Evolving Novel Behaviors via Natural Selection

A.D. Channon and **R.I. Damper**
Image, Speech & Intelligent Systems Research Group
University of Southampton, Southampton, SO17 1BJ, UK
http://www.soton.ac.uk/~adc96r
adc96r@soton.ac.uk rid@ecs.soton.ac.uk

Abstract

The traditional fitness function based methodology of artificial evolution is argued to be inadequate for the construction of entities with behaviors novel to their designers: Evolutionary emergence via natural selection (without an explicit fitness function) is the way forward. This paper further considers the question of what to evolve, the focus being on principles of developmental modularity in neural networks. To develop and test the ideas, an artificial world containing autonomous organisms has been created and is described. Results show the developmental system to be well suited to long-term incremental evolution. Novel emergent strategies are identified both from an observer's perspective and in terms of their neural mechanisms.

1 How to Evolve Novel Behaviors

The Artificial Life goal presents us with the problem that we do not understand (natural) life well enough to specify it to a machine. Therefore we must either increase our understanding of it until we can, or create a system which outperforms the specifications we can give it. The first possibility includes the traditional top-down methodology, which is clearly as inappropriate for ALife as it has proved to be for AI. It also includes manual incremental (bottom-up) construction of autonomous systems with the aim of increasing our understanding and ability to model life by building increasingly impressive systems, retaining functional validity by testing them within their destination environments.

The second option is to create systems which outperform the specifications given them and which are open to producing behaviors comparable with those of (albeit simple) natural life. Evolution in nature has no (explicit) evaluation function. Through organism-environment interactions, including interactions between similarly-capable organisms, certain behaviors fare better than others. This is how the non-random cumulative selection works without any long-term goal. It is why novel structures and behaviors emerge.

As artificial evolution is applied to increasingly complex problems, the difficulty in specifying satisfactory evaluation functions is becoming apparent – see (Zaera, Cliff & Bruten 1996), for example. At the same time, the power of natural selection is being demonstrated in prototypal systems such as Tierra (Ray 1991) and PolyWorld (Yaeger 1993). Artificial selection involves the imposition of an artifice crafted for some cause external to a system beneath it while natural selection does not. Natural selection is necessary for evolutionary emergence but does not imply sustained emergence (evermore new emergent phenomena) and the question "what should we evolve?" needs to be answered with that in mind (Channon & Damper 1998). This paper sets out to answer that question. Further discussion concerning evolutionary emergence can be found in (Channon & Damper 1998), along with evaluations of other natural selection systems. Note that an explicit fitness landscape is not a requirement for artificial selection and so an implicit fitness landscape does not imply natural selection.

The subject of how genotype lengths should change has been addressed by Harvey's 'Species Adaptation Genetic Algorithm' (SAGA) theory (Harvey 1993). He demonstrates that changes in genotype length should take place much more slowly than crossover's fast mixing of chromosomes. The population should be nearly-converged, evolving as species. Therefore the fitness landscape (actual or implicit) must be sufficiently correlated for mutation to be possible without dispersing the species in genotype space or hindering the assimilation by crossover of beneficial mutations into the species.

2 What to Evolve

Neural networks are the clear choice because of their graceful degradation (high degree of neutrality). But how should the network structure be specified? The evolutionary emergence of novel behaviors requires new neural structures. We can expect most to be descended from neural structures which once had different functions. There are many known examples of neural structures that serve a purpose different from a previous use.

Evidence from gene theory tells us that genes are used like a recipe, not a blueprint. In any one cell, at any one stage of development, only a tiny proportion of the genes will be in use. Further, the effect that a gene has depends upon the cell's local environment – its neighbors.

The above two paragraphs are related: For a type of module to be used for a novel function (and then to continue to evolve from there), without loss of current function, either an extra module must be created or there must be one 'spare' (to alter). Either way, a duplication system is required. This could be either by gene duplication or as part of a developmental process.

Gene duplication can be rejected as a sole source of neural structure duplication, because the capacity required to store all connections in a large network without a modular coding is genetically infeasible. Therefore, for the effective evolutionary emergence of complex behaviors, a modular developmental process is called for. For the sake of research validity (regarding long-term goals), this should be included from the outset.

2.1 Gruau's cellular encoding

Gruau used genetic programming (GP) (Koza 1992) to evolve his 'cellular programming language' code (Gruau 1996) to develop modular artificial neural networks. The programs used are trees of graph-rewrite rules whose main points are cell division and iteration.

The crucial shortcoming is that modularity can only come from either gene duplication (see objections above) or iteration. But iteration is not a powerful enough modular developmental backbone. Consider, for example, the cerebral cortex's macro-modules of hundreds of mini-columns. These are complicated structures that cannot be generated with a 'repeat one hundred times: mini-column' rule. There are variations between mini-columns.

2.2 Cellular automata

Many investigators have used conventional cellular automata (CA) for the construction of artificial neural networks. However, such work is more at the level of neuron growth than the development of whole networks. Although CA *rules* are suited to this latter developmental process in principle, the amount of work still to be done makes this a major research hurdle.

2.3 Diffusion models

While there are a number of examples of work involving the evolution of neural networks whose development is determined by diffusion along concentration gradients, the resulting network structures have (to date) been only basic. So as to concentrate on the intended area of research, these models have also been passed over.

2.4 Lindenmayer systems

Kitano used a context-free L-system (Lindenmayer 1968) to evolve connectivity matrices (Kitano 1990). The number of rules in the genotype was variable. Boers and Kuiper used a context-sensitive L-system to evolve modular feed-forward network architectures (Boers & Kuiper 1992). A fixed-length alphabet was used for the rules, restricting the possible network architectures. Both these works used backpropagation to train the evolved networks. Also, the resulting structures were fully-connected clusters of unconnected nodes (i.e. no links within clusters and if one node in cluster A is linked to one node in cluster B then all nodes in A are linked to all nodes in B). It may be that the results achieved reflect the workings of backpropagation more than evolution.

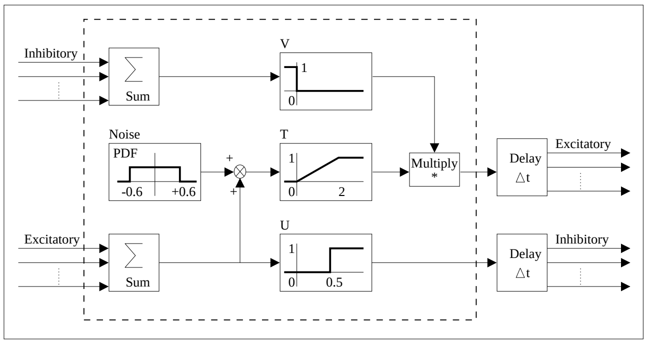

Figure 1: Schematic block diagram of a neuron, from (Cliff, Harvey & Husbands 1992).

3 Geb's Neural and Development Systems

The artificial neural networks used are recurrent networks of nodes as used successfully by Cliff, Harvey and Husbands in their evolutionary robotics work.

3.1 Developmental system

A context-free L-system was designed for the evolution of networks of these neurons. Specific attention was paid to producing a system in which children's networks resemble aspects of their parents'. Each node has a bit-string 'character' (label) associated with it, initialized at construction and modifiable during development. These characters may be of any non-zero length. A node may be a network input, a network output, or neither, as determined by an axiom (birth) network and development.

A rule matches a node if its predecessor is the start of the node's character. The longer the matching predecessor, the better the match; the best matching rule (if any) is applied. Thus ever more specific rules can evolve from those that have already been successful.

The production rules have the following form:

$$\mathcal{P} \rightarrow \mathcal{S}_r, \mathcal{S}_n \ ; b_1, b_2, b_3, b_4, b_5, b_6 \qquad \text{where:}$$

$\mathcal{P}$	Predecessor (initial bits of node's character)
$\mathcal{S}_r$	Successor 1: *replacement* node's character
$\mathcal{S}_n$	Successor 2: *new* node's character
bits:	link details [0=no,1=yes]:
(b_1, b_2)	reverse types [inhibitory/excitatory] of $(\text{input}, \text{output})$ links on $\mathcal{S}_n$
(b_3, b_4)	$(\text{inhibitory}, \text{excitatory})$ link from $\mathcal{S}_r$ to $\mathcal{S}_n$
(b_5, b_6)	$(\text{inhibitory}, \text{excitatory})$ link from $\mathcal{S}_n$ to $\mathcal{S}_r$

If a successor has no character (0 length) then that node is not created. Thus the predecessor node may be replaced by 0, 1 or 2 nodes. The 'replacement' successor (if it exists) is just the old (predecessor) node, with the same links but a different character. The 'new' successor (if it exists) is a new node. It inherits a copy of the old node's input links unless it has a link from the old node (b_3 or b_4). It inherits a copy of the old node's output links unless it has a link to the old node (b_5 or b_6).

New network input nodes are (only) produced from network input nodes and new network output nodes are (only) produced from network output nodes. Character-based matching of network inputs and outputs ensures that the addition or removal of nodes later in development or evolution will not damage the relationships of previously adapted network inputs and outputs.

3.2 Genetic decoding of production rules

The genetic decoding is loosely similar to that in (Boers & Kuiper 1992). For every bit of the genotype, an attempt is made to read a rule that starts on that bit. A valid rule is one that starts with 11 and has enough bits after it to complete a rule.

To read a rule, the system uses the idea of 'segments'. A segment is a bit string with its odd-numbered bits (1st, 3rd, 5th, ...) all 0. Thus the reading of a segment is as follows: read the current bit; if it is a 1 then stop; else read the next bit – this is the next information bit of the segment; now start over, keeping track of the information bits of the segment. Note that a segment can be empty (have 0 information bits).

The full procedure to (try to) read a rule begins with reading a segment for each of the predecessor, the first successor (replacement node) and the second successor (new node). Then, if possible, the six link-details bits are read. For example:

```
Genotype: 1 1 1 0 1 1 0 0 1 0 1 1 1 0 0 0 0 0
Decoding: +++ ->_1_ * _0_ * 0 1 1 1 0 0
            +++ _1_ ->_0_ * _1_ * 1 0 0 0 0 0
Rules: 1.  P  -> Sr ,  Sn , link bits
          any -> 1  ,  0  , 0 1 1 1 0 0
       2.         P  -> Sr ,  Sn , link bits
                  1  -> 0  ,  1  , 1 0 0 0 0 0
```

4 Experimental World

To develop and validate the above, a simple ALife system has been created. 'Geb' (after the Egyptian god of the earth) is a two-dimensional toroidal world of artificial organisms each controlled by a neural network using the developmental system above. Evolution is strictly by natural selection. There are no global system rules that delete organisms; this is under their own control.

Geb's world (figure 2) is divided into a grid of squares; uusually 20×20 of them. Individuals may not be within the same square at any one time. This gives the organisms a 'size' and puts a limit on their number. They are otherwise free to move around the world, within and between squares. As well as a position, each organism has a forward (facing) direction, set randomly at birth. Organisms are displayed as filled arcs, the sharp points of which indicate their direction. This is Geb's main algorithm:

Initialization
Every square in the world has an individual with a single-bit genotype '0' born into it.

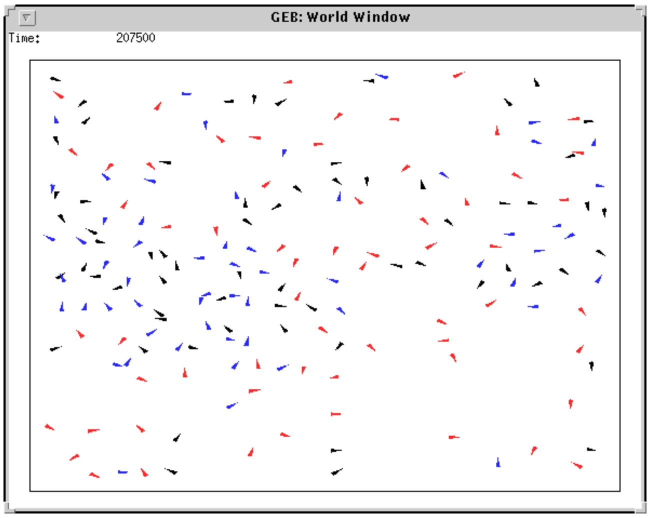

Figure 2: Geb's world.

Main Loop
In each time step (loop), every individual alive at the start of the cycle is processed once. The order in which the individuals are processed is otherwise random.
These are the steps involved for each individual:

- Network inputs are updated.
- Development – one iteration.
- Update all neural activations, including network outputs.
- Actions associated with certain network outputs are carried out according to those outputs. These actions are reproduce, fight, turn anti-clockwise, turn clockwise, and move forward.

4.1 Neural network details

The axiom network consists of three nodes with two excitatory links:

$$\text{network input } 001 \longmapsto 000 \longmapsto 01 \text{ network output}$$

The network output node's character (01) matches reproduction. The network input node's character (left input 01) matches this, without matching any of the other action characters. Finally, the hidden node's character neither matches nor is matched by the other nodes' or the action characters.

Development takes place throughout the individual's life, although necessary limits on the number of nodes and links are imposed.

4.2 Organism $\longleftrightarrow$ environment interactions

Five built-in actions are available to each organism. Each is associated with network output nodes whose characters start with a particular bit-string:

- `01*` Try to *reproduce* with organism in front
- `100*` *Fight*: Kill organism in front (if there is one)
- `101*` *Turn anti-clockwise*
- `110*` *Turn clockwise*
- `111*` *Move forward* (if nothing in the way)

For example, if a network output node has the character 1101001, the organism will turn clockwise by an angle proportional to the node's excitatory output. If an action has more than one matching network output node then the relevant output is the sum of these nodes' excitatory outputs, bounded by unity as within any node. If an action has no output node with a matching character, then the relevant output is noise, at the same level as in the (other) nodes.

Both *reproduce* and *fight* are binary actions. They are applied if the relevant network output exceeds a threshold and have no effect if the square in front is empty. Turning and moving forward are done in proportion to excitatory output.

When an organism reproduces with a mate in front of it, the child is placed in the square beyond the mate if that square is empty. If it is not, the child replaces the mate. An organism cannot reproduce with an individual that is fighting if this would involve replacing that individual. Reproduction involves crossover and mutation. Geb's crossover always offsets the cut point in the second individual by one gene, with equal probability either way – which is why the genotype lengths vary.

An organism's network input nodes have their excitatory inputs set to the weighted sum of 'matching' output nodes' excitatory outputs from other individuals in the neighborhood. If the first bit of an input node's character is 1 then the node takes its input from individuals to the right hand side (including forward- and back-right), otherwise from individuals to the left. An input node 'matches' an output node if the rest of the input node's character is the same as the start of the character of the output node. For example, an input node with character 10011 matches (only) output nodes with character's starting with 0011 in the networks of individuals to the right. Weighting is inversely proportional to the Euclidean distances between individuals. Currently the input neighborhood is a 5×5 area centered on the relevant organism.

5 Results

5.1 Kin similarity and convergence

When two Geb organisms (with networks developed from more than just a couple of production rules each) reproduce, the child's network almost always resembles a combination of the two parents' networks. Examination of networks from Geb's population at any time shows similarities between many of them. The population remains nearly-converged, in small numbers of species, throughout the evolution. The criterion of a sufficiently correlated (implicit) fitness landscape has been met by the developmental system, making it suitable for long-term evolution.

5.2 Emergence of increasing complexity

Once Geb has started, there is a short period while genotype lengths increase until capable of containing a production rule. For the next ten to twenty thousand time steps (in typical runs), networks resulting in very simple

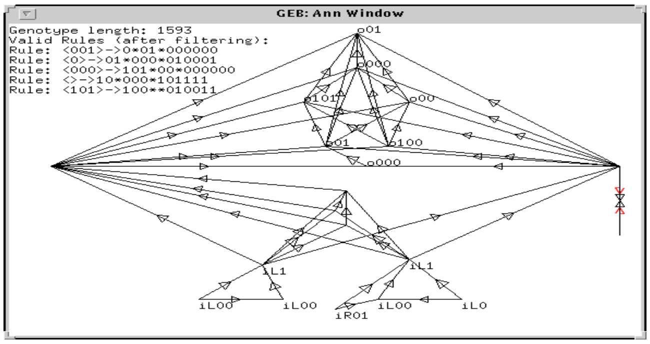

Figure 3: A dominant organism

strategies such as 'do everything' and 'always go forwards and kill' dominate the population. Some networks do better than others but not sufficiently well for them to display a dominating effect.

In every run to date, the first dominant species that emerges has been one whose individuals turn in one direction while trying to fight and reproduce at the same time. Figure 3 shows an example of such an individual, after the user had dragged the nodes apart to make detailed examination possible. Note the outputs o101, o01 [x2] and o100 (turn anti-clockwise, reproduce and fight). Note also the large number of links necessary to pass from inputs to outputs, and the input characters which match non-action output characters of the same network (o000 [x2], o00). Individuals of this species use nearby members of the same species, who are also turning in circles, as sources of activation (so keeping each other going).

Although a very simple strategy, watching it in action makes its success understandable. Imagine running around in a small circle stabbing the air in front of you. Anyone trying to attack would either have to get their timing exactly right or approach in a faster spiral – both relatively advanced strategies. These individuals also mate just before killing. The offspring (normally) appear beyond the individual being killed, away from the killer's path.

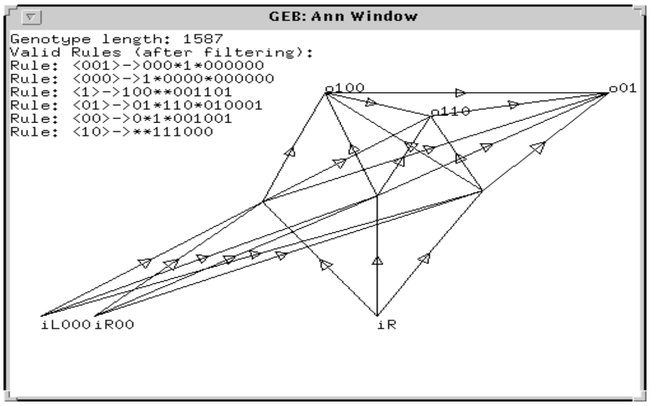

Figure 4: A rebel

Because of the success of this first dominant species,

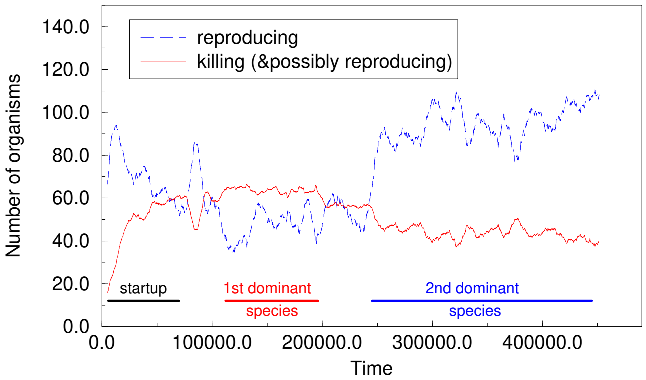

Figure 5: Typical run (running averages).

the world always has enough space for other organisms to exist. Such organisms tend not to last long; almost any movement will bring them into contact with one of the dominant organisms. Hence these organisms share some of the network morphology of the dominant species. However, they can make some progress: Individuals have emerged that are successful at turning to face the dominant species and holding their direction while trying to kill and reproduce. An example of such a 'rebel' (from the same run as figure 3) is shown in figure 4.

Running averages of the number of organisms reproducing and killing (figure 5) suggest that further species emerge. However, organisms have proved difficult to analyze beyond the above, even at the behavioral level. All that can currently be said is that they share characteristics of the previous species but are different.

6 Conclusions

The main conclusion is that the proposed approach is viable. Although the behaviors that emerged are very low-level, they are encouraging nonetheless, for the increases in complexity were in ways not specified by the design. It is difficult to evaluate any ongoing emergence, because of the difficulty in analyzing later organisms. Either tools to aid in such analysis will have to be constructed, or a more transparent system created.

In work involving pure natural selection, the organisms' developmental and interaction systems are analogous to the fitness functions of conventional genetic algorithms. While the general aim involves moving away from such comparisons, the analogy is useful for recognizing how the epistasis of fitness landscape issue transfers across: Certain ontogenetic and interaction systems can result in individuals with similar genotypes but very different phenotypes. The results show that Geb's developmental system does not suffer from this problem, making it suitable for long-term incremental evolution. This alone is a significant result for a modular developmental system.

This work has made it clear that the *specification* of 'actions', even at a low-level, results in the organisms being constrained around these actions and limits evolution. Alternatives in which the embodiment of organisms is linked to their actions need to be investigated.

Acknowledgements

This work is supported by an award from the United Kingdom's Engineering and Physical Sciences Research Council to author ADC. It is a continuation of previous work also supported by an award from the EPSRC (supervisor Inman Harvey, University of Sussex).

References

Boers, E. J., and Kuiper, H. 1992. Biological metaphors and the design of modular artificial neural networks. Master's thesis, Departments of Computer Science and Experimental Psychology, Leiden University.

Channon, A., and Damper, R. 1998. Perpetuating evolutionary emergence. http://www.soton.ac.uk/~adc96r/. Submitted to SAB98, the Fifth International Conference of the Society for Adaptive Behavior.

Cliff, D.; Harvey, I.; and Husbands, P. 1992. Incremental evolution of neural network architectures for adaptive behaviour. Technical Report CSRP256, University of Sussex School of Cognitive and Computing Sciences.

Gruau, F. 1996. Artificial cellular development in optimization and compilation. Technical report, Psychology department, Stanford University, Palo Alto, CA.

Harvey, I. 1993. Evolutionary robotics and SAGA: the case for hill crawling and tournament selection. In Langton, C. G., ed., *Artificial Life III*. Addison-Wesley.

Kitano, H. 1990. Designing neural networks using genetic algorithms with graph generation system. *Complex Systems* 4:461–476.

Koza, J. R. 1992. *Genetic Programming*. Cambridge, MA: MIT Press/ Bradford Books.

Lindenmayer, A. 1968. Mathematical models for cellular interaction in development. *Journal of Theoretical Biology* 18:280–315. Parts I and II.

Ray, T. S. 1991. An approach to the synthesis of life. In Langton, C.; Taylor, C.; Farmer, J.; and Rasmussen, S., eds., *Artificial Life II*, 371–408. Redwood City, CA: Addison-Wesley.

Yaeger, L. 1993. Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or polyworld: Life in a new context. In Langton, C. G., ed., *Artificial Life III*, 263–298. Addison-Wesley.

Zaera, N.; Cliff, D.; and Bruten, J. 1996. (Not) evolving collective behaviours in synthetic fish. In Maes, P.; Mataric, M.; Meyer, J.-A.; Pollack, J.; and Wilson, S., eds., *Proceedings of SAB96*, 635–644. MIT Press Bradford Books.