

## Research Article

# Epidemic Spread Modeling with Time Variant Infective Population Using Pushdown Cellular Automata

Senthil Athithan,<sup>1</sup> Vidya Prasad Shukla,<sup>1</sup> and Sangappa Ramachandra Biradar<sup>2</sup>

<sup>1</sup> Mody University of Science and Technology, Lakshmangarh, Rajasthan 332311, India

<sup>2</sup> SDM College of Engineering, Hubli-Dharwad, Karnataka 580002, India

Correspondence should be addressed to Senthil Athithan; [senthilathithan@hotmail.com](mailto:senthilathithan@hotmail.com)

Received 20 July 2014; Revised 3 October 2014; Accepted 5 October 2014; Published 12 November 2014

Academic Editor: Wen-Cheng Liu

Copyright © 2014 Senthil Athithan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The world without a disease is a dream of any human being. The disease spread if not controlled could cause an epidemic situation to spread and lead to pandemic. To control an epidemic we need to understand the nature of its spread and the epidemic spread model helps us in achieving this. Here we propose an epidemic spread model which considers not only the current infective population around the population but also the infective population which remain from the previous generations for computing the next generation infected individuals. A pushdown cellular automata model which is an enhanced version of cellular automata by adding a stack component is being used to model the epidemic spread and the model is validated by the real time data of H1N1 epidemic in Abu Dhabi.

## 1. Introduction

Computational models in epidemics provide insight into dynamics of the disease spread across a geographical region. Given a small amount of relevant real time data the model would give us after certain amount of time what would be the epidemic situation across the geographical region. Cellular automata (CA) models are considered to be very handy and efficient in handling the real time simulation problems in epidemiology. The reason behind the simplicity of this CA model is its ability to attain the global behavior from the local behavior by the interaction of its cells [1–3].

The power of CA has been utilized to solve wide variety of problems like solving Weyl, Dirac, and Maxwell equations [4], diffusion equation [5], and Poisson equations [6]. Epidemic spread modeling through cellular automata has been an importance to many of the researchers. An efficient epidemic spread model through CA is given by Hoya White et al. [7]; in this model the population is assumed to be constant and the rules for the CA are considered to be static. The movement of population and its effects during an epidemic

spread are given by Sirakoulis et al. [8]. The population when grouped under patches and their movement during an epidemic spread are discussed in our earlier work [9]. The spatial pattern and its dynamics along with noise in epidemics are being provided by Sun et al. [10]. Stochastic model for epidemic spread with quarantine and vaccination strategies have been provided by Wang Jeffrey [11].

In all these works discussed above, the susceptible population is infected by the infected population available in the current time step. There are no considerations for the populations who are infected at the previous time steps. If we consider the infected population at current time as  $t$ , then the population which was infected at the previous time steps is considered as  $t - 1$ ,  $t - 2$ , and so forth. This work largely focusses on how the latent infected population, that is, the ones which are infected at  $t - 1$ ,  $t - 2$ , has an effect in infecting the current set of susceptible population in a cell. This relates to a question: how could we segregate or handle the infected populations at  $t - 1$  or  $t - 2$ ? The solution is to append a stack to each cell of the cellular automaton and call the modified CA as pushdown cellular automaton or PCA [12]. The idea

TABLE 1: Structure of the cell status word.

Cell_id	CPV	<i>I</i> value	<i>R</i> value	<i>M</i> value
---------	-----	----------------	----------------	----------------

is to hold the infective population at different time steps at different levels of stack and use them for computation for the next time step accordingly.

In this research paper we begin introducing the cellular automata in Section 2.1, then in Section 2.2 we introduce the usage of cellular automata in epidemic spread simulation and the cellular automata with stack configuration would be handled in Section 2.3 followed by various modifications in stack configured cellular automata model in the next sections. The results and discussions are dealt with in Section 3 followed by conclusion and the future scope of the model.

## 2. Materials and Methods

**2.1. Cellular Automata.** Cellular automata is a universal Turing machine which is capable of solving many complex problems ranging from simulating forest fire spread, landmine detection, robotic movements, and object avoidance to understanding the spread of epidemics [15]. It consists of a grid of cells, which is the basic building component of a cellular automaton. Each cell could be in one of the finite amount of states. The next state of each cell depends on the state of the current cell and states of the neighboring cells. The neighborhood configuration can be many but mostly Von-Neumann neighborhood or Moore neighborhood is used depending upon which of the surrounding cells we consider for computing the next state of the current cell.

**2.2. Cellular Automata for Epidemic Spread.** A cellular automaton can be configured for simulating an epidemic spread. The following assumptions would be applied for configuring the cellular automata.

A cell could be in one of the following finite number of states: susceptible, infected, and recovered/removed (dead). We assume the epidemic model to be of SIR type with equal birth and death rate so the population as a whole remains constant.

The neighborhood configuration would be of Moore's neighborhood which considers the top, down, left, right, and the diagonal neighbors as well.

The population count in each cell is assumed to be constant and no movement between the cells is permitted. This assumption is for the very basic model whereas in real time the movement of population between the cells is permitted.

The initial configuration of the cellular automata and the cellular status word for each cell is as described in our work [9].

The cell status word (CSW) (see Table 1) has five parameters, namely, (1) Cell\_id which is the current cell denoted by  $C(x, y)$ , (2) CPV (critical population value) which is the fraction of infected population over total population in the cell, (3) *I* value which takes Boolean value 0/1: 0 means the cell

is free of infective population; 1 means some percentage of cell is infected so we consider that the cell is infected, (4) *R* value which will also take either 0 or 1 depending upon whether the population inside the cell is recovered from the disease or not, and (5) *M* value which is the movement flag for a particular cell used to control the movement of the population from one cell to another. If the value is 0 it means no movement possible from the cell.

The critical population value is computed by

$$CPV = \frac{IPC_{x,y}}{TPC_{x,y}}, \quad (1)$$

where IPC represents the infected population count in the cell and TPC represents the total population count in the cell. The critical population value for a cell at the next time step would be the function of the CPVs of the current cell at time  $t$  and the CPVs of the neighboring cells; that is,

$$CPV_{x,y}^{t+1} = fn(CPV_{x,y}^t, CPV_{x+1,y}^t, CPV_{x-1,y}^t, CPV_{x,y+1}^t, CPV_{x,y-1}^t, CPV_{x+1,y+1}^t, CPV_{x+1,y-1}^t, CPV_{x-1,y-1}^t, CPV_{x-1,y+1}^t). \quad (2)$$

Size of the grid or cellular array depends on the user and the type of geographical area under consideration. The bigger the cellular array is the longer it takes for the simulation to run.

**2.3. Pushdown Cellular Automata for Time Variant Infective Population.** A modified cellular automata with added stack to each cell is termed as pushdown cellular automata. The concept of time variant infective population comes to importance in the case of population movement across the cells. In each time step a certain percentage of population moves from one cell to another based on the configuration of the cellular array. During the movement a certain percentage of infective, susceptible, and recovered population moves from one cell to another and the count is considered to be random across all the three sets of populations. So some amount of infective population moves from current cell to another, and some remains within the current cell. So there can be two categories of infective population in the current cell, that is, the ones which remain from the previous time step and the ones which could come from another cell to the current one. The infective population which remain from the previous time step would be termed as  $I_{t-1}$  and the infective population which come into the current cell would be termed as  $I_t$ . The degree of infection caused by the newly infected infective population and the infection caused by the older ones would be different. So the cell update rule will be modified to accommodate the effect of these current infective population and the older ones. Here in this research work we consider three time steps  $t, t-1$ , and  $t-2$ , respectively. So the stack size would be 3 (plus one to hold the bottom symbol \$ which indicates the bottom of the stack).

TABLE 2: Structure of the cell status word.

Cell_id	SP*	I value	R value	M value
\$*				

TABLE 3: Structure of the CSW at time  $t = 0$ .

Cell_id	SP*	I value	R value	M value
*	*	\$*		

TABLE 4: Structure of the CSW at time  $t = 1$ .

Cell_id	SP*	I value	R value	M value
*	$I_{t-1}^*$	\$*		

TABLE 5: Structure of the CSW at time  $t = 2$ .

Cell_id	SP*	I value	R value	M value
$I_{t-2}^*$	$I_{t-1}^*$	\$*		

**2.4. Modification of CSW with Stack.** The cell status word is modified to hold the stack contents for storing the critical population values of the previous time steps. The modified CSW will look like Table 2.

CPV part is replaced by SP (stack pointer) which points to the top of the stack attached to the current cell. Default value/initial value of the SP would be \$ which indicates the bottom symbol of the stack and also the emptiness of the stack. The structure of the CSW during the three time steps is given by Tables 3, 4, and 5.

In every time step the current cell is updated based on the status of itself and the status of the neighboring cells.

**2.5. Configuration of PDCA.** The PDCA has the following components.

Finite set of states  $(S, I, R)$ : neighborhood configuration would be Moore neighborhood (all cells surrounding the cell to be considered for update). The neighborhood status is divided into three categories based on the number of infected cells out of the eight neighborhood cells. The status is minimum denoted by “m” when the number of the infected cells surrounding the current cell is less than 4. The status is half denoted by “h” when the number of the infected cells surrounding the current cell is either 4 or 5. The status is full denoted by “f” when the number of the infected cells surrounding the current cell is more than 5. Given below is the various neighborhood status. White indicates susceptible state, red indicates infected state, and green indicates removed or recovered state (Figure 1).

Stack of size three is attached to each CSW of the cell with the following stack symbols:

\$: special symbol to indicate the bottom of the stack,

$I_{t-1}$ : number of infective remaining from the previous time step,

$I_{t-2}$ : number of infective population remaining from the previous to previous time step.

Transition function ( $\delta$ ) is given by the following.

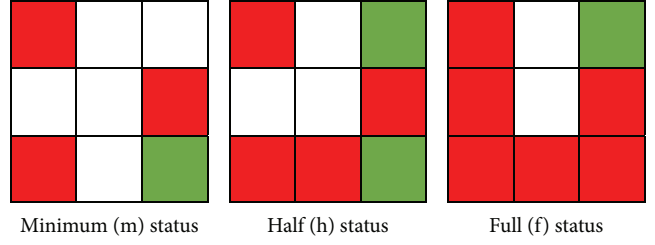


FIGURE 1: Neighborhood status configuration.

Initially all the cells are assumed to be in the susceptible state (S).

Each transition function will be of the following form:  $\delta(\langle \text{current state of the cell} \rangle, \langle \text{state of the neighborhood} \rangle, \langle \text{stack content before transition} \rangle) \rightarrow (\langle \text{state of the cell after transition} \rangle, \langle \text{stack content after transition} \rangle)$

(i)  $\delta(S, \lambda, \lambda) \rightarrow (S, \$)$  (pushing the \$ on to the stack before starting the computation).

The biological meaning of \$ can be considered as the residual infective population whose impact is not severe and remains in every cell:

(ii)  $\delta(S, m, \$) \rightarrow (S, m, I_{t-1} \$)$ .

When the cell is in susceptible state and the neighborhood is in the minimum status with the stack top being \$ then the cell remains in susceptible state and the neighborhood also in minimum status and the number of infective population that remains in the cell would be pushed as  $I_{t-1}$  on top of the stack. The biological meaning would be that current location is not infective (which means the number of infective population is less) and surrounding is also having less number of infective population; then in the next time step the cell is not infected and remains in susceptible state:

(iii)  $\delta(S, h, \$) \rightarrow (S, h, I_{t-1} \$) \mid (I, f, I_{t-1} \$)$ .

When the cell is in susceptible state and the neighborhood is in the half status with the stack top being \$ then the cell remains in susceptible state and the neighborhood also in half status and the number of infective population that remains in the cell would be pushed as  $I_{t-1}$  on top of the stack. The other choice of this state to the next time step would be changed to infective state, and the neighborhood status changes from half to full and the  $I_{t-1}$  would be pushed on top of the stack. The choice would be selected at random as in real time cases this randomness helps us in introducing the noise factor into the system which is very appropriate since the epidemic situation is a nondeterministic one. As the number of infective population increases around the surroundings the current cell can change from susceptible state to infective state:

(iv)  $\delta(S, f, \$) \rightarrow (I, f, I_{t-1} \$)$ .

When the cell is in susceptible state and the neighborhood is in full status with stack top being \$ then the cell changes its state to infective with the neighborhood remaining in

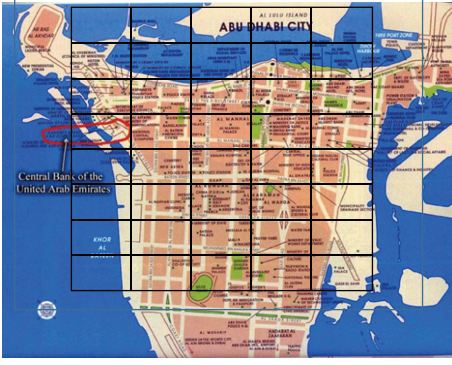


FIGURE 2: Mapping of cellular grid with map of Abu Dhabi [13, 14].

full status and  $I_{t-1}$  is pushed onto the stack. Likewise the remaining transition functions for various states of the PDCA are

- (i)  $\delta(S, m, I_{t-1}\$) \rightarrow (S, m, I_{t-2}I_{t-1}\$)$ ,
- (ii)  $\delta(S, h, I_{t-1}\$) \rightarrow (I, h, I_{t-2}I_{t-1}\$)$ ,
- (iii)  $\delta(S, m, I_{t-2}I_{t-1}\$) \rightarrow (I, m, I_{t-1}\$)$ ,
- (iv)  $\delta(I, h, I_{t-2}I_{t-1}\$) \rightarrow (I, h, I_{t-2}I_{t-1}\$) \mid (R, m, I_{t-1}\$)$ ,
- (v)  $\delta(I, m, I_{t-1}\$) \rightarrow (I, h, I_{t-2}I_{t-1}\$) \mid (R, m, \$)$ ,
- (vi)  $\delta(I, f, I_{t-1}\$) \rightarrow (I, f, I_{t-2}I_{t-1}\$) \mid (I, h, I_{t-1}\$)$ .

**2.5.1. Assumptions Regarding the Stack.** The following assumptions were taken into account with regard to the stack.

- (i) Here we consider only two time steps before  $t-1$ ,  $t-2$  so the stack size becomes three along with the bottom symbol  $\$$ . If we increase the time steps to unlimited amount, this will lead to the proportionate amount of increase in the size of stack (memory) which will definitely affect the efficiency of the computation.
- (ii) There is no movement of the population which is inside the stack.

**2.6. Setting Up the Model.** The first step in designing the cellular automata model is defining the size of the grid. The size of the grid would be inversely proportional to the speed of computation [8, 9]. The grid would be taken with the background of the city of Abu Dhabi since we would be simulating our results with the data of H1N1 database of Abu Dhabi (Figure 2).

After mapping the grid with the map of the city the boundaries are drawn for the simulation area (Figure 3). We presume that the epidemic would not spread beyond these boundaries since it is surrounded by sea. Overlapping top and bottom rows and left and right rows of the grid are not a feasible assumption as the city is surrounded by the sea and three sides.

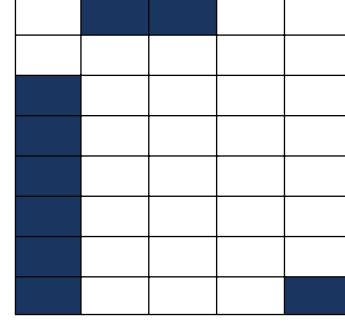


FIGURE 3: Mapped grid with boundaries blocked used for simulation.

Initial conditions are as follows:

- (i) cellular array size  $\rightarrow$  defined by the user,
- (ii) number of persons in each cell  $\rightarrow 100$ ,
- (iii)  $I$  value  $\rightarrow 0$  for all cells,
- (iv)  $R$  value  $\rightarrow 0$  for all cells,
- (v) infection period  $\rightarrow$  defined by the user as per the type of disease,
- (vi)  $M$  value  $\rightarrow 0$  no movement in initial configuration.

The critical population value (CPV) would be now calculated by the following way:

$$CPV = \frac{IPC_{x,y}^t + IPC_{x,y}^{t-1} + IPC_{x,y}^{t-2}}{TPC_{x,y}}. \quad (3)$$

The CPV of a cell in the next time step would be calculated in the same way after transition into a new state based on the transition function.

**2.7. Homogeneous PDCA without Movement of Population.** In this configuration the number of persons in each cell is kept the same and the movement of population between the cells is also restricted by giving the  $M$  value in CSW to 0. This configuration is not the right one to model the real time environment but necessary to understand the basic parameters during the simulation.

#### 2.7.1. The Algorithm for Homogeneous PDCA (Equal Population in Each Cell with No Movement)

*Step 0.* Get the input from the user: the value of  $tr$  (time period for recovery),  $tg$  (number of generations), and  $ti$  (time period



for immunity). Block the boundary cells so as to control the infection spread.

*Step 1.* Initialize the generation counter to 0 and push \$ on to the stack for all the cells.

*Step 2.* Initialize all the cells with equal number of populations.

*Step 3.* Initialize the  $M$  value of the CSW of all cells to 0 to control the movement of population.

*Step 4.* Infect the desired cell.

*Step 5.* Increment the generation counter by 1.

*Step 6.* Find out the region of cells which are infected and the region of cells which are not infected by checking the  $R$  value and  $I$  value.

*Step 7.* Based on the top symbol of the stack and the status of the neighborhood change the state of the current cell and modify the stack according to appropriate transition function.

*Step 8.* If  $R$  value is 1 then go to 9.

Else calculate the CPV value based on the cellular automata update rule to find the fraction of infected population.

*Step 9.* If the number of generations is less than  $tg$  then go to step 4.

*Step 10.* Stop.

**2.8. Heterogeneous Population with Movement.** In this configuration, the population in each cell varies and moreover the movement of population between the cells is also possible. The movement of population from a particular cell is possible only if the cell's CSW-M value is 1; otherwise movement cannot be done from that cell. If we consider a geographical area (city or town) to which the grid is mapped then all those cells which are in line with the streets or bridges of the city are enabled for movement. After deciding which cells allow the movement of population, we need to identify the amount of population that would be moving from the cell and the time of movement.

The amount of population which can move from one cell to another cell would be 5% to 7% and we can deploy two techniques to trigger the movement of the population. First one is that after some random amount of time or when the infected fraction of population reaches more than 15% of the total population the movement is done. The second choice would be more natural in the sense that the people tend to move from one place to another when there would be more infection around.

### 2.8.1. Algorithm for Heterogeneous Population with Movement

*Step 0.* Get the input from the user: the value of  $tr$  (time period for recovery),  $tg$  (number of generations), and  $ti$  (time period for immunity).

*Step 1.* Initialize the generation counter to 0 and set enable field of CSW to 1 for all those cells except the boundary cells.

*Step 2.* Initialize all the enabled cells with variable populations.

*Step 3.* Initialize the  $M$  value of the CSW of all cells which are in the pathway to 1 in order to enable the movement.

*Step 4.* Push \$ on the stack and infect the desired cells.

*Step 5.* Increment the generation counter by 1.

*Step 6.* Find out the region of enabled cells which are infected and the region of cells which are not infected by checking the  $R$  value and  $I$  value.

*Step 6.1.* Based on the top symbol of the stack and the status of the neighborhood change the state of the current cell and modify the stack according to appropriate transition function.

*Step 7.* Check if the infected percentage is greater than 15%. If yes then go to step 8; else go to step 9.

*Step 8.* Check whether the cell is movement enabled or not by checking  $M$  value to be 1. If yes then go to step 8.1; else go to 9.

*Step 8.1.* If the cell is within the boundary limits transfer the 0 to 5% amount of population taken at random to the next cell in the neighborhood to the direction in which there is a pathway available.

*Step 9.* If  $R$  value is 1 then go to 10.

Else calculate the CPV value by (3) based on the cellular automata update rule to find the fraction of infected population.

*Step 10.* If the number of generations is less than  $tg$  then go to step 4.

*Step 11.* Stop.

The PDCA could be slightly modified for considering the movement factor. The modification would be in terms of the neighborhood configuration; instead of using the traditional Moore neighborhood we can use the extended neighborhood concept so as to realistically model the situation. The extended neighborhood combines the traditional Moore neighborhood and randomly selects the cells across the neighborhood cells.

As we extend the neighborhood to broader radius say to 2 or 3 then the region of neighborhood increases for 8 to 16 or 24 cells, so the more the radius the less the probability of contact and infection. So the radius is confined to 1.

If the current cell is  $C_{i,j}$ ,  $M_{i,j}$  refers to the Moore neighborhood and  $E_{i,j}$  refers to the cells that are chosen randomly; then the extended neighborhood of

$$C_{i,j} = M_{i,j} + E_{i,j} \quad (4)$$

and accordingly the cell update rule will also change for the extended neighborhood [16]:

$$\begin{aligned} CPV_{x,y}^{t+1} = fn & \left( CPV_{x,y}^t, CPV_{x+1,y}^t, CPV_{x-1,y}^t, CPV_{x,y+1}^t, \right. \\ & CPV_{x,y-1}^t, CPV_{x+1,y+1}^t, CPV_{x+1,y-1}^t, \\ & \left. CPV_{x-1,y-1}^t, CPV_{x-1,y+1}^t \right) + E_{i,j}. \end{aligned} \quad (5)$$

This random choosing of cells in the neighborhood part and the random selection of individual for the movement give us the source for the noise addition in the model as the model we are considering is of nature which is not a deterministic one. The stack rules will remain the same and the transition functions based on stack will also be maintained similar to the previous one.

**2.9. Important Parameters in the Epidemic Spread.** Susceptible population denoted by  $S$  is the population which is not infected but prone to infection when it is around the infected set of population. In this simulation environment we consider all the population initially to be susceptible. There may be many cases where certain population can be immune to the disease under consideration, so they may not be part of the susceptible set.

Infected population denoted by  $I$  is the population which is actually infected by the disease under consideration. Certain amount of individuals from the susceptible set is infected by the infection rate denoted by " $\beta$ ." Here in our model we consider three different types of infective population; those are the ones present in the cell at current time  $I_0$ , the ones which remain from the previous time step  $I_{t-1}$ , and the ones from  $I_{t-2}$  time step. The current ones are considered in the cell and in the next time step it is pushed onto the stack as  $I_{t-1}$ . Since we have three different types of infective population, the infection rate also would be different for them; the infection rate will increase for the infected population if not treated or vaccinated.

We define three different infection rates for the three sets of infected population. They are  $\beta_0$ ,  $\beta_1$ , and  $\beta_2$  for  $I_0$ ,  $I_{t-1}$ , and  $I_{t-2}$ , respectively. We can define the relation between the three types of infection rates by

$$\beta_0 \leq \beta_1 \leq \beta_2. \quad (6)$$

The above equation is valid in case there is no treatment or vaccination. So the infection rate increases as the time progresses and the infected individuals would be capable of infecting more numbers of susceptible population:

$$\beta_0 \leq \beta_1 \geq \beta_2. \quad (7)$$

The above equation is applicable when there is a treatment or vaccination strategy is underway to the infected set of

population. Initially when the treatment starts the effect will not be visible but after certain time period the infection rate starts reducing. That is the reason why  $\beta_1$  is greater than both  $\beta_0$  and  $\beta_2$ .

Recovered populations denoted by  $R$  are those which are recovered from the infected population and become immune to the disease. The rate of recovery from the infected population is given by  $\gamma$ . The recovery rate would be directly proportional to the level of vaccination strategy used for the infected population.

Reproduction number denoted by  $R_0$  which is the important component in the epidemic environment denotes the amount of individuals infected by a single infected individual in a cell placed along with fully susceptible population:

$$R_0 = \frac{\beta}{\mu + \gamma}, \quad (8)$$

where  $\beta$  is the infection rate,  $\mu$  is the birth and death rate considered to be equal which means that the amount of birth is equal to the amount of death, and  $\gamma$  represents the recovery rate. When  $R_0$  is less than 1 then the system is in infection-free stable state. When  $R_0$  is greater than 1 the system is in endemic stable state. Always the objective is to reduce the  $R_0$  to less than one by increasing the recovery rate. The above given equation is the general one and for estimating  $R_0$  in the cellular automata paradigm with various radius of neighborhood the solution is given by

$$R_0 = 2m \left( 1 - \frac{1}{4r(r+1)} \right)^{m-1} (1 - e^{-k}), \quad (9)$$

where  $2m$  is the average number of connections per cell,  $(1 - e^{-k})$  is the probability that susceptible cells become infected, and  $r$  is the neighborhood radius.

**2.10. Controlling the Spread of Epidemics.** There are various strategies for controlling the epidemics; quarantine and vaccination stand out amongst them.

**2.10.1. Quarantine.** There are two terms in the world of epidemic which needs to be clarified before dealing with quarantine. The terms are isolation and quarantine; what is the difference between these two? The clarification rightly given by Center for Disease Control (CDC-USA) [17] is as follows.

**Isolation.** It applies to person who is known to be ill with a contagious disease.

**Quarantine.** It applies to those who have been exposed to a contagious disease but who may or may not become ill.

Quarantine strategy is implemented by blocking 9 cells with a boundary and stating that the population outside the cell is susceptible to infection. The probability of the infection would be  $1 - (1 - \beta)^m$  where  $r$  is the escape rate, which means number of persons who can go through the quarantine defense [11]. This could be more realistic because we consider no quarantine system is full proof. The quarantine is analyzed with 10% and 25% escape rate.

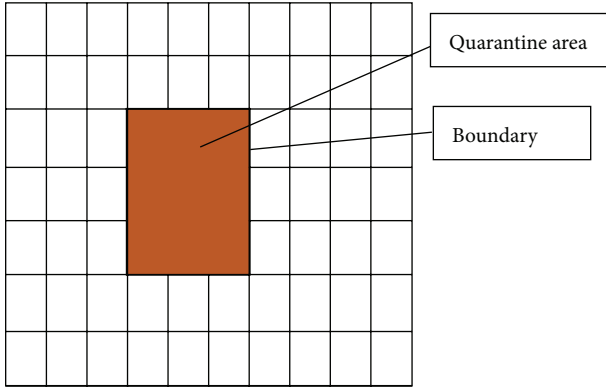


FIGURE 4: Quarantine setup.

The area to be quarantined can be identified using the full status of the stack given in the figure (Figure 4). The stack full status indicates that more numbers of infective population with different infection rates are accumulated when compared to the other region.

**2.10.2. Vaccination.** The most common and effective technique in dealing with control of epidemics is the vaccination technique. There can be many methods of vaccination: one is to continuously vaccinate for particular amount of time and the other one is to vaccinate for a small amount of time in intervals which is known as pulse vaccination. Here continuous vaccination after the 30th generation is done for another 10 generations. The epidemic spread simulation techniques help to reach cost effective technique in implementing the vaccination. The areas where the disease spread is more have to be identified and vaccinated accordingly. The important concept in vaccination is the efficacy of the vaccine; vaccine efficacy is defined as the reduction in effect of disease among people who have been vaccinated compared to the effect in unvaccinated people [18]. The influenza vaccine efficacy rate is considered to be 85% [19]. The probability of infection spread in the vaccinated area is  $(1 - r)(1 - \beta)^n$  where  $r$  is the efficacy rate. Randomly selected cells which have large number of infective population starting from the 30th generation are being vaccinated and the results have been found.

**Simulation environment:** simulation is done using MATLAB Ver. 2013 (64 bit). The processor used is Intel Core-i5 and RAM capacity is 8 GB.

### 3. Results and Discussion

**3.1. Homogeneous Model.** Homogeneous model is the model in which there is a fixed amount of population in every cell. A  $50 \times 50$  cellular grid (optimal grid size is the tradeoff between time, size, and accuracy) is used to simulate the epidemic spread as per the available surveillance data of H1N1 infection in Abu Dhabi [13, 14]. The black colored blocks in the grid are the boundary which is traced out from the Abu Dhabi city map. We have infected certain cells randomly in the left side of the grid and started the simulation and the snapshots of

TABLE 6: Number of infected population in Abu Dhabi over 12 months [13, 14].

Months	Confirmed cases	Recovered cases
9-May	1	1
9-Jun	11	11
9-Jul	172	172
9-Aug	233	228
9-Sep	134	128
9-Oct	194	189
9-Nov	178	171
9-Dec	16	14
10-Jan	8	7
10-Feb	12	12
10-Mar	5	5
10-Apr	2	2
Total	966	940

each of the 30 generations are provided here which clearly depicts how the infection spreads out in the beginning and is erased out in the end. The last two parts of the figure provide information about the cells in which the augmented stack is full and empty. The epidemic spread is simulated using the data of H1N1 infection in Abu Dhabi [13, 14] over the period of twelve months with the parameters of  $\mu = 1/12$ ;  $\beta_0 = 1.32$ ;  $\beta_1 = 1.46$ ;  $\beta_2 = 1.78$ . Average  $\beta$  comes out to be 1.52,  $\gamma = 1.39$ , and  $R_0 = 1.0319$ .

The average infection period of a cell as a whole is around 15 to 20 generations (Table 6).

Figure 6 shows the curve of SIR model which matches with the ideal epidemic SIR curve and the number of infective population peak during the 4th and 5th month, respectively. Figure 7 gives us the picture of how the number of infective population increases during the 4-5th month along with the increase in the number of stacks getting full by  $I_{t-2}$ ,  $I_{t-1}$ ,  $I_t$ .

Figure 8 shows us the number of confirmed cases with the effect of no quarantine, quarantine with 25% escape rate, and quarantine with 10% escape rate. 10% escape rate is considered to be optimum. Figure 9 gives the picture of how the number of infective population decreases with the random vaccination done every 30 generations across the cells in the grid.

**3.2. Heterogeneous Population with Movement.** Population movement is enabled in those CSWs where the stacks were full which is identified from Figure 5(k) (Figure 10). Simulation is done for another 300 generations and the infection spread reach is calculated from the start location to the various identified locations in the grid (Table 7). This analysis helps us in planning the vaccination, quarantine, and evacuation procedure which is a very important phase in epidemic situation.

### 4. Conclusion and Future Scope

The epidemic spread model using pushdown cellular automata has been considered here in this work to examine the

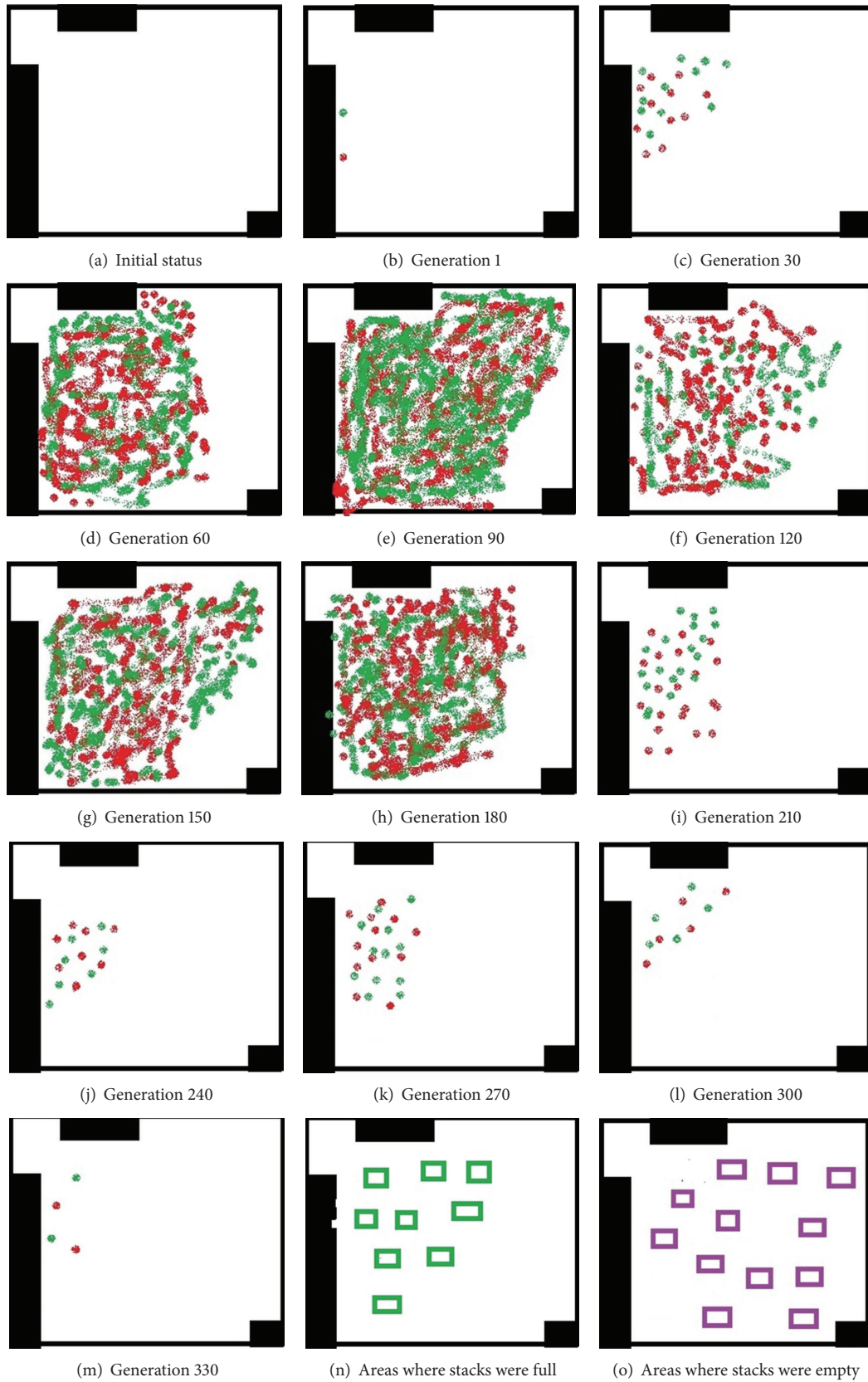


FIGURE 5: ((a) to (m)) Simulation results showing the geographical spread of infection. (n) and (o) shows full and empty stack regions.



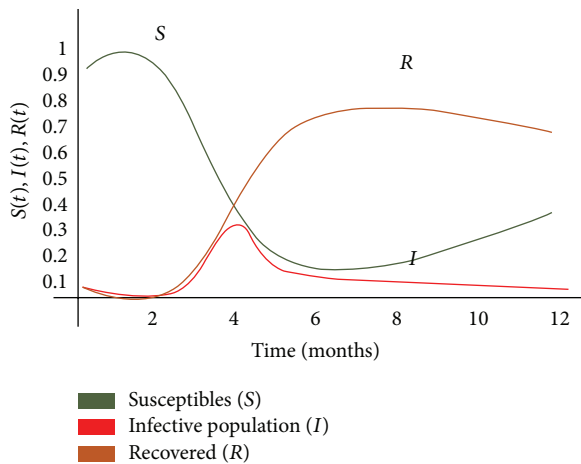


FIGURE 6: SIR curve for the simulated data.

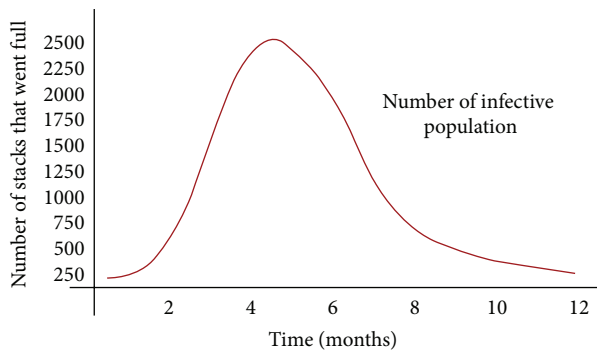


FIGURE 7: Variation of number of infective population over stack fullness.

TABLE 7: Number of generations for the infection to reach from start to specified position.

Position	Infection reaches in generation
Start	0
A	64
B	92
C	123
D	197

effects of the time variant infective population that remain in a cell for a certain time period. The effective use of the stack to hold these infective population of the previous time periods and effect of these infective population on the infection spread in the coming time steps are studied with two models: one with the homogeneous population across all the cells with no movements between the cells and the other one with different sizes of population in each cell with movement between the cells across the desired locations. The effect of quarantine and the vaccination process is also being investigated with the homogeneous model. The analysis helps us to understand the nature of the epidemic spread across the geographical region and to plan the quarantine

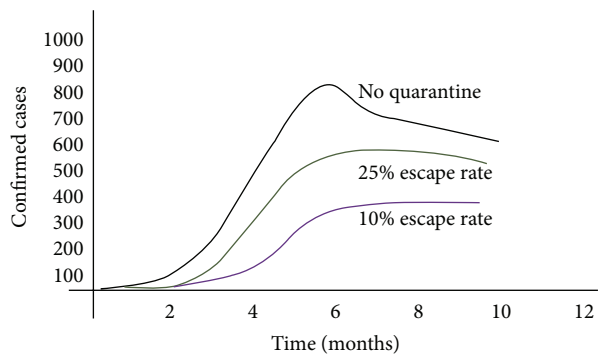


FIGURE 8: Effect of quarantine.

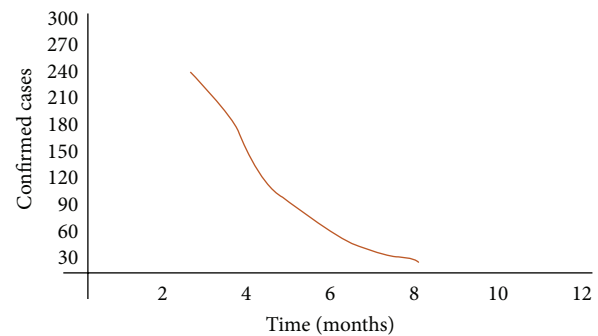


FIGURE 9: Effect of vaccination.

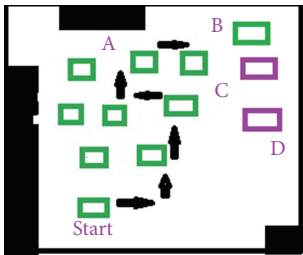


FIGURE 10: Movement enabled cells and start position.

and evacuation strategy in case of a pandemic. This model has been tested with the H1N1 data for Abu Dhabi over the period of twelve months from May 2009 to April 2010. Future work can be carried out in this area by considering 3D cellular automata which can take care of the population density like the people living in buildings on multiple floors. The other variation of considering two different types of influenza infection which has got two different recovery rates also can be explored.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] A. Ilachinski, *Cellular Automata—A Discrete Universe*, World Scientific, 2002.
- [2] M. Sipper, *Evolution of Parallel Cellular Machines: The Cellular Programming Approach*, Springer, 2004.
- [3] J. von Neumann, *Theory of Self Reproducing Automata*, University of Illinois, Urbana, Ill, USA, 1966.
- [4] I. Bialynicki-Birula, “Weyl, Dirac, and Maxwell equations on a lattice as unitary cellular automata,” *Physical Review D*, vol. 49, no. 12, pp. 6920–6927, 1994.
- [5] B. Chopard and M. Droz, “Cellular automata model for the diffusion equation,” *Journal of Statistical Physics*, vol. 64, no. 3-4, pp. 859–892, 1991.
- [6] H. Chen, W. H. Matthaeus, and L. W. Klein, “Theory of multi-color lattice gas: a cellular automaton Poisson solver,” *Journal of Computational Physics*, vol. 88, no. 2, pp. 433–466, 1990.
- [7] S. Hoya White, A. M. del Rey, and G. R. Sánchez, “Using cellular automata to simulate epidemic diseases,” *Applied Mathematical Sciences*, vol. 3, no. 20, pp. 959–968, 2009.
- [8] G. C. Sirakoulis, I. Karafyllidis, and A. Thanailakis, “A cellular automaton model for the effects of population movement and vaccination on epidemic propagation,” *Ecological Modelling*, vol. 133, no. 3, pp. 209–223, 2000.
- [9] A. Athithan, V. P. Shukla, and S. R. Biradar, “Dynamic cellular automata based epidemic spread model for population in patches with movement,” *Journal of Computational Environmental Sciences*, vol. 2014, Article ID 518053, 8 pages, 2014.
- [10] G.-Q. Sun, Z. Jin, L.-P. Song, A. Chakraborty, and B.-L. Li, “Phase transition in spatial epidemics using cellular automata with noise,” *Ecological Research*, vol. 26, no. 2, pp. 333–340, 2011.
- [11] B. Wang Jeffrey, “School’s Out ? Designing Epidemic Containment Strategies with a spatial stochastic method,” 2010, <http://www.dancingdarwin.com/sca>.
- [12] M. Kutrib, “Pushdown cellular automata,” *Theoretical Computer Science*, vol. 215, no. 1-2, pp. 239–261, 1999.
- [13] F. Ahmed, F. A. Hosani, A. Al Mannaie, and O. Harrison, “Early outcomes of pandemic influenza (H1N1) 2009 surveillance in Abu Dhabi Emirate, May-August 2009,” *Eastern Mediterranean Health Journal*, vol. 18, no. 1, pp. 31–36, 2012.
- [14] F. A. Rihan, “SIR Model of Swine Influenza Epidemic in Abu Dhabi,” *Journal of Public Health Frontier*, vol. 1, no. 4, pp. 85–89, 2012.
- [15] Nandita Gandhi and A. Senthil, *Optimized Landmine Detection Simulation Using Cellular Automata*, Lambert Academic Publications, 2011.
- [16] C. Guan, W. Yuan, and Y. Peng, “A cellular automaton model with extended neighborhood for epidemic propagation,” in *Proceedings of the 4th International Joint Conference on Computational Sciences and Optimization (CSO '11)*, pp. 623–627, Yunnan, China, April 2011.
- [17] Centers for Disease Control and Prevention, “Understand Quarantine and Isolation,” <http://emergency.cdc.gov/preparedness/quarantine/>.
- [18] G. A. Weinberg and P. G. Szilagyi, “Vaccine epidemiology: efficacy, effectiveness, and the translational research roadmap,” *Journal of Infectious Diseases*, vol. 201, no. 11, pp. 1607–1610, 2010.
- [19] K. M. Neuzil and K. M. Edwards, “Influenza vaccines in children,” *Seminars in Pediatric Infectious Diseases*, vol. 13, no. 3, pp. 174–181, 2002.

