

# Applying Inhomogeneous Probabilistic Cellular Automata Rules on Epidemic Model

Wesam M. Elsayed<sup>1</sup>

Department of Mathematics, Faculty  
of Science, Mansoura University,  
Mansoura, 35516, Egypt.  
Mansoura, Egypt.

Ahmed H. El-bassiouny<sup>2</sup>

Department of Mathematics, Faculty  
of Science, Mansoura University,  
Mansoura, 35516, Egypt.  
Mansoura, Egypt.

Elsayed F. Radwan<sup>3</sup>

Department of Computer Science,  
Faculty of Computer and Information  
Sciences Mansoura University,  
Egypt, P.O. Box: 35516

**Abstract**—This paper presents some of the results of our probabilistic cellular automaton (PCA) based epidemic model. It is shown that PCA performs better than deterministic ones. We consider two possible ways of interaction that relies on a two-way split rules either horizontal or vertical interaction with 2 different probabilities causing more of the best possible choices for the behavior of the disease. Our results are a generalization of that Hawkins et al done.

**Keywords**—Probabilistic Cellular Automata (PCA); Epidemic modeling; Optimization.

## I. INTRODUCTION

Because of the spread of diseases, a technical innovative model should be made to recover their time regions. Many researches tried to solve this problem based on medical disease feature, which suffer from unpredictable ones. Whilst a single infected host might not be significant, a disease that spreads through a large population yields serious health and economic threats. In this sense, mathematical epidemiology is concerned with modeling the spread of infectious disease in a population [see 2]. The aim is generally to understand the time course of the disease with the goal of controlling its spread. Traditionally, the majority of existing mathematical models to simulate epidemics are based on ordinary differential equations. These models have serious drawbacks in that they neglect the local characteristics of the spreading process and they do not include variable susceptibility of individuals. Specifically, they fail to simulate in a proper way (1) the individual contact processes, (2) the effects of individual behavior, (3) the spatial aspects of the epidemic spreading, and (4) the effects of mixing patterns of the individuals.

Mathematical modeling in epidemiology was pioneered by Bernoulli in 1760. Nevertheless, the work due to Kermack-McKendrick [7] can be considered as the starting point for the design of modern mathematical models. It consists of a SIR model (Susceptible-Infected-Recovered, SIR) which is a set of Ordinary Differential Equations.

Optimization methods have been developed for deterministic simulation models. However, they have not been done with more complex stochastic simulation models. For more details look in [2, 4, 5, 12]. Influenza is transmitted in a complex way from person to person. In addition, given an introduction of influenza into a population, the probability of a major epidemic and the possible size of an epidemic are high-

ly variable. Thus, the mathematical models for influenza epidemics should have a detailed contact structure and be stochastic. Besides, the epidemic process is non-linear since the incidence of new infections depends on the current number of both infectives and susceptibles in the population at a particular time. All of these factors make optimization based on traditional gradient methods, such as the Newton-Raphson method, difficult or even prohibitive. [8] Developed a stochastic approximation method whose convergence is guaranteed under mild conditions. The method, however, requires knowledge of the analytic gradient of the considered objective function [11]. However, in terms of simulation optimization, the drawback of these methods remains the unavailability of gradients. When trying to devise more realistic models we incorporate spatial parameters to better reflect the heterogeneous environment found in nature. An alternative to using deterministic differential equations is to use a two-dimensional cellular automaton (CA) to model location specific characteristics of the susceptible population together with stochastic parameters which captures the probabilistic nature of disease transmission. Cellular Automata model should be treated as a dynamical system that involves a random variable. As a result, the suggested model should be a stochastic dynamical model called probabilistic Cellular Automata model (PCA) which is an extension of CA. The state space remains the same as well as the local and synchronous character of the dynamic. The novelty is that each site updates its value randomly according to a probability distribution which depends on the neighboring sites. Also, we will apply non-uniform cellular automata where each cell satisfies the same or different rules according to which cell states are updated in a synchronous and local manner.

Usually, when a CA-based model is considered to simulate an epidemic spreading, individuals are assumed to be distributed in the cellular space such that each cell stands for an individual of the population as in our case.

In the case of our stochastic two-dimensional non uniform cellular automata model, we consider two possible ways of interaction so, we have two directions to get infected from other cells either horizontal or vertical interaction as each cell stands for an individual. This will reduce the possibility to get infection from others as in every direction there are only two neighbors affecting the central cell in every direction as seen in following sections. Stochastic optimization methods will

be used to model parameters for infectious diseases and population structures. Also, we append every rule with two factors namely certainty and coverage factors to show the non-uniform way where each cell affects another in its neighborhood and vice versa. This method of two-way split rules pave the way for more good choices of solutions as seen in section IV part B.

In this work we consider how we can model the spread of an epidemic using a two-dimensional cellular automaton using programming features of the matrix algebra package MATLAB to develop an implementation that simulates a simple model. By adding our new probabilities of getting infection either from horizontal or vertical motion to the program mentioned in [3]. Also, the effect of population vaccination can be considered in this model. A vaccination parameter  $V \in [0,1]$  must be considered in the objective function of the model. Such parameter stands for the portion of susceptible infected individuals at each time step which are vaccinated. But in the program, the user has the ability to vaccinate specific regions of the environment. This takes input from the user while the simulation is running, allowing vaccination strategies to be tested.

## II. PRELIMINARIES

### A. The Probabilistic Cellular Automata Model (PCA)

In this section, we describe the overall design of our automaton. The model is a two-dimensional grid of cells, each cell containing the same or different rules due to the fact that each epidemic can be in any one of four stages as in the SEIR model we introduced later (i.e. Susceptible, Exposed, Infected and Recovered). The susceptible individuals are those capable of contracting the disease; For many infections there is a period of time during which the individual has been infected but is not yet infectious himself; during this latent period the individual is said to be exposed. In this case we have the SEIR model in which the new class of exposed individuals (E) must be considered. The infectious individuals are those capable of spreading the disease; and the recovered individuals are those immune from the disease, either died from the disease, or having recovered, are definitely immune to it. So, different diagnosis of disease has different rules. The neighbors of any cell in the grid are the cell itself plus the four orthogonal adjacent cells (fig. 1) according to von Neumann.

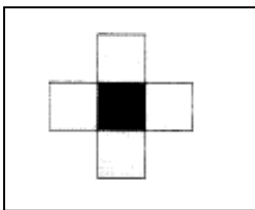


Fig. 1. The two-dimensional cellular automata cell's neighbors

Once the automaton was embedded in the grid, the cell as an individual finite state machine, began to follow the rule that is applied to it. A single cell cannot do much without interacting with other cells. The lattice starts out with an initial configuration of cells in which the following hold.

Each cell can take on one of the 4 different states:

- 1) Susceptible (i.e. healthy; designated as 0)
- 2) Exposed (designated as 1)
- 3) Infected (designated as -1)
- 4) Recovered (includes both survivors and deaths; designated as 2)

- It is supposed that the way of infection is the contact between the infected individual and the healthy individual.
- Once the healthy individuals have contracted the infection and have recovered from it, they acquire temporary immunity.

### B. Mathematical Formulation Of The PCA

Two-dimensional CA are discrete dynamical systems formed by a finite number of identical objects called cells, which are arranged uniformly in a two-dimensional space. Each cell is endowed with a state, belonging to a finite state set, that changes at every discrete step of time according to a rule, called local transition function. More precisely, a CA can be defined as a 4-uplet,  $A = (C, S, N, f)$ , where C is the cellular space formed by a two-dimensional array of  $r \times c$  cells, where r stands for rows and c for columns (see Fig. 2-(a)):  $C = \{(a, b), 1 \leq a \leq r, 1 \leq b \leq c\}$ . The state of each cell is an element of a finite set, S, in such a way that the state of the cell (a, b) at time t is denoted by  $S(a,b;t) \in S$ . The matrix  $C(t) = S(i, j; t)$  is called configuration of the CA at time t. Moreover,  $C(0)$  is the

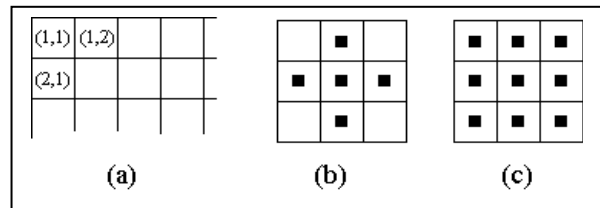


Fig. 2. (a) Cellular space (b) Von Neumann neighborhood (c) Moore neighborhood

initial configuration of the CA. The neighborhood of a cell (a,b) is the set of cells whose states at time t determine the state of the cell (a, b) at time  $t + 1$ , by means of the local transition function. Depending on the process to be modeled, one can choose an appropriate neighborhood. Nevertheless, the traditional neighborhoods considered are the Von Neumann neighborhood (see Fig. 2-(b)), and the Moore neighborhood (see Fig. 2-(c)). Note that the main cell is also considered in its neighborhood. A neighborhood is defined by means of a finite set of indices

$N = \{(x_i, y_i) : 1 \leq i \leq m\} \subset Z \times Z$  N: Neighborhood of a cell c, such that for every cell (a, b), its neighborhood,

$N_{(a,b)}$  is the set of m cells given by  $N_{(a,b)} = \{(a + x_i, b + y_i), \dots, (a + x_m, b + y_m) : (x_k, y_k) \in N\}$ .

Note that for Von Neumann neighborhood, we have  $N = \{(0,0), (-1,0), (0,1), (1,0), (0,-1)\}$ .

Cell's state [10]:

$$S_c(t+1) \in \{0, -1, 1, 2\} \quad (1)$$

$$f: S_c(t+1) \rightarrow S_c(t) \quad (2)$$

Where

$S_c$ : State of the cell at iteration  $t+1$

$f$ : Transition function (Determines how the cell's state can change).

As the cellular space is considered to be finite, boundary conditions must be taken into account in order to assure the well-defined dynamics of the CA.

### C. Probabilistic Analysis of Cellular Automata Rules

With each time step, the state of each individual cell changes according to a set of rules based on the states of the cell's neighbors. These rules can be either deterministic (certain) or stochastic (probability-based/uncertain). To determine this exactly we need to define two essential factors, if the decision rules have the form  $C \rightarrow D$ , meaning IF C THEN D, where C is the condition attribute, and D is the decision attribute of the decision rule then we have ,

The certainty factor of the decision rule  $C \rightarrow D$ , denoted by

$cer(C,D)$ , is defined as :

$$cer(C,D) = p(D/C) \quad (3)$$

Where  $cer(C,D) \in [0,1]$ . If  $cer(C,D)=1$ , then the given

decision rule is a deterministic or certain decision rule .

Otherwise,  $0 < cer(C,D) < 1$ , the given decision rule is a stochastic or uncertain decision rule.

The coverage factor of the decision rule  $C \rightarrow D$  in S, denoted by  $cov(C,D)$ , is defined as:

$$cov(C,D) = p(C/D) \quad (4)$$

In this work we consider a non-uniform case of cellular automata where different cells may contain different rules. At a given moment, only one rule is active for a cell and determines the cell's function. A non-active rules may be activated in next time steps.

Let any cell in the lattice be labeled by its position  $c=(i, j)$

where  $i$  and  $j$  are the row and column indices. A function

$S_c(t)$  is the state of cell  $c$  at time step  $t$ . The rules of the

model specify how the state  $S_c(t+1)$  is to be computed from

the states at time step  $t$  of it's neighbors

$\{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\}$  as in fig. 3:

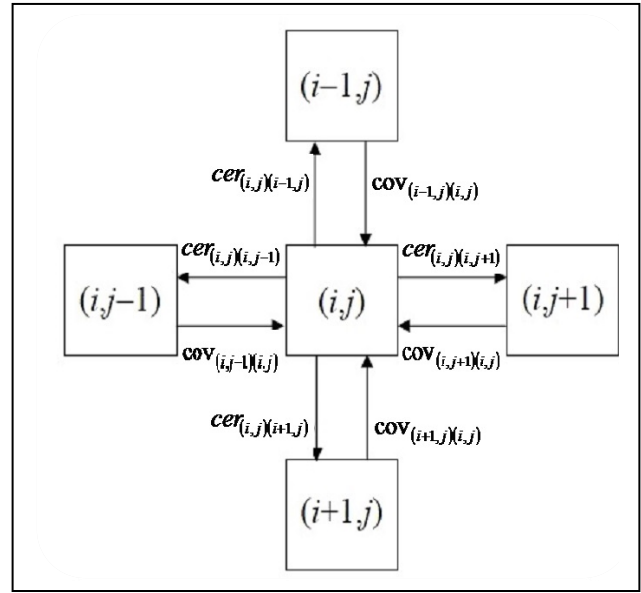


Fig. 3. Parameters of the transition rules.

Here we assume that there are two possible structures either horizontal or vertical interaction between cells each of them has different probability of infection respectively. A small probability 0.0001 of spontaneous infection is assumed. This represents the possibility of external infection e.g. infection due to traveling or imported objects [1].

In the horizontal case, the state of cell  $(i, j)$  depends only on the states of cells  $\{(i, j-1), (i, j+1)\}$  while during vertical interaction it's state depends on states of cells  $\{(i-1, j), (i+1, j)\}$ . So, the automata rules must be formalized for the two directions as follows:

5) For Horizontal interaction (with probability of infection  $P_h$ ):

$$a) \text{ If } S(i, j; t) = 0 \text{ and } S(i, j-1; t) = -1 \text{ or } S(i, j+1; t) = -1 \text{ then } S(i, j; t+1) = -1 \text{ with probability } P_h \quad (5)$$

$$b) \text{ If } S(i, j; t) = -1 \text{ then } S(i, j; t+1) = 2 \quad (6)$$

$$c) \text{ If } S(i, j; t) = 1 \text{ then } S(i, j; t+1) = 2 \quad (7)$$

$$d) S(i, j; t+1) = 1 \text{ (independent of it's interaction neighbors) with probability } 0.0001. \quad (8)$$

6) For Vertical interaction (with probability of infection  $P_v$ ):

a) If  $S(i, j; t) = 0$  and  $S(i, j-1; t) = -1$  or  $S(i, j+1; t) = -1$   
then  $S(i, j; t+1) = -1$  with probability  $P_v$  (9).

b) If  $S(i, j; t) = -1$  then  $S(i, j; t+1) = 2$  (10).

c) If  $S(i, j; t) = 1$  then  $S(i, j; t+1) = 2$  (11).

d)  $S(i, j; t+1) = 1$  (independent of its interaction neighbors) with probability 0.0001 (12).

If a cell is susceptible (0), the simulation module counts the number of infected cells that are its nearest neighbors. The simulation then calculates the probability the cell can avoid being infected  $(1-\beta)^m$ , where  $\beta$  is the infection rate (the probability a contact with an infected person is actually infectious) and  $m$  is the number of infected cells. Subtracting this value from 1 yields the probability of the cell being infected [10].

### III. THE OPTIMIZATION PROBLEM

The optimization problem is as follows: Given a limited quantity of influenza vaccine and a particular population structure and infection rate pattern for a single wave of pandemic influenza, what proportion of each stage should be vaccinated to minimize the impact of the epidemic? [ see :9]

We divide the population into four different cases: Susceptible, Exposed, Infected and Recovered that are indexed as  $i=1, \dots, 4$ . We let  $n_i$  be the number of individuals in class  $i$ , and the total population size is  $n = \sum_{i=1}^4 n_i$ . We let  $\beta$  be the infection rate, i.e. (the probability a contact with an infected person is actually infectious), where  $\beta$  either takes the value  $P_h$  or  $P_v$  depending on the movement direction. We let  $V$  be the total number of vaccine doses available before day one of the epidemic, and  $V_i$  the proportion of individuals in case  $i$  that is vaccinated. Thus, the total number of doses distributed is,

$$\begin{cases} Q = \sum_{i=1}^4 n_i V_i & \text{where,} \\ Q \leq V \end{cases} \quad (13)$$

since we cannot use more vaccine than there is available. We assume that each person vaccinated receives one dose of vaccine. Four different values of the vaccination rate are considered:  $V = 0, 0.2, 0.3, 0.4$ . Note that as  $V$  increases, the number of infected individual decreases. To reflect the impact of a single illness, we let  $w_i$  be the weight assigned to an illness in each case for minimization of the loss function. Then, the optimization problem is expressed as

$$\min \sum_{i=1}^4 \beta n_i w_i \quad (14)$$

Such that

$$\sum_{i=1}^4 n_i V_i \leq V \quad (15)$$

We concentrate on minimizing overall illness in the population as well as number of lives lost given a predetermined number of doses  $V$  of vaccine. We use weights of one, i.e.  $w_i = 1, i = 1, \dots, 4$  for minimizing illness objective function. The rules have been applied based on an optimization algorithm as illustrated in Fig. 4 showing the behavior of the epidemic through different probabilities.

*Pseudocode*

*Overall structure*

*Main function*

Get information from user and initialize variables  
Visualise state

Loop through generations:

- 1) Update count of infected neighbours for each cell
- 2) Update state of each cell based on number of infected neighbours
- 3) Visualize state
- 4) Stop

In this work we consider how we can model the spread of an epidemic using a two-dimensional cellular automaton. We will use programming features of the matrix algebra package MATLAB to develop an implementation that simulates a simple model [3] as shown in fig.4. We want to construct a model of an epidemic that will develop over a fixed  $N$  by  $N$  grid in a given number of generations. We apply a set of rules to each cell that will determine its fate in the next generation, for example whether a given cell will become infected or not. The probabilities of state changes are a set of predefined parameters. By studying the effects of varying these parameters we try to predict how the epidemic will develop over time will it eventually die out.

We apply the same strategy of vaccination used by [3] where the user have the ability to vaccinate specific regions of the environment. This takes input from the user while the simulation is running, allowing vaccination strategies to be tested. Since vaccination is not usually permanent but lasts for a period of time, it is implemented in much the same way as the temporary immunity that follows organism recovery. While the program is running, the user can click on the grid causing vaccination to be simulated at the clicked point.

```

Pseudo code for main program
Initialize constants to represent cell states (Susceptible,
Exposed, Infected, and Recovered)
Input N, the array size to use
Input  $P_h$  and  $P_v$ , the probabilities of becoming infected
during horizontal and vertical interaction
Input  $Q$ , the probability of becoming recovered.
Optionally input ca_state, the state matrix, from given
initial configuration
If no initial configuration was supplied:
    Initialize ca_state with all cells Susceptible
    Set randomly-selected cells near centre to Infected
Add a border of immune cells to ca_state
Visualize ca_state
Initialize infected_neighbours, the infected neighbour
count matrix
Loop through generations
    Loop through valid x-coordinates (2 to N+1)
        Loop through valid y-coordinates (2 to N+1)
            Update count of infected_neighbours(x,
y)
        End loop through y-coordinates
    End loop through x-coordinates
    Loop through valid x-coordinates (2 to N+1)
        Loop through valid y-coordinates (2 to
N+1)
            Update ca_state(x, y) based on in-
fected_neighbours(x, y)
        End loop through y-coordinates
    End loop through x-coordinates
    Visualize ca_state
End loop through generations
    
```

Fig. 4. Pseudo code for main program

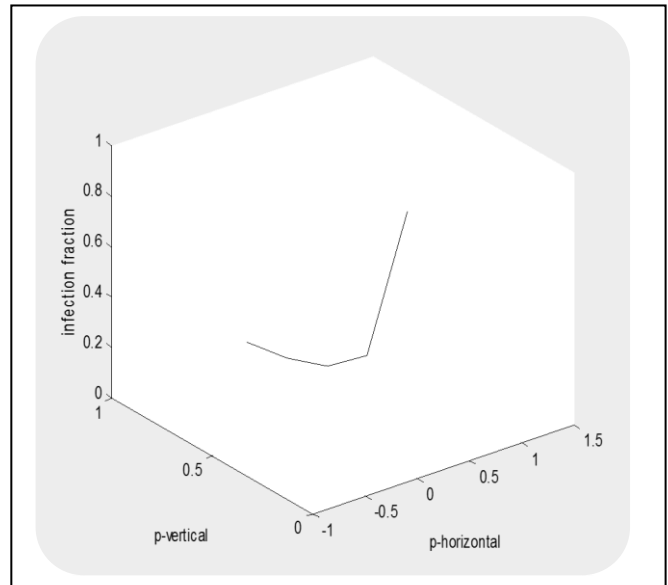
#### IV. SIMULATION AND ANALYSIS

##### A. Different scenarios based on varying $p$ -horizontal ( $P_h$ ) and $p$ -vertical ( $P_v$ )

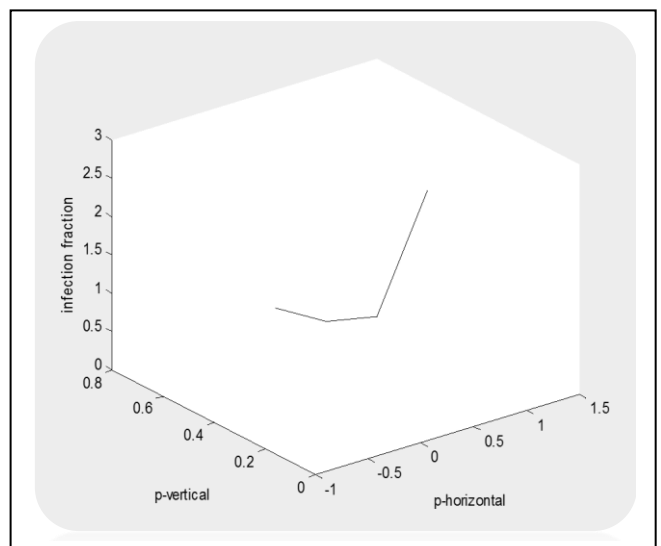
Because our cellular automaton is probabilistic (i.e. random numbers affect the chances of different scenarios arising,

even with fixed  $P_h$ ,  $P_v$  and  $Q$ ) it is necessary to run several simulations to obtain an approximate understanding of what should happen to the spread of the epidemic. We also need to test several different values of  $P_h$  and  $P_v$ . Initially, we will set  $Q$  as a constant 1 and vary  $P_h$ ,  $P_v$ . Thus an Infected cell will immediately become Recovered. We will test for  $P_h$ ,  $P_v \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ .

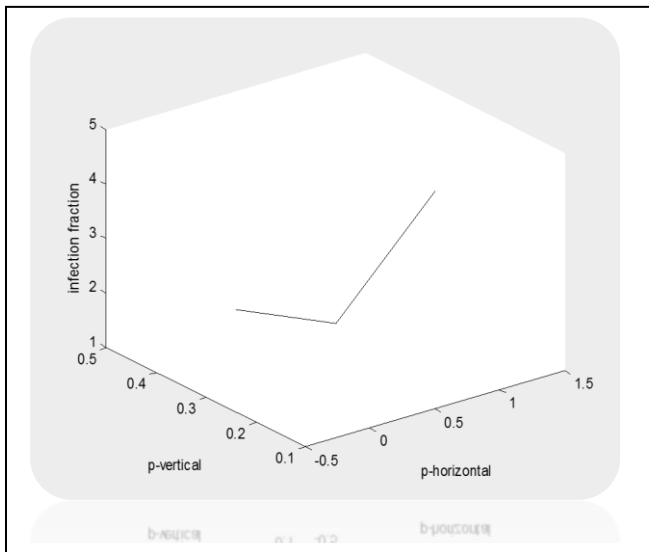
Case 1:  $P_h = 0.1$ ,  $P_v = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ .



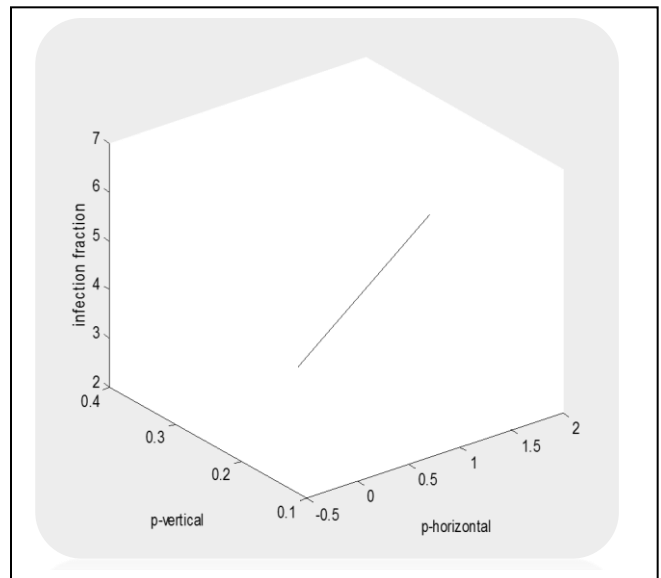
Case2:  $P_h = 0.3$ ,  $P_v = \{0.1, 0.3, 0.5, 0.7\}$ .



Case 3:  $P_h = 0.5$ ,  $P_v = \{0.1, 0.3, 0.5\}$ .



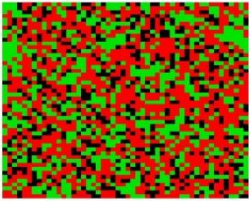
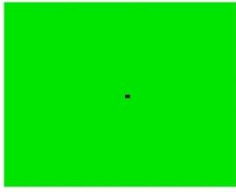
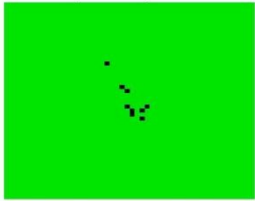
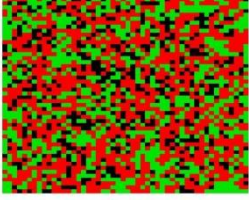

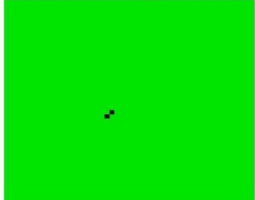
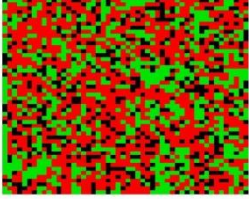
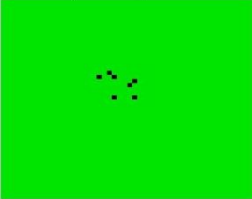
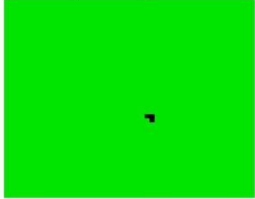
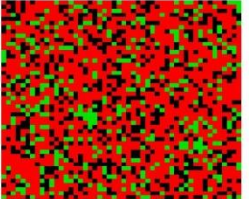
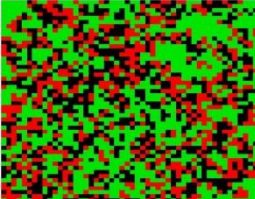
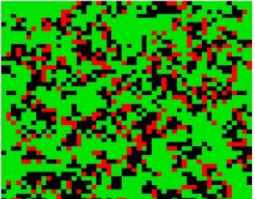
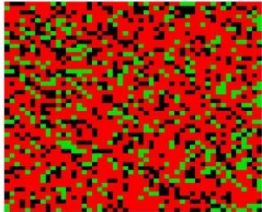
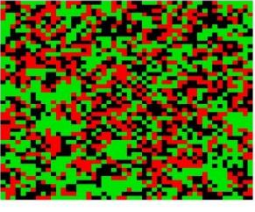
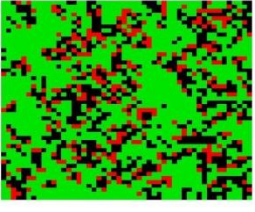
Case 4:  $p_h = 0.7, p_v = \{0.1, 0.3\}$ .

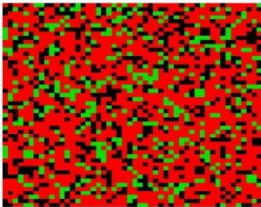
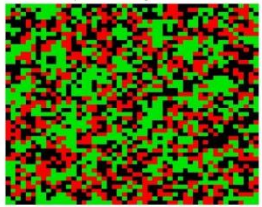
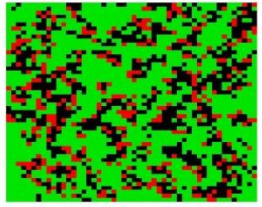
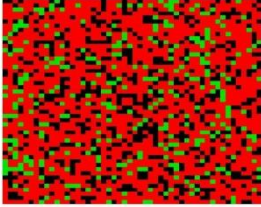
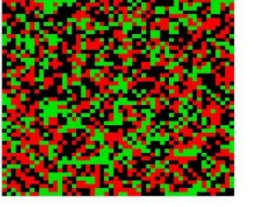



B. We run the simulations for 100 generations and observe the behavior of the disease in a grid of size  $50 \times 50$ .

TABLE I. Description of observed behaviour for a range of values of  $p_h, p_v$  and  $q$  (green is Susceptible, red is Infected, and black is Recovered)

$q$ $p_h, p_v$	0.2	0.6	1
0.1, 0.1	 Epidemic simulation - generation 100 Chaotic pattern emerges rapidly. Almost all cells Infected at any one time.	 Epidemic simulation - generation 66 disease dies out followed by all cells Susceptible within approx 66 generations.	 Epidemic simulation - generation 5 disease dies out followed by all cells Susceptible within approx 5 generations.
0.1, 0.3	 Epidemic simulation - generation 100 Chaotic pattern emerges rapidly. Almost all cells Infected at any one time.	 Epidemic simulation - generation 16 disease dies out followed by all cells Susceptible within approx 16 generations.	 Epidemic simulation - generation 4 disease dies out followed by all cells Susceptible within approx 4 generations.

<p>0.1 , 0.5</p>	<p>Epidemic simulation - generation 100</p>  <p>Chaotic pattern emerges rapidly. Almost all cells Infected at any one time.</p>	<p>Epidemic simulation - generation 30</p>  <p>disease dies out followed by all cells Susceptible within approx 30 generations.</p>	<p>Epidemic simulation - generation 4</p>  <p>disease dies out followed by all cells Susceptible within approx 4 generations</p>
<p>0.1 , 0.7</p>	<p>Epidemic simulation - generation 100</p>  <p>Chaotic pattern emerges rapidly. Almost all cells Infected at any one time.</p>	<p>Epidemic simulation - generation 66</p>  <p>disease dies out followed by all cells Susceptible within approx 66 generations</p>	<p>Epidemic simulation - generation 9</p>  <p>disease dies out followed by all cells Susceptible within approx 9 generations</p>
<p>0.1 , 0.9</p>	<p>Epidemic simulation - generation 100</p>  <p>Chaotic pattern emerges rapidly. Almost all cells Infected at any one time.</p>	<p>Epidemic simulation - generation 14</p>  <p>disease dies out followed by all cells Susceptible within approx 14 generations</p>	<p>Epidemic simulation - generation 9</p>  <p>disease dies out followed by all cells Susceptible within approx 9 generations</p>
<p>0.3 , 0.3</p>	<p>Epidemic simulation - generation 100</p>  <p>Chaotic pattern emerges Spreads more quickly.</p>	<p>Epidemic simulation - generation 100</p>  <p>Chaotic pattern emerges Spreads slowly.</p>	<p>Epidemic simulation - generation 100</p>  <p>Chaotic pattern emerges Spreads more slowly.</p>
<p>0.3 , 0.5</p>	<p>Epidemic simulation - generation 100</p>  <p>Chaotic pattern emerges Spreads more quickly.</p>	<p>Epidemic simulation - generation 100</p>  <p>Chaotic pattern emerges Spreads rapidly.</p>	<p>Epidemic simulation - generation 100</p>  <p>Chaotic pattern emerges Spreads more slowly.</p>

0.3 , 0.7	 Chaotic pattern emerges Spreads more quickly.	 Chaotic pattern emerges Spreads rapidly.	 Chaotic pattern emerges Spreads more slowly.
0.5 , 0.5	 Chaotic pattern emerges Spreads more quickly.	 Chaotic pattern emerges Spreads rapidly.	 Chaotic pattern emerges Spreads slowly.

Note that: the figures shown in table1 will be a bit different every time we execute the program as the vaccine is given randomly by each user.

## V. CONCLUSION

These epidemic scenarios presented above provide an opportunity to demonstrate the visualization capabilities of a graphical CA model. When [3] use a single probability of infection, they got only two results in which all cells are in susceptible state. But, here when dividing the cellular space into two directions of motion with two different probabilities of infection, there are more optional values for  $P_h, P_v$  differ at which generation it is obtained as shown above in table1. We introduced a theoretical model to simulate the spreading of an epidemic. It is based on transferring the problem into parametric one and the rules into restrictions, obtaining the optimization problem (14). It is solved with the chosen values of  $P_h, P_v$  and  $q$  get the minimum value of the function which minimize the impact of the epidemic.

## VI. FUTURE WORK

PCA are lattice model of spatially extended systems with probabilistic local dynamical rules of evolution. It is difficult to analyze rigorously, so the computational simulation provide an alternative tool. So the future work is to develop an iterative Probabilistic Neural Networks with fully parallel probabilistic feedback dynamic. In addition, Parallel Genetic Algorithms can be incorporated by modifying the probabilities. PGA search through the space of parameters to calibrate the model to observe data. Whilst, PNN synthesis of approaches based on prior analysis and contextual information.

## ACKNOWLEDGMENT

We thank Prof. E. Ahmed, Mathematics Dept., Faculty of Science, Mansoura University, Egypt, for his comments. Also,

special thanks for MATLAB Developers, where matrix algebra package is used for implementing that simulation model.

## REFERENCES

- [1] Ahmed E., Agiza H.N. , (1998) On modeling epidemics. Including latency, incubation and variable susceptibility, *Physica A* 253 347-352.
- [2] Anderson, R., May, R., 1991. *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press, New York.
- [3] Andrew Hawkins, Danny Roff, Adam Gundry, 2005. *Cellular Automata and Spatial Epidemics*. November 25. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.105.1458&rep=rep1&type=pdf>.
- [4] Greenhalgh, D., 1986. Control of an epidemic spreading in a heterogeneously mixing population. *Math. Biosci.* 80, 23–45.
- [5] Hethcote, H., Waltman, P., 1973. Optimal vaccine schedules in deterministic epidemic models. *Math. Biosci.* 18, 365–381.
- [6] Hoya White S., Marti'n del Rey A., Rodri'guez Sa'nchez G., (2007). Modeling epidemics using cellular automata. *Applied Mathematics and Computation* 186 193–202.
- [7] Kermack W.O., McKendrick A.G., *Proc. Roy. Soc. Edin.* (1927) A 115 700.
- [8] Robbins, H., Munro, S., 1951. A stochastic approximation method. *Ann. Math. Statist.* 22, 400–407.
- [9] Patel Rajan, Ira M. Longini Jr., M. Elizabeth Halloran, (2005) Finding optimal vaccination strategies for pandemic influenza using genetic algorithms. *Journal of Theoretical Biology* 234. 201–212.
- [10] Wang Jeffrey B., 2010. School's Out? Designing Epidemic Containment Strategies with a spatial stochastic method. <http://www.dancingdarwin.com/sca>.
- [11] Weisstein, E., 2004. Robbins–munro stochastic approximation. *MathWorld*; <http://mathworld.wolfram.com/Robbins–MonroStochasticApproximation.html>.
- [12] Wickwire, K.H., Guest, D., 1976. Optimal control policies for reducing the maximum source of a closed epidemic. *Math. Biosci.* 32, 1–14.

## AUTHORS PROFILE

Wesam Elsayed, Department of Mathematics, Faculty of Science, Mansoura University, Mansoura, 35516, Egypt



E-mail address: w\_mahmoud93@yahoo.com



Ahmed El-bassiouny , received his B.Sc. and M.Sc. from Mathematics Dept. Faculty of Science, Mansoura University, Egypt. He received his Ph.D. from University of Kiev, Soviet Union. He became a Lecturer at Mathematics Dept., Faculty of Science, Mansoura University, Egypt on

2004.

E-mail address: el\_bassiouny@mans.edu.eg



Elsayed Radwan, received his B.Sc. and M.Sc. from Mathematics Dept. Faculty of Science, Mansoura University, Egypt. He received his Ph.D. from Intelligent Lab., Graduate School of Engineering, Toin University of Yokohama, Japan. He became a Lecturer at Computer Science Dept., Faculty of Computer and Information Sciences, Mansoura University, Egypt on 2007. His areas of interest are Soft Computing, Evolutionary Computing, Pattern Recognition, and Cellular Neural Networks.

E-mail address: elsfradwan@yahoo.com