Division of Informatics, University of Edinburgh

Institute of Perception, Action and Behaviour

The Control of Dynamical Systems by Evolved Constraints: A New Perspective on Modelling Life

by

Tim Taylor

Informatics Research Report EDI-INF-RR-0148

Division of Informatics **September 2002**
http://www.informatics.ed.ac.uk/

The Control of Dynamical Systems by Evolved Constraints: A New Perspective on Modelling Life

Tim Taylor

Informatics Research Report EDI-INF-RR-0148

DIVISION *of* INFORMATICS
Institute of Perception, Action and Behaviour

September 2002

To be presented as a poster at the Artificial Life 8 conference, Sydney, Australia, December 2002.

Abstract :

It is argued that a fruitful, and as yet unexplored, avenue for artificial life research lies in modelling organisms as organisations embedded within a dynamical system environment. From this perspective, the origin and evolution of life is the progressive control of the dynamical system at a local level by constraints which are represented on an organism's genome. Such an approach shifts the focus of artificial life models away from the design of individuals, towards the interaction of an individual with its dynamic environment. It also admits no representational distinction between organism and environment. An evolutionary cellular automata system, called EvoCA, is introduced as a tool to explore these ideas. In EvoCA, an evolved individual is a collection of constraints on the state of specific cells in the CA. Results are presented of initial experiments to investigate the interaction of evolution with the dynamics of EvoCA under various regimes (as characterised by Langton's lambda parameter) and to study different ways of specifying constraints (i.e. timed and conditional genes). It is suggested that, for future experiments, it may be productive to allow evolution more opportunity to exploit the given dynamics of the environment, by using natural selection methods, rather than trying to force it in a particular direction using the artificial selection methods of genetic algorithms. A variety of planned future experiments are discussed.

Keywords : , artificial life; origin of life; open-ended evolution; epistemology

The Control of Dynamical Systems by Evolved Constraints

A New Perspective on Modelling Life

Tim Taylor

IPAB, Division of Informatics, University of Edinburgh, 5 Forrest Hill, Edinburgh, EH1 2QL, U.K.
Email: tim.taylor@ed.ac.uk Web: http://www.dai.ed.ac.uk/homes/timt/

Abstract

It is argued that a fruitful, and as yet unexplored, avenue for artificial life research lies in modelling organisms as organisations embedded within a dynamical system environment. From this perspective, the origin and evolution of life is the progressive control of the dynamical system at a local level by constraints which are represented on an organism's genome. Such an approach shifts the focus of artificial life models away from the design of individuals, towards the *interaction* of an individual with its dynamic environment. It also admits no representational distinction between organism and environment. An evolutionary cellular automata system, called EvoCA, is introduced as a tool to explore these ideas. In EvoCA, an evolved individual is a collection of constraints on the state of specific cells in the CA. Results are presented of initial experiments to investigate the interaction of evolution with the dynamics of EvoCA under various regimes (as characterised by Langton's λ parameter) and to study different ways of specifying constraints (i.e. timed and conditional genes). It is suggested that, for future experiments, it may be productive to allow evolution more opportunity to exploit the given dynamics of the environment, by using natural selection methods, rather than trying to force it in a particular direction using the artificial selection methods of genetic algorithms. A variety of planned future experiments are discussed.

Introduction

One of the key challenges facing artificial life researchers, as well as biologists, is to explain the origin of living organisms from a non-living environment (Bedau *et al.* 2000; Maynard Smith 1986).

Most ALife work on the evolution of life has employed a strong representational distinction between living organisms and their environment. Examples include Tierra (Ray 1991) and PolyWorld (Yaeger 1994). In Tierra, for instance, individuals are computer programs with associated instruction pointers, registers, stacks, etc. Interactions between an individual and its environment can only be achieved in a limited number of predefined ways, such as by the allocation of memory in order to reproduce (an interaction with the abiotic environment), or by reading machine instructions from a neighbouring program (an interaction with the biotic environment).[1]

Even in work where no such distinction exists, individuals, and the dynamical laws of the environment, are carefully crafted to achieve a particular type of behaviour. Examples of this type include von Neumann's self-reproducing automata (von Neumann 1966), simulations of autopoietic systems (Varela, Maturana, & Uribe 1974; McMullin & Varela 1997), and Holland's α-Universes (Holland 1976).

Neither of these approaches (i.e. using a strong representational distinction between living and non-living entities, or the careful crafting of the "laws of physics" of the world for a particular purpose) can inform us a great deal of how living organisms first originated from a non-living environment which, presumably, was not specifically designed to support life.

Howard Pattee, a physicist by training, has devoted much of his career to the question of the origin of life (Rocha 2001). His particular perspective is the issue of how semiotics (i.e. symbol systems, such as genomes, and their associated semantics in the context of an organism) can originate from a purely physical environment.

Pattee argues that the distinction between the material and symbolic aspects of living organisms, seen as an example of the more general epistemological distinction between laws and initial conditions, is a defining feature of life, and also a necessary condition for open-ended evolution (Pattee 1995a; 1995b). He explains the relationship between the two as follows:

> Writing symbols is a time-dependent dynamic activity that leaves time-independent structure or record. ... Symbols are read when these structures re-enter the dynamics of laws as constraints. Any highly evolved formal symbol system may be viewed as a particularly versatile collection of initial conditions or constraints, often stored in a memory, pro-

[1] Further discussion of issues relating to the representation of individuals and environments in artificial life models can be found in (Taylor 2001).

> ducing significant or functional behavior that is usefully described by locally selected rules rather than physical laws. ... [A]ll symbol systems must have material embodiments that obey physical laws. But for the reasons just stated, the lawful description of symbols, even though correct in all details, can reveal no significance. (Pattee 1995b)

The symbols recorded on the genome ultimately acquire semantics in an organism in the context of the survival value of the dynamics that they initiate (i.e. natural selection of phenotypes). It is this autonomous structure-function self-referent organisation that is entailed in Pattee's term "semantic closure".

This perspective, then, sees organisms as entities whose phenotypes are embedded within an environment viewed as a dynamical system, and whose genotypes interact with the environment by specifying constraints[2] upon its dynamics, thereby generating the phenotypes. From this point of view, the most important distinction is not between organisms and their abiotic environment, but rather between the environment as a whole (including organism phenotypes) and organism genotypes. It is the relatively time-independent genotypes, by supplying local constraints to the dynamics of the environment, that reify phenotypes as distinct entities within the environment.

Furthermore, with no representational distinction between organisms and the environment, any property of the environment could in principle become incorporated or used by an organism's phenotype. Any property or process so incorporated can be expected to be retained if it promotes the evolutionary success of the organism. From this perspective, the evolutionary acquisition of new ways of measuring the environment (e.g. new genetic machinery or new sensory capabilities) is not the problem that it is sometimes considered to be in other approaches to artificial life (Pattee 1995b; Dautenhahn, Polani, & Uthmann 2001).

The EvoCA System

A simulation platform, called EvoCA, was designed to explore this dynamical systems view of organisms and environments.[3]

EvoCA is built upon a cellular automaton (CA) system. CA were chosen because they are fairly simple, discrete time and space dynamical systems, whose behaviour has been extensively studied (Burks 1968; Wolfram 1986).

One-, two- and three-dimensional CA are allowed, although the geometry of the cells is restricted to a square in the 2D case and a cube in the 3D case. Any number of states may be specified, and neighbourhood specification is flexible.

A uniform transition function is used for all cells in the CA, and updates are performed synchronously. The transition function is loaded in from a file that specifies a list of mappings between neighbourhood configurations and new states. Any neighbourhood configuration that is not listed in the file will map to the quiescent state (which is designated as state 0). The quiescent state is defined such that if all of a cell's neighbours are in the quiescent state, then the state of the cell does not change.

On top of the CA, EvoCA provides a representation for genotypes, methods for decoding genotypes such that they interact with the dynamics of the CA, and a genetic algorithm for evolving populations of genotypes.[4]

Genotypes

Every genotype in EvoCA is associated with a specific cell on the CA. A genotype comprises a variable length list of genes. Two types of gene are available: timed and conditional. Both types specify a particular target cell (whose position is defined relative to that of the genotype) and a target state for that cell. A maximum radius is defined for each dimension of the CA to confine the position of the target cell relative to the genotype.

Each gene additionally specifies a precondition that must be satisfied in order for it to activated. Timed and conditional genes have different types of preconditions.

Timed genes specify a time (i.e. a specific iteration of the CA) at which they act. At the specified iteration, the gene sets the state of the target cell to the target state.

Conditional genes specify a watch cell and watch state. The watch cell specification is confined to the set of cells that are direct neighbours of the target cell. Whenever the specified watch cell is in the specified watch state, the conditional gene is triggered, setting the state of the target cell to the target state.

Every gene in the genotype is checked at each iteration of the CA to see whether it should be activated for that iteration. Whenever any gene is activated, its action overrides the normal CA transition function for the

[2] Throughout this paper the general term 'constraint' is used to cover initial conditions, constraints and boundary conditions. For further discussion of these concepts, and of their relationship to physical laws, the reader is referred to (Pattee 1995a).

[3] The source code for EvoCA can be downloaded from the author's webpage. Note, however, that the system runs on top of a third-party CA platform (CAMELot), produced by the Edinburgh Parallel Computing Centre, `http://www.epcc.ed.ac.uk`.

[4] Various authors have experimented with systems in which a genetic algorithm is used to evolve the transition function of a CA to achieve a particular task, e.g. (Crutchfield & Mitchell 1995). This is fundamentally different to the current approach of evolving constraints for a given transition function; this other work, from the current perspective, entails evolving the "laws of physics" of the environment rather than constraints to control a given set of laws.

target cell for that particular iteration.

The Genetic Algorithm

A fairly standard, generational genetic algorithm is used to evolve a population of individuals. Each individual is evaluated separately, and placed in the same cell at the centre of the CA array. The iteration count of the CA is reset to zero at the start of each evaluation. All cells are initially set to the quiescent state, except those which have non-quiescent states specified by timed genes acting at time zero. The CA is then allowed to run for a given number of iterations, with the genes of the genotype setting specific cell states when they become active.

The fitness function of the genetic algorithm uses a target configuration of the CA which is loaded into the system from a file at startup. The target specifies the desired state of each cell in the CA. As well as specific states, two types of wildcards may also be used: '?' means that the cell can take on any state; 'x' means that it can take on any non-quiescent state.

Cells with a target state of '?' do not participate in the fitness calculation. All other cells contribute as follows: if the current state of the cell matches the target state, the cell contributes one point to the fitness score, otherwise it contributes nothing. The overall fitness of a configuration is then the total score divided by the number of cells contributing to the calculation, and scaled to a given range (0–100 in the experiments reported here).

An individual's fitness is calculated at each iteration, excluding time zero (the initial configuration). The maximum fitness achieved at any point during the evaluation is taken as that individual's final fitness score.

To generate the initial population of P individuals, a set of P_i individuals (where $P_i \geq P$) is randomly generated and evaluated. The fittest P individuals from this set are selected to fill the initial population.

The genetic algorithm can be set up to use either tournament selection or fitness proportional (roulette wheel) selection. For tournament selection, a probability may be specified of the fittest member of the tournament being selected – in cases where the fittest member is not selected, an individual is selected from the tournament group using fitness proportional selection. Elitism may also be applied, in which case the fittest individual from the population is guaranteed to pass at least one exact copy of itself into the next generation.

In addition to one-point crossover and gene mutation, a number of other genetic operators are also available: gene insertion (a random gene is inserted into an existing genotype); gene deletion (an existing gene is deleted from a genotype); gene reversal (the order of a sequence of genes between two selected points in the genotype is reversed); and gene duplication (a sequence of genes between two selected points in the genotype is duplicated at the end of the genotype). A limit on the maximum allowable genome length is defined. This is respected both during the initial generation of random genotypes and in the action of the genetic operators.

In EvoCA, the perspective of a genome as a source of constraints for a dynamical system is taken to the extreme; genomes play *no* part in the dynamics of the system other than to specify constraints (i.e. they have no material embodiment). This is largely for practical, rather than theoretical, reasons, and means that the design of the system can be kept very simple. This simplification is not without consequences. It means that an external mechanism is required for interpreting genomes as constraints (this happens at each iteration of the CA), and for writing genomes, with noise, at reproduction (this is performed by the genetic algorithm). Another consequence is that the symbols (genes) on the genome are restricted to specifying constraints in a predefined way – in the particular design of EvoCA they are defined to map to the lowest level of the CA dynamics by constraining a specific cell to be in a specific state at a specific time. These restrictions all arise because genomes in EvoCA do not participate in the dynamics of the system at all, except through supplying constraints. This design decision is justified because of the perspective of genomes taken here – i.e. that the *fundamental* role of the genome is to supply constraints to the dynamical environment.

Langton's λ Parameter

It has been observed that the dynamics of CA can be categorised into a small number of qualitative classes. The most widely known scheme, suggested by Wolfram (1984), consists of four classes:

- Class I CA evolve, from almost all initial states, to a unique homogeneous state.
- Class II CA evolve to simple separated periodic structures.
- Class III CA evolve to chaotic aperiodic patterns.
- Class IV CA evolve complex patterns of localised structures. Wolfram conjectured that such CA are capable of universal computation.

A method of parameterising the space of CA transition functions has been suggested by Langton (1986,1991). The parameter λ is defined as follows:

$$\lambda = \frac{K^N - n_q}{K^N} \tag{1}$$

where K is the number of cell states, N is the size of the cell neighbourhood, and n_q is the total number of transitions to the quiescent state out of the K^N entries

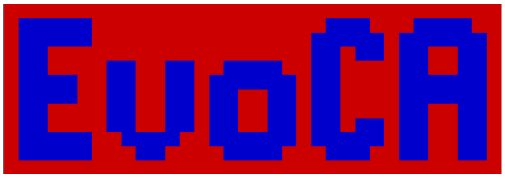

Figure 1: Target configuration. Black cells represent the quiescent state. Gray cells represent the 'x' wildcard (any state except the quiescent state).

in the transition table. A given λ value therefore defines a (typically large) set of possible transition functions for a given CA.

In general, the dynamics that emerge from any particular transition function and initial configuration cannot be accurately predicted. However, Langton observed a clear progression in the typical behaviour arising from a number of transition functions taken from the set defined by a given λ value, as λ was increased from 0.0 to 1.0. In terms of Wolfram's classes, the progression was Class I $\rightarrow$ Class II $\rightarrow$ Class IV $\rightarrow$ Class III.

The precise values of λ at which a CA moves from one class of dynamics to another depends on factors such as the dimensionality of the CA, the size of the array, the number of states and the number of neighbours. For a 2D CA with $K = 8$ and $N = 5$ (and array size 64×64), Langton found that Class I behaviour is found in the range $0.0 < \lambda < 0.2$, Class II followed by Class IV in the range $0.2 < \lambda < 0.4$, and Class III in the range $0.4 < \lambda < 1.0$ (Langton 1986).

Experiments

The experiments reported here constitute an initial investigation of using evolution to control the CA dynamics. They were also designed to study the relative utility of timed and conditional genes.

In all reported experiments, the parameters of the system were as listed in Table 1 (except where otherwise stated), and the target configuration was as shown in Figure 1. Furthermore, all timed genes were restricted to act at time zero; that is, timed genes can only affect the initial configuration of the CA.[5]

Note that the size of the array is 34×12, or 408 cells in total. The maximum genome length is set to 50, which means that a genome can, at best, directly affect the state of 12.25% of the cells. To achieve high fitness scores, a genome must therefore rely to some degree on the dynamics of the CA.

Note, however, that for some transition functions, any given target configuration may be impossible to achieve exactly (i.e. it may be a "Garden of Eden" configuration that can only exist if supplied as the initial configuration). Furthermore, for CA with Class IV dynamics, transients become indefinitely long (Wolfram 1984; Langton 1991), implying that the question of whether a particular configuration is, or is not, a Garden of Eden configuration is undecidable. This means that perfect fitness scores should not be expected in these experiments, with the particular target configuration used, or, in general, with any target configuration.

[5]This was because it was considered undesirable, both from a theoretical and an implementational point of view, to use genes which required access to a global clock.

Number of states K	8
Neighbourhood size N	5
Number of dimensions	2
Array size (X,Y)	34,12
Population size P	100
Initial population size P_i	1000
Maximum genome length	50
Evaluation duration	50
Number of generations	1000
Apply Elitism?	Yes
Tournament Selection?	Yes
Tournament Size	4
Fittest wins probability	0.75
Crossover probability	0.6
Mutation probability	0.1
Gene insertion probability	0.05
Gene deletion probability	0.05
Gene reversal probability	0.05
Gene duplication probability	0.05
Genome position (X,Y)	17,6
Radius of gene action (X,Y)	17,6

Table 1: Default parameter values. (Mutation probability specifies a per gene probability. All other probabilities relating to genetic operators specify per genome probabilities.)

A utility program was written to generate random transition function files for any given λ value. A series of 500 evolutionary runs was conducted, in the following four batches:

1. For each value of λ in the set {0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80}, 20 evolutionary runs were conducted, each using a different randomly generated transition function, to test variability across transition functions and across λ values.

2. For $\lambda = 0.80$, 20 runs were conducted, all using the same transition function, to test variability due to the stochastic nature of the genetic algorithm.

3. For each value of λ in the set {0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80}, 20 evolutionary runs were conducted, each using a different randomly generated transition function. In these experiments, all genomes were restricted to contain timed genes only; no conditional genes were allowed.

4. For each value of λ in the set {0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80}, 20 evolutionary runs were conducted, each using a different randomly generated transition

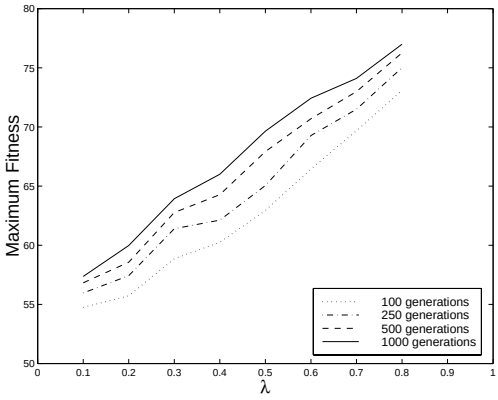

Figure 3: Performance by λ value. Ordinate is the mean value of the maximum fitness in the population, across 20 different transition functions, at various times in the evolutionary run for batch 1 runs.

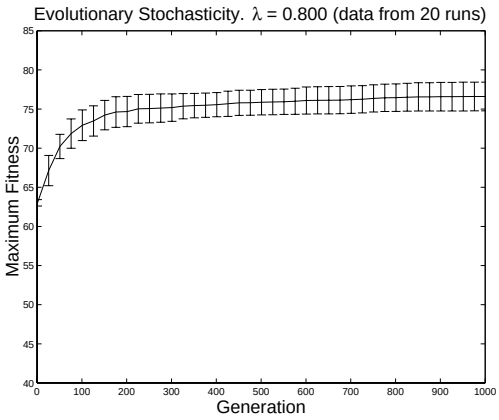

Figure 4: Evolutionary stochasticity. Graph shows the mean value of the maximum fitness in the population at each generation, and the sample standard deviation, for batch 2 runs.

function. In these experiments, the maximum genome length was raised from 50 to 200. This means that a genome can directly affect the state of 49% of the cells.

Results

The performance of the first batch of experiments is shown in Figure 2. Each graph in this figure shows that the maximum fitness increases over evolutionary time. However, in most runs, the rate of increase of fitness after 500 generations is comparatively low. The standard deviation of the fitness across the different transition functions is typically in the range 2–4.

Looking at the difference in results as λ is increased from 0.10 to 0.80, it can be seen that the performance steadily improves. This can be clearly seen in Figure 3, which shows that the improvement is roughly linear with respect to λ.

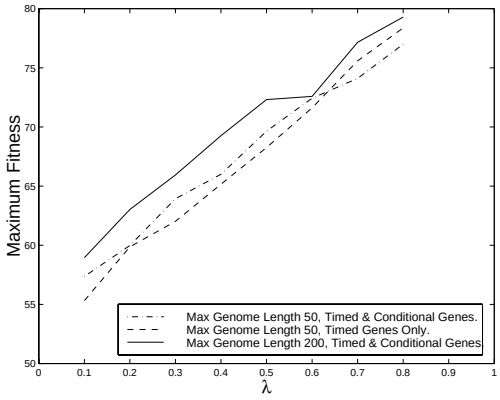

Figure 5: Performance by λ value. Ordinate is the mean value of the end-of-run maximum fitness in the population, across 20 different transition functions, for runs in batches 1, 3 and 4.

The results of the second batch of runs, which looked at the variability in performance due to the stochasticity of the genetic algorithm (for $\lambda = 0.80$), are shown in Figure 4. Comparing this graph with that in the bottom right of Figure 2, it appears that the level of variability in performance introduced by evolutionary stochasticity is comparable to that introduced by the use of different transition functions for a given λ value.

In the experiments where only timed genes were allowed (batch 3), the performance graphs (not shown) were generally very similar to those obtained in batch 1. The performance graphs (not shown) of the experiments where the maximum genome length was raised from 50 to 200 genes (batch 4) showed surprisingly little improvement over the batch 1 runs. The mean end-of-run performance for runs in each of these three batches (i.e. batches 1, 3 and 4), plotted against λ, is shown in Figure 5.

Considering all of the runs conducted, individuals generally achieved their maximum fitness at an early iteration of the CA. For runs in batches 1, 3 and 4, the iteration at which the best individual in the final generation achieved its maximum fitness is plotted in Figure 6. This graph shows that, for batch 1 runs (timed and conditional genes, maximum genome length 50), as λ rises, the median iteration at which maximum fitness is achieved decreases. However, for runs in batches 3 and 4 (timed genes only, and longer genomes, respectively), maximum fitness is generally achieved in fewer than 5 iterations for all λ values.

The relative abundance of timed genes and conditional genes in the fittest end-of-run individual, for runs in batch 1, is plotted in Figure 7. This shows that conditional genes tend to outnumber timed genes for $0.2 \leq \lambda \leq 0.4$, whereas for other λ values the relative abundances are roughly equal. The corresponding chart

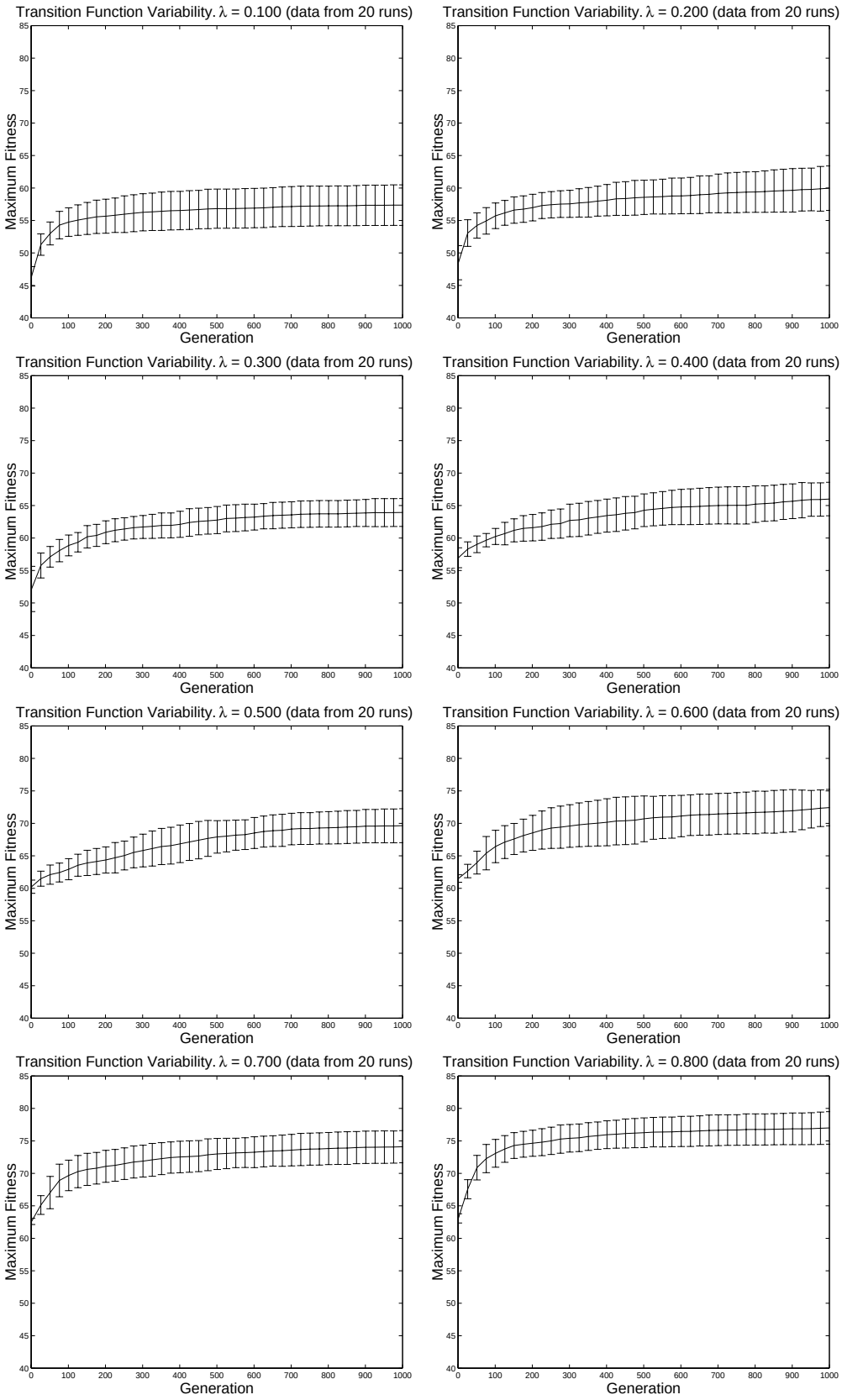

Figure 2: Performance for different transition functions, for a variety of different λ values. Graphs show the mean value of the maximum fitness in the population at each generation, together with the sample standard deviation.

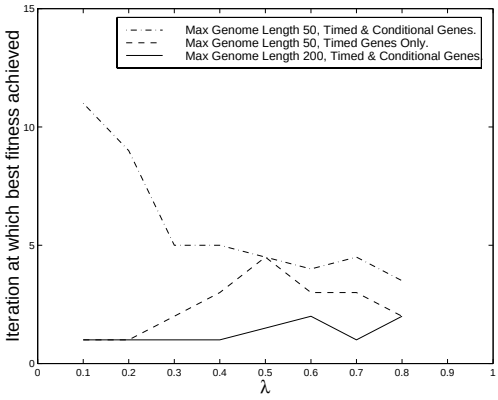

Figure 6: Iteration at which best end-of-run individual achieved maximum fitness, by λ. Medians plotted for runs in batches 1, 3 and 4.

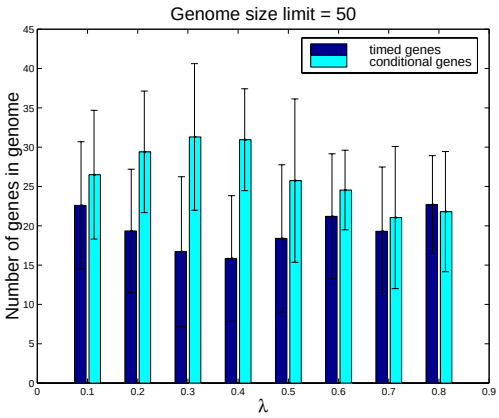

Figure 7: Relative abundance of timed genes versus conditional genes in best end-of-run individual, by λ. Means and standard deviations plotted for runs in batch 1.

for batch 4 runs, with longer genomes, is shown in Figure 8. In this case, the relative abundance of timed and conditional genes is roughly equal for all λ values.

The phenotype of the best individual found across all of the runs is shown in Figure 9. This individual achieved a maximum fitness of 84.8% at iteration 1. Compare the lower section of this figure with the target configuration shown in Figure 1 (remember that the gray cells in this figure represent the 'x' wildcard, meaning that any non-quiescent state is acceptable). The letters in the target configuration can just about be distinguished in the lower section of Figure 9, although the result is certainly not perfect.

Figure 10 shows the phenotype of another individual with good fitness, but this time where maximum fitness was achieved after iteration 1. This particular individual achieves a maximum fitness of 80.1 at iteration 3.

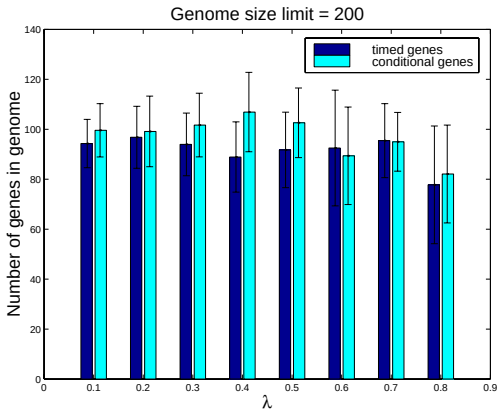

Figure 8: Relative abundance of timed genes versus conditional genes in best end-of-run individual, by λ. Means and standard deviations plotted for runs in batch 4.

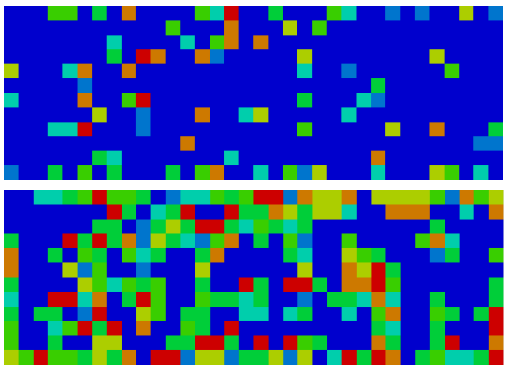

Figure 9: Phenotype of the best individual found across all runs. This was from a batch 4 run, where the maximum genome length was 200, $\lambda = 0.80$. Initial configuration is shown at the top, with that at iteration 1 below. Fitness at iteration 1 is 84.8.

Discussion

This paper has presented the results of initial investigations into the behaviour of EcoCA. Further investigations are required to build up a more thorough understanding of the system. Experiments currently under way include:

- A more thorough investigation of the role of evolutionary stochasticity, over a range of λ values.
- An investigation of the sensitivity of the system to the parameters of the genetic algorithm.
- An investigation of the sensitivity of the system to different target configurations. In particular, a method has been devised of parameterising target configurations according to a) the fraction of cells in non-quiescent states, and b) the degree of aggregation of these cells.

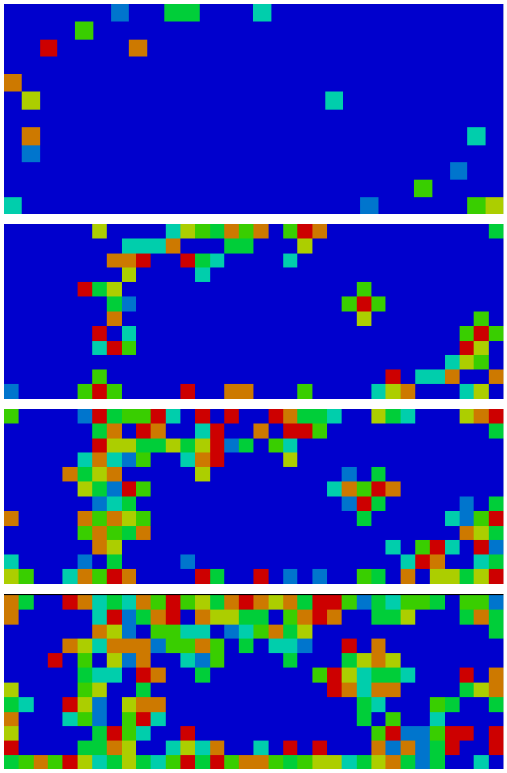

Figure 10: Phenotype of another good individual. This was from a batch 1 run, $\lambda = 0.80$. Initial configuration is shown at the top, with subsequent iterations below. Fitness at iteration 3 is 80.1.

From the experiments reported here, one of the most noticeable results is that the performance of the genetic algorithm across different λ values does not reflect the transitions between different regimes described by Langton (Langton 1986; 1991). There is a fairly linear increase in performance as λ is increased.

This may reflect the fact that the current fitness function is not rewarding *dynamics* per se, but is rather looking for a particular static target configuration.

It might be postulated that the improved performance at higher λ values simply reflects the fact that, as λ is increased, the CA dynamics are less likely to evolve to a fixed or simple periodic attractor. Each iteration of the CA is therefore more likely to generate a configuration that has not appeared before, so runs with high λ values are likely to score better simply because a greater number of distinct configurations are generated during the evaluation of each individual. However, Figure 6 indicates that this explanation is incorrect; as λ is increased, the iteration at which maximum fitness is achieved remains low (indeed, for batch 1 runs, it actually *decreases*). This suggests that individuals have difficulty in harnessing the dynamics of the environment at high λ values to reach the desired target configuration.

An alternative, and more convincing, interpretation of these results is that, as λ increases, the probability of any given cell being in the quiescent state, by definition, decreases. In the target configuration used in these experiments, over 42% of the cells were in the quiescent state. For high λ values, it would only be in the very first few iterations of the CA that this kind of proportion of cells would generally be in the quiescent state.

The additional fact that the runs with larger genomes (batch 4) produced only marginally better results than batch 1 suggests that the target configuration was genuinely hard to achieve, even if genomes could specify many more constraints. The planned runs with parameterised target configurations should elucidate this.

Figures 6, 7 and 8 provide indications of the relative utility of timed (t=0) and conditional genes.

When genomes can only specify the initial states of a small proportion of cells (e.g. batch 1 runs, with maximum genome length 50), conditional genes can effectively interact with the CA dynamics and enable an individual to achieve maximum fitness at a relatively late iteration. This is especially true for $\lambda \leq 0.4$, where the CA dynamics have simple attractors (i.e. Class I and II CA); compare the plot for batch 1 runs (maximum genome length 50, timed and conditional genes) in Figure 6, with the other two plots in that figure.[6] However, when longer genomes are allowed (batch 4), high fitnesses can be achieved within the first one or two iterations, so conditional genes should be no more useful than timed (t=0) genes in this case.

The above remarks are supported by Figures 7 and 8, which show the relative abundance of timed and conditional genes in small and large maximum genome sizes, respectively. With a low ceiling on genome size (batch 1, Figure 7), conditional genes outnumber timed (t=0) genes in runs with low λ values, especially in the region $0.2 \leq \lambda \leq 0.4$. When initial states can be specified for a greater proportion of cells (batch 4, Figure 8), the relative abundance of timed versus conditional genes is approximately equal for all λ values.

Taking a step back, the observations above suggest that trying to force the CA to adopt a particular, static, target configuration may not be the most productive approach in this kind of study. The use of a target configuration in these experiments has, in effect, been trying to push evolution down a particular path. A more fruitful approach might be to allow individuals to exploit the particular dynamics of their environment in any way they can. This can be achieved using biotic (natural) selection rather than abiotic (artificial) selection. A version of EvoCA that uses biotic selection is cur-

[6] A plausible reason for this is that, for CA with Class I or II dynamics, conditional genes can have a fairly local effect, giving a smooth fitness landscape upon which evolution can operate.

rently under development (EvoCA-B). In this version, many individuals exist concurrently on the CA array, and the dynamic activity of one individual may interfere with, or completely destroy, the activity of another. In this environment, an individual's primary goal is persistence, and its secondary goal is reproduction. However, there is no external specification of what makes a good phenotype; whether a particular dynamic organisation persists or perishes is determined by the CA dynamics and by the activity of other individuals in the neighbourhood. Individuals will need to be self-generating and self-maintaining to survive in this environment, which resonates with Maturana and Varela's concept of organisms as autopoietic organisations (Varela 1979; Maturana & Varela 1980). The use of natural selection will also be a more appropriate model of Pattee's idea of semantic closure (Pattee 1995b), as the survival value of dynamics initiated by the genotype will be determined within the dynamical system itself (as mentioned above, by the CA dynamics and the activity of neighbouring individuals), rather than by some externally defined fitness function.

Conclusion

A new perspective on modelling life has been discussed. This perspective views the origin and open-ended evolution of life as phenomena that take place within a dynamical system. It is argued that to reproduce these phenomena in a computational medium, no representational distinction should exist between phenotypes and the abiotic environment. Rather, the important representational distinction is between genotypes (viewed as relatively inert, symbolic structures) and phenotypes-plus-abiotic-environment (a dynamical system). The approach emphasises the role of the genotype in generating and sustaining phenotypes in this dynamical system environment by supplying initial conditions and constraints to the dynamics. The EvoCA system has been introduced as a tool to explore these ideas. The results of initial experiments have been presented and discussed. The genetic algorithm in EvoCA was, up to a point, able to improve the performance of individuals in generating a particular target configuration in the CA; the best end-of-run fitness achieved across the 500 runs was 84.8% (but note that, in general, 100% fitness is actually impossible to achieve in these experiments). The use of conditional genes in addition to timed genes improved performance in some situations, especially when the maximum allowable genome length was low. It has been suggested that in future work it may be more productive to allow individuals to exploit the dynamics of the environment without supplying external goals, by using natural rather than artificial selection.

Acknowledgements

The author is employed by the Hydra project, funded by the European Union's IST (FET) Programme. EvoCA is built on top of the CAMELot cellular automata system, supplied by the Edinburgh Parallel Computing Centre and developed as part of the COLOMBO Esprit project.

References

Bedau, M. A.; McCaskill, J. S.; Packard, N. H.; Rasmussen, S.; Adami, C.; Green, D. G.; Ikegami, T.; Kaneko, K.; and Ray, T. S. 2000. Open problems in artificial life. *Artificial Life* 6(4):363–376.

Burks, A. W., ed. 1968. *Essays on Cellular Automata*. University of Illinois Press.

Crutchfield, J. P., and Mitchell, M. 1995. The evolution of emergent computation. *Proceedings of the National Academy of Sciences U.S.A.* 92(23):100742–100746.

Dautenhahn, K.; Polani, D.; and Uthmann, T. 2001. Guest editors' introduction: Special issue on sensor evolution. *Artificial Life* 7(2):95–98.

Holland, J. H. 1976. Studies of the spontaneous emergence of self-replicating systems using cellular automata and formal grammars. In Lindenmayer, A., and Rozenberg, G., eds., *Automata, Languages, Development*. New York: North-Holland. 385–404.

Langton, C. G. 1986. Studying artificial life with cellular automata. *Physica D* (22):120–149.

Langton, C. G. 1991. Life at the edge of chaos. In Langton, C.; Taylor, C.; Farmer, J.; and Rasmussen, S., eds., *Artificial Life II*, volume X of *SFI Studies in the Sciences of Complexity*, 41–91. Addison-Wesley.

Maturana, H. R., and Varela, F. J. 1980. *Autopoiesis and Cognition: The Realization of the Living*. Reidel.

Maynard Smith, J. 1986. *The Problems of Biology*. Oxford University Press.

McMullin, B., and Varela, F. J. 1997. Rediscovering computational autopoiesis. In Husbands, P., and Harvey, I., eds., *Fourth European Conference on Artificial Life*, 38–47. MIT Press/Bradford Books.

Pattee, H. 1995a. Artificial life needs a real epistemology. In Morán, F.; Moreno, A.; Merelo, J.; and Chacón, P., eds., *Advances in Artificial Life: Third European Conference on Artificial Life*, Lecture Notes in Artificial Intelligence, 23–38. Berlin: Springer.

Pattee, H. 1995b. Evolving self-reference: Matter, symbols, and semantic closure. *Communication and Cognition—Artificial Intelligence* 12(1–2):9–28.

Ray, T. S. 1991. An approach to the synthesis of life. In Langton, C.; Taylor, C.; Farmer, J.; and Rasmussen, S., eds., *Artificial Life II*, volume X of *SFI Studies in the Sciences of Complexity*, 371–408. Addison-Wesley.

Rocha, L. M., ed. 2001. *The Physics and Evolution of Symbols and Codes: Reflections on the Work of Howard Pattee*. Special issue of *BioSystems* 60(1-3).

Taylor, T. 2001. Creativity in evolution: Individuals,

interactions and environments. In Bentley, P. J., and Corne, D. W., eds., *Creative Evolutionary Systems*. Morgan Kaufman. chapter 1, 79–108.

Varela, F. J.; Maturana, H. R.; and Uribe, R. 1974. Autopoiesis: The organization of living systems, its characterization and a model. *BioSystems* 5:187–196.

Varela, F. J. 1979. *Principles of Biological Autonomy*. Amsterdam: North Holland.

von Neumann, J. 1966. *The Theory of Self-Reproducing Automata*. Urbana, Ill.: University of Illinois Press.

Wolfram, S. 1984. Universality and complexity in cellular automata. *Physica D* 10:1–35.

Wolfram, S. 1986. *Theory and Applications of Cellular Automata (Including Selected Papers 1983 – 1986)*. Singapore: World Scientific.

Yaeger, L. 1994. Computational genetics, physiology, metabolism, neural systems, learning, vision and behavior or poly-world: Life in a new context. In Langton, C., ed., *Artificial Life III*, volume XVII of *Santa Fe Institute Studies in the Sciences of Complexity*, 263–298. Addison-Wesley.