# Investigating the Evolvability of Biologically Inspired CA

David Basanta[1], Mark A. Miodownik[1], Peter J. Bentley[2], and Elizabeth A. Holm[3]

[1] King's College London, Mechanical Engineering Department, Strand, London, UK
{david.basanta,mark.miodownik}@kcl.ac.uk
[2] University College London, Computer Science Department, Gower St., London, UK
p.bentley@cs.ucl.ac.uk
[3] Sandia National Labs, Albuquerque, New Mexico, USA
eaholm@sandia.gov

**Abstract.** The developmental processes studied by biologists are emergent self organised processes that are the result of natural evolution. Developmental biology can be a good inspiration for anyone interested in evolving self organised discrete systems like Cellular Automata: nature has proved that evolving self organisation is possible.
In this paper, we will describe EmbryoCA, a model of 3D Cellular Automata for pattern generation. Developmental biology has been used as an inspiration to design EmbryoCA and make the model more gradual in the hope of having a more evolvable type of CA. Also, experiments comparing the evolvability of different setups of EmbryoCA with a conventional CA model are shown.

## 1 Introduction

The study of evolvable mechanisms capable of generating rich patterns is interesting for researchers working in different scientific fields. As an example of this interest, work has been carried out to use evolutionary computing to find spatial three dimensional (3D) patterns since those patterns can be used to characterise microstructures for its use by material scientists [1].

Evolving patterns using a linear representation in which every feature is directly encoded into a gene, poses significant problems in terms of scalability and seriously limits the complexity that can be handled by such systems [2]. Nature does not work this way. In nature, non-linear, self-organising processes known colectively as embryogenesis or development, grow the phenotype from a compact representation.

Cellular Automata (CA) can be used to model developmental processes since their compact rule sets produce rich results - akin to a compact genotype developing into a complex phenotype. Unfortunately they tend to be very sensitive to small changes to their rule sets: a small change is amplified time step after time step and eventually, the results generated by two very similar CA ends up being very different. It is due to that sensitivity that is difficult to evolve them to

perform computations [3] or do something like pattern generation [4]. Evolution needs to be able to do smooth, gradual changes to be effective [5], in systems that can disrupted by minor modifications, adaptative improvements by random selection and mutation cannot occur [6].

In this paper we will describe a CA model for pattern generation whose design has been inspired by developmental and cellular biology in order to be more evolvable than conventional CA models. The next section will provide an introduction to some concepts needed to understand the EmbryoCA model and the experiments done using it. Next, the EmbryoCA model will be described and after this, the experiments performed to compare the evolvability of EmbryoCA with conventional CA will be shown. Finally the results will be discussed and conclusions presented.

## 2 Background

This section will introduce a few notions about CA, developmental biology and pattern characterisation functions that will be needed to understand the EmbryoCA model and the experiments performed.

### 2.1 Cellular Automata

Von Neumann and Ulam introduced CA in the 1940s for the study of self replication [7]. CA are dynamical systems made of a discrete number of elements arranged into a lattice (normally one, two or three dimensional ones are used). These elements, or automata, can be in a number of discrete states and change according to a finite number of rules that determine the state of an automaton given the state of the neighbours of that automaton. Time is also discrete and is divided into time steps.

CA are interesting because being simple, they can show very complex behaviours and patterns. They have been used to model and solve all sorts of problems. They have been used to study electro-static self-assembly processes [8], density classification [9], pattern formation [10], they have been used in materials science [11], testing digital circuits [12] and to model developmental processes [13].

There are several models of CA. Differences between two different CA models could be in the definition of neighbourhood, if the rules are applied synchronously or asynchronously or the number and type of states in which an automaton could be at any given time step.

### 2.2 Effector Automata

Effector Automata (EfA) is a model of CA designed and created by Lohn and Reggia to evolve self replication [14]. In the EfA model, automata are autonomous elements capable of moving, creating copies of themselves and dying, in an otherwise empty lattice. The output of a rule in an EfA is the action to be

performed by the automaton when its internal state and its configuration of neighbours are the ones specified in the rule.

### 2.3 Developmental biology

Nature has managed to create morphologies and patterns of extraordinary complexity and sophistication [15] and most of them are the result of processes studied in developmental biology. Development is the process of construction and growth of organisms that emerges from the interactions between proteins and genes and cells, with the environment [2]. Genes encode proteins and proteins perform almost all the tasks needed for development such as catalysing the synthesis of other cell components, regulating the expression of genes as well as making inter cellular communication possible. As a consequence of the interplay of these elements, structures emerge from a simple group of cells that divide, grow and change shape [16]. It is, without doubt, thanks to developmental processes that complex multicellular organisms are possible.

From a computer science or engineering perspective, development is about construction and self organisation [2] and it can successfully be applied to solve complex problems in other areas of science and engineering [17].

### 2.4 Two point correlation

Two point correlation functions are widely used in materials science to characterise patterns [18] and in general they can be a useful characterisation of 2 or 3D lattices. One of the reasons that make them quite useful in a diverse number of areas is that they are very general and hence, can be used to characterise any pattern, from a cell to a painting. The two point correlation function is described in the following equation:

$$f(d) = \frac{1}{N_{\mathrm{S}}^2} \sum_{i=0}^{N_s} n_d \tag{1}$$

where d is the correlation distance, $\mathrm{N_S}$ is the total number of particles that belong to a given phase in the matrix and $\mathrm{n_d}$ is the number of particles of the phase being characterised that are separated at distance d from particle i.

## 3 EmbryoCA

EmbryoCA, is a model of 3D CA inspired by developmental biology and built using the principles of the EfA model. The main aims of the EmbryoCA model are to be able to grow binary 3D spatial patterns [19] and to be evolvable. In this work, the evolvability of a CA model is considered as the capability of the model of being effectively and efficiently modified by evolution through gradual change.

### 3.1 EmbryoCA as a developmental biology model

The automata in the EmbryoCA model are cell-like effector automata. They are autonomous entities capable of moving in a 3D space, creating copies of themselves and dying. Each automaton has an identical rule set or genome. The genes are regulated by both the environment (that is, the other automata in the neighbourhood) and the elements created by expressing the genes that, in this model, will be called proteins. A protein may promote certain types of actions (for instance, moving to another location) or may inhibit the expression of some gene (for instance, inhibiting the expression of the gene that creates the protein that promotes the automaton to move). As usual with CA models, both time and space are discrete and each time step, the appropriate genes of each automaton are expressed and the interplay of the different proteins will determine which action, if any, will take place.

### 3.2 Description of EmbryoCA as a CA model

An EmbryoCA is specified with a list of rules that have the following format:

`if (`*`variable`* `= number) then do` *`consequence`*

where *variable* can be either the internal variable that keeps track of the number of divisions that the automaton has gone through, or the number of neighbours in one of the six directions of a semi totalistic Moore neighbourhood (north, south, east, west, up and down) [20]. There are two types of *consequences* in a rule: actions (move, divide and die) and antiactions (inhibiting the automaton from either moving, dividing or dying). At a given time, an automaton may have more than one applicable rule and a conflict resolution mechanism will decide what action to follow.

For each timestep, every automaton follows this algorithm:

1. Get list of rules whose precondition is true.
2. for every applicable rule:
   (a) if the consequence is an action, increase the counter associated with the action.
   (b) if the consequence is an antiaction, decrease the counter associated with the action.
3. Pick the action with the higher counter.
4. If selected action's counter is higher than threshold, execute action.

The Initial Configuration (IC) of an EmbryoCA is always one single automaton, or zygote, placed in the middle of a 3D lattice

Figure 1 shows one example of 3D binary pattern (transparent and blue) grown by an EmbryoCA.

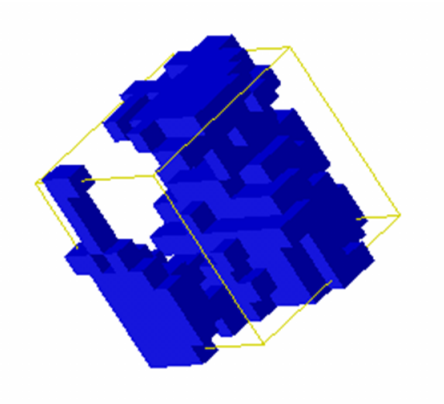

**Fig. 1.** Example of 3D binary spatially discontinous pattern grown by an EmbryoCA.

## 4 Experiments

The purpose of the experiments is to compare the evolvability of the EmbryoCA model with conventional CA models. Due to the discrete nature of CA, change is normally not gradual and the consequence is that they are not very evolvable. The experiments will assess what impact changes in the rule set have on the final 3D pattern generated.

The model of CA representing conventional CA is described in table 1. It is a 3D CA in which every cell is occupied by an automaton. Rules determine in which of the two possible states will the automaton be in the next time step. At any given time and for any automaton, one and only one rule of the rule set is applicable. What determines if a rule will be applied is the configuration of neighbours of the automaton under consideration. The neighbourhood of an automaton in this model is based on the Moore neighbourhood. In 3D that means that 26 neighbours are considered when evaluating rules. The way the neighbourhood is used to evaluate rules is semi totalistic: only the overall count in one of the six possible directions (north, south, east, west, up and down) is considered. This means that in each of the six directions, an automaton may have any number between 0 and 9 neighbours. As a consequence of this type of neighbourhood and given that every possible configuration of a neighbourhood has to be included in the rule set, the rule set of a CA in this model is made of $1 * 10^6$ rules. These types of rule sets are frequently used by researchers that want to evolve CA like [21] and [14].

A random walk will be used to compare the gradualness of change in both models of CA. A random walk can be described by the following algorithm :

1. `Randomly create a rule set and in the conventional CA, an IC.`
2. `Grow a pattern with it iterating the rule set 100 time steps, call it original pattern.`

| Feature | Value |
|---|---|
| CA dimensions | 20x20x20 |
| Automata states | 2 (B&W) |
| Neighbourhood type | Moore neighbourhood |
| Neighbours count | Semi totalistic (6 directions) |
| Rule set size | 1000000 |
| Time steps | 100 |

**Table 1.** Description of the conventional model of CA used in the experiments.

3. Make a number of random changes in the rule set: randomly pick the rules to be replaced with randomly created new ones.
4. Grow a pattern with the modified rule set in a 100 time steps, call it new pattern.
5. Characterise the new pattern with a 2 point correlation
6. Measure the difference between the characterisations obtained with the 2 point correlation function of the original and the new pattern. Record the difference.
7. Call the new pattern original pattern.
8. Repeat the process from step 3 for 1000 times.

Figure 2 shows how a two point correlation function is used to measure the difference between any two 3D patterns. This function is used to characterise with a distribution both, the CA before and after the change has been made on the rule set. The bigger the difference between the two distributions, the less similarity between the two patterns generated.

Three different systems have been compared in these experiments, one of them is a conventional CA described before and the other two are different versions of the EmbryoCA model, one with rule sets of size 100 and the other with rule sets containing 1000 rules. Each system is tested with random walks of a thousand steps each in which every CA is iterated for a 100 time steps. The rate of change over the previous rule set is 1, 5, 10 and 20% of the rules. A change of 10% means that 10 of every 100 rules are completely erased and replaced by new randomly generated ones. For each percentage, the system is tested with random walks 10 times.

## 5 Results and analysis

Figure 3 shows the main results of the experiments: the average change, as measured using a 2-point correlation function, for each of the models used in the experiments. Some of the results that can be appreciated in the figure were expected. For instance that where higher rates of change between two consecutive rule sets in the random walk are used, the differences in the phenotype grown with them are higher too. It can also be seen that the conventional CA

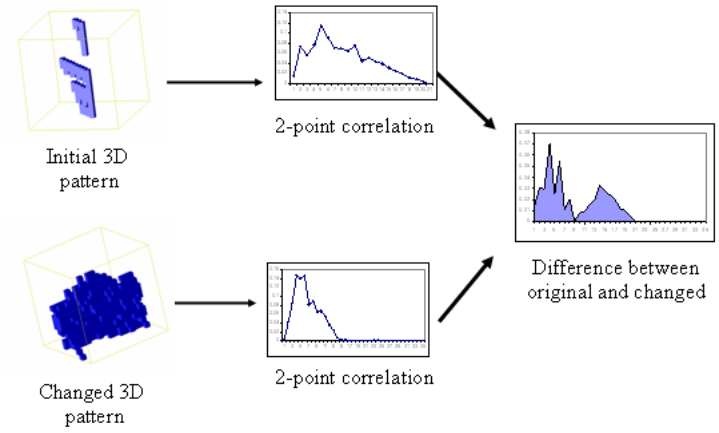


**Fig. 2.** Example of how a two point correlation function is used to characterise and compare two different rule sets.

used, whose rule set contains $1 * 10^6$ rules, is less sensitive to changes than the EmbryoCA with a hundred rules but more than the EmbryoCA with 500 rules.

Figure 4 shows a comparison between two of the random walks performed for the experiments. Both of them represent the sequence of differences between the patterns grown by CA before and after a change. In this case, the rate of change was 1% of the rules. One of the series represents the conventional CA and the other represents the EmbryoCA with 500 rules. It can be appreciated that, compared to the EmbryoCA, changes in the rule set of a conventional CA are more likely to either produce no effect whatsoever or represent a steep change in the phenotype grown as measured by the two point correlation function. It is precisely this type of behaviour that makes the evolvability of conventional CA models comparatively poor.

Figure 5 shows another comparison between one of the random walks performed with the conventional CA model and a random walk done with the EmbryoCA with 500 rules model. In this case, the rate of change between consecutive CA is 20%. While it is true that some of the changes in the rules of the EmbryoCA model can have a tremendous impact on the phenotype, for the vast majority of the cases, the EmbryoCA model is less sensitive to random mutation.

It is important to note that the gradualness shown by the EmbryoCA model is not an artifact of redundancy in the rule set (i.e., part of the rule set never being used). Whereas in the conventional CA model most rules are never used and therefore most changes in the rule set cannot have any effect on the patterns generated, in the EmbryoCA virtually all changes in the rule set will affect rules that will be used and therefore, will have an impact, albeit small, in the final

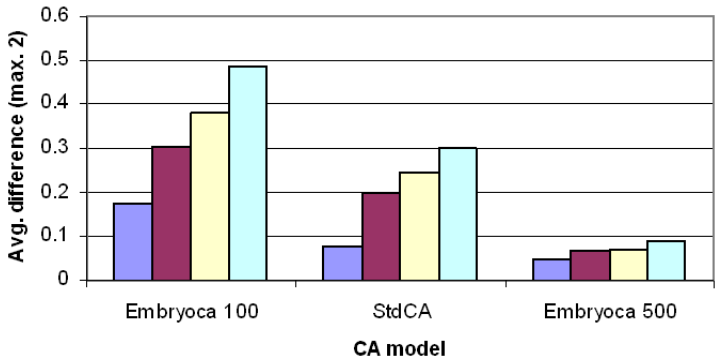


**Fig. 3.** Average difference between two consecutive rulesets, after and before a change has been made. For every CA used, rates of change equal to 1, 5, 10 and 20% of the rule set are shown grouped together. The maximum difference between any two individuals is 2.

pattern. To test that, the models were tested to see what percentage of the rule set is actually used, resulting in data shown in figures 6 and 7.

Figure 6 shows the percentage of rules used for an EmbryoCA with 100 and an EmbryoCA with 500 rules, in 100 different trials, each of them representing a different rule set.

Figure 7 represents the same results when the trials were performed using the conventional CA model. It is easy to see that every time one single rule is changed in an EmbryoCA, that has to have some repercussion in the way the pattern is grown. One reason that could explain why despite this, changes in EmbryoCA tend to be gradual is that the function of rules in the model is not to determine what an automaton should do in a specific situation, as it is the case in conventional CA models. Their function is to help promote some specific action. This means that the actions taken by an automaton are likely to be the result of a number of rules promoting that action and not the responsibility of a single rule. It is for this reason that changes in the rule set slowly modify the phenotype instead of transforming it radically.

## 6 Conclusions

Developmental processes are key to evolve sophisticated and complex designs, but without graduality, the dynamical non linear and mostly discrete systems used to model development make evolution difficult if not impossible [2]. One way to achieve graduality is to use non discrete systems to model development [22], but nature is also discrete: genes are expressed or not, reactions do or do not

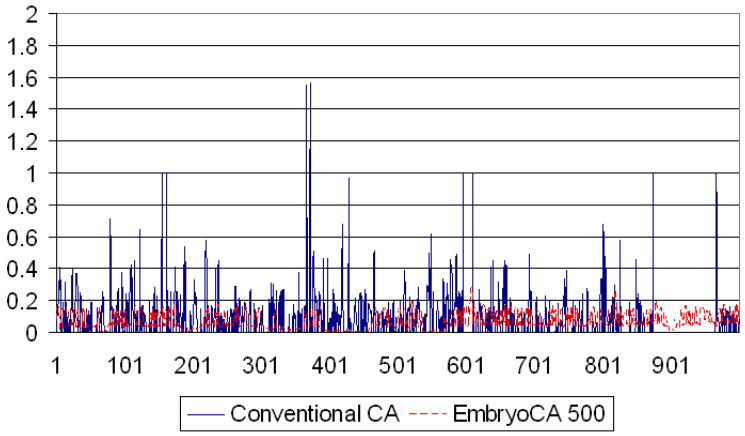


**Fig. 4.** Example of 2 random walks, one with the conventional CA and another with the EmbryoCA with 500 rules, with a rate of change of 1% of the rule set. Each time, 1% of the rules in the rule set are replaced by new ones and the difference between the patterns, before and after the change, calculated and recorded.

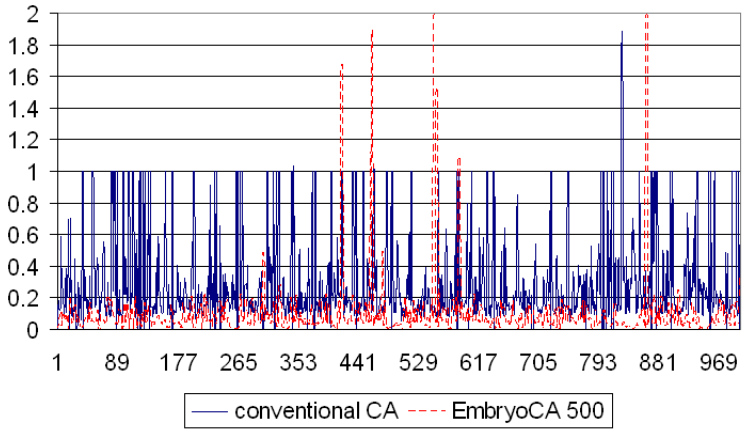


**Fig. 5.** Example of 2 random walks, one with the conventional CA and another with the EmbryoCA with 500 rules, with a rate of change of 20% of the rule set. Each time, 20% of the rules in the rule set are replaced by new ones and the difference between the patterns, before and after the change, calculated and recorded.

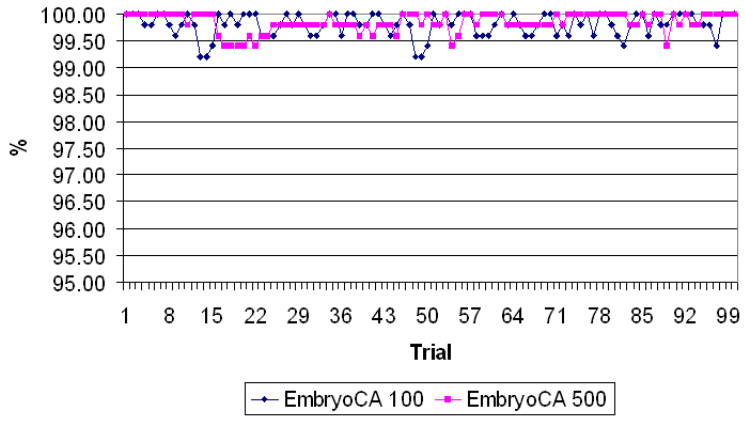


**Fig. 6.** Percentage of rules used in 100 EmbryoCA with 100 rules and with 100 EmbryoCA with 500 rules. In every case, more than 99% of the rules in the rule set were used.

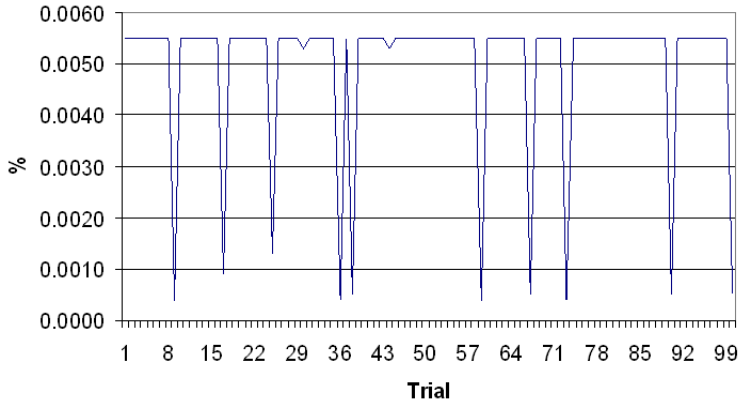


**Fig. 7.** Percentage of rules used in 100 conventional CA. The percentage of rules used is extremely low, always below $6 * 10^{-5}\%$.

take place, cells do or do not divide, there is no middle ground. And despite the discreteness of nature, development happens and this development gets shaped by evolution.

This paper has described EmbryoCA, a CA model designed to grow patterns, and has shown experiments that proof that it is a comparatively gradual CA model. This graduality means that changes in the patterns generated are proportional to the changes in the rule set. This behaviour contrasts with the more drastic effect that changes have on conventional CA models where they are more likely to either have no impact at all or change completely the resulting pattern. We believe that this graduality has the potential to make CA models that follow the principles of the EmbryoCA model more evolvable and better suited to help use development with Evolutionary Algorithms.

## Acknowledgements

This work was performed in part at Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

## References

1. Basanta, D., Miodownik, M.A., Holm, E.A., Bentley, P.J.: Designing the internal architecture of metals using a genetic algorithm. Proceedings of the Engineering design conference 2003. Professional Engineering Publishing Ltd. London, UK (2002)
2. Kumar, S., Bentley, P.J.: An introduction to computational development. In On Growth, form and development. Editors Kumar, S. and Bentley, P. J. Academic Press (2003)
3. Mitchell, M., Crutchfield, J.P., Hraber, P.T.: Evolving cellular automata to perform computations: Mechanisms and impediments. Physica D, 75, 361-369,1994 (1994)
4. Basanta, D., Bentley, P., Miodownik, M.A., Holm, E.A.: Evolving cellular automata to grow microstructures. Proc. of 6th European Conference on Genetic Programming (EuroGP 2003), 14-16 April 2003 (2003)
5. Dawkins, R.: The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe Without Design. Penguin (1986)
6. Kauffman, S.: The origins of order. Self-organization and selection in evolution. Oxford University Press, Oxford, UK (1993)
7. von Neumann, J.: Theory of self-reproducing automata. University of Illinois Press (1966)
8. Spillman, J.B., Zeng, T., Claus, R.O.: Modeling the electro-static self-assembly process using stochastic cellular automata. Smart Materlials and Structures **11** (2002) 623–630
9. Mitchell, M.: Computation in cellular automata. In T. Gramss, S. Bornholdt, M. Gross, M. Mitchell, and T. Pellizzari, Nonstandard Computation, pp. 95–140. Weinheim: VCH Verlagsgesellschaft, 1998. SFI Working Paper 96-09-074 (1996)

10. Miller, J.: Evolving developmental programs for adaptation, morphogenesis and self-repair. ECAL 2003, Dortmund, Germany (2003)
11. Raabe, D.: Cellular automata in materials science with particular reference to recrystallization simulation. Annual review of materials research,2002 (2002)
12. Corno, F., Reorda, M.S., Squillero, G.: Exploiting the selfish gene algorithm for evolving hardware cellular automata. Proceedings of the 2000 congress on evolutionary computation CEC00. IEEE Press (2000)
13. Bentley, P.J., Kumar, S.: The abcs of evolutionary design: Investigating the evolvability of embryogenies for morphogenesis. Genetic and Evolutionary Computation Conference (GECCO '99) July 14-17, 1999, Orlando, Florida, USA (1999)
14. Lohn, J., Reggia, J.: Discovery of self-replicating structures using a genetic algorithm. 1995 IEEE International Conference on Evolutionary Computing. (1995)
15. Ball, P.: The self made tapestry. Oxford University Press, Oxford (1999)
16. Wolpert, L.: Principles of development. 2nd edition. Oxford University Press, Oxford (2002)
17. Kumar, S., Bentley, P.: On growth, form and computers. Elsevier Academic Press, London (2003)
18. Mason, T.A., Adams, B.L.: Use of microstructural statistics in predicting polycrystalline material properties. Metall. Mater. Transactions 30A(1999) p. 969-979. (1999)
19. Basanta, D., Bentley, P., Miodownik, M.A., Holm, E.A.: Evolving and growing microstructures of materials using biologically inspired ca. Proc. of 2004 NASA/DoD conference on Evolvable Hardware (EH 2004), Seattle, 24-26 June 2004 (2004)
20. Moore, E.F.: Machine models of self reproduction. American Mathematical Society Proceedings of Symposia in Applied Mathematics 14 17-33 (1962)
21. Mitchell, M., Crutchfield, J., Das, R.: Evolving cellular automata with genetic algorithms: A review of recent work (1996)
22. Bentley, P.J.: Evolving fractal proteins. Proc of ICES 2003, the 5th International Conference on Evolvable Systems: From Biology to Hardware (2003)