Long-Term Evolutionary Dynamics in Heterogeneous Cellular Automata

David Medernach[1], Taras Kowaliw[2], Conor Ryan[1], and René Doursat[3]
1. Department of Computer Science and Information Systems, University of Limerick, Ireland
2. Institut des Systèmes Complexes, Paris Ile-de-France (ISC-PIF), CNRS, Paris, France
3. School of Biomedical Engineering, Drexel University, Philadelphia, USA

ABSTRACT

In this work we study open-ended evolution through the analysis of a new model, HetCA, for "heterogeneous cellular automata". Striving for simplicity, HetCA is based on classical two-dimensional CA, but differs from them in several key ways: cells include properties of "age", "decay", and "quiescence"; cells utilize a *heterogeneous* transition function, one inspired by genetic programming; and there exists a notion of *genetic transfer* between adjacent cells. The cumulative effect of these changes is the creation of an *evolving ecosystem of competing cell colonies*. To evaluate the results of our new model, we define a measure of phenotypic diversity on the space of cellular automata. Via this measure, we contrast HetCA to several controls known for their emergent behaviours—homogeneous CA and the Game of Life—and several variants of our model. This analysis demonstrates that HetCA has a capacity for long-term phenotypic dynamics not readily achieved in other models. Runs exceeding one million time steps do not exhibit stagnation or even cyclic behaviour. Further, we show that the design choices are well motivated, as the exclusion of any one of them disrupts the long-term dynamics.

Categories and Subject Descriptors

F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*Self-modifying machines*; I.2.2 [**Artificial Intelligence**]: Automatic Programming—*Program synthesis*

Keywords

cellular automata, open-ended, evolution, genetic programming, artificial ecosystem

1. INTRODUCTION

It is a truism that the complexity of life increases with time, even if the means or measures of it are controversial. Recreating this effectively endless self-generation of new mechanisms and capabilities would be fascinating for its insight into our own origins, and enticing as a harnessable creative force. In this paper, we introduce a new artificial life model based on a discrete dynamical systems framework. We have extended classical 2D cellular automata (CA), our chief modification being the allowance for heterogeneous transition functions. These changes convert a CA system into a new kind of "ecosystemic" model, where different genomes compete for existence. The value of such a model resides precisely in its simplicity: we aim to observe that long-term dynamics can be achieved quite naturally, given appropriate and plausible hypotheses. To demonstrate this open-endedness, we will show that our device is capable of supporting long-term dynamical behaviour, more so than control models such as homogeneous CA and the Game of Life. In particular, we will exhibit examples of the strategies discovered by our system, which are characterized by the *emergence of competitive behaviour*.

Our paper is organized as follows: in the remainder of this section, we review open-ended evolution and cellular automata. Next, we introduce our novel modifications to cellular automata, HetCA (for "heterogeneous cellular automata"), and describe their typical effects. Following this, we define a measure of phenotypic diversity and use it to illustrate the long-term dynamics of HetCA relative to several controls. Finally, we explore some specific examples of system outputs to discuss their behaviour qualitatively.

1.1 Artificial Life Models

A major motivation for work in artificial life is to create *open-ended evolution*, or systems in which novel artifacts are continuously produced. The first example of work of this kind, by Barricelli, consisted of a one dimensional matrix of simple rules which would copy and move [10]. Perhaps the best known Alife model, however, is Ray's Tierra [20]. In this world, there is competition between replicating computer programs in a virtual machine. Later, inspired by Tierra, another evolutionary system called Avida became popular and was extended in many directions [1]. High-level systems exist as well, where agents execute complex predefined functions, such as eating or fighting in Polyworld [28]. In all these cases, there is no particular specified goal: interesting and sometimes unexpected phenomena emerge from the interaction of individuals and their environment.

An important insight into the majority of such successful systems is that they involve *ecosystemic interactions*. Several researchers [8, 21, 14] claim that ecosystemic models are well suited to the generation of life-like behaviours, and perhaps even creativity. Further, an artificial system shown to

GECCO'13, July 6-10, 2013, Amsterdam, The Netherlands.

pass Bedau et al.'s test for unbounded evolutionary activity had its success attributed to ecosystemic interactions [5].

1.2 Cellular Automata

Cellular automata are an abstract framework created by von Neumann and Ulam in the 1940's to study self-replicating machines, which soon became a general model of discrete dynamical systems. Since then, CA have been intensely studied experimentally and theoretically [12, 27].

Here, by "classical" two-dimensional CA we mean:

- a toroidal lattice "world" of cells, $W = \{c = (x, y)\}$, with $|W| = w \times h$ (width and height, here 800×600)
- a neighbourhood for each cell: $\nu(c) = \{c, c_1, ..., c_{|\nu|}\}$, where $|\nu_{\mathrm{V}}| = 5$ denotes the von Neumann neighbourhood and $|\nu_{\mathrm{M}}| = 9$ the Moore neighbourhood.
- a discrete time, $t = 0, 1, ...$
- an alphabet of cell states, $\Sigma = \{s_1, ..., s_{|\Sigma|}\}$
- a state $s(c, t) \in \Sigma$ for each cell $c \in W$ at each time t
- a single system-wide transition function $\varphi : \Sigma^{|\nu|} \to \Sigma$

A CA is initialized with random states across the world. At time $t + 1$, each state is updated in parallel according to

$$s(c, t + 1) = \varphi(\mathbf{s}(\nu(c), t)),$$

where $\mathbf{s}(\nu(c), t) = \{s(c, t), s(c_1, t), ...\}$ is a $|\nu|$-tuple representing the collective state of c's neighbourhood including itself. Therefore, an uncompressed representation of a CA transition function requires $|\Sigma|^{|\nu|}$ elements, or a listing for every possible collective neighbourhood state, and the total number of such rules is an astronomical $|\Sigma|^{|\Sigma|^{|\nu|}}$.

Cellular Automata are known to be a model of computation, as they can simulate a universal Turing machine [27]. Even some simple CA have the property of computational universality, such as Conway's Game of Life, a two-dimensional CA in which binary cells switch on or off according to a simple counting rule [6]. Furthermore, CA—or minor variants thereof—are often used as predictive models of natural phenomena [7, 27], artificial life models of self-reproduction [22, 18, 29], and models of morphogenesis, sometimes as part of the creation of engineered design [15, 17, 9, 25, 16, 2].

1.3 Evolutionary Cellular Automata

Given the importance of CA as models, and the size of the space involved, the search for "useful" CA rules is an attractive but difficult problem. There are several theoretical results on the impossibility of predicting analytically the outcome of a given CA rule set starting from a given configuration. The reverse attempt at generating local rules to create a target global pattern is also not possible in the general case. Under these constraints, a natural choice for exploring CA with desired properties is an evolutionary search. Unfortunately, CA are difficult to evolve naïvely [11, 13], and when used in an application, variants of the traditional cellular automaton framework are often used to improve evolvability [15]. There are several attempts to use representations inspired by *genetic programming* (GP) for the (homogeneous) transition function, often in the context of generating self-replicating structures [18, 3].

A key feature of traditional CA resides in a consistent transition function for all cells, making them good models of homogeneous physical laws or intra-organismal cellular growth. However, some experiments have also explored the use of *heterogeneous* transition functions [26, 23, 24].

Algorithm 1 HetCA update rule: accepts a world state $\mathbf{s}(W, t) = \{s(c, t)\}_c$, outputs the next world state $\mathbf{s}(W, t+1)$.

for each cell in the world ($c \in W$) **do**
  increment cell age $\to a(c, t + 1) := a(c, t) + 1$
  if cell is in *quiescent* state? ($s(c, t) = q$) **then**
    transfer genome from random eligible neighbour $\to$ $\varphi_{c,t+1} := (\Phi_{c,t} \neq \varnothing)\ ?\ \mathcal{U}[\Phi_{c,t}]\ :\ 0$
    if transfer happened? ($\varphi_{c,t+1} \neq 0$) **then**
      update state $\to s(c, t + 1) := \varphi_{c,t+1}(\mathbf{s}(\nu_{\mathrm{M}}(c), t))$
      reset age $\to a(c, t + 1) := 0$
    end if
  else if cell is in *decaying* state? ($s(c, t) = d$) **then**
    if cell older than decay end? ($a(c, t+1) > a_{\mathrm{dec}}$) **then**
      set state to quiescent $\to s(c, t + 1) := q$
    end if
  else if cell is in *living* state? ($s(c, t) \in \Lambda$) **then**
    if cell older than life end? ($a(c, t + 1) > a_{\max}$) **then**
      set state to decaying $\to s(c, t + 1) := d$
      reset age $\to a(c, t + 1) := 0$
    else
      transfer genome from eligible neighbour $\to$ $\varphi_{c,t+1} := (\Phi_{c,t} \neq \varnothing)\ ?\ \mathcal{U}[\Phi_{c,t}]\ :\ \varphi_{c,t}$
      possibly mutate genome with probability $p_{\mathrm{mut}} \to$ $\varphi_{c,t+1} := \mathrm{mutate}(\varphi_{c,t+1})$ (see Section 2.4)
      update state $\to s(c, t + 1) := \varphi_{c,t+1}(\mathbf{s}(\nu_{\mathrm{M}}(c), t))$
    end if
  end if
end for

2. THE HETCA MODEL

Our model HetCA diverges from classical 2D cellular automata in three ways:

- cells have properties of *decay* and *quiescence*;
- each cell contains its own transition function, also called *genome*, which has the ability to evolve over time and be transferred to its neighbours;
- these genomic transition functions are represented in a parsimonious way (see Section 2.2).

None of these concepts alone is novel: their conjunction, however, is. In short, HetCA is initialized as a world state $\mathbf{s}(W, 0)$ of cells with randomly initialized state values and genomes. We will refer to cells which are neither decaying nor quiescent as *living* cells. For each cell in W, the following processes run in parallel:

- **age:** old living cells become decaying cells, old decaying cells become quiescent
- **transfer:** cells can accept genetic material from living neighbours, which can make quiescent cells living cells again.
- **mutate:** a living cell's genetic material can change
- **update:** a living cell executes its genome, i.e. applies its transition function to its neighbourhood

A formal description is available in Algorithm 1. We describe these processes in more detail below.

2.1 Cell Quiescence and Decay

Since the HetCA alphabet comprises two special states, "quiescent" and "decay", and the other states are "living", we denote it here by $\Sigma = \{q, d\} \cup \Lambda$, where $\Lambda = \{l_1, ..., l_{|\Sigma|-2}\}$. Our metaphor is that living cells "age" and eventually "die", becoming inert for some period of time. During the decay

process, they continue ageing and eventually become quiescent, creating "free space".

A living cell or decaying cell c ages by incrementing an internal counter $a(c,t)$. If a living cell's counter passes a threshold $a_{\max}$, its state is converted to decaying: $s(c,t) = d$, its genome is discarded, and its age is reset: $a(c,t) = 0$. If a decaying cell's age counter passes another threshold, a_{dec}, then it is turned into a quiescent cell: $s(c,t) = q$.

Our motivation for including decay and quiescence was to create a form of *competition* for the cells. Decaying matter is a challenge for cell colonies, as it makes simple maximal growth unsustainable: cells need to "learn" to survive around decay, e.g. use it as a means of defending their genetic territory. To encourage this trend, we set the amount of time necessary for the elimination of decay to a much larger value than the lifespan of a living cell: $a_{\text{dec}} \gg a_{\max}$.

2.2 Heterogeneous Transition Functions and Genetic Transfer

By "heterogeneous" CA, we mean that each living cell c contains its own (potentially unique) transition function, which may also vary over time, thus is denoted by $\varphi_{c,t}$. The world W is initialized with a uniform random sampling of cell states $\{s(c,0)\}_{c\in W}$ and transition functions $\{\varphi_{c,0}\}_{c\in W}$. Then, each cell determines its next state according to its own transition function: $s(c,t+1) = \varphi_{c,t}(\mathbf{s}(\nu_{\text{M}}(c),t))$.

Moreover, a transition function can be randomly *transferred* between neighbouring cells, but under two conditions: only (1) from a living cell to a living or quiescent cell, and (2) if this function is "resolved" in the new neighbourhood around that cell, meaning that its output must be a living cell, too. The motivation for this requirement is to discourage random occurrences of decaying and quiescent cell states because these states are "sinks" (as they are associated with a loss of transition function, hence are not updated).

We now describe this process more formally. We will use ν_{M} to resolve transition functions, and ν_{V} to chose neighbourhoods for resolution. Given a transition function φ, we denote by $\mathscr{R}_t(\nu_{\text{M}}(c),\varphi)$ the fact that a neighbourhood $\nu_{\text{M}}(c)$ "resolves" φ at time t and define it by

$$\mathscr{R}_t(\nu_{\text{M}}(c),\varphi) \;\Leftrightarrow\; \varphi(\mathbf{s}(\nu_{\text{M}}(c),t)) \in \Lambda.$$

We also denote by $\lambda_t(c)$ the subset of $\nu_{\text{V}}(c)$ containing the living neighbours of c at t: $\lambda_t(c) = \{c' \in \nu_{\text{V}}(c) \,|\, s(c',t) \in \Lambda\}$. Then, if cell c is living or quiescent at time t, it may receive at time $t+1$ the transition function from one of its living neighbours c' chosen randomly, only if that new function is resolved the current neighbourhood state of c. This reads

$$\varphi_{c,t+1} = \mathcal{U}[\Phi_{c,t}]\,, \text{ where}$$
$$\Phi_{c,t} = \{\varphi_{c',t} \,|\, c' \in \lambda_t(c) \text{ and } \mathscr{R}_t(\nu_{\text{M}}(c),\varphi_{c',t})\}$$

is the set of eligible neighbouring transition functions, and $\mathcal{U}$ denotes a uniformly random draw from a set of elements. Note that if the original cell c was quiescent, the set of neighbouring transition functions might be empty. In this case, there can be no transfer and c remains quiescent.

2.3 Transition Function Representation

We now describe the genomic format of $\varphi_{c,t}$. Given that uncompressed transition functions require an enormous state space $|\Sigma|^{|\Sigma|^{|\nu|}}$, and also recalling that transition functions are notoriously difficult to evolve, we elected to use a form of

```
int CA-LGP(int n1, ..., int n4 ) {
  double R1 = n1;
  double R2 = n2;
  double R3 = n3;
  double R4 = n4;
  double R5 = 0;
  double R6 = 1;
  double R7 = 0;
  double R8 = 4;
  //double R9 = 2;
  R1 = delta(R1, R6);
  R1 = R5 + R1;
  R7 = delta(R2, R6);
  R1 = R1 + R7;
  R7 = delta(R3, R6);
  //R9 = pow(R7, R7);
  R1 = R1 + R7;
  R7 = delta(R4, R6);
  R6 = R1 + R7;
  R5 = R8 - R6;
  //R2 = safeDiv(R2, R7);
  return argmax {R5, R6};
                i-N
}
```

N neighbourhood registers

|Σ| state registers (initialized genetically)

additional registers (initialized genetically)

length n_{reg}

program statements length less than n_{prog}

output is always index of maximum state register

Figure 1: Example CA-LGP program in pseudo-java notation. This program accepts a von Neumann (4-cell) neighbourhood, and assumes an alphabet of two cell states, $\Sigma = \{0,1\}$. It first computes the number of state-1 cells in the neighbourhood, then the number of state-0 cells. Finally, it outputs the central state reflecting the majority. Note the presence of neutral code, displayed in grey.

compression of their genomic representation. To this end, we developed a new representation for CA transition functions inspired by linear genetic programming (LGP) [4]. LGP is a desirable choice as its programs are easy to initialize and mutate, fast to execute, naturally modular and resistant to bloat, and make use of neutral code, which is believed to increase evolvability [4, 19].

We name our new representation of φ a **CA-LGP** program. Like any CA transition function, it maps the space of neighbourhood states to a new cell state: $\Sigma^{|\nu|} \to \Sigma$, but also, like any LGP representation, it provides an evolvable representation framework in which φ can be decomposed into an alphabet of elementary functions tuned by a few parameters. A CA-LGP program φ consists of:

- a list of $n_{\text{reg}} = |\nu| + |\Sigma| + n_{\text{add}}$ registers, written $\{R_i\}$:
  - $|\nu|$ *neighbourhood registers* holding the state values $\mathbf{s}$, the input to the program
  - $|\Sigma|$ *state registers*, genetically-specified constants
  - n_{add} *additional registers*, also genetically-specified constants
- a list of at most n_{prog} program statements, each of the form $R_i = \text{op}(R_j, R_k)$ for some operator op and some register indices i, j, k.
- a return statement, which returns the state associated with the maximum value state register.

An example of CA-LGP program is shown in Figure 1.

Table 1: Function set.

op. name	**action** on inputs (x, y)
abs	$\|x\|$
plus	$x + y$
delta	1, if $\|x - y\| < 1/10000$; 0 o.w.
dist	$\|x - y\|$
inv	$1 - x$
inv2	safeDiv$(1, x)$
magPlus	$\|x + y\|$
max	$\max\{x, y\}$
min	$\min\{x, y\}$
safeDiv	x/y if $\|y\| > 1/10000$; 1 o.w.
safePow	x^y, if defined; 1 o.w.
thresh	1, if $x > y$; 0 o.w.
times	xy
zero	1, if $\|x\| < 1/10000$; 0 o.w.

2.3.1 CA-LGP Genetic Initialization

A CA-LGP program can be initialized randomly by:

- choosing a number of additional registers
- specifying initial values for the state registers and additional registers with integers chosen uniformly randomly in the range $\{0, ..., |\nu|\}$. This maximum value is small enough to not encumber evolvability, but large enough to potentially act as a divisor for normalization (say, to compute averages).
- specifying each program statement by selecting four values uniformly randomly for i, j, k, and op, among the available registers and function set (Table 1).

2.4 Genetic Mutation

Given some particular CA-LGP, we can apply a mutation operator to generate a genetically similar CA-LGP. For this, we randomly choose either a micro- or a macro-mutation:

- Micro-mutation: for each additional register and for each program statement, a mutation is applied with a small probability and re-initializes one component of the statement.
- Macro-mutation: we choose one of the following, with equal probability:
  - If the program size is less than maximum n_{prog}, a randomly initialized program statement is added.
  - If the program size is greater than 2, a randomly selected program statement is removed.

The probability of a living cell being mutated, i.e. executing the above algorithm, is set to a constant $p_{\text{mut}} = 0.0008$.

3. MODEL BEHAVIOUR

As previously mentioned, HetCA is initialized with a uniformly random mixture of cell states and genomes. The vast majority of genomes turn to quiescence or decay within the first few time steps. This is not surprising since two of our cell states, q and d, are sinks, and randomly initialized genomes have no reason to avoid such states.

Usually, a small number of genomes (between one and three) survive this initial extinction, forming small groups of cells in between fields of decay[1]. Once decaying cells turn to quiescent, these surviving clumps quickly grow to cover the entire world. This often leads to a second or even third extinction event, after which the change from decay to quiescence becomes desynchronized. Usually, these early extinction events disappear before time step 10,000.

Following this, various populations come and go, that is, loose clusters of similar phenotypic patterns can be seen throughout the world, with new patterns materializing and old patterns disappearing with time. Occasionally, a new phenotypic pattern will emerge and quickly cover the entire world, indicating that a beneficial mutation has occurred. Informal experimentation convinced us that using different parameter values for the size of the world, and for $a_{\max}$ and a_{dec}, led to similar results, qualitatively speaking. The values chosen for experimentation in the following sections are largely arbitrarily, and we expect similar values to produce the same trends.

Our motivation for including quiescent and decay types was to encourage competition. Specifically, we expected genomes to "learn" to use decay as a barrier for their "territory", which would allocate them space to grow but block genetic transfer from neighbouring cells. Indeed, in many runs we have observed periods where populations selected voluntary decay with high probability. Since voluntary decay is a form of "genetic suicide", we view this as evidence that indeed, there are evolvable means of exploiting decay for survival advantage. Figure 2 shows one such successful example of a species relying on voluntary decay.

4. PHENOTYPIC DIVERSITY

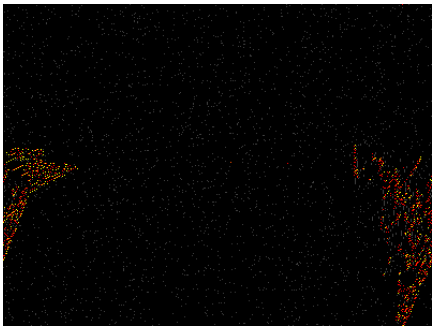
To quantify the qualitative results seen above, we designed a series of *measures of the diversity* of the system. The notion of "genotypic distance" was rejected, however: given that most transition functions φ encounter only a small proportion of all potentially available neighbourhood states, any representation of φ inevitably contains a lot of meaningless information, which would add too much noise to a direct comparison. This difficulty in measuring the distance between genetic programs, or any computer program for that matter, is well known. Therefore, we chose to focus on measuring the *differences in phenotypic patterns* over time, indirectly measuring the arrival of new "species" via the organizational behaviour that they exhibited in the world.

4.1 Choice of a phenotypic diversity measure

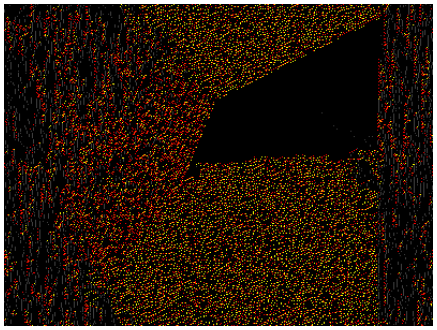
Our goal is to automatically measure the phenotypic patterns associated with the speciation events observed above. This would allow us to detect these events and estimate the number of *species* in our world, keeping in mind that genomically distinct cells can be phenotypically indistinguishable from each other. Formally, a "species" here refers to a family of genomes $\{\varphi_{c,t}\}$ which have identical phenotypic effect in the world (and, naturally, should not be confused with a cell state $s \in \Sigma$, which any cell of any species may potentially assume). Thus, we want to define a metric that shows maximal variance for the events that we consider significant in our qualitative observations.

The simplest metrics involve the *number of cells* in each state (i.e. each colour in Figure 2) at each time step. We

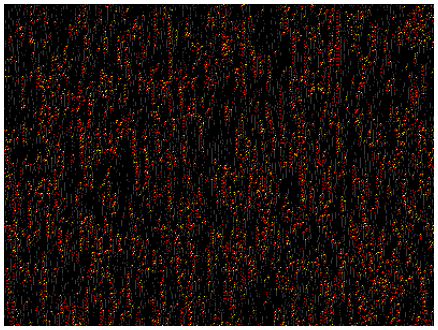
[1]It is not uncommon for all genomes to die following the initial extinction events, either immediately (before iteration 200) or, less frequently, after the first re-growth stage (before iteration 1500). This leaves a trivial world devoid of genomes. Given our particular parameters, these extinction events occur approximately 75% of the time. In these cases we simply restart the simulation, which is not computationally expensive since the run time to an extinction is short.



$t = 1350$: Two distinct genomes have generated two colonies. One is sparse, making prodigious use of voluntary decay; the second, less sparse, grows more quickly.



$t = 1950$: The two species have percolated, with the less sparse species occupying more space.



$t = 3300$: The sparse species has eliminated the other species entirely.

Figure 2: View of a HetCA run *(best viewed in colour)*. Quiescent cells are drawn in black, decay cells in grey, and living cells are drawn in other colours based on state.

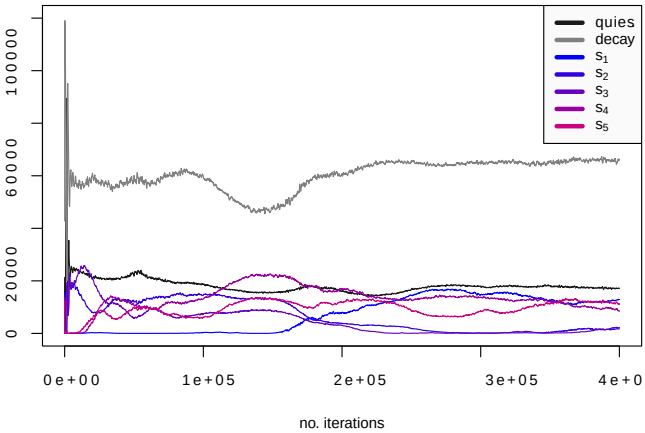


Figure 3: Plot of the number of cells by state over time in a typical HetCA run. The steep increases and decreases in the early iterations correspond to initial periods of rapid growth and extinction.

deal here with five different living states (Figure 3) and denote these metrics by $\{N_i(t)\}$ with $i = -1, 0, 1, ..., 5$, where $i = -1$ and 0 represent states q and d respectively (thus here, $|\Sigma| = 7$ and $|\Lambda| = 5$). However, while these time variations in "isostate" population sizes (on the long term) may constitute some indication of genotypic changes in the world, hence changes in species, it is at best highly indirect (again, states are *not* genotypes). We cannot exclude the possibility that several distinct species generate similar proportions of cell states. In fact, given the incremental nature of evolution, this possibility is very likely.

Our first attempt at a better diversity measure is based on *entropy*. We define the spatial frequency of a given state $s_i \in \Sigma$ as $\rho(s_i) = N_i/|W|$, and define the global entropy by

$$H(t) = \frac{1}{\log|\Sigma|} \sum_{i=-1}^{|\Lambda|} -\rho(s_i) \log \rho(s_i).$$

Next, we measure the variance of the distribution of cell states. First, we determine the "median" cell state s_{med}, i.e. the state whose frequency of occurrence is median. From this, we write the "gross" cell variance in the world as:

$$\sigma_{\text{gross}} = \frac{1}{\sqrt{|W|}} \left(\sum_{c \in W} \big(1 - \delta(s(c), s_{\text{med}})\big) \right)^{\frac{1}{2}} = \sqrt{1 - \rho(s_{\text{med}})},$$

where $\delta(s_i, s_j)$ is the Kronecker delta between two states. We also tried versions of σ_{gross} involving only living cells (i.e. $i = 1, ..., |\Lambda|$) but without significantly different results.

To measure the local organization of cell states, we would like a continuous metric showing the divergence of a particular neighbourhood from the "typical" neighbourhood. However, our cell states are distinct types with no natural ordering, making it difficult to define a "state distance". Initial attempts using δ were not particularly discriminating. Instead, we based our metric on the frequencies of cell states.

First, we compute $s_{\max}, s_{\min} \in \Sigma$, the states which occur with maximum and minimum frequency, respectively. We define the normalized frequency of each state $s_i \in \Sigma$ to be:

$$\widehat{\rho}(s_i) = \frac{\rho(s_i) - \rho(s_{\min})}{\rho(s_{\max}) - \rho(s_{\min})} = \frac{N_i - N_{\min}}{N_{\max} - N_{\min}}.$$

Next, for each cell $c \in W$ we compute the *local state frequency mean* and *local state frequency variance* as

$$\mu_{\text{loc}}(c) = \frac{1}{9} \sum_{c' \in \nu_{\text{M}}(c)} \widehat{\rho}(s(c')) \quad \text{and}$$

$$\sigma_{\text{loc}}(c) = \frac{1}{3} \left(\sum_{c' \in \nu_{\text{M}}(c)} \big(\widehat{\rho}(s(c')) - \mu_{\text{loc}}(c)\big)^2 \right)^{\frac{1}{2}}.$$

This last measure allows us to quantify how different a particular local neighbourhood is from the expected cell states in a more continuous way. Using σ_{loc}, we can now define the *global mean* and *global variance* (both of the local state frequency variance) over the world as

$$\mu_{\text{glob}} = \frac{1}{|W|} \sum_{c \in W} \sigma_{\text{loc}}(c) \quad \text{and}$$

$$\sigma_{\text{glob}} = \frac{1}{\sqrt{|W|}} \left(\sum_{c \in W} \big(\sigma_{\text{loc}}(c) - \mu_{\text{glob}}\big)^2 \right)^{\frac{1}{2}}.$$

The global measure σ_{glob} is an indication of the degree of "distinctiveness" of the neighbourhoods of the world relative to the expected neighbourhood. Thus it is sensitive not only

to differing proportions of cell states over space, but also to the local structure of those cell states.

4.2 Comparison of model versions

To demonstrate the capacity of HetCA to generate long-term phenotypic diversity, we contrast it with several control groups:

- **HetCA-a4**: HetCA with $a_{\max} = 4$ and $a_{\text{dec}} = 1850$.
- **HetCA-a7**: HetCA with $a_{\max} = 7$ and $a_{\text{dec}} = 1850$.
- **HetCA-noDec**: HetCA with $a_{\max} = 4$ and $a_{\text{dec}} = 0$.
- **HetCA-noMut**: HetCA-a4 with $p_{\text{mut}} = 0$, i.e. all genomes remain as initialized.
- **ClassicCA**: a randomly initialized classical CA, in which the universal transition function φ was also generated randomly (by CA-LGP).
- **GoL**: the classic Game of Life in a randomly initialized world and with a binary alphabet $\Sigma = \{0, 1\}$.

First, to develop our intuition, we show some typical evolution graphs for the σ_{glob} metric for each of the above groups (Figure 5). In these particular runs, we see that HetCA-noMut, ClassicCA, and GoL are all nearly trivial, in the sense that there are no significant phenotypic events happening in any of them. This was expected and contributes to validating our measure of diversity: since these three controls are all non-evolutionary, they are likely to produce homogeneous behaviour, which is reflected by a flat diversity curve. In contrast, both HetCA curves (red and blue) show very substantive changes over time, while also displaying (noisy) plateaus during certain intervals. The HetCA-noDec curve (green) also shows significant changes over time.

To be sure, these three non-evolutionary models are theoretically capable of universal behaviour, thus in principle can generate patterns that would maximize our measure. Yet, our intuition is that these sorts of configurations are very rare. Interestingly, the ClassicCA model generates behaviour that might be characterized as "chaotic" by CA standards (i.e. CA class 3) [27]. Yet, these cases typically do not register as "diverse" by our metric because they typically involve the repetition of similar regions across space.

HetCA-noDec, which is an evolutionary group, generates more interesting results. In this case, the curve is largely random, increasing and decreasing without any apparent pattern. Our hypothesis is that this behaviour is due to a lack of competition, i.e. genomes have little means to defend themselves against competitors, hence the populations fluctuate randomly.

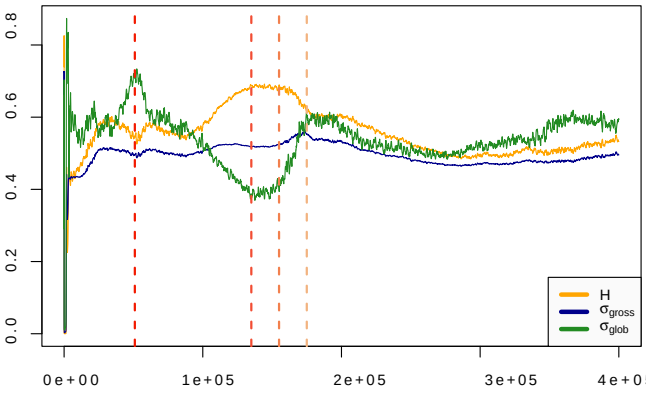
The HetCA-a4 and a7 runs, however, appear to generate larger plateaus with dramatic changes interspersed, the sort of phenomena that could be associated with "punctuated equilibria" of evolutionary innovation.

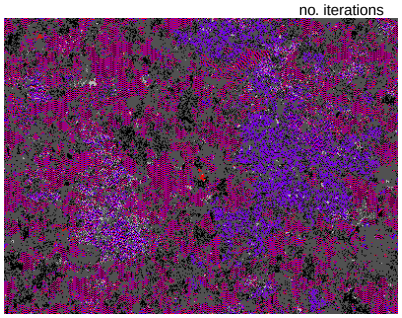
After looking at these graphs, we now want to compare these groups more quantitatively. To this aim, we define a single-value statistic, the *phenotypic variance*, as follows:

$$V_T[\sigma] = \frac{1}{\sqrt{T}} \left(\sum_{t=0}^{T-1} (\sigma(t) - E_T[\sigma])^2 \right)^{\frac{1}{2}},$$

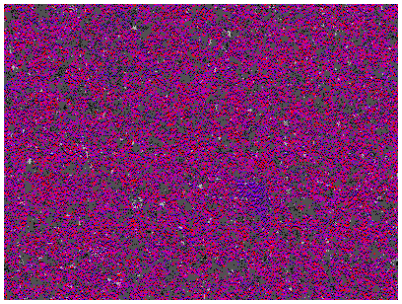
where σ stands for σ_{glob}, T is some total time of observation, and $E_T[\sigma]$ is the mean of $\sigma(t)$ over time, written:

$$E_T[\sigma] = \frac{1}{T} \sum_{t=0}^{T-1} \sigma(t)$$

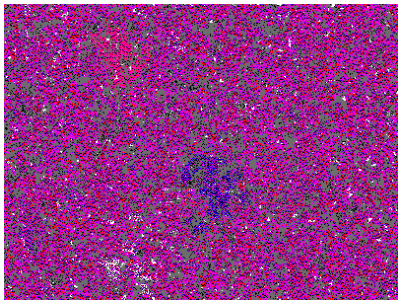




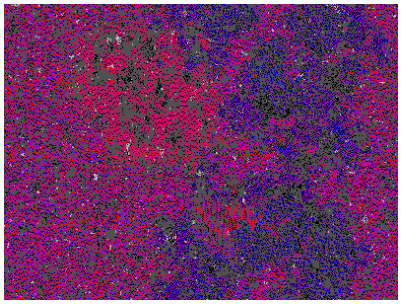
$t = 51,000$
Several competing species: high values for σ_{glob}.



$t = 135,000$
One species dominating: low values for σ_{glob}.



$t = 155,000$
Growth of new species (small blue cluster near centre, small pink cluster in upper left): σ_{glob} values increasing.



$t = 175,000$
A new mix of several competing species: high σ_{glob} values.

Figure 4: Correlation between phenotypic events and diversity measures: (*top*) a plot of our proposed measures over time; (*below*) screen shots of particular iterations corresponding to significant events.

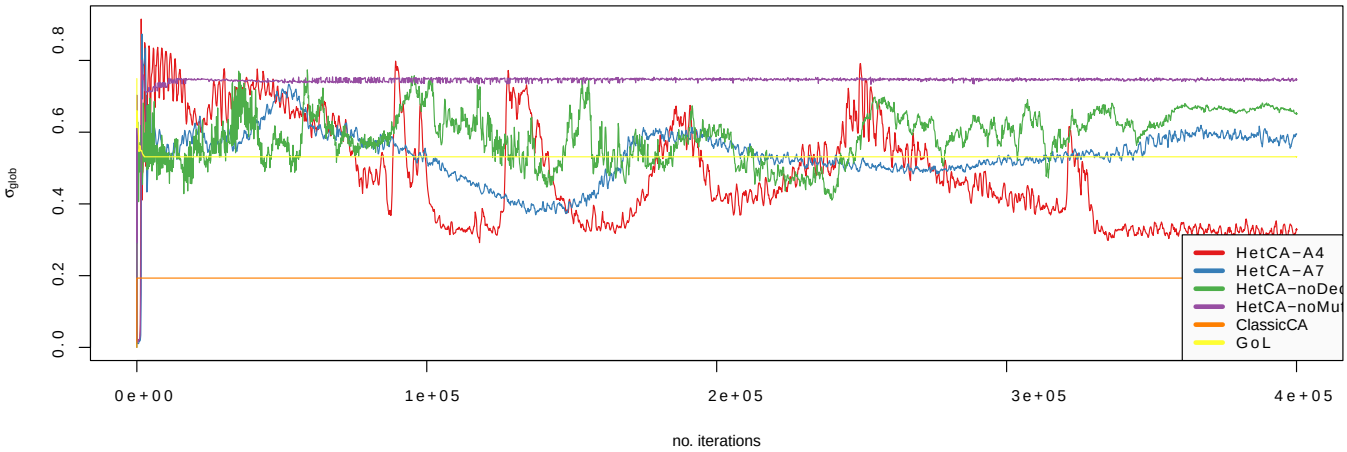


Figure 5: Plot of the σ_{glob} measure over time for HetCA (with two values of $a_{\max}$) and four control groups.

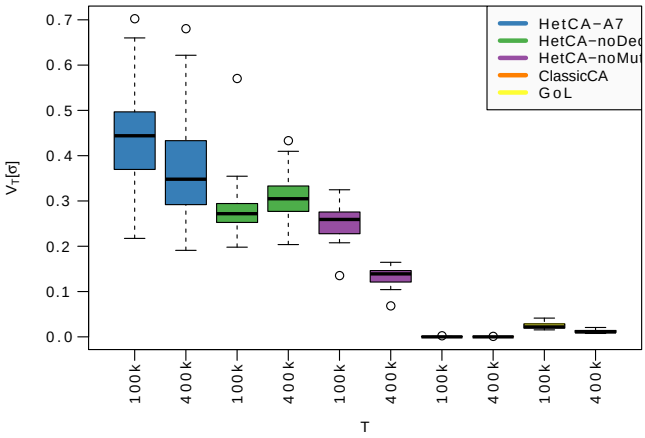


Figure 6: Comparison of values for $V_T[\sigma]$ by group, each for two durations $T = 100\text{k}$ and $T = 400\text{k}$.

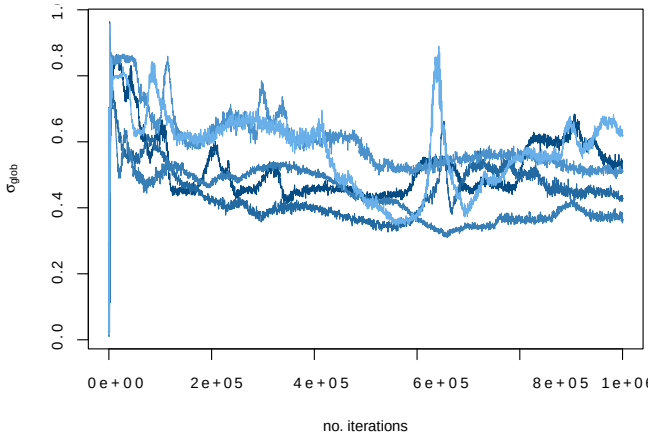


Figure 7: Plot of σ_{glob} over time for five independent long runs of HetCA-a7.

Note that $V_T[\sigma]$ is the variance *over time* of a measure that is itself an instantaneous variance *over space*. Next, we report the experimental *average phenotypic variance* (APV) $\overline{V_T}[\sigma]$ for each control group calculated over 100 independent runs based on random initial conditions. Figure 6 shows for each group a compact boxplot representing the distribution of $V_T[\sigma]$ values over different runs and their average $\overline{V_T}[\sigma]$.

Both the ClassicCA and GoL groups have consistently low APV. The HetCA-noMut group begins with high a phenotypic variance, but then decreases with time. This is likely due to an initial period of conflict between the randomly generated genomes, which later diminishes in intensity following the initial growth and extinction events. Thus, as we expected, there is little long-term variance in any of these three groups.

The two control groups closest in their σ curves over time (Figure 4) are the HetCA-a7 and HetCA-noDecay groups (blue and green). This is also reflected in Figure 6 by their APV clearly greater than all the other groups. Furthermore, HetCA-a7 is significantly higher than HetCA-noDecay (confirmed by a Welch's two-sample t-test yielding a P-value $p < 0.01$, even in the closer case of $T = 400\text{k}$ time steps). On the other hand, however, it seems that the APV of HetCA-a7 is decreasing as the averaging window widens, while the APV of HetCA-noDecay is increasing.

To explore this further, several very long runs of $T = 1\text{M}$ time steps were executed in both groups. In theory, given that the world is discrete and its states are finite, there must exist some point in time at which the dynamics of the world should loop back and become cyclic. However, due to the huge numbers of location-genome-state combinatorial possibilities, even the very long runs were not enough to reveal these types of cycles (Figure 7).

The phenotypic variance for the two groups is contrasted more precisely over different durations in Figure 8. Here, it is clear that the APV of HetCA-a7 initially decreases while that of HetCA-noDecay increases. By duration 400k, however, both values have plateaued. By duration 1M, there is no significant difference in phenotypic variance inside either group, compared to duration 400k. Still, at duration 1M, the phenotypic variance of HetCA-a7 is significantly greater than that of HetCA-noDecay (with P-value $p < 0.05$, where this slightly greater uncertainty is probably due to less data points on these long runs).

5. CONCLUSIONS

We have presented a simple extension of cellular automata, HetCA, consisting of three important changes from classical CA: transition function heterogeneity, transition function mutability, and cell decay and quiescence. Our hypothesis was that these changes would generate long-term dynamics. To test this, we developed a measure of phenotypic

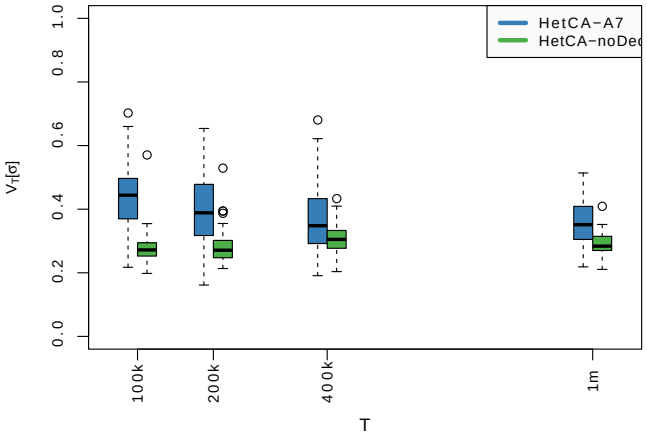


Figure 8: Comparison of values for $V_T[\sigma]$ by T for the HetCA-a7 and HetCA-noDecay groups.

diversity correlating with our intuition regarding significant phenotypic events.

Results showed that our model *HetCA was capable of long-term phenotypic dynamics*, sustaining a high level of variance over very long runs. Moreover, *HetCA displayed greater behavioral diversity* than classical cellular automata, such as the Game of Life. Finally, comparison between model variants showed that *all three changes were instrumental* in the generation of this long-term diversity.

6. ACKNOWLEDGMENTS

We thank Nicolas Bredeche (ISIR, Université Pierre et Marie Curie Paris 6) and Evelyne Lutton (INRIA Saclay) for valuable discussions. This work was financially supported by the Région Île-de-France, via the ISC-PIF.

7. REFERENCES

[1] C. Adami, C. Brown, and W. Kellogg. Evolutionary learning in the 2D artificial life system "Avida". In *Artificial Life IV*, pages 377–381, 1994.

[2] M. Bidlo and Z. Vašíček. Cellular automata-based development of combinational and polymorphic circuits: A comparative study. In G. Hornby, L. Sekanina, and P. Haddow, editors, *Evolvable Systems: From Biology to Hardware*, pages 106–117. Springer, 2008.

[3] M. Bidlo and Z. Vašíček. Evolution of cellular automata using instruction-based approach. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2012.

[4] M. Brameier and W. Banzhaf. *Linear Genetic Programming*. Springer, 2006.

[5] A. Channon. Unbounded evolutionary dynamics in a system of agents that actively process and transform their environment. *Genetic Programming and Evolvable Machines*, 7:253–281, 2006.

[6] J. Conway, R. Guy, and E. Berlekamp. *Winning Ways for Your Mathematical Plays, Vol. 2*. CRC Press, 2003.

[7] A. Deutsch and S. Dormann. *Cellular Automaton Modelling of Biological Pattern Formation: Characterization, Applications and Analysis*. Birkhauser, 2005.

[8] A. Dorin. The virtual ecosystem as generative electronic art. In G. R. et. al., editor, *European Workshop on Evolutionary Music and Art, Applications of Evolutionary Computing (EvoWorkshops)*, pages 467–476, 2004.

[9] R. Doursat. The growing canavas of biological development: multiscale pattern generation on an expanding lattice of gene regulatory networks. *InterJournal: Complex Systems*, 1809, 2006.

[10] D. Fogel. Nils Barricelli - artificial life, coevolution, self-adaptation. *IEEE Computational Intelligence Magazine*, 1(1):41–45, 2006.

[11] N. Ganguly, B. Sikdar, A. Deutsch, G. Canright, and P. Chaudhuri. A survey on cellular automata. Technical Report 9, Centre for High Performance Computing, Dresden University of Technology, 2003.

[12] A. Ilachinski. *Cellular Automata: A Discrete Universe*. World Scientific, 2001.

[13] T. Kowaliw and W. Banzhaf. Augmenting artificial development with local fitness. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 316–323, 2009.

[14] T. Kowaliw, A. Dorin, and J. McCormack. Promoting creative design in interactive evolutionary computation. *IEEE Transactions on Evolutionary Computation*, 16(4):523–536, 2012.

[15] T. Kowaliw, P. Grogono, and N. Kharma. Bluenome: A novel developmental model of artificial morphogenesis. In K. D. et al., editor, *6th Conference on Genetic and Evolutionary Computation (GECCO)*, 2004.

[16] T. Kowaliw, P. Grogono, and N. Kharma. The evolution of structural form through artificial embryogeny. In *IEEE Symposium on Artificial Life (IEEE-ALIFE)*, pages 425–432. IEEE, 2007.

[17] J. Miller. Evolving a self-repairing, self-regulating, french flag organism. In K. D. et al., editor, *6th Conference on Genetic and Evolutionary Computation (GECCO)*, pages 129–139. Springer-Verlag, 2004.

[18] Z. Pan and J. Reggia. Computational discovery of instructionless self-replicating structures in cellular automata. *Artificial Life*, 16:1064–5462, 2010.

[19] R. Poli, W. Langdon, and N. McPhee. *A Field Guide To Genetic Programming*. Lulu Enterprises, 2008.

[20] T. Ray. Evolution, ecology and optimization of digital organisms. Santa Fe Institute Technical Report 92-08-942, 1992.

[21] M. Rönkkö. An artificial ecosystem: Emergent dynamics and lifelike properties. *Artificial Life*, 13(2):159–187, 2007.

[22] H. Sayama. Self-protection and diversity in self-replicating cellular automata. *Artificial Life*, 10(1):83–98, 2004.

[23] M. Sipper. Computing with cellular automata: Three cases for nonuniformity. *Phys. Rev. E*, 57:3589–3592, Mar 1998.

[24] M. Sipper and M. Tomassini. Computation in artificially evolved, non-uniform cellular automata. *Theoretical Computer Science*, 217(1):81–98, 1999.

[25] G. Tufte. Gene regulation mechanisms introduced in the evaluation criteria for a hardware cellular development system. In *NASA/ESA Conference on Adaptive Hardware and Systems*, pages 137–144, 2006.

[26] G. Vichniac, P. Tamayo, and H. Hartman. Annealed and quenched inhomogeneous cellular automata (INCA). *Journal of Statistical Physics*, 45:875–883, 1986.

[27] S. Wolfram. *A New Kind of Science*. Wolfram Media Inc., 2002.

[28] L. Yaeger. Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or polyworld: Life in a new context. In *Artificial Life III*, pages 263–298, 1993.

[29] A. Yinusa and C. Nehaniv. Study of inheritable mutations in von Neumann self-reproducing automata using the golly simulator. In *2011 Symposium on Artificial Life (ALIFE)*, pages 211–217, 2011.