) 0.5; %nonpharmaceutical interventions (NPIs)
%u_v = @(t) 0.5; %green pass and other measures
accumulation = zeros(1,N); %accumulation factor for the frustration function
ratio = 6; %rationality: in the paper it was beta.

% EPIDEMIC PARAMETERS
% Import the real data from Italy. The vector Z contains the real number of
% positive people (infected I) day by day; vector D contains the real
% number of Deaths in Italy day by day since 24/02/2020 until 02/06/2022.
[Z,D,Guariti,totale_casi] = ...
csvimport('Andamento_Nazionale_Italia.csv', 'columns', ...
    {'totale_positivi', 'deceduti', 'dimessi_guariti', 'totale_casi'});
Z = Z/(60000000); %italian population is 60mln
D = D/60000000; %italian population is 60mln
detection_rate = 0.4; %detection rate: not every positive has been founded
DD = 10; %detection_delay taken from the paper: in Italy we go from 9 to 14
phi = 0.1; %probability for an R to re-become S
sigma = 0.5; %effectiveness of self-protective behaviour. 0<=sigma<=1
alpha = 0.7; %effectiveness of the vaccine. 0<=alpha<=1
alpha_2 = 0.5; %protection against death for a vaccinated person

% PARAMETERS FOR THE VACCINATION PAYOFF FUNCTION
global c1
global c2
global c3
global c4

c1 = 12;
c2 = 1.1;
c3 = 2; %the bigger, the more fear of the vaccine
%eterogeneous immediate cost of vaccination [vax, novax]
c4 = [0.5*ones(1,N*0.9),1.5*ones(1,N*0.1)];
g = 0.0018;

% Loading the vectors containing the number of vaxinations
load('V.mat');
load('V_tot.mat');
```

```
108
109 %%
110 %%% SIMULATION
111
112 cases = 5; %number of different cases to test
113
114 plot_matrix_I = zeros(cases,T); %matrix for store all the simulations
115 plot_matrix_D = zeros(cases,T); %matrix for store all the simulations
116 plot_matrix_yes = zeros(cases,T); %matrix for store all the simulations
117 plot_matrix_vax = zeros(cases,T); %matrix for store all the simulations
118 %(in each row there is the average among all sim of the same configuration)
119
120 find_policy = zeros(cases,7); %matrix that aims to find the optimal policy
121 %evaluating different values of u(t) and u_v(t)
122
123 % we fix a configuration and test several u(t) and u_v(t)
124 %fixed parameters
125 delta = 0.002; %delta
126 lambda = 0.008; %lambda
127 mu = 0.14; %mu
128 c = 0.5; %c
129 gamma = 0.51; %gamma
130 k = 12; %k
131
132 hbar=waitbar(1,'','Name','Iterazioni');
133 for index = 1:cases %index for u(t) (now I have only one index)
134
135     %u = @(t) 0.6*(t<=(60+20*index)) + 0.5*(t>(60+20*index));
136     %u_v = @(t) 1 - u(t);
137     u = @(t) 0.2 + 0.1*index; %npi for the self-protective behavior
138     u_v = @(t) 1 - u(t); %npi for the vax (u_v(t) = 1-u(t))
139     %we can assume the budget constraint u(t) + u_v(t) = 1
140     %even if it should be u(t) + u_v(t) <= 1
141     save_u = 0.2 + 0.1*index; %u to save in the matrix later
142     save_u_v = 1-save_u;
143
144     SIM = 50; %simulations using the same configuration
145
146     model_I = zeros(SIM,T);
147     model_D = zeros(SIM,T);
148     model_yes = zeros(SIM,T);
149     model_vax = zeros(SIM,T);
150     simtime = 0; %variable for calculate the time of simulations
151
152                 %hbar=waitbar(0,'','Name','Iterazioni');
153                 for sim = 1:SIM
154                     waitbar((((index-1)*SIM)+sim)/(SIM*cases),hbar,...
155                 sprintf('sim = %d / %d',(((index-1)*SIM)+sim),SIM*cases));
156     tic %evaluate the simulation time
```

```
157
158     %reinitialize the state matrix for each parameters configuration
159     %state of the individual: decision prudent, health state, vax state
160     state = zeros(3,N,2);
161     %On the 27/12/2020 the population infected in Italy was about the 1%
162     %of the total, that means 10 people every 1000 individuals (our size).
163     %But we also have to adjust the number dividing by the detection_rate,
164     %so we need 10/detection rate = 24.
165     state(2,randperm(N,24),1) = 1; %initial health state on 27/12/2020
166     state(1,randperm(N,N),1) = 1; %initial behaviour on 27/12/2020
167
168 for t = 1:T-1 %in t=1 there is the initial state
169
170     model_I(sim,t) = sum(state(2,:,1)==1)/N; %count the infected people
171     model_D(sim,t) = sum(state(2,:,1)==3)/N; %count the death people
172     model_yes(sim,t) = sum(state(1,:,1)==1)/N; %count the prudent people
173     model_vax(sim,t) = sum(state(3,:,1)>0)/N; %count the vaccinated people
174
175     for j = K*N+1:2*K*N %random contacts in second half of contact_layer
176         %take two random integers between 1 and N
177         contact_layer(j,:) = randperm(N,2);
178     end
179
180     for i = 1:N %for each individual
181
182         %compute the number of infected contacts for current individual i
183         N_i = 0; %initialization
184         for j = 1:2*K*N
185         if (contact_layer(j,1)==i && state(2,contact_layer(j,2),1)==1) ...
186         || (contact_layer(j,2)==i && state(2,contact_layer(j,1),1)==1)
187             N_i = N_i + 1;
188         end
189         end
190
191         %update the accumulation using recursive formula
192         accumulation(i) = gamma*(accumulation(i) + c*state(1,i,1));
193
194         % DECISIONS BASED ON GAME THEORY
195         %stochastic daily decision for adopting self-protective measures
196         if rand(1) < ...
197         (exp(ratio*Payoff_vax(i,t,0))/(exp(ratio*Payoff_vax(i,t,0))+1))
198             state(1,i,2) = 1; %behave prudent
199         else
200             state(1,i,2) = 0; %behave not prudent
201         end
202
203         %stochastic decision (if v(i) = 0) for getting vaccinated
204         if state(3,i,1) == 0 && rand(1) < ...
205         (exp(ratio*Payoff_vax(i,t,1))/(exp(ratio*Payoff_vax(i,t,1))+1))
```

```
206             state(3,i,2) = 1; %get vaccinated, but not yet immune
207         elseif state(3,i,1) == 1 && rand(1) < 1/14 %external infection
208             % interval between the first and the second dose. Here we use
209             % 14 as the days needed for being immune after getting vaccin.
210             %thanks to the elseif there is no the possibility to be immune
211             %the same day of the vaccination.
212             state(3,i,2) = 2; %become immune
213         else
214             state(3,i,2) = state(3,i,1); %remains unchanged
215         end
216 
217         %EPIDEMIC DYNAMICS
218         %infection without vaccine protection
219         if state(2,i,1)==0 && state(3,i,1)<2
220             if rand(1)< ...
221             (1-sigma*state(1,i,1))*(1-(1-lambda)^(N_i)) || rand(1)<1/N
222                 state(2,i,2) = 1; %infected
223             end
224         %infection with vaccine protection
225         elseif state(2,i,1)==0 && state(3,i,1)==2
226             if rand(1)< ...
227             (1-alpha)*(1-sigma*state(1,i,1))*(1-(1-lambda)^(N_i)) || ...
228             rand(1)<(1-alpha)*1/N
229                 state(2,i,2) = 1; %infected
230             end
231         %recovery
232         elseif state(2,i,1)==1 && t>1 && rand(1)<mu
233             state(2,i,2) = 2; %recover
234             %death without vaccine
235             if state(3,i,1)<2 && rand(1)<delta
236                 state(2,i,2) = 3; %dead
237             end
238             %death with vaccine
239             if state(3,i,1)==2 && rand(1)<(1-alpha)*delta
240                 state(2,i,2) = 3; %dead
241             end
242         %an R returns to be Supsectible
243         elseif state(2,i,1)==2 && rand(1)<phi
244             state(2,i,2) = 0; %again Supsectible
245         else
246             state(2,i,2) = state(2,i,1);
247         end
248     end
249     %update the state matrix
250     state(:,:,1) = state(:,:,2); %update state for the next time step
251 end %end t
252 
253 model_I(sim,T) = sum(state(2,:,1)==1)/N;
254 model_D(sim,T) = sum(state(2,:,1)==3)/N;
```

```
% for the last step T, we copy the prudent and vaccinated at step T-1.
model_yes(sim,T) = model_yes(sim,T-1);
model_vax(sim,T) = model_vax(sim,T-1);

simtime_single = toc;
simtime = simtime + simtime_single; %evaluate time
                end %end for the SIM
                %close(hbar)

find_policy(index,1) = save_u; %store the value of u
find_policy(index,2) = save_u_v; %store the value of u_v
find_policy(index,3) = simtime/SIM; %avg time for 1 simulation
find_policy(index,4) = mean(mean(model_yes(:,:))); %avg prudent people
find_policy(index,5) = mean(mean(model_vax(:,:))); %avg vaccinated people
find_policy(index,6) = mean(mean(model_I(:,:))); %avg infected people
find_policy(index,7) = mean(max(model_I(:,:),[],2)); %avg max of infected
find_policy(index,8) = mean(model_D(:,T)); %avg deaths at time T.

plot_matrix_I(index,:) = mean(model_I(:,:),1);
plot_matrix_D(index,:) = mean(model_D(:,:),1);
plot_matrix_yes(index,:) = mean(model_yes(:,:),1);
plot_matrix_vax(index,:) = mean(model_vax(:,:),1);

end
close(hbar)

toc %evaluate total time

%% PLOTS
figure
gr = tiledlayout(1,3);
title(gr,"Epidemic Plots")
x = 1:1:T;

%plot infected Z(t)
nexttile
hold on
title("Infected people Z(t)")
xlabel("time")
ylabel("fraction of population")
legend
for p = 1:cases
    %plot(x,plot_matrix_I(p,1:T),'DisplayName',...
    %['Changing point t=', num2str((60+20*p)), '.'],'LineWidth',1.1)
    plot(x,plot_matrix_I(p,1:T),'DisplayName',...
['u(t)=', num2str((0.2+0.1*p)), ', u_v(t)=',num2str(1-(0.2+0.1*p)),'.'],...
    'LineWidth',1.1)
end
plot(x,Z(1+308+DD:T+308+DD)/detection_rate, ...
```

```
304 "k--",'DisplayName','real Z_r(t)')
305 xlabel("time")
306 ylabel("fraction of population")
307 hold off
308
309 %plot fraction of people adopting self-protective measures
310 nexttile
311 hold on
312 title("Prudent people (X_i(t) = 1)")
313 legend
314 for p = 1:cases
315     plot(x,plot_matrix_yes(p,:),'DisplayName',...
316 ['u(t)=', num2str(0.2+0.1*p), ', u_v(t)=',num2str(1-(0.2+0.1*p)),'.'],...
317     'LineWidth',1.1)
318     %plot(x,plot_matrix_yes(p,:),'DisplayName',...
319     %['Changing point t=', num2str((60+20*p)), '.'],'LineWidth',1.1)
320     xlabel("time")
321     ylabel("fraction of population")
322 end
323 hold off
324
325 %plot the fraction of people vaccinated
326 nexttile
327 hold on
328 title("Vaccinated people (V_i(t) > 0)")
329 legend('Location','northwest')
330 for p = 1:cases
331     plot(x,plot_matrix_vax(p,:),'DisplayName',...
332 ['u(t)=', num2str(0.2+0.1*p), ', u_v(t)=',num2str(1-(0.2+0.1*p)),'.'],...
333     'LineWidth',1.1)
334     %plot(x,plot_matrix_vax(p,:),'DisplayName',...
335     %['Changing point t=', num2str((60+20*p)), '.'],'LineWidth',1.1)
336     xlabel("time")
337     ylabel("fraction of population")
338 end
339 hold off
```