Using a Genetic Algorithm to Evolve Cellular Automata for 2D/3D Computational Development

Shape Generation

Arturo Chavoya
arturo.chavoya-pena@sip.univ-tlse1.fr

Yves Duthen
yves.duthen@univ-tlse1.fr

Université de Toulouse 1 *
1, Place Anatole France
31042 Toulouse, France
Tel. +33 (561) 12 87 93

ABSTRACT

Form generation or morphogenesis is one of the main stages of both artificial and natural development. This paper provides results from experiments in which a genetic algorithm (GA) was used to evolve cellular automata (CA) to produce predefined 2D and 3D shapes. The GA worked by evolving the CA rule table and the number of iterations that the model was to run. After the final chromosomes were obtained for all shapes, the CA model was allowed to run starting with a single cell in the middle of the lattice until the allowed number of iterations was reached and a shape was formed. In all cases, mean fitness values of evolved chromosomes were above 80%.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE—*Problem Solving, Control Methods, and Search*

General Terms

Algorithms

Keywords

Genetic Algorithms, Cellular Automata, Computational Development

1. INTRODUCTION

Computational Development is the study of artificial models of cellular development, with the objective of understanding how complex structures and forms can emerge from a few initial cells [3]. One of the crucial stages of an organism's development is that of form generation or morphogenesis, where the fundamental layout of the individual is to be defined. Self-organization plays a central role in shape formation, since within individual cells there exists no preconceived blueprint specifying how to construct a shape. Shapes emerge mainly from the local interaction of cells.

*Image Synthesis & Virtual Reality Team

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
ACM 1-59593-186-4/06/0007.

Cellular automata (CA) have previously been used to study shape formation, as they provide an excellent framework for modeling local interactions that give rise to emergent properties in complex systems [2]. On the other hand, genetic algorithms (GAs) have been extensively used in the past in a wide range of applications, and in particular they have previously been used to evolve CA to perform specific tasks [1, 2, 4]. In this paper we describe research on using GAs to evolve CA to create predefined 2D and 3D shapes.

2. CELLULAR AUTOMATA

Two different regular lattices were studied, a 2D (33×33) and a 3D ($17 \times 17 \times 17$) lattice. The set of cell states was defined as $\Sigma = \{0, 1\}$, where 0 means an empty cell and 1 an occupied or active cell. The three different interaction neighborhood templates η considered are presented in Figure 1. The CA's rule ϕ was defined as a lookup table that determined, for each local neighborhood, the state (empty, occupied) of the objective cell at the next time step [4]. For a binary-state CA, these update states are termed the rule table's "output bits". The lookup table input was defined by the binary state value of cells in the local interaction neighborhood $\eta_0\eta_1...\eta_n$. Starting with an active cell in the middle of the lattice, the CA algorithm was applied allowing active cells to reproduce according to the CA rule table and until the indicated number of iterations was attained. Asynchronous updating of cells was chosen for the CA implementation. During an iteration of the CA algorithm, the order of reproduction of active cells was randomly selected.

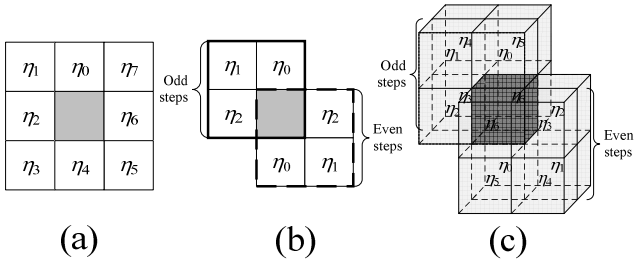

Figure 1: Interaction neighborhoods. (a) Moore, (b) 2D Margolus, and (c) 3D Margolus. The objective cell is depicted in a darker color.

3. GENETIC ALGORITHM

The GA uses tournament selection with mutation and single-point crossover as genetic operators. The initial population consisted of 500 binary chromosomes chosen at random. Tournaments were run with sets of 3 individuals randomly selected from the population. Crossover and mutation rates were 0.60 and 0.015, respectively. Finally, the number of generations was set at 50.

Chromosomes contained two fields: the *control field*, which coded for the number of steps that the CA algorithm was allowed to run, and the *action field*, representing the CA lookup table's output bits in lexicographical order of neighborhood. Fitness was defined as $f = (ins - \frac{1}{2}outs)/des$, where ins is the number of filled cells inside the desired shape, $outs$ is the number of filled cells outside the desired shape, and des is the total number of cells inside the desired shape [2]. During the course of a GA experiment, each chromosome produced in a generation was fed to the corresponding CA model, which was allowed to run for as many iterations as indicated in the chromosome's control field. Fitness was evaluated after the model stopped and a shape was formed. This process continued until the maximum number of generations was reached and then the individual with the highest fitness value was selected.

4. RESULTS

The GA was used to evolve the lookup table and the number of iterations for the desired shapes. Starting with one active cell in the middle of the CA lattice, cells were allowed to reproduce using the lookup table found by the GA and for as many iterations as indicated in the chromosome's control field. Since the CA algorithm uses asynchronous updating with the order of reproduction of cells randomly selected, a particular shape and fitness can slightly change on different runs of the CA algorithm for the same chromosome. For this reason, fitness mean and standard deviation from 100 runs of the CA algorithm are reported for all final chromosomes.

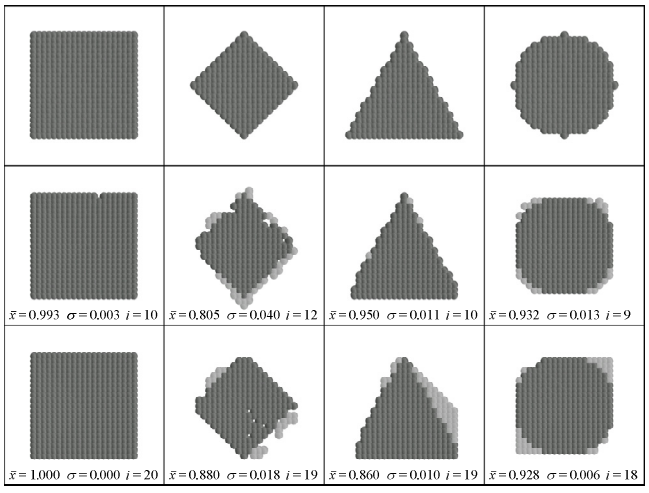

Figure 2: 2D shapes. Desired shapes (upper row), and shapes obtained with the Moore (middle row), and the 2D Margolus (lower row) models.

Figure 2 shows results for 2D shapes. For ease of visualization, cells that fall outside the desired shape are shown in light gray. Fitness mean ($\bar{x}$) and standard deviation (σ) from 100 runs of the final chromosomes, as well as the evolved number of iterations (i) (coded in the control field), are shown below the corresponding shape.

The desired shapes and the evolved 3D shapes are presented in Figure 3. As in the 2D case, cells outside the desired shape are shown in light gray. Fitness mean ($\bar{x}$) and standard deviation (σ) from 100 runs of the CA algorithm, and the evolved number of iterations (i) are also presented.

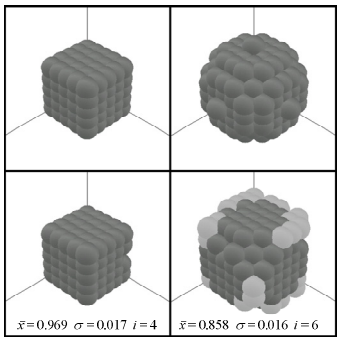

Figure 3: 3D shapes. Desired shapes (upper row), and shapes obtained with the 3D Margolus model (lower row).

5. CONCLUSIONS

In general, the framework used proved to be suitable for modeling simple shapes, but more work is needed to explore shape formation of more complex forms, both in 2D and 3D. It is also desirable to study shape formation using a more elaborate model, such as that provided by genetic regulatory networks. Furthermore, in order to build a more accurate model of the development process, interaction with the environment and other artificial organisms is necessary.

The long-term goal of this work is to study the emergent properties of the artificial development process. It can be envisioned that one day it will be feasible to build highly complex structures arising mainly from the interaction of myriads of simpler entities.

6. REFERENCES

[1] R. Breukelaar and T. Bäck. Using a genetic algorithm to evolve behavior in multi dimensional cellular automata: emergence of behavior. In *GECCO '05*, pages 107–114, 2005.

[2] H. de Garis. Artificial embryology and cellular differentiation. In P. J. Bentley, editor, *Evolutionary Design by Computers*, pages 281–295. Morgan Kaufmann Publishers, Inc., San Francisco, USA, 1999.

[3] S. Kumar and P. J. Bentley. An introduction to computational development. In S. Kumar and P. J. Bentley, editors, *On Growth, Form and Computers*, chapter 1, pages 1–44. Academic Press, New York, NY, USA, 2003.

[4] M. Mitchell, J. P. Crutchfield, and R. Das. Evolving cellular automata with genetic algorithms: A review of recent work. In *Proceedings of the First International Conference on Evolutionary Computation and its Applications (EvCA'96), 1996.*, Moscow, Russia, 1996.