Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

AN ANALYSIS OF EMERGING BEHAVIORS IN LARGE-SCALE QUEUEING-BASED SERVICE SYSTEMS USING AGENT-BASED SIMULATION

Wai Kin (Victor) Chan

Department of Decision Sciences and Eng. Systems
Rensselaer Polytechnic Institute
CII 5015, 110 Eighth Street
Troy, NY 12180 U.S.A.

ABSTRACT

This paper considers a large-scale service system consisting of a number of service areas (cells). Each cell contains a queueing model that operates continuously and independently from the queueing models in other cells. Each cell changes its state between alive and dead based on certain rules that depend on the queueing status of its own queue, the neighboring queues, and the whole community, while satisfying a constraint on the number of live cells in a neighborhood. The objective is to examine emerging behaviors from the interactions of the cells under various rules. Chaotic, deterministic, and in-between emerging behaviors are presented.

1 INTRODUCTION

A complex system often contains a large set of interacting individual sub-systems, each of which could be an autonomous, goal-driven, and adaptive intelligent entity (agent). For example, a large-scale healthcare network could consist of a number of objects, such as hospitals, medical supply warehouses, blood banks, etc. In many cases, each object (agent), although operating based on its own strategies, cooperates with each other in accordance with certain mutual rules or agreements. Real-world examples can be found in blood supply chain (Katsaliaki and Brailsford 2006, Mustafee et al. 2006), organ transplantation supply chain (Pritsker et al. 1995), general supply chains (Pathak et al. 2007), electricity markets (Conzelmann et al. 2004, Koritarov 2004), and social networks (Newman et al. 2006).

Because of the agents' autonomous characteristic, people have been applying different kind of decentralized methods to analyze the behaviors of these complex systems. Agent-based simulation (ABS) is one of such methods that has been developed rapidly over the past decade. The basic idea of ABS is to simulate these real-world systems with a group of interacting agents modeled as computer programs to study the interactions of the agents and/or emerging system behaviors. The theoretical basis of ABS lies mainly in c*ellular automata* (*CA*), *complex system modeling* (*CSM*), *artificial life* (*AL*), and *swarm intelligence* (*SI*) (Bonabeau et al. 1999, Macal and North 2005).

The work in CA is pioneered by Von Neumann (1966) and Stanislaw M. Ulam (Beyer et al. 1985). CA considers a set of cells residing in an n-dimensional lattice. It studies complex behaviors emerging from some simple rules that govern the interactions of the cells. In a conventional CA, each cell is nothing but an object following only the rules. One of the famous models is the *Conway's Game of Life* (Berlekamp et al. 1985), in which the only action that each cell performs is to change its state based on the number of neighboring live cells. See Toffoli and Margolus (1987) and Wolfram (1983) for more details on CA.

Comparing ABS and CA, one can see that ABS provides more flexibilities and functionalities (such as learning and adaptation) in each agent and can accommodate more complicated interactions and rules between the agents and their environment. Nevertheless, being able to produce complex behaviors using simple rules is the beauty of CA and one of the main reasons for its popularity and broad applications in many areas (Wolfram 2003).

The research reported here lies in between conventional CA and general ABS. We retain the basic CA lattice-form of environment and try to keep the interaction rules simple; yet we move a step closer to general ABS by allowing the cells to model more realistic real-world objects and using more general interaction rules.

Specifically, we generalize the basic framework of CA to allow each cell to be a queueing node. The interactions among the cells are based on the status of its queue, its neighbors' queues, the whole community's queues, and the state of its neighboring cells. Therefore, the cells' transition rules are no longer fixed, but change as the system evolves. Details of the model are provided in the next section.

978-1-4244-2708-6/08/$25.00 ©2008 IEEE

The goal of this work is to construct a conceptual decentralized model to examine emerging behaviors from a large set of queueing systems. Each queueing system can be viewed as a simplified representation for a healthcare unit/facility, service center, or any supply-demand chain in which a certain type of stochastically arriving demand is fulfilled by a set of resources. The objective of this research is to study the interactions among these large number of queues.

Emerging behaviors here refer to a certain pattern of live and dead cells distribution—a live cell means a service center should be built while a dead cell implies the service center should be shut down. Identifying these patterns provides values to decision-makers in designing where to set up these centers and how they evolve in time. In the case of a chaotic pattern, a temporal study on the number of live cells within a particular region might reveal the scarcity of the supply in that region. People doing market study might also be interested in the distribution and the related statistics of the live and dead centers estimated from these evolving patterns.

The organization of this paper is as follows. Section 2 defines the model and the interaction rules. Section 3 presents some emerging behaviors, including chaotic, deterministic, and in-between patterns. Section 4 draws a conclusion on this study and suggests future extensions.

2 THE MODEL

The model includes an environment, a set of cells (each contains a queue), and the interaction rules among the cells.

The *environment* is a two-dimensional lattice space considered in conventional CAs. Each grid of the lattice is a *cell*. Each cell contains one queueing system whose input process and output process are independent from other queues. Figure 1 shows an example with 9 cells and 9 queues. The model *evolution* (*interactions of cells*) is carried out by incorporating three types of information (self, local, and global) and one constraint. *Self information* refers to the queueing status of a cell. *Local information* is the queueing status of the neighboring queues. *Global information* represents the overall queueing status from the whole community. The *constraint* is a set of rules that each cell must follow. For example, a constraint could be "there must be at last one live cell within a neighborhood." The *time* of the model is continuous. All queues are running continuously without interruption. However, the interactions of cells take place only at discrete time points, i.e., every single time unit (see more details in the following).

The formal definition of the model is provided in the following. The model space, $\mathbb{L}$, is a two-dimensional lattice consisting of $r \times c$ cells, i.e., $\mathbb{L} = \{(i,j) \mid i = 1,\dots;\ j = 1,\dots,c\}$. Each cell, C_{ij}, is occupied by a queue, Q_{ij}, independent from all other queues. Let $\mathbb{Q}$ and $\mathbb{N}$ be, respectively, the set of all the queues and the set of neighbors corresponding to each queue, i.e., $\mathbb{Q} = \{Q_{ij} \mid (i,j) \in \mathbb{L}\}$ and $\mathbb{N} = \{\boldsymbol{N}_{ij} \mid (i,j) \in \mathbb{L}\}$. Denoted by $\mathbb{T}$ the set of transition rules associated with each queue, i.e., $\mathbb{T} = \{\boldsymbol{T}_{ij} \mid (i,j) \in \mathbb{L}\}$.

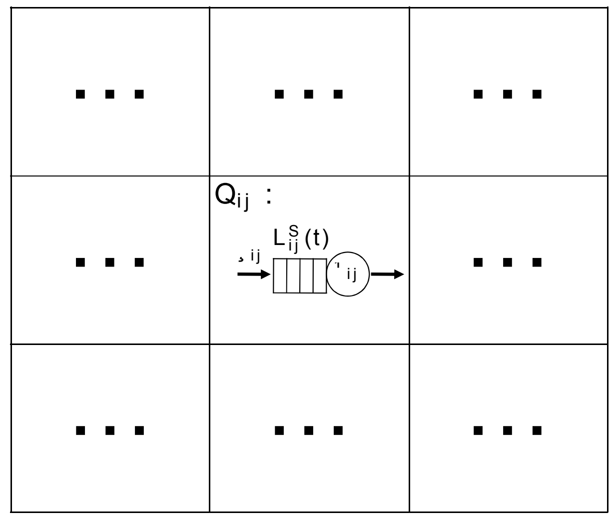


Figure 1: A Model with 9 Cells

The agent-based queueing network can be expressed as:

$$\mathbb{N} = \{\mathbb{Q}, \mathbb{N}, \mathbb{T}, \mathbb{L}\}.$$

Each queue is a generic queueing model. It can be, for example, a certain service system with resources to satisfy a particular type of demand. Let $\mathcal{F}^{\lambda}_{ij}(\lambda_{ij})$ and $\mathcal{F}^{\mu}_{ij}(\mu_{ij})$ be the distributions of the demand and the service, respectively. Each queue has R_{ij} resources (i.e., servers). The notation for each queue is:

$$Q_{ij} = \{\mathcal{F}^{\lambda}_{ij}(\lambda_{ij}), \mathcal{F}^{\mu}_{ij}(\mu_{ij}), R_{ij}\}.$$

The Moore neighborhood is used in this paper. Therefore, the neighbors of Q_{ij} are represented by

$$\boldsymbol{N}_{ij} = \{Q_{kl} \mid (k,l) \neq (i,j) \in \mathbb{L}, |k-i| \leq 1, |l-j| \leq 1\}.$$

Since the time of the model is continuous, each queue resembles a conventional queueing system. However, model evolution is carried out at discrete time points, i.e., every one-unit time interval. As a result, the status of each queue is sampled only at $t = 1, 2, \dots$.

This model uses three factors and one constraint to determine the state of a cell. They are the factors coming from the cell itself (*self*), its neighbor (*local*), the whole population (*global*), and the constraint restricting the number of life cells in a neighborhood. In the following, we first introduce these factors and the constraint one by one and then discuss how to consolidate them to determine the final state of a cell.

The self factor is a cell's queue size, which represents the degree of demand for the service inside a cell. Let $L_{ij}^{S}(t)$ be the queue size of Q_{ij} at sample time t. This self factor influences the cell's state by the following rule: (1) C_{ij} should die at the beginning of time $t+1$ if its queue size is smaller than a threshold, X_{ij}^{S}, at the end of time t; (2) C_{ij} should be alive at the beginning of time $t+1$ if its queue size is larger than a threshold, Y_{ij}^{S}, at the end of time t; and (3) C_{ij} remains its state at the beginning of time $t+1$ if its queue size is between X_{ij}^{S} and Y_{ij}^{S}, inclusively, at the end of time t. Denoted by $f_{ij}^{S}(t)$ the self factor, which equals –1 if it suggests that the cell should die, 0 if unchanged, and 1 if alive. We have

$$f_{ij}^{S}(t+1)=\begin{cases}-1 & L_{ij}^{S}(t)<X_{ij}^{S}\\ 0 & X_{ij}^{S}\le L_{ij}^{S}(t)\le Y_{ij}^{S}.\\ 1 & Y_{ij}^{S}<L_{ij}^{S}(t)\end{cases}$$

The local factor comes from a cell's neighbors. Let $L_{ij}^{L}(t)$ be the average queue size of all C_{ij}'s neighbors, i.e.,

$$L_{ij}^{L}(t)=\frac{1}{|\boldsymbol{N}_{ij}|}\sum_{\forall Q_{kl}\in\boldsymbol{N}_{ij}}L_{kl}^{S}(t).$$

This local factor influences C_{ij}'s state by the following rule: (1) C_{ij} should die at the beginning of time $t+1$ if its average neighbor queue size is smaller than a threshold, X_{ij}^{L}, at the end of time t; (2) C_{ij} should be alive at the beginning of time $t+1$ if its average neighbor queue size is larger than a threshold, Y_{ij}^{L}, at the end of time t; and (3) C_{ij} remains its state at the beginning of time $t+1$ if its average neighbor queue size is between X_{ij}^{L} and Y_{ij}^{L}, inclusively, at the end of time t. Denoted by $f_{ij}^{N}(t)$ the local factor, which equals –1 if it suggests that the cell should die, 0 if unchanged, and 1 if alive. We have

$$f_{ij}^{L}(t+1)=\begin{cases}-1 & L_{ij}^{N}(t)<X_{ij}^{L}\\ 0 & X_{ij}^{L}\le L_{ij}^{N}(t)\le Y_{ij}^{L}.\\ 1 & Y_{ij}^{L}<L_{ij}^{N}(t)\end{cases}$$

The global factor is the average queue size of the whole community, denoted by $L^{G}(t)$, which is

$$L^{G}(t)=\frac{1}{|\mathbb{N}|}\sum_{\forall Q_{kl}\in\mathbb{N}}L_{kl}^{S}(t).$$

Similar to the self and local factors, the global factor affects C_{ij}'s state by the following rule: (1) C_{ij} should die at the beginning of time $t+1$ if the average global queue size is smaller than a threshold, X_{ij}^{G}, at the end of time t; (2) C_{ij} should be alive at the beginning of time $t+1$ if the average global queue size is larger than a threshold, Y_{ij}^{G}, at the end of time t; and (3) C_{ij} remains its state at the beginning of time $t+1$ if the average global queue size is between X_{ij}^{G} and Y_{ij}^{G}, inclusively, at the end of time t. Denoted by $f_{ij}^{G}(t)$ the global factor, which equals –1 if it suggests that the cell should die, 0 if unchanged, and 1 if alive. We have

$$f_{ij}^{G}(t+1)=\begin{cases}-1 & L^{G}(t)<X_{ij}^{G}\\ 0 & X_{ij}^{G}\le L^{G}(t)\le Y_{ij}^{G}.\\ 1 & Y_{ij}^{G}<L^{G}(t)\end{cases}$$

Observe that X_{ij}^{G} and Y_{ij}^{G} allow different cells to have different degrees of assessment of the global factors; therefore, although $L^{G}(t)$ is the same for all cells, $f_{ij}^{G}(t)$'s are distinct.

Another important factor influencing the state of a cell is the number of neighboring live cells. In reality, it is uncommon to have several service centers next to each others. This leads to the following constraint (rule) governing the number of live cells within a Moore neighborhood: (1) C_{ij} will die at the beginning of time $t+1$ if the number of neighboring live cells is greater than a threshold, Y_{ij}^{C}, at the end of time t (i.e., there are too many service centers around); (2) C_{ij} will be alive at the beginning of time $t+1$ if the number of neighboring live cells is smaller than a threshold, X_{ij}^{C}, at the end of time t (i.e., there are too few service centers around); and (3) C_{ij} remains it's state at the beginning of time $t+1$ if the number of neighboring live cells is between X_{ij}^{C} and Y_{ij}^{C}, inclusively, at the end of time t. Denoted by $f_{ij}^{C}(t)$ the factor representing the neighboring live cells, which equals –1 if it forces the cell to die, 0 if unchanged, and 1 if alive. We have

$$f_{ij}^{C}(t+1)=\begin{cases}1 & A_{ij}(t)<X_{ij}^{C}\\ 0 & X_{ij}^{C}\le A_{ij}(t)\le Y_{ij}^{C},\\ -1 & Y_{ij}^{C}<A_{ij}(t)\end{cases}\qquad(1)$$

where $A_{ij}(t)=\{Q_{kl}\mid Q_{kl}\in N_{ij},S_{ij}(t)=1\}$ is the set of live neighbors of C_{ij} at time t and $S_{ij}(t)$ is the state of C_{ij} and is equal to 1 if C_{ij} is alive and –1 otherwise. It should be pointed out that in this paper $S_{ij}(t)$ is only an indicator of the state of C_{ij}; it does not enable nor disable its queue;

that is, Q_{ij} keeps running normally no matter whether C_{ij} is alive or not.

For simplicity, the effects of these three factors and the constraint are combined by weighting each of them and summing them up. Let F_{ij} be the liveness (total factor) governing the state of C_{ij}. Then, we have

$$F_{ij}(t) = \alpha_S f_{ij}^S(t) + \alpha_L f_{ij}^L(t) + \alpha_G f_{ij}^G(t) + \alpha_C f_{ij}^C(t),$$

where α_S, α_L, α_G, and α_C are the weighted coefficients for the self, local, global, and constraint factors, respectively. It is assumed that $\alpha_S + \alpha_L + \alpha_G = 1$ and $\alpha_C > 1$ so that the constraint is enforced.

To make the model more realistic, a constant sojourn time that a cell must stay in a particular state upon entering is assumed. The reasoning is that this can avoid repeatedly opening and closing a service center at every two consecutive iterations. Let D_{ij} be the constant sojourn time that a cell must remain in its current state and d_{ij} be the number of iterations (times of intervals) since C_{ij}'s last state change.

Summarizing the discussion above, the state of C_{ij} is determined by the following rule: (1) C_{ij} will die at the beginning of time $t+1$ if its state has been changed for more than D_{ij} iterations and its liveness at time t is smaller than a threshold, X_{ij}^F; (2) C_{ij} will be alive at the beginning of time $t+1$ if its state has been changed for more than D_{ij} iterations and its liveness at time t is greater than a threshold, Y_{ij}^F; and (3) C_{ij} will remain in its current state at the beginning of time $t+1$ if its state has been changed for less than D_{ij} iterations or its liveness at time t is between X_{ij}^F and Y_{ij}^F, inclusively. Formally, this rule is

$$S_{ij}(t+1) = \begin{cases} -1 & d_{ij} > D_{ij} \text{ and } F_{ij}(t) < X_{ij}^F \\ S_{ij}(t) & d_{ij} \le D_{ij} \text{ or } X_{ij}^F < F_{ij}(t) < Y_{ij}^F. \\ 1 & d_{ij} > D_{ij} \text{ and } Y_{ij}^F < F_{ij}(t) \end{cases}$$

3 EMERGING BEHAVIORS

The model defined in the previous section does not make any assumptions on the queueing systems. However, for simplicity, in this section, we implement the model using 30×30 (= 900) *M*/*M*/1 queues. The software platform, AnyLogic (Borshchev et al. 2002), is used to create the model. Due to the model size and the number of parameters it has, we limit our experimental study on evaluating how the values of X^C and Y^C affect the distribution of live and dead cells. Since each queue is an *M*/*M*/1, the expected queue size is $E[Q_{ij}] = \rho_{ij}^2/(1-\rho_{ij})$ with a variance equal to $Var\{Q_{ij}\} = \rho_{ij}^2(1+\rho_{ij}-\rho_{ij}^2)/(1-\rho_{ij})^2$, where $\rho_{ij} = \lambda_{ij}/\mu_{ij}$ is the traffic intensity.

The values of the thresholds, X_{ij}^S, Y_{ij}^S, X_{ij}^L, Y_{ij}^L, X_{ij}^G, and Y_{ij}^G are determined based on the argument that a service center should be built [closed] if the queue size is larger [smaller] than the mean by a certain percent of the standard deviation. In the example below, 30% of the standard deviation is selected. For instance, $f_S(t+1) = -1$ if $L_{ij}^S < X_{ij}^S = E[Q_{ij}] - 0.3\sqrt{Var\{Q_{ij}\}}$, $f_S(t+1) = 1$ if $L_{ij}^S > Y_{ij}^S = E[Q_{ij}] + 0.3\sqrt{Var\{Q_{ij}\}}$, and $f_S(t+1) = 0$ otherwise. The model data is provided in Figure 2 and the experimental results are displayed in Figure 5 in Appendix A.

Parameter/Quantity	Value
r	30
c	30
$\lambda_{ij} = \lambda, \forall(i,j) \in \mathbb{L}$	0.9
$\mu_{ij} = \mu, \forall(i,j) \in \mathbb{L}$	1
$R_{ij} = R, \forall(i,j) \in \mathbb{L}$	1
$E[Q_{ij}] = E[Q], \forall(i,j) \in \mathbb{L}$	8.1
$Var[Q_{ij}] = Var[Q], \forall(i,j) \in \mathbb{L}$	88.29
$X_{ij}^S = X^S, \forall(i,j) \in \mathbb{L}$	5.28
$Y_{ij}^S = Y^S, \forall(i,j) \in \mathbb{L}$	10.92
$X_{ij}^L = X^L, \forall(i,j) \in \mathbb{L}$	5.28
$Y_{ij}^L = Y^L, \forall(i,j) \in \mathbb{L}$	10.92
$X_{ij}^G = X^G, \forall(i,j) \in \mathbb{L}$	5.28
$Y_{ij}^G = Y^G, \forall(i,j) \in \mathbb{L}$	10.92
$X_{ij}^C = X^C, \forall(i,j) \in \mathbb{L}$	X^C (to be varied)
$Y_{ij}^G = Y^G, \forall(i,j) \in \mathbb{L}$	Y^C (to be varied)
α_S	0.5
α_L	0.4
α_G	0.1
α_C	3
$X_{ij}^F = X^F, \forall(i,j) \in \mathbb{L}$	-0.3
$Y_{ij}^F = Y^F, \forall(i,j) \in \mathbb{L}$	0.3
$D_{ij}, \forall(i,j) \in \mathbb{L}$	$D_{ij} \sim \text{UNIF}\{1,10\}$
α_G	0.1
α_C	3

Figure 2: Experimental Data

The experiment is performed by varying the two thresholds, (X^C, Y^C), in the constraint for the number of live neighboring cells, including the following cases: (0, 0), (0, 1), (1, 1), (1, 2), (2, 2), (2, 3), (3, 3), (3, 4), (4, 4), (4, 5), (5, 5), (5, 6), (6, 6), (6, 7), (7, 7), (7, 8), (8, 8). Figure 5 shows the distributions of live cells for all these cases. In Figure 5, different colors are used to visualize the number of live neighboring cells. The color schemes are shown in Figure

3. If the cell is dead, then the color is white (or null). All distributions in Figure 5 are obtained after 20,000 iterations.

Number of Live Neighbors	Color
0	Light gray
1	Gray
2	Blue
3	Red
4	Green
5	Magenta
6	Orange
7	Yellow
8	Cyan

Figure 3: Color Schemes for Cells with Different Number of Live Neighbors

It can be seen from Figure 5 that Cases (0, 0) to (2, 3) and cases (5, 6) to (8, 8) produce patterns in seemingly chaotic forms, which also evolve in an unforeseeable format.

Case (4, 4) yields a quite stable pattern after around 20K iterations. This pattern contains several squares expanding equally from the center to the boundary of the environment, without connecting to each other. This also creates two diagonal lines of live cells in red since these red cells have 3 live neighbors (while all the rest live cells, in blue, have 2). This pattern is stable mainly because the thresholds, $\left(X^C, Y^C\right)$ = (4, 4), force all dead cells to have 6 live neighbors and the live cells to have 2 live neighbors (the blue cells) or 3 live neighbors (the red cells on the diagonal lines), keeping all dead cells to die and live cells to live from iteration to iteration (See Figure 5). Although the queue sizes of all the *M*/*M*/1 queues keep changing at every iteration, the whole system has reached an equilibrium (or the lowest-energy point), in which the constraint in Eq. (1) is always binding at $f_{ij}^C(t) = -1$ or $f_{ij}^C(t) = 1$, causing the constraint factor, $f_{ij}^C(t)$, to dominate all the other terms in $F_{ij}(t)$ and subsequently, determine the state change of each cell, $S_{ij}(t)$. It is also observed that this stationary pattern can be reached from all arbitrary initial states that we have simulated.

On the other hand, Cases (3, 4), (3, 3), (4, 5), and (5, 5) generate some patterns that lie in between chaotic and deterministic—that is, "*on the edge of chaos*" (Langton 1990). These patterns all have four curly lines, each starting at the center and ending at one of the corners of the environment. These four curly lines are neither fixed nor periodic (within our experiment length), but evolve from iteration to iteration and never disappear. Figure 4 in Appendix A shows four snapshots of Case (5, 5) at iterations 20K, 80K, 160K, and 320K.

4 CONCLUSION AND FUTURE WORK

Large-scale service or supply-chain systems often exhibit behaviors emerging from different regulations or rules. Some of these behaviors are foreseeable, yet they have been largely ignored due to leaking knowledge about the complex interaction consequence from a large amount of intelligent objects. See the examples given in the literatures cited in Section 1.

This paper is one attempt to understand these complex interactions. A set of 900 queueing models are considered and their interactions are examined. Several emerging behaviors are observed from different rules. These include behaviors of chaotic, deterministic, and on-the-edge-of-chaos. The latest is of particular interest because the evolution follows certain non-deterministic and non-totally stochastic form, resembling some of the real-world phenomena.

There is certainly a fidelity issue of using a simple queueing system to model a big service center or supply-demand facility; therefore, one possibility is to extend the queues to more general queueing systems or networks. Other future work may include (1) examining how low and medium traffic intensities influence the system behaviors, (2) allowing jobs (demand) jockeying among neighboring cells, and (3) making the service processes in the queues dependent on each others.

Another interesting extension is to model random neighborhood. This would make the model more realistic because agents in the real world could interact with other agents at different remote locations.

ACKNOWLEDGMENTS

The authors wish to thank the National Science Foundation through grant CMMI-0644959 for partial support of the research reported here.

A APPENDIX

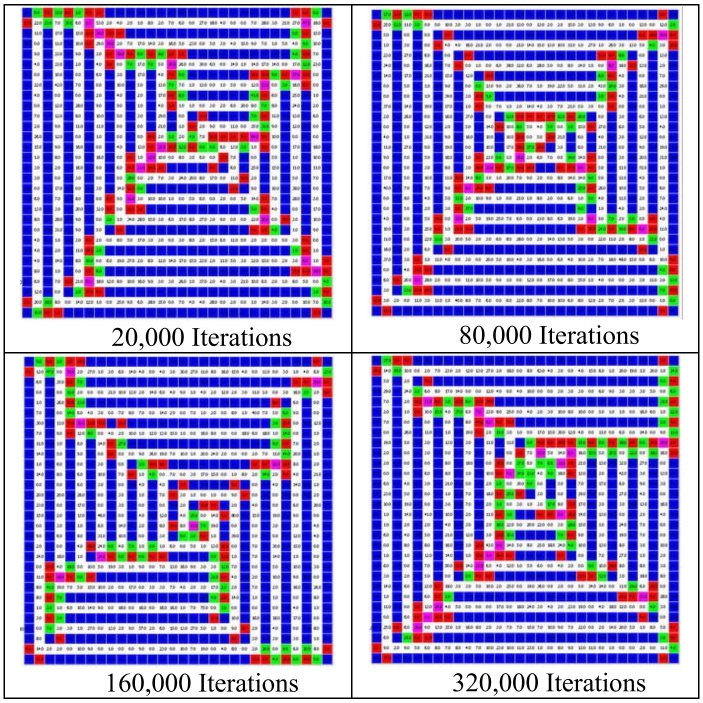
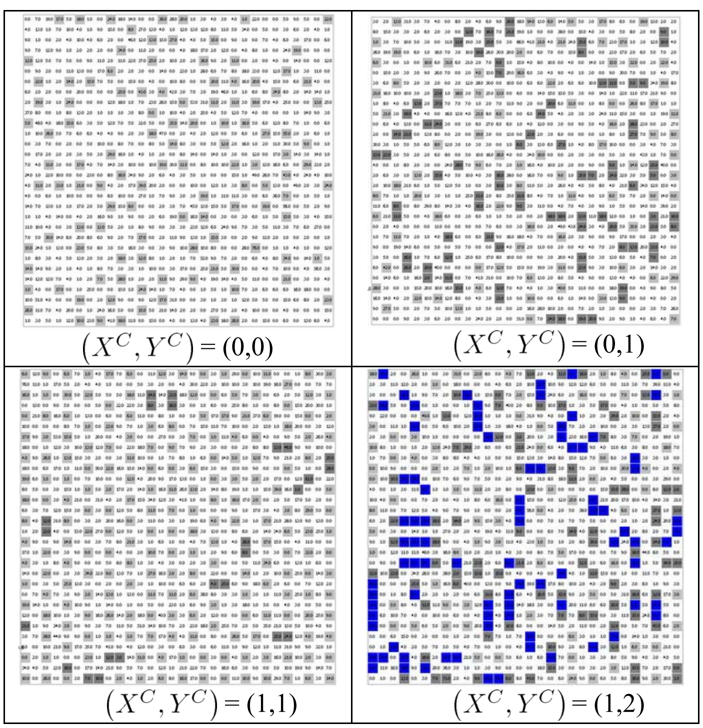
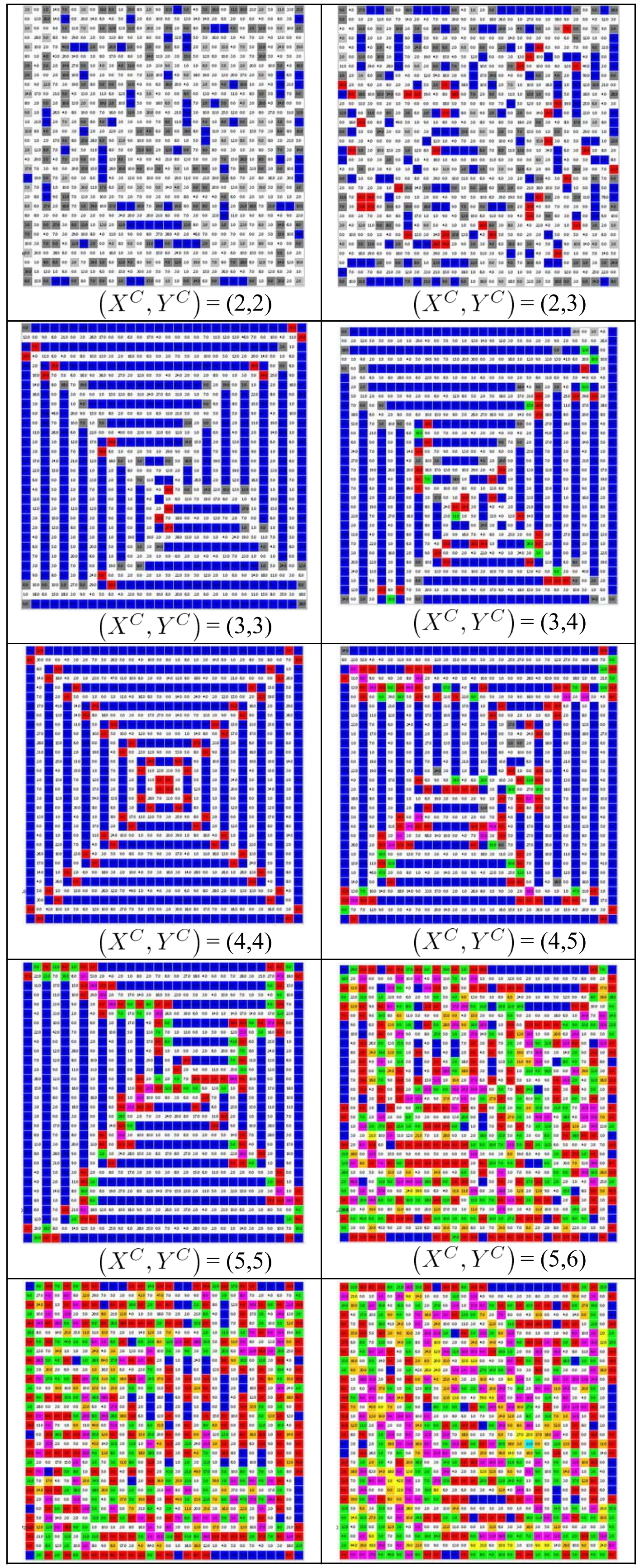


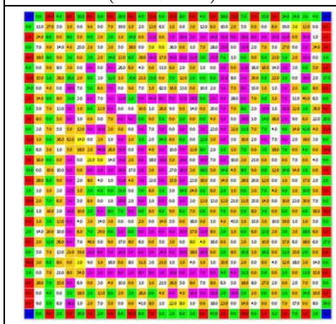
Figure 4: "On-The-Edge-Of-Chaos" Pattern under $\left(X^C, Y^C\right) = (5,5)$



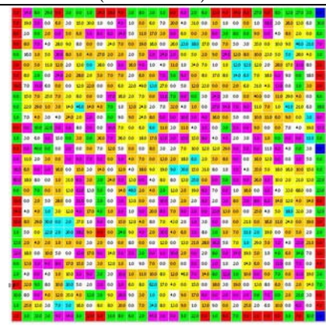


$(X^C, Y^C) = (6,6)$

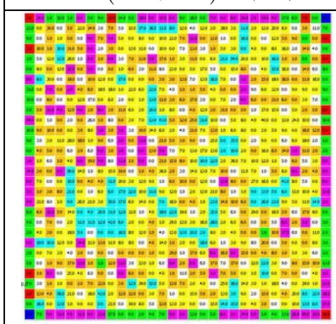
$(X^C, Y^C) = (6,7)$



$(X^C, Y^C) = (7,7)$



$(X^C, Y^C) = (7,8)$



$(X^C, Y^C) = (8,8)$

Figure 5: Chaotic, Deterministic, and On-The-Edge-Of-Chaos Patterns under Various Rules

REFERENCES

Berlekamp, E. R., J. H. Conway and R. K. Guy 1985. *Winning Ways for Your Mathematical Plays*, Academic Press.

Beyer, W. A., P. H. Sellers and M. S. Waterman 1985. Stanislaw M. Ulam's Contributions to Theoretical Theory. *Letters in Mathematical Physics* 10: 231-242.

Bonabeau, E. A., M. A. Dorigo and G. A. Theraulaz 1999. *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, New York.

Borshchev, A., Y. Karpov and V. Kharitonov 2002. Distributed Simulation of Hybrid Systems with Anylogic and HLA. *Future Generation Computer Systems* 18(6): 829-839.

Conzelmann, G., M. J. North, G. Boyd, R. Cirillo, V. Koritarov, C. M. Macal, P. Thimmapuram and T. Veselka 2004. Simulating Strategic Market Behavior Using an Agent-Based Modeling Approach--Results of a Power Market Analysis for the Midwestern United States. In *6th IAEE European Energy Conference on "Modeling in Energy Economics and Policy"*, ed.

Katsaliaki, K. and S. C. Brailsford 2006. Using Simulation to Improve the Blood Supply Chain. *J Oper Res Soc* 58(2): 219-227.

Koritarov, V. 2004. Real-World Market Representation with Agents. *IEEE Power and Energy Magazine* 2(4): 39-46.

Langton, C. G. 1990. Computation at the Edge of Chaos: Phase Transitions and Emergent Computation. *Physica D* D42,: 12-37.

Macal, C. M. and M. J. North 2005. Tutorial on Agent-Based Modeling and Simulation. In *Proceedings of the 2005 Winter Simulation Conference*, ed. Kuhl, M. E., N. M. Steiger, F. B. Armstrong and J. A. Joines, 2-15. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Mustafee, Taylor, Katsaliaki and Brailsford 2006. Distributed Simulation with Cots Simulation Packages: A Case Study in Health Care Supply Chain Simulation. In *Proceedings of the 2006 Winter Simulation Conference*, ed. Perrone, L. F., Wieland, F. P., Liu, J., Lawson, B. G., Nicol, D. M., Fujimoto, R. M., 1136-1142. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Newman, M., A.-L. Barabasi and D. J. Watts 2006. *The Structure and Dynamics of Networks*, Princeton University Press.

Pathak, S. D., D. M. Dilts and G. Biswas 2007. On the Evolutionary Dynamics of Supply Network Topologies. *IEEE Transactions on Engineering Management* 54(4): 662-72.

Pritsker, A. A. B., O. P. Daily and K. D. Pritsker 1996. Using Simulation to Craft a National Organ Transplantation Policy. In Proceedings of the 1996 Winter Simulation Conference, ed. Charnes, J. M., D. J. Morrice, D. T. Brunner and J. J. Swain, 1163-1169. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Toffoli, T. and N. Margolus 1987. *Cellular Automata Machines : A New Environment for Modeling*, MIT Press.

Von Neumann, J. 1966. *Theory of Self-Reproducing Automata*, University of Illinois Press.

Wolfram, S. 1983. Universality and Complexity in Cellular Automata. *Physica D: Nonlinear Phenomena* 10D: 1-35.

Wolfram, S. 2003. *A New Kind of Science*, Wolfram Media, Champaign.

AUTHOR BIOGRAPHY

WAI KIN (VICTOR) CHAN is an Assistant Professor of the Department of Decision Sciences and Engineering Systems at Rensselaer Polytechnic Institute. He received a B. Eng. and an M. Eng. in electrical engineering from, respectively, Shanghai Jiao Tong University and Tsinghua University, and a M.S. and Ph.D. in industrial engineering and operations research from University of California, Berkeley in 2001 and 2005, respectively. His research interests include discrete-event simulation, agent-based simulation, mathematical programming, and their applications in energy markets, social networks, service systems, and manufacturing. His e-mail address is <chanw@rpi.edu>.