



UNIVERSITY OF MILANO - BICOCCA

School of Science

Department of Informatics, Systems and Communication

Master's degree course in Computer Science

Covid-19 infection inspired diffusion dynamics: generalization of a membrane-based simulator

Supervisor: Prof. Claudio Zandron

Co-supervisor: Prof. Giuditta Franco

Master's Thesis by:
Sandro Erba
Matricola 856327

Academic Year 2023-2024

Contents

1 State of the art	5
1.1 Membrane systems	5
1.1.1 New properties for membrane rewriting rules	7
1.1.2 Rule properties	8
1.1.3 Managing probabilities	9
1.2 Epidemiological modeling	10
1.2.1 SIR and SEIR model	10
1.2.2 Basic Reproduction Number	13
1.2.3 SIS and SIRS, SEIS and SEIRS model	14
1.2.4 Recent models	14
1.3 MVT summary	18
1.4 LOIMOS summary	19
2 Definition of the model	21
2.1 Resolution of conflicts	21
2.2 SEJIRS model	24
2.3 Behavior	26
2.3.1 Prudence parameter	27
2.3.2 Likelihood to change province	28
2.3.3 Caution factor	29
2.3.4 Vaccination will	29
3 Formalization of the model	31
3.1 The P system	31
3.2 Membranes	31
3.3 Objects	34
3.4 Rules	35
3.4.1 Infection rules	35
3.4.2 Contagion rules	36
3.4.3 Symptoms rules	37
3.4.4 Movement rules	38
3.4.5 Recovering rules	39
Recovering through hospitalization	40
Recovering through specialized antivirus	40

Recovering by zeroing the viral load	41
3.5 Routine and SPP	41
4 The simulator	44
4.1 Implementation	44
4.2 GUI and simulation	45
4.3 Scalability of the scenario	47
4.4 Generalization of infections	49
5 Results	51
5.1 Dataset	51
5.2 Tuning and Validation	52
5.2.1 Validation against Lombardy	57
5.2.2 Validation against PP	57
5.2.3 Validation against targeted quarantines	60
5.2.4 Validation against SPP	62
5.2.5 Validation of SEJIRS model	64
5.3 Computational complexity	65
6 Conclusions	70
6.1 Future developments	72
Bibliography	74

Introduction

In recent years, the study of infectious diseases and their transmission dynamics has gained increasing importance, particularly in light of the COVID-19 pandemic. The ability to simulate and predict the spread of diseases is crucial for public health planning, allowing policymakers to evaluate intervention strategies and optimize resource allocation. Computational models serve as powerful tools in this regard, enabling researchers to analyze the complex interactions that govern disease propagation within populations.

One such approach is LOIMOS, introduced in "P systems in the time of covid-19" [1], a Java-based simulator that leverages membrane computing to model the progression of infectious diseases. Membrane computing [2], [3], a branch of natural computing, provides a framework for representing biological and epidemiological processes through hierarchical compartmentalization. LOIMOS employs this paradigm to simulate the spread of infections across multiple levels, capturing interactions between individuals and environmental factors that influence transmission dynamics, achieving the results presented in [4] and [5].

A different approach to the same problem is presented in "A dynamic behavior epidemiological model by membrane systems" [6] from the universities of Milano-Bicocca, Verona, and Trieste. For this reason, it is abbreviated to MVT. MVT is a Python-based simulator designed to enhance computational efficiency and incorporate additional behavioral factors. MVT streamlines certain features of LOIMOS to improve performance while introducing new elements related to individual decision-making, such as vaccination choices, mobility patterns, and risk perception. These additions aim to create a more realistic representation of human responses during an epidemic, acknowledging that disease spread is influenced not only by biological mechanisms but also by social and behavioral dynamics.

While the first model possesses a focus on epidemiological rules, with an emphasis on an accurate representation of the dynamics of viral load growth, the second takes a different approach, based primarily on human behavior in response to infection levels and the containment policies that are put in place. The objective of this work is to integrate LOIMOS and MVT into a unified model that preserves the strengths of both starting models. By combining the hierarchical structure and detailed biological interactions of LOIMOS with the behavioral adaptability of MVT, the resulting model seeks to provide a more comprehensive and flexible simulation framework. This integration requires

smoothing out the formal and implementation differences between the two systems, merging the two architectures into a single model and validating it with simulations that confirm the positive results.

A key feature of the final model is its flexibility, making it adaptable to a wide range of infectious diseases and epidemiological scenarios. Instead of being tailored to a specific pathogen or outbreak, the framework is designed to accommodate different transmission mechanisms, intervention strategies, and population behaviors. This adaptability increases its usefulness as a research tool, enabling the simulation of disparate scenarios with minimal modifications.

Ultimately, this unified and generalized model aims to improve our understanding of how biological factors and human behavior interact during an epidemic. By providing a more versatile simulation platform, it can support decision-making in public health, helping to design targeted interventions and evaluate potential outcomes under diverse epidemiological conditions.

The thesis is organized into six chapters and each one is dedicated to a specific theme. Chapter 1 describes the current state of the art, both from the point of view of membrane computing and epidemiological modeling. The chapter ends with a summary of the two starting models, which is useful to understand the original work developed in this thesis. Chapter 2 describes the main conflicts that the two models present. For each of these, the solution chosen to resolve it is presented. The features concerning the behavior of individuals, present in MVT but not in LOIMOS, are also described, particularly how they are integrated into the model.

After resolving the conflicts present, we can move on to the formalization of the model, presented in Chapter 3. Every aspect that makes up the resulting P system is extensively described and documented. Much attention is focused on rewriting rules, which manage the entire model.

The implementation phase, described in Chapter 4, was carried out in collaboration with *Francesco Reiff* from the University of Verona, who contributed significantly to the coding and testing process. This collaboration fostered a valuable exchange of ideas that strengthened both the development and validation of the simulator. The chapter also presents the results achieved in adaptability, studied from a theoretical point of view but realized during implementation.

Chapter 5 presents the validation of the model. After introducing the comparison database and the parameters that have undergone a tuning phase, the simulator is subjected to numerous tests to verify its effectiveness. This chapter consolidates the features that theoretically enhance the work. The final chapter summarizes the obtained results, draws conclusions, and outlines potential directions for future developments.

Chapter 1

State of the art

1.1 Membrane systems

Membrane systems, also referred to as P systems, are a computational model inspired by the structural and functional properties of biological cells [7]–[9]. These systems provide a discrete representation of cellular processes, simulating chemical reactions and interactions, while abstracting from traditional chemical reaction mechanisms. Both the starting models in this study and the one to be obtained belong to a specific class of P systems: P systems with active membranes [10]. P systems with Active Membranes expand upon basic Membrane Systems by introducing electric polarization and specialized rules for membrane division. The formal definition is shown first, which is useful for later presenting the characteristics they possess. These systems are defined as:

$$\Pi = (V, H, \mu, w_1, w_2, \dots, w_m, R)$$

Where:

- V : Alphabet.
- H : Set of membrane labels.
- μ : Membrane structure.
- w_i : Initial multiset of chemicals in region i .
- R : Finite set of evolution rules.

These systems have gained significant attention in theoretical computer science and bioinformatics due to their ability to model complex biochemical networks. They offer an alternative computational paradigm that can be used for problem-solving in non-traditional ways. Key elements of these systems include a well-defined cellular structure, chemical substances, polarizations, and cellular reactions. Let's see these features in more detail.

- **Cellular Structure:** A typical membrane structure consists of compartments with external relationships. The Skin membrane, which is the outermost layer, encloses the internal membranes and separates the system from the external environment. Internal membranes may be elementary (not containing other membranes) or non-elementary (containing other membranes). In the second case, a nested membrane is thus created, which makes it possible to see a membrane system as possessing a tree structure. Such a hierarchical arrangement plays a crucial role in controlling information flow and regulating biochemical reactions within the system.
- **Chemical Substances:** Each region of the system contains chemicals, represented as multisets of symbols over an alphabet. Each symbol has a multiplicity representing its current quantity. The only limit on existence imposed on these symbols is that set by the capacity of the membrane in which they exist.
- **Polarization:** This feature is present in P systems with active membranes, but not in classical P systems. Each membrane and object may have a charge of +, - or 0. This polarization governs the communication behavior between membranes. There may be polarized rules that require the chemicals produced to move in membranes marked with a certain charge [11]. Polarization-dependent rules introduce an additional layer of complexity, allowing for selective permeability and directed transport of molecules.
- **Reactions:** Reactions are transitions executed in a membrane, and multiple transitions form a computation. Each reaction is described using a rewriting rule and a target destination. Chemicals on the left-hand side of a rule are transformed into those on the right, and there are 3 types of rules:
 - $v \rightarrow xy$ with $v, x, y \in V^*$ (non-cooperative, a symbol is turned into a multiset independently from the context);
 - $vw \rightarrow xy$ with $v, w, x, y \in V^*$ (cooperative, the reaction evaluates the objects to reach the goal more efficiently);
 - $vc \rightarrow xc$ with $v, c, x \in V^*$ (catalyst, in this case, the object c acts as a catalyst and triggers a reaction, turning v into x);

These are some examples of rewriting rules:

- $[v \rightarrow w]_h$ with $v, w \in V^*$ (evolution rules)
- $v[]_h \rightarrow [w]_h$ with $v, w \in V^*$ (in communication rules)
- $[v]_h \rightarrow w[]_h$ with $v, w \in V^*$ (out communication rules)
- $[v]_h \rightarrow [[w]_j]_h$ with $v, w \in V^*$ (membrane creation with object evolution)

– $[[v]_{h_1}]_{h_2} \square_{h_3} \rightarrow \square_{h_2} [[w]_{h_1}]_{h_3}$ with $v, w \in V^*$ (membrane movement)

The δ symbol represents membrane dissolution: the rewriting rule is applied to the objects, new ones are generated, and at the end, the membrane dissolves. The rewriting rules associated with this membrane cease to exist, and the contents are released into the membrane immediately outside. This mechanism allows the system to dynamically alter its topology.

A particular feature of P systems with active membranes is that they possess division rules. This feature made it possible to ‘solve’ complex problems, such as HPP and SAT [12], [13]. In these rules, membranes are split into two copies. If they possess chemicals or membranes, their contents are replicated.

– Elementary division:

$$[v]_h^\alpha \rightarrow [w]_{h_1}^\beta [y]_{h_2}^\gamma$$

– Non-elementary division:

$$\square_{h_1}^+ \cdots \square_{h_k}^+ \square_{h_{k+1}}^- \cdots \square_{h_n}^-]_h^\alpha \rightarrow [\square_{h_1}^+ \cdots \square_{h_k}^+]_{h_{n+1}}^\beta [\square_{h_{k+1}}^- \cdots \square_{h_n}^-]_{h_{n+2}}^\gamma$$

The ability to duplicate membranes enables massive parallelism, which is one of the reasons P systems are considered a promising approach for unconventional computing.

- **Computational step:** A model transition consists of a computational step, and during a step each chemical within the relevant membranes executes one of its possible rules. These rules are applied in a completely parallel and independent manner in each membrane. The system output is a set of symbols that comes out of the Skin membrane in a halting computation.

1.1.1 New properties for membrane rewriting rules

Now that the main characteristics of a P system have been described, we move on to the analysis of an interesting work that seeks to improve its capabilities to effectively employ such systems to model biological systems. We mainly refer to the work "Enhancing P systems for Complex Biological Simulations" by A.A. Elsayed, R. Ceprià, A. Hafez, C. Llorens, and J.M. Sempere [14].

Membrane systems have proven to be a powerful computational paradigm for modeling biological systems. The ability to model parallel and non-deterministic interactions makes P systems well-suited for simulating complex systems, such as immune responses to infections or vaccines. Recent advancements in natural computing and bioinformatics highlight the need for adaptive computational models capable of handling biological complexity. Traditional P systems often face limitations when applied to complex biological processes, such as immune system responses. The cited paper proposes an extension of P system rules to improve their representational fidelity in biological simulations, particularly in

modeling interactions within immune responses and nervous system dynamics. By extending P system rules, the cited work [14] aligns with efforts in computational biology to refine simulations of biochemical networks and cellular interactions. The paper in question identifies key limitations in traditional P system rules and introduces novel mechanisms to enhance their modeling capacity. Specifically, it introduces:

- New rules for managing empty membranes, an aspect often overlooked in classical models.
- Probabilistic parameters in reaction rules, allowing for more nuanced and realistic biological simulations.
- Visualization tools that aid in debugging and analyzing membrane structures and interactions, not exploited in this thesis.

Another key innovation in the cited work is the treatment of empty membranes, which in biological terms would either die or dissolve. In standard P systems, membranes dissolve when a specific object is present, but the authors introduce the symbol ϕ to represent the absence of objects. For example, an empty membrane X dissolving is represented as:

$$[\phi]_X \rightarrow \partial$$

where ∂ ensures a rule structure without carrying object meaning. Additionally, ϕ can be combined with other objects to impose an only if condition. For instance:

$$[\phi, v]_X \rightarrow \partial$$

dissolves membrane X if it contains only object v . This mechanism enhances the biological accuracy of P system simulations.

1.1.2 Rule properties

In a formal definition of P systems, a rule consists of an input and an output. In [14], the authors associate additional properties to each rule, such as annotations, probabilities, priority, and execution time. The general format is:

$$\text{input} \rightarrow \text{output} \quad (\text{prop_key} = \text{prop_value})$$

For example, in the rule $[(a)^2]_X \rightarrow [(a)^2, b]_X$ two a objects inside membrane X generate a new b object.

The paper cited at the beginning defines the following rule properties (prop_key):

- *name*: Distinguishes between similar rules with the same input but different outputs.

- *time*: Specifies the execution time in units (e.g., seconds, hours). Timed rules delay output production while consuming inputs. The time unit determines how system clocks trigger events.

Example: $[(a)^2]_X \rightarrow [(a)^2, b]_X \quad (\text{time} = 2h)$

- *probability*: Introduces stochastic behavior, making simulations more realistic. Even if input conditions are met, a rule may execute probabilistically rather than deterministically. This, by recognizing that biological processes do not always follow a simple and deterministic path, adds a level of realism to simulations.

The next section will deal in detail with the possibility of adding probabilities and priorities to the rules. This is a widely used aspect of the model described in this thesis and has an important impact in its formalization.

1.1.3 Managing probabilities

The probabilistic execution of rules in P systems is implemented by three components: Input Population Percentage (IPP), Rule Priority Index (RPI), and Rule Probability (RP).

- **Input Population Percentage (IPP)**: IPP controls the fraction of objects or membranes affected by a rule within a membrane, introducing probabilistic interactions instead of deterministic application. This enhances biological realism by simulating processes that occur only in a subset of the population. The rule r_1 below applies to 20% of the a objects in membrane X, causing their disappearance, while 80% remain unchanged:

$$[a]_X \rightarrow []_X \quad (\text{IPP} = 0.2, \text{name} = r_1)$$

- **Rule Priority Index (RPI)**: RPI resolves conflicts when multiple applicable rules exist within the same membrane. A lower RPI value indicates higher priority, ensuring sequential execution. This is critical when different object concentrations imply distinct biological states. The rule r_2 ($RPI = 1$), which allows a to grow in number, will take precedence over the rule r_3 ($RPI = 2$), which eliminates a instead:

$$[a, (b)^2]_X \rightarrow [(a)^2, (b)^2]_X \quad (RPI = 1, \text{name} = r_2)$$

$$[a, b]_X \rightarrow [b]_X \quad (RPI = 2, \text{name} = r_3)$$

- **Rule Probability (RP)**: RP assigns a probability to rule execution, adding stochastic behavior. If conditions are met, the rule executes with the specified probability. Rule r_4 executes 90% of the time when applicable. When triggered, it affects only 80% of the objects due to *IPP*:

$$[a]_X \rightarrow [a, b]_X \quad (\text{IPP} = 0.8, \text{RP} = 0.9, \text{name} = r_4)$$

RP and IPP act independently: RP determines execution probability, while IPP defines the proportion of affected objects.

1.2 Epidemiological modeling

Epidemiological models are essential tools for understanding the spread of infectious diseases and evaluating intervention strategies. Over the years, various models have been developed, each capturing different aspects of disease transmission dynamics. This chapter provides an overview of foundational models, highlighting their assumptions, limitations, and applicability to real-world scenarios.

1.2.1 SIR and SEIR model

In this thesis, a membrane system will be used to predict the course of infectious diseases. The modern history of models for the representation of infectious diseases takes shape with the SIR model, in 1927, by W. O. Kermack and A. G. McKendrick [15]. The acronym SIR identifies three categories of individuals: S stands for Susceptible, i.e. healthy individuals who may contract the disease; I stands for Infectious, i.e. individuals who have already contracted the infection, and R stands for Recovered, i.e. those who have recovered. In the SIR model it is assumed that infected individuals become immediately infectious (there is no incubation period) and that cured individuals acquire permanent immunity and can no longer take part in the transmission process.

The total population is divided into three classes, each represented by a function of time: $S(t)$, $I(t)$ e $R(t)$. The total number of individuals, denoted by N , is given by the sum of the three functions:

$$N = S(t) + I(t) + R(t)$$

When a Susceptible individual comes into contact with an Infectious individual, there is a probability that an infection will occur. Consequently, the susceptible population decreases by an amount equal to the number of infected individuals in a given time interval, while the infectious population increases by the same amount. The number of newly infected individuals per unit of time is defined as incidence.

The incidence makes it possible to describe the evolution of the susceptible population in the following way:

$$\frac{dS}{dt}(t) = S'(t) = -\text{incidence}$$

cN represents the total number of contacts the infectious individual has, where c is the contact rate *per capita* and N is the total number of individuals. The probability of a contact occurring with a susceptible individual is given by $\frac{S}{N}$. Consequently, the number of contacts between an infective and a susceptible in the unit of time is $cN \cdot \frac{S}{N} = cS$. However, not all contacts necessarily lead to infection, so the probability p that a contact will transmit the disease is introduced. We then have pcS as the number of susceptibles infected per unit time by a single infectious person. We define β , given by $\beta = pc$, as the constant

transmission rate. Consequently, the incidence (obtained as the total number of susceptibles contracting the infection in the unit of time) is given by βSI . Therefore, the evolution of the Susceptibles population can be described by the following differential equation:

$$S'(t) = -\beta SI$$

Infected individuals pass to the class I . We now denote by α the probability that an infected individual heals in the unit of time. The differential equation describing the dynamics of the infected is thus presented:

$$I'(t) = \beta SI - \alpha I$$

Healed individuals enter the class of the recovered with this differential equation:

$$R'(t) = \alpha I$$

To sum up, this is the system that represents the SIR model:

$$\begin{cases} S'(t) = -\beta SI \\ I'(t) = \beta SI - \alpha I \\ R'(t) = \alpha I \end{cases}$$

A representation of this can be seen in Fig.1.1.

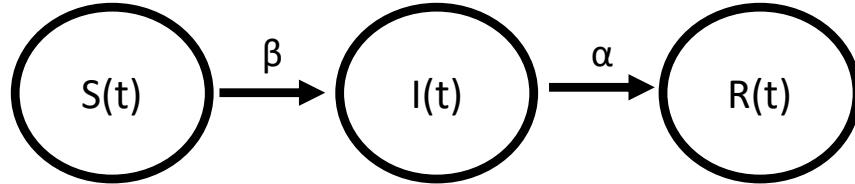


Figure 1.1: SIR model.

Demography management can be added to the model just presented. A SIR model with demographics [16] makes it possible to add and remove individuals following the birth and death rates of the population. In addition, it is possible to assign a different death rate according to category to simulate infections (such as COVID-19) that can lead to death.

The SIR model with demography is a basic formulation that does not lend itself adequately to the study of the spread of many infectious diseases due to their complexity. For this reason, a more realistic model than the SIR model is analyzed in this section: the SEIR model [17].

In many infectious diseases, an individual does not immediately become infectious after being infected. In these cases, the pathogen needs an interval

of time to replicate and establish itself within the new host. This interval is referred to as the latent period and is the time during which an individual is infected but not yet infectious.

Another fundamental concept in epidemiology is the incubation period, which is the time interval between infection and the appearance of symptoms. It is important to note that the latent and incubation periods may not coincide, as they refer to distinct processes. In our model it was chosen, for simplicity's sake, to make them coincide.

In order to include the latent period in the SIR model, we introduce a new class, called Exposed, denoted by the letter E . This class represents individuals who have been infected but are not yet infectious and is generally placed after the susceptible class. The SEIR model can be described by the following system of differential equations:

$$\begin{cases} S'(t) = -\beta SI \\ E'(t) = \beta SI - \eta E \\ I'(t) = \eta E - \alpha I \\ R'(t) = \alpha I \end{cases}$$

where:

- η represents the rate per person at which an individual becomes infectious.
- The variable $I(t)$ refers exclusively to infectious individuals.
- The other variables and parameters (S, R, β, α) retain the same meaning as in the SIR model.

The SEIR model thus allows a more realistic description of the spread of infectious diseases characterized by a latent period, offering a significant extension compared to the SIR model. A representation of this can be seen in Fig.1.2. As mentioned for the SIR model, there is also an implementation with demography for the SEIR model. The birth rate Λ and the death rate per person μ , both constant, are added. The SEIR model with demographics will then look like this:

$$\begin{cases} S'(t) = \Lambda - (\beta I + \mu)S \\ E'(t) = \beta SI - (\eta + \mu)E \\ I'(t) = \eta E - (\alpha + \mu)I \\ R'(t) = \alpha I - \mu R \end{cases}$$

In this system, it can easily be seen that the total population is no longer constant, but tends to a maximum value calculated as:

$$N(t) \rightarrow \frac{\Lambda}{\mu} \quad \text{when } t \rightarrow \infty$$

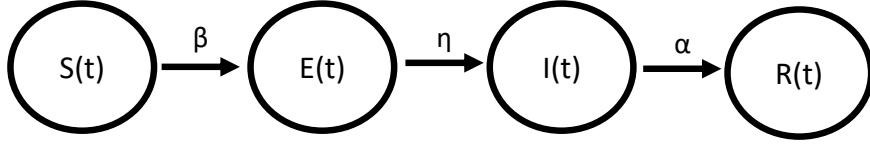


Figure 1.2: SEIR model.

1.2.2 Basic Reproduction Number

It is now possible to deal with a very important parameter [18] in these models: the basic reproduction number, abbreviated to BRN. This parameter, denoted \mathcal{R}_0 , is given by the expression:

$$\mathcal{R}_0 = \frac{\beta\Lambda}{\mu(\alpha + \mu)}$$

From an epidemiological point of view, \mathcal{R}_0 represents the average number of secondary cases generated by a single infectious individual in a population initially composed entirely of susceptibles. Let us analyze this in detail:

1. Upon the arrival of a patient zero in a fully susceptible population, the number of susceptible individuals will be equal to the total population calculated above, expressed as $\frac{\Lambda}{\mu}$.
2. An individual leaves the infectious state with a rate equal to $\alpha + \mu$, so $\frac{1}{\alpha+\mu}$ represents the average time during which it remains infectious.
3. The incidence of new infections is given by βSI . If it is assumed that there is only one infectious person in the population ($I = 1$) and that all others are susceptible ($S = \frac{\Lambda}{\mu}$), then the number of new infections per unit time is equal to $\beta\frac{\Lambda}{\mu}$.
4. Consequently, the total number of infections caused by a single infectious person during the period in which he remains infectious will be given by:

$$\mathcal{R}_0 = \frac{\beta\Lambda}{\mu(\alpha + \mu)}.$$

The value of \mathcal{R}_0 is crucial [19] as it represents a threshold for the dynamics of the system, providing essential information on the propagation of the disease:

- If $\mathcal{R}_0 > 1$, the disease becomes endemic, i.e. it stabilizes and remains in the population indefinitely.
- If, on the other hand, $\mathcal{R}_0 < 1$, the number of infected gradually decreases until it disappears, stopping the spread of the infection.

1.2.3 SIS and SIRS, SEIS and SEIRS model

Although the SEIR model adds more detail than the SIR, it is still far from modeling a plausible reality. An important feature in many infectious diseases is that infection-acquired immunity is not lifelong, and recovered subjects will eventually become susceptible again. In the two models just presented, this was impossible, as one can only leave the R class due to death. To remedy this shortcoming, the SIS and SIRS [20] models have been introduced. As can be guessed from the name, in the former the recovered class is completely removed, and individuals who have overcome the disease will return to the susceptible class. In this model, therefore, there is no acquired immunity. In the SIRS model, on the other hand, individuals who recover are temporarily placed in the recovered class. Here, after a period of time determined in the creation of the model, individuals lose this immunity and return to the susceptible class.

The two features presented, i.e. the presence of the E class and the possibility of returning susceptible, are both present in the SEIS [21], [22] and SEIRS [23] models, which can be seen as a fusion of those described so far. The scheme of the four models is shown in Fig.1.3 and Fig.1.4, while in Table1.1 there is a recap of the models presented and some examples of diseases with which they can be associated.

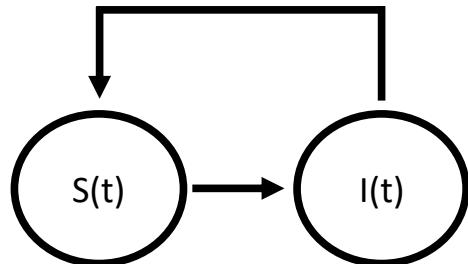
Model	Key Feature	Examples of Diseases
SIR	Permanent immunity	Measles, chickenpox
SEIR	Permanent immunity, latent period	COVID-19, Ebola
SIS	No immunity	Common cold, urinary tract infections
SIRS	Temporary immunity	Influenza, COVID-19
SEIS	No immunity, latent period	Tuberculosis
SEIRS	Temporary immunity, latent period	Dengue fever, some strains of influenza

Table 1.1: Comparison of epidemiological models.

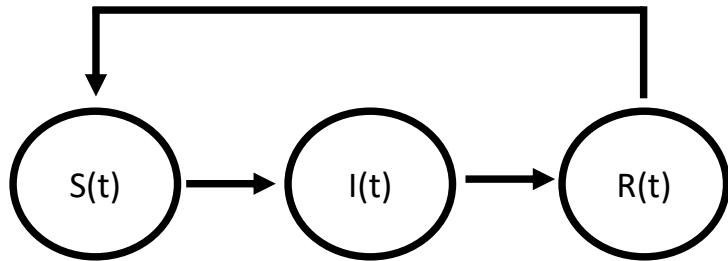
1.2.4 Recent models

The next two works described are models created ad-hoc to simulate scenarios of particular interest for this thesis, i.e. relating to SARS and COVID-19. Although this can also be done with simpler models, as is done in the work [24] with SEIS, it is interesting to observe how more complex models can represent this problem. It will certainly come in handy for the study and creation phase of our model.

The first work presented is the study by A. B. Gumel et al. on a mathematical model of the 2003 SARS outbreak [25]. The study underscores the importance of quarantine and isolation in controlling the epidemic in the absence of treatments or vaccines. SARS (Severe Acute Respiratory Syndrome), caused by the SARS-CoV coronavirus, first appeared in Guangdong in November 2002 and ended in July 2003, with 8098 cases and 774 deaths across 32 countries. Its mortality rate averaged 15%, rising to 50% among the elderly. The WHO declared



(a)



(b)

Figure 1.3: SIS (a) and SIRS (b) model

a global emergency in March 2003, implementing quarantine and isolation as key strategies. Despite initial setbacks due to poor protocols, these measures were eventually refined, allowing for the successful containment and elimination of SARS by August 2003.

The model examines the dynamics of six classes: susceptibles S , asymptomatic individuals E , quarantined individuals Q , symptomatic individuals I , isolated individuals J , and recovered individuals R . In this context, quarantine refers to the separation of individuals before symptoms appear, while isolation applies only to symptomatic individuals.

The model also takes demographics into account: let $\mu > 0$ be the natural death rate common for each class; let Π be the rate per unit of time at which new susceptibles enter the region. This parameter takes into account births, immigration, and emigration; let p be the rate per day at which tourists, supposedly infected but asymptomatic, enter the region. Note that this parameter p is set equal to zero in the theoretical formulation of the model because the estimated value is small and has been seen to have a negligible effect, whereas

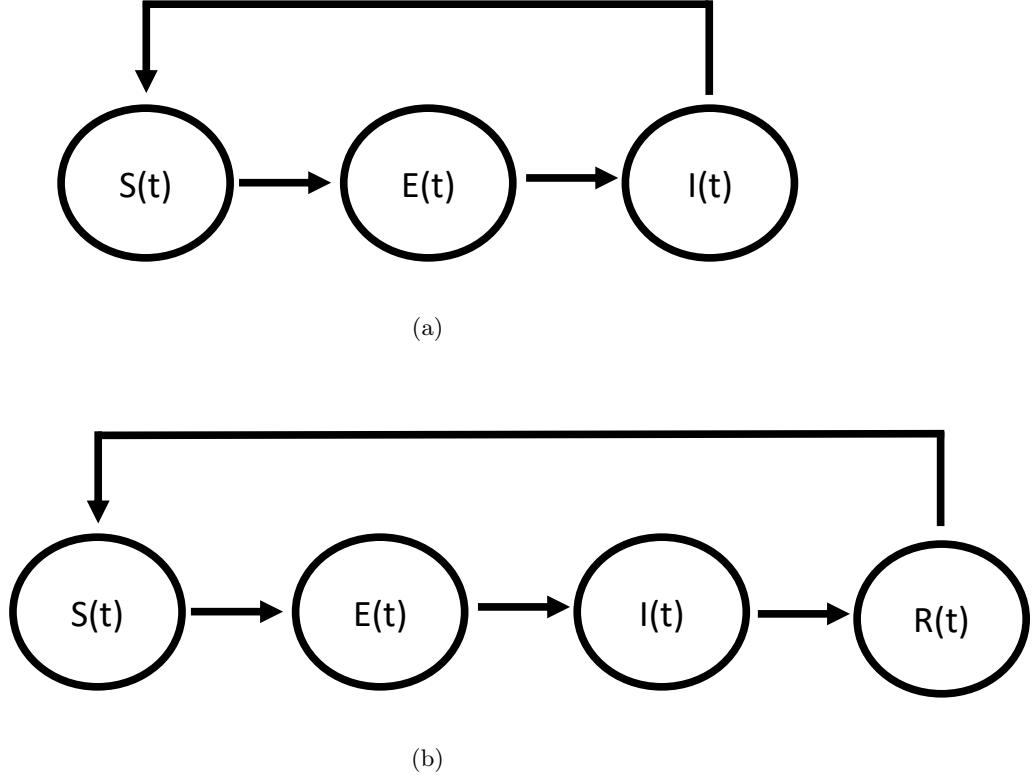


Figure 1.4: SEIS (a) and SEIRS (b) model

the numerical simulations were performed with $p > 0$. In addition, there are two classes, I and J , which have death rates increased by d_1 and d_2 , respectively. These are the disease-induced death rates, which are different for the two classes. A scheme of the model, called SEIRQJ, is shown in Fig.1.5. For simplicity, all rates that characterized the movement from one class to another have been hidden. The impact this model had in the creation of the work in this thesis will be presented later.

The second advanced model presented in this paper is the most recent of all and is called SIDARTHE [26]–[29]. The SIDARTHE model is an epidemiological framework developed to analyze the spread of COVID-19 in Italy, and can be seen in Fig.1.6. The acronym SIDARTHE stands for susceptible S , infected I , diagnosed D , ailing A , recognized R , threatened T , healed H and extinct E , representing different stages of the disease progression. This model distinguishes between diagnosed and undiagnosed cases, as well as varying symptom severity, to provide a comprehensive understanding of the epidemic’s dynamics. By differentiating between diagnosed and undiagnosed individuals, the SIDARTHE

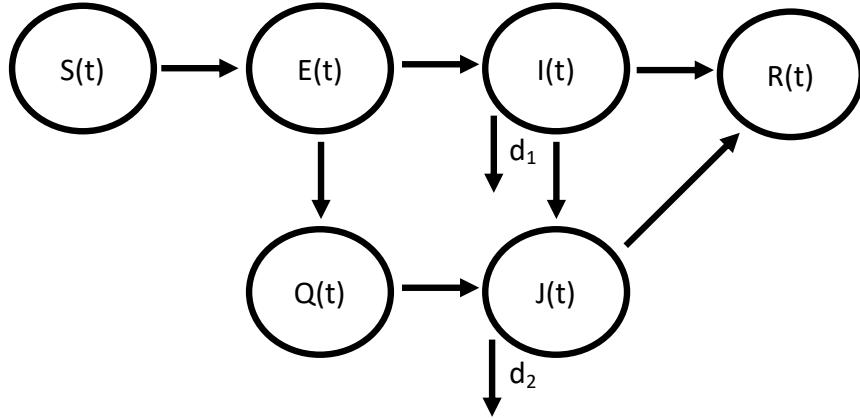


Figure 1.5: SEIRQJ model.

model accounts for the higher transmission potential of undiagnosed cases, who are less likely to be isolated. This distinction helps explain discrepancies in perceived case fatality rates and the overall severity of the epidemic. The model also emphasizes the importance of implementing combined interventions, such as social distancing, widespread testing, and contact tracing, to effectively control the spread of the virus. Simulation results from the SIDARTHE model have been compared with real data from the COVID-19 epidemic in Italy, demonstrating its validity and usefulness in predicting various scenarios based on different countermeasures. The model's insights have been instrumental in understanding the impact of public health interventions and guiding policy decisions during the pandemic. The SIDARTHE model is very complex and will not be analyzed in detail in this section. It contains many interesting ideas that have been extrapolated and incorporated into the SEJIRS model, presented in section 2.2. The SEJIRS model was designed in this thesis to represent the P system obtained by merging the two initial models.

The final model from which significant insights were drawn is the SEITR model, specifically the one described in "Qualitative analysis of a stochastic SEITR epidemic model with multiple stages of infection and treatment" [30]. This work introduces the class T_j , representing individuals undergoing treatment. Its distinctive feature lies in the presence of multiple stages of infection, each with different effects on individuals, denoted by the index j . For each infection stage j , there is a corresponding class of infected individuals I_j and a class of treated individuals T_j . Although this structure increases complexity, it closely aligns with how the model presented in this thesis handles symptom progression and isolation measures.

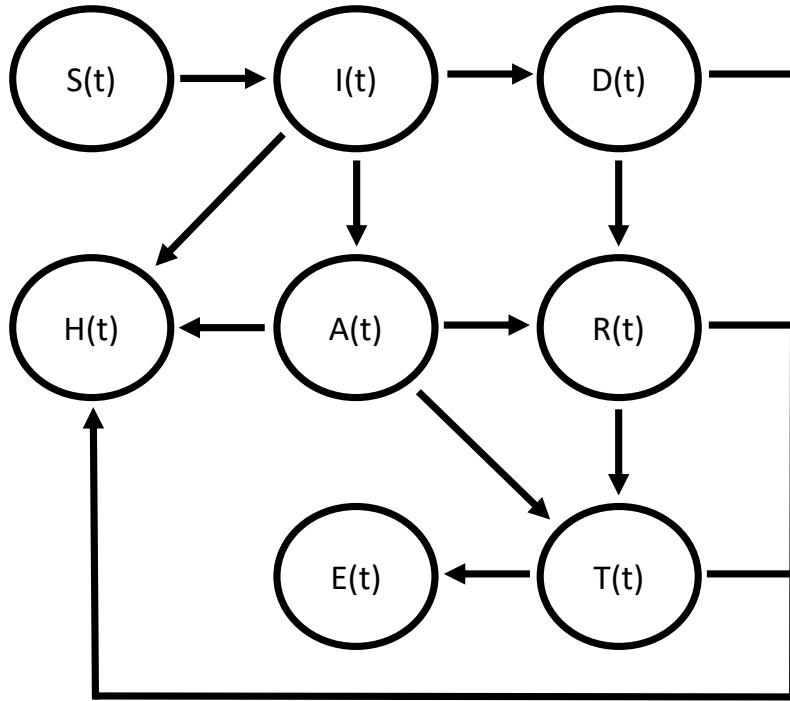


Figure 1.6: SIDARTHE model.

1.3 MVT summary

The first work presented in detail is "A dynamic behavior epidemiological model by membrane systems" [6], called MVT. MVT is the starting work, both in terms of theoretical model and code, written in Python. Some of its characteristics are now summarized.

- **Membranes:** For each province, the model has only four types of buildings: schools, workplaces, hospitals, and common areas, each with its own label. Each individual membrane has a province of origin and a province of destination, identification number, age divided into 3 age groups (6-19, 20-59, 60+), and other personal information.
- **Infection Rules:** The probability of getting infected is based on some parameters like the infection rate of the location, the number of infected present, and the awareness about the disease. A new contagion leads to an incubation period (Inc) of 5 days, then one of infection (I) for 7 days, and finally immunity (Imm) for 180 days. This disease management cycle

is determined by simple rules of cooperation. Infected individuals can be taken to a hospital where they begin a course of hard treatment that will lead to recovery. In the hospital, however, every hour they have a determined probability of dying.

- **Simulation:** The simulation is carried out for one year with a population of 25000 and vaccination rates at 0%, 20%, 40%, 60%, 80%. The Caution Parameter f^* , representing the percentage of infected people necessary for individuals to adopt behaviors that halve infection probabilities, is changed. The impact of varying these parameters on the simulation is presented.
- **Relevant features:** Behavior and provinces. The population adopts behaviors in response to changes in the epidemiological situation. These include adaptive behaviors based on infection prevalence, dynamically influencing infection rates, and vaccination processes.

The model is divided into a fixed number of provinces that share the population equally. Individuals move between these regions with varying probabilities depending on the number of infected people in the destination region, consolidating the behavioral aspect of the population.

1.4 LOIMOS summary

The second work to be integrated into this model is "P systems in the time of COVID-19" [1], where LOIMOS is presented. LOIMOS has a predictive statistical approach and uses a cell-like P system with active membranes without polarization. Some of its characteristics are now summarized.

1. **Membranes:** LOIMOS has twelve types of place membranes: house, ICU (Intensive Care Unit), hospital, hospital aux (movement from hospital to ICU), post-hospital, residence, elderly day center, workplace, common area, school divided into primary and secondary and leisure area. People are defined by time, day, 4 age ranges (0-12, 13-19, 20-59, 60+), role, viral load (v_1 , *antiv* and *antivesp*), symptoms (E₁, E₂, E₃ or E₄), address.
2. **Infection Rules:** There are over 30000 rules and a table with infection percentages for different ages and places. An infected person can send an object of type virus to another. They can recover or worsen depending on the type of infection, and when the infection ends the individual becomes immune with four different types of immunity. The most serious infection type, which is integrated into the final model, has an amount x of v_1 objects, with $0 \leq x \leq 1000$. The higher this value is, the more infected the individual is. The patient has antivirus objects (*antiv*) that can become specialized (*antivesp*), which eliminate the viral load (v_1) and can lead to recovery.
3. **Simulation:** Three 290-day simulations are run, with a population of around 10000 individuals. In the first simulation everything remains

open and there are no movement restrictions, in the second there is a quarantine, and in the third there are quarantine and restriction. This led to an interesting comparison of results, where the population is divided according to the type of infection.

4. **Relevant features:** LOIMOS has interesting features concerning symptom management and type of infection. Objects E_1 , E_2 , E_3 , E_4 are used in individuals to denote the level of symptoms, which result in differences in the way the population moves.

- E_2 : stay at home.
- E_3 : go to the hospital, may die.
- E_4 : go to the ICU, may die.

With a high viral load, thus a high v_1 count, an individual might worsen his symptoms and go from E_i to E_{i+1} . If this is not the case, the patient can remain in the same E_i with a certain probability or he can heal and return to E_1 , always based on the level of viral load. Different levels of symptoms lead to different probabilities of death.

In LOIMOS there are four types of infection, each characterized by 3 parameters:

- Innate immunity: can be efficient or inefficient
- Acquired immunity: can be lacking, normal or weak
- Symptoms: can be mild to no, or symptomatic

And the possible combinations that create LOIMOS infections are:

- Efficient innate immunity, lacking acquired immunity, mild to no symptoms
- Efficient innate immunity, normal acquired immunity, mild to no symptoms
- Inefficient innate immunity, normal acquired immunity, symptomatic
- Inefficient innate immunity, weak acquired immunity, symptomatic

In LOIMOS, there are also four types of immunity, each with its characteristics and consequences for the individual.

Chapter 2

Definition of the model

This chapter presents the preliminary work carried out to define the new model. After describing the characteristics of LOIMOS and MVT, it was necessary to iron out the conflicts present in order to be able to proceed with the formalization of the new model. After this step, it was possible to create the epidemiological model for the work presented in this thesis. This was done by taking inspiration from the models described in the previous chapter. Describing and visualizing this model gives a clear idea of the life cycle that the disease will have in the resulting work. Afterward, the main characteristics of this model with regard to the behavior of the population, which are of great impact on the thesis work carried out, are described.

2.1 Resolution of conflicts

We discuss now the main conflicts encountered between LOIMOS and MVT, and the decisions taken to resolve them. When not specified otherwise in Table 2.1, MVT is considered, since it is the starting model.

1. **Programming language:** The programming language used was the first problem to be addressed. LOIMOS uses a specific approach for membrane management and rule application in Java. Adding a new rule requires not insignificant writing work. MVT, on the other hand, uses object-oriented programming in Python. We opted to use Python, as the code lent itself better to the addition of new features, being simpler.
2. **Place membranes:** The number of place membrane types in LOIMOS (12) is higher than in MVT (4). These increased the computational weight, but added detail and realism to the simulation. It was therefore decided to add some of the place membranes present in LOIMOS, testing their effectiveness on a case-by-case basis. Houses, for example, were completely absent from the MVT model, and individuals spent the night in the common area. Nonetheless, not all place membranes originally present in LOIMOS

Conflicts	Taken from
Programming language	MVT
Place membranes	LOIMOS
Infection and death probability	New
Behavior	MVT
Infection Management	Both
Symptoms	LOIMOS
Movement and Quarantine	MVT
Weekend	LOIMOS
Immunity	MVT
Hospitalization	Both
Output	MVT
Recovery	Both

Table 2.1: Summary of conflict resolution

were added, to maintain computing efficiency. Those considered in the end are the following: houses, hospitals, ICUs, workplaces, schools, leisure centers, and the common area.

3. **Infection and death probability:** The starting models have different probabilities of infecting individuals and death rates. This is also because the probability of infection varies according to the membranes of the location, and the probability of death varies according to the healthcare facility in which one is located. As mentioned above, the place membranes are not the same for the two models, nor for the model in this thesis, thus leading to different values. The infection and death rates of LOIMOS and MVT, and the discussion about how these were chosen and tuned to obtain the final ones for the present model, can be found in section 5.2, dedicated to parameter tuning.
4. **Social Behavior:** MVT models people's behavior was chosen, in particular for three cases: inter-provincial displacement, willingness to vaccinate, and caution factor in becoming infected. This is a key feature of the MVT model that has been fully maintained, controlled, and expanded with the Prudence Parameter.
5. **Infection Management:** MVT adds a suffix to each individual based on health status: *Iinc* (Initial incubation), *I* (Infection), and *Imm* (Immunity). It is a simple method since the rules are cyclic, do not include probabilities, and each day the next step is applied. LOIMOS uses objects like viruses,

harmless infections, antivirus, and antibodies in quantities from 0 to 1000. These interact with each other through cooperative rules with fixed probabilities. They also influence symptom management, which is not present in MVT. In LOIMOS, infection management is handled with a high level of detail, representing one of its key strengths. For this reason, it has been incorporated into the new model.

6. **Symptoms:** MVT ignores the symptomatic aspect of the disease. In LOIMOS, there are four levels of symptoms (E_1 , E_2 , E_3 , E_4) with fixed probability values for transitioning from one to the next. According to these symptoms, people are brought to hospitals and ICUs, and individuals decide to stay at home for preventive isolation. Symptoms play a crucial role in LOIMOS and are closely linked to infection management. Ignoring them would result in a significant shortcoming in the thesis work, as it would lack a key parameter for modeling many infectious diseases. For this reason, they are integrated into the model.
7. **Movement and Quarantine:** Since LOIMOS is not divided into provinces, only in MVT individuals can move from one province membrane to another. In LOIMOS, movement between membranes is still present, but only between different places. Nevertheless, in MVT, individuals' common sense is the only possibility to prevent infections between provinces. There are no quarantine rules or movement restrictions imposed by strict regulations as governments might implement. Only in LOIMOS there is the possibility of applying quarantines or curfews. It was decided to combine the two models by keeping the provinces and by adding the option to restrict travel outside one's own house.
8. **Weekend:** A small note is made regarding the management of free time. In LOIMOS, each individual is given options for spending the weekend, during which they will not follow their usual routine and will go to places they don't usually visit (e.g., a teenager visiting an elderly relative in a nursing home). MVT mainly ignores this, with far fewer places where individuals can move. With the addition of new locations, the proposed model in this thesis was able to implement weekend behaviors, with different routines depending on the age group.
9. **Immunity:** In MVT, once an individual is considered cured, he or she acquires immunity and can no longer be infected for 180 days. In LOIMOS, the disease is treated more elaborately, with immunity acquired in different ways, meaning that the individual could become infected again. However, this requires the 4 different types of infection present in LOIMOS, which adds too much complexity to the model, so the MVT implementation is retained.
10. **Hospitalization:** In MVT there is a probability of being moved to a hospital every hour, and this has three effects: it prohibits the individual from going outside thus preventing new infections, halves the probability

of death, and most importantly, it starts a course of treatment that leads to cure after 7 days. In LOIMOS one goes to the hospital when reaches symptoms of E_3 , and ICU on E_4 symptoms. It was decided to maintain a mix between the two models, for better accuracy. Symptoms are the reason why individuals move to hospitals, but the three functions they have in MVT are maintained. ICUs have been added and they work in a similar way like LOIMOS, continuing the cycle of care that began in the hospital.

11. **Output:** Whereas LOIMOS has an output membrane and generates a large CSV file containing dozens of columns, MVT only outputs four values per day: currently infected, new daily cases, prevalence, and deaths. The simplicity of the MVT output is chosen with the addition of seconds for time complexity, information about the SEJIRS model, and the generation of the graphs presented later.
12. **Recovery:** MVT allows individuals to recover after a course of treatment in a hospital. LOIMOS gives this option upon reaching a certain number of *antivesp* objects, or if the disease declines because it is defeated by the developed antibodies. Both approaches have been retained and joined together.

2.2 SEJIRS model

The works presented in section 1.2 enabled the creation of a similar scheme, adapted to the fusion of LOIMOS and MVT models. After describing how the conflicts concerning the functioning of disease, symptoms, recovery, and immunity were resolved, a modeling in this regard consolidates the result obtained. Although done in a bland manner, such a schematization gives more validity to the resulting model, making it comparable to similar works. For this reason, an effort was made to represent the epidemiological model of the membrane system created in this work. The result can be seen in Fig.2.1. It is abbreviated as SEJIRS and, as it was mainly inspired by the last two models presented, i.e. the one for SARS in 2003 (with letters SEIRQJ) and SIDARTHE, a glossary of each class and variable is necessary in order to understand and distinguish them.

- $S(t)$: As in all the models described so far, it represents the susceptible class. These individuals, characterized by a ‘Healthy’ status in the code, can become infected and enter the incubation state.
- $E(t)$: As in the exposed class, this denotes individuals who have contracted the disease, but who are not yet infectious and do not present symptoms, so they are not in isolation. An individual in this class can either recover, if their antibodies are effective and defeat the disease while it is incubating, or worsen and become infectious. In our simulator, therefore, the latent period is equivalent to the incubation period. Such individuals in this

model have an ‘Incubation’ status and can be associated with members of class E of the models presented above.

- $I(t)$: this class represents people who are infected and can infect others. Individuals in this class have the status of ‘Infected’, and its functioning is comparable to that of the class of the same name in the initial models. This class incorporates some classes from SIDARTHE, as will be explained below. In the model of this thesis as in SIDARTHE there are symptoms, and infected individuals can be asymptomatic. In that case, an asymptomatic individual belonging to I class can be compared to one belonging to class I of SIDARTHE. If, on the other hand, mild symptoms develop in the individual, he may not be able to recognize them. He will therefore be associated with class A of SIDARTHE. If he realizes he is infected, there is a probability (dictated by the PP described in section 2.2.1) that he will become an isolated individual. Only in this case, this class can be associated with the J of Gumel’s model and the R of SIDARTHE. An individual belonging to the infected I class has only two possible fates: to recover or to suffer a worse of symptoms.
- $J_3(t)$: When an individual presents symptoms of a medium level, they enter this class and are labeled as being in isolation. The number 3 denotes the medium level of symptoms (E_3), described later. In this class individuals are infectious but, if enough rooms are available, they are kept in isolation at the hospital. Even if the hospitals are occupied, an individual in this condition does not have the possibility of leaving home. If they cannot recover, they may die or suffer a further deterioration of their health condition. The parameter d_1 is, as in the model for SARS [25], the probability that an individual in this class will die. It can therefore be associated with the J class of that model or the T class of SIDARTHE.
- $J_4(t)$: Individuals in this class present severe symptoms, such as respiratory problems, and require intensive care. They will try to be taken to an available Intensive Care Unit (ICU), and have a parameter of death d_2 , with $d_2 > d_1$. They are obviously to be considered as isolated individuals. As with the previous class, they can be associated with members of class J in the first advanced model or with T in the second. One can see J_3 and J_4 as a division of these classes, dictated by the different parameters of death and level of symptoms. Such a division is made in [30], where infections are categorized into stages. An individual in this class is at the worst stage of the infection, so will only recover or die.
- $R(t)$: As with all previous models (except for SIDARTHE), this class represents the recovered, denoted by the status ‘Recovered’. Being inspired by a model with temporary immunity, there is a period after which an individual in this class will return to the S class. Since there is no possibility of death in this class, the only possible transition for its members is to return to the susceptible state (S), leading to a rapid increase in its

population. It is intuitive to understand that, if high immunity times are chosen, the model will only have one peak infection and then stabilize.

- Quarantine: The quarantine class, which includes all quarantined individuals, is not explicitly incorporated into the model as a distinct state. This is because entering quarantine is not inherently part of the disease progression or treatment cycle. Instead, quarantine is treated as an external condition that can be applied to all individuals, regardless of their health status. During a simulation, a predefined quarantine period can be set, during which all individuals remain confined to their homes while retaining their SEJIRS model state. Unlike SEIRQJ, where quarantine is modeled as a specific state called Q , here it is conceptually present as an optional mechanism rather than an intrinsic part of the model.
- Extinct: The class E from the SIDARTHE model, which includes all deceased individuals, is not explicitly included in the scheme. Instead, death-related data is recorded for output purposes without the need to introduce a dedicated class.
- Demography: The demography, mentioned in section 1.2.1, is not included in the model. It would introduce birth and death rates for individuals, regardless of their health status. This is because the meaningful simulations will only last for one year, a short time to make sense of the addition of a complicated feature. Furthermore, the number of deaths per COVID-19 is the only variable related to the deceased that is of interest in the simulation. Indeed, none of the parameters μ, Λ, Π, p concerning non-disease-related deaths, births, immigration and emigration, and tourists are present.

In the results chapter, in particular section 5.2.5, we describe in detail how the cardinality of the classes changes during a simulation.

2.3 Behavior

A model that simulates reality, to be effective and reliable, should model every aspect of it. An important and therefore not negligible factor in this type of model is the behavior of individuals [31], [32]. Each individual thinks in his or her own way and makes decisions that impact the population according to his or her criteria, and these are so complex as to be practically impossible to model perfectly. However, it remains important and interesting to add parameters that allow individuals in the simulation to choose for themselves, eliminating the risk of a population behaving as a mass that reacts to stimuli from its environment in the same way. To avoid this problem, the behavior of individuals is exploited to allow them to make decisions. In this work, there are four areas where this feature can be found.

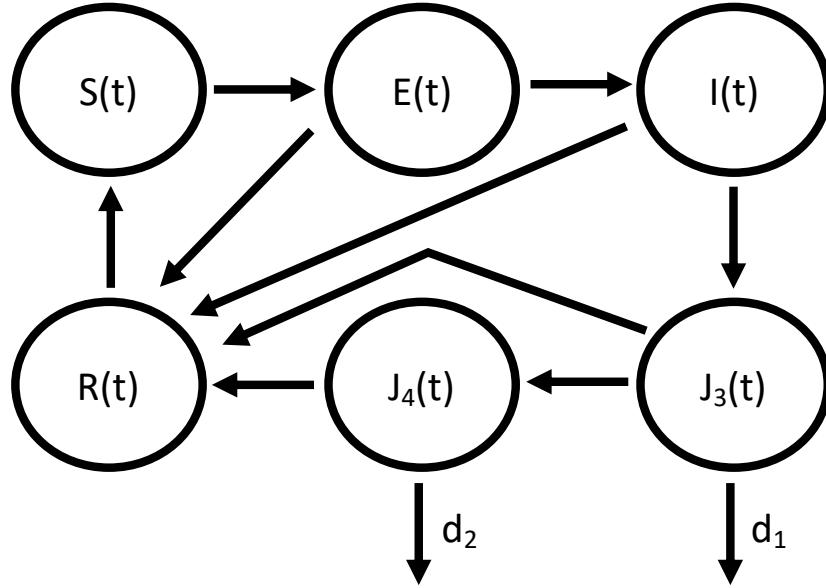


Figure 2.1: SEJIRS model.

2.3.1 Prudence parameter

As will be presented later, infected individuals have a symptoms parameter that describes their level, and individuals with symptoms are aware of their situation. If they have a low level of symptoms (E_2) they can isolate themselves, or they may ignore being infected for various reasons like personal urgency, underestimation of symptoms, lack of interest in the pandemic, or more [33]. This may lead the individual to go out anyway but with a lower probability. To model this probability, the Prudence Parameter (PP) is introduced, with $0 \leq PP \leq 1$. As PP increases, individuals become more cautious, reducing the probability that they leave the house while having symptoms.

- With $PP = 0$, we model a population that is totally unaware of the disease and the probability of coming out with low symptoms (E_2) remains the same as without symptoms (E_1).
- With an intermediate PP value, e.g. $PP = 0.3$, you have a population that is about 50% less likely to go out with weak symptoms. This parameter is applied in a very trivial way. In the rule that allows an individual to perform his or her routine, the probability of performing it is reduced by $(1 - PP)^2$. An example is given in equation 2.1.
- With $PP = 1$, the population is fully aware of the risks and symptoms of

the disease, and the probability of going out infected is zero. The effect is the same as in the LOIMOS modeling, with the obligation to stay at home with E_2 . This does not eliminate the infection, but it certainly has a beneficial impact on disease containment.

$$[ind_{E_2}]_{House} \xrightarrow{\text{Movement Probability} * (1 - PP)^2} [ind_{E_2}]_{School} \quad (2.1)$$

Note that this rule only applies to individuals with symptoms in E_2 . It does not reflect the reality that individuals with strong symptoms (E_3 and E_4) continue to go to work or school and even change regions.

This parameter can be employed to model populations inclined to disregard regulations imposed by governmental authorities. Despite the existence of explicit guidelines designed to mitigate the spread of the pandemic, such as the obligation to undergo diagnostic testing when exhibiting clear symptoms and the subsequent requirement to self-isolate; populations characterized by a limited degree of civic awareness may be prone to non-compliance. This tendency has been extensively documented in [34] and [35] concerning the Italian context. Consequently, it may be justifiable to assign a medium or low value to PP , even when the disease is well understood and thoroughly investigated, if the population under analysis may not adhere to prescribed directives.

Another important advantage this parameter brings will be described in section 4.4, dedicated to generalization. The effects this parameter has on the population and the epidemic trend will be studied in section 5.2.2, dedicated to results and parameter validation.

2.3.2 Likelihood to change province

The model is divided into provinces and these play an important role in the routines of individuals. Each individual has a province of origin, which is where he or she resides, and a province of destination, which is where he or she will perform his or her weekly routines. In a large part of the population, i.e. 80 percent, these two provinces coincide. The individual in question will therefore perform his duties without changing provinces. In the remaining cases, the provinces are assigned randomly. This generates a proportion of individuals who must cross a province in order to perform their routine. In order to do this, they will have to spend an hour in the common area of the destination province, which is shared by all the individuals making this move. This operation carries a high risk of being infected and individuals are aware of this. For this reason, the willingness to move between provinces was modeled.

The probability of changing province is defined as $\left(1 - \frac{\phi_Y}{\text{total population}_Y}\right)$, where ϕ_Y represents the number of infected individuals in province Y. This means that the willingness to move towards an arrival point is defined by the epidemiological context of the destination province. The higher the percentage of infected people at the destination, the less likely it is that individuals will travel to that province to perform their routine.

2.3.3 Caution factor

This feature has been retained by the MVT model because it is workable and plausible with reality, so more information about it can be found in [6]. As the count of infectious cases grows, people tend to adopt more cautious behaviors to reduce the risk of infection. This trend can be represented through a non-negative, decreasing function, shown in Eq.2.2, with the number of infections acting as the variable. Let M denote the information regarding infection cases, N represents the total population and $a > 0$:

$$\psi(M) = \frac{1}{1 + a \frac{M}{N}} \quad (2.2)$$

If it is replaced $f = \frac{M}{N}$ and $a = \frac{1}{f^*}$, where $\frac{1}{f^*}$ is the fraction that halves the risk of contagion and f^* is very small, Eq.2.3 is obtained:

$$\psi(f) = \frac{1}{1 + \frac{f}{f^*}} \quad (2.3)$$

The function describes a situation where, when the values of f and f^* coincide, the result returned is $\frac{1}{2}$. This implies that the value of f^* represents the proportion of infected individuals required for the adopted behavior to reduce the probability of infection transmission by half. It can now be described in more detail how the behavior is modeled:

- If $f^* = 1$, the probability of infection is only halved when $f = 1$, which means that the entire population is infected;
- If $f^* = 0.1$, the probability of infection will be reduced by half when one tenth of the population in question is infected. For values of $f > 0.1$, the probability of infection will be further decreased.

This places particular importance on the parameter f^* , and consequently on the value a . This value, called simply the Caution Factor, is set to $a = 0.001$ during all the simulations. Its effects on the population are not tested and it is not considered in the validation phase as it is a parameter that has already been consolidated in the previous model.

2.3.4 Vaccination will

As in the previous section, this feature is inspired by MVT and uses the ratio of f to f^* , here referred to as x . We then have $x = \frac{f}{f^*}$ where f^* is a very low value.

To model the will to get vaccinated is used an increasing function:

$$\omega(x) = 1 + A \frac{x^2}{1 + x^2} \quad (3)$$

where:

- As before, $f = \frac{M}{N}$ is the ratio of infected to total population.
- f^* is a very low value, set to 0.01 in the simulations. The x parameter is normalized, making it suitable for modeling;
- A is the amplitude parameter of the modulation.

For a generic individual, the vaccination process can be described as:



with $0 \leq P(v) \leq 1$ representing the probability interval to get vaccinated. Official data [36]–[39] states that a complete two-dose cycle of Oxford AstraZeneca COVID-19 vaccine has an expected effectiveness value of 81%, varying in a range between 72-87%. Using a probability density function, with an expected value of $E[x] = 81\%$, vaccine effectiveness is assigned to each vaccinated individual. The following formula is applied, assuming the distribution is uniform:

$$f(x) = \frac{1}{up - lw} \quad (2.4)$$

where:

- $f(x)$ is the probability density function.
- up is the upper bound of the range, set to 87.
- lw is the lower bound of the range, set to 72.

Vaccine Efficacy (VE) is assigned using a random number:

$$VE(ind_{id}) = \text{rand} \cdot (up - lw) + lw \quad (2.5)$$

Corresponding chemical value, e.g. 81, will be created and assigned to the individual in question, as follows: $ind_{id,VE=81}$.

The vaccine duration (VD) is associated and assigned accordingly. The duration of a vaccine's protective effect is directly linked to its effectiveness: once the effectiveness is determined within the defined range, a corresponding duration is calculated to represent how long the protection remains active.

The utility that a vaccine brings to an individual is now described. When one comes into contact with an infected individual, there is a probability of infection that varies according to several parameters. There is a reduction in this probability due to the Vaccine Efficacy of the individual at risk of infection, simply multiplying the probability of infection by $(1 - VE)$.

Chapter 3

Formalization of the model

3.1 The P system

Having described the state of the art, anticipated some of the features the model possesses, and described how conflicts have been resolved, it is time to introduce the formal definition of the model [7] presented in this paper. It is, as described above, a cell-like P system with active membranes of degree $m \geq 1$, and without polarization. The P system in question is defined by the tuple:

$$\Pi = (V, H, \mu, w_1, w_2, \dots, w_m, R) \quad (3.1)$$

where:

1. V is the alphabet of objects. Each membrane contains objects, represented with multisets to map strings of symbols onto an alphabet. For example ab^2c^3 indicates a single instance of the chemical a , 2 copies of b and 3 copies of c .
2. H is the alphabet of labels for membranes. They have a label to distinguish them from different membranes of the same type.
3. μ is the initial membrane structure, of degree m , with all membranes labeled with elements of H . A membrane with label h is represented as $[]_h$.
4. w_i are strings over V specifying the multiset of objects initially in the i -ht regions defined by μ .
5. R is a finite set of evolution rules.

3.2 Membranes

The membranes that compose the model are now described. It is possible to introduce a tree-like membrane structure, starting from the skin membrane, and

continuing with the membranes described in the list shown later. A representation of this can be seen in Fig.3.1.

1. **Skin:** The outermost membrane delimiting the P system is called the skin membrane.
2. **Provinces:** The number of provinces in the model, abbreviated as np , is a parameter chosen by the user at the launch of the simulation. The provinces of the model, denoted by PV_p for $1 \leq p \leq np$, contain all the places membrane.
3. **Membranes that simulate places:** Places membranes have an integer value representing their capacity, which means the number of objects and membranes that can be held. Each place membrane has its own capacity (see section 4.3) and the number of places varies according to the population.
 - nr = number of regular structures per province, different for each facility. It is calculated on the population and capacity of each facility so there is always available space.
 - ns = number of sanitary structures per province, different between hospitals and ICUs. It is calculated according to the population and the rate of hospitals and ICUs in Lombardy. There will not always be places available for individuals in need.
 - nh = number of houses per province, calculated on the population with an average of 3 individuals per house, varying from 1 to 6 individuals.
 - (a) **House_{p,h}** for $1 \leq p \leq np$ and for $1 \leq i \leq nh$. Each house is a family unit where family members spend evenings and nights, and some days on weekends.
 - (b) **Hospital_{p,s}** for $1 \leq p \leq np$ and for $1 \leq s \leq ns$. A place where the severely infected go, i.e. with symptoms equal to E3 or E4. It allows a course of treatment to begin that will lead to the individual's recovery, if not interrupted earlier due to death.
 - (c) **ICU_{p,s}** for $1 \leq p \leq np$ and for $1 \leq s \leq ns$. A place where severely infected people go, i.e. those with E4 symptoms. Here, the probability of death is lower than by staying at home, and the course of treatment is continued (or started, if the patient is taken from somewhere other than a hospital).
 - (d) **School_{p,r}** for $1 \leq p \leq np$ and for $1 \leq r \leq nr$. A place where young individuals go to study.
 - (e) **Workplace_{p,r}** for $1 \leq p \leq np$ and for $1 \leq r \leq nr$. A place where adult people go to work.

- (f) **Common area_p** for $1 \leq p \leq np$. A single, crowded, indoor location, representing both transport (metro, buses, trains) and highly concentrated areas (shopping centers, supermarkets). It is therefore used by individuals of all ages to move between provinces or to spend a few hours during the week doing necessary shopping. At weekends it is possible to spend a few hours there.
- (g) **Leisure center_{p,r}** for $1 \leq p \leq np$ and for $1 \leq r \leq nr$. A place where individuals can spend their free time. The elderly can spend their days there, while young and adult individuals will visit it in the evenings on weekdays or at weekends. To minimize the number of membranes in the model, it is used as a multipurpose membrane. During the day it is considered an open place (squares, streets, parks) and during the evening it becomes a closed place (cinemas, discos, bars, and clubs). Contagion rates vary accordingly.
4. **Individual:** Membranes that simulate people, denoted by ind_{id} for $1 \leq id \leq population$. All people are considered of the same type. To define characteristics of the person, like the age group, a number of objects are defined that provide these characteristics within the region delimited by the membrane (that is inside the host).

The membrane structure of the P system is defined in such a way that within the skin membrane, there are all the provinces. Inside these provinces, there are places on the same level (houses, hospitals, workplaces, and others), and finally, the hosts (individuals) are inside these places. The hosts always go from one place to another, perhaps crossing the provinces, but they never go outside the P system environment. To move into a place, the membrane that models the host enters into the membrane that models the place using a movement rule.

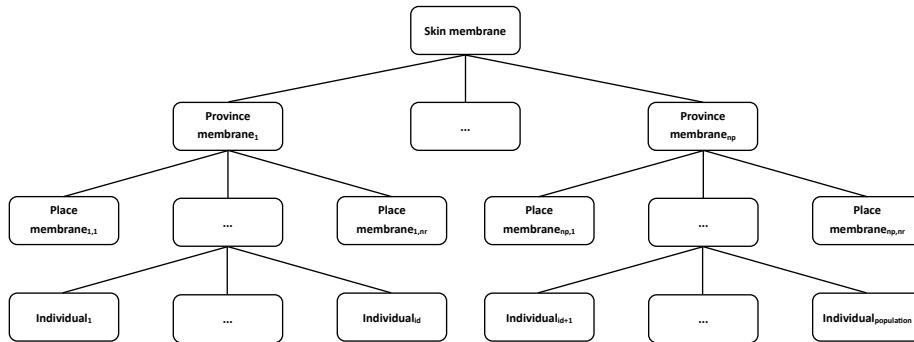


Figure 3.1: Tree structure of the membrane system.

3.3 Objects

Inside the membranes just presented there are other membranes, as already discussed, or chemicals. These substances represent information, execute rewriting rules and allow the model to evolve and process data. For the sake of clarity, an explanatory list of some objects in the model is now presented.

1. **Day_i:** for $1 \leq i \leq 7$. It denotes the day of the week and is inside each host to control its schedule. Individuals have different schedules during the week than on weekends.
2. **Hour_i:** for $0 \leq i \leq 23$. It denotes every day's hour and controls the schedule inside every individual.
3. **Objects to define the age:** Each individual has an object that describes the age group to which he or she belongs. These can be:
 - young: represents an individual between the ages of 6 and 20. His trips are mainly to go to school, maybe passing through the common area, and leisure centers.
 - adult: represents an adult aged between 21 and 59. It mainly travels to workplaces, maybe passing through the common area, and to leisure centers.
 - elderly: indicates an elderly person aged 60 or older. Generally, they move towards the common area and the leisure centers during the afternoons.
4. **Objects related to viral load:** Many types of objects are defined, let's see the most important ones. An infected host can pass v_1^5 objects to another host, infecting him. The infection is simulated by a quantity x of object v_1 ; the more v_1 objects, the more infected the patient is. The membrane representing the patient has *antiv* objects (antibodies) that can eliminate v_1 with a small probability, and can transform into *antivesp* objects (specialized antibodies), that defeat v_1 more effectively. If a predefined number of *antivesp* objects are obtained the infection disappears completely, curing the host. When the infection ends, the host is marked as recovered, making it immune to new infections for a certain editable period of time.
5. **Symptoms objects:** These denote the host's health status. E_1 means the host is well (but may be asymptomatic), E_2 is mildly symptomatic, E_3 is severely ill, and E_4 is critically ill. Each status triggers different processes: E_2 can cause the host to stay home depending on the Prudence Parameter, E_3 requires hospitalization and E_4 requires ICU treatment. As seen in the SEJIRS model, in E_3 and E_4 death can occur with different probabilities. The obligation to stay at home for individuals with E_2 , present in LOIMOS, is modified by the possibility of going out wearing masks or similar devices.

This is discussed in more detail in section 2.3.1, dedicated to the Prudence Parameter.

6. **Infection Number Object:** Denoted by ϕ where $1 \leq \phi \leq n$ and n is the number of people in the considered place. It is a local object present in every place that indicates the number of infected people in the considered place membrane. It is used to calculate the probability of contagion.

Individuals have many other items describing information such a number that sequentially identifies it, their address, remaining days of hospitalization (if any), province of origin and destination, and many others. The additional alphabet symbol V is added, and used as suffix to represent information relating to vaccination status.

For example, a young individual, vaccinated, moving from PV_1 to PV_2 is represented as ind_{young,V,PV_1,PV_2} . Additional information characterizing an individual has been omitted in this example.

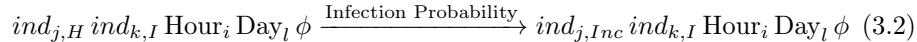
3.4 Rules

As described in section 1.1.2, the rules in this P system have additional properties that are essential for modeling an effective simulator. Every rule $\alpha \xrightarrow{p,q} \beta$ has two parameters $p \in [0, 1]$ is a probability value that models the stochastic aspects necessary for a reliable simulation of real systems. Rules have to be applied with priority, and the parameter $q \in \mathbb{N}$ is this priority value. This allows P systems to be effectively used in probabilistic modeling of biological networks. In some rules presented in this work, if there are no other similar rules, then the priority parameter is omitted as it cannot be compared.

3.4.1 Infection rules

The infection rates presented in the LOIMOS work classify individuals based on groups that take into account age, health status, and possible symptoms resulting from the infection. Therefore the general idea behind the rules is to describe a dynamic process of infection among individuals in specific places. The probability of infection, based on the number of infected individuals present in a specific place, the vaccination status of the involved individuals, and the caution factor, determines whether or not to place an individual into an incubation state. If this occurs, the individual in question receives five v_1 charges representing his current viral load.

The infection rules for different types of place membranes involving individuals can be summarized and generalized with this evolution rule:



- Infection Probability: base infection rate $\cdot \frac{\phi}{\text{total individuals}} \cdot \psi(M)$

- Base infection rate: specific to the type of membrane (see Table 5.1).
- ϕ : Current infection count in the location.
- Hour_i and Day_l: respectively the *i*-th hour of the day and the *l*-th day of the week.
- Infection status: H for Healthy, I for Infected and Inc for Incubation.

3.4.2 Contagion rules

The main evolution rules of contagion for an individual are given, ordered by decreasing priority. All these rules take place within the individual membrane and involve chemicals inside it.

- $antiv^{200}, v_1^{200} \xrightarrow{0.012,4} antivesp, v_1_ino, antiv^{199}, v_1^{199}, sint^{200}$

The individual fights the disease and a new *antivesp* is created. The probabilities for applying this rule depend on the individual's state of health and can be observed in Table 5.3. Since it takes a certain amount of time for an infection to occur, the 200 *v₁* objects ensure the necessary time for the infection to grow up to 20%.

- $antivesp, v_1 \xrightarrow{1,3} antivesp, v_1_ino, sint$

Each specialized antibody fights the infection with a certain probability, generating harmless viruses.

- $antiv, v_1 \xrightarrow{0.001,3} antiv, v_1_ino, sint$

Each non-specialized antibody fights against the infection with a lower probability, generating harmless viruses.

- $v_1_ino, phag \xrightarrow{1,2} phag$

Harmless viruses are eliminated by phagocytosis chemicals.

- $v_1 \xrightarrow{0.035,2} v_1, v_1, sint$

Viruses (*v₁*) not occupied by the other rules are free to grow.

- $v_1 \xrightarrow{1,0.1} v_1, sint$

Symptoms (*sint*) increase according to the viral load present. Each of these evolution rules generates a certain amount of *sint* depending on how many *v₁* are involved in the left-hand side of the rule.

3.4.3 Symptoms rules

During the symptomatology process, each host is associated with a single object E_x , where $x \in 1, 2, 3, 4$, which represents its current status. At the outset of the simulation, the system is initialized with a predefined number of this E_x objects, one for each individual.

- E_1 : the host has no symptoms, and may or may not be infected
- E_2 : the host has mild symptoms, may or may not realize he is infected
- E_3 : the host has severe symptoms and needs hospitalization
- E_4 : the host has critical symptoms and needs ICU

The *flag* object serves to determine if the host has attained a high viral load. If this condition is met, the *flag* object creates a *cont* object, which initiates the corresponding actions linked to the elevated viral load. Again, these evolution rules take place within the individual membrane and involve chemicals present within it, and are ordered by decreasing priority.

- $E_1, cont \xrightarrow{1,3} E_2$

This rule changes the state from asymptomatic to mild symptoms. This occurs the first time a dangerous viral load is reached.

- $E_2, cont \xrightarrow{0.0015,3} E_3$
- $E_3, cont \xrightarrow{0.001,3} E_4$

This rule changes the state from mild to severe and from severe to critical symptoms. This occurs if the host maintains a high viral load. The probability of this rule is deduced from the probability of a host worsening to a severe or a critical status. Compared to LOIMOS they were lowered, previously they were 0.003 and 0.0025 respectively.

- $sint^{700}, flag \xrightarrow{1,2} cont, flag$

When there are 700 *sint* objects, they take the *flag* object and generate the high viral load state represented by *cont*. The object *cont* will trigger the next processes.

- $E_2, cont \xrightarrow{1,2} E_2, cont$
- $E_3, cont \xrightarrow{1,2} E_3, cont$
- $E_4, cont \xrightarrow{1,2} E_4, cont$

If the *cont* object exists then the host remains in the same state. Given that the viral load is dangerous the health state, if it does not worsen, is maintained.

- $clear_cont, cont \xrightarrow{1,2} \lambda$
- $flag \xrightarrow{1,1} flag, clear_cont^{1000}$
- $clear_cont \xrightarrow{1,1} \lambda$

The three above rules are used when the viral load is not high. When do not exist 700 *sint* objects, the *flag* object is not selected, so the rules defined above are executed. The *flag* object generates *clear_cont* that delete all the *cont* objects and eliminate the host's symptoms.

- $E_2 \xrightarrow{1,1} E_1$
- $E_3 \xrightarrow{1,1} E_1$
- $E_4 \xrightarrow{1,1} E_1$

There is no *cont* object so the viral load is not dangerous, the host is cured and the symptoms disappear fully.

- $sint \xrightarrow{1,1} \lambda$

The object *sint* is used to know how many objects v_1 there were in a previous step. They are only used to know if a dangerous viral load is reached, in the fourth rule. If they are not used, then are deleted.

Infected individuals with medium or high-level symptoms (E_3 or E_4) can, every hour, trigger a dissolving rule that results in death. The probabilities for the activation of this rule are now presented. The same data can be seen in Table 5.2, which compares them with those of the LOIMOS model.

- Probability of death for seriously ill patient (E_3): $1.6 \cdot 10^{-5}/\text{h}$.
- Probability of death for critical patient (E_4) in ICU: $10^{-4}/\text{h}$.
- Probability of death for critical patient (E_4) outside the ICU: $2 \cdot 10^{-4}/\text{h}$.

3.4.4 Movement rules

These rules are used in various cases but they can be divided into two main uses: to move through different regions or to move through places in the same region to perform one's daily routine. Let us now look at the first, which can be generally described with the rule 3.3.

$$[[ind_{PV_x, PV_y} \text{Hour}_i \text{Day}_l]_{PM}]_{PV_x} \xrightarrow{\text{Movement Probability}} [[ind_{PV_x, PV_y} \text{Hour}_i \text{Day}_l]_{CA}]_{PV_y} \quad (3.3)$$

where $y \in PV_i \setminus \{x\}$ with $1 \leq i \leq np$, $x \neq y$, PM is a generic place membrane and CA is the Common Area of the destination province PV_y . The elements in the rewriting rules can be interpreted as follows:

1. ind_{PV_x, PV_y} : an individual traveling from province x to province y .
2. $Hour_i$ and Day_l : respectively the i -th hour of the day and the l -th day of the week.
3. Movement Probability: as described in the appropriate section (2.2.2), this value also depends on the epidemiological situation in the destination province. In any case, it will be 0 if the movement between regions is not in the routine set up for that individual at time i on day l .

In simpler terms, a person who is characterized by a destination province different from the origin one will move to a different province membrane according to this rule. It follows from rule 3.3 that the movement rules can also be applied between membranes nested on several levels, such as this individual moving from his place membranes in one province to the common area in another province. It would be impossible for an individual to reside directly within a province without any place membranes to contain him.

The rule in 3.3 describes the movement of an individual between two different provinces. Although it appears to be a different behavior, the movement of an individual between two places (such as home and work) follows the same rules. There will no longer be a willingness as it is the individual's duty to follow the imposed routine. There will also be a lower level of depth in the membranes as the individual remains in the same province. An example is shown in the rule 3.4.

$$[ind_{adult} \text{ Hour}_i \text{ Day}_l]_{House} \xrightarrow{\text{Movement Probability}} [ind_{adult} \text{ Hour}_i \text{ Day}_l]_{Workplace} \quad (3.4)$$

1. ind_{adult} : an adult individual who could go to work.
2. $Hour_i$ and Day_l : respectively the i -th hour of the day and the l -th day of the week.
3. Movement Probability: If i is equal to 7 and l is $1 \leq l \leq 5$, it is a working day, so the probability of moving will be 1, 0 otherwise.

Movement rules then bring the simulations to life, allowing individuals to follow their own routines. This will be described in more detail in the section on routines. The rules presented are all from the membrane movement category. There are none belonging to the out or in communication categories rules, these would compromise the structure in Fig.3.1.

3.4.5 Recovering rules

Having described how an individual can become infected, how the disease grows in his or her membrane, and how symptoms manifest and develop, the ways in which an individual can be cured are now shown. There are three ways to recover from the infection without dying:

Recovering through hospitalization

An infected individual with average or severe symptoms (E_3 or E_4) every hour has a probability of 0.03 of being moved to hospital, if there is one in his region that can host him. This parameter has been maintained from MVT, and a similar process is done for ICUs. If one is already in a hospital and the individual presents severe symptoms (E_4), then he/she will be taken immediately to the ICU. If the individual was in another place membrane, there is a probability of 0.05 that he/she will be taken directly to an ICU. This value is higher than that for hospitals because the situation is more severe. It is important to remember that the number of places in ICUs is limited, and a good simulation should take into account setting the correct parameter. In both cases, the hospitalization cycle starts, which lasts 7 days (settable parameter), during which the rule 3.5 will be applied. At the end of the set days the individual will become Recovered, immune to the infection for 180 days and his viral load will be zeroed, as shown in the rule 3.6.

$$[ind_{I,hdl=i}]_{Hospital} \xrightarrow{1} [ind_{I,hdl=i-1}]_{Hospital} \quad (3.5)$$

$$[ind_{I,hdl=0}]_{Hospital} \xrightarrow{1} [ind_{R,idl=180,E_1,v_1=0}]_{House} \quad (3.6)$$

1. hdl : hospitalization days left.
2. idl : immunity days left.
3. E_i : individual's level of symptoms, equal to E_1 in this case.
4. Infection status: I for Infected and R for Recovered.

Both rules just presented remain valid for ICUs and function in the same way. The variable hdl is shared between the two facilities because the cycle of care is not interrupted in the transfer from the hospital to the ICUs (or vice versa).

Recovering through specialized antivirus

Another way in which healing is possible is by achieving sufficient specialized antivirus (*antivesp*) to be immune to the disease. This allows the entire viral load to be defeated to zero, as seen in the rule 3.7. The rate of antibody generation and the number needed to heal (set at 40) are therefore a key parameter in the model, and its tuning is presented in section 5.2.

$$[ind_{I,antivesp \geq 40}]_{House} \xrightarrow{1} [ind_{R,idl=180,sym=E_1,v_1=0}]_{House} \quad (3.7)$$

Recovering by zeroing the viral load

This is the most intuitive and obvious way, also understandable from the contagion rules. If antivirus (*antiv*) and specialized antivirus (*antivesp*) manage to bring the viral load (v_1) to zero by fighting it, then the individual is considered cured. This leads to the consequences already presented in the other evolution rules, completely curing the host and making it immune to the disease for 180 days.

3.5 Routine and SPP

This section is dedicated to the movements of individuals, in particular the routine they follow from day to day. The MVT model did not have weekends, while LOIMOS did, and it was decided to implement them in the current model. Some places are not present in the tables summarizing the movements of individuals: these places are hospitals and ICUs, which can only be accessed by individuals with average or severe symptoms. The routines of individuals who are housed in these places do not allow any movement until the course of treatment is finished. Moreover, because of the *PP* there is a possibility that some individuals with low symptoms do not follow these routines and remain self-isolated in their houses. As can be guessed, a worker with symptoms might not go to the workplace and spend the entire day at home. The cells in the following tables that contain more than one place membrane represent a population that is divided into several places at that particular time. These tables provide a key to understanding which places and times are the most dangerous in the simulator.

- Routine for weekday: Table 3.1 shows the place membranes that each individual, divided by age group, might visit during the week. It is interesting to note that some young and adult individuals, before and after spending their day at school or in the workplace, pass through the common area. This happens because they have to cross their province of origin to get to the destination province, where the place membranes to which they are headed are located. The consequences of this will be described in the section 5.2.4.
- Routine for weekend: Table 3.2 shows the routines of individuals during weekends. Although there are many options for going out, the majority of the population will spend a good portion of the weekend at home. The day starts with some individuals in leisure centers due to individuals having spent the evening out. After that, during the day, people of all ages can move quite freely between places, creating many opportunities for contagion. Weekends have a higher computational cost than weekdays, due to this high movement.

During the week, as already described in the section on travel between provinces, individuals can cross provinces. This brings together individuals

Hour	Young	Adult	Elderly
0-6	House	House	House
7	House, CA	House, CA	House
8	School	Workplace	House, CA
9-16	School	Workplace	House, CA, LC
17-18	House, CA	House, CA	House, CA, LC
19	House, CA	House	House, CA, LC
20	House	House	House, CA, LC
21-23	House, LC	House, LC	House

Table 3.1: Daily schedule for the weekday, where LC stands for Leisure Centers and CA stands for Common Area.

Hour	Young	Adult	Elderly
0-3	House, LC	House, LC	House
4-7	House	House	House
8	House, CA	House, CA	House, CA
9-11	House, CA, LC	House, CA, LC	House, CA, LC
12-14	House	House	House
15-18	House, CA, LC	House, CA, LC	House, CA, LC
19-21	House	House	House
22-23	House, LC	House, LC	House

Table 3.2: Daily Schedule for the weekend, where LC stands for Leisure Centers and CA stands for Common Area.

who would otherwise have never met. The movement of flows of people is an interesting and important aspect of the model, which required some attention, also discussed in the section on behavior. For this, the number of individuals having a province of origin equal to the province of destination is not random but dictated by a parameter called Same Province Percentage, abbreviated to SPP. In the simulations shown, this is set at 0.8, so we have 80% of the population having identical origin and destination provinces. The effect this parameter has on the simulation is studied and shown in the section on the validation of the results, specifically in section 5.2.4. It can be guessed that a low parameter leads to more movements between the common areas of the provinces, causing more infections and raising the number of deaths.

In addition to the one made for healthcare facilities, there is another exception that obliges individuals not to follow these routines. These are the quarantine periods that can be imposed when the simulator is launched. During these periods no individual may leave his house membrane, and this is done by setting the probabilities of all movement rules in the model to zero. Any individuals residing in medical facilities will have to finish their course of treatment before being returned healthy to their homes. The infection progress of individuals obviously remains active, and contacts in houses become the only ones that can lead to new infections. The probability of death remains the same during these periods. The quarantine periods add utility to the model, because it is useful to understand whether it makes sense to do it and how long to make it last in order to maximize its effectiveness. The validation section will discuss this possibility more specifically, and will also show, in section 5.2.3, the variation of curves with and without targeted quarantines.

Chapter 4

The simulator

The development of the simulator and the subsequent parameter tuning was carried out in collaboration with *Francesco Reiff* from the University of Verona. His contribution was particularly significant in the design and refinement of the simulation framework, ensuring both computational efficiency and consistency with the epidemiological model. The collaboration involved extensive discussions on possible modifications to the implementation, carefully evaluating which changes were most appropriate. While the model structure was designed in this thesis, *Francesco Reiff* contributed significantly to the implementation phase, focusing on writing the code and conducting extensive testing. Additionally, the initial conflict resolution process was carried out together, ensuring consistency and correctness in the simulation framework. Furthermore, the collaboration with the University of Verona, represented in this context by *Francesco Reiff* and co-supervisor *Giuditta Franco*, facilitated a productive exchange of ideas that benefited both parties, enriching the development of the simulator and the overall research process.

4.1 Implementation

The implementation in this thesis follows an Object-Oriented Programming (OOP) approach, similar to MVT, rather than using pre-existing code designed for the membrane computing paradigm, as seen in LOIMOS or others [40], [41]. This is done for convenience in code modification, simulation efficiency, and minimization of required changes, as the initial program is MVT, which already follows this paradigm.

Consequently, certain criteria had to be followed when writing the code. The OOP facilitates the translation of a P systems structure into a programming language, making the process more intuitive. Individuals are represented as distinct programming objects, while their characteristics, captured by smaller sub-objects, are defined through specific parameters. Place membranes are transformed into larger programming objects designed to manage additional data

structures, allowing for the tracking of individual positions and the application of rules based on contextual logic. Province membranes are interpreted as collections of place membranes, serving to oversee the broader epidemiological context and exchange objects with other province membranes.

- Using object-oriented principles, entities such as individuals, contextual resources, and status information are modeled as objects, enabling an effective translation of P systems into a programming language.
- The hierarchical organization of the membranes forms a tree structure that organizes the various compartments, aligning perfectly with the requirements of an epidemiological model.
- Rewriting rules are reinterpreted as functions and methods within the program, representing key processes like infection spread, movement of individuals, vaccination campaigns, and virus incubation. These functions formalize the computations needed to simulate real-world dynamics.

Following the membrane computing paradigm, each class attribute cannot change more than once per hour because it can be part of only one rule per step. Attributes that have just increased or decreased (by generating or removing chemicals) will only be read with the updated value in the next step, or the membrane computing paradigm will have failed. The rules applied follow priorities and probabilities of occurrence, and the movement of membranes must be calibrated so that the skin can only contain regions, which can only contain places, which can only contain individuals, which can only have attributes that characterize them. For a more concrete example of this concept, hour and day have been defined as objects, and so they must be for a properly functioning membrane system, but in the code, these are simple variables in a loop. The result is the same, but conceptually, they are two very different things. Thanks to OOP, there was no need to *flag*, *cont*, or *clear_cont* object for symptom management: these were only present in LOIMOS. As mentioned in 2.1, Python was retained as the programming language, which is excellent for its simplicity, easy data manipulation, and support for OOP.

4.2 GUI and simulation

Since this simulator could find a great deal of use in many fields, especially those not directly related to IT such as virology, infectious diseases, healthcare, and many others, a graphical interface was developed that could also be used by experts in these fields, even with no knowledge of the technicalities at the basis of the simulator. Fig.4.1 shows the five screens that compose it. Although it does not present elaborate graphics, all the most important parameters are listed in this interface and their editing has been greatly simplified. Before the GUI was created, to change a parameter it was necessary to find every call to that parameter throughout the code and fix it, hoping not to forget any of them.

All modifiable values not present in the GUI, such as SPP , are listed at the beginning of the main model class, to make their modification feasible even for less experienced users. The addition of the GUI, although a trivial feature, greatly broadens the target group of users to whom the work proposed in this thesis may be of interest.

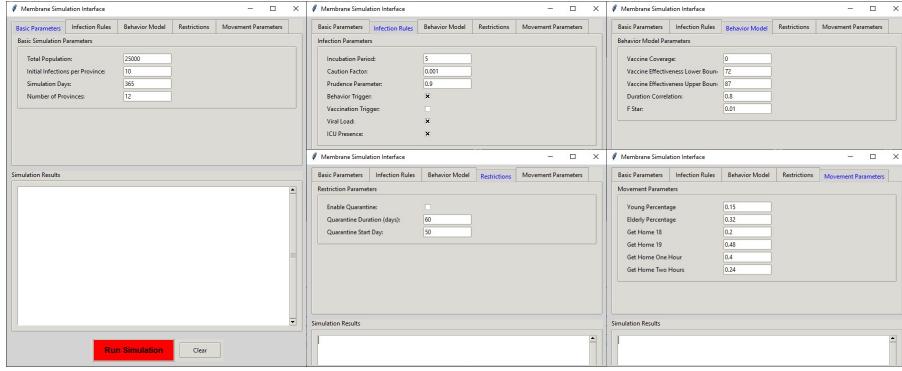


Figure 4.1: GUI of the simulator.

Now that the required input parameters have been presented, the operation of a simulation, consisting of a computation divided into steps, is described. A computation works from an initial starting state towards an end state through a number of discrete steps, called configurations. A configuration of such a system at time t is defined by the membrane structure μ_t and the multisets of objects associated with each region in μ_t . Each configuration involves iterating through all membranes in the P system and the application of rules, which occurs in both a maximally parallel and non-deterministic manner [42]. The rules section, just described, presents some of the most important and descriptive rules of the model.

A computation step, i.e. the transition from one configuration to the next, is carried out as the simulation progresses over time, specifically one hour for each step. In each of these, each object can only perform one rule, after which it moves on to the next step. Working through step-by-step, a computation usually halts when no further evolution can take place (i.e. when no rules are able to be applied). Since this is a simulation, among the inputs to be provided is the number of steps beyond which to stop. It is therefore evident that the model was created to not have a fixed number of steps, and is capable of processing information forever, theoretically.

The pseudocode shown in Algorithm 2 outlines the simulation process, structured into two main functions:

- **create_scenario()**: Initializes the environment by generating provinces, places, and individuals with the correct demographic distribution. Initial infections are also introduced.

- `run_simulation(days, hours per day)`: Governs the epidemic progression, handling mobility, infections, health status updates, hospitalizations, and data recording.

When the set number of steps is met, in this case, dictated by days, the output stage is reached. This was very different in the two starting models. In LOIMOS you get lots of information for each hour, to be exact, a CSV file with 224 columns. In MVT the results are the values of the current infected, prevalence of the virus, new daily cases, and deaths, but for every day, not every hour. In the resulting model, there are the values in MVT for each step, time data describing how many seconds were required to complete that step and the cardinality of each of the classes of the SEJIRS model. Graphs corresponding to the number of infected and dead obtained are also automatically generated. In the next chapter, dedicated to the validation of the results, the outputs of the calculations will be analyzed in more detail. All the material shown can be found in the GitHub link [43].

In addition to conflict resolution, many features have been added to the simulator, which is now adaptable both from the point of view of the population and the environment in which this population moves and from the point of view of the infection that is being simulated.

4.3 Scalability of the scenario

The last major task of this work is the generalization of the model, in particular on two different aspects. The starting simulators are in fact specific to COVID-19 and present a scenario that is already configured and ready to go, but not scalable. In LOIMOS, all input membranes have to be entered in a previously prepared file, and this is not easy to modify. In MVT the number of regions is fixed. For these and other reasons adjustments were necessary to allow the code to generalize, both from the point of view of the simulated scenario and the type of infection.

In MVT there is an unrealistic implementation of place membranes. There is a variable called *numberOfStructures*, set to 50, which instantiates for each province that number of workplaces, schools and hospitals. The capacity of these increases as the population grows until it reaches unreliable values, but the number of structures always remains the same. In the new simulator, a realistic capacity searched on the network was assigned to each type of place membrane:

- School: 300
- Workplace: 200
- Leisure centers: 200

These places membranes are generated in sufficient quantity to contain the entire population. Another type of approach is designed for healthcare facilities, namely hospitals and ICUs. These have a capacity of 150 and 1 respectively (one is

Algorithm 2 Pseudocode of the simulator.

Input: Simulation parameters, days, population
Output: Epidemic evolution data, seconds, SEJIRS class cardinality

```
1: function CREATE_SCENARIO
2:   for all province in provinces do
3:     Create province membrane instance
4:     Calculate the correct number of place membranes
5:     Add place membranes
6:     Create individuals with the correct age distribution
7:     Assign individuals to houses
8:     Introduce initial infections
9:   end for
10:  end function

11: function RUN_SIMULATION(days, hours per day)
12:   Create CSV file
13:   for all day in days do
14:     for all hour in hours per day do
15:       if not a quarantine day and is the correct hour then
16:         Move individuals between provinces through common areas
17:         Move workers to workplaces
18:         Move students to schools
19:         Move elderly to leisure centers
20:         Simulate infections in all places with individuals
21:         Move individuals back home
22:       end if
23:       Trigger infection and vaccination progress
24:       Check for deaths
25:       Discharge recovered individuals from hospitals
26:       Check for hospitalization based on hospital capacity
27:       Track infections and update the scenario
28:     end for
29:     Reduce recovery days in hospitalized individuals
30:   end for
31:   Write data to CSV file and create charts
32: end function
```

completely isolated inside an ICU). The number of these facilities is initialized to reflect the healthcare coverage that, on average, exists in Europe. This makes it possible to simulate the overcrowding of medical facilities, a problem discussed and of great interest during drastic moments of the COVID-19 epidemic. Having sufficient hospital capacity to contain the entire population is unrealistic.

A big step in the scaling of the scenario was certainly to make the number of provinces modifiable from the graphic interface. Another change concerning the provinces was the arrangement of origin and destination provinces. During the creation of an individual, the latter was randomly assigned, leading (in the case of 12 provinces) to only 8% of the population having the same origin and destination province. It was not possible to adjust this value comfortably without intervening in the code.

This problem was solved by simply inserting a parameter, usually set at 80%, which instantiates 20% of the population with different destination provinces and origin provinces named Same Province Percentage (*SPP*). By raising this parameter, scenarios could be simulated where, for example, the provinces are poorly connected and individuals consequently rarely cross them. Lowering it could simulate a city divided into neighborhoods with a high flow of people between them. Combined with the possibility of varying numbers of individuals and provinces, the simulator is now much more flexible.

4.4 Generalization of infections

Another level of generalization was undertaken with regard to the type of disease. Although the simulations were performed with COVID-19 parameters, and therefore the work in this thesis is related to this disease, these can be varied to suit new epidemics. LOIMOS and MVT handle the growth of infection in two completely different ways: the former assigns a viral load, consisting of many variables to each individual, while the latter has a cycle with unchanging days, in which one goes from Incubation to Infected to Recovered. Although the former is closer to reality as far as COVID-19 is concerned, there are infections whose behavior does not reflect that proposed by a viral load model. This is why, in the GUI of the new simulator, there is a checkbox that allows the user to decide what type of development the infection being simulated has: viral-loaded or fixed-day. In the case of working with an epidemic about which little is yet known, it would be unlikely to fit the model with the parameters proposed by LOIMOS, so the more generic option present in MVT would be more appropriate.

Another checkbox that can be activated or deactivated is the one regulating the presence or absence of ICUs. These structures, which are of vital importance for the COVID-19 outbreak, may be superfluous in simulations related to different infections. For this reason, their presence can be removed.

In general, the assignment of many parameters has been moved to the GUI, making the simulation easier to modify for any type of user. From there, it is also possible to decide on the starting day and duration of a quarantine period. This will prohibit any person from leaving, which is useful in order to find the

best time to impose a quarantine that minimizes the negative consequences of the disease.

A final note should be made regarding the Prudence Parameter (PP), introduced in section 2.3.1: as PP increases, individuals become more cautious, reducing the probability that they leave the house while having symptoms. Whereas LOIMOS forced symptomatic individuals to stay at home, this parameter adds more flexibility to the model, allowing it to adapt to different situations. This parameter has an important function in the generalization of the model. So far, work has been done on a simulator dedicated to COVID-19, a widely studied and globally known disease with recognizable symptoms. If it were to return, perhaps with a more aggressive variant, society would probably be able to recognize it and act accordingly. Widespread warnings would be sufficient to make the entire population realize that the risk of a similar pandemic is recurring, leading to increased caution. In this case, therefore, the population has a high PP value. If on the other hand, a new disease is being simulated, with symptoms that are not yet clear and well delineated, it would be difficult for the population to recognize it. A low PP value will allow this scenario to be modeled perfectly, with individuals becoming vectors of the disease without realizing it, as they mistake mild symptoms for milder, negligible diseases [44].

Chapter 5

Results

5.1 Dataset

An important part of the work performed is undoubtedly the tuning of the parameters and subsequent validation of the results. Even though the numerous parameters in the simulation were studied and derived from previous work, they led to simulations with higher-than-expected numbers of infected individuals and deaths. This discrepancy arises from a comparison with a database from the Lombardy region, which was not used in previous models. Moreover, each model was effective and achieved excellent results when executed in its own environment. In this thesis, multiple parameters from both models are being merged, and they must be carefully synchronized and precisely tuned. In particular, the values concerning the virus and its spread, such as the probability of infecting other individuals, the growth of the virus from hour to hour, or the various probabilities of death depending on the individual's health condition and symptoms, were tuned to allow the simulations to generate plausible data. This is, after all, the ultimate goal of the aforementioned work.

Performing a valid mathematical comparison and giving a rigorous definition of *plausible* is a non-negligible challenge in the field of epidemiological modeling. The real world takes into account so many factors and parameters that a simulator will, to a greater or lesser extent, have to neglect and approximate. The chapter on behavior is a clear example of this. Predicting the course of future and little-known diseases is a speculation that is unlikely, if not with time, to find scientific proof. In order to overcome this problem and meet this challenge, it has been necessary to research and analyze past data. This thesis is in fact dealing with a disease that has already peaked and has been extensively studied, generating various databases on the subject.

To carry out the tuning and refinement of the simulator, the database related to Lombardy [45] was exploited. This contains data from February 2020 to the present day concerning each Italian region, one of the European states most affected in the first phase of the epidemic [46]. In particular, the Lombardy

region, which has already been studied in many previous works [47]–[49]. Among the numerous data in the database, it was decided to tune the model to the two most impactful in the population, i.e. the number of infected individuals and the number of deaths. The graphs obtained from this database are shown in Fig5.2 and Fig5.1.

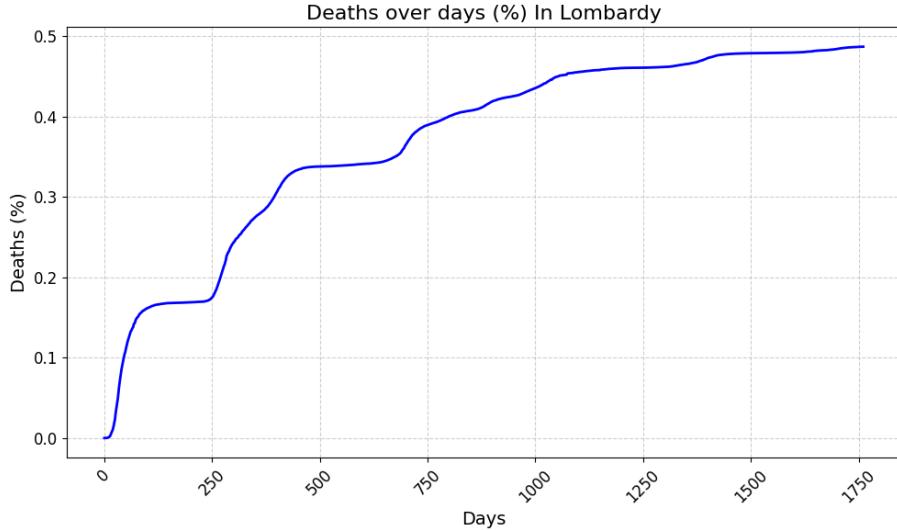


Figure 5.1: Lombardy death.

5.2 Tuning and Validation

In the preliminary phase of the thesis, the model achieved results similar to those of LOIMOS, but too high compared to Lombardy. MVT shows simulations with 25000 individuals in 365 days, and with a low caution value a prevalence of about 2000 is achieved. This corresponds to 8% of the population, which is a plausible value and close to the data shown. At the end of the simulation, there are 284 deaths, corresponding to more than 1% of the total. This value is quite high, if one compares it with the one in Lombardy. It should be remembered that since it is a cumulative value, having double the result (1% vs. 0.5%) in only 365 days, compared to the 1500 days in the Lombardy database, denotes a too high death rate.

LOIMOS has 10320 people in 291 days. Without quarantine there are more than 2500 infected, so more than 25% of the population. With quarantines and countermeasures, it drops to 1200 as a peak, which is 10%, reached after only 30 days. These figures do not reflect the Lombardy database and need to be downgraded. These analyses also led to the conclusion that the probability of being infected taken from LOIMOS should be slightly reduced.

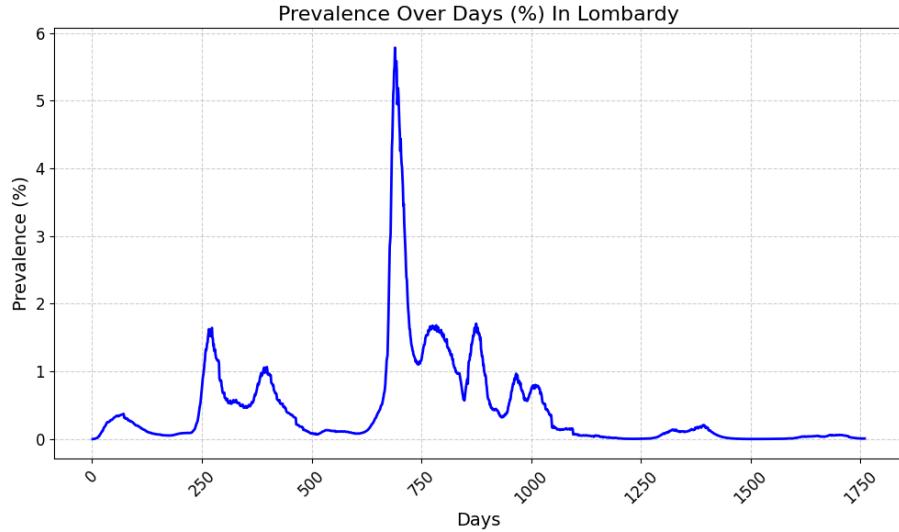


Figure 5.2: Lombardy prevalence.

Initially, the number of dead and infected was far above the recorded parameters, even more than 10 times. The latest simulations instead achieve the results shown in Fig5.3 and Fig5.4 As in MVT, these graphs are the result of simulations run with a population of 25000 for a whole year.

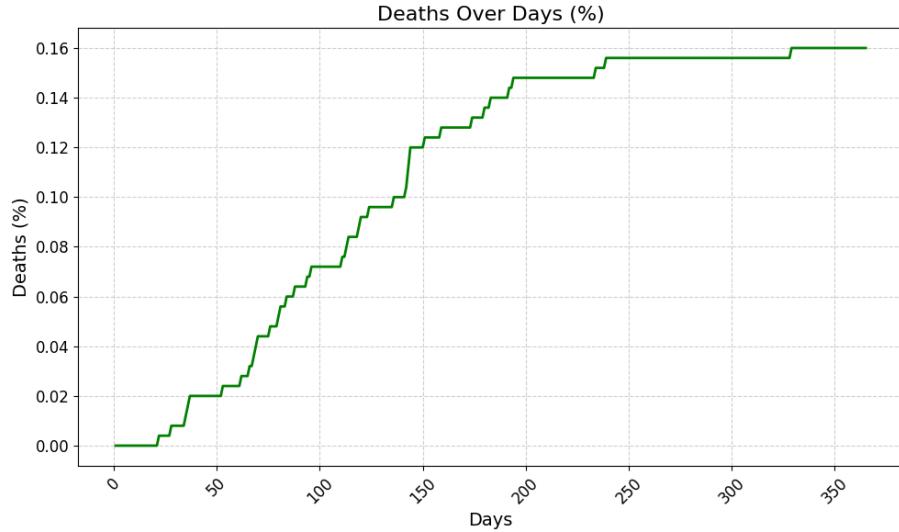


Figure 5.3: Simulator deaths.

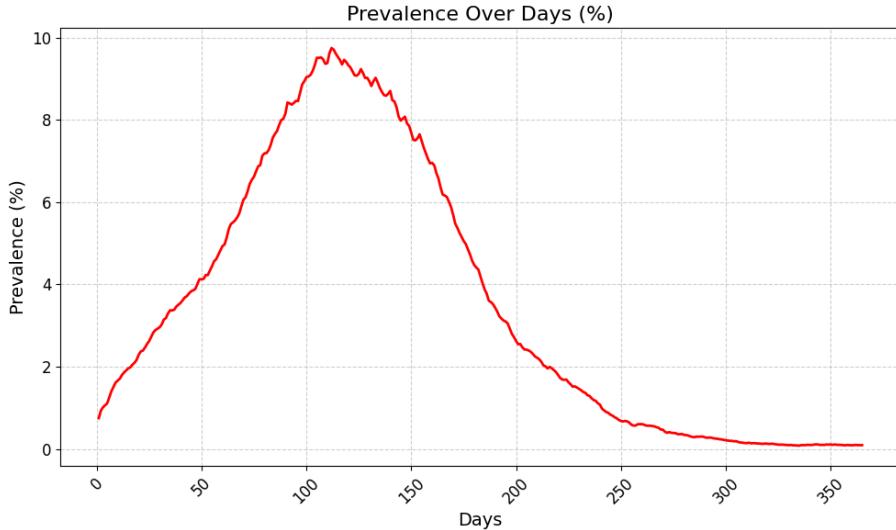


Figure 5.4: Simulator prevalence.

To achieve these parameters a careful tuning phase was necessary. This involved many of the variables presented so far, but in particular the following four:

- **Infection rate:** The probabilities of infection were different, though similar, for each model. Table 5.1 shows them all. All probabilities are marked as ‘young and adult’/‘elderly’, highlighting that in the models there is the same probability of being infected whether you are young or adult. An x means that it is not possible for individuals of that age group to go into that membrane. There is a notation to be made. While MVT is actually divided into the 3 age categories, LOIMOS has 4 categories concerning the types of infections, which are difficult to map into ages. The category "E-inn/N-acq/N or E-inn/L-acq/N asymptomatic" was associated with young and adult, while the category "I-inn/W-acq/S or I-inn/N-acq/S weak symptoms + 60 years old" are the elderly. In the table, you can see that it is not possible to get infected in ICUs or hospitals. This change was made because, while in LOIMOS there are some adults working in the healthcare sector, in the model of this thesis there is no such possibility. It was deemed unnecessary to add this option to avoid complex rules, such as adults going to the hospital even if they are healthy, but not occupying beds. The choice was justified by the fact that the proportion of individuals working in healthcare is minimal, and they have a similar probability of becoming infected as other workers. Despite the fact that they work in close contact with infected individuals, they have the necessary knowledge and tools to avoid being infected. As far as individuals in hospitals and

ICUs are concerned, they are all already infected, and as soon as they finish their course of treatment, they are sent home, so the absence of methods for infection does not create any differences. However, in accordance with reality, the possibility of healthy individuals going to the hospital for reasons other than the studied disease was not added. The option, applied in the real world, of dividing the hospital into sections dedicated to infectious diseases, leaving the healthy area of the facility completely aside, was considered valid. To better simulate the data from the Lombardy region, the infection rates were reduced by 15%. This adjustment accounts for the widespread use of face masks, as recorded in a database spanning from 2020 to the present. Face masks have been estimated to significantly reduce transmission rates. However, due to inconsistent usage, incorrect wearing, and their application being limited to specific locations, the overall reduction was estimated at approximately 15%. This correction ensures that the model accurately reflects the real-world mitigation measures in place during the studied period.

Place	LOIMOS	MVT	Tuned
Common Area	0.02 / 0.2	0.02 / 0.2	0.017 / 0.17
House	0.02 / 0.2	not present	0.035 / 0.35
Workplace	0.02 / x	0.02 / x	0.017 / x
School	0.03 / x	0.03 / x	0.017 / x
Retirement homes	0.04 / 0.4	not present	not present
Hospitals	0.05 / 0.5	0.05 / 0.5	0 / 0
ICUs	0.05 / 0.5	not present	0 / 0
Leisure Centers (Day)	0.06 / x	not present	0.026 / 0.07
Leisure Centers (Night)	0.06 / x	not present	0.052 / 0.07

Table 5.1: LOIMOS, MVT, and final simulator infection rate divided as ‘young and adult’/‘elderly’

- **Death rate:** Similar to the above, Table 5.2 shows the initial and post-tuning death rates, which have been drastically reduced. Every hour, individuals with certain characteristics have a probability of dying. In MVT this probability is unique and fixed at 5×10^{-4} for individuals in hospital.

The LOIMOS parameters are given in the table, where those resulting from parameter tuning are also present. LOIMOS reported, in the first simulations, a very high death rate. In order to balance these exaggerated numbers, the death rates have been brought down to one thirtieth of the initial ones.

The LOIMOS parameters are given in the table, alongside the adjusted values obtained through parameter tuning. The initial simulations

with LOIMOS reported an excessively high death rate, likely due to overestimated fatality assumptions in the early stages of the pandemic. To better align with real-world data, particularly for the Lombardy region, the death rates were reduced to one-thirtieth of their initial values. This correction accounts for advancements in medical treatments, improved hospital capacity, and better patient management strategies that significantly lowered mortality compared to early projections.

Condition	LOIMOS	Tuned
E_3	$5 \cdot 10^{-4}$	$1.6 \cdot 10^{-5}$
E_4 in ICU	$3 \cdot 10^{-3}$	10^{-4}
E_4 outside ICU	$6 \cdot 10^{-3}$	$2 \cdot 10^{-4}$

Table 5.2: LOIMOS and final simulator death rate

- **Antivesp generation:** Each individual, depending on age and level of symptoms, has a certain probability of generating antivesp (specialized antibodies). These antibodies fight the disease extremely effectively. In the initial LOIMOS simulations, the probability of generating antivesp was too low, leading to the disease being excessively deadly. This was corrected by tripling the generation rates. The adjustment was made to reflect a more accurate immune response, as the initial model did not properly account for the large number of individuals who recover due to their body's own immune system. In the real world, many individuals recover without the need for hospital intervention, and a large portion of patients are cured thanks to their own immune system. However, in LOIMOS, the number of patients recovering in hospitals was initially very high, while the number of people who could recover by generating antivesp was too low. After introducing capacity limits for hospitals, it became clear that these would never be sufficient to meet the demand. This outcome was problematic for the new simulator, where reducing the number of hospital beds to align with real-world conditions would have led to an increase in the number of infected individuals rather than a decrease. To correct this, the immune response was strengthened by increasing the probability of antivesp generation, making individuals more capable of combating the disease independently and preventing the unintended effects of overloading the hospital system. The new antivesp creation rates are shown in Table 5.3.
- **Antivesp needed for immunity:** Upon reaching a certain number of antivesp, the individual is able to defeat the disease independently and becomes cured. The number in question has a strong impact on the simulation and is currently set at 40, which ensures that the disease duration for each individual is consistent with the average duration of illness observed in the fixed day cycles of MVT. This number remains

	E ₂	E ₃	E ₄
Young	0.024	0.018	0.012
Adult	0.012	0.009	0.006
Elderly	0.004	0.003	0.002

Table 5.3: Antivesp creation rates according to symptoms and age.

easily modifiable.

5.2.1 Validation against Lombardy

This subsection is dedicated to the comparison between the results obtained from the simulator and the Lombardy data. No direct comparison is made in a single graph as the Lombardy trend contains numerous peaks, whereas the model presented tends to simulate only one. Several simulations with different parameters were therefore run, and it can be seen that each of these is able to follow the course of the Lombardy curve for a certain stretch. Subgraphs of this curve are shown, as the Lombardy database covers almost five years and to simulate it in full would be too expensive. In Fig.5.5 a short simulation is presented, which after only 90 days is decreasing. This occurs because the infection parameters were set lower than those that led to the results in Fig.5.4. Despite this, a match can be found with a peak in the Lombardy graph, particularly from day 370 to day 460. A similar argument can be made for the graphs in Fig.5.6 and Fig.5.7. The latter, although it does not capture accurately how quickly the infection spread in Lombardy, manages to accurately predict the peak value reached. A comparison in the number of deaths between data from Lombardy and a one-year simulation is shown in Fig.5.8. This consolidates the drastic reduction in death parameters carried out in the tuning phase.

5.2.2 Validation against PP

Fig.5.9 and Fig.5.10 compare different simulations for prevalence and the number of deaths, respectively, varying only *PP*. The values chosen for this parameter were 0.25, 0.5, 0.75, 0.9 and 1.

As immediately apparent, the parameter introduced in this work has a drastic impact on the population. Simulations with $PP = 0$ were not carried out as they would make little sense. These would model a population in which all individuals, regardless of their age or the importance of their routine, continue to follow it and leave the house despite presenting symptoms. Even if these symptoms are mild, it is unthinkable that no individual, especially an elderly person, would stay at home when these symptoms occur. In the other simulations in this paper, the parameter was always set to 0.9. It must be remembered that symptoms do not occur immediately with the infection but may take several days before they occur, allowing the individual unaware of his or her condition to go out and spread the disease.

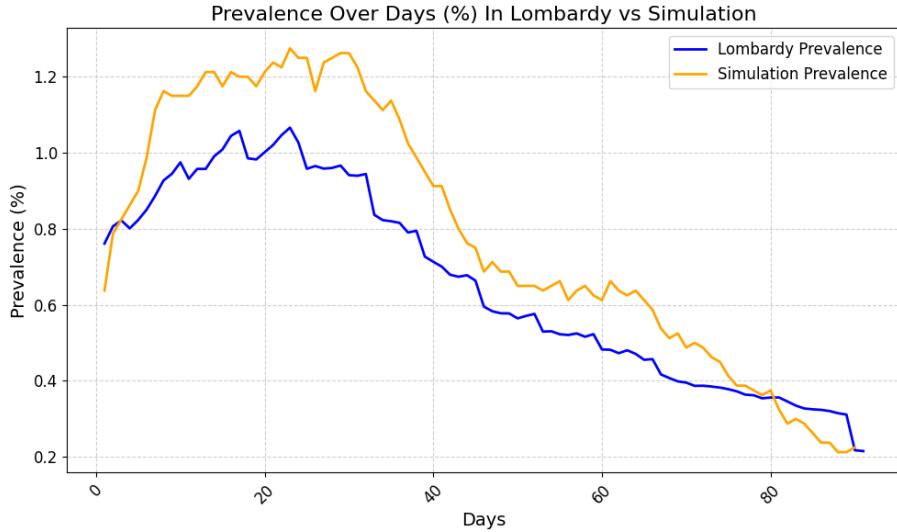


Figure 5.5: Prevalence comparison from day 370 to day 460.

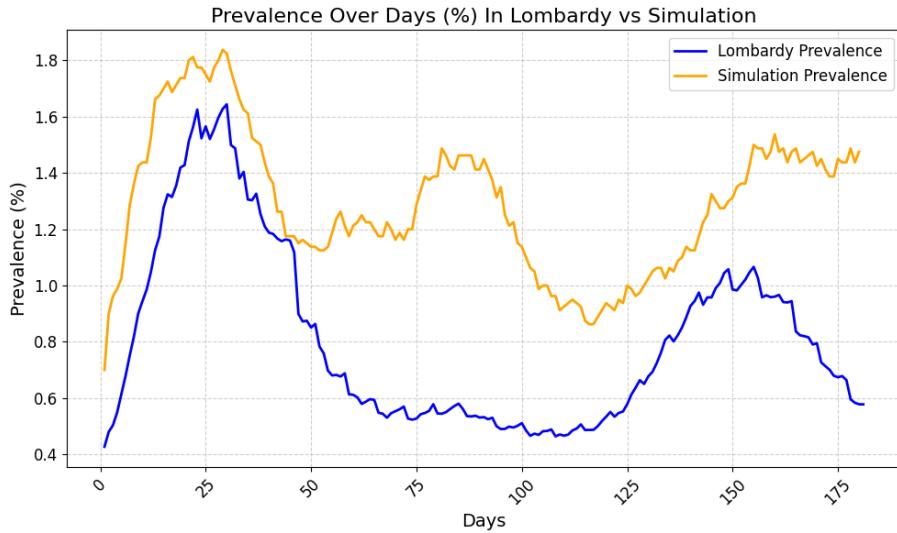


Figure 5.6: Prevalence comparison from day 240 to day 420.

Looking at the graphs, particularly the prevalence graph in Fig.5.9, one immediately notices how low PP parameters lead to drastic results, outside the reliable values observed in the Lombardy graphs. This suggests that a low PP parameter, below 0.25, does not model reality well and generates results outside the norm. Just as simulations with $PP = 0$ are not run, those with a too low

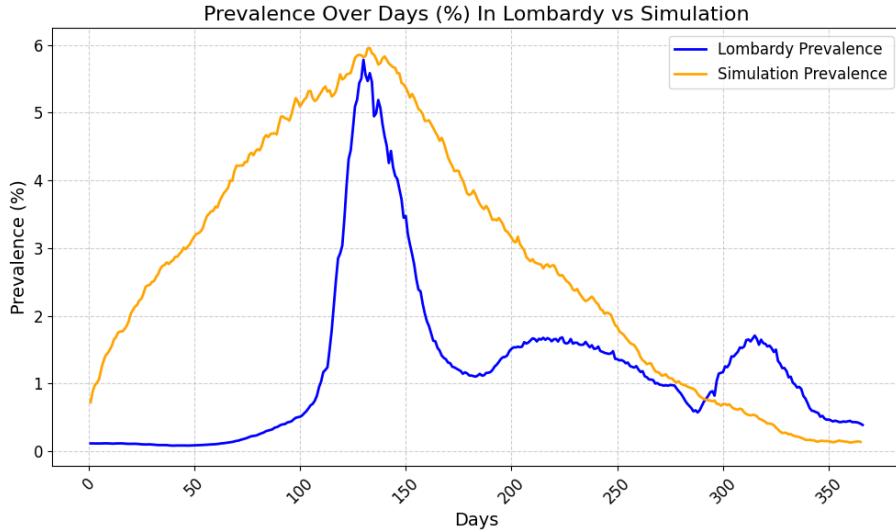


Figure 5.7: Prevalence comparison from day 550 to day 915.

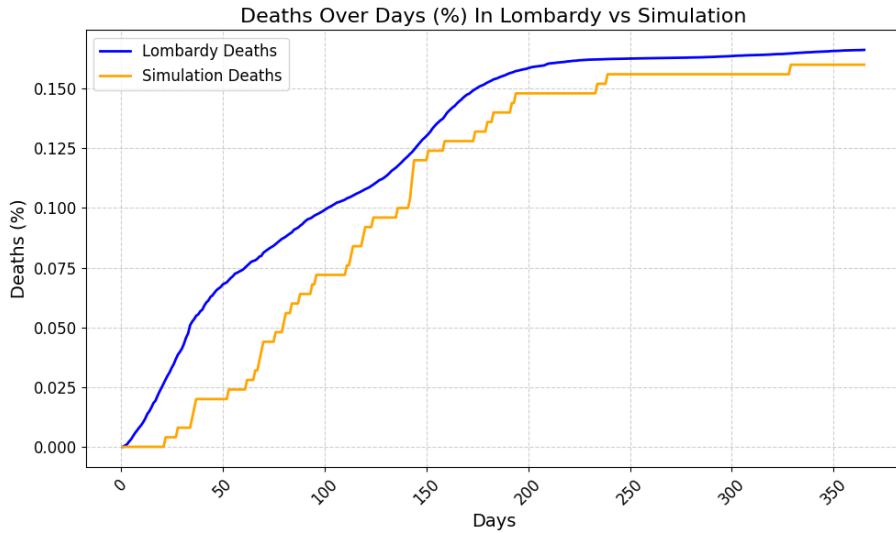


Figure 5.8: Death comparison from day 250 to day 615.

PP value are also unreliable. In general, if one aims to have reliable predictions for the COVID-19 trend, this parameter should never fall below 0.75. This is mainly due to the fact that the parameter tuning was done with a $PP = 0.9$, a decision made in order to maintain the characteristics of LOIMOS, which was developed as if it had a $PP = 1$. However, it remains interesting to note how

exaggerated misbehavior of the population can lead to such drastic consequences. This suggests that educational and awareness-raising campaigns to prevent the spread of infectious diseases, simply by self-isolation at home when the possible symptoms arise, can have a beneficial effect on the population as a whole. Table 5.4 summarizes the most significant values for each of these simulations.

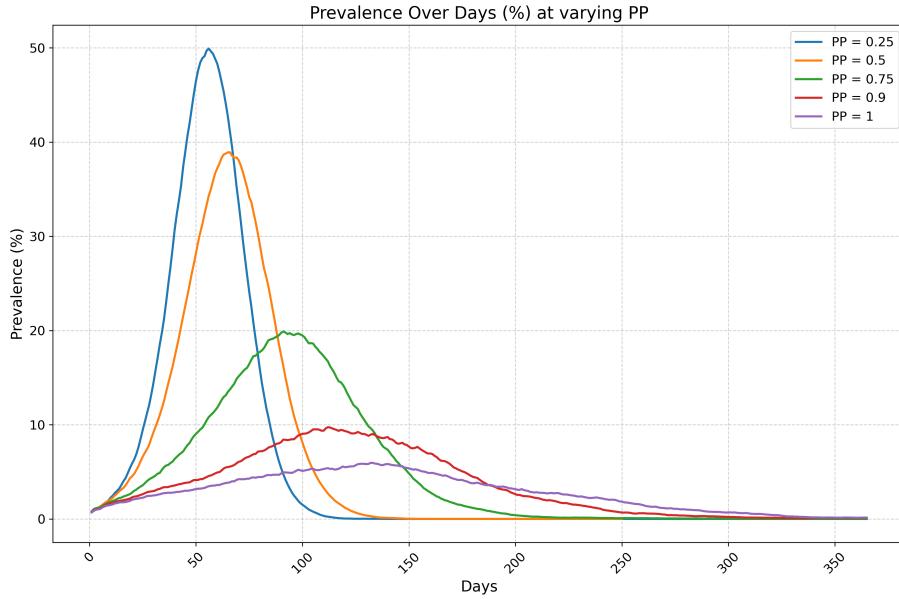


Figure 5.9: Prevalence varying the PP.

Type of simulation	Max prevalence	Final deaths
PP = 0.25	12481	101
PP = 0.5	9729	64
PP = 0.75	4976	48
PP = 0.9	2436	40
PP = 1	1464	30

Table 5.4: Different result varying the PP.

5.2.3 Validation against targeted quarantines

In this subsection, simulation results following the application of quarantine regulations are presented. The possibility of introducing quarantines is a very powerful weapon to combat infectious diseases, and one of the most effective. Although it does not lead to a definitive solution to the problem, only to a gain in time, it makes it possible to greatly reduce the number of new cases and

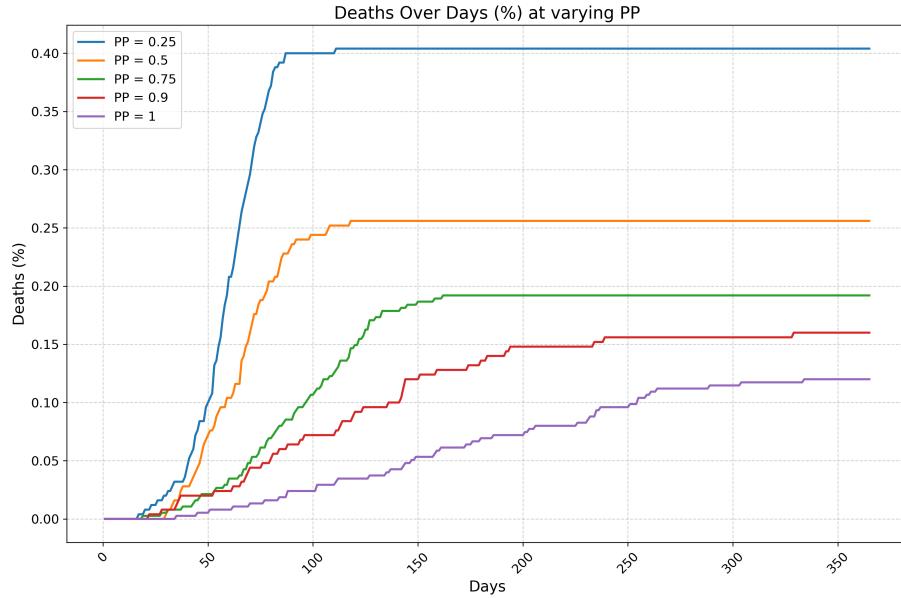


Figure 5.10: Deaths varying the PP.

keep hospital overcrowding under control. Despite these great advantages, the option of completely banning individuals from leaving their homes is an option that is only used in the most problematic cases. This in fact leads to much dissent in the subject population, both for psychological and mental well-being reasons, and for more concrete economic reasons. Prohibiting a population from leaving their homes is a choice that weighs on the economies of the families but, above all, on that of the government that makes this choice. For this reason, in setting up a quarantine, the aim is to maximise its effect by using as few days as possible. This simulator, therefore, aims to predict the course of the disease as quarantines change so that the most effective scheduling can be chosen.

To show the results achieved in this area a comparison of three simulations is shown in Fig.5.11. Again, the parameters are identical, except of course for the quarantine days set. In the first simulation, also taken as an example in the previous section with a PP = 0.9, no quarantines are carried out. The simulation with the addition of a quarantine period was prepared by knowing the course the disease might take and carrying out the closure of public places on the rising edge of the maximum peak.

A single, 60-day quarantine is carried out, starting on day 50. In the picture, it is shown with a dotted line of the corresponding color, orange. This led to a drastic drop in the number of infected, bringing them closer to the initial condition. With the end of the quarantine, however, this number rose sharply again. It still remained well below the value achieved without quarantine. This confirms that a sensible placement of a period of movement restrictions can slow

down and lower the negative impact of the disease.

An even more interesting result is provided by the third simulation. This has the characteristic of having a quarantine period split into three sub-periods. Each of these lasts 20 days and is alternated with 20 open days. In general, therefore, the cost of this maneuver will be similar to the previous one, as the total number of closed days is always 60. Initially, it follows a pattern almost identical to the two models, and similar to that with a quarantine period from day 50. At the end of the first sub-period of 20 days, it can be seen that the number of infected stops decreasing, stabilizes, and is about to rise again. It is fortunately stopped by the second quarantine sub-period, which drastically reduces the number. The third sub-period seems almost superfluous. By the end of this one, the number of infected has been drastically reduced, falling below the initial numbers in the simulation. It can then be seen that this number rises little by little, but it takes more than about a year to reach values similar to those present before the first sub-period. Although it is not known what peak may be reached later on, it can be assumed that it will not exceed that obtained without quarantines.

The months gained will surely be useful in the search for a cure if there is none, or for the distribution of the vaccine if it is present. Indeed, as it has been described above, the purpose of these restrictive regulations is primarily to slow down and control the disease, not to eradicate it.

Fig.5.12 shows the death data for the same simulations. It can be seen that from day 50 onwards, the line for the simulation without quarantines plunges and reaches high values, while the curve with quarantines remains much lower and the one with split quarantines tapers to almost flat. For a clearer view of the results, the values obtained are shown in Table 5.5.

Type of simulation	Max prevalence	Final deaths
No quarantine	2436	40
One quarantine	1611	32
Splitted quarantine	991	11

Table 5.5: Different result varying the quarantine.

5.2.4 Validation against SPP

The SPP parameter, presented in section 3.5, stands for Same Province Percentage, with $0 \leq SPP \leq 1$. It represents the percentage of the population whose province of destination is the same as their province of origin. The concept of province is, however, very general. If SPP is set to 0, the presence of provinces is unnecessary as the population moves between them without any restriction. With an $SPP = 1$ there is total isolation between one province and another, and it is as if we were running np separate simulations in parallel, where np is the number of provinces. The split into provinces can be used to divide the same city into neighborhoods, small towns between them, or entire conurbations with

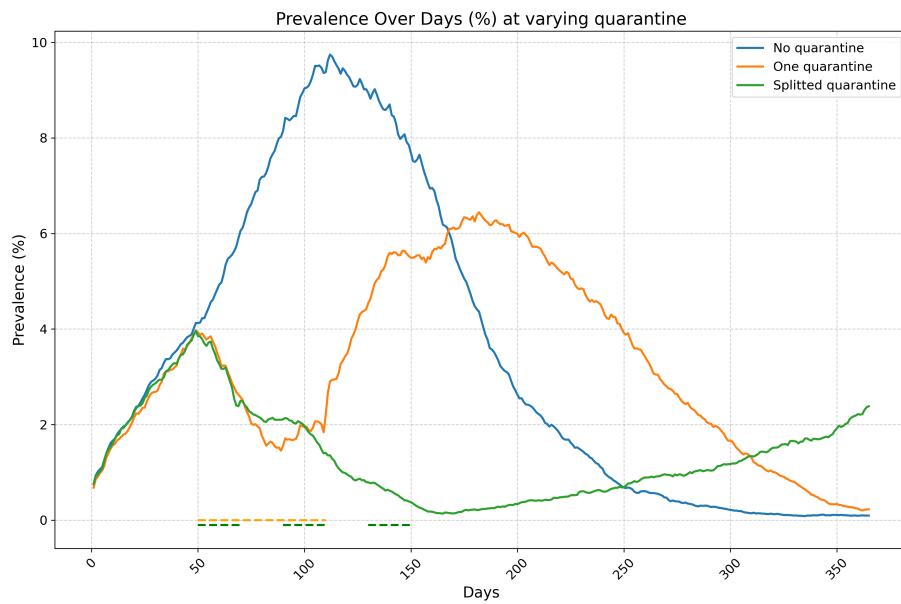


Figure 5.11: Prevalence varying quarantine periods (dotted lines).

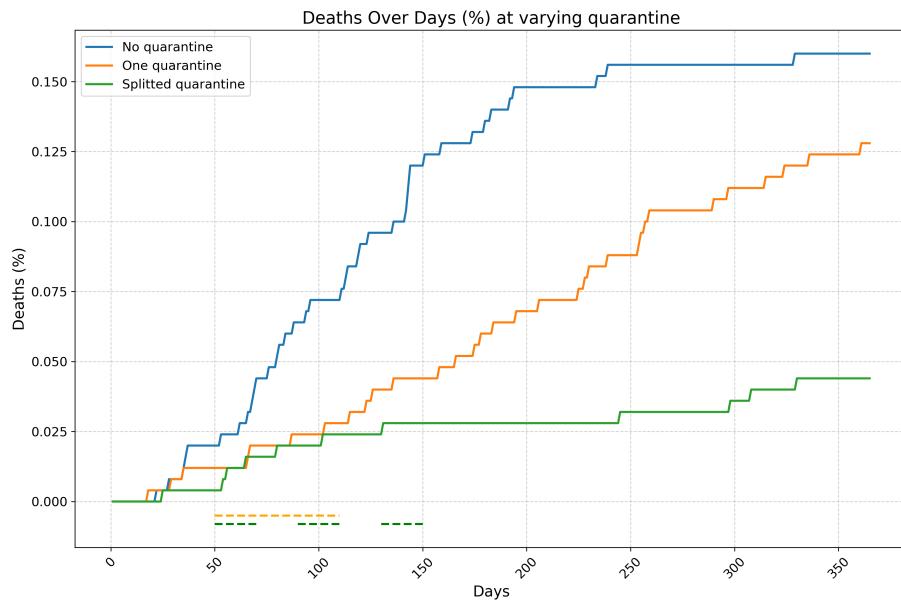


Figure 5.12: Deaths varying quarantine periods (dotted lines).

areas of similar population. The presence of this parameter allows these options to be better modeled since each has a different flow of people passing through these zones.

The effect of this parameter on the simulations is now shown. A graph with five simulations is shown, carried out with a SPP value of 0, 0.5, 0.65, 0.8 and 0.95. Unlike the previous parameters, varying the SPP does not lead to large deviations in the infected curve. In Fig.5.13 it can be seen that as this parameter increases, the number of infected tends to decrease. The strong probabilistic component of the model leads to unexpected variations, especially for parameters that have a moderate impact. This can be seen from the simulation curve with $SPP = 0.8$, which has a higher number of infected than the one with $SPP = 0.65$. The slight increase in the number of infected in the simulations with more freedom of movement results from the crossing of provinces. In order to move between provinces, individuals must take public transport, and this operation is modeled by having to spend an hour in the common area of the destination province.

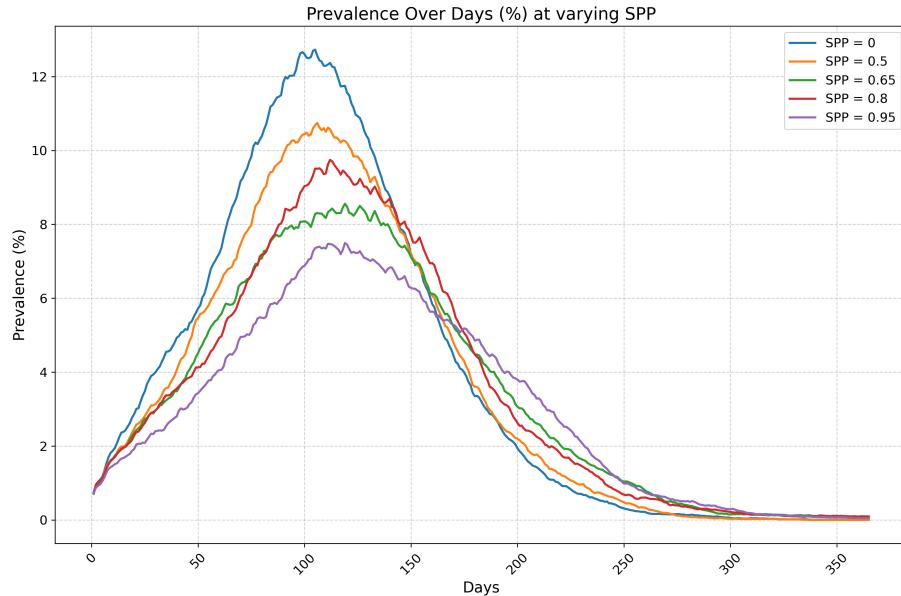


Figure 5.13: Prevalence varying SPP.

5.2.5 Validation of SEJIRS model

The final validation performed concerns epidemiological modeling. A simulation is conducted in which, daily, the number of individuals belonging to the classes of the SEJIRS model, presented in the dedicated section, is tracked. This type of simulation is widely used to validate epidemiological models as it provides

important insights into the movement of individuals between the model's classes. The results can reveal specific and interesting trends in the cardinalities of the classes, such as those described below and shown in Fig. 5.14.

- **Variability and model noise:** The fluctuating pattern is visible when zooming in on the graph in Fig.5.15, especially in E and J_3 curves, revealing the stochastic nature of the model and the presence of random variations. However, when observing the curve in its entirety, these fluctuations fade, and only the overall growth and decline trends remain apparent.
- **Growth and decline trends:** The orange I curve follows an approximately logistic growth phase, peaking around day 120 before rapidly declining. The descent appears steeper than the initial increase, suggesting a strong recovery mechanism. Additionally, the S and R curves behave oppositely, with recovery rates matching infection rates, leading to an intersection point where the population is nearly evenly split between susceptible and recovered individuals. Around day 140, both S and R curves undergo a concavity change, which coincides with the peak of infections. From a mathematical perspective, one could interpret the trend of class I as the derivative of class R , highlighting the direct relationship between infections and recoveries.
- **Long-term stability and equilibrium:** Around day 250, most curves begin to stabilize, approaching an asymptotic phase. This suggests that the system reaches a near-equilibrium state, where transitions between classes slow down significantly. The infected population stabilizes, and the E and $Deaths$ curves flatten, indicating minimal further changes.
- **Cyclic behavior and return to susceptibility:** The model introduces a periodic component, as recovered individuals transition back to the susceptible class after 180 days. This periodicity differs from classic SIR models but aligns with diseases that grant temporary immunity. Over time, the S and R curves mirror each other, with S increasing as R decreases, completing a full cycle. New outbreaks would restart the same pattern, making the system inherently cyclic, except for the number of deaths, which continuously increases.

5.3 Computational complexity

The computational complexity of the model is now discussed. Although there are a large number of parameters, those that have the greatest impact on the average time to run a simulation day are the population and the number of provinces into which it is divided. In the simulations performed, the seconds required to run each day were calculated and added to the CSV file, and these are now shown in the following graphs. The code has been executed on hardware with the following features:

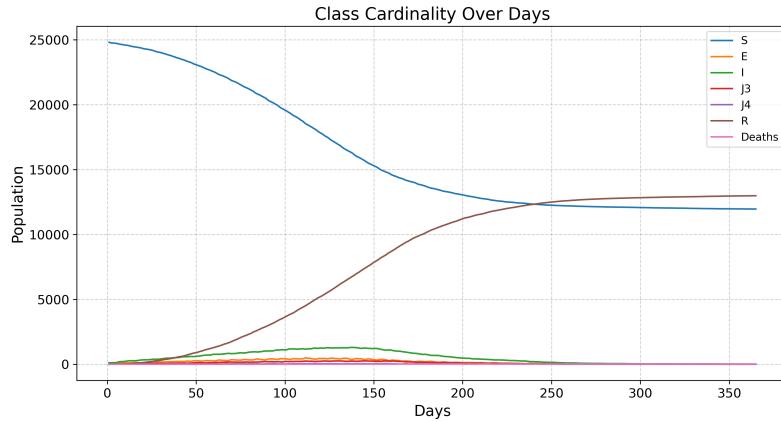


Figure 5.14: Class cardinality on SEJIRS model.

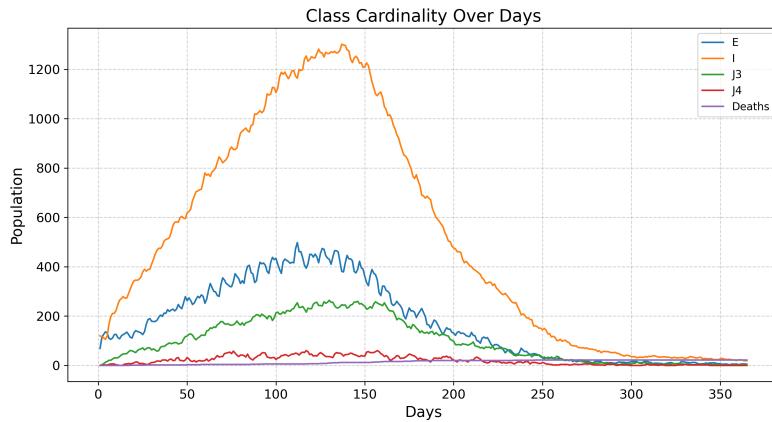


Figure 5.15: A detail of the class cardinality on SEJIRS model.

- CPU: Intel(R) Core(TM) i5-4310M CPU @ 2.70GHz
- RAM: 8 GB
- Operating System: Windows 10 Pro, Version 20H2
- Software Environment: Python 3.12.6, NumPy 2.1.1, Pandas 2.2.3, Matplotlib 3.9.2

The analysis of computation time is divided into two different scenarios, each illustrated by a dedicated graph:

- **Effect of increasing population and provinces proportionally:** The graph in Fig.5.16 shows that as the population increases, the time required to simulate a single day grows linearly. This behavior is consistent with the fact that the number of provinces was increased proportionally to the population, maintaining a constant ratio of approximately 4000 individuals per province. Consequently, the computational workload per province remains unchanged, leading to a proportional increase in simulation time with population size.

The linear trend is confirmed by the regression line, whose slope indicates the growth rate of execution time per additional unit of population. The intercept represents a fixed computational component, likely due to operations independent of the population itself. The strong adherence of the regression line to the experimental data confirms that this linear relationship holds within the considered range.

Additionally, the number of simulated days varies, but its impact on execution time is negligible since the average per-day execution time is used. Therefore, as long as the population-to-province ratio remains unchanged, the time required to process a simulation day follows a linear pattern.

- **Effect of increasing population while keeping the number of provinces constant:** The graph in Fig.5.17 shows the relationship between population size and execution time when the number of provinces remains fixed at 10. Unlike the previous scenario, where the population and the number of provinces scaled proportionally, here the computational workload per province increases as the population grows.

The trend observed follows a linear growth pattern, as confirmed by the regression line. The increase in execution time is directly proportional to the population size, indicating that as the number of individuals per province rises, the computational workload per province increases at a constant rate.

The regression equation highlights this linear relationship, with the slope representing the rate of increase in execution time per additional unit of population. The negative intercept likely represents initial computational overhead, which becomes negligible at larger population values. These results are expected, as the pseudocode in section 4.2 does not contain nested loops iterating over the population, ensuring that execution time scales linearly with population size.

The increase in computation time can be attributed to several factors related to population interactions within the simulation. One key issue is the high number of interactions occurring within the same province. As the population grows, the algorithm struggles to scale efficiently due to internal infection dynamics, hospitalizations, and vaccination processes. However, the primary cause of computational slowdowns is not these individual factors, but rather the

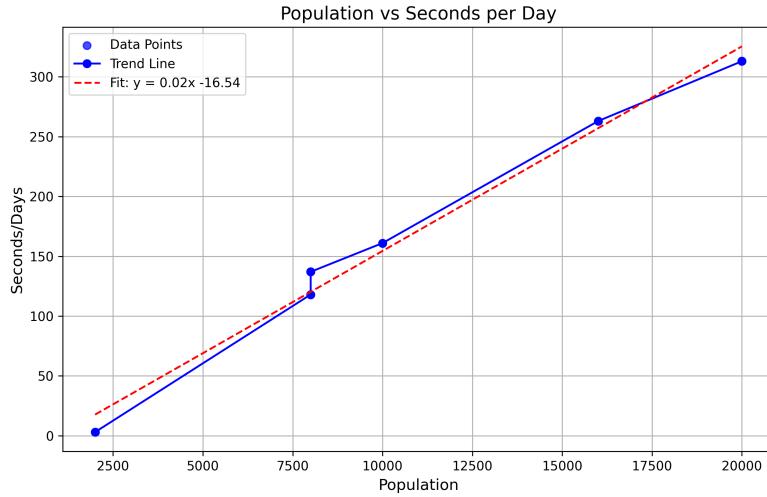


Figure 5.16: Variation of time to run a day as the population and the number of provinces vary.

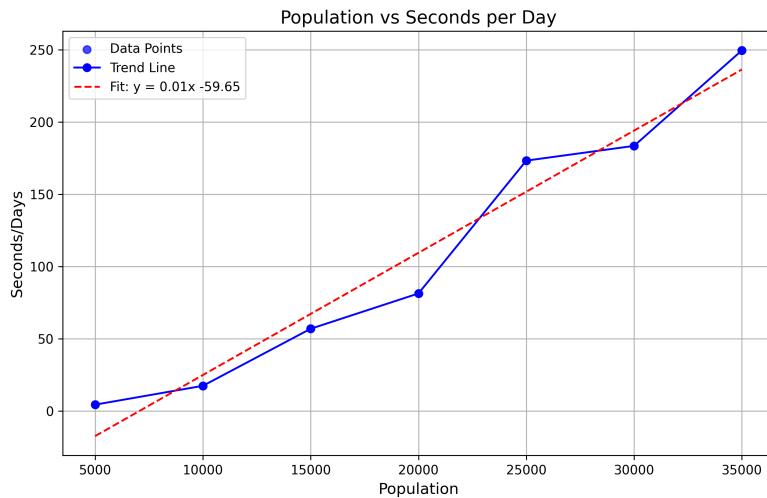


Figure 5.17: Variation of time to run a day as the population varies while the number of provinces (10) remains the same.

interactions between individuals, especially in enclosed environments such as households.

Household infections represent a significant computational challenge. Since

the number of households grows proportionally with the population, each household must evaluate the health status of its members to determine new infections. This process is computationally expensive but essential. Unlike the starting MVT model, which did not account for household-based transmission, excluding this aspect is not a viable option. Individuals spend a considerable portion of their time at home, particularly during quarantine periods, which were not present in the initial model just mentioned. Therefore, household infections must be included in the model to ensure its accuracy. To optimize this process, infections in homes are performed only during the hours when homes are actually occupied and when individuals are awake and interacting. This approach reduces unnecessary computations while preserving the realism of the simulation.

From the study in this section it can be deduced that as long as a good ratio of individuals per province is maintained, there is a linear time growth. If a simulation with a large population value is required, it is recommended to increase the number of provinces. Even if these subdivisions do not reflect a precise geographical division, provinces in the model represent urban areas with a certain degree of isolation and do not strictly define real provinces. This degree can be changed, and the percentage of the population moving between provinces, as described in section 3.5, adjusted as desired. The definition of a province is therefore very flexible and adaptable to the context.

Chapter 6

Conclusions

In this thesis, the use of P systems, based on membrane computing, was investigated in the field of epidemiological research. The first objective of the work was to combine the features of two pre-existing models to obtain one that integrates them both. The result was fully achieved, as the model presented possesses both the ability to simulate individual behavior and a non-trivial complexity in the management of infections.

One of the key advantages of using P systems in this context is their intrinsic ability to perform parallel computations. This characteristic allows for the efficient simulation of large-scale epidemiological models while maintaining flexibility and scalability. The approach adopted makes it possible to incorporate different aspects of infectious disease dynamics, including behavioral responses to changing epidemiological conditions. In addition to providing a detailed formalization, this work introduces the SEJIRS epidemiological model.

Beyond the integration of the two original models, this work also aimed at extending the simulation framework to create a more adaptable model. The goal is to develop a system capable of simulating various infectious diseases by adjusting key parameters and mechanisms. To achieve this level of generalization, numerous features and measures were implemented, enhancing both the flexibility of the simulated scenarios and the range of diseases that can be studied. The scalability of the scenario was improved by allowing the number of provinces to be adjusted, dynamically varying the number of place membranes with population size, and introducing the Same Province Percentage (*SPP*) parameter to control the percentage of individuals moving between different provinces. Additionally, the model supports different infection dynamics, enabling the choice between viral-loaded or fixed-day infections, the inclusion or exclusion of ICU place membranes, quarantine enforcement, and the Prudence Parameter (*PP*), which determines whether symptomatic individuals remain isolated.

These improvements significantly expand the generalization of the model compared to the initial versions, making it a more versatile tool for epidemiological research. This adaptability allows the model to be applied to different epidemiological contexts, providing valuable insights into disease

spread and control strategies. The addition of a graphical interface accessible to all types of users and the easy adaptability of the code consolidate this achievement.

Throughout this process, numerous simulations were carried out to verify that the model produces the expected results under varying conditions. The simulations were specifically designed to test the model's performance using data parameters such as a population of 25000 individuals split between 12 provinces, and a simulation duration of 365 days. Key parameters such as the SPP (which controls the proportion of individuals with the same origin and destination province) and the PP (prudence parameter for symptomatic individuals) were varied. Additionally, quarantine measures were tested with different durations, ICU availability was considered, and active viral load was included. All behaviors described in section 2.3 were implemented to ensure the model captured the full range of dynamics. The parameter tuning process, conducted in the final phase of the thesis, led to significant improvements in the model's performance. After fine-tuning the parameters, the model was rigorously tested using the data outlined in section 5.2, ensuring that it produced reliable results and met the expected outcomes.

Regional databases, particularly from the Lombardy region, were used to refine the model. Once the final version was consolidated, it underwent further testing to evaluate its performance across different aspects. The model's response to changes in the Prudence Parameter confirms the importance of the behavior of individuals. The effect of quarantines was also studied, obtaining a large reduction in the infected ones. This validation is crucial to understanding the effectiveness of the model, as the placement of targeted quarantines could be one of the main uses of the simulator.

The variation in the cardinalities of the SEJIRS classes was also calculated, with particular attention to the variability and stochastic behavior of the model. For instance, when analyzing the fluctuations in the E and J_3 curves, the random variations are evident, reflecting the model's inherent noise. The model also revealed interesting patterns, such as the opposing behaviors of the susceptible (S) and recovered (R) curves. As the simulation progresses, long-term stability is observed, with curves reaching equilibrium around day 250. The system also exhibits cyclic behavior, with recovered individuals transitioning back to the susceptible class after 180 days, completing a full cycle of infection, recovery, and susceptibility.

An analysis of computation times has shown that their growth is directly influenced by two key factors: population size and the number of provinces. The results confirm that execution time follows a linear trend in both scenarios analyzed, aligning with theoretical expectations based on the algorithm's structure. When the population and the number of provinces increase proportionally, maintaining a fixed ratio of individuals per province, the time required to simulate a single day grows linearly. This is due to the fact that the computational workload per province remains unchanged, leading to a proportional increase in execution time as more provinces are introduced. The regression analysis strongly supports this linear relationship, with the slope

indicating the rate of increase in execution time per additional unit of population. Similarly, when the population increases while keeping the number of provinces constant, execution time still exhibits linear growth. This confirms that the computational complexity per province scales proportionally with the number of individuals, without introducing additional exponential factors.

These findings highlight the efficiency of the model in handling large-scale simulations. Furthermore, the model's flexibility enables adaptation to different urban configurations by adjusting the number of provinces, making it suitable for a wide range of applications in population dynamics and epidemiological modeling.

As future developments, we aim to refine the proposed model, by ensuring that it remains robust once generalized to work for other infectious diseases. The combination of computational efficiency, behavioral epidemiology, and model accuracy establishes a strong foundation for further research in this field.

6.1 Future developments

- **Incorporating demographics into the SEJIRS model:** A future enhancement to the model could involve integrating demographic factors, such as births and deaths in the population independent of the disease, into the SEJIRS framework and, consequently, into the simulator. This would allow for a more nuanced simulation of disease transmission, taking into account the entry of new individuals into the susceptible class (S), as well as the potential for individuals to die from any class.
- **Parallelization to improve the computational efficiency:** One important area for improvement is the parallelization of the model. By distributing computations across multiple processors, the model's performance could be significantly enhanced, especially when simulating large populations or running long-term projections.
- **Seasonal variations in behavior:** Introducing seasonal and monthly variations could improve the realism of the model. For instance, different chances of leaving home or getting infected could be adjusted based on the time of year, accounting for factors like weather conditions, holidays, or cultural events that influence human behavior.
- **Modeling household dynamics:** Another development could focus on further differentiating household structures. This could include modeling homes inhabited by singles or groups of young people, as their behaviors and interactions may significantly impact the spread of infectious diseases.
- **Additional place membranes:** The inclusion of more place membranes could enhance the complexity of the model, enabling more precise simulations of interactions between different environments. A good example could be universities: incorporating university environments into the model could provide insights into how infection spreads in academic settings.

Specific contagion parameters and routines for students—such as campus life, dormitory living, and class schedules—could be modeled to better reflect this population’s dynamics.

- **Travel restrictions between provinces:** Another possible extension is the incorporation of travel restrictions between provinces, simulating how limiting movement for certain periods impacts the spread of disease. This could be useful for studying the effects of partial movement restrictions as a more targeted intervention during an outbreak.
- **Optimization and learning methods for further improvement:** Exploring the application of optimization techniques or machine learning algorithms, regardless of the specific type used, could lead to further improvements in the model’s efficiency and accuracy. These methods could help in identifying key parameters that influence disease spread and improve predictions.
- **Model refinement and extension:** The model could be tested with parameters and databases from other infectious diseases to evaluate its adaptability. By applying the SEJIRS framework to diseases beyond the initial scope, the model’s flexibility and utility could be further validated, paving the way for broader epidemiological applications.

Bibliography

- [1] F. Baquero, M. Campos, C. Llorens, and J. M. Sempere, “P systems in the time of covid-19,” *Journal of Membrane Computing*, vol. 3, pp. 246–257, 2021.
- [2] G. Păun, G. Rozenberg, and A. Salomaa, *The Oxford handbook of membrane computing*. Oxford University Press, 2009.
- [3] G. Paun, *Membrane computing: an introduction*. Springer Science & Business Media, 2012.
- [4] M. Campos, J. M. Sempere, J. C. Galán, *et al.*, “Simulating the impact of non-pharmaceutical interventions limiting transmission in covid-19 epidemics using a membrane computing model,” *MicroLife*, vol. 2, uqab011, 2021.
- [5] M. Campos, J. M. Sempere, J. C. Galán, *et al.*, “Simulating the efficacy of vaccines on the epidemiological dynamics of sars-cov-2 in a membrane computing model,” *Microlife*, vol. 3, uqac018, 2022.
- [6] D. Valcamonica, A. D’Onofrio, M. M. Fareed, G. Franco, and C. Zandron, “A dynamic behavior epidemiological model by membrane systems,” Università degli Studi di Milano-Bicocca, Università degli Studi di Verona, Università degli studi di Trieste, 2024.
- [7] S. Verlan, “Using the formal framework for p systems,” in *International Conference on Membrane Computing*, Springer, 2013, pp. 56–79.
- [8] R. Freund and S. Verlan, “A formal framework for static (tissue) p systems,” in *Membrane Computing: 8th International Workshop, WMC 2007 Thessaloniki, Greece, June 25–28, 2007 Revised Selected and Invited Papers 8*, Springer, 2007, pp. 271–284.
- [9] R. Freund, I. Pérez-Hurtado, A. Riscos-Núñez, and S. Verlan, “A formalization of membrane systems with dynamically evolving structures,” *International Journal of Computer Mathematics*, vol. 90, no. 4, pp. 801–815, 2013.
- [10] A. Păun, “On p systems with active membranes,” in *Unconventional Models of Computation, UMC’2K: Proceedings of the Second International Conference on Unconventional Models of Computation, (UMC’2K)*, Springer, 2001, pp. 187–201.

- [11] G. Păun, “Computing with membranes,” *Journal of Computer and System Sciences*, vol. 61, no. 1, pp. 108–143, 2000.
- [12] C. Zandron, C. Ferretti, and G. Mauri, “Solving np-complete problems using p systems with active membranes,” in *Unconventional Models of Computation, UMC’2K: Proceedings of the Second International Conference on Unconventional Models of Computation,(UMC’2K)*, Springer, 2001, pp. 289–301.
- [13] L. Pan and A. Alhazov, “Solving hpp and sat by p systems with active membranes and separation rules,” *Acta Informatica*, vol. 43, pp. 131–145, 2006.
- [14] A. E. R. C. A. H. C. Llorens and J. Sempere, “Enhancing p systems for complex biological simulations,” 2024.
- [15] J. N. Davidson, *William ogilvy kermack, 1898-1970*, 1971.
- [16] A. Franceschetti and A. Pugliese, “Threshold behaviour of a sir epidemic model with age structure and immigration,” *Journal of Mathematical Biology*, vol. 57, no. 1, pp. 1–27, 2008.
- [17] F. Brauer, “Compartmental models in epidemiology,” *Mathematical epidemiology*, pp. 19–79, 2008.
- [18] K. Susvitasari, “Stochastic model of sir epidemic modelling,” *arXiv preprint arXiv:1803.01496*, 2018.
- [19] L. J. Allen and A. M. Burgin, “Comparison of deterministic and stochastic sis and sir models in discrete time,” *Mathematical biosciences*, vol. 163, no. 1, pp. 1–33, 2000.
- [20] C. Vargas-De-León, “On the global stability of sis, sir and sirs epidemic models with standard incidence,” *Chaos, Solitons & Fractals*, vol. 44, no. 12, pp. 1106–1110, 2011.
- [21] N. Bame, S. Bowong, J. Mbang, G. Sallet, and J.-J. Tewa, “Global stability analysis for seis models with n latent classes,” *Mathematical Biosciences & Engineering*, vol. 5, no. 1, pp. 20–33, 2008.
- [22] H. Wan *et al.*, “An seis epidemic model with transport-related infection,” *Journal of theoretical biology*, vol. 247, no. 3, pp. 507–524, 2007.
- [23] O. N. Bjørnstad, K. Shea, M. Krzywinski, and N. Altman, “The seirs model for infectious disease dynamics.,” *Nature methods*, vol. 17, no. 6, pp. 557–559, 2020.
- [24] M. Naim, F. Lahmudi, and A. Namir, “Threshold dynamics of an seis epidemic model with nonlinear incidence rates,” *Differential Equations and Dynamical Systems*, vol. 32, no. 2, pp. 505–518, 2024.
- [25] A. B. Gumel, S. Ruan, T. Day, *et al.*, “Modelling strategies for controlling sars outbreaks,” *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 271, no. 1554, pp. 2223–2232, 2004.

- [26] G. Giordano, F. Blanchini, R. Bruno, *et al.*, “A sidarthe model of covid-19 epidemic in italy,” *arXiv preprint arXiv:2003.09861*, 2020.
- [27] G. Giordano, F. Blanchini, R. Bruno, *et al.*, “Modelling the covid-19 epidemic and implementation of population-wide interventions in italy,” *Nature Medicine*, vol. 26, no. 6, pp. 855–860, 2020.
- [28] M. Higazy, “Novel fractional order sidarthe mathematical model of covid-19 pandemic,” *Chaos, Solitons & Fractals*, vol. 138, p. 110 007, 2020.
- [29] A. N. Akkilic, Z. Sabir, M. A. Z. Raja, and H. Bulut, “Numerical treatment on the new fractional-order sidarthe covid-19 pandemic differential model via neural networks,” *The European Physical Journal Plus*, vol. 137, no. 3, p. 334, 2022.
- [30] O. M. Otunuga and M. O. Ogunsolu, “Qualitative analysis of a stochastic seir epidemic model with multiple stages of infection and treatment,” *Infectious Disease Modelling*, vol. 5, pp. 61–90, 2020.
- [31] J. M. Epstein, “Modelling to contain pandemics,” *Nature*, vol. 460, no. 7256, pp. 687–687, 2009.
- [32] E. Bonabeau, “Agent-based modeling: Methods and techniques for simulating human systems,” *Proceedings of the national academy of sciences*, vol. 99, no. suppl_3, pp. 7280–7287, 2002.
- [33] E. Y. Hills S, “Factors associated with non-adherence to social distancing rules during the covid-19 pandemic: A logistic regression analysis.,” *BMC Public Health*, 2021. DOI: <https://doi.org/10.1186/s12889-021-10379-7>.
- [34] R. Webster, S. Brooks, L. Smith, L. Woodland, S. Wessely, and G. Rubin, “How to improve adherence with quarantine: Rapid review of the evidence,” *Public Health*, vol. 182, pp. 163–169, 2020, ISSN: 0033-3506. DOI: <https://doi.org/10.1016/j.puhe.2020.03.007>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0033350620300718>.
- [35] L. Carlucci, I. D’Ambrosio, and M. Balsamo, “Demographic and attitudinal factors of adherence to quarantine guidelines during covid-19: The italian model,” *Frontiers in Psychology*, vol. 11, 2020, ISSN: 1664-1078. DOI: [10.3389/fpsyg.2020.559288](https://doi.org/10.3389/fpsyg.2020.559288). [Online]. Available: <https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2020.559288>.
- [36] M. D. Knoll and C. Wonodi, “Oxford–astrazeneca covid-19 vaccine efficacy,” *The Lancet*, vol. 397, no. 10269, pp. 72–74, 2021.
- [37] T. Cerqueira-Silva, V. de Araújo Oliveira, V. S. Boaventura, *et al.*, “Influence of age on the effectiveness and duration of protection of vaxzevria and coronavac vaccines: A population-based study,” *The Lancet Regional Health–Americas*, vol. 6, 2022.
- [38] M. D. Hitchings, O. T. Ranzani, M. Dorion, *et al.*, “Effectiveness of the chadox1 vaccine in the elderly during sars-cov-2 gamma variant transmission in brazil,” *MedRxiv*, pp. 2021–07, 2021.

- [39] A. Sheikh, J. McMenamin, B. Taylor, and C. Robertson, “Sars-cov-2 delta voc in scotland: Demographics, risk of hospital admission, and vaccine effectiveness,” *The Lancet*, vol. 397, no. 10293, pp. 2461–2462, 2021.
- [40] I. Pérez-Hurtado, D. Orellana-Martín, G. Zhang, and M. J. Pérez-Jiménez, “P-lingua in two steps: Flexibility and efficiency,” *Journal of Membrane Computing*, vol. 1, pp. 93–102, 2019.
- [41] V. Manca, A. Castellini, G. Franco, L. Marchetti, and R. Pagliarini, “Metabolic p systems: A discrete model for biological dynamics,” *Chinese Journal of Electronics*, vol. 22, no. 4, pp. 717–723, 2013.
- [42] G. Păun, “Introduction to membrane computing,” in *Applications of Membrane Computing*, G. Ciobanu, G. Păun, and M. J. Pérez-Jiménez, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–42, ISBN: 978-3-540-29937-0. DOI: 10.1007/3-540-29937-8_1.
- [43] S. Erba, <https://github.com/SandroErba/MVT>, 2025.
- [44] F. Mowbray, L. Woodland, L. E. Smith, R. Amlôt, and G. J. Rubin, “Is my cough a cold or covid? a qualitative study of covid-19 symptom recognition and attitudes toward testing in the uk,” *Frontiers in public health*, vol. 9, p. 716 421, 2021.
- [45] <https://github.com/pcm-dpc/COVID-19/tree/master/dati-regioni>.
- [46] R. G. Remuzzi Andrea, “Covid-19 and italy: What next?” *The Lancet*, 2020.
- [47] C. L. Rovetta A Bhagavathula AS, “Modeling the epidemiological trend and behavior of covid-19 in italy,” *Cureus*, 2020.
- [48] A. Lombardi, N. Amoroso, A. Monaco, S. Tangaro, and R. Bellotti, “Complex network modelling of origin–destination commuting flows for the covid-19 epidemic spread analysis in italian lombardy region,” *Applied Sciences*, vol. 11, no. 10, 2021, ISSN: 2076-3417. DOI: 10.3390/app11104381. [Online]. Available: <https://www.mdpi.com/2076-3417/11/10/4381>.
- [49] D. Cereda, M. Tirani, F. Rovida, *et al.*, *The early phase of the covid-19 outbreak in lombardy, italy*, 2020. arXiv: 2003.09320 [q-bio.PE]. [Online]. Available: <https://arxiv.org/abs/2003.09320>.