Artificial Life Journal, Volume 2, Number 1, pages 1-35, 1995.

Studying Artificial Life Using a Simple, General Cellular Model

Moshe Sipper
Department of Computer Science
Tel Aviv University
Tel Aviv 69978, Israel
e-mail: moshes@math.tau.ac.il

Abstract

Some of the major outstanding problems in biology are related to issues of emergence and evolution. These include: (1) how do populations of organisms traverse their adaptive landscapes? (2) what is the relation between adaptedness and fitness? (3) the formation of multi-cellular organisms from basic units or cells. In this paper we study these issues using a model which is both *general* and *simple*. The system, derived from the CA (cellular automata) model, consists of a two-dimensional grid of interacting organisms which may evolve over time. We first present designed multi-cellular organisms which display several interesting behaviors including: reproduction, growth, mobility. We then turn our attention to evolution in various *environments*, including: an environment in which competition for space occurs, an IPD (Iterated Prisoner's Dilemma) environment, an environment of spatial niches, and an environment of temporal niches. One of the advantages of AL models is the opportunities they offer in performing in-depth studies of the evolutionary process. This is accomplished in our case by observing not only phenotypic effects but also such measures as: fitness, operability, energy and the genescape. Our work sheds light on the problems raised above, and offers a possible path towards the long term two-fold goal of ALife research: (1) increasing our understanding of biology and (2) enhancing our understanding of artificial models, thereby providing us with the ability to improve their performance.

1 Introduction

A major theme in the field of Artificial Life is the emergence of complex behavior from the interaction of simple elements. Natural life emerges out of the organized interactions of a great number of non-living molecules, with no global controller responsible for the behavior of every part [Langton, 1989]. Closely related to the concept of emergence is that of evolution, in natural settings as well as in artificial ones.

Several major outstanding problems in biology are related to these two themes, emergence and evolution, among them [Taylor and Jefferson, 1994]: (1) how do populations of organisms traverse their adaptive landscapes- through gradual fine-tuning by natural selection on large populations, or alternatively in fits and starts with a good bit of chance to "jump" adaptive valleys in order to find more favorable epistatic combinations? (2) what is the relation between adaptedness and fitness, that is, between adaptation and what is selected for? It is now understood that natural selection does not necessarily maximize adaptedness, even in theory [Mueller and Feldman, 1988]. Chance, structural necessity, pleiotropy, historical accident and more detract from the 'optimization in nature' argument [Gould and Lewontin, 1979]. Another major theme is the formation of multi-cellular organisms from basic units or cells. Other related problems include: the origin of life, cultural evolution, origin and maintenance of sex, structure of ecosystems [Taylor and Jefferson, 1994].

This is just a partial list of open problems amenable to study by AL modeling. AL research into such issues holds a potential two-fold benefit: (1) increasing our understanding of biology and (2) enhancing our understanding of artificial models, thereby providing us with the ability to improve their performance (e.g. robotics, evolving software).

Our main interest in this paper lies in studying evolution, adaptation and multi-cellularity in a model which is both *general* and *simple*. Generality implies two things: (1) the model supports universal computation and (2) the basic units encode a general form of interaction rather than some specialized action (e.g. an IPD strategy, see Section 4.3). Simplicity implies that the basic units of interaction are modest in comparison to Turing machines. If we imagine a scale of complexity with Turing machines occupying the high end then simple machines are those that occupy the low end, e.g. finite state automatons. These two guidelines (generality and simplicity) allow us to evolve complex behavior with the ability to explore, in-depth, the inner workings of the evolutionary process (we shall come back to this point in the discussion in Section 5).

The model presented is essentially derived from the cellular automata model. Cellular automata (CA) are dynamical systems in which space and time are discrete. The states of cells in a regular grid are updated synchronously according to a local interaction rule. Each cell obeys the same rule and has a finite (usually small) number of states [Toffoli and Margolus, 1987]. The model was originally conceived by John von Neumann in the 1950's [von Neumann, 1966].

The CA model is perhaps the simplest, general model available. It is simple in that the basic units are small, local, finite state machines (rules). It is general since: (1) CA's support universal computation [Culik II *et al.*, 1990, Codd, 1968, von Neumann, 1966], and (2) the rules represent a general form of local interaction. The model has been applied to the study of general phenomenological aspects of the world, including: communication, computation, construction, growth, reproduction, competition and evolution [Burks, 1970, Smith, 1969, Toffoli and Margolus, 1987].

The main difficulty with the CA approach seems to lie with the extreme low-level representation of the interactions. CA's are programmed at the level of the local physics of the system and therefore higher-level cooperative structures are difficult to evolve in CA's [Rasmussen *et al.*, 1992]. Hence our intent is to increase the "capacity" for evolution while preserving the essential features of the CA model: (1) massive parallelism, (2) locality of cellular interactions and (3) simplicity of cells (finite state machines).

The basic model is detailed in Section 2 and the evolutionary aspect is presented in Section 4.1. We delineate below the three basic features by which it differs from the CA model [Sipper, 1994]:

1. Whereas the CA model consists of uniform cells, each containing the same rule, we consider the non-uniform case where different cells may contain different rules.
2. The rules are slightly more complex than CA rules.
3. Evolution takes place not only in *state space* as in the CA model, but also in *rule space*, i.e. rules may change (evolve) over time.

Thus, we have a grid of interacting, simple organisms (rules) which may evolve over time. The course of evolution is influenced by the nature of these organisms as well as by their *environment*. In nature the role of the environment in generating complex behavior is well known, e.g. as has been noted by [Simon, 1969] who described a scene in which the observed complexity of an ant's path is due to the complexity of the environment and not necessarily a reflection of the complexity of the ant. In our model, each rule is considered to have a certain *fitness*, depending upon the environment under consideration. As opposed to models such as GA (genetic algorithms) where each entity is independent, interacting only with the fitness function (and not the environment), in our case fitness depends on interactions of evolving organisms operating in an environment.

Note that the term 'environment' can convey two meanings: in the strict sense it refers to the surroundings, excluding the organisms themselves (e.g. in nature: sun, water, weather, terrain, etc') while the broad sense refers to the total system, i.e. surroundings + interacting organisms (e.g. ecosystem). In what follows the term is used in the strict sense, however we attain an environment in the broad sense, i.e. a total system of interacting organisms (see also [Bonabeau and Theraulaz, 1994]).

We consider various environments, including: the 'basic' environment where rules compete for space on the grid, an IPD (Iterated Prisoner's Dilemma) environment, an environment of spatial niches, and an environment of temporal niches. One of the advantages of AL models is the opportunities they offer in performing in-depth studies of the evolutionary process. This is accomplished in our case by observing not only phenotypic effects (i.e. evolution of cell states as a function of time) but also fitness, operability, energy and the genescape.

Our approach may be viewed as a non-uniform CA with enhanced rules [Sipper, 1994] and as such it is related to other works. [Garzon, 1990] presents two generalizations of cellular automata, namely discrete neural networks and automata networks. These are compared to the original model from a computational point of view which considers the classes of problems such models can solve.

In [Lee *et al.*, 1990, Qian *et al.*, 1990] adaptive stochastic cellular automata are considered which are essentially non-uniform automata whose rules are drawn from the same probability distribution function (PDF). Adaptation and learning are accomplished by evolving the PDF using gradient descent. Their approach focuses on the learning aspect where an automaton is trained to solve some problem (e.g. pole balancing). An important issue is the coding of the problem onto the CA structure, which is non-trivial and may be complex. Furthermore, the global state of the grid is considered, e.g. when generating a reward/penalty signal (our model is local).

The work of [Vichniac *et al.*, 1986] discusses a one-dimensional inhomogeneous CA in which a cell probabilistically selects one of two rules, at each time step. They showed that complex patterns appear characteristic of class IV behavior (see also [Hartman and Vichniac, 1986]). We shall discuss other models, which are more closely linked with ours, in the relevant sections ahead.

Our approach is different than the above works and is more in the spirit of Artificial Life where cellular automata provide us with "logical universes" [Langton, 1986]. These are: "synthetic universes defined by simple rules ... One can actually construct them, and watch them evolve." [Toffoli and Margolus, 1987].

In the next section we detail the basic model (without evolution which is presented in Section 4.1). In Section 3 we present designed multi-cellular organisms which display several behaviors including: reproduction, growth, mobility. These are interesting in and of themselves and also serve as motivation for the following section (Section 4) in which we turn our attention from the human watchmaker to the blind one, focusing on evolution. A discussion of our results is provided in Section 5.

2 The basic model

The two-dimensional CA model consists of a two-dimensional grid of cells, each containing the same rule, according to which cell states are updated in a synchronous, local manner. In our model the grid consists of cells which are either *vacant*, containing no rule, or *operational* containing a finite state automaton (rule) which can, in one time step:

1. Access its own state and that of its immediate neighbors (grid is toroidal).

2. Change its state and the states of its immediate neighbors. Contention occurs when more than one operational neighbor attempts to change the state of the same cell. Such a situation is resolved randomly, i.e. one of the contending neighbors "wins" and decides the cell's state at the next time step. Note that the cell itself is also a contender, provided it is operational.

3. Copy its rule into a neighoring *vacant* cell. Contention occurs if more than one operational neighbor attempts to copy itself into the same cell. Such a situation is resolved randomly, i.e. one of the contending neighbors "wins" and copies its rule into the cell. Note that in this case the cell itself is not a contender since it must be vacant in the first place for contention to occur.

At each time step every operational rule[1] simultaneously executes its appropriate rule entry, i.e. the entry corresponding to its current neighborhood configuration. Thus, state changes and rule copies are effected as explained above. Note that a vacant cell may be in any grid state as it can be changed by operational neighboring cells.

Whereas a rule in the CA model accesses the states of its neighbors but may only change its own state, our model allows state changes of neighboring cells and rule copying into them. Thus, our rules may be regarded as being more "active" than those of the CA model. Furthermore, different cells may contain different rules (non-uniformity). The third feature of our model as presented in Section 1 is the evolution which takes place in rule space, i.e. rules evolve as time progresses. This is detailed in Section 4.1. We first turn our attention to multi-cellular organisms in the next section.

3 Multi-cellularity

In this section we present a number of multi-cellular organisms which are composed of several cells, consisting of rules as described in Section 2. The organisms discussed below are designed rather than evolved and our intent is to demonstrate that interesting behaviors can arise using the dynamics described above. In the next section (Section 4) we shall focus on evolution. At this point the term 'multi-cellular' is loosely defined so as to refer to any structure composed of several cells, acting in unison. In Section 5 we examine more carefully the meaning of the term 'cell' and expand upon the general issue of multi-cellular organisms versus uni-cellular ones. The cellular space considered throughout this section is 3-state, 9-neighbor where states are denoted $\{0, 1, b\}$.

3.1 A self-reproducing loop

Our first example involves a simple self-reproducing loop motivated by Langton's work [Langton, 1984, Langton, 1986] who described such a structure in uniform cellular automata. His loop was later simplified by [Byl, 1989, Reggia *et al.*, 1993]. Langton's loop (motivated by [Codd, 1968]) makes dual use of the information contained in a description to reproduce itself. The structure consists of a looped pathway, containing instructions, with a construction arm projecting out from it. Upon encountering the arm junction the instruction is replicated, with one copy propagating back around the loop again and the other copy propagating down the construction arm, where it is translated as an instruction when it reaches the end of the arm.

The important issue to note is the two different uses of information, interpreted and uninterpreted, which also occur in natural self-reproduction, the former being the process of *translation*, and the latter *transcription*. In Langton's loops translation is accomplished when the instruction signals are "executed" as they reach the end of the construction arms, and upon the collision of signals with other signals. Transcription is accomplished by the duplication of signals at the arm junctions [Langton, 1984].

[1] Throughout this paper we use the terms operational cell and operational rule interchangeably.

The loop considered in this section consists of five cells and reproduces within six time steps . The initial configuration consists of a grid of vacant cells (i.e. with no rule) with a single loop composed of five cells in state 1, each containing the loop rule (Figure 1a). The arm extends itself by copying its rule into an adjoining cell, coupled with a state change to that cell. The new configuration then acts as data to the arm, thereby providing the description by which the loop form is replicated. When a loop finds itself blocked by other loops it "dies" by retracting the construction arm. Figure 1b shows the configuration after several time steps.

The loop rule is given in Figure 2. Note that most entries are identity transformations, i.e. they transform a state to itself, thereby causing no change (only 40 entries of the 3^9 are non-identity). In his paper [Langton, 1984] compares the self-reproducing loop with the works of [von Neumann, 1966] and [Codd, 1968], drawing the conclusion that although the capacity for universal construction, presented by both, is a *sufficient* condition for self-reproduction it is not a *necessary* one. Furthermore, as Langton points out, naturally self-reproducing systems are not capable of universal construction. His intent was therefore to present a simpler system that exhibits non-trivial self-reproduction. This was accomplished by constructing a rule in an eight-state cellular space, in which the dual nature of information, i.e. translation and transcription is utilized.

In the loop presented above simple transcription is accomplished as an integral part of a cell's operation, since a rule can be copied, i.e. treated as data. Once a rule is activated it begins to function by changing states in accordance with the grid configuration, thereby performing translation on the surrounding cells (data). Essentially, the loop operates by transcribing itself onto a neighboring cell while simultaneously writing instructions (in the form of grid states) that will be carried out at the next time step.

In Langton's system each grid cell initially contains the rule that supports replication whereas in our case the grid cells are initially vacant and the loop itself contains all the information needed. In both cases reproduction is not coded entirely into the "transition physics" but rather is "actively directed by the configuration itself" where "the structure may take advantage of certain properties of the transition function physics of the cellular space" [Langton, 1984]. Thus interest in such systems arises since they display an interplay of active structures taking advantage of the characteristics of cellular space.

3.2 Reproduction of passive structures by copier cells

In the previous section we described a self-reproducing loop, which exhibited a two-fold utilization of information, i.e. translation and transcription. In this section we examine a system of reproduction consisting of passive structures copied by active (mobile) cells. The motivation for our approach lies in the information flow in protein synthesis, where passive mRNA structures are translated into amino acids by active tRNA cells. Each tRNA cell matches one specific codon in the mRNA structure and synthesizes one amino acid. Note that our system is extremely simple with regards to the workings of the living cell and therefore the above analogy is (highly) abstracted.

Our system consists of stationary structures composed of vacant grid cells comprising

```
11
111

time = 0
```

```
11
1110

time = 1
```

```
11
11100

time = 2
```

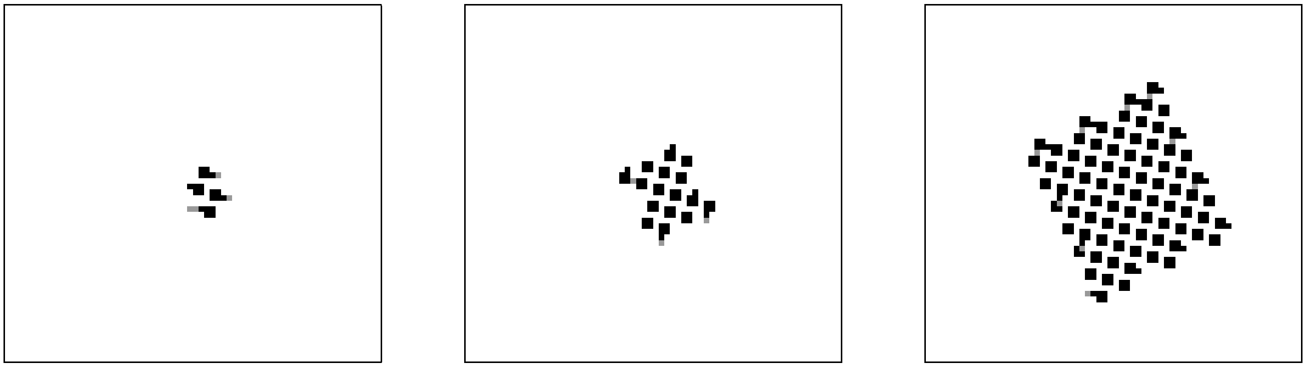
```
11
11101
    1
time = 3
```

```
11
11101
   11


time = 4
```

```
11
11011
   11
   1

time = 5
```

```
 1
11
11 11
   11
   1
   0
time = 6
```

(a)



time = 12   time = 28   time = 66

Black squares represent cells in state 1, non-filled squares represent cells in state 0 and white squares represents cells in state *b*.

(b)

Figure 1: **Self reproducing loop.**

$$\begin{array}{|c|c|c|}\hline 1 & & \\ \hline 1 & 1 & \\ \hline & & \\ \hline\end{array} \rightarrow \begin{array}{|c|c|c|}\hline 1 & & \\ \hline 1 & 1 & 0 \\ \hline & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline & & \\ \hline 1 & 0 & \\ \hline & & \\ \hline\end{array} \rightarrow \begin{array}{|c|c|c|}\hline & & \\ \hline 1 & 0 & 0 \\ \hline & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline & & \\ \hline 0 & 0 & \\ \hline & & \\ \hline\end{array} \rightarrow \begin{array}{|c|c|c|}\hline & & \\ \hline 0 & 1 & \\ \hline & 1 & \\ \hline\end{array}$$

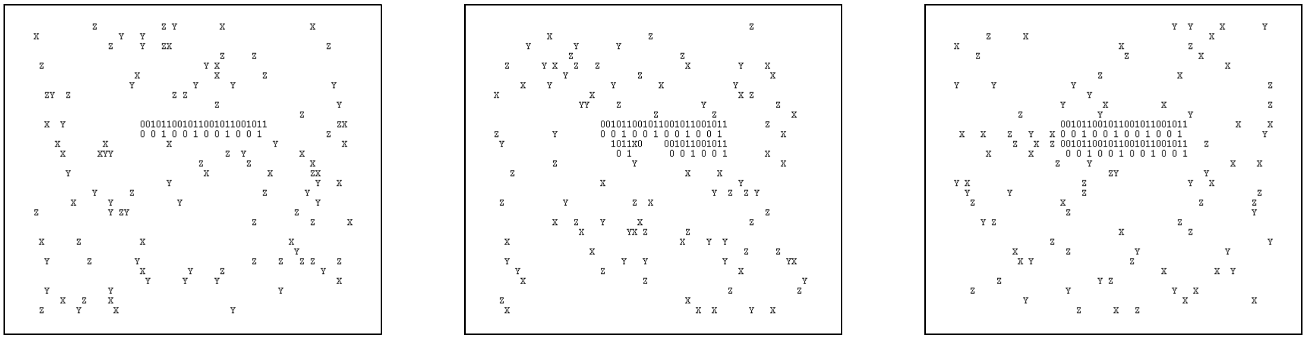
$$\begin{array}{|c|c|c|}\hline 0 & 1 & \\ \hline & 1 & \\ \hline & & \\ \hline\end{array} \rightarrow \begin{array}{|c|c|c|}\hline 0 & 1 & \\ \hline 1 & 1 & \\ \hline & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline & & \\ \hline 1 & 0 & 1 \\ \hline & 1 & 1 \\ \hline\end{array} \rightarrow \begin{array}{|c|c|c|}\hline & & \\ \hline 1 & 1 & 1 \\ \hline & 1 & 1 \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline 1 & 0 & 1 \\ \hline & 1 & 1 \\ \hline & & \\ \hline\end{array} \rightarrow \begin{array}{|c|c|c|}\hline 1 & 0 & 1 \\ \hline & 1 & 1 \\ \hline & 1 & \\ \hline\end{array}$$

$$\begin{array}{|c|c|c|}\hline 1 & & \\ \hline 1 & 1 & 0 \\ \hline & & 1 \\ \hline\end{array} \rightarrow \begin{array}{|c|c|c|}\hline 1 & & \\ \hline 1 & 0 & 0 \\ \hline & & 1 \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline 1 & & \\ \hline 1 & 0 & 1 \\ \hline & & 1 \\ \hline\end{array} \rightarrow \begin{array}{|c|c|c|}\hline 1 & & \\ \hline 1 & & 1 \\ \hline & & 1 \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline 1 & & \\ \hline 1 & 1 & 1 \\ \hline & & 1 \\ \hline\end{array} \rightarrow \begin{array}{|c|c|c|}\hline 1 & & \\ \hline 1 & & 1 \\ \hline & & 1 \\ \hline\end{array}$$

$$\begin{array}{|c|c|c|}\hline & & \\ \hline 1 & 1 & \\ \hline 1 & 1 & 0 \\ \hline\end{array} \rightarrow \begin{array}{|c|c|c|}\hline & 1 & \\ \hline 1 & 1 & \\ \hline 1 & 1 & 0 \\ \hline\end{array}$$

In all rule entries a state change from b to $0/1$ also involves a rule copy (note that all cells are initially vacant, i.e. with no rule, except the ones comprising the initial loop).
Each of the above entries consists of three further rotations (not shown). All other entries preserve the configuration.

Figure 2: **Self reproducing loop rule.**



time = 0 time = 435 time = 813

Figure 3: **Reproduction by copier cells.**

the passive data to be copied. The copy ("synthesis") process is accomplished by three types of copier cells, denoted X, Y, and Z which are mobile units, "swimming" on the grid, seeking an appropriate match (remember that cellular mobility is possible by using rule copying, see Section 2). When such a match occurs the cell proceeds to create the appropriate sub-structure, as in the case of a tRNA cell synthesizing the appropriate amino acid. The final result is a copy of the original structure.

The process is demonstrated in Figure 3. The initial configuration consists of a passive structure with X,Y and Z cells randomly distributed on the grid (Figure 3, *time* = 0). Each time step the copier cells move to a neighboring vacant cell (shown as white squares) at random, unless a match is found which triggers the synthesis process. Figure 3 (*time* = 435) shows the process at an intermediate stage and at the final stage (*time* = 813) when the copy has been produced.

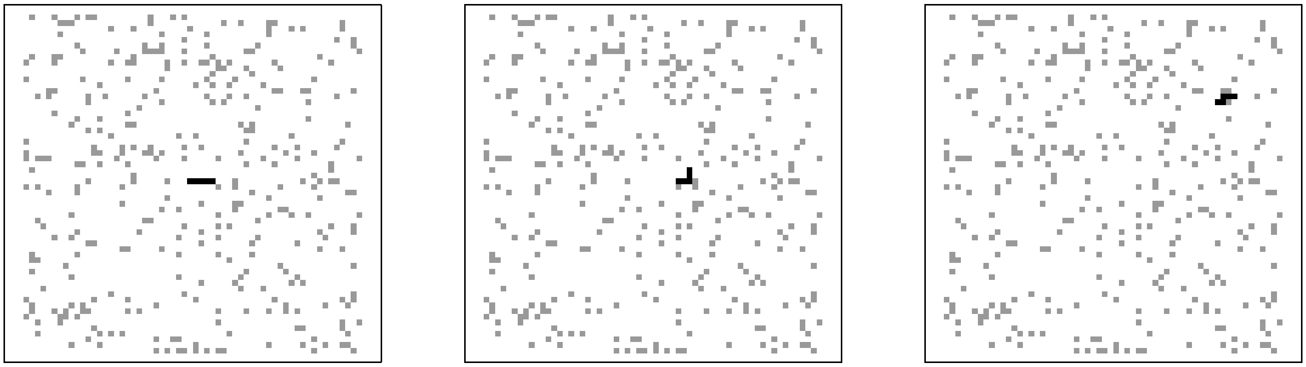
The X cell rule is detailed in Figure 4 (actually the rule template is shown. Y and Z cell rules may be analogously derived). The left rule is the match seeker, specifying the "codon" of the X cell. Once a match is found the cell builds a copy by applying the other two rules. After application of the right rule the copy has been constructed and the X cell dies. Note that most entries in the rule table specify a move to a random vacant cell in state b.

The copy created is not an exact duplicate but rather a "complementary" one. The reason for this is that we wish to avoid endless copying which would occur had an exact duplicate been created. Since our model is inherently local we cannot maintain a global variable specifying that the synthesis process has been completed. The only way to avoid an endless chain of duplicate sub-structures is by locally specifying that a copy has been completed. This is accomplished by creating a complementary sub-structure, which does not match any copier cell and is not duplicated further.

$$\begin{array}{|c|c|c|}\hline 0 & 0 & * \\ \hline 0 & X & * \\ \hline & & * \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline 0 & 0 & * \\ \hline 0 & & * \\ \hline & X,0 & * \\ \hline \end{array} \qquad \begin{array}{|c|c|c|}\hline 0 & & * \\ \hline & X,0 & * \\ \hline & & * \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline 0 & & * \\ \hline X,0 & 0 & * \\ \hline & & * \\ \hline \end{array} \qquad \begin{array}{|c|c|c|}\hline * & 0 & \\ \hline * & X,0 & 0 \\ \hline * & & \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline * & 0 & \\ \hline * & 0 & 0 \\ \hline * & & 0 \\ \hline \end{array}$$

'*' denotes the set of states: $\{0, 1, b\}$.
'X' denotes an X rule in a cell in state b. '$X, 0$' denotes an X rule in a cell in state 0.
All other entries specify a move to a random vacant cell in state b.

Figure 4: **Reproduction by copier cells: X cell rule.**



time = 0 time = 5 time = 33

Black squares represent cells in state 1, non-filled squares represent cells in state 0 and white squares represents cells in state b.

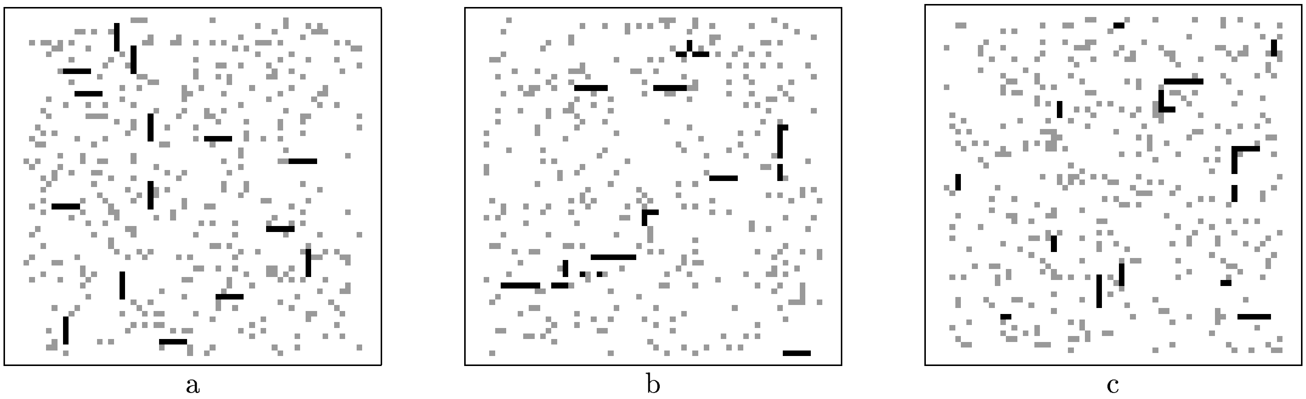
Figure 5: **A system consisting of a single worm.**

3.3 Mobility

In this section we introduce a worm-like structure which has the capacity to move freely on the grid. The system consists of worms which are active, mobile structures composed of operational cells in state 1, and blocks which are vacant cells in state 0. When a worm encounters a block it turns by 90 degrees and continues its movement (if there is a block obstructing the turn then the worm destroys it).

Figure 5 presents a system with a single worm, behaving as described above. When several worms are placed on the grid, interaction among them yields interesting phenomena (Figure 6). The following behavioral patterns are observed when two worms meet: one of them splits into two, both worms merge into one, a worm looses part of its body, both emerge unscathed. In all cases the resulting worms behave in the same manner as their ancestors.

The rule is detailed in Figure 7. Its simplicity is possible due to the power offered by our model (see Section 5). The emergent behavior is complex and exhibits different forms of interaction between the organisms inhabiting the grid. A worm acts as a single high



(a) an initial configuration of the system.
(b), (c) system configurations after several time steps.

Figure 6: **A system consisting of several worms.**

order structure and upon encountering other worms it may split, merge, shrink or emerge unscathed.

It is interesting to observe the formation of such a high order structure which operates by applying local rules. The worm rule essentially specifies how the head and tail sections operate independently. The overall effect is that of a single organism whose parts operate in unison. Living creatures may also be viewed in this manner, i.e. as a collection of independent cells operating in unison, thereby achieving the effect of a single "purposeful" organism (see Section 5).

3.4 Growth and replication

In this section we examine an enhancement of our basic model in which the following feature is added to the three presented in Section 2:

4. A cell may contain a *small* number of different rules. At a given moment only one rule is *active* and determines the cell's function. A non-active rule may be activated or copied into a neighoring cell.

This feature could serve as a possible future enhancement in the evolutionary studies as well (Section 4). At this point we present a system involving the growth and replication of complex structures which are created from grid cells and behave as multi-cellular organisms once formed. The system consists initially of two cell types, builders (A cells) and replicators (B cells), floating around on the grid.

Figure 8 demonstrates the operation of the system. At time 0 A and B cells are distributed randomly on the grid and there are two vacant cells in state 1 acting as the core of the building process. The A cells act as builders by attaching ones at both ends of the growing structure. Once a B cell arrives at an end growth stops there by attaching a zero (time 111).

$$\begin{array}{|c|c|c|}\hline * & * & * \\ \hline 1 & 1 & \\ \hline * & * & * \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline * & * & * \\ \hline 1 & 1 & 1 \\ \hline * & * & * \\ \hline \end{array} \qquad \begin{array}{|c|c|c|}\hline * & & * \\ \hline * & 1 & * \\ \hline * & 1 & * \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline * & 1 & * \\ \hline * & 1 & * \\ \hline * & 1 & * \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|}\hline * & * & * \\ \hline 1 & 1 & 0 \\ \hline * & * & * \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline * & 1 & * \\ \hline 1 & 1 & 0 \\ \hline * & * & * \\ \hline \end{array} \qquad \begin{array}{|c|c|c|}\hline * & 0 & * \\ \hline * & 1 & * \\ \hline * & 1 & * \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline * & 0 & * \\ \hline * & 1 & 1 \\ \hline * & 1 & * \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|}\hline * & * & * \\ \hline + & 1 & 1 \\ \hline * & * & * \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline * & * & * \\ \hline + & & 1 \\ \hline * & * & * \\ \hline \end{array} \qquad \begin{array}{|c|c|c|}\hline * & 1 & * \\ \hline * & 1 & * \\ \hline * & + & * \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|}\hline * & 1 & * \\ \hline * & & * \\ \hline * & + & * \\ \hline \end{array}$$

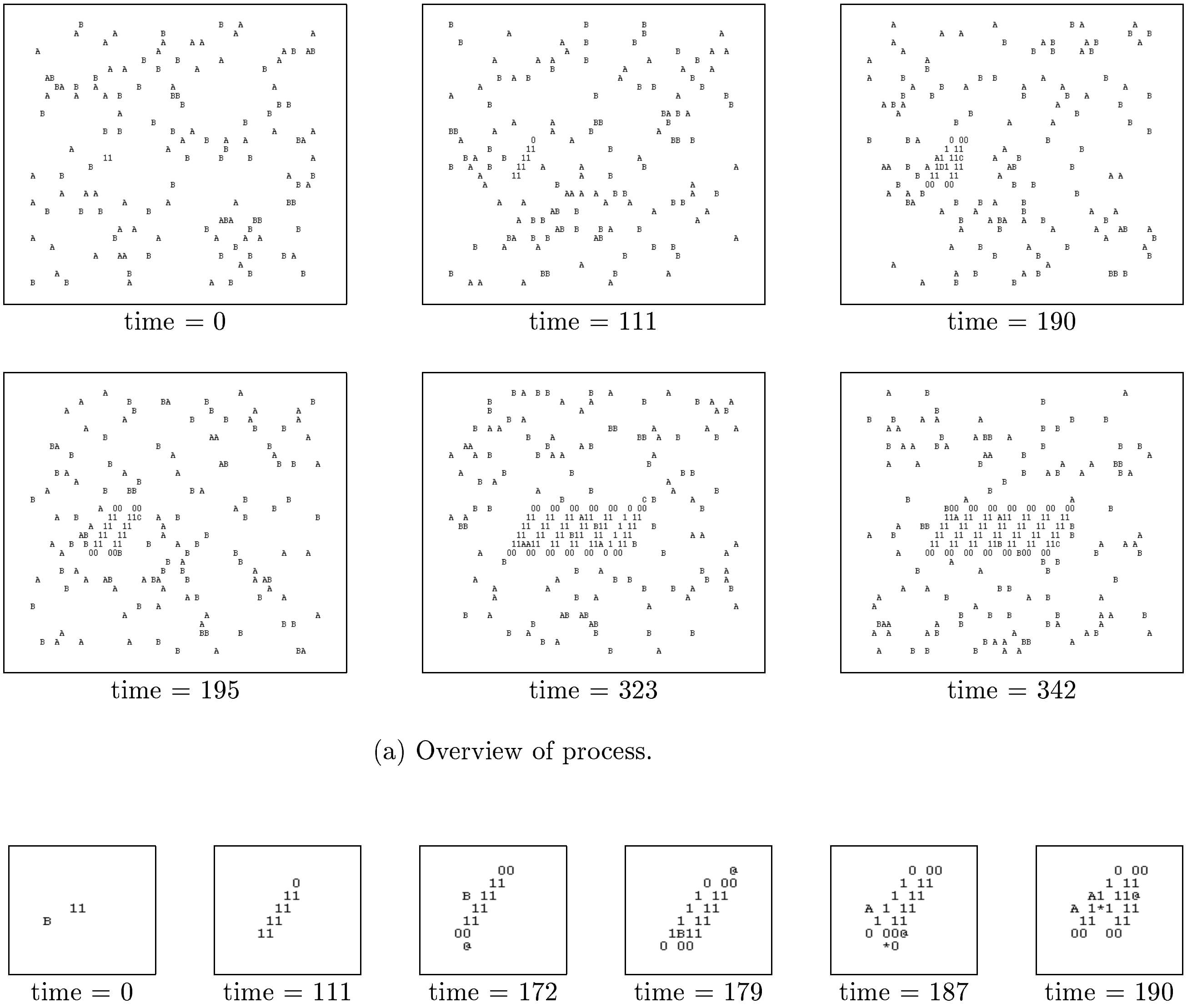
'*' denotes the set of states: $\{0, 1, b\}$.
'+' denotes the set of states: $\{0, b\}$.
All other entries preserve the configuration.

Figure 7: **Worm rule.**

When a B cell arrives at the upper end of a structure already possessing one zero a C cell is *spawned*, which travels down the length of the structure to the other end. If that end is as yet uncompleted the C cell simply waits for its completion (time 172). The C cell then moves up the structure, duplicating its right half which is also moved one cell to the right (time 179). Once the C cell reaches the upper end it travels down the structure, spawns a D cell at the bottom and begins traveling upward while duplicating and moving the right half (time 187). Meanwhile the D cell travels upwards between two halves of the structure and joins them together (time 190).

This process is then repeated. The C cell travels up and down the right side of the structure, creating a duplicate half on its way up. As it reaches the bottom end a D cell is spawned which travels upward between two disjoint halves and joins them together. Since joining two halves occurs every second pass the D cell dies immediately every other pass (e.g. time 195).

There are interesting features to be noted in the process presented. Replication should begin only after the organism is completely formed, i.e. there are two distinct phases of development. However there can be no global indicator that such a situation has occurred (see also Section 3.2). Our solution is therefore local: a B cell upon encountering an upper end which already has one zero completes the formation of that end and releases a C cell which travels down the length of the structure. This cell will seek the bottom end or *wait* for its completion. Only at such time when the structure is complete will the C cell begin the replication process.



(a) Overview of process.

(b) Zoom of intermediate stages. *C* cells are represented by @, *D* cells by ∗.

Figure 8: **Growth and replication.**

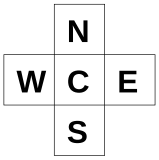


Figure 9: **Cell neighborhood.**

Replication involves two cells operating in unison where the C cell duplicates half of the structure while the D cell "glues" two halves together. Again it is crucial that the whole process be local in nature since no global indicators can be used.

The rules involved in the system are given in Appendix A. The spawning of C and D cells are provided for by the added feature above which specifies that a cell may contain a small number of different rules, where only one is active at a given moment. Therefore, the initial B cells can contain all three rules: B,C,D.

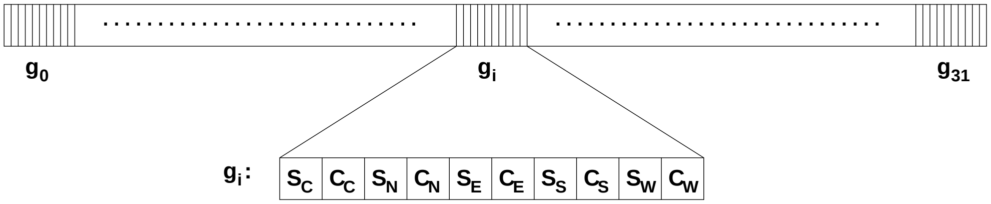
The design of our system is even more efficient than that however, requiring only two rule tables, one for A cells and one for $B/C/D$ cells. Each entry of the $B/C/D$ rule table is only used by one of the cells (i.e. the entries are mutually exclusive). At a given moment the cell has one active rule (which determines its type). If the table entry to be accessed belongs to the active rule- it is used, otherwise a default state change occurs. The default transformation is a move to a random vacant cell for B cells and no change for C and D cells.

4 Evolution

4.1 Evolution in rule space

The previous section presented a number of designed multi-cellular organisms using the model delineated in Section 2. These organisms demonstrate the capability of our model in creating systems of interest. This comes about by increasing the level of operation with respect to the 'physics' level of CA (Section 1). In this section we study evolution as it occurs in our model. Though at this point we have not yet evolved organisms as complex as those of the previous section we have nonetheless encountered several interesting phenomena. We shall also present various tools with which the evolutionary process can be investigated.

The cellular space considered in this section is 2 state, 5-neighbor (Figure 9) where states are denoted $\{0, 1\}$. We chose this space due to practical considerations as well as the desire to study the simplest possible two-dimensional space. Evolution in rule space is achieved by constructing the *genome* of each cell specifying its rule table as depicted in Figure 10. There are 32 genes corresponding to all possible neighborhood configurations. Each gene consists of 10 bits encoding the state change to be effected on neighboring cells (including itself) and whether the rule should be copied to neighboring cells or not (including itself).



g_i - gene i corresponds to neighborhood configuration i, where i is equal to the binary representation of the neighboring cell states in the order: CNESW.

S_X - state change to cell in direction x (0/1).

C_X - copy rule to cell in direction x (0-don't copy, 1-copy).

Figure 10: **Rule genome.**

When discussing specific genes we will use the following symbolism:

$$CNESW \Rightarrow Z_c Z_n Z_e Z_s Z_w$$

where $CNESW$ represents a neighborhood configuration, $Z_c Z_n Z_e Z_s Z_w$ represents the respective S_x and C_x bits using the following notation for Z_x:

	$C_x = 0$	$C_x = 1$
$S_x = 0$	$Z_x =$'0'	$Z_x =$'$-$'
$S_x = 1$	$Z_x =$'1'	$Z_x =$'$+$'

At each time step every operational rule simultaneously executes its appropriate rule entry by referring to the gene corresponding to its current neighborhood states, i.e. state changes and rule copies are effected as delineated in Section 2. This is followed by application of two genetic operators: crossover (re-combination) and mutation. These operators are well known in the context of Genetic Algorithms [Holland, 1975, Goldberg, 1989] in which genomes are also represented as strings (usually binary).

Crossover is performed in the following manner: at each time step every operational cell selects an operational neighbor at random. Let (i, j) denote the grid position of an operational cell and (i_n, j_n) the grid position of the randomly selected operational neighbor. Crossover is performed between the genomes of the rules in cell (i, j) and cell (i_n, j_n), with probability p_{cross}. The (single) crossover site is selected with uniform probability over the entire string and the resulting genome is placed in cell (i, j). If the cell has no operational neighbors then no crossover is effected. Note that the crossover operator is somewhat different than the one used in Genetic Algorithms due to its 'asymmetry': cell (i, j) selects cell (i_n, j_n) while cell (i_n, j_n) may select a different cell, i.e. cell (i', j') such that $(i', j') \neq (i, j)$. It is felt that this slightly decreases the coupling between cells, thus enhancing locality and generality.

Mutation is applied to the genome of each operational rule, after the crossover stage, by inverting each bit with probability p_{mut}. Note that both operations are insensitive to gene boundaries which is also the case in biological settings. In summary, at each time step every operational rule performs its appropriate action, after which crossover and mutation are applied.

It is important to note the difference between our approach and Genetic Algorithms. Though we apply genetic operators in a similar fashion there is no selection mechanism operating on a global level using the total fitness of the entire population. As we shall see (Section 4.3) fitness will be introduced, albeit in a local manner consistent with our model (see also [Collins and Jefferson, 1992]). Note also that in the standard GA model each entity is an independent coding of a problem solution interacting only with the fitness function, never "seeing" the other entities in the population nor the general environment that exists (see also [Ray, 1994a]). In contrast, in our case fitness depends on interactions of evolving organisms operating in an environment.

Taking a 'hardware' point of view, we note that the resources required by our model only slightly exceed those of CA. Since both models are local in nature each cell must retain a copy of the rule in its own memory[2]. Furthermore, the size of our genome is 320 bits as compared to the CA rule which requires 32 bits. Note that in this context rule copying is straightforward requiring only a simple memory transfer. We maintain that on the scale of complexity (Section 1) our enhanced rule is very close to the low end alongside with the CA rule.

4.2 Initial results

Our first experiments were performed by running the model described above using an initial random population of rules. The parameters used are detailed in Table 1.

In this setup the only limitation imposed by the environment is due to the finite size of the grid, i.e. there is competition between rules for occupation of cells. The final grid obtained is one in which most cells are operational (approximately 96%). The rule population consists of different rules with some notable commonalities among them. The average value of the number of $C_c = 1$ bits in the rule genomes is approximately 31. This bit indicates whether the rule should be copied to the cell it occupies in the next time step ($C_c = 1$) or not ($C_c = 0$) and it is observed that almost all such bits in the genomes equal 1. Thus, a simple strategy has emerged which specifies that a rule, upon occupation of a certain cell, remains there, thereby preventing occupation by another rule (which can only enter a vacant cell).

Another commonality observed, among runs, was the average distribution of C_x bits in the genomes of the rules present on the final grid. The percent of C_x bits equaling 1 is 63% and those equaling 0 is 37%. These ratios are approximately $1 - 1/e$ and $1/e$, respectively, and appeared regularly in all simulations. Since the C_x bits in the genome indicate how

[2] Although simulations of CA on serial computers may optimize memory requirements by retaining a single copy of the rule this in no way impairs our 'hardware' argument.

General parameters	time steps	$3000 - 30000$
	grid size	40x50
	p_{cross}	0.9
	p_{mut}	0.001
Initialization parameters	$p_{operational}$	0.5
	$p(S_x = 1)$	0.5
	$p(C_x = 1)$	0.5

p_{cross}: probability of crossover.
p_{mut}: probability of mutation.
$p_{operational}$: probability of cell being operational in initial grid.
$p(S_x = 1)$: probability of S_x bits of genome equaling 1 (state change bits, see Figure 10).
$p(C_x = 1)$: probability of C_x bits of genome equaling 1 (copy rule bits, see Figure 10).

Table 1: **Simulation parameters**

"active" a rule is it is evident that activity is essential for survival, in the context of the simple scenario described. The average percentage of S_x bits in the genomes was equal, i.e. approximately 50% for each bit, indicating no preference for a specific state.

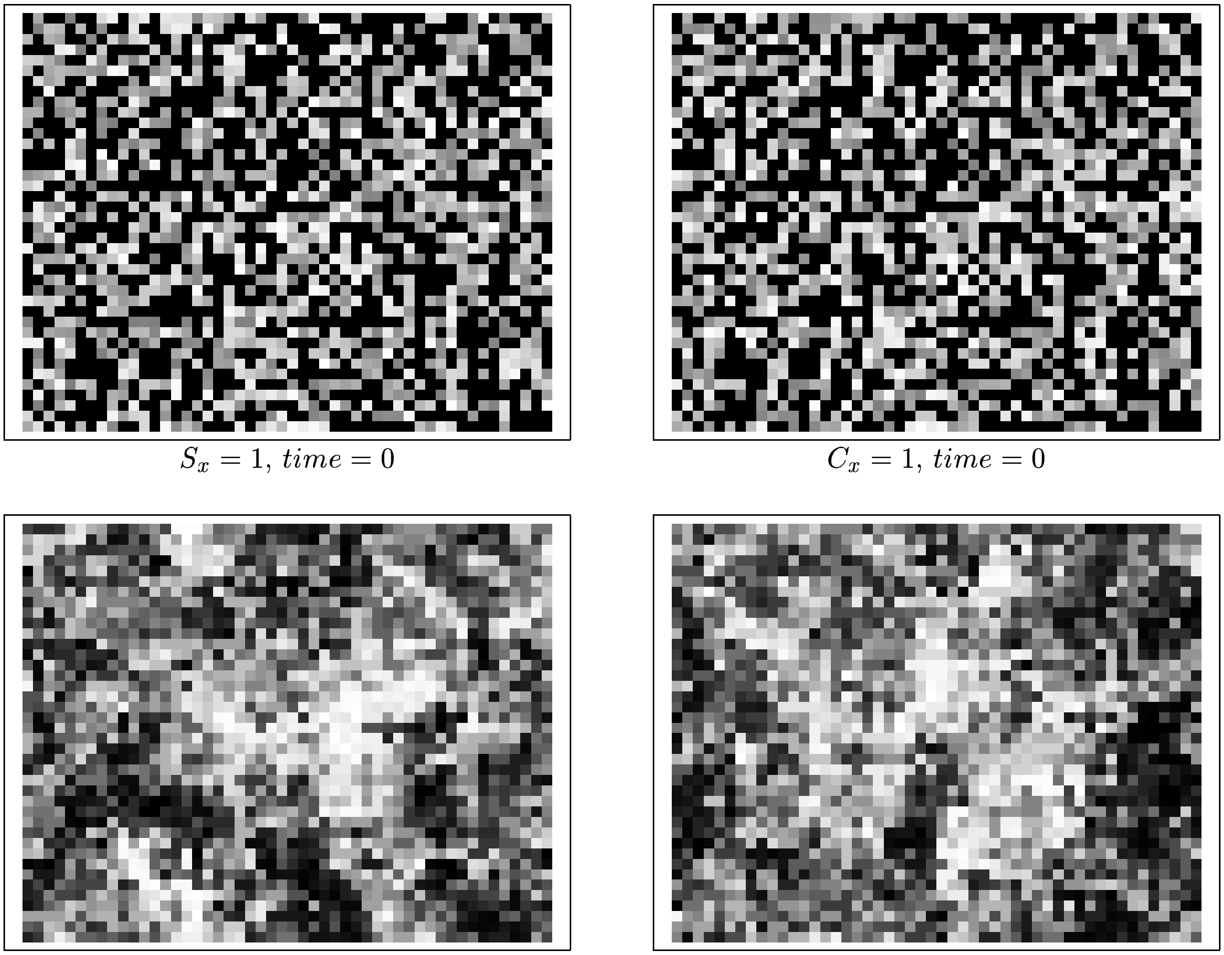
The results described were essentially the same for different values of the parameters in Table 1. One case did, however, prove slightly different, namely $p_{mut} = 0$, i.e. using crossover alone. Here all cells in the final grid were operational with the C_c bits of all genomes equaling 1 (i.e. 32 $C_c = 1$ bits). Thus it is evident that the initial population consists of sufficient genetic material such that perfect survivors can emerge. Mutation in this case hinders survival, however we must bear in mind that the environment is simple and thus there appear to be no local minima which can only be avoided by using mutation. As we shall see ahead this is not the case for more complex environments.

Another interesting phenomena was observed by looking at the $S_x = 1$ and $C_x = 1$ grids. The $S_x = 1$ grid is constructed by computing for each cell the total number of S_x bits which equal 1 for the rule genome in that cell. The $C_x = 1$ grid is constructed analogously for C_x bits. A typical run is presented in Figure 11. It is evident that clusters are formed according to state preference ($S_x = 1$ grid) and according to activity ($C_x = 1$ grid).

A final experiment performed in the context of the scenario described so far was the removal of the constraint that a rule may only copy itself into a vacant cell. When run with $p_{mut} = 0$, i.e. no mutations, one rule remained on the grid occupying all cells (i.e. all cells were operational). This rule is the perfect survivor with all C_x bits in its genome set to 1.

4.3 Fitness in an IPD environment

In this section we enhance our model by adding a measure of a rule's *fitness*, specifying how well it performs in a certain *environment*. The environment explored is defined by the



$S_x = 1$, $time = 0$

$C_x = 1$, $time = 0$

$S_x = 1$, $time = 30000$

$C_x = 1$, $time = 30000$

Figure 11: $S_x = 1$ **and** $C_x = 1$ **grids (see text).**

Iterated Prisoner's Dilemma (IPD), a simple game which has been investigated extensively as a model of the evolution of cooperation. IPD provides a useful framework for studying how cooperation can become established in a situation where short-range maximization of individual utility leads to a collective utility minimum. The game was first explored by [Flood, 1952] (see also [Poundstone, 1992]) and became ubiquitous due to Axelrod's work [Axelrod and Hamilton, 1981, Axelrod, 1984, Axelrod, 1987, Axelrod and Dion, 1988]. These studies involve competition between several *strategies*, which are either fixed at the outset or evolve over time. An evolutionary approach was also taken by [Lindgren, 1992, Lindgren and Nordahl, 1994a] where genomes represent finite memory game strategies with an initial population containing only memory 1 strategies. The memory length is allowed to change through neutral gene duplications and split mutations, after which point mutations are applied which can then give rise to new strategies. Simulations of this model revealed interesting phenomena of evolving strategies in a punctuated equilibria manner [Eldredge and Gould, 1972].

The fact that the physical world has spatial dimensions has also come into play in the investigation of IPD models. A CA approach was applied by [Axelrod, 1984] in which each cell contains a single strategy and simultaneously plays IPD against its neighbors. The cell's score is then compared to its neighbors and the highest scoring strategy is adopted by the cell at the next time step. In this case evolution was carried out with a fixed set of strategies, i.e. without application of genetic operators. In [Nowak and May, 1992] the dynamics of two interacting memoryless strategies were considered: cooperators and defectors (also known in the IPD literature as AllC and AllD). Spatiotemporal chaos was observed when interactions occured on a two-dimensional grid. A spatial evolutionary model was also considered in [Lindgren and Nordahl, 1994b] where the representation of strategies and adaptive moves were identical to those of [Lindgren, 1992].

It is important to note the difference of the above approaches from ours. The models discussed above were explicitly intended to study various aspects of the evolution of cooperation using the IPD model. Thus, *strategies* are the basic units of interaction, whether fixed or evolving over time (e.g. by coding them as genomes and performing genetic operators). In contrast we use IPD to model an *environment* and our basic unit of interaction is the rule discussed in Section 4.1. Our genome does not represent an IPD strategy, but rather a general form of local interaction pertinent to our model. Our intention is to study such interacting cells in various environments, one of which is defined in this section by IPD. Thus, rather than using IPD explicitly in the form of strategies, it is applied implicitly through the environment.

At each time step every operational cell plays IPD with its neighbors where a value of 1 represents cooperation and a value of 0 represents defection. The payoff matrix is as follows (presented for row player):

	Cooperation (1)	Defection (0)
Cooperation (1)	3	0
Defection (0)	5	1

The cell's fitness is computed as the sum of the (four) payoffs, after which the following takes place: each (operational) cell which has an operational neighbor with a higher fitness than its own "dies", i.e. becomes vacant. Crossover and mutation are then carried out as described above with one small difference: the crossover probability p_{cross} is not fixed, but is equal to $(f(i,j) + f(i_n, j_n))/40$, where $f(i,j)$ is the fitness of the cell at position (i,j), $f(i_n, j_n)$ is the fitness of the selected operational neighbor for crossover (see Section 4.1). In summary the (augmented) computational process is as follows: at each time step the grid is updated by rule application, then fitness is evaluated according to IPD after which operational cells with fitter operational neighbors become vacant. Finally, crossover and mutation are applied as explained above.

Simulations revealed the following evolutionary phenomenon which is depicted in Figure 12 (parameters used are those of Table 1, except for p_{cross} computed as discussed above). The figure presents a typical run starting from a random grid ($time = 0$). At $time = 1050$ we observe that approximately half the cells are operational ones in state 0, surrounded by vacant cells in state 1. This configuration, which we term *alternate defection*, is one in which the operational cells attain the maximal fitness (payoff) of 20. However, this is not a stable configuration. At some point in time a small cluster of cooperating operational cells emerges ($time = 1500$) which spreads rapidly throughout the grid ($time = 1650$). The final configuration is one in which most cells are cooperating operational ones with a fitness of 12 ($time = 2400$).

The notion of a cluster of cooperation in a spatial IPD model was discussed in [Axelrod, 1984] (albeit without rule evolution, see above). He used the term "invasion by a cluster", emphasizing that a single cooperating cell does not stand a chance against a world of defectors. As noted above our model is more complex, involving evolutionary mechanisms and a general genome which does not specifically code for IPD strategies. Nonetheless we see that the IPD environment induces cooperation, with a noteworthy transition phenomenon in which widespread defection prevails.

Cooperation is achieved by a multitude of *different* rules, i.e. with different genotypic makeup. Upon inspection of these rules we detected a significant commonality among them found in gene g_{31} which is usually:

$11111 \Rightarrow +++++$

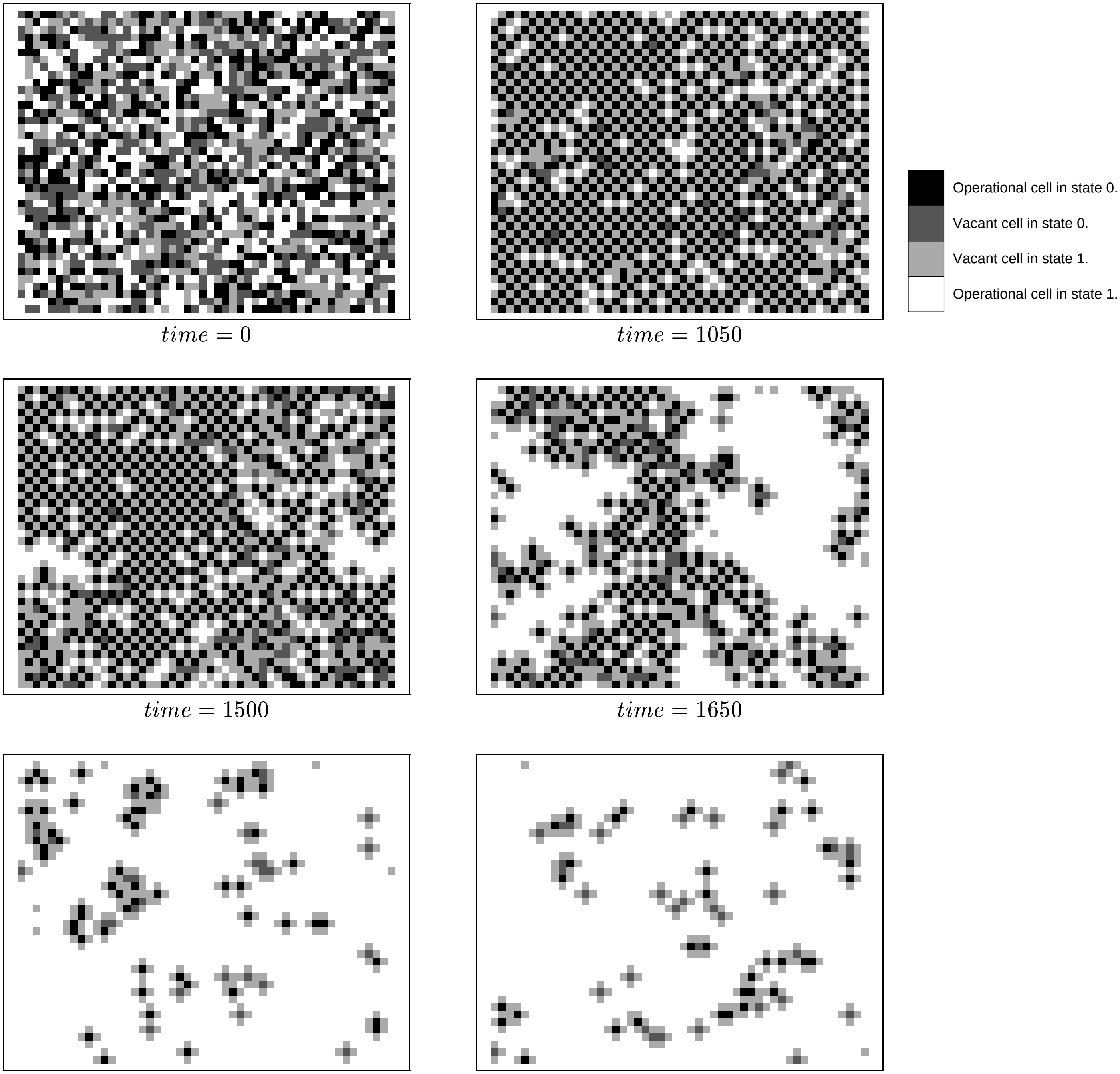
or, in some cases a C_x bit may be 0, where $x \neq c$ (i.e. not the central copy rule bit), e.g.:

$11111 \Rightarrow +++1+$

Thus, we see how cooperation is maintained, by having this gene activated once stability is attained, essentially assuring that the cell remains operational and in state 1 (cooperate) with operational cooperating neighbors. Occasional "cheaters" have been observed, i.e. rules with gene g_{31} such as:

$11111 \Rightarrow -++++$

These are rules which remain operational at the next time step but in a state of defection. However, they are unsuccessful in invading the grid, and we have not observed a return to defection after cooperation has been attained.



$time = 0$

$time = 1050$

$time = 1500$

$time = 1650$

$time = 1800$

$time = 2400$

Figure 12: **Iterated Prisoner's Dilemma.**

It is noteworthy that the final grid consists of rules which essentially employ only one gene of the 32 present in the genome. This may be compared to biological settings where only part of the genome is expressed, while other parts are of no use. Thus, one of the aforementioned features of our model is demonstrated, namely the general coding of cellular rules, as opposed, e.g. to explicit coding of IPD strategies. Evolution take its course, converging to a stable "strategy" consisting of a *multitude* of *different* rules (genomes), whose commonality lies in a specific part of the genome, the part which is *expressed*, i.e. responsible for the *phenotype*. Our rules can be viewed as simple organisms specified by the genome of Figure 10 where evolution determines which genes are expressed and their exact allelic form. We can view this setup as the formation of a sub-species of cooperating organisms, where members are defined by their phenotypic effects, rather than their exact genetic makeup. Whereas the genomes differ greatly (in terms of the precise alleles present), their phenotypes are similar (cooperation) due to a critical gene, g_{31}, which is the one expressed.

When p_{mut} is set to 0, two patterns have been observed to emerge: cooperation or absolute alternate defection (Figure 13). While cooperation is as before, among different rules, absolute alternate defection is achieved with only one surviving rule. Each such run produced a different survivor with an important commonality found in gene g_{15} which is one of the following:

$01111 \Rightarrow + - - - -$

or

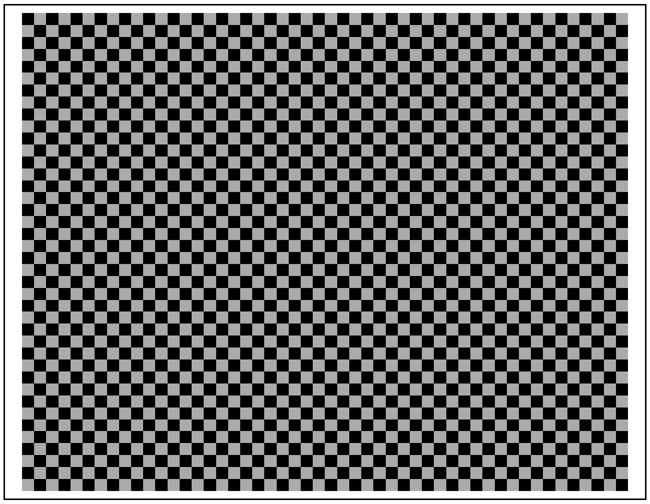
$01111 \Rightarrow 1 - - - -$

Thus, when the grid configuration is such that all operational rules are in state 0 surrounded by vacant cells in state 1, g_{15} is activated causing the current cell's state to become 1 and the rule to be copied into all neighboring cells, with their state changed to 0[3]. This is an interesting strategy in that an operational cell insures cooperation of the cell it occupies and then defects to a neighboring cell. The case of $p_{mut} = 0$ demonstrates the importance of mutation which causes small perturbations that are necessary to invoke cooperation, as opposed to less complex environments where mutation was less useful (Section 4.2).

We next explore the following modification: fitness is allowed to accumulate over a small period of time ($3 - 5$ steps). The death of operational cells still occurs at each time step as before (i.e. when a fitter operational neighbor exists), however, they stand a better chance of survival since their recent fitness histories are taken into account. It was observed that cooperation did not emerge, rather the state attained was that of alternate defection. Thus in a harsher environment, inflicting immediate penalty on unfit cells, cooperation emerges while in a more forgiving environment defection wins.

Cooperation also emerges when the grid is run with a different initial rule population involving only two types of rules: cooperators and defectors. The S_x bits of cooperators are set to 1 while those of defectors are set to 0. The C_x bits are initialized randomly and all

[3] Note that though every vacant cell is contended by four operational neighbors they are all identical and so there is no importance as to who wins. Also note that when the center cell remains operational (as in the first g_{15} gene) it immediately dies since its fitness is 0.



time = 10000

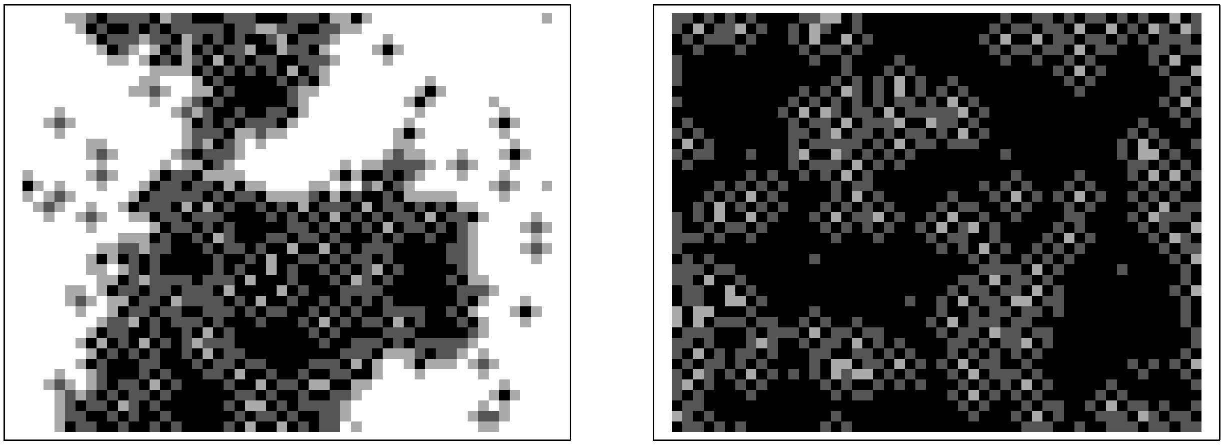
Figure 13: **Iterated Prisoner's Dilemma, $p_{mut} = 0$: absolute alternate defection.**

cells are operational at $time = 0$ (crossover and mutation are effected as above, $p_{mut} > 0$).

Let p_{coop} denote the probability of a cell being a cooperator in the initial grid. When $p_{coop} = 0.9$ we observe that at first there is a "battle" raging on between cooperators and defectors (Figure 14). However, the grid then shifts to alternate defection and finally to cooperation as in Figure 12. When p_{coop} is set to 0.5, i.e. an equal proportion (on average) of cooperators and defectors in the initial population, there is at first an outbreak of defection (Figure 14). Again, however, the grid shifts to alternate defection and then to cooperation. This evolutionary pattern is also observed for $p_{coop} = 0.1$. Thus even when there is a majority of defectors at $time = 0$ cooperation prevails.

4.4 Energy in an environment of niches

In this section we introduce the concept of *energy*, which serves as a measure of an organism's activity, with the intent of enhancing our understanding of phenomena occurring in our model. Each cell is considered to have a finite value of energy units. At each time step energy units are transferred between cells in the following manner: when an operational cell attempts to copy its rule into an adjoining vacant cell an energy unit is transferred to that cell. Thus, an operational cell loses a energy units where a equals the number of $C_x = 1$ bits with x representing a vacant neighbor, i.e. the number of copies the cell attempts to perform (not necessarily successfully since contention may occur, see Section 2). Note that the total amount of energy is conserved since an operational cell's loss is a vacant cell's gain. All cells hold the same amount of energy at the outset and no bounds are set on the possible energy values throughout the run.



$p_{coop} = 0.9$, $time = 150$

$p_{coop} = 0.5$, $time = 150$

Figure 14: **Cooperators and Defectors.**

To study the idea of energy we explore an environment consisting of spatial niches where each cell (i, j) possesses a *niche id* equal to:

$$n_d(i,j) = \lfloor i/10 + j/10 \rfloor \bmod 5$$

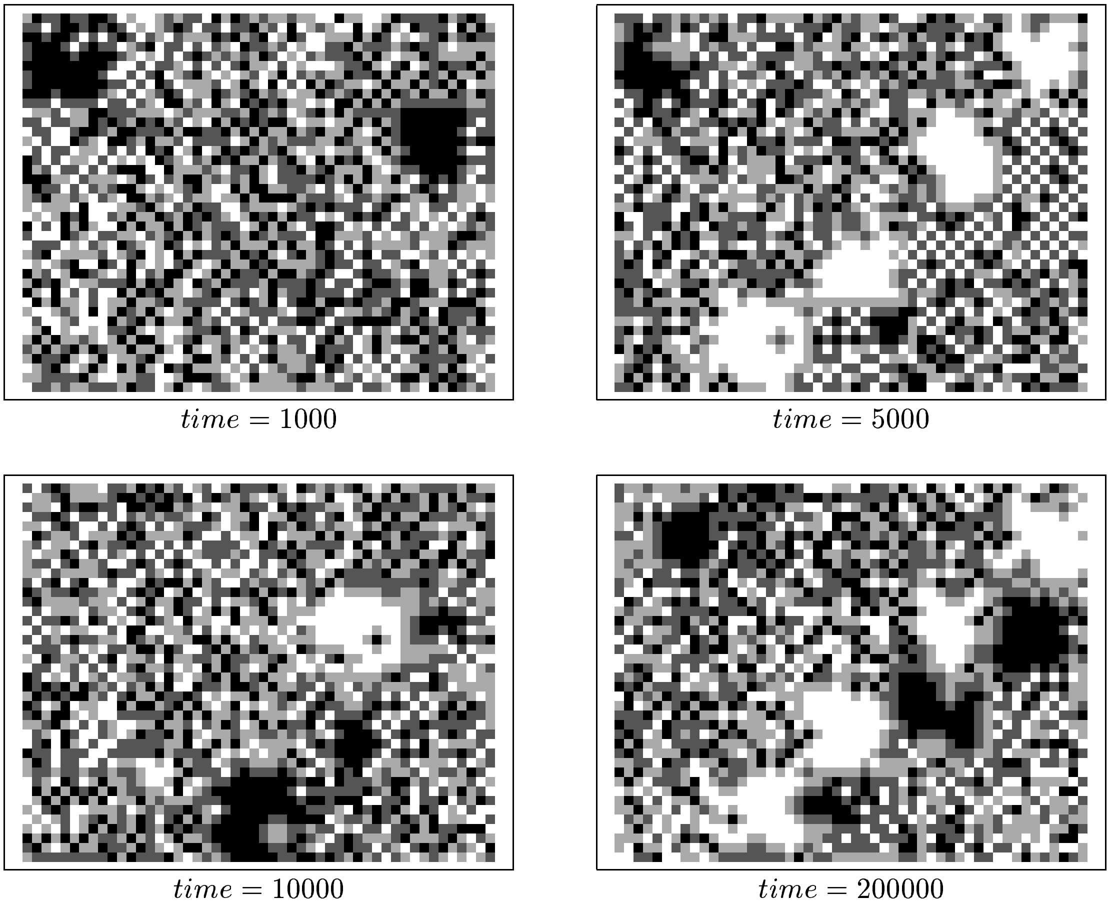
The n_d value indicates the desired number of neighbors in state 1. A cell's fitness, at time t, is defined as:

$$f^t(i,j) = 4 - \mid n_d(i,j) - n_o^t(i,j) \mid$$

where $n_o^t(i,j)$ is the number of adjoining cells in state 1, at time t. As in Section 4.3, p_{cross} is not fixed, but is equal to $(f(i,j)+f(i_n,j_n))/8$, where $f(i_n,j_n)$ is the fitness of the selected operational neighbor. Also, an operational cell with a fitter operational neighbor "dies", i.e. becomes vacant (Section 4.3).

Figure 15 shows the grid at various times and Figure 16 shows the energy map where a darker shade corresponds to lower energy. Observing the grid it is difficult to discern the precise patterns that emerge, however the energy map provides a clear picture of what transpires. At $time = 1000$ we note that boundaries begin to form, evident by the higher energy borders (lighter shades). These correspond to cells positioned in between niches which remain vacant, thus becoming highly energetic. At $time = 5000$ and $time = 10000$ we see that the borders have become more pronounced. Furthermore, regions of low (dark) energy appear corresponding to niches with $n_d = 0, 4$. This indicates that there is a lower degree of activity in these areas, presumably since these niches represent an "easier" environment. At $time = 200000$ the energy map is very smooth indicating uniform activity, with clear borders between niches.

A different environment considered is one of temporal niches, where n_d is a function of time rather than space, i.e. $n_d(t) = \lfloor t/1000 \rfloor \bmod 5$. We generated energy maps at



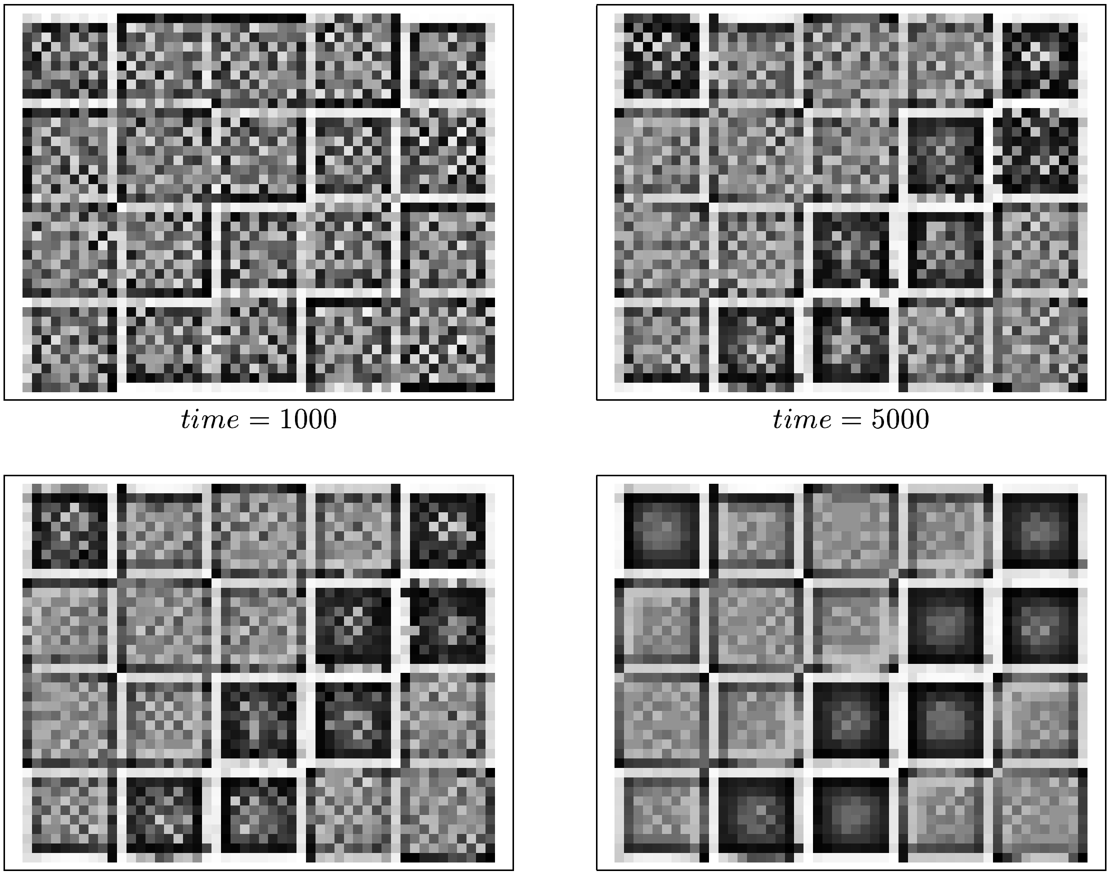
$time = 1000$

$time = 5000$

$time = 10000$

$time = 200000$

Figure 15: **Spatial niches: grid.**



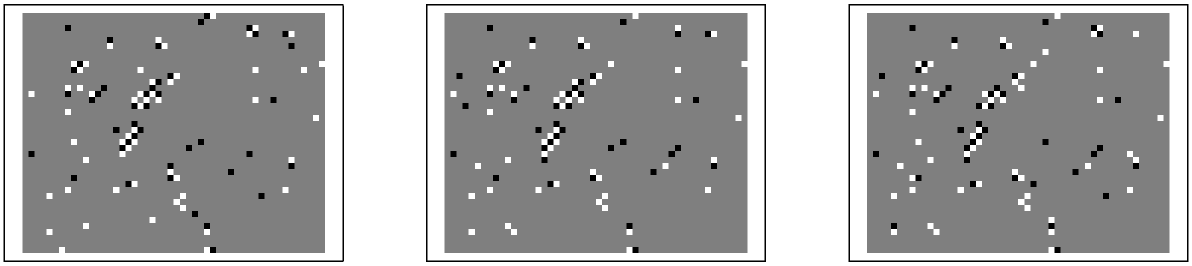
$time = 1000$

$time = 5000$

$time = 10000$

$time = 200000$

Figure 16: **Spatial niches: energy.**



$time = 16000$ $time = 17000$ $time = 18000$

Gray squares represent energy values within 2 standard deviations of the average, white squares represent extreme high values (outside the range), black squares represent extreme low values.

Figure 17: **Temporal niches: energy.**

points in time where niche shifts occur, i.e. $\lfloor t/1000 \rfloor - 1$, and observed an interesting phenomenon. After a few thousand steps the energy pattern stabilizes and the correlation between successive intervals is close to unity. Figure 17 depicts a typical case (for clarity we show a map of deviations from average, though the correlation was computed for the original maps). Thus, there are regions of extensive activity and regions of low activity, which persist through time.

A different aspect of the evolutionary process is considered in Figure 18 which shows the number of operational cells and their average fitness as a function of time. Highest fitness is obtained at temporal niches corresponding to $n_d = 4$ ($time = 5000, 10000, 15000, 20000$). At these points in time there is a drastic change in the environment (n_d shifts from 4 to 0) and we observe that fitness does not usually climb to its maximal value (which is possible for $n_d = 0$). A further observation is the correlation between fitness and operability. We see that fitness rises in exact correlation with the number of operational cells. Thus, the environment is such that more cells can become active (operational) while maintaining high fitness.

Such a situation is not always the case. Consider, for example, the IPD environment of Section 4.3 whose fitness and operability graphs are presented in Figure 19. Here we see that at a certain point in time fitness begins to decline, however the number of operational cells starts rising. This is the shift from alternate defection to cooperation discussed in Section 4.3. We note that in the IPD environment cells cannot all be active, yet maintain the highest fitness. In this case lower fitness is opted for, attaining a higher number of operational cells.

A different version of temporal niches was also studied in which n_d shifts between the values 0 and 4 every 1000 time steps. In some cases we obtained results as depicted in Figure 20, noting that after several thousand time steps adaptation to environmental changes

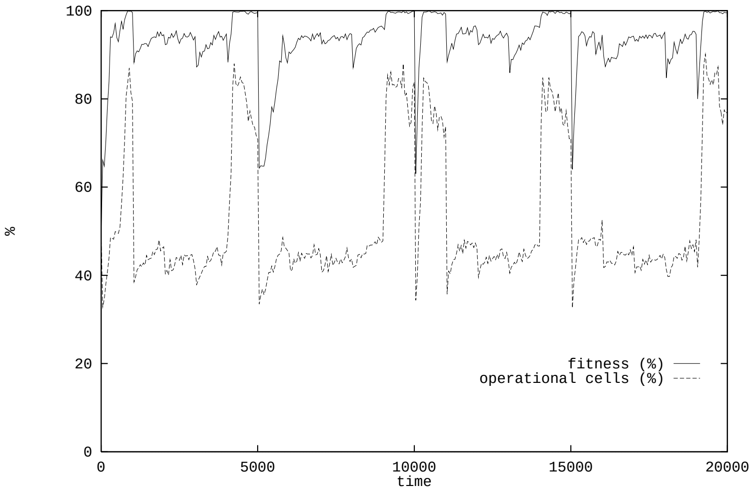


Figure 18: **Temporal niches:** $n_d = 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0 \ldots$

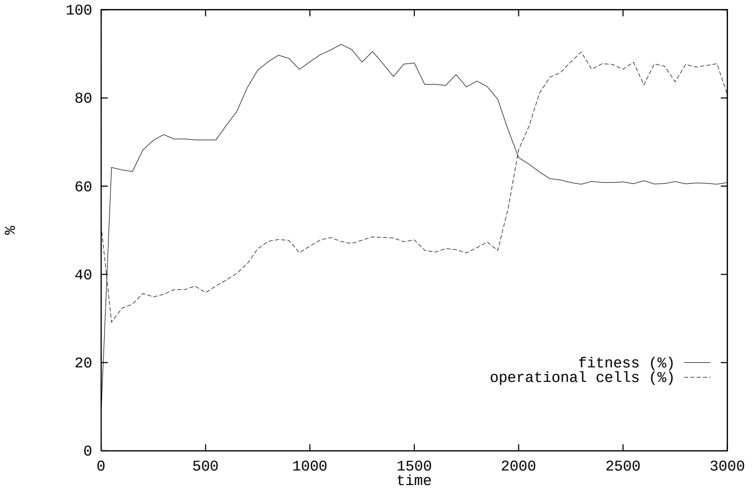


Figure 19: **Iterated Prisoner's Dilemma: fitness, operability.**

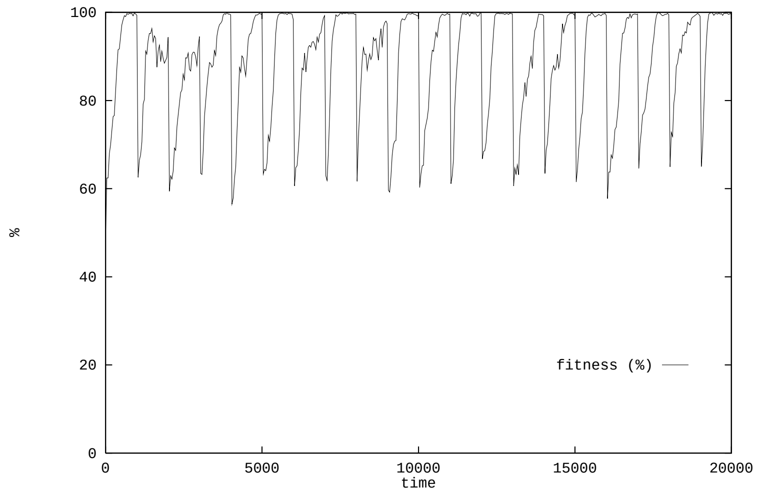


Figure 20: **Temporal niches:** $n_d = 0 \to 4 \to 0 \ldots$

becomes "easier". This could be evidence of *Preadaptation*, a concept which is used to describe the process by which an organ, behavior, neural structure, etc., which evolved to solve *one* set of tasks is later utilized to solve a *different* set of tasks. Though the concept is rooted in the work of [Darwin, 1866] it has recently been elaborated by [Gould, 1982, Gould and Vrba, 1982, Mayr, 1976].

An artificial life approach to preadaptation was taken by [Stork *et al.*, 1992] who investigated an apparent "useless" synapse in the current tailflip circuit of the crayfish, which can be understood as being a vestige from a previous evolutionary epoch in which the circuit was used for swimming instead of flipping (as it is used today). They performed simulations in which the task of the simulated organism is switched from swimming to flipping, and then back to swimming again, observing that adaptation is much more rapid the second time swimming is selected for. This was explained in terms of evolutionary memory in which "junk" genetic information is used [Stork *et al.*, 1992]. Here, "junk", stored for possible future use is contrasted with "trash" which is discarded. Thus, apparent useless information can help regain fitness quickly at some future time when environmental changes occur. In Section 4.5 we examine the genescape, which allows us to directly observe the interplay of genes. Indeed, we note that evolutionary memory can be of use since different genes are responsible for the two niches discussed above ($n_d = 0, 4$).

4.5 The genescape

In their paper [Bedau and Packard, 1992] discuss how to discern whether or not evolution is taking place in an observed system, defining evolutionary activity as the rate at which useful genetic innovations are absorbed into the population. They point out that the rate at which new genes are introduced does not reflect genuine evolutionary activity, for the new genes may be useless. Rather, *persistent usage* of new genes, is the defining characteristic of genuine evolutionary activity.

The model studied by [Bedau and Packard, 1992] is that of strategic bugs in which a bug's genome consists of a look-up table, with an entry for every possible combination of states. They attach to each gene (i.e. each table entry) a "usage counter", which is initialized to zero. Every time a particular table entry is used the corresponding usage counter is incremented. Mutation sets the counter to zero, while during crossover genes are exchanged along with their counters. By keeping track of how many times each gene is invoked, waves of evolutionary activity are observed through a global histogram of gene usage plotted as a function of time. As long as activity waves continue to occur, the population is continually incorporating new genetic material, i.e. evolution is occuring [Bedau and Packard, 1992]. While this measure is extremely difficult to obtain in biological settings, it is easy to do so in artificial ones, providing insight into the evolutionary process.

We have applied the idea of usage counters to our model. Each gene in our genome corresponds to a certain neighborhood configuration (input), specifying the appropriate actions to be performed (output). In this respect it is similar to the strategic bugs model of [Bedau and Packard, 1992] and usage counters are attached to each gene and updated as described above[4]. In [Bedau and Packard, 1992] the usage distribution function is defined, which is then used to derive the $A(t)$ measure of evolutionary activity. Since our genome is small (32 genes) we have opted for a more direct approach in which we study the total usage of each gene throughout the grid as a function of time. This measure is computed by summing the usage counters of all operational cells at a given time. Our measurements can then be presented in a three dimensional plot denoted the *genescape*, meaning the evolutionary genetic landscape.

The genescape of the environment studied in Section 4.2 is shown in Figure 21. Recall that in this case no explicit environmental constraints are placed and the only (implicit) one is therefore due to the finite size of the grid, i.e. there is competition between rules for occupation of cells. The genescape shows that usage is approximately constant (after an initial rise due to an increase in the number of operational cells) and uniform. No gene is preferred since the environment is such that all contribute equally to fitness. The constant usage count is consistent with our parameters (p_{cross} and p_{mut}). This situation may be considered as a "flat" genescape serving as a baseline for comparison with other environments[5].

Figure 22 shows the genescape of the IPD environment (Section 4.3). We observe that gene g_{15} initially comes to dominate, later to be overtaken by g_{31}, representing the shift from

[4] There is one small difference: In the model of [Bedau and Packard, 1992] crossover does not occur across gene boundaries and therefore does not set the respective counter to zero, whereas in our model crossover can occur anywhere along the genome. Thus, a counter is reset whenever crossover occurs within its gene (as well as when the gene mutates).

[5] Note that other parameters did reveal interesting phenomena even for this simple environment, as noted in Section 4.2.

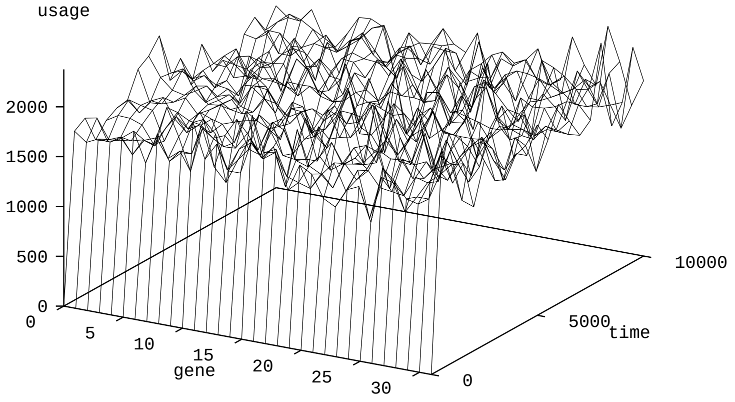


Figure 21: **Genescape: no environmental constraints.**

alternate defection to cooperation. Smaller peaks are also apparent, coexisting alongside g_{15}. These occur for genes, g_i such that $i < 15$, i.e. those genes representing a central cell state of defection (0). Thus the dominance of g_{15} is not totalistic as is later the case with g_{31}. This gene, g_{31}, shows a small usage peak from the start, essentially biding its time until the "right" moment comes, when cooperation breaks through. This is reminiscent of punctuated equilibria results, where phenotypic effects are not observed for long periods of time, while evolution runs its course in the (unobserved) genotype.

The genescapes of the temporal niches environments of Section 4.4 are presented in Figures 23 and 24. Observing Figure 23a, we note how usage peaks shift from g_0 (for niche id $n_d = 0$) to g_{31} (for $n_d = 4$) as time progresses. Closer inspection provides us with more insight into the evolutionary process (Figure 23b). It is noted that gene g_{16} competes with g_0 when $n_d = 0$ and g_{15} competes with g_{31} when $n_d = 4$, with g_0 and g_{31} predominating eventually. This competition is explained by the fact that n_d specifies the desired number of neighbors in state 1, without placing any restriction on the central cell, thus promoting competition between two genes where one eventually emerges as the "winner".

When intermediate n_d values are in effect ($n_d = 1, 2, 3$) we observe multiple peaks corresponding to those genes representing the appropriate number of neighbors (Figure 23b). As the environment changes (through n_d) different *epistatic* effects are introduced. The lowest degree of epistasis occurs when $n_d = 0, 4$ and the highest when $n_d = 2$. It is interesting to compare these results with those obtained by [Kauffman and Weinberger,

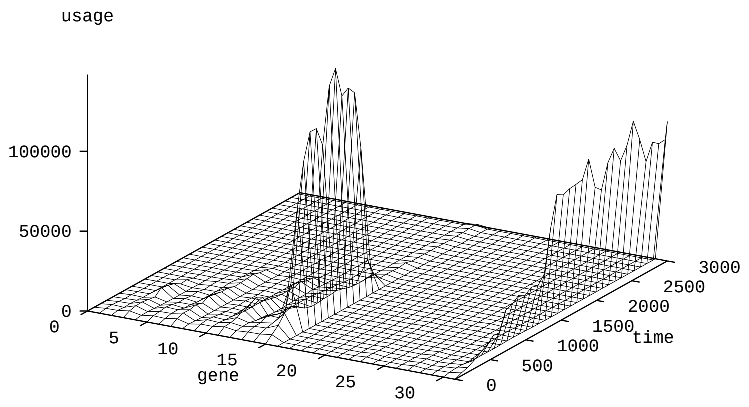


Figure 22: **Genescape: IPD.**

1989, Kauffman and Johnsen, 1992] in which the NK model is employed. The NK model describes genotype fitness landscapes engendered by arbitrarily complex epistatic couplings. An organism's genotype consists of N genes, each with A alleles. The fitness contribution of each gene depends upon itself and epistatically on K other genes. The central idea used in the NK model is that the epistatic effects of the A^K different combinations of A alternative states of the other K genes on the functional contribution of the Ath state of each gene are so complex that their statistical features can be captured by assigning fitness contributions at random from a specified distribution. Tuning K from low to high increases the epistatic linkages thus providing a tunable rugged family of model fitness landscapes.

The main conclusions offered by [Kauffman and Weinberger, 1989, Kauffman and Johnsen, 1992] are that as K increases relative to N (i.e. as epistatic linkages increase) the ruggedness of the fitness landscapes increases by a rise in the number of fitness peaks, while the typical heights of these peaks decrease. The decrease reflects the conflicting constraints which arise when epistatic linkages increase. In the NK model epistatic linkages are made explicit using the K parameter with fitness contributions assigned randomly. We have presented an environment in which the n_d (niche) value changes, thereby causing *implicit* changes in the degree of epistasis. Essentially, $K = 1$ for $n_d = 0, 4$, $K = 7$ for $n_d = 1, 3$ and $K = 11$ for $n_d = 2$. Our usage results of Figure 23 correspond with the conclusions offered by [Kauffman and Weinberger, 1989, Kauffman and Johnsen, 1992]. As K increases the number of usage peaks increase while their heights decrease. Note that we do not measure fitness as in the NK model but rather usage, which can be regarded as a more "raw" measure. Also, fitness contributions are not made explicit but rather are implicitly induced by the environment. Although our viewpoint is different the results obtained are analogous, enhancing our understanding of epistatic environmental effects.

4.6 Synchrony verses asynchrony

One of the prominent features of the CA model is its synchronous mode of operation meaning that all cells are updated simultaneously at each time step. Recently, it has been observed that when asynchronous updating is used (i.e. one cell is updated at each time step) results may be different. For example, in [Huberman and Glance, 1993] it was shown that when asynchrony is introduced in the model of [Nowak and May, 1992] (see Section 4.3) a fixed state is arrived at rather than the chaotic spatiotemporal behavior induced by the synchronous model. Asynchrony has also been shown to "freeze" the game of life, i.e. convergence to a fixed point occurs, rather than complex, class IV phenomena of the synchronous model [Bersini and Detour, 1994] .

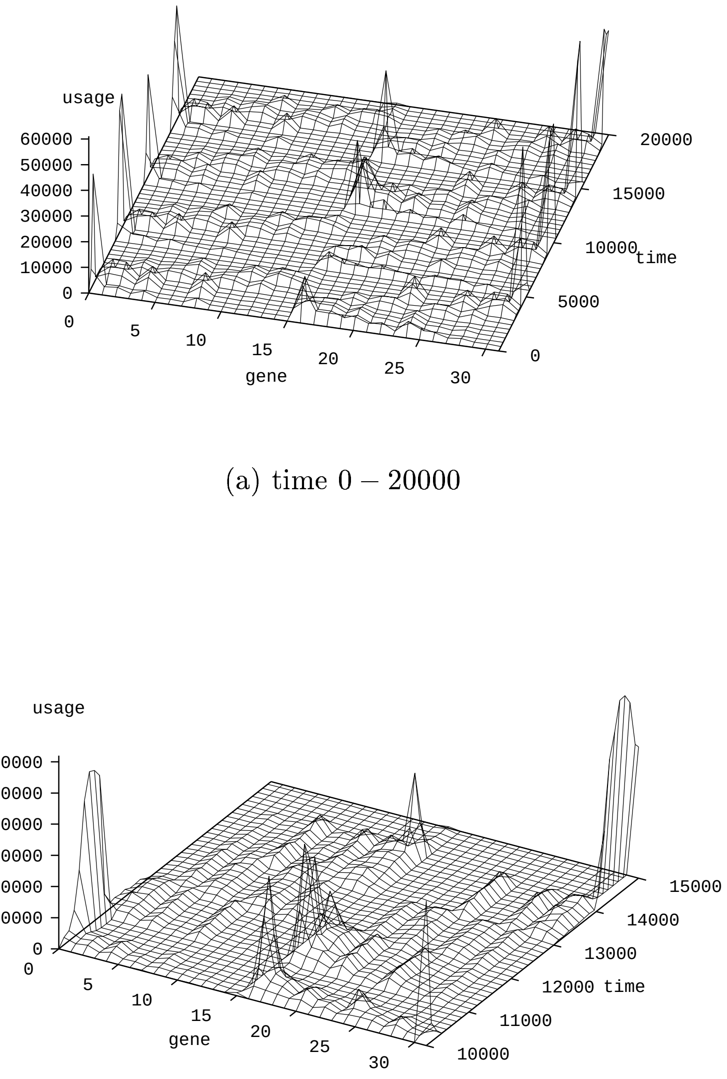
The issue raised by these investigations (see also [Lumer and Nicolis, 1994]) is the relevance of results obtained by CA models to biological phenomena. Indeed [Huberman and Glance, 1993] have argued that patterns and regularities observed in nature require asynchronous updating since natural systems posses no global clock. It may be argued that from a physical point of view synchrony is justified: since we model a continuous spatial and temporal world we must examine each spatial location at every time step, no matter how small we choose these (discrete) steps to be. However, as we move up the scale of complexity of the basic units, synchrony seems to be less justified. For example, IPD is usually aimed at investigating social cooperation where the basic units of interaction are complex organisms (e.g. humans, societies).

The simulations described in the previous sections were conducted using synchronous updating. Due to the arguments raised above we were motivated to investigate the issue of asynchrony by repeating some of our simulations using asynchronous updating. Results obtained were different than for synchronous updating, e.g. the asynchronous runs of the IPD environment (Section 4.3) produced no "interesting" configurations as for the synchronous case.

We then experimented with two forms of *partial* asynchrony: (1) *sparse updating*: at each time step a cell is updated with probability p_{sparse}, and (2) *regional updating*: at each time step a fixed size, square region of the grid is updated. Sparse updating produced "uninteresting" results, i.e. as in the asynchronous case. However, with regional updating we observed that the synchronous updating results were repeated, provided the region size exceeded a certain value, which is about 100 cells (i.e. a 10x10 square).

It is noteworthy that sparse updating did not "work" even for high values of p_{sparse} (e.g. 0.2) while regional updating produced results identical to the synchronous case[6]. We also

[6] Note that a region size of 10x10 is equivalent (on average) in terms of the number of cells updated per time step to $p_{sparse} = 0.05$ for a 40x50 grid.



(a) time $0 - 20000$

(b) zoom of time $10000 - 15000$

Figure 23: **Genescape: temporal niches,** $n_d = 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0 \ldots$

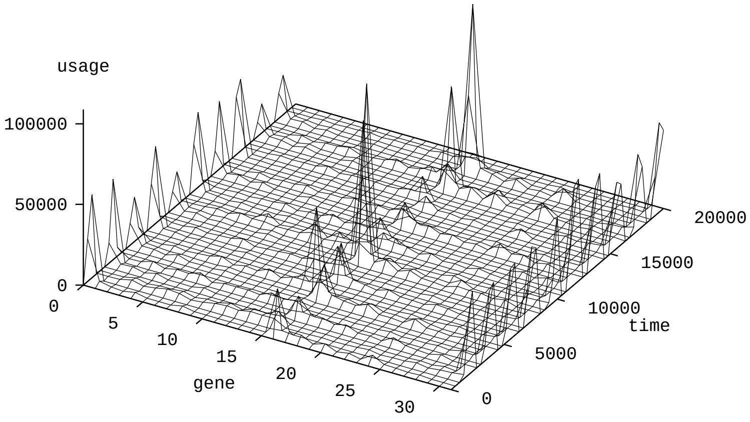


Figure 24: **Genescape: temporal niches,** $n_d = 0 \rightarrow 4 \rightarrow 0 \ldots$

experimented with larger grids and obtained the same results without increasing the region size (i.e. about 10x10). While it cannot be ascertained that this size is constant, it seems safe to conjecture that it grows sub-linearly with grid size.

The regional updating method, though not completely asynchronous is nonetheless interesting, especially since the region size seems to grow sub-linearly with grid size. From a 'hardware' point of view this is encouraging since implementations can be made easier by using local (regional) synchronization rather than global, thereby facilitating scaling. We noted that a minimal amount of activity must simultaneously take place in order for "interesting" patterns to emerge, i.e. there is certain threshold of interaction. The crucial factor here is not the total number of cells updated per time step, but rather the simultaneous activity of a (small) area. This is evident by the failure of the sparse updating method verses the success of regional updating. The importance of "regions" of evolution has also been noted in biological settings [Mayr, 1976, Eldredge and Gould, 1972].

The issue of synchrony verses asynchrony in spatially distributed systems is still an open question. For example, in a recent paper asynchronous simulations were carried out revealing chaotic spatial organization [Lindgren and Nordahl, 1994b], results which were contrasted with those of [Huberman and Glance, 1993]. Our model may yet reveal interesting phenomena for the case of complete asynchrony when other types of environments are employed. At present we have a strong case for partial asynchrony in the form of regional updating, which, due to the small region size, is close to complete asynchrony.

5 Discussion

In this paper we presented a system of simple organisms interacting in a two-dimensional environment, which have the capacity to evolve. We first turned our attention to designed multi-cellular organisms displaying several interesting behaviors. These included: a self-reproducing loop, replication of passive structures by copier cells, mobile organisms, two-phased growth and replication. These organisms offered motivation as to the power of our model in creating systems of interest. This comes about by increasing the level of operation with respect to the 'physics' level of CA (Section 1).

A related work is that of *embryonics*, standing for embryological electronics [Mange and Stauffer, 1994, Marchal *et al.*, 1994, Durand *et al.*, 1994]. This is a CA based approach in which three principles of natural organization are employed: multi-cellular organization, cellular differentiation and cellular division. Their intent is to create an architecture which is complex enough for (quasi) universal computation yet simple enough for physical implementation. The approach represents another attempt at confronting the aforementioned problem of CA, namely the low level of operation.

An important distinction made by the embryonics group is the difference between uni-cellular and multi-cellular organisms. In biological terms a cell can be defined as the smallest part of a living being which carries the complete plan of the being, that is its genome [Mange and Stauffer, 1994]. In this respect the self-reproducing automata of [von Neumann, 1966] and [Langton, 1984] are uni-cellular organisms: the genome is contained within the entire configuration. An important common point between both the embryonics approach and ours is that true multi-cellular organisms are formed: our cell is equivalent to a biological cell in the sense that it contains the complete genome (rule table). A creature in our model consists of several cells operating in unison, thereby achieving the effect of a single "purposeful" organism. It is interesting to compare Langton's self-reproducing loop which is uni-cellular with ours (Section 3.1) which is multi-cellular. This illustrates our concept of raising the level of operation: Langton's loop demonstrates how uni-cellular replication can be attained whereas our loop starts from there and goes on to achieve multi-cellular replication. In this strict sense our model may be viewed as a kind of 'macro' CA consisting of higher level basic operations. We also observe in our model that each cell acts according to a specific gene (entry), which is a simple form of locally-based cellular differentiation. Such approaches offer us new paths in the development of complex machines as collections of simpler cells. Such machines can be made to display an array of biological phenomena, including: self-repair, self-reproduction, growth and evolution [Mange and Stauffer, 1994].

After our initial investigation of multi-cellularity we turned our attention to evolution in rule space which occurs through changes in the genotypes representing the rules by which the organisms operate. At first we placed no explicit environmental constraints, thereby retaining only the implicit constraint due to the finite size of the grid. We observed that a simple strategy emerged in which an organism (as defined by its rule) "sits tight" upon occupation of a certain cell. We can view this as the formation of simple replicators which replicate within their own cell (at each time step) as well as into (possibly) vacant cells. It was also noted that rules tend to self-organize (spatially) in accordance with their levels of activity (C_x bits) and state preferences (S_x bits). These results are interesting, demonstrating that even a very simple environment, with but one constraining factor, is sufficient in order to guide the evolutionary process through regular spatiotemporal patterns.

The IPD environment revealed several interesting phenomena. The evolutionary path taken passes through a state of alternate defection, in which approximately half of the cells are operational, attaining a maximal fitness. However, this is not a stable configuration, since a small cluster of cooperation eventually emerges, taking over most of the grid.

One of our observations concerns the importance of mutation in complex environments. In the simple environment of Section 4.2 mutation proved to be a hindrance, preventing the evolution of perfect survivors. However, as environments grew more complex, mutation became a crucial factor. For example, in the IPD environment defection can prevail when the mutation rate is set to zero, however cooperation always emerges when this rate is small, yet non-zero. It seems that mutation is necessary to help the evolutionary process from getting stuck in local minima (see also [Goldberg, 1989]).

The emergence of cooperation depends not only on the mutation operator but also on the harshness of the environment. When the environment is more forgiving cooperation does not necessarily emerge and defection may prevail, whereas in a harsher environment defection always "steps down" in favor of cooperation. This may have implications to real-life situations in which survival in a harsher environment requires more cooperation.

As discussed above (Section 4.3) our IPD environment is different than other IPD models in that our genome is general and does not code for specific actions, e.g. strategies. Cooperation emerges between a multitude of different organisms, whose commonality lies in the *expression* of a specific gene, a situation which may be regarded as the formation of a sub-species.

One of the advantages of AL models is the opportunities they offer in performing in-depth studies of the evolutionary process. This was accomplished in our case by observing not only phenotypic effects (i.e. evolution of cell states as a function of time) but also fitness, operability, energy and the genescape. The energy concept was introduced as a measure of an organism's activity, where each rule copy costs one unit of energy. We applied this measure in environments consisting of spatial and temporal niches. For the case of spatial niches we observed the difficulty in discerning phenotypic effects (the grid), whereas the energy map provided us with a clear picture of the evolutionary process: regions of higher and lower activity, with high energy boundaries between them. The environment of temporal niches presented us with an interesting phenomenon in which adaptation takes place (as evident by taking note of the fitness graph), with small clusters of extreme energetic activity forming regularly.

An additional measure introduced is the genescape, which depicts the incorporation of new genetic material into the population. The epistatic interplay of genes is highlighted by studying such plots. In the IPD case we noted that the transition from alternate defection to cooperation occurs through a shift from one gene (g_{15}) to another (g_{31}). It was observed that while the phenotypic effect of g_{31} occurs only after several hundred time steps it is constantly evolving, albeit at a low (dormant) rate of activity. This provides us with insight on punctuated equilibria phenomena, which may be partly explained by the difference between observed effects (phenotypes, e.g. fossil record), and unobserved effects (genotypes).

As the environment changes through time (temporal niches) organisms adapt by traversing their adaptive landscapes. By studying the genescape we were able to observe the subtle interplay of epistatic couplings, noting shifts from single-peaked to multi-peaked, rugged terrains. Thus we gain a deeper understanding than is possible by observing only the grid, i.e. phenotypic effects.

A tentative analogy may be put forward, between our organism and the hypothetical, now extinct, RNA organism [Joyce, 1989]. These were presumably simple RNA molecules capable of catalyzing their own replication. What both types of organisms have in common is that a single molecule constitutes the body plus the genetic information, and effects the replication. The inherent locality and parallelism of our model add credence to such an analogy by offering closer adherence to nature. However, we must bear in mind that only a superficial comparison may be drawn at this stage since our model is highly abstracted in relation to nature and has been implemented only for an extremely small number of "molecules". Further investigations along this line using artificial life models may enhance our understanding of the RNA world theory. The analogy between RNA organisms and other types of digital organisms has been noted in [Ray, 1994a].

In Section 1 we delineated two basic guidelines, generality and simplicity, which served us in the definition of our model. In their paper [Jefferson *et al.*, 1992] present a number of important properties a programming paradigm must have to be suitable as a representation for organisms in biologically motivated studies. We discuss these below in light of our model:

1. *Computational completeness.*, i.e. Turing machine equivalence. Since our model is an enhancement of the CA model this property holds. We also noted that from a 'hardware' point of view the resources required by our model only slightly exceed those of CA (Section 4.1).

2. A *simple, uniform model of computation.* This is essentially what we referred to as simplicity (of basic units) and generality (the second meaning, i.e. general encoding, see Section 1). This property is intended to prevent the system from being biased towards a particular environment.

3. *Syntactical closure* of genetic operators. In our case all genomes represent a legal rule table encoding. This property also enables us to start with a random population, thereby avoiding bias.

4. The paradigm should be *well conditioned* under genetic operators. This requirement is less formal meaning that evolution between successive time steps is usually "well behaved", i.e. discontinuities occur only occasionally. This property can be assessed using the genescape.

5. *One time unit* of an organism's life must be specified. In our case time is discrete, with an organism accepting input (neighborhood states) and taking action (output) in a single time step.

6. *Scalability*. This property must be examined with some care. If we wish to add sensory apparatus (in our case increase the neighborhood and/or the number of states) then the genome grows exponentially since it encodes a finite state automaton table. However, complexity can increase through the interaction of several organisms. Indeed, a central goal of AL research is the evolution of multi-cellular creatures. As noted above such organisms are parallel devices, composed of simple basic units and may therefore scale very well. At this point we have demonstrated that multi-cellularity can be attained, albeit by design (Section 3). Scalability is also related to the issue of asynchrony which was discussed in Section 4.6.

The model presented in this paper provides insight into issues involving adaptation and evolution. There are still, however, many limitations that should be addressed. We have modeled an environment in the strict sense, i.e. excluding the organisms themselves (Section 1). Although we achieved an environment in the broad sense, i.e. a total system of interacting organisms, the dichotomy between organisms and their environment is still a major obstacle to overcome [Jefferson *et al.*, 1992] (see also [Bonabeau and Theraulaz, 1994]). Another central issue discussed above is the formation (evolution) of multi-cellular organisms. It is clear that much more research is needed in this direction.

The evolutionary studies we performed were carried out in rather small grids (consisting of only a few thousand cells). It seems reasonable to assume that in order to evolve "interesting" creatures a larger number of units is required. Models such as ours which consist of simple, locally connected units lend themselves to scaling through the use of parallel or distributed implementations. For example, Ray has recently suggested creating a network-wide reserve for the digital Tierra creatures [Ray, 1994b]. He hopes that by increasing the scale of the system by several orders of magnitude, new phenomena may arise that have not been observed in the smaller scale systems.

It is hoped that the development of such AL models will serve the two-fold goal of: (1) increasing our understanding of biology and (2) enhancing our understanding of artificial models, thereby providing us with the ability to improve their performance. AL research opens new doors providing us with novel opportunities to explore issues such as adaptation, evolution and emergence which are central both in natural environments as well as man-made ones.

Acknowledgments

I am grateful to Daniel Sipper for his suggestions which helped improve this paper. Many thanks go to Daniel Mange, Eduardo Sanchez and Pierre Marchal for our fruitful discussions and for their general encouragement.

A Growth and replication: specification of rule

A-cell

Formation of ones:

$$\begin{array}{|c|c|c|}\hline & 1 & 1 \\ \hline & \mathrm{A} & \\ \hline & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & 1 & 1 \\ \hline & 1 & \\ \hline & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline & & 1 \\ \hline & \mathrm{A} & 1 \\ \hline & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & & 1 \\ \hline & 1 & 1 \\ \hline & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline & & \\ \hline & \mathrm{A} & \\ \hline 1 & 1 & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & & \\ \hline & 1 & \\ \hline 1 & 1 & \\ \hline\end{array}$$

$$\begin{array}{|c|c|c|}\hline & & \\ \hline 1 & \mathrm{A} & \\ \hline 1 & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & & \\ \hline 1 & 1 & \\ \hline 1 & & \\ \hline\end{array}$$

B-cell

Formation of zeros:

$$\begin{array}{|c|c|c|}\hline & 1 & 1 \\ \hline & \mathrm{B} & \\ \hline & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & 1 & 1 \\ \hline & 0 & \\ \hline & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline & & \\ \hline & \mathrm{B} & \\ \hline 1 & 1 & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & & \\ \hline & 0 & \\ \hline 1 & 1 & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline & & 1 \\ \hline & \mathrm{B} & 0 \\ \hline & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & & 1 \\ \hline & 0 & 0 \\ \hline & & \\ \hline\end{array}$$

Formation of zero and spawning of *C*-cell:

$$\begin{array}{|c|c|c|}\hline & & \\ \hline 0 & \mathrm{B} & \\ \hline 1 & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & & \\ \hline 0 & 0 & \\ \hline 1 & \mathrm{C} & \\ \hline\end{array}$$

C-cell

Downward movement:

$$\begin{array}{|c|c|c|}\hline 0 & 0 & \\ \hline 1 & \mathrm{C} & \\ \hline & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline 0 & 0 & \\ \hline 1 & & \\ \hline \mathrm{C} & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline 1 & 1 & \\ \hline 1 & \mathrm{C} & \\ \hline & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline 1 & 1 & \\ \hline 1 & & \\ \hline \mathrm{C} & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline 1 & 1 & \\ \hline 0 & \mathrm{C} & \\ \hline & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline 1 & 1 & \\ \hline 0 & & \\ \hline \mathrm{C} & & \\ \hline\end{array}$$

Beginning of upward replication movement and spawning of *D*-cell:

$$\begin{array}{|c|c|c|}\hline 0 & 0 & \\ \hline & \mathrm{C} & \\ \hline & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline 0 & 0 & \mathrm{C} \\ \hline \mathrm{D} & 0 & \\ \hline & & \\ \hline\end{array}$$

Upward replication movement and transfer of one position to the right:

$$\begin{array}{|c|c|c|}\hline 1 & 1 & \\ \hline 0 & \mathrm{C} & \\ \hline 0 & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline 1 & 1 & \mathrm{C} \\ \hline & 0 & 0 \\ \hline & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline 1 & 1 & \\ \hline 1 & \mathrm{C} & \\ \hline 0 & 0 & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline 1 & 1 & \mathrm{C} \\ \hline & 1 & 1 \\ \hline 0 & 0 & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline 1 & 1 & \\ \hline 1 & \mathrm{C} & \\ \hline 1 & 1 & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline 1 & 1 & \mathrm{C} \\ \hline & 1 & 1 \\ \hline 1 & 1 & \\ \hline\end{array}$$

$$\begin{array}{|c|c|c|}\hline 0 & 0 & \\ \hline 1 & \mathrm{C} & \\ \hline 1 & 1 & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline 0 & 0 & \mathrm{C} \\ \hline & 1 & 1 \\ \hline 1 & 1 & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline 0 & 0 & \\ \hline 1 & \mathrm{C} & \\ \hline 0 & 0 & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline 0 & 0 & \mathrm{C} \\ \hline & 1 & 1 \\ \hline 0 & 0 & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline & & \\ \hline 0 & \mathrm{C} & \\ \hline 1 & 1 & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & & \mathrm{C} \\ \hline & 0 & 0 \\ \hline 1 & 1 & \\ \hline\end{array}$$

End of upward replication movement:

$$\begin{array}{|c|c|c|}\hline & & \\ \hline & \mathrm{C} & \\ \hline 0 & 0 & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline 0 & \mathrm{B} & \\ \hline\end{array}$$

D-cell

Move to bottom left-hand side of structure (start position):

$$\begin{array}{|c|c|c|}\hline & 0 & 0 \\ \hline & \mathrm{D} & 0 \\ \hline & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline \mathrm{D} & 0 & 0 \\ \hline & & 0 \\ \hline & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline & 0 & \\ \hline & \mathrm{D} & \\ \hline & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline \mathrm{D} & 0 & \\ \hline & & \\ \hline & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline & 0 & 0 \\ \hline & \mathrm{D} & \\ \hline & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline \mathrm{D} & 0 & 0 \\ \hline & & \\ \hline & & \\ \hline\end{array}$$

Imediate death in case two half structures do not exist:

$$\begin{array}{|c|c|c|}\hline & & \\ \hline & \mathrm{D} & 0 \\ \hline & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & & \\ \hline & & 0 \\ \hline & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline 1 & & \\ \hline & \mathrm{D} & 0 \\ \hline & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline 1 & & \\ \hline & & 0 \\ \hline & & \\ \hline\end{array}$$

Upward replication movement:

$$\begin{array}{|c|c|c|}\hline & 1 & \\ \hline 0 & \mathrm{D} & 0 \\ \hline & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & 1 & \mathrm{D} \\ \hline 0 & 0 & \\ \hline & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline & 1 & \\ \hline 1 & \mathrm{D} & 1 \\ \hline 0 & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & 1 & \mathrm{D} \\ \hline 1 & 1 & \\ \hline 0 & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline & 1 & \\ \hline 1 & \mathrm{D} & 1 \\ \hline 1 & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & 1 & \mathrm{D} \\ \hline 1 & 1 & \\ \hline 1 & & \\ \hline\end{array}$$

$$\begin{array}{|c|c|c|}\hline & 0 & \\ \hline 1 & \mathrm{D} & 1 \\ \hline 1 & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & 0 & \mathrm{D} \\ \hline 1 & 1 & \\ \hline 1 & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline & 0 & \\ \hline 1 & \mathrm{D} & 1 \\ \hline 0 & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & 0 & \mathrm{D} \\ \hline 1 & 1 & \\ \hline 0 & & \\ \hline\end{array} \qquad \begin{array}{|c|c|c|}\hline & & \\ \hline 0 & \mathrm{D} & 0 \\ \hline 1 & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & & \mathrm{D} \\ \hline 0 & 0 & \\ \hline 1 & & \\ \hline\end{array}$$

Death after completion of upward movement:

$$\begin{array}{|c|c|c|}\hline & & \\ \hline & \mathrm{D} & \\ \hline 0 & & \\ \hline\end{array} \to \begin{array}{|c|c|c|}\hline & & \\ \hline & & \\ \hline 0 & & \\ \hline\end{array}$$

Note that an *A* cell dies after attaching a one to the structure, a *B* cell either dies or spawns a *C* cell after attachment of zero. All other entries of *A* and *B* cell rules specify a move to a random vacant cell while those for *C* and *D* rules specify no change.

References

[Axelrod and Dion, 1988] R. Axelrod and E. Dion. The further evolution of cooperation. *Science*, 242:1385–1390, 1988.

[Axelrod and Hamilton, 1981] R. Axelrod and W. D. Hamilton. The evolution of cooperation. *Science*, 211:1390–1396, 1981.

[Axelrod, 1984] R. Axelrod. *The Evolution of Cooperation.* Basic Books, Inc., New-York, 1984.

[Axelrod, 1987] R. Axelrod. The evolution of strategies in the iterated prisoner's dilemma. In L. Davies, editor, *Genetic Algorithms and Simulated Annealing*, pages 32–42. Pitman, London, 1987.

[Bedau and Packard, 1992] M. A. Bedau and N. H. Packard. Measurement of evolutionary activity, teleology, and life. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, volume X of *SFI Studies in the Sciences of Complexity*, pages 431–461, Redwood City, CA, 1992. Addison-Wesley.

[Bersini and Detour, 1994] H. Bersini and V. Detour. Asynchrony induces stability in cellular automata based models. In R. A. Brooks and P. Maes, editors, *Artificial Life IV*, pages 382–387, Cambridge, Massachusetts, 1994. The MIT Press.

[Bonabeau and Theraulaz, 1994] E. W. Bonabeau and G. Theraulaz. Why do we need artificial life? *Artificial Life Journal*, 1(3):303–325, 1994. The MIT Press, Cambridge, MA.

[Burks, 1970] A. Burks, editor. *Essays on cellular automata.* University of Illinois Press, Urbana, Illinois, 1970.

[Byl, 1989] J. Byl. Self-reproduction in small cellular automata. *Physica D*, 34:295–299, 1989.

[Codd, 1968] E. F. Codd. *Cellular Automata.* Academic Press, New York, 1968.

[Collins and Jefferson, 1992] R. J. Collins and D. R. Jefferson. AntFarm: Towards simulated evolution. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, volume X of *SFI Studies in the Sciences of Complexity*, pages 579–601, Redwood City, CA, 1992. Addison-Wesley.

[Culik II *et al.*, 1990] K. Culik II, L. P. Hurd, and S. Yu. Computation theoretic aspects of cellular automata. *Physica D*, 45:357–378, 1990.

[Darwin, 1866] C. R. Darwin. *The Origin of Species.* Penguin, London, 1866. 1st edition reprinted, 1968.

[Durand *et al.*, 1994] S. Durand, A. Stauffer, and D. Mange. Biodule: an introduction to digital biology. Technical report, LSL, Swiss Federal Institute of Technology, Lausanne, Switzerland, September 1994.

[Eldredge and Gould, 1972] N. Eldredge and S. J. Gould. Punctuated equilibria: and alternative to phyletic gradualism. In T. J. M. Schopf, editor, *Models in Paleobiology*, pages 82–115. Freeman Cooper, San Francisco, 1972.

[Flood, 1952] M. M. Flood. Some experimental games. Technical Report RM-789-1, The Rand Corporation, Santa Monica, CA, 1952.

[Garzon, 1990] M. Garzon. Cellular automata and discrete neural networks. *Physica D*, 45:431–440, 1990.

[Goldberg, 1989] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[Gould and Lewontin, 1979] S. J. Gould and R. C. Lewontin. The spandrels of San Marco and the Panglossian paradigm: A critique of the adaptationist programme. *Proceedings of the Royal Society of London B*, 205:581–598, 1979.

[Gould and Vrba, 1982] S. J. Gould and E. S. Vrba. Exaptation- a missing term in the science of form. *Paleobiology*, 8:4–15, 1982.

[Gould, 1982] S. J. Gould. Darwinism and the expansion of evolutionary theory. *Science*, 216:380–387, 1982.

[Hartman and Vichniac, 1986] H. Hartman and G. Y. Vichniac. Inhomogeneous cellular automata. In E. Bienenstock, F. Fogelman, and G. Weisbuch, editors, *Disordered Systems and Biological Organization*, pages 53–57. Springer-Verlag, Berlin, 1986.

[Holland, 1975] J. H. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor, Michigan, 1975.

[Huberman and Glance, 1993] B. A. Huberman and N. S. Glance. Evolutionary games and computer simulations. *Proceedings of the National Academy of Sciences USA*, 90:7716–7718, August 1993.

[Jefferson *et al.*, 1992] D. Jefferson, R. Collins, C. Cooper, M. Dyer, M. Flowers, R. Korf, C. Taylor, and A. Wang. Evolution as a theme in Artificial Life: The Genesys/Tracker system. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, volume X of *SFI Studies in the Sciences of Complexity*, pages 549–578, Redwood City, CA, 1992. Addison-Wesley.

[Joyce, 1989] G. F. Joyce. RNA evolution and the origins of life. *Nature*, 338:217–224, March 1989.

[Kauffman and Johnsen, 1992] S. A. Kauffman and S. Johnsen. Co-evolution to the edge of chaos: Coupled fitness landscapes, poised states, and co-evolutionary avalanches. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, volume X of *SFI Studies in the Sciences of Complexity*, pages 325–369, Redwood City, CA, 1992. Addison-Wesley.

[Kauffman and Weinberger, 1989] S. A. Kauffman and E. D. Weinberger. The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of Theoretical Biology*, 141:211–245, 1989.

[Langton, 1984] C. G. Langton. Self-reproduction in cellular automata. *Physica D*, 10:135–144, 1984.

[Langton, 1986] C. G. Langton. Studying artificial life with cellular automata. *Physica D*, 22:120–140, 1986.

[Langton, 1989] C. G. Langton, editor. *Artificial Life*, volume VI of *SFI Studies in the Sciences of Complexity*. Addison-Wesley, Redwood City, CA, 1989.

[Lee *et al.*, 1990] Y. C. Lee, S. Qian, R. D. Jones, C. W. Barnes, G. W. Flake, M. K. O'Rourke, K. Lee, H. H. Chen, G. Z. Sun, Y. Q. Zhang, D. Chen, and C. L. Giles. Adaptive stochastic cellular automata: theory. *Physica D*, 45:159–180, 1990.

[Lindgren and Nordahl, 1994a] K. Lindgren and M. G. Nordahl. Cooperation and community structure in artificial ecosystems. *Artificial Life Journal*, 1(1/2):15–37, 1994. The MIT Press, Cambridge, MA.

[Lindgren and Nordahl, 1994b] K. Lindgren and M. G. Nordahl. Evolutionary dynamics of spatial games. *Physica D*, 75:292–309, 1994.

[Lindgren, 1992] K. Lindgren. Evolutionary phenomena in simple dynamics. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, volume X of *SFI Studies in the Sciences of Complexity*, pages 295–312, Redwood City, CA, 1992. Addison-Wesley.

[Lumer and Nicolis, 1994] E. D. Lumer and G. Nicolis. Synchronous verses asynchronous dynamics in spatially distributed systems. *Physica D*, 71:440–452, 1994.

[Mange and Stauffer, 1994] D. Mange and A. Stauffer. Introduction to embryonics: Towards new self-repairing and self-reproducing hardware based on biological-like properties. In N. M. Thalmann and D. Thalmann, editors, *Artificial Life and Virtual Reality*, pages 61–72, Chichester, England, 1994. John Wiley.

[Marchal *et al.*, 1994] P. Marchal, C. Piguet, D. Mange, A. Stauffer, and S. Durand. Embryological development on silicon. In R. A. Brooks and P. Maes, editors, *Artificial Life IV*, pages 365–370, Cambridge, Massachusetts, 1994. The MIT Press.

[Mayr, 1976] E. Mayr. *Evolution and the Diversity of Life.* Harvard University Press, Cambridge, MA, 1976.

[Mueller and Feldman, 1988] L. D. Mueller and M. W. Feldman. The evolution of altruism by kin selection: New phenomena with strong selection. *Ethology and Sociobiology*, 9:223–240, 1988.

[Nowak and May, 1992] M. A. Nowak and R. M. May. Evolutionary games and spatial chaos. *Nature*, 359:826–829, October 1992.

[Poundstone, 1992] W. Poundstone. *The Prisoner's Dilemma.* Doubleday, New York, 1992.

[Qian *et al.*, 1990] S. Qian, Y. C. Lee, R. D. Jones, C. W. Barnes, G. W. Flake, M. K. O'Rourke, K. Lee, H. H. Chen, G. Z. Sun, Y. Q. Zhang, D. Chen, and C. L. Giles. Adaptive stochastic cellular automata: applications. *Physica D*, 45:181–188, 1990.

[Rasmussen *et al.*, 1992] S. Rasmussen, C. Knudsen, and R. Feldberg. Dynamics of programmable matter. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, volume X of *SFI Studies in the Sciences of Complexity*, pages 211–254, Redwood City, CA, 1992. Addison-Wesley.

[Ray, 1994a] T. S. Ray. An evolutionary approach to synthetic biology: Zen and the art of creating life. *Artificial Life Journal*, 1(1/2):179–209, 1994. The MIT Press, Cambridge, MA.

[Ray, 1994b] T. S. Ray. A proposal to create a network-wide biodiversity reserve for digital organisms. unpublished, 1994. see also: *Science*, Vol. 264, May, 1994, page 1085.

[Reggia *et al.*, 1993] J. A. Reggia, S. L. Armentrout, H.-H. Chou, and Y. Peng. Simple systems that exhibit self-directed replication. *Science*, 259:1282–1287, February 1993.

[Simon, 1969] H. Simon. *The Sciences of the Artificial.* The MIT Press, Cambridge, Massachusetts, 1969.

[Sipper, 1994] M. Sipper. Non-uniform cellular automata: Evolution in rule space and formation of complex structures. In R. A. Brooks and P. Maes, editors, *Artificial Life IV*, pages 394–399, Cambridge, Massachusetts, 1994. The MIT Press.

[Smith, 1969] A. Smith. Cellular automata theory. Technical Report 2, Stanford Electronic Lab., Stanford University, 1969.

[Stork *et al.*, 1992] D. G. Stork, B. Jackson, and S. Walker. "Non-optimality" via preadaptation in simple neural systems. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, volume X of *SFI Studies in the Sciences of Complexity*, pages 409–429, Redwood City, CA, 1992. Addison-Wesley.

[Taylor and Jefferson, 1994] C. Taylor and D. Jefferson. Artificial life as a tool for biological inquiry. *Artificial Life Journal*, 1(1/2):1–13, 1994. The MIT Press, Cambridge, MA.

[Toffoli and Margolus, 1987] T. Toffoli and N. Margolus. *Cellular Automata Machines.* The MIT Press, Cambridge, Massachusetts, 1987.

[Vichniac *et al.*, 1986] G. Y. Vichniac, P. Tamayo, and H. Hartman. Annealed and quenched inhomogeneous cellular automata. *Journal of Statistical Physics*, 45:875–883, 1986.

[von Neumann, 1966] J. von Neumann. *The Theory of Self-Reproducing Automata.* University of Illinois Press, Illinois, 1966. Edited and completed by A.W. Burks.