STUDYING THE SPREAD OF DISEASES USING GEOGRAPHICAL DATA AND IRREGULAR TOPOLOGIES WITH CELL-DEVS

Román Cárdenas

Laboratorio de Sistemas Integrados
Universidad Politécnica de Madrid
ETSI Telecomunicación, Avda. Complutense 30
Madrid 28040, SPAIN
r.cardenas@upm.es

Cristina Ruiz Martin
Gabriel Wainer

Dept. Of Systems and Computer Engineering
Carleton University
1125 Colonel By Drive
Ottawa ON K1S 5B6, CANADA
{cristinaruizmartin, gwainer}@sce.carleton.ca

Peter Dobias
Mark Rempel

Defence Research and Development Canada
Centre for Operational Research and Analysis
60 Moodie Drive
Ottawa, ON, K1A0K2, CANADA
{peter.dobias, mark.rempel}@drdc-rddc.gc.ca

ABSTRACT

Modeling and Simulation (M&S) techniques have been proven to be effective to understand how diseases spread and assess the effectiveness of decisions aimed to control them (e.g., mobility restrictions). Recently, governments used this approach to determine the evolution of the COVID-19 pandemic. In this context, M&S tools that consider geographical information can improve the quality of the simulations. This research presents a methodology that allows modelers to prototype disease spread models that include geographical information. The model can be easily parameterized for other geographical regions and diseases. We present a case study of a disease spread model to show how this methodology works.

Keywords: Cell-DEVS, Geographical Model, Irregular Topologies, Disease Spread.

1 INTRODUCTION

Understanding how a disease is spread is critical to make decisions aimed to control it. There are many factors that affect how a disease spreads, such as the capacity to isolate symptomatic individuals and how close contacts are traced and quarantined (Fraser, et al. 2004). These two factors can be controlled by decision makers, but there are others, such as the behavior of individuals, that cannot be fully controlled. For example, the COVID-19 pandemic has revealed that the effectiveness of some measures that would work in an ideal scenario is reduced due to behavioral factors. When dealing with lockdown measures, some studies revealed that repression did not play a significant role in compliance. Instead, moral, and social motivations produced better results (Kuiper, et al. 2020).

ANNSIM '21, July 19-22, 2021, Fairfax, VA, USA;

The intrinsic complexity of human behavior and the large number of factors affecting the spread of a disease makes simulation a useful tool to study these problems. We can simulate how a specific decision works in different scenarios, for instance, quarantines and lockdowns with different compliance rates. With such analysis we could determine the minimum compliance rate to have a given outcome.

Recently, governments have been using this kind of analysis to determine the evolution of the COVID-19 pandemic and decide which measures should be in place. For example, Currie, et al. (2020) highlight the importance of simulation for making decisions that (1) affect disease transmission, (2) determine how resources should be allocated, and (3) influence the priorities for care.

When developing simulation models in the context of spread of diseases, geographical information (such as population demographics, borders, or transportation infrastructure among territories) can improve the quality of the results obtained. For example, Zhong, Huang and Song (2009) introduced a geographical Cellular Automata (CA) corresponding to a Susceptible-Infected-Recovered-Susceptible (SIRS) model with multiple infection phases. The cell space represents irregular shapes with varying dimensions corresponding to a geographical location. The length of the boundaries with the adjacent regions, and road links between regions were used to determine the neighborhood. Carlucci, D'Ambrosio and Balsamo (2020) conducted a survey to analyze the effect of sociodemographic variables such as age, education, marital status, geographic area and region, employment status, and annual income and their effect in compliance with quarantine measures in Italy. They found out that women exhibited significantly higher levels of adherence to quarantine guidelines than men. There were also significant differences based on age groups. The 18-29 age group had significantly lower adherence to the rules than the other groups.

In most cases, these models are developed using ad-hoc techniques, and they are difficult to apply to other geographical areas. Here, we present a methodological approach that allows modelers to define disease spread models that include geographical information that can be easily parameterized for other geographical regions. This approach is focused on providing rapid prototyping for model development. We define a method based on Cell-DEVS specifications with irregular neighborhoods. We present a case study of a disease spread model based on the model introduced by Zhong, Huang and Song (2009) to show how the methodology works.

The rest of the paper is organized as follows. In section 2, we present related work on the study of the spread of diseases and how geographical information has been considered in the literature. We also introduce the formalism used in this research: cell-DEVS. In section 3, we define the model. In section 4, we focus on the implementation details relevant to understand the proposed methodological approach. Section 5 focuses on the analysis of some simulation results. Finally, in section 6, we present conclusions and future work.

2 BACKGROUND

There are numerous models to study the spread of diseases based on the foundational research by Kermack and McKendrick (1927), who introduced a model that classifies the population into three different categories: (1) Susceptible – i.e., individuals who can get infected with the disease, (2) Infected – those who have contracted the disease, and (3) Recovered – people that recovered from the disease. The success of this SIR model has led to several improvements and adaptations to study the spread of different diseases. For example, Li and Muldowney (1995) include a fourth category: Exposed – individuals who do not transmit the disease, but which eventually become infected. To model fatalities, different authors (for instance, Matadi (2014)) include a category for those who die because of the disease.

Different combinations of these categories exist, such as SEIR, SIRS, SIRD, or SEIRDS. Models based on these categories calculate the number of individuals in the different categories using models described with differential equations. Many of these models have been applied to study the spread COVID-19. For example, Caccavo (2020) adapted a SIRD model that adequately describes the outbreaks in China and Italy. However, these models rely on the law of large numbers and cannot capture interactions between groups across time and space.

Other approaches, such as metapopulation (Ajelli, et al. 2010, Chen, et al. 2019) and cellular models, have adapted these mathematical models to integrate the spatial aspects of the spread of the disease. For example, Bin, Sun, and Chen (2019) used population density, gender, and age structure to set the evolution rules of a SEIRDS the model using CA. They applied their model to study the spread of influenza A (H1N1). Fuentes and Kuperman (1999) also used CA combined with SIR models to study the effect on immunity in the spread of a disease. Recently, Cárdenas, et al. (2020) replicated the results of CA models using Cell-DEVS, focusing on the advantages of Cell-DEVS over CA for disease spread models.

Some models include spatial aspects in a more realistic way. For example, Zhong, Huang and Song (2009). presented a geographical CA corresponding to a SIRS model with multiple infection phases. Each infection phase has associated a different probability of spreading the virus and a different recovery rate. In this model, each cell represents a geographical location. They have irregular shapes and different dimensions based on the geographical location they represent. The neighborhood of a specific cell is defined based on the length of the boundaries with the adjacent regions, and road links between them.

In many cases, the geographical characteristics of these models are specific for a region. These are difficult to apply to other regions We present a method to develop rapid prototypes of geographical models for the spread of diseases. These models that can be easily adapted to other geographical regions by just modifying the parameters of the model (i.e., without changing their implementation). In that way, the modeler does not need to go over the verification process every time the model is adapted to a different geographical area.

2.1 The Cell-DEVS Formalism

The Cell-DEVS formalism (Wainer 2017) allows defining n-dimensional cell spaces in which each cell is an independent discrete-event model based on the Discrete EVent System Specification (DEVS) (Zeigler, Praehofer and Kim 2000). Cell-DEVS divides the space of the system into a collection of cells. Each cell's next state depends on its current state and input events received from neighboring cells or any other DEVS model. Figure 1 shows a cell and a 2-dimensional Cell-DEVS model.

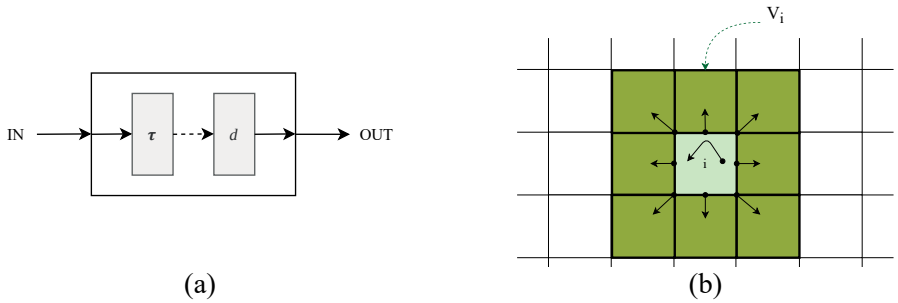


Figure 1. Cell-DEVS model schematic.

As shown in. Figure 1(a), when cells receive any input event, they trigger their local computing function τ to compute their corresponding next state. Cell state changes are transmitted as output events after a time delay specified by a delay function *d*. State changes are sent to neighboring cells. Figure 1(b) depicts an illustrative example in which the neighborhood set of cell *i*, V_i, corresponds to a Moore neighborhood (Gray 2003). Cell-DEVS allows cells to interact with other DEVS models, tools, data sets, and visualization mechanisms, making it easy and efficient to build complex cellular models. Although most Cell-DEVS models represent the space as a regular lattice of cells, the formalism allows defining irregular neighborhoods for each cell.

In this research, we use the Cadmium simulator (Belloli, et al. 2019). Cadmium is a C++17 header-only DEVS M&S library. Cadmium follows a strong-typed approach to ensure that all the DEVS models comply with the DEVS formalism before simulating them. It also supports the description of Cell-DEVS models.

As far as our knowledge, Cadmium is the first simulation engine that enables the definition of Cell-DEVS models with irregular topologies and complex neighborhood sets.

3 MODEL DEFINITION

In this section, we present a space-based pandemic model for scenarios with irregular topologies based on geographical data using the Cell-DEVS formalism. The model is based on previous work proposed by Zhong, Huang, and Song (2009), and it is an extension of an early version (Cárdenas, et al. 2020). We adapted the model to support irregular geographical topologies, and we include additional compartments and configuration parameters to model the pandemic.

In our model, each cell represents a region of the scenario space. For example, if the scenario corresponds to a city, each cell could represent a health unit. Eq. (1) shows the tuple that represents the state of cell *i*:

$$\theta_i = \{P_i, S_i, I_{(i,1)}, \cdots, I_{(i,T_I)}, R_{(i,1)}, \cdots, R_{(i,T_R)}, D_i\} \tag{1}$$

P_i is the total population of cell *i*. S_i is the population ratio that is susceptible to the disease. $I_{(i,x)}$ represents the ratio that has been infected during *x* consecutive days. A person can be infected for up to T_I consecutive days. $R_{(i,x)}$ corresponds to the ratio of individuals that recovered and have been immune to the disease during *x* consecutive days. After T_R days being immune, recovered people become susceptible again. Finally, $D_i$represents the people that died because of the pandemic. The sum of all the compartments that comprise the cell state must be equal to 1, as stated in Eq. (2):

$$S_i + \sum_{n=1}^{T_I} I_{(i,n)} + \sum_{n=1}^{T_R} R_{(i,n)} + D_i = 1 \tag{2}$$

The new state of cell *i*, $\theta'_i = \{P_i, S'_i, I'_{(i,1)}, \cdots, I'_{(i,T_I)}, R'_{(i,1)}, \cdots, R'_{(i,T_R)}, D'_i\}$, is computed every time the local computing function is triggered. Figure 2 shows how people transition from one compartment to another when the cell state changes.

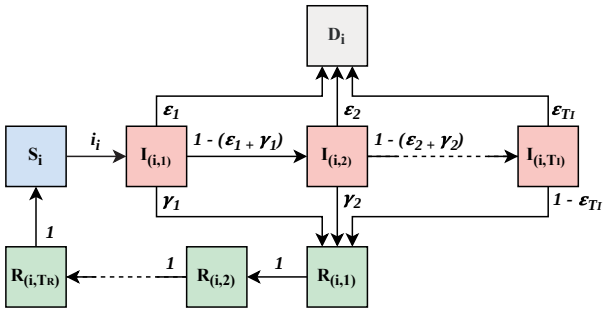


Figure 2. Transitions between epidemiological compartments of cell *i*.

Eq. (3) describes how the compartments ratio changes every time the local computing function is triggered.

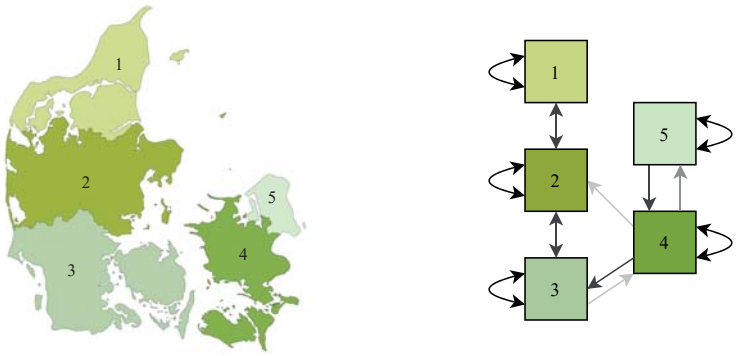
$$\begin{aligned}
S'_i &= S_i \cdot (1 - i_i) + R_{(i,T_R)} \\
I'_{(i,x)} &= \begin{cases} S_i \cdot i_i & , \text{if } x = 1 \\ I_{(i,x-1)} \cdot \left(1 - (\varepsilon_{x-1} + \gamma_{x-1})\right) & , \text{if } 1 < x \leq T_I \end{cases} \\
R'_{(i,x)} &= \begin{cases} I_{(i,T_I)} \cdot \left(1 - \varepsilon_{T_I}\right) + \sum_{n=1}^{T_I - 1} I_{(i,n)} \cdot \gamma_n & , \text{if } x = 1 \\ R_{(i,x-1)} & , \text{if } 1 < x \leq T_R \end{cases} \\
D'_i &= \sum_{n=1}^{T_I} I_{(i,n)} \cdot \varepsilon_n
\end{aligned} \tag{3}$$

The terms γ_x and ε_x correspond to the *recovery* and *mortality factors* (i.e., the probability of a person that has been infected during x consecutive days recovering and dying, respectively). Also, i_i represents the ratio of susceptible people that got infected since the last time the local computing function was triggered:

$$i_i = \sigma \cdot min\left\{1, \sum_{j \in V_i}\left(\rho_{i,j} \cdot \frac{P_j}{P_i} \cdot \sum_{n=1}^{T_I} \upsilon_n \cdot I_{(j,n)}\right)\right\} \quad (4)$$

where σ is the *susceptibility factor* (i.e., the probability of a susceptible individual getting infected if exposed to the illness), υ_x is the *virulence factor* (i.e., the probability of a person that has been infected during x consecutive days exposing others to the disease), and $\rho_{i,j}$ corresponds to the *correlation factor* (i.e., the degree of influence of cell j over the state of cell *i*).

The correlation factor between cells allows considering the effect of some cells over others differently. The way we compute the correlation factor is a configuration parameter of the presented model. For instance, when working with irregular topologies in which each cell represents a geographical region, we can compute $\rho_{i,j}$ as the length of the shared border divided by the perimeter of the region represented by cell *i*. Alternatively, if we have enough data about the area of interest, $\rho_{i,j}$ could depend on the average mobility of people from cell *j* to cell *i*. Figure 3 illustrates how geographical data can be processed to generate a Cell-DEVS model that supports irregular topologies. For each geographical region in Figure 3(a), a cell with the same identification number is created in Figure 3(b). The arrows between cells represent the cell neighborhoods. The color of the arrows interconnecting cells indicates the degree correlation factor.



(a) (b)

Figure 3. Irregular Cell-DEVS schematic.

Cell *i* is usually a neighbor of itself, and its autocorrelation factor, $\rho_{i,i}$, is usually close to 1 (in Figure 3(b), these autocorrelation arrows are black). Correlation factors can by directed (i.e., $\rho_{i,j} \neq \rho_{j,i}$, if $i \neq j$) For example, $\rho_{3,4}$ is high, as people of region 4 usually move to region 3. However, $\rho_{4,3}$ is close to 0 (in the figure, the arrow's color is light gray). It is also possible that cell *i* belongs to the neighborhood of cell *j*, but cell *i* does not consider cell *j* as a neighbor (this is the case between cells 2 and 4 in Figure 3(b)).

The versatility of the proposed model allows exploring the effect of different countermeasures to reduce the impact of the pandemic under study. For instance, when studying a disease that presents airborne transmission (e.g., COVID-19), it is easy to develop a prototype that modifies the virulence and/or susceptibility of the individuals to emulate the use of masks. Additionally, correlation factors between neighboring cells can be modified to emulate mobility restrictions from one cell to another.

4 BUILDING A MODEL WITH GEOGRAPHICAL DATA

This section illustrates how to define a simulation scenario from geographical data. The framework is publicly available in our GitHub repository, in which we enumerate all the dependencies required to compile the application, as well as instructions to run simulations.

Let us assume that we want to simulate a pandemic scenario over the geographical data presented in Figure 4. Each cell of the scenario must be configured according to the available geographical data and the data that characterizes the spread of the disease under study.

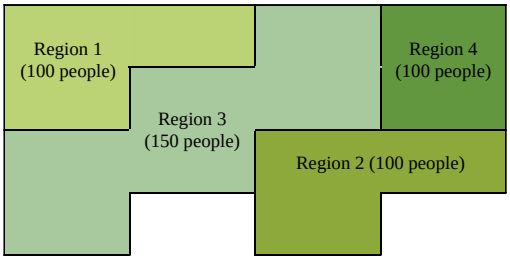


Figure 4. Example of geographical data for simulating the spread of a disease.

From the geographical data, we must obtain two configuration parameters: *a)* the population of each cell (depicted in the figure), and *b)* the correlation factor between each cell. In this example, we define the correlation factor $\rho_{i,j}$ as the length of the border shared between regions *i* and *j* divided by the perimeter of cell *i*. However, it is possible to define different approaches to compute the correlation between cells (e.g., the percentage of people that moves daily from cell *i* to cell *j*). Table 1 shows all the correlation factors in this scenario. Rows correspond to the cell *i* of the correlation, whereas columns represent the cell *j*.

Table 1. Correlation factor of cells in the scenario.

	Region 1	**Region 2**	**Region 3**	**Region 4**
Region 1	1	0	0.5	0
Region 2	0	1	0.25	0.167
Region 3	0.3	0.15	1	0.1
Region 4	0	0.25	0.25	1

The rest of the parameters of the model depend on the disease under study. Table 2 shows all the parameters used to characterize the disease in our case study. The table shows the Virulence, Recovery, and Mortality factors for each population group. In this example, the population is divided in 5 groups, all of them with the same factors. The population is divided based on age, level of education or income among other factors.

Table 2. Configuration parameters of the disease.

Configuration Parameter	**Value**
Maximum days of infection T_I	5
Maximum days of recovery T_R	10
Susceptibility factor σ	1
Virulence factor υ_x	$0.05, \forall x \in \{1, \cdots, 5\}$
Recovery factor γ_x	$0.1, \forall x \in \{1, \cdots, 5\}$
Mortality factor ε_x	$0.05, \forall x \in \{1, \cdots, 5\}$

The framework allows rapid prototyping of multiple pandemic scenarios. To do so, it generates any scenario using a JavaScript Object Notation (JSON) configuration file. Once the models comprising the study are

completed and they are compiled, the simulation framework can run multiple scenarios using the external JSON configuration file. Code 1 shows this JSON object for our case study.

Code 1. Description of default cell configuration.

```
"default": {
    "state": {
        "population": 100,
        "susceptible": 1,
        "infected": [0, 0, 0, 0, 0],
        "recovered": [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        "deceased": 0
    },
    "config": {
        "susceptibility": 1,
        "virulence": [0.05, 0.05, 0.05, 0.05, 0.05],
        "recovery": [0.1, 0.1, 0.1, 0.1, 0.1],
        "mortality": [0.05, 0.05, 0.05, 0.05, 0.05],
    }
}
```

The file must contain a `"default"` JSON object that describes the default configuration of all the cells present in the scenario. Here, the default configuration contains all the parameters shown in Table 2 and the most recurring initial state of all the cells in the scenario (i.e., 100 inhabitants, all of them susceptible to the disease). Additionally, the configuration file must contain a JSON object for every cell. The object corresponding to each cell is treated as a JSON patch (Internet Engineering Task Force 2013). Thus, the JSON object must contain only those fields that differ from the default configuration. Code 2 contains all the configuration patches for the cells in the example.

Code 2. Region-specific configuration patches.

```
"region_1": {
    "state": {
        "susceptible": 0.99,
        "infected": [0.01, 0, 0, 0, 0]
    },
    "neighborhood": {"region_1": 1, "region_3": 0.5}
},
"region_2": {
   neighborhood": {"region_2": 1, "region_3": 0.25, "region_4": 0.167}
   },
"region_3": {
   "state": {"population": 150},
    "neighborhood": {"region_1": 0.334, "region_2": 0.15,
                     "region_3": 1, "region_4": 0.1}
   },
"region_4": {
    "neighborhood": {"region_2": 0.25, "region_3": 0.25, "region_4": 1}
}
```

Regions 2 and 4 have the default configuration. Thus, we only must indicate their neighborhood. The neighborhood is described in the `"neighborhood"` object. For each neighboring region j, we indicate the correlation factor $\rho_{i,j}$. In case of the region 3, we only must modify the population of the region, as this region has 150 inhabitants. Inside the `"neighborhood"` object, we only have to overwrite the `"population"` value. Finally, let us assume that, initially, 1% of the population of region 1 are infected. In this case, we also must overwrite the `"susceptible"` and `"infected"` parameters of the region corresponding to this region, as they do not coincide with the default configuration.

Figure 5 shows how to conduct experiments with this framework. A Cell-DEVS generator reads from a geographical data source (for instance, a file defined in GeoJSON (Internet Engineering Task Force 2016)) and builds the corresponding scenario configuration file. The Cell-DEVS generator identifies which regions conform the scenario, obtain their population, and computes the correlation factor between them. The function that computes the correlation factor can be redefined to explore different neighborhood configurations. Once the scenario JSON configuration file is generated, the epidemic simulator presented earlier executes the scenario and returns the simulation results, which can be analyzed.

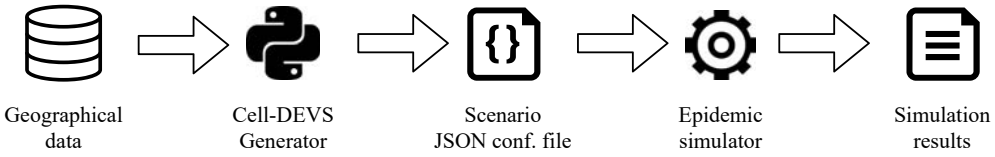


Figure 5. Pandemic simulator workflow.

5 SIMULATING COVID SCENARIOS

This section shows different simulations conducted on the model presented in Section 4. We first introduce a base case study scenario using the parameters in Table 2, in which we use the population age segments presented earlier. This base scenario was used to show the effect of different control measures in the spread of the disease. We also show how to modify the parameters of the model if the variables that define the behavior of the disease (i.e., virulence, recovery, and mortality factors) change.

Figure 6 shows the simulation results for our base scenario, built using a Jupyter notebook application available for studying the results of the simulation.

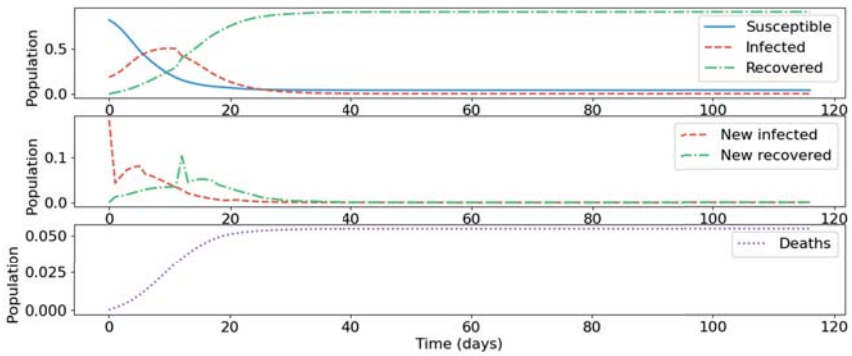


Figure 6. Base simulation Scenario

The top part of the figure represents the percentage of Susceptible (blue line), Infected (red line) and Recovered (green line) population over time. The middle diagram shows the percentage of newly infected (red line) and newly recovered individuals (green line) at a given time. Finally, the bottom part represents the number of peopled that died due to the disease. Every figure presented in this section follows this same structure. We can see that the population becomes infected quickly (red line in the top diagram) and that new infections decrease after a couple of days. The highest percentage of newly infected individuals is detected is at the beginning of the simulation (red line in the middle diagram), which is causing the new recovery peak between day 10 and 12 (middle diagram, green line). At the end of the pandemic, approximately 5% of the population died (bottom of the figure).

This basic case study shows how to analyze different scenarios by simply modifying the parameters in the JSON file. For example, we can study the effect of a disease with a higher mortality rate in the elderly population (e.g., 0.1 instead of 0.05). To do this, we simply modify the mortality rate for the elder age group

in Code 1, defining mortality as "`"mortality": [0.05, 0.05, 0.05, 0.05, 0.1]`" and using this file to run the model again. The remaining parameters remain unchanged.

The simulation results for this scenario are shown in Figure 7. Compared to the base scenario, fatalities (bottom of the figure) have increased from 5% to 15%. The curve for infected (top of the figure, red line), susceptible (top of the figure, blue line), and newly infected (middle of the figure, red line) remains the same. The parameter we modified did not affect these results. However, the percentage of recovered individuals (top of the figure, green line) and new recovered individuals (middle of the figure, green line) has slightly decreased. This has occurred because the remaining population either recovers or dies.

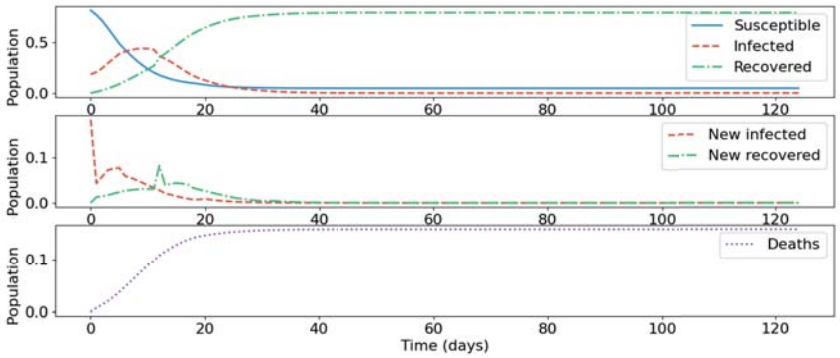


Figure 7. Simulation scenario: mortality is higher for elderly people.

We can also study the effect of a change in the spread of the disease. In this case, we can study how the pandemic will evolve if there are new variants of the virus that are more transmissible (i.e., the virulence factor is higher). To do this analysis, we just need to modify the "virulence" parameters in the JSON file. In this case, we set the virulence 5 times higher than the original value: `"virulence": [0.25, 0.25, 0.25, 0.25, 0.25]` (Code 1). The remaining parameters do not change.

In Figure 8, we present the simulation results for this new scenario. In this case, we can see that the population gets infected earlier (top of the figure, red line).

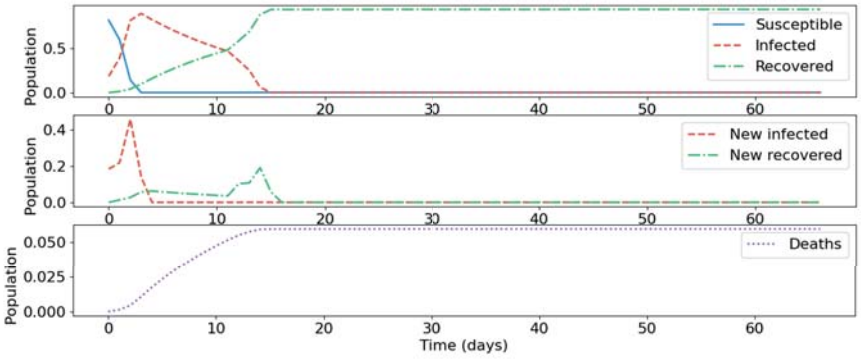


Figure 8. Simulation scenario: higher virulence.

Everyone gets infected in 15 days compared with the over 20 days in the base scenario. Additionally, the infection peak (i.e., the amount of people infected at the same time) is also higher (top of the figure, red line). The infection peak is an important measure as it is directly related with the number of people that may require hospital care. At the bottom of the figure, we can appreciate that the percentage of deaths remains the same because the mortality factor has not change and we are not considering the effect of having hospitals overloaded.

In this scenario, we expect that a reduction of the number of contacts per people will reduce the transmission of the disease. This can be modeled including a new parameter to our model: the mobility factor. If we decide to introduce a new parameter that was not originally available in the JSON configuration file, we need to modify the implementation of our model. In this case, we modify the implementation of the model to be able to study the effects of a lockdown, a measure that governments are taking to control the spread of diseases. To do so, we introduce two new parameters: (1) the percentage of infected population that triggers the lockdown, and (2) the reduction on the percentage of infected population that ends the lockdown. To replicate the results in the base scenario (Figure 6), we need to set the first parameter to 100%. The value of the second parameter is not relevant to be able to replicate the results because the lockdown will never be triggered.

In Figure 9, we show the results of this new model. We configure the model with a value of 20% for the percentage of infected population that triggers the lockdown and 10% for the reduction on the percentage of infected population that ends the lockdown. In this study, we assume an ideal scenario where there is no disobedient population. At the top part of the figure, we can see that the percentage of infected population (red line) is always under 20%. When the lockdown is in effect, the number of infections decreases. In this scenario, the lockdown is long enough to extinguish the disease. There is a high percentage of the population that did not get infected (top of the figure, blue line). Even though the mortality rate remains the same, the number of deaths (bottom of the figure) decreases because less people had the disease.

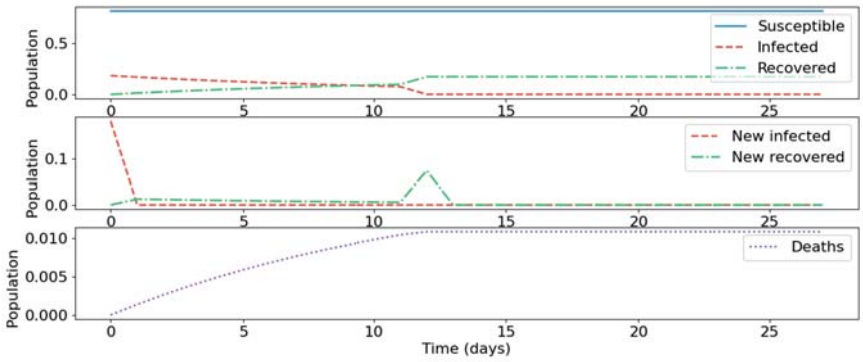


Figure 9. Simulation scenario: lockdown with no disobedient population.

The results presented Figure 9 can be modified by including an additional parameter: the percentage of disobedient population. With this parameter, we can study different scenarios with different disobedient rates and adjust the threshold to trigger the lockdown to get the expected outcome of the lockdown measure. All these studies can be done with the version of the model available in our GitHub repository.

6 CONCLUSIONS AND FUTURE WORK

In this research, we have presented a methodology based on Cell-DEVS with irregular topologies for rapid prototyping of new rules for the spread of communicable disease based on SIR models built as cell spaces. In our models, each cell represents a geographical region. To implement the model that exemplifies our methodology, we used Cadmium simulator. We used Cadmium because it is the only Cell-DEVS simulator engine that allows to implement Cell-DEVS models with irregular topologies.

The models we developed are parameterized and they can easily be calibrated for different geographical regions. All the model parameters are defined in an external JSON file. Therefore, once the model is implemented and compiled, we can study different scenarios without modifying the implementation. We just need to modify the parameters in the JSON file. It is important to remark that one of these parameters is the neighborhood of each cell. Therefore, we can easily model different geographical areas.

Future research will focus on adding individual interactions within the population of a cell. The future work will also be focused on calibrating and validating the model using real data from Canada obtained during the COVID-19 pandemic. To calibrate the model, we can use different values for the parameters defined in the JSON file. It is important to remark that, for this analysis, the implementation does not need to be modified. If the model ends up being not valid, we will modify the equations that defines the model, and we will reimplement the equations in our cell. The advantage of Cell-DEVS and our implementation in Cadmium is that changing the equations in the model is an easy task.

ACKNOWLEDGEMENTS

This project has been funded by NSERC Alliance research grants. We would like to acknowledge Binyamin Brion who worked on model implementation, and Kevin Henares for implementing the Python script to transform the geographical data into input files for the studies.

REFERENCES

Ajelli, M., B. Gonçalves, D. Balcan, V. Colizza, H. Hu, J. J. Ramasco, S. Merler, and A. Vespignani 2010. "Comparing Large-Scale Computational Approaches to Epidemic Modeling: Agent-Based versus Structured Metapopulation Models". *BMC Infect Dis* vol. 10.

Belloli, L., D. Vicino, C. Ruiz-Martin, and G. Wainer 2019. "Building DEVS Models with the Cadmium tool". In *Proceedings of the Winter Simulation Conference*, edited by N. Mustafee, M. Rabe, KH. G. Bae, C. Szabo, and S. Lazarova-molnar, pp. 45–59. National Harbor, MD, IEEE.

Bin, S., G. Sun, and C.C. Chen 2019. "Spread of Infectious Disease Modeling and Analysis of Different Factors on Spread of Infectious Disease Based on Cellular Automata". *International Journal of Environmental Research and Public Health* vol. 16 (23), pp. 4683.

Caccavo, D. 2020. "Chinese and Italian COVID-19 Outbreaks can be Correctly Described by a Modified SIRD Model". *medRxiv.*

Cárdenas, R., K. Henares, C. Ruiz-Martín, and G. Wainer 2020. "Cell-DEVS Models for the Spread of COVID-19". In *14th International Conference on Cellular Automata for Research and Industry*, pp. 239. Lodz, Poland, Springer Nature.

Carlucci, L., I. D'Ambrosio, and M. Balsamo 2020. "Demographic and Attitudinal Factors of Adherence to Quarantine Guidelines During COVID-19: The Italian Model". *Frontiers in Psychology* vol. 11, pp. 2702.

Chen, S., C. Lanzas, C. Lee, G. L. Zenarosa, A. A. Arif, and M. Dulin 2019. "Metapopulation Model from Pathogen's Perspective: A Versatile Framework to Quantify Pathogen Transfer and Circulation between Environment and Hosts." *Sci Rep* vol. *9*, pp. 1694.

Currie, C. S. M., J. W. Fowler, K. Kotiadis, T. Monks, B. S. Onggo, D. A. Robertson, and A. A. Tako 2020. "How Simulation Modelling Can Help Reduce the Impact of COVID-19." *Journal of Simulation* vol. 14, pp. 83-97.

Fraser, C., S. Riley, R. M. Anderson, and N. M. Ferguson 2004. "Factors that Make an Infectious Disease Outbreak Controllable". In *Proc. of the National Academy of Science*, pp. 6146-61521.

Fuentes, M. A., and M. N. Kuperman 1999. "Cellular Automata and Epidemiological Models with Spatial Dependence". *Physica A: Statistical Mechanics and its Applications* vol. *267* (3-4), pp. 471-486.

Gray, L. 2003. "A Mathematician Looks at Wolfram's New Kind of Science". *Notices of the American Mathematical Society* vol. 50 (2), pp. 200-211.

Internet Engineering Task Force. 2013. *JavaScript Object Notation (JSON) Patch.* April. Accessed March 19, 2021. https://tools.ietf.org/html/rfc6902.

Internet Engineering Task Force. 2016. *The GeoJSON Format.* August. Accessed March 19, 2021. https://tools.ietf.org/html/rfc7946.

Kermack, W. O., and A. G. McKendrick. 1927. "A Contribution to the Mathematical Theory of Epidemics". In *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical character*, pp. 700-721. The Royal Society of London.

Kuiper, M. E., A. L. de Bruijn, C. R. Folmer, E. Olthuis, M. Brownlee, E. B. Kooistra, A. Fine, and B. van Rooij 2020. "The Intelligent Lockdown: Compliance with COVID-19 Mitigation Measures in the Netherlands". *Amsterdam Law School Research Paper No. 2020-20. General Subserie Research Paper No. 2020-02.*

Li, M. Y., and J. S. Muldowney 1995. "Global Stability for the SEIR Model in Epidemiology". *Mathematical Biosciences* vol. 125 (2), pp. 155-164.

Matadi, M. B. 2014. "The SIRD Epidemial Model". *Far East J. of Applied Maths* vol. 89 (1), pp. 1-14.

Wainer, G. 2017. *Discrete-Event Modeling and Simulation: a Practitioner's Approach.* CRC Press.

Zeigler, B. P., H. Praehofer, and T. G. Kim 2000. *Theory of Modeling and Simulation. Integrating Discrete Event and Continuous Complex Dynamic Systems.* Academic Press.

Zhong, S., Q. Huang, and D. Song. 2009. "Simulation of the Spread of Infectious Diseases in a Geographical Environment." *Science in China Series D: Earth Sciences* vol. 52, pp. 550-561.

AUTHOR BIOGRAPHIES

ROMÁN CÁRDENAS is pursuing the Ph.D. in Electronic Systems Engineering in Cotutelle between Technical University of Madrid and Carleton University. His research interests include modeling and simulation with applications in the IoT domain. He can be reached at r.cardenas@upm.es.

CRISTINA RUIZ-MARTÍN received a Ph.D. in Cotutelle between Universidad de Valladolid and Carleton University (2018). She is an Instructor at Carleton University. Her research interests include modeling and simulation in decision making. She can be reached at cristinaruizmartin@sce.carleton.ca.

GABRIEL WAINER received the Ph.D. degree from Université d'Aix-Marseille III. He is a Full Professor in Carleton University. His current research interests are related with modeling methodologies and tools, parallel/distributed simulation, and real-time systems. His e-mail is gwainer@sce.carleton.ca.

PETER DOBIAS received his Ph.D. degree in Physics from University of Alberta. He currently works at the Centre for Operational Research and Analysis, Defence Research and Development Canada. He does research in complex systems, optimization, wargaming, and capability-based analysis. His e-mail is peter.dobias@drdc-rddc.gc.ca.

MARK REMPEL received his MSc in Physics from University of Saskatchewan. He currently works at the Centre for Operational Research and Analysis, Defence Research and Development Canada. He does research in complex systems, optimization, wargaming, and capability-based analysis. His e-mail is mark.rempel@drdc-rddc.gc.ca.