Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

DISCRETE ELEMENT MODELS AND REAL LIFE DUALS

Ross A. Gagliano

Department of Mathematics and Computer Science
Georgia State University
Atlanta, Georgia 30303-3083 USA
(404) 651-2245
matrag@gsusgi2.gsu.edu

Michael R. Lauer

Datatrac Corporation
1325 Northmedow Parkway
Roswell, Georgia 30076 USA
(800) 827-8722
michael@datatrac.om.org

ABSTRACT

Some interesting results are obtained using a class of models defined as discrete element. One of the best examples of this class is the Game of Life whose history is traced to cellular automata. Through duals, the results link these discrete element models with standard mathematical models, the specific one involved here is a logistic equation. Some implications are drawn from Lauer's blending of two discrete element models: the Game of Life and the Prisoner's Dilemma. The totally cooperative aspects of "births" and "deaths" in the standard Game of Life are contrasted with the competitive features of the algorithms of strategy in the Prisoner's Dilemma. Of particular significance are the optimal population values for varying levels of "selfish" births and the counterintuitive answers to a number of significant questions that are raised in a series of simulation experiments.

1 INTRODUCTION

We begin with a few operational definitions and a brief discussion of the concepts used in this paper. First, in a broad sense of mathematics which encompasses both pattern recognition and symbol manipulation, we distinguish mathematical models from discrete element models. These two are related, nonetheless, as can be shown through the use of so-called duals.

For our purposes, a *mathematical model* is defined as an equation or function whose varied form can include arithmetic, algebraic, Boolean, differential, or integral expressions. A distinguishing feature of these models is that their mathematical specification is functional and macroscopic or top-down. Mathematical models abound in scientific computation, particularly in finite element analysis (FEA) and finite element methods (FEMs).

FEA is used to represent differential aspects of continuous systems, and a FEM assists in solving resulting differential equations. Solutions to these expressions often are found by finite approximation techniques where continuous functions have piecewise continuous replacements defined on particular topologies; e.g., a polygon.

Conversely, a *discrete element model* is embodied in a computer simulation which can be decomposed into the data structures that represent the attributes of each component or element of a system, and the algorithms that dynamically exercise the model. Such discrete element specification is usually microscopic or bottom-up.

Discrete element models generally exhibit three features [9]. First, each element in some total collection is identified by attributes such as name, identification number, location, class, type, etc. Second, the time-varying values of these sets of attributes are determined by rules based on: how each individual element is "connected" others, its disposition to change, its strategy, and its previous responses, etc. Lastly, the aggregate patterns, or distributions of population values, quite often provide insight beyond that available through the individual data.

The very recent success in experimenting with discrete element models is due in no small way to the availability of personal computers. Computer generated patterns, such as fractals, chaos, artificial neural networks, and computer generated art, are often derived using discrete element models. However, discrete event simulation, with its central concept of "waiting lines" or queues, may be the best known area of simulation and, moreover, a fertile area for contrasting mathematical and discrete element models.

In queuing models, each element is discretely specified and tracked in time (or modeled), and the necessary data structures and algorithms are incorporated into a simulation programming language.

Such languages are Simscript, GPSS, Siman, or Simula, or a higher level general purpose language such as FORTRAN, Pascal, Modula, Ada, or C. Within a formal subject area called Queuing Theory, duals are developed that employ closed-form, continuous mathematical expressions that describe the limiting values of the "waiting line" processes.

Lastly, a *mathematical dual*, analogous to the philosophical notion of duality, or two modes of thought, can be either a set of consistent forms of representation or two models that yield comparable results. In this paper, we shall use the latter definition of the term dual to infer yielding similar results through two different representational forms. As noted previously [13], "mathematics is often defined as the science of space and number ... and it was not until the recent resonance of computers and mathematics that a more apt definition became fully evident: mathematics is the science of patterns."

2 HISTORY OF CELLULAR AUTOMATA

The evolution of discrete element models generally, and the Game of Life specifically, can be traced to the work in cellular automata, or a theory of machines that incorporate cells as components. This area of study is also known as tessalation automata and comprises a portion of the formal area within computer science. Cellular automata have become valuable modeling and simulation tools in science and technology. These tools extend across the spectrum of disciplines from physics, chemistry, and biology, to philosophy, sociology, and political science.

Cellular automata started with the work of Alan Turing in the late 1930's, one of whose contribution was the notion of a Turing machine or "universal computer" [5]. In the forties, John von Neumann extended Turing's ideas with efforts towards a self-replicating automaton, or a robot that could build a copy of itself using standard components [4]. Stanislaw Ulam suggested the use of an array of rectangular cells, assembled on something like a chessboard, where each cell takes on one of a number of possible states [13]. Edward Moore, known for the Moore automaton [4], proposed the use of the eight-cell neighborhood (four adjacent and four orthogonal) for the rectangular grid. Thus, this neighborhood is sometimes called the "Moore neighborhood" which has became a standard neighborhood or template particularly for the two-dimensional Game of Life as described below.

3 ONE-DIMENSIONAL GAME OF LIFE

The Game of Life has many variations: three-dimensional, two-dimensional, two-sided using two different colored pieces or "chips", and even one-dimensional. First suggested by Stephen Wolfram [19], the one-dimensional Game of Life is essentially a linear version of the more common two-dimensional approach. The one-dimensional consists of a linear array of cells of some maximal length typically defined by the screen width of the computer system on which it is generated [16].

Each cell can take on a number of states determined by transition rules, which we will later call strategies, such as the following. If the present value (or state) of a given cell are added to the values of its two adjacent neighbors, then its next state of that cell is a function of the sum.

For example, if the computed state or sum of a given cell is as shown on the top line:

0 1 2 3 4 5 6 7 8 9
0 2 3 0 1 0 1 0 0 0

then the next state is given by the values on the second line. In words, if its sum is 0, then its next state is 0; if 1 then 2; if 2 then 3; etc. Thus, if we start with a single cell whose initial state of '1' in the center, the sequence of sums and states will be as follows. The states are given on the left and the sums are given on the right. If the states are represented by special symbols or colors, extremely appealing patterns can be generated.

1	111
222	24642
31113	3454543
10001	1110111
2220222	246444642
311111113	34533333543
100000001	11100000111
22200000222	

4 TWO-DIMENSIONAL GAME OF LIFE

The original Game of Life was developed in the sixties at Gonville and Caius College, part of Cambridge University, by John Conway [12]. It started out as a manual game that could be exercised on a chess board with a set of chips initially placed on certain squares. Later, it was implemented on various mainframe computers of the day.

Rules were developed to determine whether chips would be located on particular cells at successive time steps. Sets of chip patterns were called configurations. After the initial placement of chips on the board, successive configurations depend only on the immediately previous configuration in the following way.

Each chip (or "filled" location) monitors its "Moore neighborhood," or the four adjacent and four diagonal cells. This neighborhood forms the <u>template</u> within which its number of "neighbors" or filled locations is counted. The rules (strategies) are as follows:

- If a particular chip has either 2 or 3 neighbor chips in its template, then it "survives" to the next configuration.
- If it has either 0 or 1 neighbors, then it said to "perish of loneliness."
- If it has four or more neighbors, then it "dies of overcrowding."
- If there is no chip at a particular location, then one is created (a "birth") if the unfilled location has exactly three neighbors in the template; otherwise, there is no change.

All "births" and "deaths" occur simultaneously each time period or generation, a series of which produces life "histories." When automated, these histories can be analyzed visually. Interesting initial configurations for visual patterns include:

a. The Blinker - { * * * }

b. The Glider -
```
{ * *   }
{ *   * }
{ *     }.
```

c. The Arch of Pi -
```
{ * * * }
{ *   * }
{ *   * }
```

b. A Viral Block -
```
{ * *  * * }
{ * *  * * }
{    y     }
{ * *  * * }
{ * *  * * }.
```

Successive configurations of the Blinker produce intermittent orthogonal bars, while the Glider appears to "move" in a diagonal direction. The Arch of Pi provides symmetrical patterns that can be studied from the point of view of population logistic experiments. The Viral Block is stable unless a single chip is placed at y after the initial configuration, from which point the configuration disappears.

There are two ways to track the states of this system: the first is by visual patterns that provide evidence of symmetry, structure, and motion or "movement" of the chips. The second is by plotting population levels at each discrete time period whose trajectories mirror common logistics models [8] through traces of aggregate numeric values.

5 POPULATION DYNAMICS

The activity of these systems can be monitored by plotting the population levels at discrete time periods. The resultant trajectories mirror common logistic models as shown below in Figure 1. The traces of aggregate numeric values create graphs known as S- or J-curves in biology which are plots of population logistic equations, one of which is next discussed.

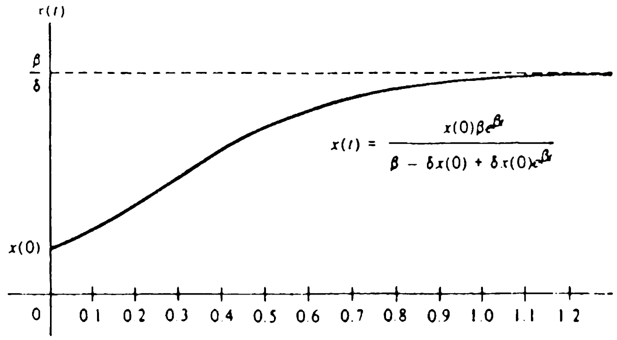

Figure 1: Population Logistic Curve

An important relationship of mathematical models and discrete element models is this dual between the Game of Life and its population dynamics curve. It is a fairly straightforward exercise to develop a discrete element model of the Game of Life and mimic the results with a standard differential equation model. The metric for such a comparison, as developed and used in many ecological studies, is the number of elements (in this case, chips) that are present at each time period. In real environments, such counts are observed to obey logistic expressions (laws) such as the Verhulst-Pearl equation:

$$\frac{dN(t)}{dt} = b * N(t) \frac{K - N}{K}$$

where N(t) is the population size at time t, K is the maximal population, and b is a constant real-valued parameter of change. The population size, N(t), can increase to some maximal value (K), decrease to zero (we usually do not track negative populations), or (with minor changes) oscillate. The plot of the population data of the Pi initial configuration is shown below in Figure 2. Note that the shape resembles that the J-curve previously discussed and shown in Figure 1.

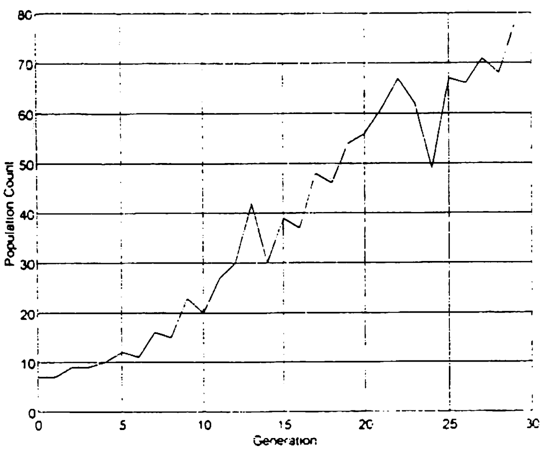

Figure 2: Game of Life Pi-configuration Data

6 LAUER'S EXTENSION

Lauer's extension to the two-dimensional Game of Life changes the rules slightly by incorporating ideas of competition versus cooperation. This approach allows the study of the ramifications of the addition of "greedy" or purely self-interested strategies (rules): i.e., locally optimal and apparently indifferent to the global impact [17]. We have previously explored such ideas in investigating the interdependence of strategies in a ring setting [10,11].

In attempting to ascertain whether there exists a non-zero optimal value of "selfish" births, Lauer [15] modified the standard birth and death rules with the addition of concepts based on the Prisoner's Dilemma [14], as discussed below. Using the standard Game of Life to establish the baseline or totally cooperative environment, it was conjectured that varying percentages of selfish births might yield more productive; i.e., larger sized, populations. It was then decided to test this conjecture through a simulation study.

Lauer's modified rules for the Game of Life are as follows:

- When a chip or filled node is surrounded by 4 or more chips, then instead of automatically dying if it is selfish, it "kills" an appropriate number of its neighbors until it has only three so that it can legally survive. For each round that it kills neighbors in this manner, it increments its vitality factor by one. It can also kill an additional neighbor (the rule of orneriness).
- If a chip has either 0 or 1 neighbor and is selfish, instead of automatically dying, it can survive the round if its vitality count is greater or equal to one. In each situation like this, it must decrement its vitality by one.
- To have a birth at an empty cell, the rules of exactly 3 neighbors is changed to 3 OR 4 neighbors with the designation of a new chip as either selfish or unselfish (normal) based on the given selfishness parameter of the population.

The insight into these rule changes is taken from the Prisoner's Dilemma (PD) as follows. PD was developed from a matrix game devised by Dresher and Flood in the 1950's at the RAND Corporation [7]. The name PD has been attributed to the mathematician Tucker who proposed the approach as an analogy to the technique often used by police in dealing with criminal suspects (prisoners) and the strategies that the suspects possibly use in making decisions (hence, the dilemma associated with their choosing). Their choices usually are to either "squeal" which means to turn state's evidence (if in fact guilty) or to stay "mum" which means to say nothing.

In its most general but simplest case, the PD can be represented as a two-player game where each player chooses one of two possibilities, either to "cooperate with" or to "defect from" the other player. A 2 X 2 matrix that provides the corresponding payoffs for these decisions is shown in Table I. How the PD relates to the Game of Life is that the strategies that players follow determine which action (whether to cooperate or defect) they take. Decisions made taking into account the PD payoffs seem to favor cooperative strategies (e.g., a very simple strategy called Tit-for-Tat) [1,3].

The necessary conditions for the PD payoff matrix are that:

$$T > R > L > S$$

and that

$$(T + S) / 2 \; < \; R.$$

Table I: Prisoner Dilemma Payoff Matrix

	Player A: Cooperates	Player B: Defects
Cooperates	(R,R)	(S,T)
Defects	(T,S)	(L,L)

Sequential rounds of PD "encounters" are called Computer Tournaments [2]. In each of the rounds of such a tournament, each player "exchanges" payoff values according to the PD matrix. Normally, participating players in each encounter are randomly paired, and their actions to cooperate or defect are determined as a function of their particular strategy and their recent experiences.

7 THE SIMULATION EXPERIMENTS

To summarize the Game of Life, we have a number of individual cells or nodes, each of which responds to its environment without regard to self-awareness or self-preference, but blindly follows a common set of rules. The environment itself exhibits a characteristic fecundity as manifested in the birth of new cells when certain conditions are met. We proposed a set of experiments to explore an aspect of the nature of life itself, to investigate what would happen if we were to inject an element of self-interest into this universe of selflessness.

Before considering the possible ramifications of such a step, we speculated that the Game of Life has the inherent potential to be modified in such a way. It turns out that the Game of Life does possess a unique capacity to model self-interest because:

1) Based on the dual that produces the well known population logistic curve, there is a basis for believing that by investigating this modification we are legitimately analyzing processes that apply to populations;
2) The traditional rule, that if a node is surrounded by 4 or more other nodes, then it dies selflessly of overcrowding, is changed. The modified version is that when a node is thus surrounded, it dies as before unless designated a selfish node; in which case, it kills as many of its neighbors as necessary to reduce the local population within its template to a point where it can survive itself. This is clearly an ideal paradigm of brute self-interest.

What made such extensions and these experiments particularly intriguing for the authors was that the possibilities appeared to be endless and potentially counterintuitive, with possibly profound ramifications for not only numerous applications [6], but also an evolving methodology [9].

In the simulations runs, all of the data beyond 0% selfish births were obtained by averaging the ending population for 600 replications on various size rectangular grids (matrices). This allowed us to determine beyond what point the results would become conceptually repetitious, though still variable and apparently unpredictable. The smallest matrix selected was a 7 by 7, the medium a 21 by 21, and the larger a 41 by 41.

For the medium size grid (a 21 by 21 matrix), stability is achieved before the 100th generation, while it takes approximately 200 generations for the larger matrix (41x41) to stabilize. Subsequent trials with varying percentages of selfish births were run up to those points for each size matrix.

8 RESULTS OF SIMULATION EXPERIMENTS

In discussing the results, let us conjecture some implications for the new rules. Recall that the major issue that guided the design of the simulation experiments was that of injecting an element of self-interest into the standard two-dimensional Game of Life. Clearly, the selfish nodes would benefit directly in terms of continued survival, but there were several additional open questions that needed to be addressed to assess the real impact of the selfish nodes on the selfless ones. Thus, we addressed these four questions:

1) Would the population at the end of some number of generations be greater than, equal to, or less than that achieved by the standard (or utterly selfless) opulations?
2) Is there, in fact, an optimal non-zero percentage of selfishness in a generally cooperative or selfless population?
3) Would the community as a whole be more or less stable with a selfish element that without one?
4) Does introducing a limited dose of selfishness into the population increase or decrease the overall population levels and mortality rate?

In answering these questions, it should be noted that we initially discovered that there is a non-zero value for selfish births that does cause an optimal value for the size of the population after one hundred generations. As shown in Figure 3, the value of 95 elements is the ending population for 0% selfish births (the standard Game of Life case), which peaks, however, later at 5 % of selfish births, but continues to drop as expected as the parameter (percentage of selfish births) increases from around 9 % and beyond.

To answer question #1, we examined the results for the medium and larger matrices to compare the maximum populations achieved by various strategy mixes. Referring to Table II, we see that the selfless strategy (0% selfish births) was able to fill about 30-35% of the matrix at the maximum, while 5% selfish births increased the percentage of fill to between 42 and 46%.

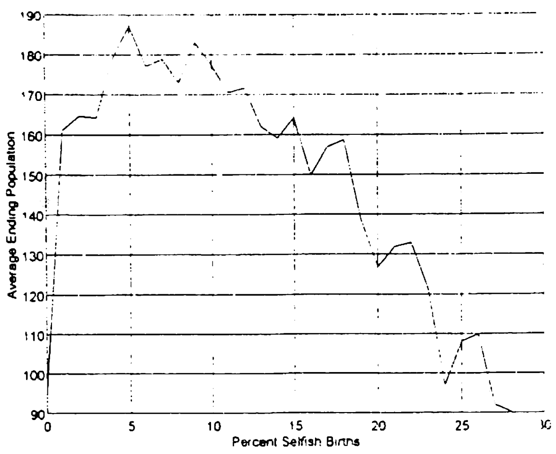

Figure 3: Average Ending Population as a Function of Selfish Births

This result is surprising because by including nodes that kill from 1 to 5 of their neighbors in order to survive themselves, we have created a compound strategy that is more productive than the simple, purely cooperative strategy alone. Thus, by including selfishness in a selfless population, we achieve an increased population level over the simple, selfless strategy.

Table II: Productivity of Selfish and Selfless Strategies

% Selfish Births	Max % Filled 21x21 grid	Max % Filled 41x41 grid
0	31.1 (1)	34.9 (3)
5	42.5 (2)	45.5 (4)

(1) Population of 137 achieved in the 98th generation.
(2) Average population of 187.28 achieved in the 99th generation.
(3) Population of 588 achieved in the 189th generation.
(4) Average population of 765.46 achieved in the 199th generation.

To answer question #2, we again refer to Figure 3 where we can see that at 5% selfish births we do, in fact, find an optimal non-zero value for the population level. We also note that although this is the optimal value for this experiment, the advantage in productivity persists all the way up to 26% selfish births.

We conclude that this is no fluke, no mere "sweet spot," that produces these results. There appears to be something about the inclusion of self-interest that has a very powerful effect on the population, and that is counterintuitively beneficial for the entire group.

Or is it? Does the inclusion of selfishness actually affect the average lifespan of the community as a whole?

To answer this, question #3, we refer to Figure 4 to examine the case where the available resources are exceedingly scarce. This is modeled by reducing the space, or the number of cells available, to the smallest (7 by 7) matrix. Here we find, at the end of 50 generations for the base case (0% selfish births), a catastrophic result: there are no survivors!

Figure 4 also makes clear the surprising truth of the matter: in the case of extremely limited resources, the inclusion of any amount of selfishness -- from 1% up to 100% (0-28% is shown) -- is required to guarantee the very survival of the community itself. In this case, the optimal result is achieved in the range of 10-11% selfish births.

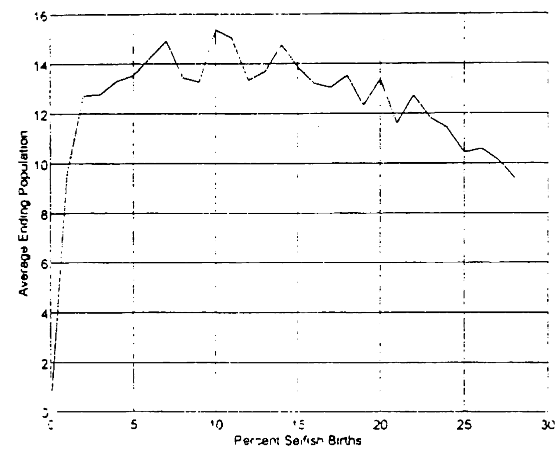

Figure 4: Pi-configuration with Limited Resources

Alternatively, to answer question #3, we can look at the average age of the population at the end of 100 generations. Here we find in Figure 5 that the community tends to stabilize in age beyond 10% selfish births. However, the average age for selfish nodes (the solid line graph in the figure) peaks near thirty generations, whereas the selfless nodes (the '+') remain below 5 on average. The combination, the average for the entire community (shown as the '*'), peaks slightly above 5 but remains near 5. What is startling is that the selfish nodes tend to decrease in average age while the selfless tend to increase for the increase of percent of selfish births shown.

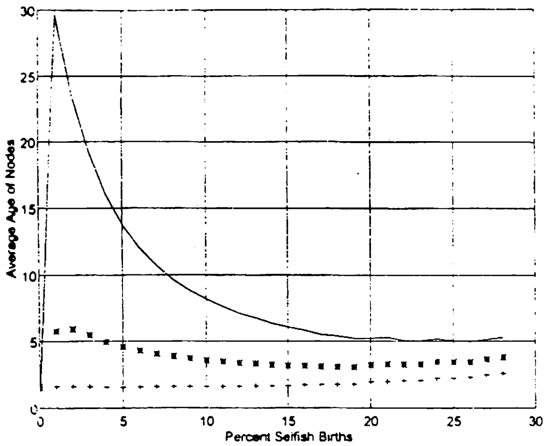

Figure 5. Average Age of Nodes

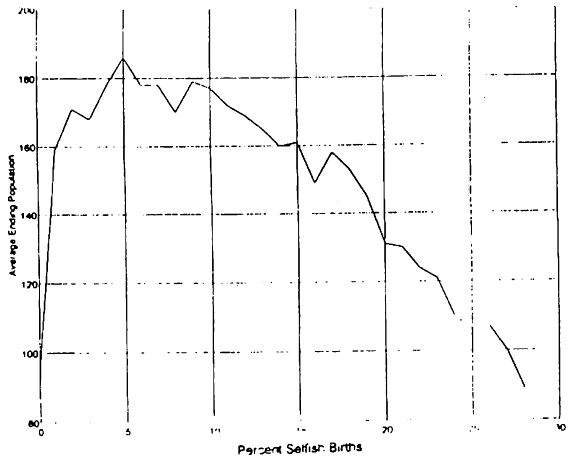

Figure 6. Average Ending Population

To answer question #4, refer to Figures 6 and 7 which show the relationship between the total number of deaths in 100 generations and the ending population. With 0% selfish births, there are 95 survivors and a total of 3808 deaths. The number of survivors does not return to this original level until we reach 27% selfish births, and in this range we find a startling result in the number of deaths (in Figure 7). As might be expected, as soon as we increase the percentage of selfish nodes in the population, there should be, and is, an immediate increase in the total number of deaths. This is apparently the price that must be paid for maximizing the overall population level. However, once we get above 18% selfish births, the number of overall deaths drops BELOW the 3808 level (of the 0% selfish births), yet we still achieve an advantage in ending population level that continues until the 27% level.

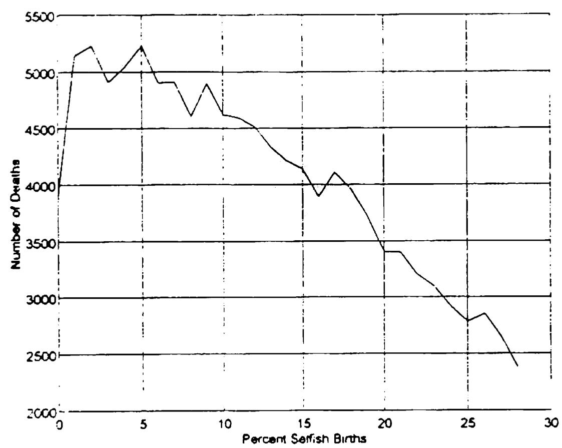

Figure 7. Number of Deaths

9 SUMMARY

In this paper, we have attempted to show the power of a class of models called discrete element through a series of simulation experiments. Some realism was demanded of the two-dimensional Game of Life by the inclusion of Prisoner Dilemma-like strategies, called Lauer's extension. With that modification, and the use of the mathematical dual of the population logistic curves, some surprising results were obtained. The results were more fully explicated through a series of questions and answers that provide some insight, results that are counterintuitive, and a few surprises.

We did not develop these models to ascertain what kinds of expectations that these results might contradict. Further, it is not certain how to justify drawing more generalized conclusions from the experimental data from our simulations. Ideally, the results should be abstracted into some forms of formulae, rules, or laws.

What we have attempted to show is that by including various levels of so-called selfish nodes in the initial population of the two-dimensional Game of Life through the control of the percentage of selfish births, we have found that:

1) For certain classes of resource allocation problems, there may be a more efficient packing of the available space, or a more efficient utilization of available resources;

2) The optimal amount of selfishness in an otherwise selfless population is not zero, but may be substantially higher than is initially suspected;

3) In certain extreme cases of resource availability, some degree of selfish behavior appears to be crucial to the survival of the group (the community as a whole);

4) The group benefits from the inclusion of some selfishness in terms of stability (as measured by average age) as well as in terms of overall population level;

5) For certain values, including selfishness into an otherwise totally cooperative population not only increases the productivity, but it also actually decreases the overall mortality rate.

We seek further corroboration and other general implications of these unexpected findings. The authors currently are working on several additional papers that demonstrate implications of these results for applications in several other fields; e.g., biology, economics, sociology, political science, and philosophy. At the very least, ourhope is that this contribution will provide a more comprehensive modeling approach with findings that support a spreading modern realization that life may be a great deal more complex than we first imagined.

Suffice it to say that attempts to model real world processes confronts us with the fact that life is far too complex to be comprehended in a single, simple theory. Therefore, to say that it would be best for all if everyone in a society acted selflessly is surely contradicted by modern market theory [18]. Further, to say that it would be best for all if everyone acted as the perfect 'economic man' is certainly contradicted by the results of these simulation experiments.

ACKNOWLEDGMENT

The authors wish to thank Phillip Mitchem for his support in the preparation of this paper.

REFERENCES

[1] Axelrod, R. A. and W. D. Hamilton, "The Evolution of Cooperation" Science, Vol. 211, 1390 - 1396, 27 March 1981.

[2] Axelrod, R. **The Evolution of Cooperation**. New York: Basic Books, Inc., 1984.

[3] Behr, R. L. "Nice Guys Finish Last - Sometimes," Journal of Conflict Resolution, Vol. 25, No. 2, 289-300, June 1981.

[4] Burks, A. W. (ed.) **Essays on Cellular Automata**. University of Illinois Press, 1970.

[5] Codd, E. F. **Cellular Automata**. New York: Academic Press, 1968.

[6] Dawkins, R. **The Selfish Gene**. Oxford University Press, 1989.

[7] Dresher, M. **Games of Strategy, Theory and Applications**, Englewood Cliffs, NJ:Prentice-Hall, 1961.

[8] Gagliano, R. A., "Some Reflections on Societal Modeling," Proceedings of the 7th Annual Pittsburgh Conference on Modeling and Simulation, University of Pittsburgh: Instrument Society of America, 766-770, April 1976.

[9] Gagliano, R. A. **Discrete Element Models and Simulation** (in preparation).

[10] Gagliano, R. A., M. D. Fraser and M. E. Schaefer, "The Simulation of Decentralized Control: A Hostless Resource Allocation Model," Simulation, Vol. 58, No. 6, 398-408, June 1992.

[11] Gagliano, R. A. and M. E. Schaefer, "Decentralized Control of Access to Resources in a Network: Interdependence of Strategies in a Challenge Ring Model," in M. D. Fraser (ed.) Advances in Control Networks and Large Scale Parallel Distributed Processing Models, Chapter 5, 145-170, Ablex Publishing Co., 1991.

[12] Gardner, M., "Mathematical Games: The Fantastic Combinations of John Conway's New Solitaire Game 'Life'," Scientific American, 120-123, October 1971.

[13] Gardner, M., "Mathematical Games: On Cellular Automata, Self-reproduction, the Garden of Eden and the Game 'Life'," Scientific American, 112-117, February 1972.

[14] Hofstadter, D. R. "The Prisoner's Dilemma Computer Tournaments and the Evolution of Cooperation," in **Metamagical Themas**, New York: Basic Books, Inc., 1985 (originally appeared in Scientific American, May 1983).

[15] Lauer, M. R. "Real Life or a New Perspective on SOB's," CSc 836 Term Paper, Georgia State University, December 1993.

[16] Perry, K. E. "Abstract Mathematical Art," Byte, 181-192, December 1986.

[17] Railey, M. R. "Dynamic Control Structures for Cooperating Processes," PhD Thesis, University of Illinois, January 1986.

[18] Schelling, T. C. **Micromotives and Macrobehavior**. New York: W. W. Norton & Co., 1978.

[19] Wolfram, S. "Cellular Automata as Models of Complexity," Nature, October 1984.

AUTHOR BIOGRAPHIES

ROSS A. GAGLIANO is an Associate Professor of Computer Science at Georgia State University. He received his Ph. D. from Georgia Tech in 1976. His research interests include software engineering, computer architecture, and the aspects of modeling and simulation as discussed in this paper.

MICHAEL R. LAUER is a graduate student in Computer Science at Georgia State University. He received B. S. in Computer Science from Georgia State. He is currently employed as a software engineer.