EDITOR: Nargess Memarsadeghi, nargess.memarsadeghi@nasa.gov

DEPARTMENT: YOUR HOMEWORK ASSIGNMENT

Modeling and Simulation of Space-Based Pandemic Scenarios Using an Open-Source Platform

Gabriel Wainer, *Carleton University, Ottawa, ON, K1S 5B6, Canada*
Román Cárdenas, *Universidad Politécnica de Madrid, 28040, Madrid, Spain*
Kevin Henares, *Universidad Complutense de Madrid, 28040, Madrid, Spain*
Cristina Ruiz-Martín, *Carleton University, Ottawa, ON, K1S 5B6, Canada*

The classic susceptible-infected-recovered (SIR) models provide a good approach for modeling the spread of communicable diseases. However, this model is not suitable to understand the spatial implications on the spreading of the disease or the impact of individual interactions. Our open-source platform uses an extension of the classic SIR models for rapidly prototyping different aspects of virus spread and infection of the population using a spatial approach. This platform is useful for studying the spread of the disease and analyzing the simulation results with advanced visualization tools.

Mathematical models are useful for predicting the spread of diseases, their effects, and for assessing the effect of preventive measures. A now traditional model proposed by Kermack and McKendrick[1] introduced an approach to characterize the propagation of a disease by classifying the population in compartments, and by defining differential equations that describe how individuals transition from one compartment to another. This so-called "SIR" model considers individuals susceptible to the disease, those who are infected, and those who recovered. Although SIR-based models are useful for characterizing the spread of diseases at a macroscopic level, they are not adequate to understand the spatial dimensions of the phenomenon. Instead, in the Cell-DEVS formalism,[2] we can build space-based compartmental models using the model by dividing the space into units (called "cells," with uniform or nonuniform topologies) that represent segments of the space. Schuster[3] introduced a method for building SIR models using cellular automata and lattice gases. Here, we explore a Cell-DEVS implementation of a new SIR epidemiological model that includes a new compartment for those who are deceased, and the possibility that recovered individuals could lose immunity and become susceptible to the disease again (called a susceptible-infected-recovered-deceased-susceptible or "SIRDS" model). We also show how to model a stratification of the population into segments according to their age. We focus on how to use the model and the tools to conduct different experiments with ease.

MODELING SPACE-BASED MODELS WITH CELL-DEVS

Cell-DEVS describes a model of a system of interest by considering the spatial nature of its components and divides the model space into a collection of cells arranged in a grid. Each cell behaves as a timed finite state machine whose next state depends on its current state and inputs receiving from neighboring cells. Figure 1 shows a schematic view of a cell and a 2-D Cell-DEVS model (the figure shows a square topology but close neighbors can be defined using any kind of polygons).

1521-9615 © 2021 IEEE
Digital Object Identifier 10.1109/MCSE.2021.3075760
Date of current version 23 July 2021.

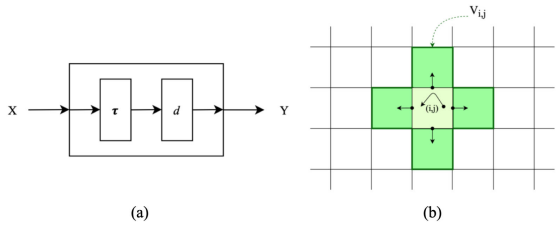

FIGURE 1. Cell-DEVS model. (a) Schematic of a cell. (b) 2-D Cell-DEVS.

Figure 1(a) shows how a single cell is organized. Cell-DEVS uses continuous time and each cell reacts to discrete events received as inputs. When an input is received, a cell is activated and invokes the local computing function τ, which computes a new state for the cell. Changes in cell state must be communicated to the neighboring cells after waiting for a time specified by the delay function d. The delay function allows complex timing behaviors in each cell.

The neighborhood set $V_{i,j}$ in Figure 1(b) contains all the cells that are neighbors of cell i, j. In the example, $V_{i,j}$ corresponds to a widely used topology, called the von Neumann neighborhood.[4]

A SIRDS MODEL IN CELL-DEVS

Our *SIRDS* Cell-DEVS model, which is based on a compartmental Cellular Automata model presented in White *et al.*,[5] divides the population into multiple age segments. At a given time t, the state of each cell i, j is defined as

$$\theta^t_{i,j} = \left\{P_{i,j}, S^t_{i,j}, I^t_{i,j}, R^t_{i,j}, D^t_{i,j}\right\} \tag{1}$$

where $P_{i,j}$ corresponds to the total population of the cell, $S^t_{i,j}$ is the portion of susceptible population, $I^t_{i,j}$ represents those who are infected, $R^t_{i,j}$ those who recovered from the disease, and $D^t_{i,j}$ is the proportion of individuals deceased due to the pandemic. The model divides the population into N different age segments, and all the model parameters are organized as N-tuples. For example, $P_{i,j}[n]$ is the total population of cell i, j in the age segment n.

When an external event activates the cell i, j, its local computation function computes the new infections at time t for every age segment n according to

$$i^t_{i,j}[n] = S^{t-1}_{i,j}[n] \cdot \sigma[n] \cdot \min\left\{1, \sum_{\alpha,\beta \in V_{i,j} k \in 1,\ldots,N} \frac{P_{\alpha,\beta}[k] \cdot I^{t-1}_{\alpha,\beta}[k]}{P_{i,j}[n]} \cdot m^{\alpha,\beta}_{i,j}[k] \cdot \nu[k]\right\} \tag{2}$$

where σ is the *susceptibility rate* (the probability that an exposed person becomes infected), $m^{\alpha,\beta}_{i,j}$ is the *mobility factor* (the probability of an individual of cell α, β moving to cell i, j), and ν is the *virulence rate* (the probability that an infected individual spreads the disease). These configuration parameters depend on the age segment. For example, $\sigma[n]$ can be higher for age segments that represent elderly people, whereas $m^{\alpha,\beta}_{i,j}[n]$ can be higher for young individuals.

Additionally, a fraction of infectious individuals become recovered depending on the *recovery rate* γ (the probability that an infected person recovers), another fraction dies according to the *mortality rate* ε (the probability that an infected person dies), and part of the recovered population loses the immunity to the disease depending on the *immunity rate* ω (the probability that a recovered person remains immune to the disease). Equation (3) shows how these values are computed

$$\begin{aligned} r^t_{i,j}[n] &= I^{t-1}_{i,j}[n] \cdot \gamma[n] d^t_{i,j}[n] = I^{t-1}_{i,j}[n] \cdot \varepsilon[n] s^t_{i,j}[n] \\ &= R^{t-1}_{i,j} \cdot (1 - \omega[n]). \end{aligned} \tag{3}$$

Finally, the cell state is updated as detailed in

$$\begin{aligned} S^t_{i,j}[n] &= S^{t-1}_{i,j}[n] + s^t_{i,j}[n] - i^t_{i,j}[n] \\ I^t_{i,j}[n] &= I^{t-1}_{i,j}[n] + i^t_{i,j}[n] - (r^t_{i,j}[n] + d^t_{i,j}[n]) \\ R^t_{i,j}[n] &= R^{t-1}_{i,j}[n] + r^t_{i,j}[n] - s^t_{i,j}[n] \\ D^t_{i,j}[n] &= D^{t-1}_{i,j}[n] + d^t_{i,j}[n]. \end{aligned} \tag{4}$$

This model has been implemented using the Cadmium simulator,[6] a C++ environment that allows defining and executing Cell-DEVS models. The SIRDS model, along with instructions to run simulations and visualize the simulation output, can be found in our public GitHub repository.[a]

SIMULATING THE SIRDS MODEL

The first example in the repository includes a configuration file to simulate the SIRDS model using COVID-19 parameters found at `config/scenario.json` in the model's GitHub repository. In this scenario, we define a uniform lattice of 25 × 25 cells.[b] There are 100 people in each cell, divided into four age segments (20 children, 40 young adults, 20 adults, and

[a]https://github.com/SimulationEverywhere-Models/CiSE-Pandemic
[b]https://github.com/SimulationEverywhere-Models/CiSE-Pandemic/tree/main/visualization

20 elderly individuals). By default, all the individuals belong to the susceptible compartment at the beginning of the simulation, regardless of their age segment

```
"shape": [25, 25],
"state": {
  "population":   [20, 40, 20, 20],
  "susceptible": [ 1,  1,  1,  1],
  "infected":     [ 0,  0,  0,  0],
  "recovered":    [ 0,  0,  0,  0],
  "deceased":     [ 0,  0,  0,  0]
},
```

For all the age segments, the virulence rate ν (i.e., the probability of an infected person infects another person) is set to 0.02. The recovery rate γ (i.e., the probability of getting recovered from the illness) is higher for younger individuals, whereas the mortality rate ε (i.e., the percentage of infected people that dies daily due to the disease) increases with age (as in the case of COVID-19). In this scenario, recovered people never lose immunity to the disease (i.e., the immunity rate ω is 1)

```
"config": {
  "susceptibility":[1, 1,     1,    1   ],
  "virulence": [0.02,  0.02,  0.02, 0.02],
  "recovery":  [0.08,  0.08,  0.06, 0.04],
  "mortality": [0,     0.001, 0.01, 0.02],
  "immunity":  [1,     1,     1,    1   ]
},
```

We use a von Neumann neighborhood, as presented in Figure 1. The mobility factor is higher in young and adult individuals, whereas children and the elderly move less

```
"neighborhood": [{
  "type": "von_neumann",
  "range": 1,
  "mobility": [0.3, 0.7, 0.5, 0.2]
}]
```

Finally, we define an alternative initial cell configuration, which starts with 1% of the young inhabitants infected

```
"infection_epicenter": {
  "state": {
    "susceptible": [1, 0.99, 1, 1],
    "infected":    [0, 0.01, 0, 0],
  }
}
```

The cell 0,0 (i.e., the upper left corner of the scenario) is the only cell using this alternative configuration as a starting value

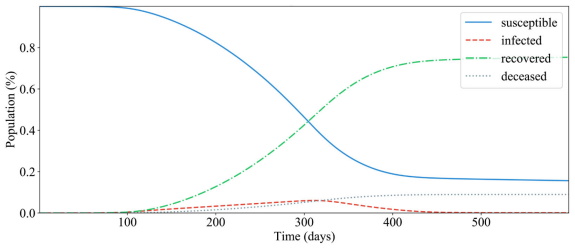

FIGURE 2. Overall epidemic trajectory of the scenario.

```
"cell_map": {
    "infection_epicenter": [[0, 0]]
}
```

You can execute this scenario by running the application and providing the path to the configuration file as follows:

```
cd bin
./CiSE-Pandemic ../config/scenario.json
```

Note that it is very important to run the application from a t1erminal whose working directory is the bash-bin/folder (i.e., the one containing the application executable).

VISUALIZATION OF THE RESULTS

A Jupyter Notebook is available in the GitHub repository[c] to analyze the simulation results and generate charts to interpret the evolution of the disease. The visualization framework includes charts with the evolution of susceptible, infected, recovered, and deceased individuals. You can visualize the state of each cell as well as the overall state of the scenario. Additionally, you can select which age segment you want to analyze.

Figure 2 shows the overall results of the example scenario. At the beginning of the simulation, almost 100% of the population is susceptible to the disease, and it progressively decreases as the pandemic spreads. At day approximately 325, we find the peak of the infection. At the end of the simulation, 70% of the population has recovered from the disease, 10% of the population is deceased, and 20% of the population remained uninfected (susceptible).

While Figure 2 gives us a general idea of how the disease will spread, we may be interested in comparing the effect of the illness on different age segments.

[c]https://github.com/SimulationEverywhere-Models/CiSE-Pandemic/tree/main/visualization

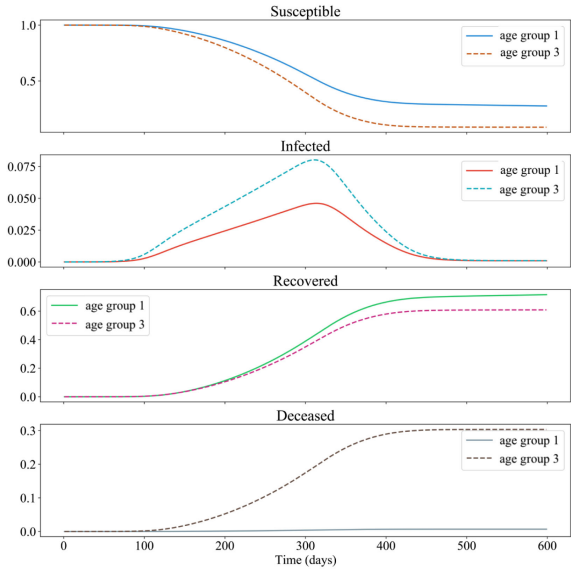

FIGURE 3. Comparison of the epidemic trajectory in young and elder people.

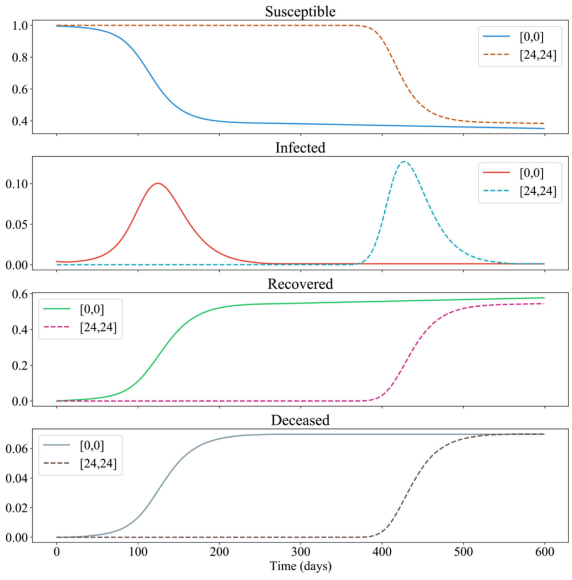

FIGURE 4. Comparison of the epidemic trajectory in cells (0,0) and (24,24).

The visualization tol enables the selection of two age segments to compare them. Figure 3 displays the evolution of each compartment for young and elderly people.

The percentage of young people that died due to the disease is negligible compared to the elderly people ratio, which is around 30%. While the peak of the infected ratio happens at the same time, the age segment corresponding to elder people reaches 7.5% of its population being infectious at the same time. On the other hand, the peak infectious ratio for young people is around 4%. The Jupyter notebook can compare the spread of the disease in different cells. Figure 4 shows the epidemic trajectory in the cells (0,0) and (24,24).

Cell (0,0) is the epicenter of the pandemic in our scenario, and the infection spreads beginning at the start of the simulation. On day 120, cell (0,0) reaches its peak number of infections. However, the pandemic starts to grow at day 375 in the cell (24,24), which is in the opposite corner cell of the scenario. This case study shows the flexibility that this simulation environment brings to conduct a variety of simulations, including changing parameters ranging from the age of the population up to the effect of the disease. The visualization tool eases the exploration of the simulation outcome and enables the comparison of the effect of the pandemic depending on spatial locations or age segments.

RUNNING NEW SIMULATIONS

Let us assume now that the virus has extended and it now affects 16 cells in the upper left corner of the scenario. The file `config/activities/1_base_scenario.json` describes the configuration of this base scenario. Based on this initial scenario, we propose the following activities.

Activity 1

Run a simulation of this new scenario. Analyze the ratios of all the compartments at the end of the simulation.

Activity 2

Using face masks is an effective method for reducing the airborne transmission of the disease. Let us assume that policymakers are considering making the use of masks mandatory for all the population, except for kids. In addition, we expect that young individuals will disobey this restriction and wear a mask only half of the time. Using a mask reduces σ to 0.5.

Run a simulation with the corresponding changes in the configuration file. How would this policy affect the epidemic transmission?

Activity 3

Let us assume a vaccine for the disease has been found. Unfortunately, we can only vaccinate 20% of

the population. Policymakers are considering vaccinating all the age segments evenly. Run a simulation considering that 20% of the population of all the age segments belong to the recovered compartment since the beginning. Does this change reduce the overall ratio of deceased people compared to the base scenario?

Activity 4

Policymakers are now considering vaccinating only the oldest population segment, as this age segment is the most affected by the pandemic. Run a simulation considering that 100% of this age segment are assigned to the recovered compartment since the beginning of the simulation. Is this vaccination strategy more successful in reducing the number of deceased people?

Activity 5

So far, we did not consider immunity loss for recovered individuals. Repeat activity 4, but setting the immunity rate to 0.999. How many waves can you identify in the simulation results? Which age compartment shows the highest ratio at the end of the simulation?

CONCLUSION

Modeling the spread of the infectious diseases with Cell-DEVS allows modelers to quickly prototype new phenomena related to the disease, and to study the simulation results with ease. The use of a formal approach allows the modelers to focus on modeling aspects and detect errors quickly without worrying about the simulation aspects, which are handled by a simulation engine. Advanced simulation tools like the open-source tool presented in this article provide new means for studying the spread of diseases and proposing new strategies to beat the pandemic.

REFERENCES

1. W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," *Proc. Roy. Soc. London. Ser. A.*, vol. 115, no. 2, pp. 700–721, 1927, doi: 10.1098/rspa.1927.0118.
2. G. Wainer, *Discrete-Event Modeling and Simulation*. Boca Rotan, FL, USA: CRC Press, 2009.
3. M. D. Schuster, "Infection modeling case study: Discrete spatial susceptible-infected-recovered model," *Comput. Sci. Eng.*, vol. 23, no. 1, pp. 83–88, Jan./Feb. 2021, doi: 10.1109/MCSE.2020.3038955.
4. L. Gray, "A mathematician looks at Wolfram's new kind of science," *Notices Amer. Math. Soc.*, vol. 50, no. 2, pp. 200–211, 2003.
5. S. Hoya White, A. Martín del Rey, and G. Rodríguez Sánchez, "Modeling epidemics using cellular automata," *Appl. Math. Comput.*, vol. 186, pp. 193–202, 2007, doi: 10.1016/j.amc.2006.06.126.
6. L. Belloli, D. Vicino, C. Ruiz-Martín, and G. Wainer, "Building DEVS models with the Cadmium tool," in *Proc. Winter Simul. Conf.*, 2019, pp. 45–59, doi: 10.1109/WSC40007.2019.9004917.

GABRIEL WAINER is a professor with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. He received the Ph.D. degree from the Université d'Aix-Marseille III, France, in 1998. He is a Fellow of the Society for Modeling and Simulation International. Contact him at gwainer@sce.carleton.ca.

ROMÁN CÁRDENAS is currently working toward the Ph.D. degree in electronic systems engineering in Cotutelle with Carleton University. He received the M.Sc. degree in telecommunication engineering from the Technical University of Madrid (UPM), Madrid, Spain, in 2019. Contact him at r.cardenas@upm.es.

KEVIN HENARES is currently working toward the Ph.D. degree with the Complutense University of Madrid, Madrid, Spain. He received the M.Sc. degree in computer engineering from Complutense University, Madrid, in 2018. Contact him at khenares@ucm.es.

CRISTINA RUIZ-MARTÍN is an instructor with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. She received the Ph.D. degree in Cotutelle between the Universidad de Valladolid and Carleton University in 2018. Contact her at cristinaruizmartin@sce.carleton.ca.