



HAL
open science

Using cellular automata and genetic algorithm to generate 2D forms

F Aïboud, N Grangeon, Sylvie Norre

► **To cite this version:**

F Aïboud, N Grangeon, Sylvie Norre. Using cellular automata and genetic algorithm to generate 2D forms. META 2010, Oct 2010, Djerba, Tunisia. hal-01741125

HAL Id: hal-01741125

<https://hal.science/hal-01741125>

Submitted on 22 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Using cellular automata and genetic algorithm to generate 2D forms

F. Aïboud, N. Grangeon and S. Norre

LIMOS, 24 avenue des Landais. BP 10125. 63173 AUBIERE cedex, France
[fazia.aiboud, sylvie.norre, nathalie.grangeon}@moniut.univ-bpclermont.fr](mailto:{fazia.aiboud, sylvie.norre, nathalie.grangeon}@moniut.univ-bpclermont.fr)

Keywords: Cellular automata, genetic algorithm, mutual information.

1. Introduction

The morphogenesis is an important process which allows living things to develop organised structures thanks to the interactions of cells either among themselves or with their environment. In order to understand these processes, we are interested in producing 2D predefined shapes from a cell with cellular automata models.

Cellular automata are discrete dynamic systems in space and time with simple local interactions but complex global behaviour [1]. They consist in a regular cells lattice. Each cell can take a state chosen among a finite set of states. The state of a cell evolves according to the states of its neighbouring cells through evolutionary rules. These rules are applied synchronously for each cell during a number of iterations.

The number of the evolutionary rules increases with the number of considered neighbouring cells. When we use a cellular automaton with 2 states and a simple neighbourhood system such as Von Neumann system with 4 neighbouring cells, 2^{16} evolutionary rules are possible to evolve the cellular automaton. Each rule gives behaviour for the cellular automata.

2. Statement of the problem

In this paper, we consider generation of 2D shapes and we use a 2D cellular automaton. Each cell can take two states “0” and “1” representing respectively an empty and an occupied cell. The new state of each cell depends on its state and states of its neighbouring cells through evolutionary rules. To generate new states, we apply synchronously these rules for all cells.

To generate 2D shapes with cellular automaton, two problems arise. The first is how to choose a rule allowing to the cellular automaton to evolve toward the desired behaviour (desired shape). The second is how often to apply this rule to attain this shape. In the literature, several works propose genetic algorithms to solve these problems. Mitchell et al [2] and Back et al [3] have applied a genetic algorithm to evolve behaviour in 1D cellular automaton. Garis [4] and Chavoya [5] have generated predefined forms with cellular automata piloted by genetic algorithm.

The model presented by Chavoya is a cellular automaton with two states: empty cell and occupied cell. The evolutionary rules are applied only to empty cells, the occupied cells remain unchanged. Moreover, the used fitness function does not consider all the transitions between states. This model presents the advantage to be easy to implement and to have an efficient solution encoding. But, we cannot use it to modelize hollow shapes. Also, when the number of states increases, the solution encoding is complicated and the fitness function used to compare the forms is difficult to calculate.

3. Proposition

In this paper, we propose an improvement of Chavoya’s model. This model is a cellular automaton which allows to the occupied cells to change states allowing creating different full and hollow shapes. This model allows considering biological phenomena such apoptosis and necrosis. We use a genetic algorithm to determine the different evolutionary rules of cellular automata and the number of iterations necessary to obtain a desired shape. To do so, we propose a new encoding and new fitness function.

Solution encoding

Each chromosome is coded in binary according to two fields: a control field and an action field. The control field finds the necessary number of iterations to the cellular automaton for obtaining the desired shape; it is the number of necessary bits for coding the maximal number of iterations. The length of this field depends on the size of the lattice of the cellular automaton. The action field is the vector of new states of cells according to the combination of its neighbouring cells. The length of this field is the number of possible combinations of the neighbourhood system.

Fitness function

The fitness function proposed to measuring the similarity between the obtained shape X and the desired shape Y is the mutual information. It is the average information provided by the probability of a realization on the probabilities of another realization Y [6]. It is given by the following equation:

$$MI(X, Y) = H(X) + H(Y) - H(X, Y).$$

$H(X)$ and $H(Y)$ are marginal entropies. They are defined as follows:

$$H(X) = -\sum_{x=1}^n p(X = x) \ln p(X = x) \quad \text{and} \quad H(Y) = -\sum_{y=1}^n p(Y = y) \ln p(Y = y).$$

With:

- x and y are respectively the different states of cells in X and Y
- n is the number of states.
- $H(X, Y)$ is the joint entropy.

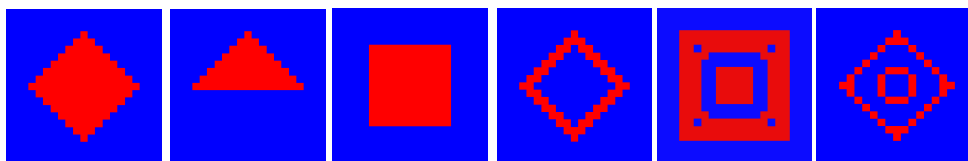
$$H(X, Y) = -\sum_{x=1}^n \sum_{y=1}^n p(X = x, Y = y) \ln(p(X = x, Y = y)).$$

$p(X = x, Y = y)$ is the joint probability of X and Y , it can be presented as two-dimensional histogram.

The advantage of this measure is the consideration of all possible transitions from one state to another. These transitions are given by the joint probabilities with x a state of a cell in the obtained shape and y state in the shape to generate. When two forms are identical, their mutual information is maximal.

4. Results

We have applied our model to generate different full and hollow shapes (square, triangular and diamond shapes). Our approach gives promising results. Some different full and hollow forms tested are given in the figure above:



Different tested forms.

References

- [1]: S.Wolfram, "Universality and complexity in cellular automata". Physica D10. North Holland 1984.
- [2]: M.Mitchell, J.P. Crutchfield and R. Das, "Evolving Cellular Automata with Genetic Algorithms: A review of recent work". 1st International Conference on Evolutionary Computation and its Applications. Russia1996.
- [3]: T.Back and R.BreuKelaar, "Using Genetic Algorithms to Evolve Behavior in Cellular Automata". 4th International Conference on Unconventional Computation. Spain 2005.
- [4]:H.D.Garis, "Artificial embryology and cellular Differentiation". Evolutionary Design by Computers, Morgan Kaufmann Publishers. USA1999.
- [5]: A.Chavoya, "Using a genetic algorithm to evolve cellular automata for 2D/3D computational development". 8th annual Conference on Genetic and Evolutionary Computation. USA 2006.
- [6]: R.Gray, "Entropy and Information Theory". Springer-Verlag. USA 1990.