

Studying the Emergence of Multicellularity with Cartesian Genetic Programming in Artificial Life

Joseph A. Rothermich

School of Computer Science
University of Birmingham
Edgbaston, Birmingham B15 2TT
United Kingdom
msc84jar@cs.bham.ac.uk

Julian F. Miller

School of Computer Science
University of Birmingham
Edgbaston, Birmingham B15 2TT
United Kingdom
j.miller@cs.bham.ac.uk

Abstract

A study is made to learn how multicellular organisms might have emerged from single-celled organisms. An understanding of this phenomenon might provide a better understanding of natural and man-made multicellular systems. The experiment performed is an Artificial Life simulation that uses Cartesian Genetic Programming to evolve the behaviors of individual cells. Cells have the ability to sense chemicals around them, signal, divide, and move towards or away from chemicals. Interesting group behavior emerged from the simple instruction sets used by the cells.

1 Introduction

The origin of multicellularity is one of the primary mysteries in modern science. The historical occurrence of single-celled organisms evolving into multicelled organisms has occurred at least three times in the history of the earth signifying that it was not a chance occurrence (Bonner, 2000). It is not currently understood why multicelled organisms evolved. A genetic mutation or an environmental change may have created a situation in which cells that adhered or grouped together might have had better fitness, increasing their chances for survival.

One example from nature that allows for the observation of multicellular emergence is the Cellular Slime Mold (*Dictyostelium discoideum*). The study of *Dictyostelium* is of immense importance in attempts to understand many issues connected with the development of multicellularity and cell signaling and motility (Kessin, 2001). *Dictyostelium* are single celled organisms that, when running out of local food resources, temporarily form into multicelled slug-like creatures. Their combined form allows them to quickly move to a new location where new energy sources might be found. After the

cells have moved as a group to a new location, a fruiting body is formed and cells break off and return to a single-celled existence. It was once thought that a leader cell would signal the others to aggregate, however it was later discovered by Keller and Segel (1970) that the cells had no leader.

From a computer science perspective, a cell can be viewed as a simple computer program. The cell has inputs and performs an action based on those inputs. Advances in computing theory, parallel processing hardware and networking have made it easier to imagine systems that are made up of individual cells, all working together to perform as a system. In this scenario, the system would be a function of the emergent properties of its components. Individual cells or programs would not be functional without the group and the functioning of the system would be greater than the sum of its parts. Creating systems in this manner may be necessary in order to continue progress in computational complexity and robustness. However, the human mind has a hard time understanding (Resnick, 1994; Johnson, 2001) and designing (Ray, 1994) decentralized systems. This paper aims to make progress on this problem by evolving a decentralized system in an Artificial Life model.

Although this paper focuses on a model of a natural ecology, modeling the emergent properties of non-biological systems is also an active area of research. Studies have been made to increase our understanding of how economic markets and pricing work (Cliff and Bruten, 1997; Epstein and Axtell, 1996) and the dynamics of city growth (Krugman, 1996; Shelling 1978).

Previous models for the study of cells and multicellularity have been created by (Agarwal, 1995; Fleischer and Barr, 1992; Ray, 1994; Resnick, 1994). This paper differs in that cell behavior is evolved through a form of genetic programming called Cartesian Genetic Programming (CGP).

CGP was used to create a genotype that maps cell inputs to behaviors or actions. The program of each cell was evolved over time by applying mutation

during the process of cell division. Cells competed or cooperated for limited energy sources in the environment. The cell's phenotypes were observed to see what types of individual and group behaviors evolved.

2 Cellular Slime Mold Simulation

In order to test the programmed environment and cell behaviors, a non-evolving experiment was performed before the primary experiment took place. The goal was to model Cellular Slime mold behavior during their aggregation phase. The rules for cell behavior were similar to those used by Resnick's StarLogo Implementation (1994). Cells start in random positions and constantly release a chemical signal. They have the ability to sense the strength of chemical straight ahead, 45 degrees to the right and 45 degrees to the left. The cells are hard-coded to move towards the strongest signal with some amount of randomness.

This experiment exhibited behaviors similar to the grouping of Slime Mold cells. Just as in the StarLogo implementation, a degree of randomness was necessary in the cell's movement in order to create large groups of cells.

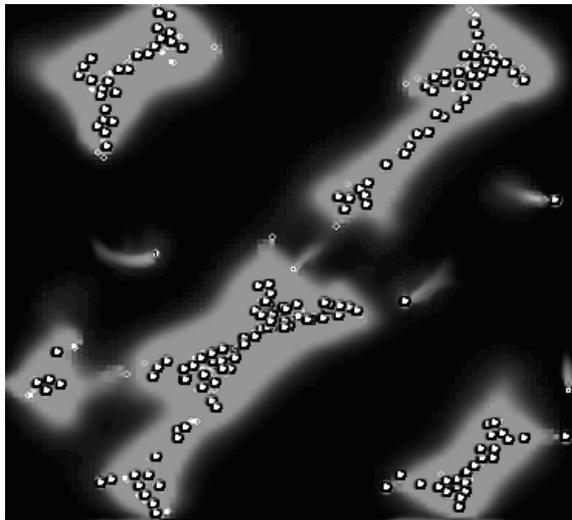


Figure 1: Slime Mold Aggregation (500 Cells)

3 Experiment

The purpose of the experiment was not to evolve cells that are similar to their counterparts in nature, but to grow cells that evolve to be successful in their own simulated ecology. The following section describes the setup and results from the experiment.

3.1 Cell Inputs

Natural cells have the ability to sense and absorb chemicals from their environment. They can either absorb chemicals directly, or the cell can accumulate the chemical in a receptor that only allows chemicals to be absorbed after a certain threshold has been passed. There are many chemicals that have an effect on cells in nature. A set of imaginary chemicals was used in this experiment. One chemical represented an energy source for the cells. The other chemical was emitted by the cells themselves, simulating cell-cell signaling.

Cells do not have any information about their location in the world. They only know what chemicals are present around them and the concentration gradients for each. This experiment followed a similar approach (i.e., in a grid world, the cells do not know their x, y coordinates). The cells also do not really know who their immediate neighbors are. They can sense the chemical being emitted by a neighbor cell, but they do not know more than that.

A threshold parameter similar to receptors in nature was used for cell inputs. This allowed the sensitivity of the cells to vary. In this implementation the parameter was a global value for all cells, but it could easily be adapted to be part of each cell's evolvable genotype.

Cells in this experiment also have the internal input of knowing how much energy they have. This means the actions they choose might be dependent upon their current health.

3.2 Cell Functions

3.2.1 Movement

Cells have the ability to move towards or away from chemicals gradients. This movement is called chemotaxis. Some cells such those with flagella have polarity and need to turn their bodies in order to change their direction. Other cells have cilia and can follow a gradient path directly with out having to turn.

The cells created in this experiment had similar functionality. Whether or not cells had polarity was controlled via a parameter on the program control panel.

A small degree of randomness was also added to the movement of the cells to keep their movement close to that of natural cells.

3.2.2 Cell Division

The cells in the experiment also have the ability to divide. Once a cell divides, the cell itself no longer exists and is replaced by two offspring. Each offspring has half of the energy of the original cell plus a small energy cost for the division process. The offspring inherit the genes of the parent cell plus some possible mutation.

3.2.3 Chemical Signaling

Cells were given the ability to release a chemical into the environment. This gives the cells a way to communicate with each other similar to the way that Cellular Slime Molds signal with cyclic AMP signals during their aggregation phase.

3.3 Cartesian Genetic Programming

In Cartesian Genetic Programming (CGP) a genotype is an integer string that encodes an indexed, feedforward, acyclic graph (Miller and Thomson, 2000). Unlike the parse tree representation in the standard GP (Koza, 1992), a genotype-phenotype mapping is used to create the graph phenotype from the integer string genotype. Each node in the genotype contains two types of genes: connection genes that represent how the inputs to the node are connected to program inputs or the outputs of other nodes, and a function gene that represents the operation that the node carries out on the inputs it receives. The nodes that are not involved in the linked path between the inputs and outputs of the program are inactive in the phenotype. Such nodes have no effect on the behavior of the phenotype. However a point mutation operator can re-connect inactive nodes or disconnect active ones. This allows neutral drift to take place. This has been shown elsewhere to be extremely beneficial to the search process for a number of problems (Miller and Thomson, 2000, Vassilev and Miller, 2000, Yu and Miller, 2001).

3.4 Implementation Details

3.4.1 Genotype Representation

As in standard CGP, the genome consists of a number of genes that are linked to form a graph. The right-most gene is executed first, and each gene can be connected to any other gene on its left. The final gene on the left is always a 'do nothing' function. During the decoding of the phenotype, if no action has been reached while traversing the graph, then eventually the left-most 'do nothing' gene will be called and the cell will not perform an

action. This case occurs less often as the genotype length parameter is increased. For these experiments, a genome length of 20 genes was used.

Each gene consists of an array of four numbers, representing the function (1), an external input (2), and two connections (3) and (4). The representation of a single gene is shown in Figure 2.

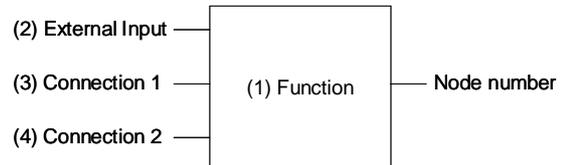


Figure 2: A Single Gene In The CGP Representation

A list of functions is shown in Figure 3. As an example, if a gene is coded with function one, it will move towards the chemical referenced in the external input (2). If a cell is coded with function three, the cell will check for the presence of the chemical referenced in (2) and then execute the gene referenced in the connection (3) or (4) accordingly.

The parameters UPPER_THRESHOLD and LOWER_THRESHOLD used in functions five and six are local variables for each cell that are considered part of the evolving genotype.

0. Do nothing
1. Move towards (2)
2. Move away from (2)
3. If (2) is present do (3), else do (4)
4. If (2) is present do (4), else do (3)
5. If energy is above UPPER_THRESHOLD do (3), else do (4)
6. If energy is below LOWER_THRESHOLD do (3), else do (4)
7. Perform (3) then divide
8. Perform (3) then release chemical signal

Figure 3: Cell Functions

In order to fully explain the decoding process for a phenotype, it is helpful to give an example. Figure 4 shows a sample genotype with five genes. The positions containing functions are underlined. The same genotype can be represented pictorially as shown in Figure 5.

0	1	2	3	4	5
0000	<u>7</u> 100	<u>1</u> 010	<u>1</u> 012	<u>3</u> 100	<u>3</u> 024

Figure 4: Example Cell Genotype In CGP

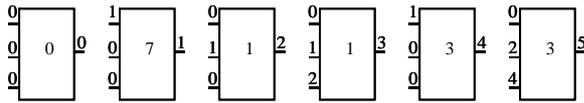


Figure 5: Pictorial Representation Of Example Genotype

The genotype is mapped to a phenotype by traversing the graph starting at the right-most gene. The function assigned to gene five is function 3: *If (2) is present, do (3) else do (4)*. The value in position (2) is zero, which represents the chemical used to produce energy. If this chemical is present, then decode and perform the gene identified by position (3): which is gene number two. Otherwise, perform the gene referenced in position (4): which is gene number four. This process continues until an action is found.

The fully decoded phenotype for this example describes the cell's behavior and is a series of IF-ELSE statements. The phenotype is shown in Figure 6.

```

If Chemical 0 is present
  Move towards Chemical 0
Else
  If Chemical 1 is present
    Do nothing
  Else
    Do nothing

```

Figure 6: Example Decoded CGP Cell Phenotype

3.4.2 Genetic Operators

When a cell divides, one of its offspring is mutated and the other inherits the parent's genes exactly. This allows for elitism so that well conditioned genotypes are not lost in the next generation.

Mutation consists of randomly changing one of the integer values in the genotype's CGP string. Only one value is changed per mutation. A mutation can modify the function, chemical input, or connections in a gene. Since only a portion of the genotype is decoded into the phenotype, mutations often do not affect the behavior of a cell.

A second mutation was implemented by adding or subtracting a random number to the upper and lower energy threshold variables.

3.4.3 Cell Population and Fitness Evaluation

A random population of cells was created for each execution. The initial, maximum and minimum number of cells in a population were

controlled by parameters on the user interface. If the population size fell below the minimum, new random cells would be created. When the maximum population size was reached, cells were no longer permitted to divide.

A steady state algorithm was used in which cells would continue living and dividing until they no longer had energy and died. Cells needed to compete or cooperate for a limited amount of chemicals necessary for energy production. This chemical was released into the environment at single or multiple points in the environment for short periods of time. After the chemical was released from a location for a period of time, an emission would start from a new random location.

Fitness was not explicitly measured, but instead fit individuals were cells sharing the same genes being able to perpetuate the existence of their genes. Cells sharing the same genetic origins were referred to as cell families in this paper. Genes producing cells that had a long life and didn't self-replicate were considered equally fit to those that produced cells that divided often as long as both strands of genes continued to remain active in the environment.

3.5 Results and Analysis

The evolving cell experiment produced cells that exhibited interesting behaviors. It was entertaining to witness an initial population of cells, each with their own, often odd, behavior. Some cells would circle around, others would do nothing and signal, and others would move quickly and efficiently towards energy sources. As time steps passed, the cells with losing strategies would die off and cells with better fitness would either continue to live or produce offspring. Although the function set of the cells did not allow for cells to physically group together as multicelled organisms, cooperative strategies emerged which could be considered first steps towards multicellularity.

3.5.1 General Observations

The populations of cells during the experiments usually tended to find a working strategy and use it for a while. Usually an improved mutated phenotype would evolve and eventually dominate the population. Sometimes, even though a population consisted mostly of a successful family of breeding cells, a few older cells would exist. These cells did not divide but merely had the instruction to move towards energy sources. They had been lucky enough to continue finding new sources after each old source ran out. Since they did not divide and split their energy in half, they were able to keep more energy for themselves. This store of energy

gave them a longer time to search out a new energy source before running out of energy. In the long term, however, these cells died off eventually. Ultimately they would not find a new energy source in time or cells that had already found the new source would block them out.

Families of cells that divided showed much better success at long-term survival. By accepting the energy cost of cell division, cells were weakened individually. In spite of this, as a group they were stronger. The families of cells were able to distribute their stored energy into more cells so that a greater area of the environment could be explored for new sources. It didn't matter that a large percentage of the cell family died by not finding sources. The probability of at least a few of the cells finding an energy source was high and those cells would absorb energy and divide to create the next generation. This was an example of cell cooperation. The cells were sacrificing their own fitness for the fitness of the group. As mentioned by (Bonner, 2000; Michod and Roze, 1999), the movement from competing cells to cooperating cells might be one of the first requirements for the development of multicellularity.

3.5.2 Genetic Neutrality

Rapidly changing environmental conditions during the Cambrian Period are possibly the cause of the genetic diversity and progress for which the period is known (Kirschvink, Ripperdan, and Evans, 1997). In the spirit of this idea, the simulation's energy sources were increased and decreased over time. This created a world where cells needed to be able to adapt rapidly to their new environment.

A high degree of neutrality was created in the populations over time. Even though one phenotype might begin to dominate the population, mutations would constantly be changing the genotype of the new cells. Only rarely did a mutation affect the phenotype behavior. After many generations, a population would often consist of a very diverse set of genotypes which all mapped approximately to the same phenotype.

When the environmental conditions changed as described above, the neutrality and genetic drift seemed to have a positive effect on the population's ability to adapt to the new conditions. The diversity of genotypes due to neutrality led to a wide range of phenotypes being tested in the new environments. It seemed much easier for mutations to find a better-suited phenotype as a result of this diversity. It is possible that neutrality made families of cells sharing the same genetic origins much more robust.

3.5.3 Collision Detection and the Huddle Strategy

The ability to control whether or not two cells could occupy the same space was available as a runtime parameter on the user interface. Collision detection prohibited cells from moving into the same space on the grid. The value of this setting had strong impacts on which cells were successful and how the cells evolved. In cell division, however, two offspring cells were created in the same location as their parent.

In order for a family of cells to be successful, it was important that it reached energy sources quickly and took advantage of them. When collision detection was turned on, it was possible for groups of cells to monopolize the energy source and physically block out other cells.

If the population was of moderate size and there was little competition for the limited supply of the energy-producing chemical, then usually cells evolved to simply move towards the food. However, when there was strong competition for food and collision detection was enabled, more strategic behaviors were required. Occasionally, cells evolved to move towards food, but once there, move towards each other's chemical signal instead. This behavior resembled a huddling action around the food that blocked out competition from entering. Signaling and moving towards the signal concentration kept the group as tight as possible. If instead a cell just moved towards the highest level of food in its immediate environment, the cell might move around a little. Since many cells are quickly absorbing the food coming from the source (the food drop-off point), a cell might sense a temporarily higher level of food in the direction opposite the food drop-off point—just outside of the cluster. While a cell might get a short-term benefit from moving away from the group, it might also lose its position and allow competition to enter. In this case, short-term greedy behavior hurt the cell and the average success of its family of cells.

3.5.4 Dividing Only While Located at an Energy Source

While it was important for a cell to produce offspring—helping their cell family to explore and increase chances of long-term survival—it was dangerous to divide unless energy was high or an energy source was near. Each time a cell divided, its energy was more than cut in half. If a cell divided when its energy was low, it hurt its chances of finding an energy source before running out of energy and dying.

When experiments were run over a large number of time-steps, cells would often evolve to only divide if they had a high amount of energy and were thus probably near an energy source. An example of a

cell with this behavior is shown in genotypic form in Figure 7 and its phenotype is shown in Figure 8.

0	0	0	0	7	1	0	0	4	1	0	0	2	0	0	1
5	0	2	0	7	0	4	4	0	0	3	1	8	1	1	1
1	0	1	2	0	0	7	4	1	0	2	5	5	1	10	1
1	0	1	1	4	0	10	12	4	0	11	9	0	1	6	2
6	1	10	7	2	0	7	9	6	1	2	17	6	1	10	7

Figure 7: The Genotype Of A Cell That Divides Only When It Is Near An Energy Source

```

If Energy is lower than 8 then
    Move Towards Chemical 0
Else
    Release Chemical Signal and
    Divide and
    Do Nothing
  
```

Figure 8: The Phenotype Of A Cell That Divides Only When It Is Near An Energy Source

This program allowed the cells to divide only when they had a significant level of energy. If they were low on energy, they would move towards the energy-producing chemical. Once they found a source and their energy increased, they would stop moving, collect energy and divide. This often worked because it meant that they could position themselves directly at an energy source and divide as much as possible making the group strong. If cells had used another strategy and did not divide, the single cell would grow strong but only up until the maximum energy level—the rest of the energy would have been wasted.

This strategy was very successful so the population was dominated by similar phenotypes. If collision detection was turned off, almost all of the cells in the population would be located at a single position during their dividing phase. While collecting energy and dividing, the cells would develop a massive population that would stay at a food source and grow. Once the food source was gone, cells would continue to divide until the cells had an energy level below the threshold of 8. Then the cells would all go at the same time to search for a new food source. This mass exodus of leaving looked almost like a tidal wave of cells. Since all of the cells had been occupying the same location, a small amount of the energy-producing chemical was distributed throughout the environment. The cells would take advantage of this residual amount which helped them stay alive long enough to find the next energy source.

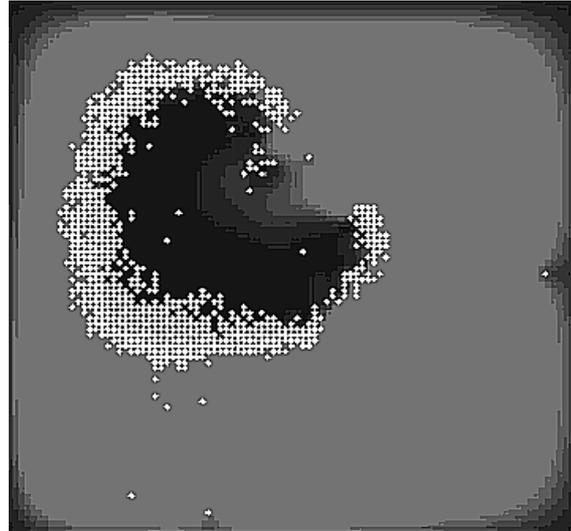


Figure 9: A Mass Exodus of Cells Looking for a New Energy Source

The mass exodus after an energy source was depleted is shown in Figure 9. The center of the group of cells is where the previous energy source was located.

4 Conclusions

The goal of this research was to explore the origins of multicellularity in an artificial life setting to gain insight into the evolution of emergent behaviors. Two primary experiments were conducted, a simulation of the aggregation of cellular slime molds and an evolving ecology of cells that were programmed with Cartesian Genetic Programming.

The slime mold aggregation experiment showed behavior characteristic of slime molds in nature. Cells were created in random locations and began to aggregate into a single group, resembling a more complex organism.

The use of Cartesian Genetic Programming was successful in producing cell behaviors that were at the same time novel and successful. The cells evolved several interesting strategies that sometimes benefited from group behavior. Behavior such as huddling and blocking were discovered. Creative uses of evolving thresholds were also used by groups of cells to improve cell division strategies. Instead of intuitive behavior, what evolved often resulted in surprising strategies that would not have been thought of otherwise. While the cells did not have the ability to permanently join together and become multicellular organisms, cooperative strategies were witnessed that could be considered first steps towards multicellularity.

The implementation provided a good base for the experiments performed in this research and should

provide a good test bed for a wide variety of future experiments.

5 Next Steps

Further development can be made to the program that might lead to more complex cellular behavior. Additional functionalities that might provide interesting results are cell adhesiveness, growing sizes of cells, and the ability for groups of cells to move at different rates than single cells.

The code used for these experiments could fairly easily be extended to allow predator/prey scenarios, parasites, symbiotic relationships, bottom feeders, etc. Other ideas are the use of a constant or periodic source of energy such as sunlight, the addition of more chemicals, and possibly even a current or stream-like effect. Combining adhesiveness with the effects of a current might show why it is helpful for organisms to attach to something stationary to capture passing food. Adding functions to the cells or the increasing the complexity of the ecology using these ideas or others might lead to more creative group behavior and might show additional insight into multicellular development.

One of the effects of using Cartesian Genetic Programming to evolve cell behavior was the high degree of neutrality and redundancy apparent in the cell's genotypes. It was observed that this feature seemed to have a positive effect when evolving the cells, especially during times of rapid environmental change. More work could be done in studying the specific impacts of neutrality and genetic drift in this context. Perhaps a comparison could be made between a population that allowed neutrality and one that didn't. To prohibit neutrality, genetic mutations could be restricted to those that made a change to the phenotype mapping.

References

- Agarwal, Pankaj (1995). The cell programming language. *Artificial Life*, 2 (1):3777.
- Bonner, John Tyler. (2001). *First Signals: The Evolution of Multicellular Development*. Princeton, NJ: Princeton University Press.
- Epstein, Joshua M. and Axtell, Robert (1996). *Growing Artificial Societies: Social Science from the Bottom Up*. Cambridge, MA: The MIT Press.
- Fleischer, Kurt and Barr, Alan H. (1992) A simulation testbed for the study of multicellular development: The multiple mechanisms of morphogenesis. In C. Langton (ed), *Artificial Life III*, London: Addison-Wesley, pp. 389-416.
- Johnson, Steven. (2001). *Emergence: The connected lives of ants, brains, cities, and software*. London: Penguin Press.
- Keller, E. and Segel, L. (1970). Initiation of slime mold aggregation viewed as an instability, *Journal of Theoretical Biology*. 26, pp. 399-415.
- Kessin, Richard H. (2001). *Dictyostelium: Evolution, Cell Biology, and the Development of Multicellularity*. Cambridge: Cambridge University Press.
- Kirschvink, Joseph L.; Ripperdan, Robert L. and Evans, David A. (1997). Evidence for a Large-Scale Reorganization of Early Cambrian Continental Masses by Inertial Interchange True Polar Wander. *Science*, 277:541-545.
- Koza, John R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.
- Krugman, Paul. (1996). *The Self-Organizing Economy*. Malden: Blackwell Publishers.
- Michod, R. E. and Roze, D. (1999). Cooperation and conflict in the evolution of individuality. III. Transitions in the unit of fitness. In: C. L. Nehaniv, editor, *Lectures on Mathematics in the Life Sciences*, American Mathematical Society, Vol. 26: 47-91.
- Miller, Julian and Thomson, Peter (2000). Cartesian Genetic Programming. In R. Poli, J.F. Miller, W. Banzhaf, W.B. Langdon, J.F. Miller, P. Nordin, T.C. Fogarty (eds), *Proceedings of the 3rd International Conference on Genetic Programming (EuroGP2000)*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, Vol. 1802, pp. 15-17.
- Ray, Thomas S. (1994). An evolutionary approach to synthetic biology: Zen and the art of creating life. *Artificial Life*, 1(1/2), pp. 179-209.
- Resnick, Mitchel (1994). *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*. Cambridge, MA: MIT Press.
- Shelling, Thomas C. (1978). *Micromotives and Macrobehavior*, New York: W. W. Norton.
- Vassilev, Vesselin K. and Miller, Julian F. (2000) The advantages of Landscape Neutrality in Digital Circuit Evolution. In J.F. Miller, A. Thompson, P. Thomson, and T. Fogarty (eds), *Proceedings of the 3rd International Conference on Evolvable Systems: From Biology to Hardware (ICES2000)*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, Vol. 1801, pp. 252-263.
- Yu, Tina and Miller Julian F. (2001) Neutrality and Evolvability of a Boolean Function Landscape. In J.F. Miller, M. Tomassini, P. L. Lanzi, C. Ryan, W. Langdon (eds), *Proceedings of the 4th International Conference on Genetic Programming (EuroGP2001)*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, Vol. 2038, pp. 204-217.