LEARNING EMERGENCE

ADAPTIVE CELLULAR AUTOMATA FAÇADE TRAINED BY ARTIFICIAL NEURAL NETWORKS

MSc Adaptive Architecture & Computation | Bartlett | UCL | 2009

Maria Eleni Skavara

This dissertation is submitted in partial fulfillment of the requirements for the degree of Master of Science in Adaptive Architecture & Computation from University College London

Bartlett School of Graduate Studies | University College of London | September 2009

I, Maria Eleni Skavara confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Maria Eleni Skavara

Abstract

This thesis looks into the possibilities of controlling the emergent behaviour of Cellular Automata (CA) to achieve specific architectural goals. More explicitly, the objective is to develop a performing, adaptive building facade, which is fed with the history of its achievements and errors, to provide optimum light conditions in buildings' interiors. To achieve that, an artificial Neural Network (NN) is implemented. However, can an artificial NN cope with the complexity of such an emergent system? Moreover, can such a system be trained to compute and yield patterns with specific regional optima, using simple inputs deriving from its environment? Both Backpropagation and optimisation using Genetic Algorithms (GA) are tested to reassign the weights of the network and several experiments are conducted regarding the structure and complexity of both CA and NN. Here it is argued that in fact, it is possible to train such a system although the level of success is strongly dependent on the degree of complexity and the level of resolution and accuracy. By taking advantage of the structural attributes of certain CA that go beyond just a higher order stability, this dissertation suggests that such an evolutionary, computational approach can lead to adaptive and performative architectural spaces of high aesthetic value.

Word count: 9692

Keywords

Complexity, Emergence, Supervised Learning, Cellular Automata, Artificial Neural Networks, Supervised Learning, Backpropagation, Generative Algorithm, Adaptive

Contents

Acknowledgements

I would like to express my gratitude to those who helped me develop and accomplish this thesis starting from my tutors, Sean Hanna and Ruairi Glynn for their support, patience and inspiration,

Alasdair Turner for his valuable guidance during the implementation of my program,

my AAC colleagues, and especially Agata Guzik, Michal Piasecki and Sahar Fikouhi, for the constructive discussions on our common interests during the last months,

Konstantinos Papagiannopoulos for being there for me unconditionally,

and the people in Bartlett's workshop for assisting me in creating physical models and rapid prototypes to bring my adaptive fa[ça]de into life.

Introduction

The notion of controlling emergent systems is not new in the architectural discourse, even though architectural practice has only recently started keeping pace. Cybernetic architecture in the 1970's and 1980's and pioneers of computer-aided design such us Nicholas Negreponte and John Freizer, set the foundation for an innovative algorithmic design approach which looks into the inherent attributes of emergent systems and uses them for particular aspirations. The idea of creating machines that use artificial intelligence to imitate and adapt to natural environment were already introduced in late 60s by Gordon Pask with his 'musicolour machine'. However, how can these ideas be embedded in a consistent architectural process and change the way architectural problems are tackled?

The aim of this thesis is twofold, which is reflected in both the nature of background literature and methodology of this dissertation. The interest is equally set on the scientific discussion of whether a Neural Network can cope with the complexity of the emergent behaviour of CA and on the merits and potential of controlling this complexity in an architectural façade. The building façade is intentionally chosen as an architectural entity consisting of multiple requirements and attributes. Being by definition the regulator between internal and external space and between urban and human scale, constitutes an example of high complexity on its own.

Dealing with and controlling a complex emergent system like CA requires a deeper understanding of its inherent qualities. Being on the edge of order and chaos, these systems can be considered as controversial because to some extent they take control of the buildings behaviour out of the architects' hands. Various questions are raised regarding the unique character of CA, which in many cases exhibits degree of global stability, thus presenting a wide range of local interactions. Seeking for the critical balance between complexity and order, these questions led to numerous experiments with different CA rules ranging from *continuous* simple rules to complex *class four* rules and alternatives for the number of their possible states. An ongoing research conducted by Melanie Mitchell and her team in Santa Fe University (Mitchell, 1996) explores different ways of optimising and guiding CA configurations for predefined goals, and essentially highlights the difficulty to retain control on the complex emergence of CA patterns. However, this thesis aims to investigate the potential of having power over more detailed structural attributes of CA patterns to respond to their constantly changing environment.

But what is the purpose of using CA on a façade? The idea behind this suggestion lies in the distinct characteristic of several CA rules to transmit information in a top-to-bottom fashion and spreading signals throughout the grid. However, it very much depends on the type of CA in terms of both complexity and initial conditions, as it will be further discussed in this dissertation. Even more than that, the interest here is set predominantly into investigating the potential of acquiring regional optima on the CA lattices to control the light penetration specific areas of the interior. This thesis suggests that utilising the above intrinsic attributes, feeding the façade with a set of values measured on the top of the building, can be enough to generate a responsive shading system enhancing patterns of high aesthetic level in the same time. In other words, the initial configuration is provided with information and is responsible for spreading it to the rest of the grid. This could be enough on its own for simple tasks in a relatively stable environment. In the case of buildings though, conditions are significantly more complicated and change over the course of the year.

Thus, the further step of using an artificial Neural Network to train the system for a sample set to achieve explicit goals throughout the year, bypassing a series of exhaustive calculations. Placing the façade in a specific virtual model sets the guidelines regarding constant and varying conditions. Orientation, built surrounding environment and optimum light levels had to be considered as givens to allow for further experimentation on the learning side of things. Even though no thorough study of materiality and scale is conducted, a general outline regarding the aggregation of the panels and the translucence of their material needed to be decided for setting a consistent work-flow of assumptions and calculations and a foundation to build the experiments on. Given these, the idea is to train the façade in order to introduce it or to get it familiar with its context and let it then evolve.

Choosing an evolutionary approach in dealing with this architectural entity shifts the objective from an explicitly calculative hard-coded computerisation to controllable evolutionary computation. The scope of this system is not to come up with predefined patterns, but to be trained to understand how these patterns are created and become sophisticated enough to generate various different ones retaining local optima leading to a high level of responsiveness to variable conditions. Once this is achieved, it is suggested that setting some general requirements are enough for the system to evolve efficient patterns.

A selected set of data is used to train and test the system. However, aiming to develop a sophisticated façade, rather than one that purely memorises a set of 'right answers', the validation of the system becomes crucial and this is realised against "novel data". To that respect, the two different methods used in this thesis constitute and endeavour to find out the resolution of any possible control on that complex system. The trials using a multi-layer Neural Network with Backpropagation focus more on a thorough one-to-one mapping of the Initial Conditions of the façade's lattice, whilst the experimentations on optimising the connections of the Neural Network with a GA highlight the importance of obtaining local required averages, regardless of how the CA pattern starts off.

To what extent is it possible to obtain 'living' buildings with emergent behaviour, which respond to architects' and inhabitants' goals? Can a façade exhibiting beautiful, complex CA patterns be also controlled to yield a responsive behaviour to its context? Can Artificial Neural Networks cope with the inherent complexity of Cellular Automata and train a system adequately to be considered adaptive? This thesis proves that it is indeed possible, however it suggests that the answer to these questions is not a matter of "yes" or "no" and shifts the interest in a rather iterative process of findings and new experiments, which in turn generate new questions, subject to further research.

Background & Literature Review

Emergence and Complexity

Emergence is widely used when describing evolving systems that build up during their life span from simple configurations to highly complex ones. But how is the term 'emergence' defined? It turns out that due to the undergoing research through different disciplines there is not a unique answer to this question. John Holland in his 'Emergence from Chaos to Order' (Holland J., 1998) argues that: "it is unlikely that a topic as complicated as emergence will submit meekly to a concise definition, and I have no such definition to offer." Steven Johnson refers to emergence as the properties of a system that cannot be deduced from its parts and highlights the distinct characteristic of emergent systems to exhibit a movement from low-level rules to higher-level sophistication, due to what he calls a "bottom-up mindset". (Johnson S., 2001). A similar definition is suggested by Francis Heylighen; he describes emergent systems as higher order systems or 'wholes' which in general cannot be reduced to the properties of the lower order subsystems or 'parts'. Heylighen talks about a spontaneous creation of 'organised wholes' out of 'disordered' collections of interacting parts. (Heylighen F., 1989).

Some of the definitions mentioned highlight the properties of emergent systems without actually defining them, while others appear rather vague to be used in a programmatic approach. When the term emergence was introduced in architecture, it constituted a buzzword often included in sentences including cybernetics, evolutionary, morphogenetic or complexity. In creating emergent architectural systems though, the focus should be shifted towards the mathematical and computational primitive components that yield emergence, rather than the resulting complexity itself. Coupled with the advance of digital fabrication and the liberation from strict Cartesian geometries, an evolutionary design is increasingly dominating the current architectural form-finding process. These systems exhibit high levels of complexity on their own and one has no control on them, unless a comprehensive, solid foundation is set. The notion of emergence in architectural design process implies a multidisciplinary, programmatic approach, where various disciplines cooperate with each other in an inseparable fashion, much like the term itself suggests.

In nature, emergent systems are characterised by non-linear behaviour and their long-term target is to maintain themselves through time, in other words to survive in complex environments. How can these attributes be integrated into an architectural configuration? Cybernetics was the first attempt to do so, but since there was not a solid explanation of the natural phenomena at the time, often endeavours to mimic natural complex systems led to dead-end. Feedback is a key element in this approach. In order to maintain themselves, 'self-organised' systems have to somehow keep track of their preceding achievements and errors and revise their form and performance to achieve equilibrium though time, although one should keep in mind that most systems have a life span. Therefore, there are nested systems that cooperate with each other, starting from the parallel collective behaviour of separate components, to the whole system and its environment where dynamic exchanges occur. In the case of architecture, the notion of dominant individual landmarks is replaced with a dialogue between each building with its surrounding built and natural environment and its inhabitants, which in turn belong to a larger system. The behaviour of the parts leads to emergent wholes through repetition and interaction of simple rules, characteristic of Cellular Automata (CA).

Cellular Automata

> ***"Space in CA is partitioned into discrete volume elements called 'cells" and time progresses in discrete steps. Each cell of space is in one of a finite number of states at any one time. The physics of this logical universe is a deterministic, local physics. "Local" means that the state of a cell at time t+ 1 is a function only of its own state and the states of its immediate neighbors at time t."***
> ***(Langton C., 1990)***

These systems are deterministic because once the rule they obey and their initial state are set its future evolution is uniquely determined. (Find more on the history of CA in Appendix 1).

Contrary to our basic intuition about the way things normally work, Wolfram proves that simple rules and IC can lead to systems with immense level of complexity. He supports that the reason why our intuition guides us in believing that to create complex systems by definition we need to start with complexity is that we mostly focus on the result we want to achieve rather than the system that produces it. And is difficult to argue this opinion when for so many years we had no idea about handling such emergent behaviours even though they are so common in nature.

> ***"And in a sense the most puzzling aspect of it is that it seems to involve getting something from nothing."***
> ***Wolfram S. (2002)***

How can this notion of 'getting something out of nothing' be used in the architectural discourse? It seems like the only way to do so, is to comprehend the deeper structural principles embedded in these systems and focus on their iterative processes and activity that lead to the resulting structure itself. Steven Wolfram (Wolfram, 2002) conducted an extensive study on Cellular Automata, listing numerous possible CA rules and observing their behaviour through time. What he noticed is that, although the diversity one sees when he examines them is at first overwhelming, definite themes begin to emerge. In an attempt to classify the 256 rules he listed according to the complexity and transmission of information they present, Wolfram proposed the four qualitative classes of CA behavior (see Appendix 1 for table).

It's worth mentioning here that although Wolfram's four classes are numbered according to the complexity they exhibit, class 4 is in many cases between classes 2 and 3 because of the overall activity of the pattern. That is, class 1 and 2 systems quickly settle down to states with no further activity. But class 3 systems continue to have many cells that change at every step, so that they in a sense maintain a high level of activity forever. And the main reason why one sees class 4 systems in the middle of these classes is because the activity that they show neither dies out completely, as in class 2, nor remains at the high level seen in class 3.

Being a deterministic complex system, CA are characterised by sensitivity to initial conditions. One slight change in the first row can have major impact on the resulting pattern. As in most attributes of CA, the four distinct classes have different sensitivity to the initial configuration. In class 1, any change made soon dies out and the final outcome returns to the one before the change. In class 2, the change has more permanent effect on the pattern, but it remains localised, in comparison to class 3 where the change spreads to the whole system. Finally, in class 4, changes also spread but sporadically. Looking at these characteristics from a computational point of view, the above can be translated in how the four classes handle and spread given information. This will be pointed out through the experiments of this thesis, where various levels of consistency between IC and overall patterns yield different patterns and change the degree of control on their structural characteristics.

The rules analaysed above were the starting point of Wolfram's study on CA and probably the simplest rules. Systems can become much more complex when, for instance, instead of binary they can inherit more different states. Increasing the number of possible states in this kind of explicit rules yields a vast number of possible neighbourhoods. In this case, "totalistic" rules are introduced, where the sum of the neighbouring cells is taken into account and not the specific position of each state in the neighbourhood. Here, the rule is working as a lookup table with the possible sums of the states and the state that the new evolved cell should get. One could expect that increasing the complexity of the rules, the complexity of the resulting patterns would increase respectfully. It turns out that no significant change in terms of complexity was noticed when adding more complex rules, leading to the conclusion that beyond a certain point no more complexity can be yielded, as Wolfram states.

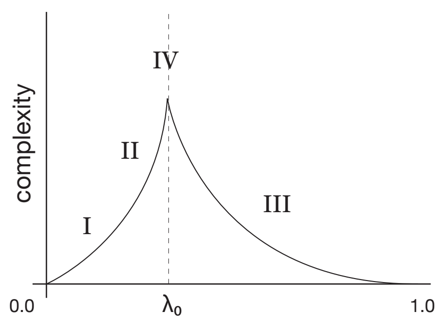
Continuous cellular automata consist another category of CA. Since some of the experiments conducted in this thesis are based on that type of CA, it's worth mentioning briefly their main properties. Here, each cell is not just either black or white, but instead can have any of a continuous range of grey levels. Similar to totalistic CA, the sum of the neighbours is calculated and its average is used to determine the grey level of the cell in the next step. When the average is 'thresholded', the cell inherits its state based on the average of itself and its neighbours in the previous instance unless this average is higher than the threshold, where the cell's state depends on what the rule graph instructs.

Wolfram tried this kind of CA to test whether the complexity found on the simpler CA he initially generated was due to the limitation of the cells to get explicit distinct states. Not only did Wolfram found out that despite the presence of continuous gray levels, complex patterns were still formed, but he showed that in fact considerable complexity is apparent even in extremely simple rules.

Another important contribution to the knowledge we have about Cellular Automata was realised by Chris Langton who did another type of classification introducing the lambda parameter (λ) (see Appendix 2 for definition and formulas). He was trying to find out the conditions under which CA support the basic operations of information transmission, storage, and modification. The idea that CA are capable of supporting universal computation has been known since their invention by Ulam and von Neumann in the late 1940s and later on by Codd, Smith and Conway with his infamous 'Game of Life'. As Langton points out, all of these approaches are based on common fundamental properties of the dynamics found in underlying transitions physics. Briefly, these features describe the capability of storing information locally for long time, transmitting information by signals over large distances, and interaction between the previous two.

> ***"This turns out to be a tractable problem, with a somewhat surprising answer; one which leads directly to a hypothesis about the conditions under which computations might emerge spontaneously in nature."***
> ***(Langton C., 1990)***

After several experiments on CA rules, Langton came up with a correlation between the λ parameter and the dynamical activity in the CA patterns (see Appendix 2). A relation between Wolfram's classification and Langton's λ parameter is illustrated in the following graph.



Langton's λ parameter in correlation to Wolfram's classification

Previous studies on controlling CA for explicit goals

The difficulty of comprehending the unique emergent behaviour of CA and of using them to obtain desired behaviour so far has made it difficult for researchers of various disciplines to use them thoroughly, although a number of applications have been realised in urban growth and human movement simulations, as well as in chemical reactions.

An example of using CA for explicit goals is Wolfram's 'window opacity controlled by CA' study. This is essentially based on the global stability yielded by certain CA rules, known already from Chris Langton and his λ parameter. He developed a window system that can take different overall averages in terms of transparent and opaque glass panels. The idea is based on the characteristic of certain one-dimensional CA to have the same ratio of the different states both in the Initial Configuration and the overall pattern. Therefore, to alter the overall opacity of the window, all one has to do is to change the average in the first row.

Melanie Mitchell and her team from Santa Fe University have realised series of experiments in an endeavour to optimise emergent systems to the extent that they can be used for solving global goals. Mitchell's experimentation on GA driven CA focuses on creating sophisticated parallel computations not with random-looking patterns, but with simpler coordinated patterns that produce desired global results. Even earlier, Norman Packard and his team experimented on evolving CA with GAs and John Koza, the pioneer of genetic programming applied the GP paradigm to process CA from simple random-number generation. Mitchell's attempt was to program a density-classification problem based on a simple binary CA where the state of the majority in the neighbourhood defines the state of the cells in each time step and found that this can only be possible within limited parts of the lattice. Information, being strongly dependent on the small neighbourhood, does not travel long distances, so the outcome patterns consisted of black and white stripes. Therefore, she applied a GA to the above model, where each individual was a binary string of a CA rule out of a vast number of possible CA rules, which were too many for any kind of exhaustive search. The goal was to achieve a density of 1/2; that is of half white and half black cells.

Mitchell's experiment led to concluding that the rules that survived through the 100 runs, did not exhibit coordinated order or information flow but only had local impact on their neighbourhood. In cases where there is some ambiguity between back and white cells, a "signal" is propagated.

> ***"The creation and interactions of these signals can be interpreted as the locus of the computation being performed by the CA - they form its emergent program... Evolving CA with GAs also gives us a tractable framework in which to study the mechanisms by which an evolutionary process might create complex coordinated behaviour in natural decentralised distributed systems"***
>
> ***Mitchell M., 1998***

The CA facade

The importance of resolution is a key issue in this thesis. Aiming to adapt to Nature with emergent systems very similar to the ones we find in Nature itself, it is critical to set the resolution of our observation. Natural systems appear to be smooth and continuous, even though when observed closer they often consist of discrete components and critical transitions between their inherent phases. Sand, for example, seems to have a flowing nature, although it consists of individual grains.

Unlike Mitchell's experiment on global stability and Wolfram's 'controlled opacity window', here the attempt is to achieve specific local averages in the shaded and non-shaded regions of the lattice in a constantly altering environment. In a sense this thesis tries to look into an intermediate scale of several areas of the lattice, which is neither reflected in the mobile, interactive behaviour of local structures like 'gliders', nor in a global stability. How can the structural characteristics of certain CA rules coupled with their overall activity, feed the system to obtain these specific regional optimums maintaining a kinetic, performative outcome? And given this, how could a Neural Network cope with this emergent complexity and use the computational capabilities of such systems to respond to explicit local goals? Finally, to what extent should one be looking into a specific direct mapping between input and a given answer or into a type of local optima, thus bypassing the in-between complexity?

The question of in what level it is more effective to understand and handle the complexity that emerges in Cellular Automata to obtain an adaptive skin is a fundamental motivation to experiment on both Supervised Learning and GA optimisation.

Methodology & Results

Virtual Model

In the previous chapter the importance and uniqueness of CA was highlighted together with methods, which can potentially use the computational behaviour of CA to achieve specific goals. Under this scope, this thesis experiments on how the above can lead to the development of a responsive building façade. Shifting the interest from purely computational experimentations to a specific, real life problem requires a number of assumptions and the creation of a virtual model. These constants are used both to place the façade into a surrounding virtual environment and to define the goals to be met. The south facing façade consists of a lattice of 11 by 31 shading panels. Another building is placed opposite to it, casting different shadows throughout the year. The aim is to train this system to respond to different sun positions during the year, that is different shadows, generating successful CA patterns that acquire specific states' averages in the shaded and non-shaded areas of the façade. A series of light sensors is placed on the rooftop, feeding the system with data and constituting the sole input of the facade. The states of the cells of the grid are correlated to respective number of tilting angles of the panels, resulting to different light intensity in the interior of the building.

A quantitative correlation between tilting angles and light penetration would require specific material constraints and detailed light measurements, which are beyond the scope of this thesis. Therefore a qualitative correlation, rather than a numeric representation is used in the mode of a lookup table that shows the hierarchy of the states in terms of light density. In this table, the most open panel equals to zero and the completely closed to a number equal to the maximum state that the cell can take. In other words, after the states are thresholded thought the table, the lowest the average, the higher the light penetration. So when looking for a CA configuration with the most open panels, we are looking for a total state average around zero. In respect to the required light density to be achieved, a number of simple criteria are predefined. For the shaded areas the criteria is to achieve a specific low average equal to 1/3 of the maximum average of the states of the shaded panels and for the non-shaded panels a formula depending on the sun's azimuth and altitude is used to calculate the required average. For example, in the case of the 4-state continuous CA, the formula below is used to find the optimum average of the shaded panels.

$$optimumAverage = 1 - \frac{abs(azimuth)}{5 * \max(azimuth)}$$

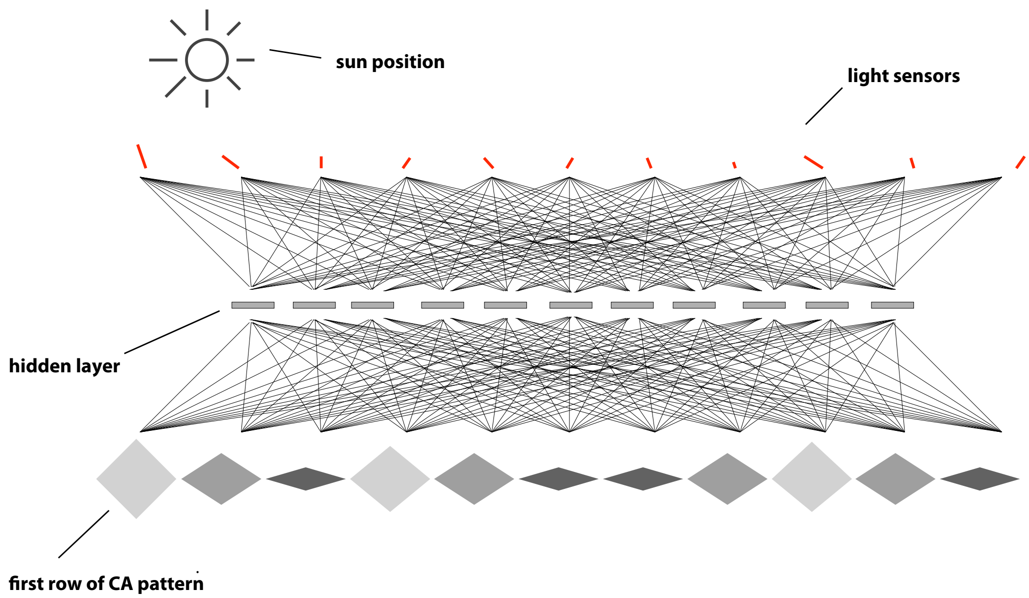
A hard-coded list of 500 different sun positions throughout the year is created with sun's altitude and azimuth – everything ones needs to know to calculate solar path and shadows for a given environment and orientation. In this data, the array of shaded panels is also hard-coded to avoid calculating it in every iteration saving computational power.

Using the value of altitude and azimuth and applying the formula,

$$100 \quad \frac{1}{(1 + sq(abs(azim0 - sensors[p].angleAz)) + sq(abs(alt0 - sensors[p].angleAlt)))}$$

an array of sensor values is generated for each sun position, which is again hard-coded to be accessed later on in the program. Having set the virtual model, calculated the shadows and listed the solar data, now it is time to start experimenting on the learning side of things.

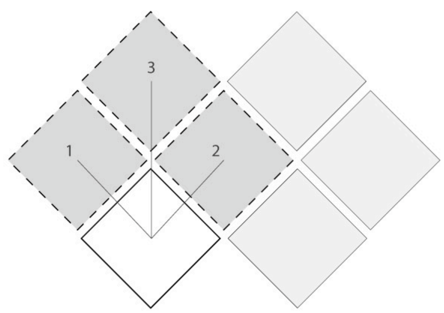
Aiming to create a learning façade emerging from simple CA rules, the choice of setting up an artificial NN primarily supports the idea of avoiding a series of explicit calculations, whilst taking advantage of the inherent computational properties of CA. The idea behind this endeavour is fundamentally based on training the system for a sample training set well enough to empower it to perform successfully in every other case. Therefore, in this case, the attempt is focused on finding an efficient Learning Machine and the appropriate type of CA to reduce the training process increasing its responsiveness against 'novel data'. The NN consists of an input layer of an array of 11 sensor values, the output layer of the 11 panels of the first row, which can take any of the different states of the CA rule, and one hidden layer of neurons, as shown in the image below.

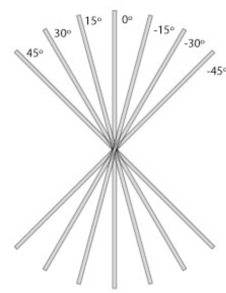


Input, hidden and output layer of neurons

3 different CA rules

Having discussed earlier about how different CA rules exhibit different behaviour and yield various levels of complexity, led to testing 3 different CA rules to understand how the CA type itself affects the performance of backpropagation and GA. To be able to make a fair comparison between the two methods tested, and given the constraints described in the 'virtual model', these 3 specific CA rules are tested for both methods. Starting from the simpler rule and continuing with the more complex ones, the aim is to look into the capabilities of the system to handle different degrees of complexity and structural characteristics. All 3 rules have a neighbourhood consisting of 3 panels, located above, left and right of the current panel. In every case the number of states is correlated to respective number of tilting angles and the rules are totalistic.





Neighbourhood

Tilting angles for 7 states

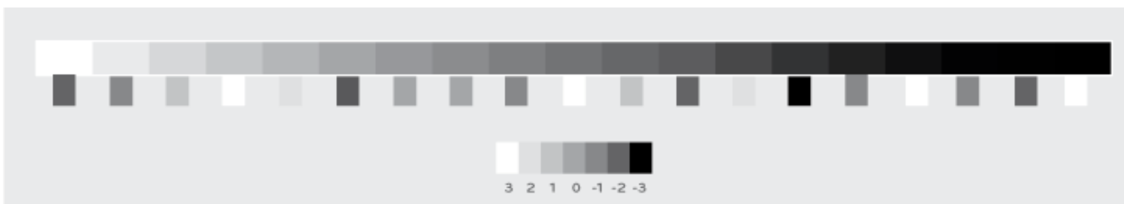
In totalistic CA, the number of possible states, for k different colours, can be calculated with the formula 3*k-2. The first CA type tested here uses a 4-state continuous rule therefore the number of possible states is 10. The states are thresholded as shown in the image below. Lets call it 'rule 1' for referencing it later on.

Rule 1

The second CA type, ‘rule 2’, is driven again by a continuous rule but this time there are 7 possible states, thus 19 possible averages.

Rule 2

The last rule, ‘rule 3’, used in the experiments is an even more complex CA type, yielding more random-like patterns. Similarly to the ‘rule 2’, it has 7 possible states but instead of being continuous, it is discrete.



Rule 3

See Appendix 3 for the patterns generated on the façade from these 3 types of CA.

Training the façade with backpropagation

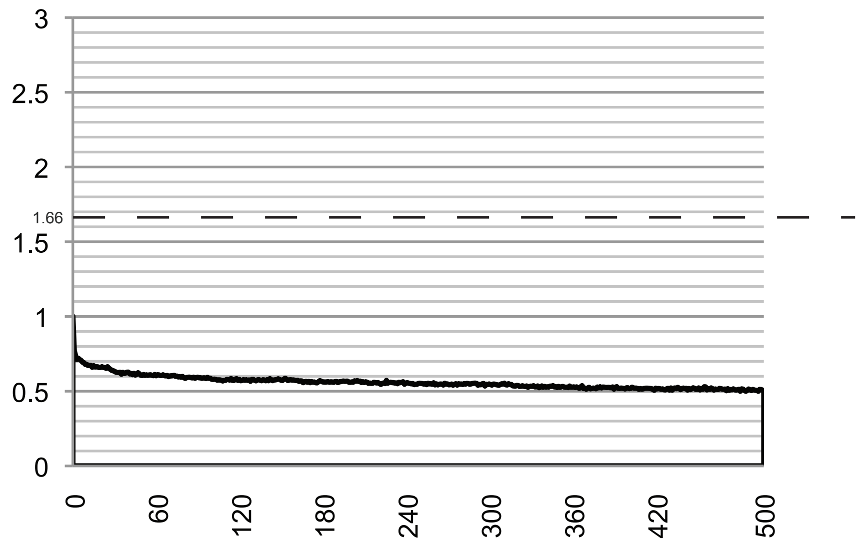
A simple 'feedforward' artificial neural network is a set of connected 'neurons', which can be activated by the input and the connections between them are weighted, usually with real-valued weights. When inputs are activated, the activation spreads in a forward fashion from the input neurons to any hidden layers and finally to the output layer over the weighted connections. Usually, the activation coming into a neuron from other neurons is multiplied by the weights of the connections over which it spreads, and then is added together with other incoming activation. The result is normally thresholded.

In most applications, the network learns a correct mapping between input and output patterns via a learning algorithm. In back-propagation, after each output is produced, the answer is compared to the known right one and the weights between the neurons are adjusted accordingly. Each of these iterations is called 'training cycle', and a complete pass of training cycles through the set of training inputs is called 'training epoch'. This type of procedure is defined as 'Supervised Learning'.

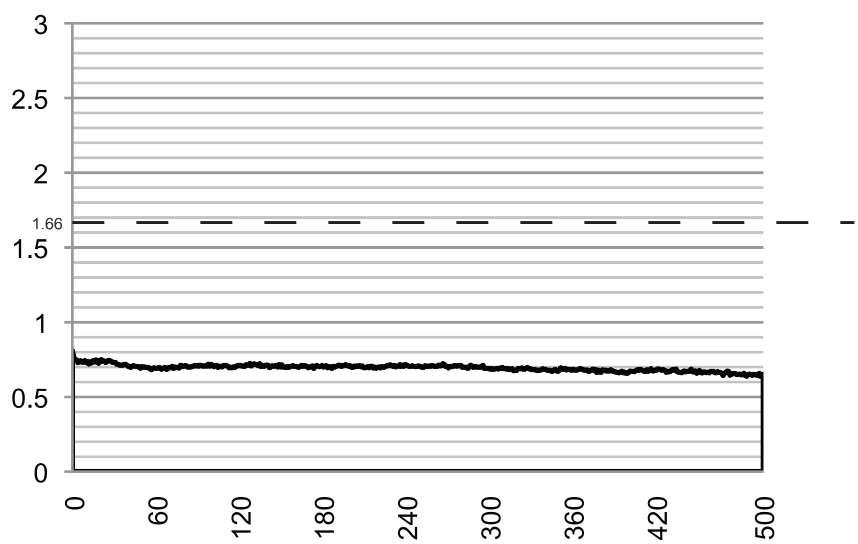
Going back to the on of the main questions of this thesis, can Supervised Learning cope with the complexity of CA, enhancing adaptability to the light levels of the environment? Using a multilayer NN with backpropagation requires that we know the answer for each input. To do so, a list of 1000 random CA patterns of the given rule is generated. Using these and applying the criteria regarding the optimum averages described in the 'virtual model' results to an optimum first row for each sun position. The Initial condition (IC) of the most suitable CA pattern in each case is hard-coded. This, together with the array of light sensor values, constitutes the data set of the NN; that is the input and the desired output. The 1000 random CA configurations are far from an exhaustive list of the possible patterns of a given rule, which would be impossible to list.

The data-set is then split in two parts - training set and testing set - with the first using 2/3 of the data set and the latter using the remaining 1/3. The network is run for 500 learning epochs in every CA case.

In 'rule 1', there appears to be a certain level of learning in the system. The graphs below illustrate the errors for the training and the testing set during the 500 iterations. Using a 4-state CA rule with states from 0 to 3, the maximum error one can get is 3. The error with randomly generated first rows is calculated as the mean of the possible errors for each state, in this case 1,66.



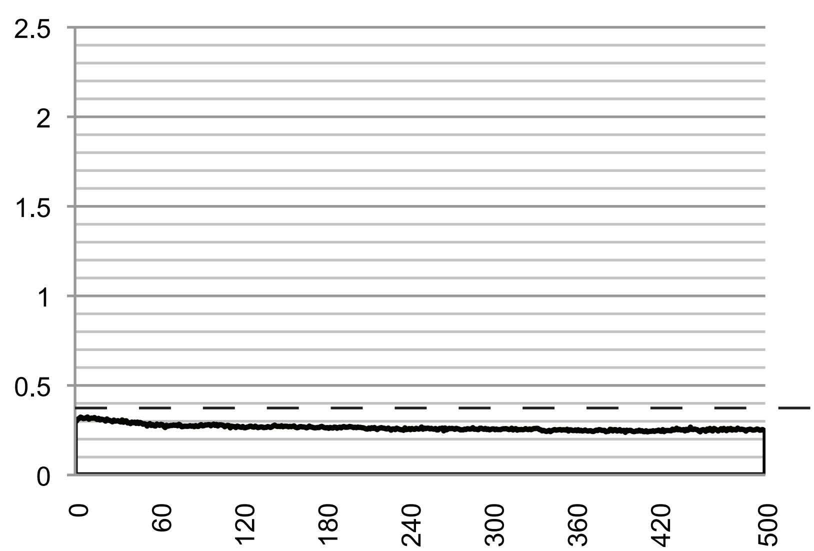
Graph 1. Training set error 4-state continuous CA



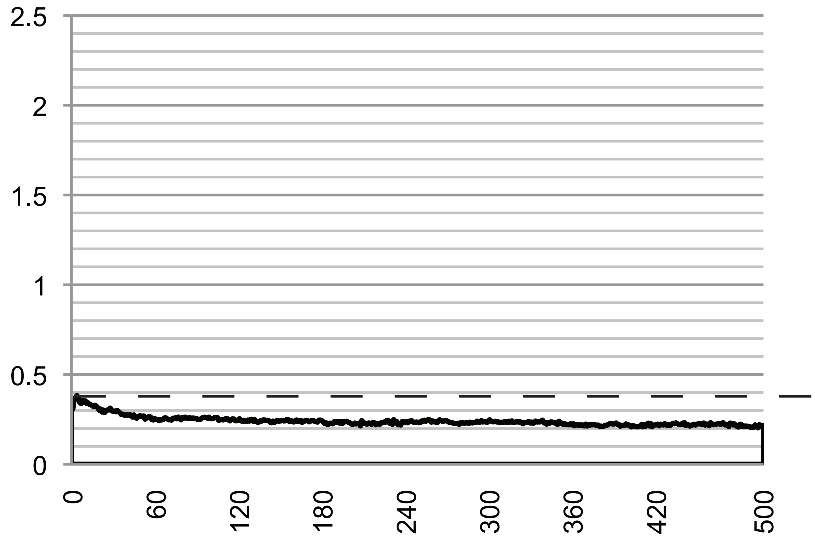
Graph 2. Testing set error 4-state continuous CA

Although in both sets the error is lower than one could get with random initial configurations, not significant learning is achieved. As one would expect, the network is more efficient in the sun positions of the training set because similar sun positions in the testing set do not necessarily require similar first rows of the CA pattern. That is, the system gradually learns what the response should be and adjusts the weights of the connections accordingly, but when it comes to test its performance against new data, it does not perform with the same degree of success. However the difference between the training and testing set is not as significant as in the other CA rules that follow.

Dealing with this kind of complexity, an issue is raised regarding the actual performance of the façade in terms of shading. As described earlier, one tiny change in the IC of the deterministic CA can modify radically the resulting patterns. In other words, although the error of the NN might be low enough to be considered successful, this small error might have dramatic impact on the lattice making it completely unsuccessful in terms of shading. In the same way that a completely different IC can yield a rather efficient configuration, even though the NN error will be high due to the different values in the output. So, to trace this 'actual' performance error, another graph is worth illustrating. That of the overall averages in the shaded and non-shaded areas of the lattice compared to the optimum ones from the predefined set of criteria.

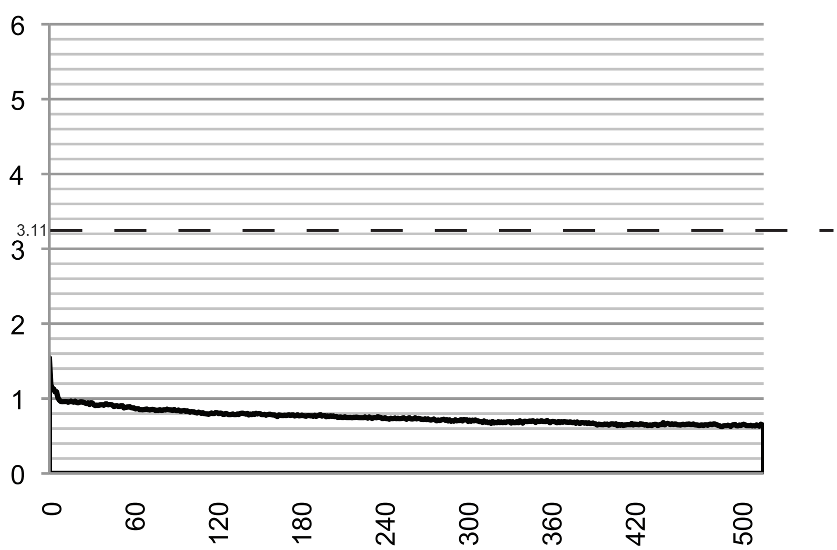


Graph 3. Training set 4-state continuous CA: tracing the shading performance error

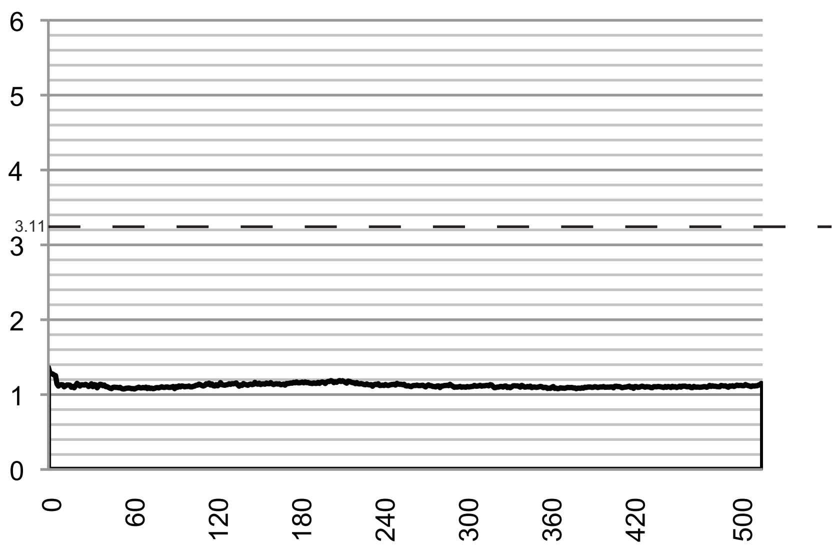


Graph 4. Testing set 4-state continuous CA: tracing the shading performance error

Training and testing set appear to have similar errors in terms of shading performance. The error when using random IC would fluctuate around the values that the graph starts off. The maximum performance error in theory is around 2.5, although when dealing with a CA it is almost impossible to obtain the highest and the lowest averages on the graph.



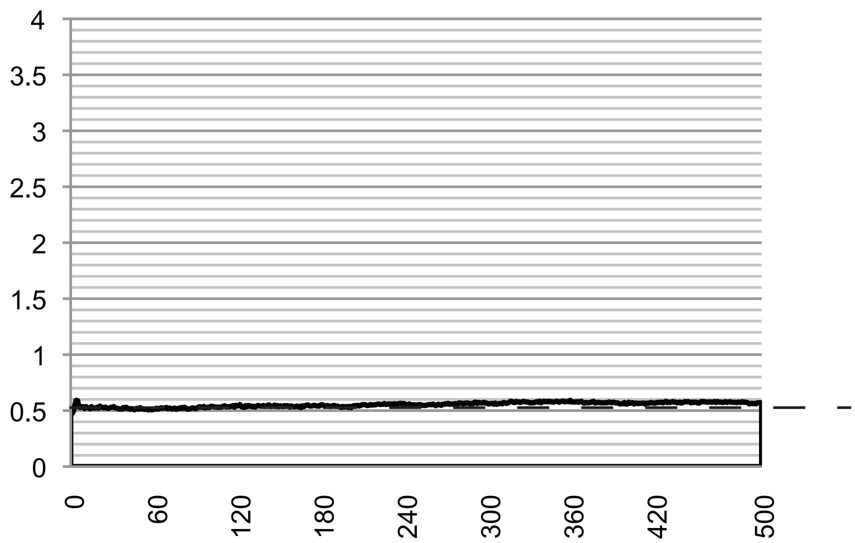
Graph 5. Training set error 7-state continuous CA



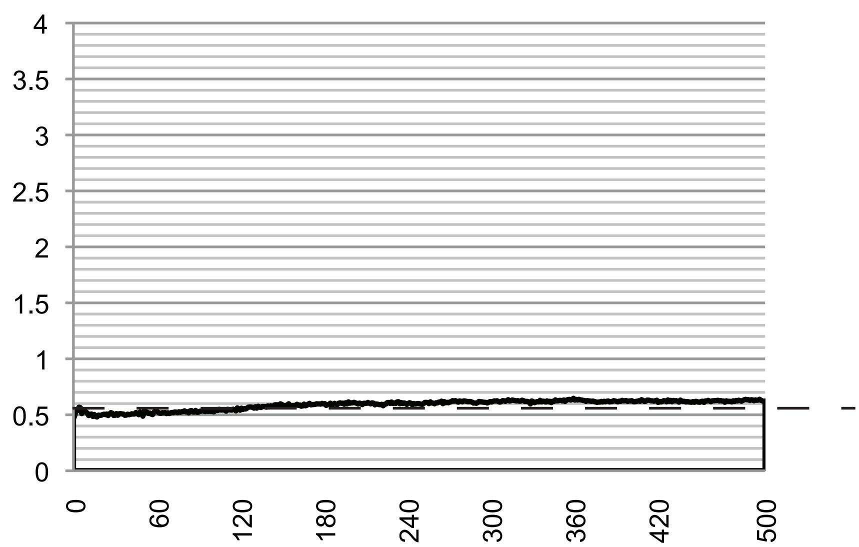
Graph 6. Testing set error 7-state continuous CA

In the case of 'rule 2', shown in the graphs above, again the errors in the training and testing set are lower than with random first rows, but not much improvement is noticeable during the learning epochs. Also, there appears to be a lower performance for the testing set whose error fluctuates around 1.2 and even has the tendency to slightly increase after 2/3 of the iterations.

Regarding the shading overall performance, as can be seen in the following graphs, not only does the error fluctuate around the error when using random IC, but it actually gets worse than that after 1/3 of the iterations.

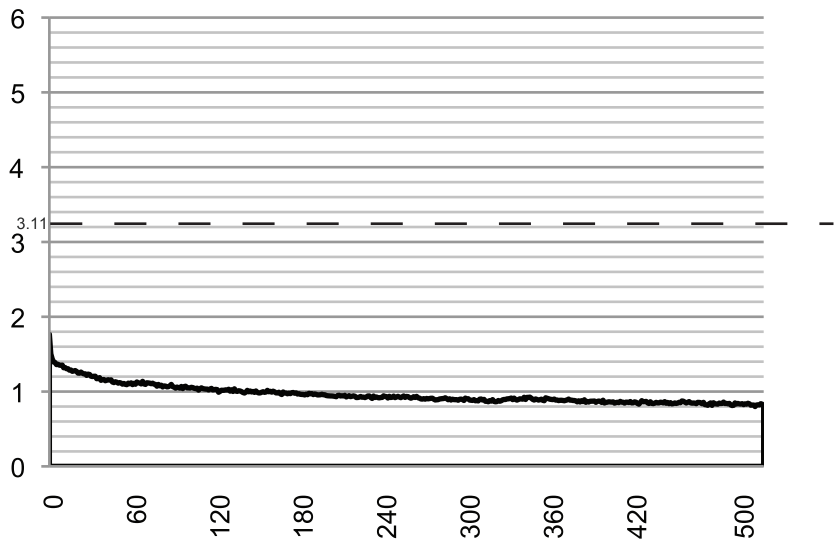


Graph 7. Training set 7-state continuous CA: shading performance error

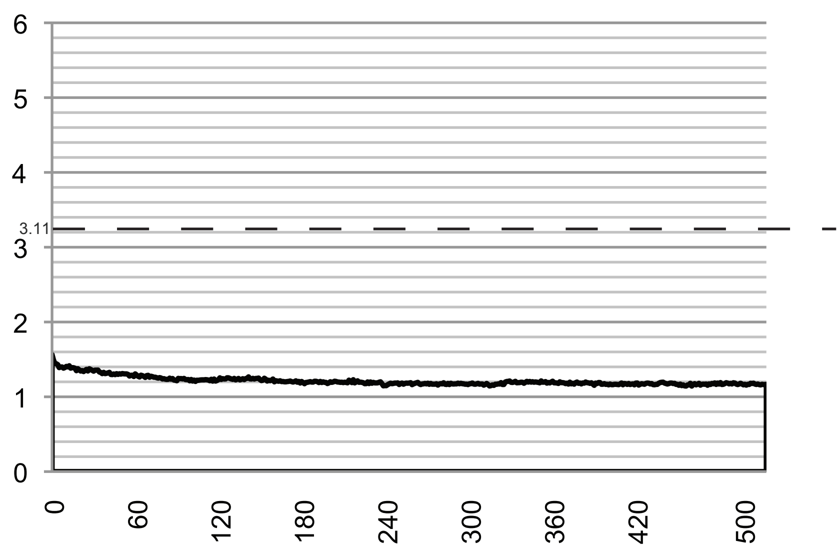


Graph 8. Testing set 7-state continuous CA: shading performance error

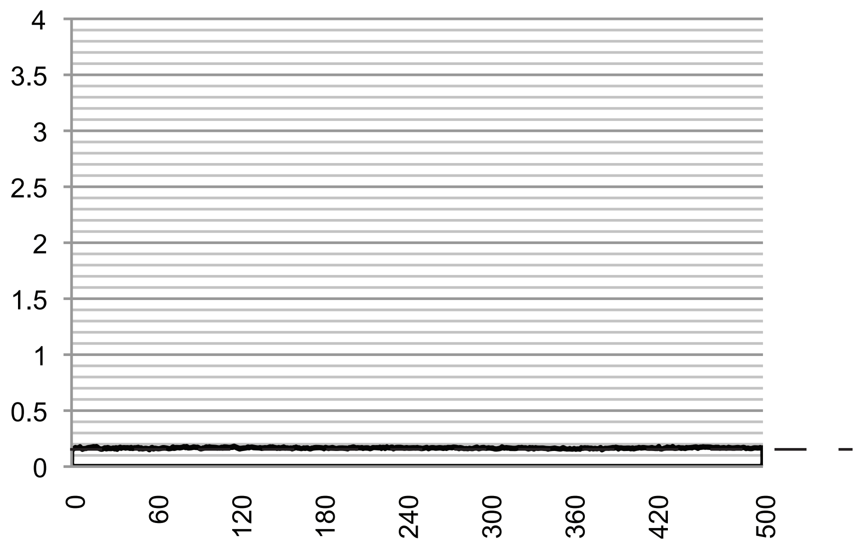
Finally, when the backpropagation was conducted with the 'rule 3', the most complex CA type in these experiments, some interesting results are yielded. First, the network seems to be learning fairly well the optimum first row for both training and testing sets. Second, and more interesting, it appears to start off with a very low error in terms of actual shading performance for both sets, but it retains approximately this value during the whole iteration set.



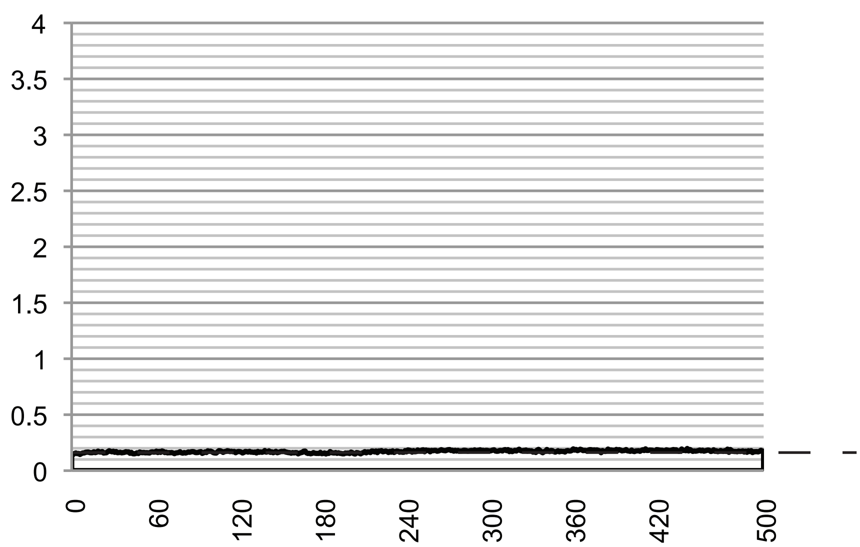
Graph 9. Training set error 7-state discrete CA



Graph 10. Testing set error 7-state discrete CA



Graph 11. Training set 7-state discrete CA: tracing the shading performance error



Graph 12. Testing set 7-state discrete CA: tracing the shading performance error

Optimisation of NN weights with Genetic Algorithm

Aiming to achieve optimum light conditions and to create a 'living skin' that adapts to its actual environment, the interest is set more on the shading efficiency of the façade, rather than the one-to-one comparison in its first row. Given that the outcome of the NN when applying backpropagation was not significantly successful and that it required a predefined answer for each sun position, here the same network is used but this time a GA is applied in order to optimise the weights of the connections between both input - hidden and hidden-output layers. The merit of using a GA lies on the freedom to let the system evolve to obtain low actual performance error and on the fact that there is no limitation in the CA selection, cause there is simply no selection at all. Bypassing the one-to-one comparison of the first row the system can yield any pattern of the given rule, as long as it is efficient in terms of light penetration.

A genetic algorithm can be implemented in many ways in a neural network. It can be used to adjust the weights of the connections in a fixed network or change the network architecture itself by altering the number of neurons and their interconnections or the learning rule. Here, the first way is tested.

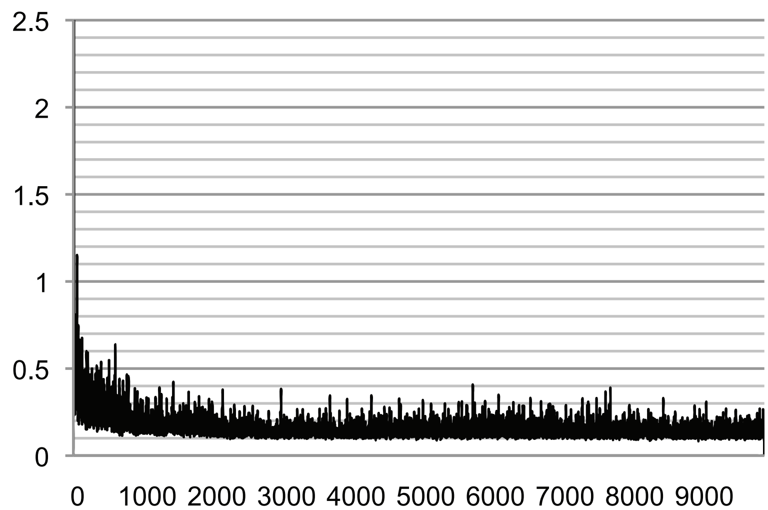
David Montana and Lawrence Davis (1989) realised the first endeavour to evolve the weights of a fixed neural network in a classification problem, by replacing backpropagation with genetic algorithms. Alternative weight-training schemes are often suggested as a way to overcome typical problems appearing in backpropagation like the tendency to get stuck at local optima in weight space, or the absence of a "teacher" to supervise several learning tasks.

In Montana's and Davis' experiment, each chromosome was a set of all the weights in the network consisting of real values rather than bits, as usually. In order to evaluate the fitness of a given chromosome, the weights were assigned to the links of the network and the network was run against the training set. The sum of the errors of all the training cycles was returned and the lower the error, the higher the fitness. A random initial population of 50 connections was generated with weights valued between +1 and -1 and rank selection was used. During their research, Montana and Davis experimented with different types of operations regarding mutation and crossover. The main points are that they added a random value between -1 and +1 to randomly selected non input weights and that the crossover takes two parent weight sets from the offspring and uses randomly selected weights from each parent to create one new child. The performance of the GA was compared to that of using backpropagation through 200 generations for the GA and 5000 complete epochs for the backpropagation. They reasoned that two GA networks were equivalent to one backpropagation iteration, cause the later consists of two parts - forward and backward, and they proved that the GA was outperforming backpropagation in their experiment, obtaining better weight vectors more quickly. Having mentioned this experiment does not prove that GA performs better in every weight optimisation task.

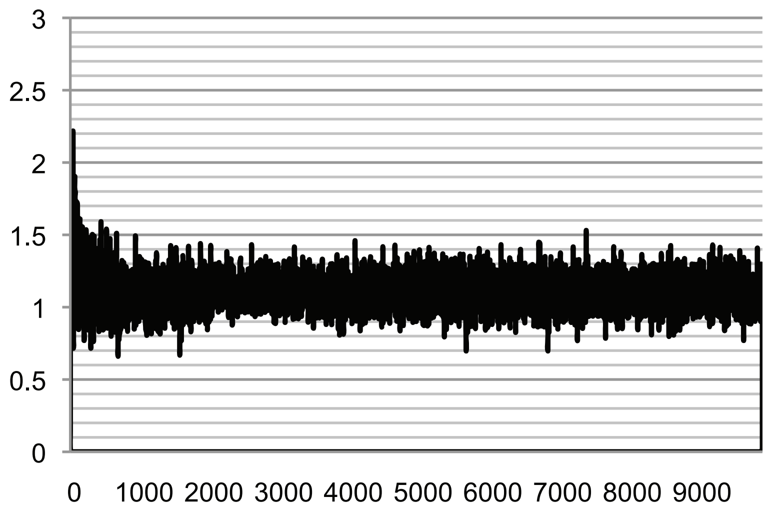
Similarly, to Montana's and Davis', and later Mitchell's (Mitchell M., 1998), optimisation of weights, the structure of the façade's network is fixed and the phenotype of the GA only consists of the two layers of weights between the 3 layers of neurons. In each generation, a population of 50 individuals, that is sets of weights, is produced. All the members of the population are evaluated in terms of shading performance and the lower the shading error, the higher the fitness. They are then ranked according to their fitness, and the operations of crossover and mutation are applied. The first is applied with 50% probability and the latter with 5%. The new parents are selected with a deviation around the best individuals and they in turn generate new offspring. In this case, a limited data-set of 50 sun positions is used to train the network, subset of the set of 800 sun positions in backpropagation.

To make a fair comparison between backpropagation and GA optimisation, GA is applied for two different scenarios. The first uses the shading error to calculate the fitness, regardless of the IC or specific patterns, as described above. However, the GA is tested in the context of a one-to-one mapping for the first row as well, similarly to backpropagation and in both cases the network is tested against the remaining sun positions in the dataset, contrary to common GA optimisation tasks. In an attempt to create a robust façade, training it for a limited sample of sun positions, testing its performance against novel data remains crucial in any case.

Having an additional layer of complexity in the system, that of CA, is GA optimisation proven to be more effective in the case of the façade? As will be illustrated in the graphs below, GA indeed outperforms backpropagation. Not only does it improve the learning capability of the network, but also it directly optimises the shading performance, which constitutes the real goal anyway. In the case of 'rule 1', the shading error drops really quickly and fluctuates between low values during the generations although he error is not significantly lower than that of backpropagation.



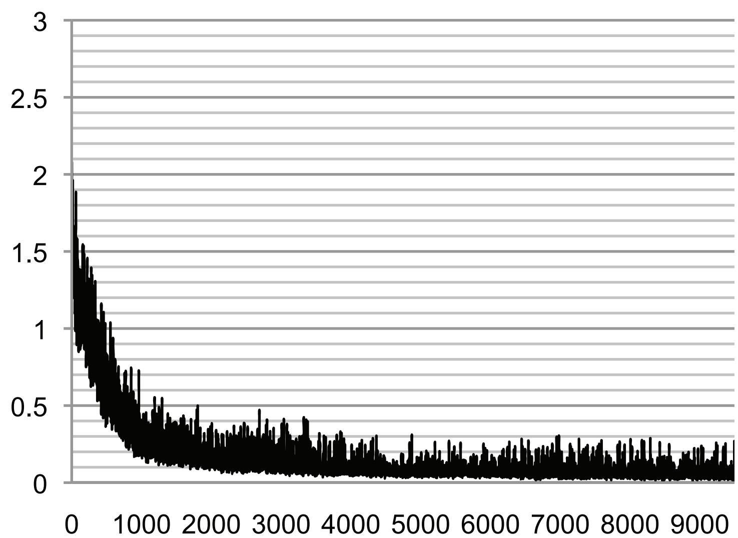
Graph 13. 4-state continuous CA: optimising with shading error for 10000 generations



Graph 14. 4-state continuous CA: tracing the first row error for 10000 generations

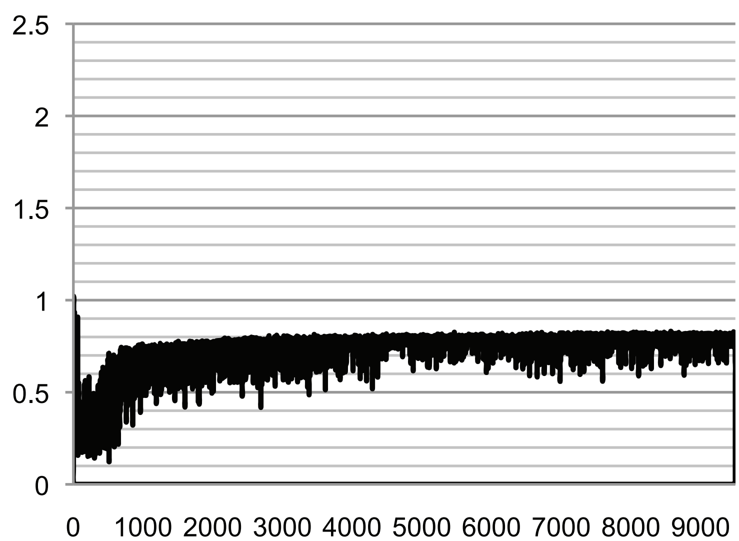
What is interesting here, is to see the correlation between the limited sample of CA used in backpropagation in relation to the unlimited selection of IC in GA. Looking at the graph above, one can see that the error drops until the first 1000 iterations and then ranges between approximately 1.5 and 0.7. As one might expect, there are much better IC for each sun position, in the exhaustive possible first rows that the GA can generate.

But when the fitness of the GA is evaluated only regarding the comparison of the first row of the dataset and that of the NN and not the shading error, then it outperforms significantly backpropagation. In the case of ‘rule 1’, the error drops down to almost 0 after 10000 weight generations. What is even more remarkable is its performance when tested against the rest of the sun positions excluded from the training set. The shading error for this type of testing set, using the optimised weights is as low as 0.1.



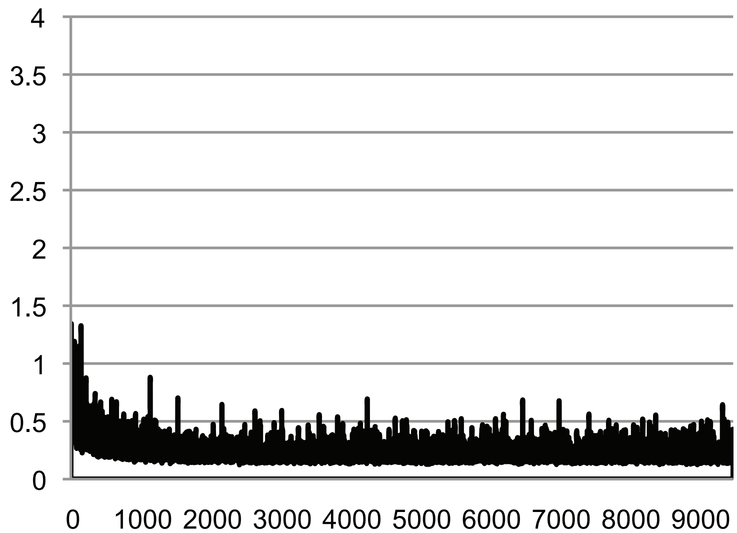
Graph 15. 4-state continuous CA: using the first row error to evaluate the NN, 10000 generations

In this case, when the shading error is calculated, even though it does not feed the system anymore, the network appears to overfit after less than 1000 generations.



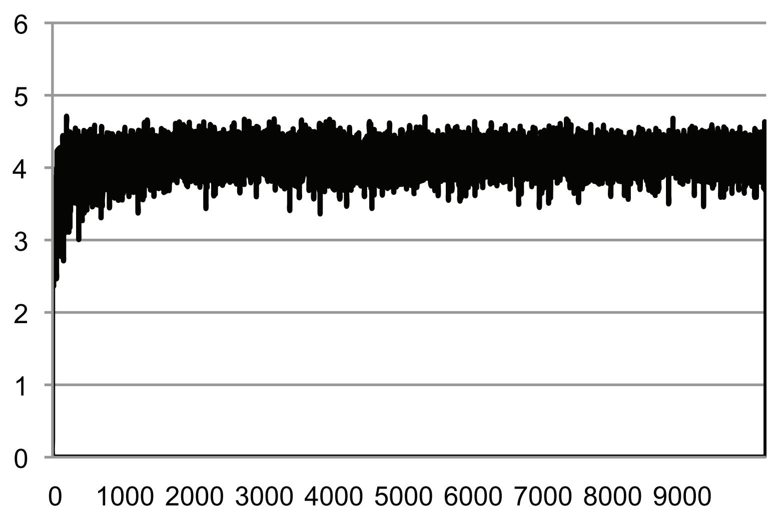
Graph 16. 4-state continuous CA: using the first row error to evaluate the NN, here showing the shading error for 10000 generations

In ‘rule 2’, the 7-state continuous one, the system appears to improve during the optimisation process but less than the previous case and exhibits a higher variation in the values even after 10000 iterations.



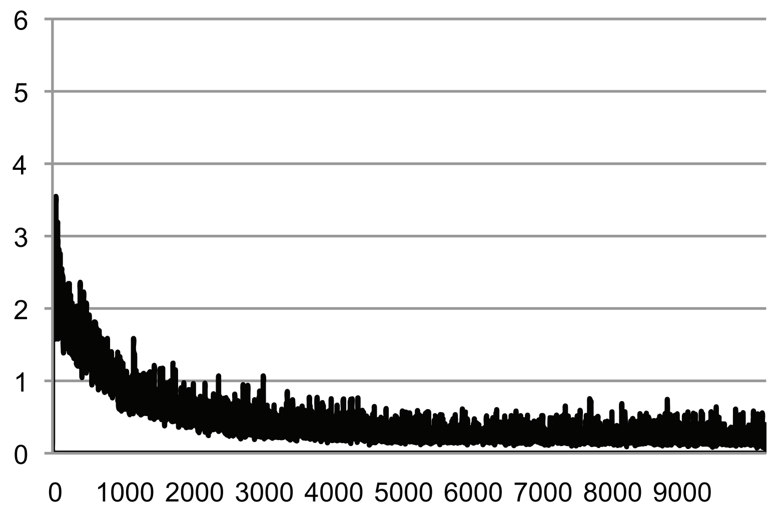
Graph 17. 7-state continuous CA: optimising with the shading error for 10000 generations

What is really interesting here is that when it comes to calculate the error of the one-yo-one comparison of the first row generated by the GA optimised weights to the ones of the initial optimum ones, the error is exceptionally high.

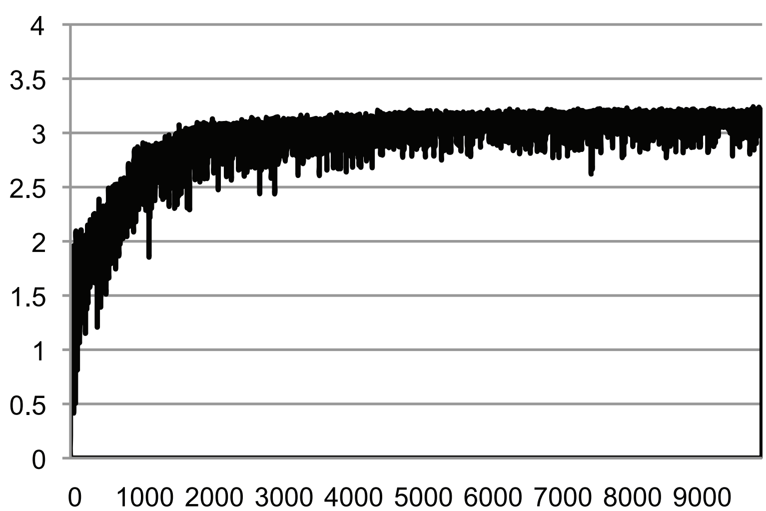


Graph 18. 7-state continuous CA: optimising with the shading error, showing the first row error after 10000 generations

Similarly to 'rule 1', the results produced when optimising the weights with the error of the first row in the case of 'rule 2', although the network learns quickly the IC of the data set, the shading error increases dramatically after few generations, as shown in the graphs below. The 2 graphs are almost inverted. The more the system learns the optimum IC from the selected CA set, the more the actual performance gets worse.

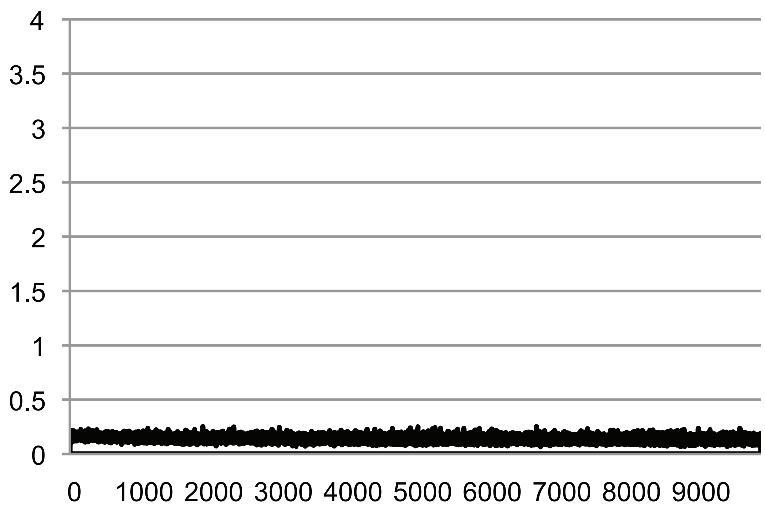


Graph 19. 7-state continuous CA: optimising with the first row error for 10000 generations

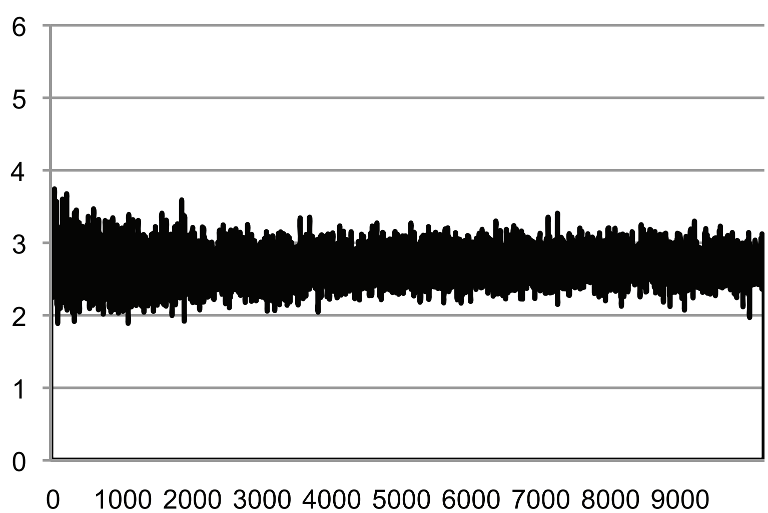


Graph 20. 7-state continuous CA: optimising with the first row error, showing the shading error, during 10000 generations

Finally, for 'rule 3', the shading error is very low from the very first generations. However, almost no change is noticed after that. And as noticed in 'rule 2', the correlation between the IC that produce successful averages in the façade, are far from the ones in the initial set of optimum patterns based on the criteria applied earlier in backpropagation.

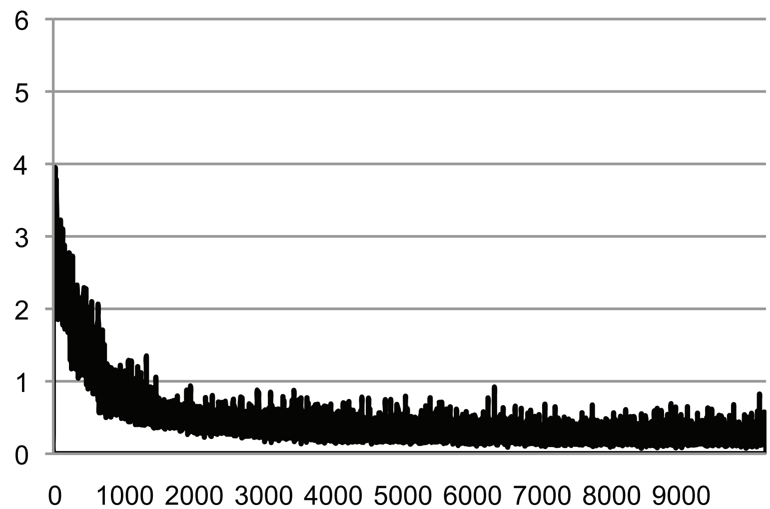


Graph 21. 7-state discontinuous CA: optimising with the shading error for 10000 generations

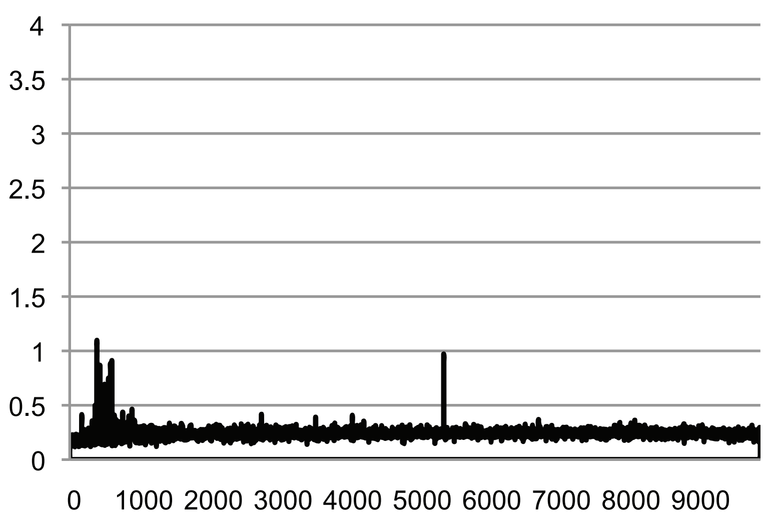


Graph 22. 7-state discontinuous CA: optimising with the shading error for 10000 generations, here showing the first row error

Finally, when using this complex 'rule 3' CA and when optimising the weights to learn the specific first row answers, the efficiency of the network is outstanding, but what is more remarkable is that unlike the two previous rules, the shading performance is very high.



Graph 23. 7-state discontinuous CA: optimising with the first row error for 10000 generations



Graph 24. 7-state discontinuous CA: optimising with the first row error for 10000 generations, here showing the shading performance error

Discussion

In the experiments described earlier, the importance of the structural properties of certain CA rules when seeking for regional optimum averages was highlighted. Trying 3 different CA types for both backpropagation and GA optimisation of weights, aimed to get a clearer idea of the impact of CA's deterministic behaviour to the controllability on the evolving patterns in certain areas of the lattice. Starting from the simple 4-state continuous 'rule 1' and building up to the more complex, discrete, 7-state 'rule 3', revealed and proved some of the intuitions one might have about the degree of complexity such a network can handle but, in the same time, led to controversial results. Creating a system with multiple layers of complexity, allows for different interpretations of the results showed in the graphs earlier. To what extent is the CA itself, the structure of the NN or the type of GA optimisation responsible for any efficient or inefficient performance? And how might a limited dataset have impact on the results?

'Rule 1' appears to be a robust example of CA rule in the context of the experimentations on the local optimums. In both backpropagation and GA optimisation, the relationship between IC and shading performance and training and testing sets seems to be very strong, contrary to more the complex rules tested. GA outperforms backpropagation both when evaluating the fitness with the shading error and the error in the IC when compared to the prerecorded optimums (graphs 1-4 and 13-16).

In 'Rule 2' on the other hand, the two methods lead to very different results. In backpropagation, the network learns the 'first row' answer to some extent, but the shading performance does not seem to improve at all (graphs 5-8). In GA optimisation, the weights are improved dramatically after a few generations and, under this scope, again GA outperforms backpropagation. But when the fitness is evaluated with the specific error in the IC, the shading performance overfits dramatically (graph 20). Dealing with a more complex rule, the controllability on such a system becomes questionable, fact that is even more obvious in 'rule 3' (graphs 9-12 and 21-24).

In backpropagation, an issue is raised about the limited selection of randomly generated CA patterns. How well can a small sample perform compared to an exhaustive list of all the possible configurations of each rule? The experiments showed that although the selection achieves certain required local goals, there appear to exist much more effective patterns outside the limited dataset. This can clearly be seen in the graphs 14, 18 and 22 of GA optimisation shown earlier. In all 3 rules, when the shading performance is optimised and the error drops really low, the one-to-one mapping error of the first row appears to maintain average values during the numerous generations, leading to the conclusion that the GA, being liberated to use any possible first row, chooses IC outside the dataset. This does not imply that the pre-chosen patterns were not efficient at all, since the same criteria regarding the local averages were applied in both cases. However, as one might expect, it is unlikely that the network will predominately use configurations from the list of selected CA cause this constitutes just a minor sample compared to the whole range of possible patterns.

In the same way, it is not surprising that when optimising the weights with the error in the IC compared to the optimum IC calculated for backpropagation the shading error is not low enough to be considered successful. Apart from the restricted dataset, there is another reason why this happens. When finding the optimum CA patterns for each sun position, it is not always the case that the local required averages are accurately equal to the averages of the patterns chosen. Applying the criteria to the 1000 randomly CA patterns, aimed to find the best of these given patterns. However, they do not necessarily generate exactly the optimum averages, they just outperform the rest of the patterns on the list. Keeping these in mind, it can be explained why evaluating the network with just the first row error does not necessarily lead to efficient instances of the façade. This is implied by how the façade performs when tested against the evaluation set.

In GA optimisation, on the other hand, a limited selection of 50 different sun positions is used to evaluate the network. Again, this has a certain impact on the results, but not regardless of the CA type used in each case. And, because of the outstanding performance of the network against the validation set, it does not appear to have negative effect on the learning capacity of the system. For continuous rules, an early overfiting of the shading performance error is observed, when using the first row error to evaluate the fitness, as early as after a few generations (graphs 16, 20). This is partially because of the limited selection of data and of the certain levels of global stability each of the CA rules exhibits. But when the discrete 'rule 3' is tested in the same context, the shading error retains low values throughout the total number of generations, although it shows signs of instability in certain iterations (graph 24). This does not imply in any way that the complex 7-state discontinuous rule is easier for the network to learn. Here, the radically different behaviours and the lack of stability appear to be crucial in this experiment. Yielding patterns with distributed states in the whole lattice enables the façade to obtain optimum local averages in certain configurations.

However, this lies more on its inherent class 4, random-like character rather than to an outstanding achievement of the NN. The opposite happens with rules of high deterministic character. Minor changes in their IC yield completely different patterns, often unsuccessful, although their first row error might be considered notably low. This leads back to the crucial discussion about the level of resolution. Having several layers of complexity in the system suggests various ways of experimentation and does not provide us with a unique answer in the question of what the resolution of observation and control should be. However, after conducting the experiments explained earlier, one can conclude that trying to obtain specific IC is not necessarily advantageous to the direction of enhancing regional optimum averages.

One example altering components of the systems is when using different number of possible states in the CA rule. This seems to alter the learning capability of the network, especially when it comes to evaluating the actual performance of the facade. And this is one characteristic example of how careful one has to be when explaining these results. Cause increasing the possible states, might suggest different CA dynamics in the same way that it adds output neurons leveraging the complexity of the network. Which of the two appears to be more crucial and in a sense more responsible for any lack of performance? Here it is argued that the dynamic activity plays actually more important role than the structure of the NN itself. With the appropriate number of neurons, in theory the network can learn anything. It is the inherent structural characteristics that matter though in the case of the façade. Aiming to train the network for a sample of sun positions, thus shadows, developing the perfect learning scheme for them is not enough. Cause the performance of the façade will eventually be subject to testing it for 'novel' data. To that respect, again GA outperforms backpropagation when tested against the evaluation dataset. As mentioned earlier, this does not constitute the typical application of GA optimisation tasks, but here the evaluation dataset is introduced to both backpropagation and GA after a number of learning epochs and generations respectively to make a fair comparison between the two. When optimising according to the shading error of the lattice, the error of the evaluation set is as low as approximately 0.1 for rules 1 and 2 and 0.2 for the complex 'rule 3', proving that with GA optimisation outperforms backpropagation in the correlation between testing and training set too.

The complexity of the system itself has been previously pointed out. However, evolving a system that exhibits a nested complexity in multiple layers, suggests an iterative trial-and-error process of altering aspects of the system. For example, the structure of the NN is fixed in all of the experiments and the same happens with the 3 specific CA rules. Experimentation on these elements would yield another set of observations seeking for further tests. It is worth mentioning though, that for instance, using different lengths for the hidden layer of the network did not affect the learning process significantly. Similarly, experimenting with numerous different CA rules did not necessarily lead to notable different results from the ones one gets from the experiments on the 3 types of CA showed here. Another experimentation could include an optimisation of the CA rule itself during an optimisation process. Nevertheless, the system of CA, NN and GA optimisation can be considered as a whole, which similarly to CA, cannot be separated from its parts.

Perceiving the whole façade as a unitary entity, suggests that any alteration in its components would affect the resulting behaviour. Moreover, regarding the façade, there are additional layers that go beyond the specific algorithmic methodology regarding the aesthetic value of the resulting façade. The balance between aesthetic and performative criteria and responsiveness to its environment is crucial to that respect, suggesting that both aspects should constitute a coherent pair that feed each other in the context of the building and its surrounding space. Algorithmic approach here is neither used just for the shake of using it nor it is isolated from the natural world. What is proposed in this thesis is a convergence between virtual and real world, or, in other words, an 'eversion' – as Marcos Novak describes this integration.

Conclusions & further thoughts

Starting from a concise idea of creating a performing, adaptive façade, this dissertation was built upon a series of experiments in an endeavour to comprehend how an evolutionary approach could facilitate novel architectural configurations. Using the paradigm of the inherent attributes of Cellular Automata to perform computations and distribute information in space and time, the hypothesis set is twofold; can a building skin be trained to enhance adaptability to its environment coupled with a high level of aesthetic value? Through a flow of experimentations on controlling the complexity of CA with backpropagation and optimisation with Genetic Algorithms, it is proven that the façade can be trained to handle the structural features of Cellular Automata to achieve local optima and constitute an adaptive, kinetic architectural entity. Minimising the series of explicit calculations usually conducted in shading systems and shifting the interest on the behavioural merits of such system, the outcome of the experiments realised here confirm that with the appropriate type of CA and training method the system can then evolve successfully in its surrounding context.

Looking into achieving regional optimum averages, rather than a pure global stability the 3 different CA rules were tested in the same context for both methods. The findings from the various experiments showed that by applying a GA optimisation to the weights the connections between the neurons of a NN of a fixed structure, a high level of learning can be achieved. The considerably high performance against both training and evaluating sets enables the façade to evolve efficiently in its surrounding context, yielding complex patterns. The experiments also suggested that simpler rules, like ‘rule 1’ are easier for the network to learn and manipulate. However, a level of learning is observed even in very complex rules. Finding the balance between complexity and control becomes crucial and subject to architects’ aspirations.

Highlighting the importance of resolution, this thesis tried to tackle the problem of gaining control in specific regions of the lattice of CA patterns by both controlling the initial conditions of the patterns and optimising the desired averages in these regions. To that respect the GA outperformed backpropagation, liberating the façade to evolve freely without being constrained to choose from limited selection of configurations. In a sense, this reflects the idea that emergent systems have the ability to perform computations beyond our current perception. Even more than that, comprehending these intrinsic characteristics empowers architects to realise a programmatic approach that shifts the interest from the final product to the iterative activity of such evolving structures. Contrary to the current perception, it is argued that no compromise between aesthetic merit and pragmatic goals needs to be made.

Following the idea of Cybernetics to create machines enriched with behaviour similar to natural systems and taking the further step of dealing with specific architectural structures, this adaptive façade emphasises the desire for adaptability to the natural environment through a programmatic procedure. Being a part of the natural and built amalgam and consisting of several layers of complexity on its own, it constitutes a fractal of nested systems. Based on the idea that technology and computation can bring closer virtual and real world, this evolutionary approach expresses a thirst to embed computational and evolutionary methods to an architectural discourse that aims to a humanised and playful architecture. This thesis is the outcome of an aspiration to explore the performative and computational characteristics of algorithms found in Nature in the context of an environment, which is not lacking materiality or scale. To that respect, an endeavour to create a physical manifestation of the idea of the façade led to a kinetic installation that expresses in a materialised way the beauty of the performing CA patterns when trying to fulfill inhabitants' optimum light environment. Images of the installation can be found in appendix 4.

Building on the achievements to date, exciting possible scenarios can be explored, including enrichment of the desirable achievements and inputs in various scales of the space-time lattice of Cellular Automata. Time, being discrete in CA, could also become part of this system, adding another layer of complexity to the current system. Potentially, people could also feed the system with their desires and needs; either by manually operating certain inputs on the grid or using a series of sensors that map the light levels of the interior. The notion of the façade trying to accomplish inhabitants' predefined goals could potentially be leveraged to a more direct fashion where people and façade work collaboratively as a unitary entity. Regardless of the level of complexity and the number of constraints one might add to this system, what is remarkable is that the façade will still try to achieve the required goals performing to some extent in an unexpected and unpredictable way and constituting a performing, intelligent organism.

References

Holland J. (1998), Emergence from Chaos to Order, Oxford University Press

Frazer J.H. (1995), An Evolutionary Architecture, Architectural Association, London

Pask G., (1968), A Comment, a Case History and a Plan

Johnson S. (2001), Emergence, the Connected Lives of Ants, Brains, Cities and Software, Scribner, New York

Heylighen F. (1989), Self-organization, Emergence and the Architecture of Complexity

Hensel M., Menges A., Weinstock M. (2004), Emergence: Morphogenetic Design Strategies, AD, issue May/June, London

Chopard B., Droz M. (1998), Cellular Automata Modeling of Physical Systems, Cambridge University Press

Wolfram S. (2002), A New Kind of Science, Wolfram media

Langton C. (1990), Computation at the Edge of Chaos: Phase Transitions and Emergent Computation.

Montana D. J. and Davis L. D (1989), Training feedforward networks using genetic algorithms. In Proceedings of the International Joint Conference on Artificial Intelligence, Morgan Kauffmann.

Moraleza F. J., Crutchfield J. and Mitchell M. (1998), Evolving Two-Dimensional Cellular Automata to Perform Density Classification: A Report on Work in Progress.

Mitchell M. (1998), An Introduction to Genetic Algorithms. Cambridge, Mass.: MIT Press

Mitchell, M. (2009), Complexity: A Guided Tour, Oxford University Press

Mitchell, M. (1997), Computer Science Application: Evolving Cellular Automata to Perform Computation

Mitchell, M. (1996), Evolving Cellular Automata with Genetic Algorithms: a Review of Recent Work

Miller J. H., Page E. S., (2007), Complex Adaptive Systems: An Introduction to Computational Models of Social Life (Princeton Studies in Complexity)

Beer D. R., (1996), Toward the evolution of dynamical neural networks for minimally cognitive behavior

Turner A., First steps in evolution of animats for pedestrian simulation

Appendices

Appendix 1

Brief history of experimentations on Cellular Automata

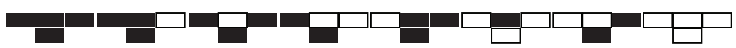
The pioneer of CA is John von Neumann who at the end of 40's as involved in the first digital computers. He tried to imitate the way human brain works and to create mechanisms performing self-control and self-repair. He looked into ways to create self-replicating systems, at the time related to an endeavour to increase computers' efficiency. He addressed this question with a fully discrete universe made up of cells with finite number of information/bits and discrete time steps (29 states). This led to a Cellular space, now known as CA. He claimed that a "Universal computation" could be achieved in this Cellular space simply by setting initial configurations, which would then lead to the solution of any computer algorithm. The notion of universal computation through simple rules in cellular lattices proves wrong the common idea that in order to create complexity, a complex program is necessary.

In 1970, mathematician John Conway introduced Game of Life, which although binary, was interesting due to the complex, interacting structures that emerged. The most known are the so-called 'gliders', a particular arrangement of cells which have the property to move across space, along straight trajectories. And what is more interesting is that the Game of Life is capable of computational universality. The universality of CA made several authors claim that the physical world itself could be considered as a large CA.

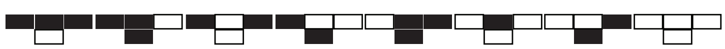
In the early 1980s, Stephen Wolfram started investigating how simple programs evolve and in particular Cellular Automata. He suggested that CA exhibit similar behaviours found in continuous systems, thus could be investigated on mathematical models allowing for explicit calculations due to their 'boolean' nature.

In his 'A New Kind of Science' (Wolfram S., 2002), Wolfram approaches complexity as a fundamental independent phenomenon and looks into its possible origins. He argues that complexity is not necessarily product of complex initial conditions, but that of rather simple ones. What is characteristic in CA is that one is not able to predict how they will evolve after discrete time steps and the only way to know is to run the CA for these steps and see what happens in their visual representation. Cause there is no computation quicker than computation itself. Following consistent research and experimentation on numerous CA, he came up with an extensive list of CA rules known as 'Wolfram rules' and classified them based on the complexity they exhibit.

Starting with very simple rules of binary CA with one cell in the initial configuration, like the uniform rule 250 below:



or the repetitive rule 90 below:

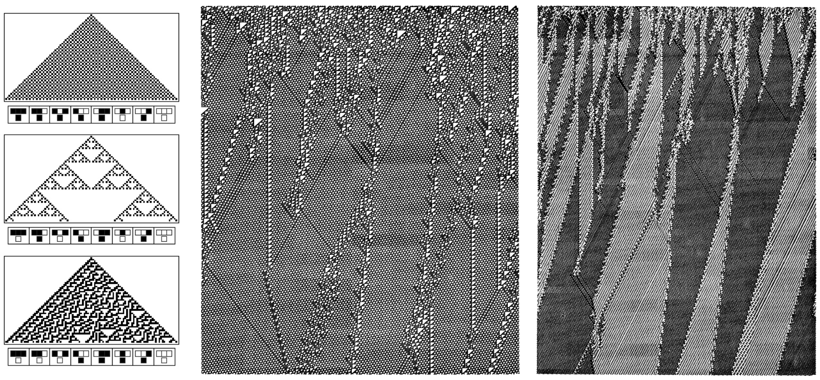


and then proceeding to the next step of slightly modifying the rule, reveals the nested, intricate, patterns one can get. Although, in these cases the patterns yielded are still highly regular, rule 30 produces a pattern that exhibits high irregularity and complexity. Wolfram points out that none of the tests he has ever done on this rule show any meaningful deviation at all from perfect randomness.

> ***"Over and over again we will see the same kind of thing: that even though the underlying rules for a system are simple, and even though the system is started from simple initial conditions, the behavior that the system shows can nevertheless be highly complex. And I will argue that it is this basic phenomenon that is ultimately responsible for most of the complexity that we see in nature."***
> ***Wolfram S. (2002)***

There are cellular automata whose behavior is in effect still more complex and in which even brief averages of the states become very difficult to predict, like rule 110. As

Wolfram describes, the pattern obtained with this rule shows a remarkable mixture of regularity and irregularity. And as the system progresses, a variety of definite localised structures are produced. And although each of these structures works in a fairly simple way and they evolved from very simple initial conditions (IC), their various interactions can have very complicated effects.



CA patterns | Stephen Wolfram

Wolfram's CA classification

Class I

Pattern evolves to homogenous state, in other words to limit points, regardless of its initial configuration. Wolfram makes an interesting parallelism with pendulums where no matter how quickly one swings them, they will eventually stay hang straight down.

Class II

CA evolves to simple separated periodic structures or to limit cycles. These patterns are characterised by repetitive behaviour and absence of long-range communication.

Class III

Class III patterns yield chaotic aperiodic patterns. Apparent randomness is generated which is not always the result of random initial conditions and high sensitivity on them.

Class IV

Class IV CA patterns yield complex patterns of localized structures, certain number of which persist forever and the pattern is characterised by a combination of randomness and order. Another general feature is that with appropriate IC they can mimic the behaviour of all sorts of other systems, as Wolfram points out.

Appendix 2

Langton's lambda parameter and CA behaviour

For a given set of possible transition functions *D* of a CA with *K* states and *N* neighbours, we pick an arbitrary state $s \in \Sigma$ and call it the quiescent state s_q. Let there be *n* transitions to this special quiescent state in a transition function Δ. The remaining $K^N - n$ transitions in *Δ* are filled by picking randomly and uniformly over the other *K -1* states in *Σ - sq*. Then

$$\lambda = \frac{K^N - n}{K^N}$$

If $n = K^N$, then all of the transitions in the rule table will be to the quiescent state *sq* and

$$\lambda = 0.0$$

If $n = 0$, then there will be no transitions to *sq* and

$$\lambda = 1.0$$

When all states are represented equally in the rule table, then

$$\lambda = 1.0 - \frac{1}{k}$$

$\lambda \approx 0.00$	All dynamical activity dies out after a single time step, leaving the arrays uniform in state S1. The area of dynamical activity has collapsed.
$\lambda \approx 0.05$	The dynamics reaches the uniform s1, fixed point after approximately 2 time steps.
$\lambda \approx 0.10$	The homogeneous fixed point is reached after 3 or 4 time steps.
$\lambda \approx 0.15$	The homogeneous fixed point is reached after 4 or 4 time steps.
$\lambda \approx 0.20$	The dynamics reaches a periodic structure, which will persist forever. Transients have increased to 7 f o 10 time steps as well. Note that the evolution does not necessarily lead to periodic dynamics.
$\lambda \approx 0.25$	Structures of period 1 appear. Thus, there are now three different possible outcomes for the ultimate dynamics of the system, depending on the initial state. The dynamics may reach a ***homogeneous*** fixed point consisting entirely of state s1, or it may reach a ***heterogeneous*** fixed point, consisting mostly of cells in state s1, with a sprinkling of cells stuck in one of the other states, or it may settle down to periodic behavior. Notice that the transients have lengthened even more.
$\lambda \approx 0.30$	Transients have lengthened again.

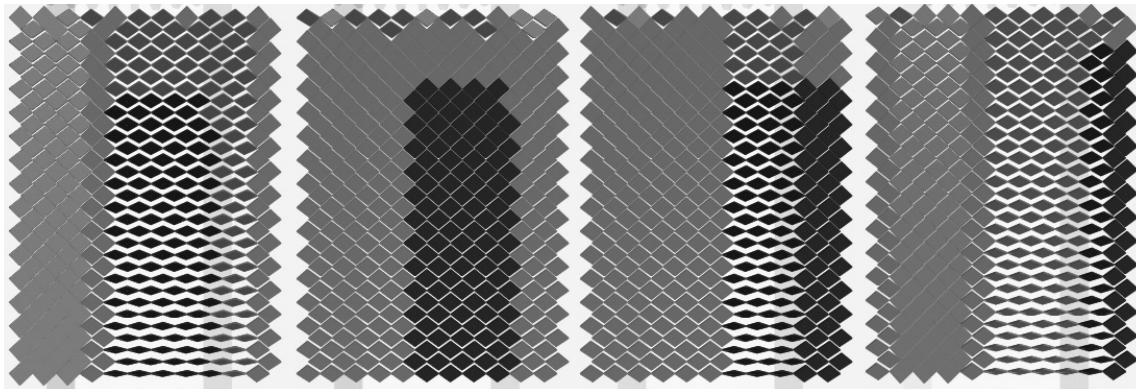
$\lambda \approx 0.35$	Transient length has grown significantly, and a new kind of periodic structure with a longer period has appeared. Most of the previous structures are still possible hence the spectrum of dynamical possibilities is broadening.
$\lambda \approx 0.40$	Transient length has increased to about 60 time steps, and a structure has appeared with a period of about 40 time steps. The area of dynamical activity is still collapsing down onto isolated periodic configurations.
$\lambda \approx 0.45$	Transient length has increased to almost 1,000 time steps. Here, the structure on the right appears to be periodic, with a period of about 100 time steps. However, after viewing several cycles of its period, it is apparent that the whole structure is moving to the left, and so this pattern will not recur precisely in its same position until it has cycled at least once around the array. Furthermore, as it propagates to the left, this structure eventually annihilates a period 1 structure after about 800 time steps. Thus, the transient length before a periodic structure is reached has grown enormously. It turns out that even after one orbit around the array, the periodic structure does not return exactly to its previous position. It must orbit the array 3 times before it repeats itself exactly. As it has shifted over only 3 sites after its quasi-period of 116 time steps, the true period of this structure is 14,848 time steps, Here, the area of dynamical activity is at a balance point between collapse and expansion.
$\lambda \approx 0.50$	Typical transient length is on the order of 12.000 time steps. After the transient, the dynamical activity settles down to periodic behavior, possibly of period one as shown in the figure. Although, the dynamics eventually becomes simple, the transient time has increased dramatically. The general tendency now is that the area of dynamical activity ***expands*** rather than contracts with time. There are, however, large fluctuations in the area covered by dynamical activity, and it is these fluctuations, which lead to the eventual collapse of the dynamics.
$\lambda \approx 0.55$	We have entered a new dynamical regime in which the transients have become so long that - for all practical purposes - they am the steady state behavior of the system over any period of time for which we can observe them. Whereas before, the dynamics ***eventually*** settled down to periodic behavior, we are now in a regime in which the dynamics typically settles down to effectively ***chaotic*** behavior.
$\lambda \approx 0.60$	The dynamics are quite chaotic, and the transient length to “typical” chaotic behavior has decreased significantly, The area of dynamical activity expands more rapidly with time.
$\lambda \approx 0.65$	Typical chaotic behavior is achieved in only 10 time-steps or so. The area of dynamical activity is expanding at about one cell per time-step in each direction, approximately half of the maximum possible rate for this neighborhood template.
$\lambda \approx 0.70$	Fully developed chaotic behaviour is reached in only 2 time-steps. The **area** of dynamical activity is expanding. Even more rapidly.
$\lambda \approx 0.75$	After only **a** single time-step, the array is essentially random and remains so thereafter. The area of dynamical activity spreads at the maximum possible rate.

(Langton C., 1990)

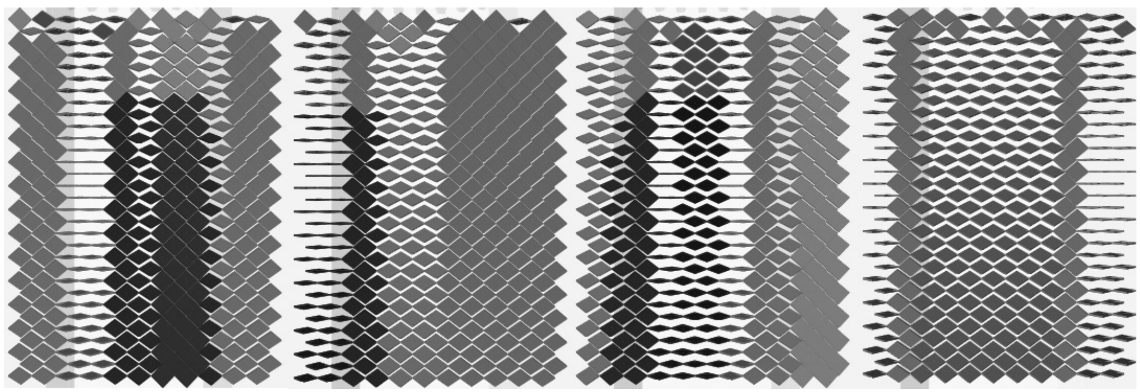
Langton came to a number of interesting conclusions regarding λ parameter and its relation to the computational behaviour of the CA configurations. His first conclusion was in respect to the relationship between the range of transmitted information (transient length) in CA lattices and the size of the lattice. For low λ values, there is no strong dependence on the size of the array, whilst for λ = 0.45 or higher the transient length appears to increase exponentially as the size of array increases. Beyond λ = 0.5 though, although the dynamics is now settling down to effectively chaotic behavior instead of periodic behavior, the transient lengths are getting shorter with increasing λ, rather than longer. For λ = 0.75 the CA patterns exhibit maximum dynamical disorder. The transition range supports both static and propagating structures, like 'gliders' or 'blinkers' in Conway's Game of Life. With the first used as 'signals' and the later as 'storage' one can conclude again that this type of emergent system could be used for universal computing. Finally, one important conclusion in Langton's research is that there appears to be a critical limit, a fine line, between synthesis and degeneration.

> ***"…we find that there exist an upper limit as well as a lower limit on the 'complexity" of a system if the process of synthesis is to be non-degenerative, constructive, or open ended. We also find that these upper and lower bounds seem to be fairly close together and are located in the vicinity of a phase-transition. As the systems near the phase-transition exhibit a range of behaviors which reflects the phenomenology of computations surprisingly well, we suggest that we can locate computation within the spectrum of dynamical behaviors at a phase-transition here at the 'edge of chaos'."***
>
> ***(Langton C., 1990)***

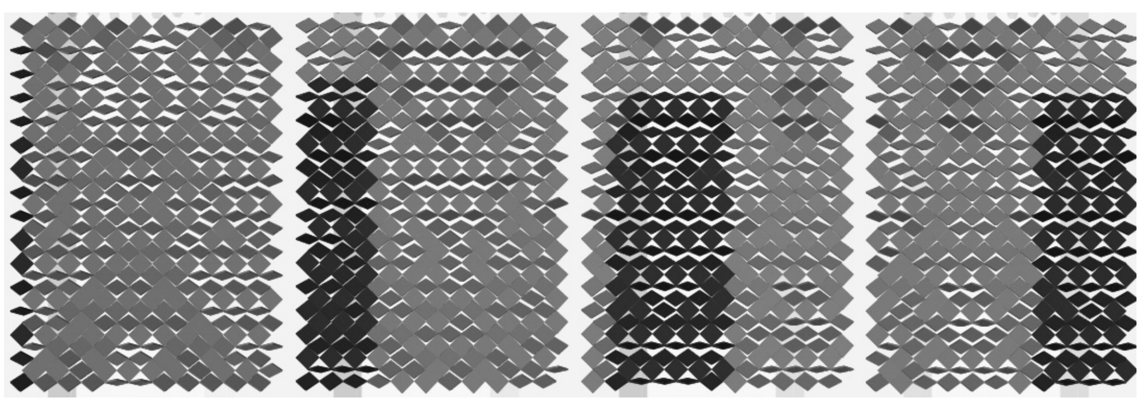
Appendix 3



4-state continuous CA implemented on the façade | Screenshots of the program running in Processing 1.0

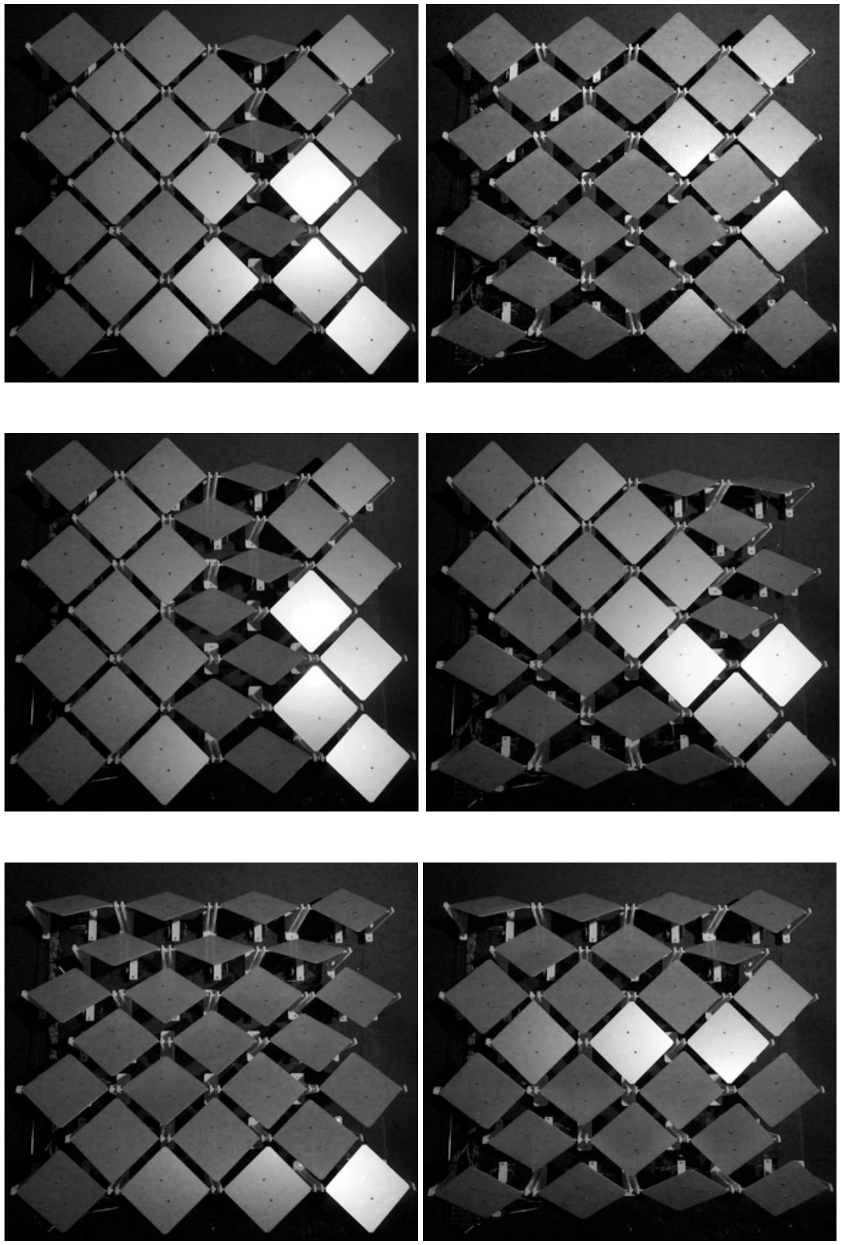


7-state continuous CA implemented on the façade | Screenshots of the program running in Processing 1.0



7-state discrete CA implemented on the façade | Screenshots of the program running in Processing 1.0

Appendix 4



The physical kinetic model