# The Evolutionary Emergence of Socially Intelligent Agents

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

Evolutionary emergence is the key to generating increasingly socially intelligent agents. In order to generate agents with novel behaviors beyond our manual design capability, long-term incremental evolution with continuing emergence within a social environment is called for. Purely artificial selection models are argued to be fundamentally inadequate for this calling and a new natural selection system containing simple virtual agents is presented. Each agent is controlled by a genetically determined neural network – controllers suited to both incremental evolution and the goal of intelligent behaviors. Resulting evolutionary emergent social behaviors are reported alongside their neural correlates. In one example, the collective behavior of one species clearly provides a selective force which is overcome by another species, demonstrating the perpetuation of evolutionary emergence via naturally-arising social coevolution.

**Keywords:** emergence, evolution, intelligence, natural selection, novelty, social agents

## 1. Introduction

The aim of generating intelligent behaviors in artificial agents presents us with the problem that we do not understand such intelligence well enough to program it into a machine. Therefore, we must either increase our understanding 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 inappropriate. Ryle (1949) long ago pointed out the futility of attempts to define intelligence, since they must invoke the 'ghost in the machine' fallacy: we can never observe intelligence directly; we can only define that some behaviors are more intelligent than others.

The first option also includes manual incremental (bottom-up) construction of agents with the intention of increasing our understanding and ability to model intelligence. The aim here is to build increasingly impressive agents, retaining functional validity by testing them within their destination environments. However, bearing in mind the fundamentally distributed nature of intelligent behaviors, it is unlikely that human designers will be capable of manually producing intelligence beyond a rudimentary level.

Further, the first option tends to be at odds with the general-purpose nature of intelligence. To argue that creating general-purpose intelligence is too vague or hard a problem and that research should deal first with specific, static behaviours is analogous to advising Charles Babbage to think first about designing a machine to sort coins (say), before tackling the more demanding issues involved for a general-purpose computing machine. Such arguments completely miss the point. The way forward is to tackle the issue in its entirety.

The second option is to create systems which outperform the specifications given them and which are open to producing intelligent behaviors comparable with those of (albeit simple) natural agents. Evolution in nature has no (explicit) evaluation function. Through agent-environment interactions, 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.

## 2. The need for sociality and natural selection

Because this approach requires evolution to be self-incremental, other parts of the environment (not just our so far isolated agent) must also evolve. The most obvious solution is to realize other agents as the evolvable part of the environment. Thus coevolution can occur through agent-agent interactions – socially.

The use of coevolutionary models is fast becoming a dominant approach in the adaptive behavior field. This is essentially a response to the problems encountered when trying to use artificial selection to evolve complex behaviors. However, artificial selection has kept its hold so far – most systems still use fitness functions. Much of this work is based on the 'Red Queen' or 'Arms Race' phenomenon (see Cliff and Miller, 1995; Dawkins and Krebs, 1979), an early example of which is Hillis' coevolution of sorting networks and their test cases. Hillis concluded his paper with the statement: "It is ironic, but perhaps not surprising, that our attempts to improve simulated evolution as an optimization procedure continue to take us closer to real biological systems" (Hillis, 1990, page 233).

As with Hillis' paper, the reasoning given for imposing coevolution is often that it provides "a useful way of dealing with the problems associated with static fitness landscapes" (Bullock, 1995, section 5). It appears that few of those working with artificial selection intentionally use coevolution as a step towards intrinsic evolution. Notably, Reynolds (of 'Boids' fame) worked towards more automatic evolution by coevolving simulated mobile agent controllers which competed with each other in games of 'tag' (Reynolds, 1994). This eliminated the need to design a controller in order to evolve a controller, as required in his previous work (Reynolds, 1992).

Emergence is related to qualitatively novel structures and behaviors which are not reducible to those hierarchically below them. It poses an attractive methodology for tackling Descartes' Dictum: "how can a designer build a device which outperforms the designer's specifications?" (Cariani, 1991, page 776). Most importantly, it is *necessary* for the generation of agents with intelligent behaviors beyond our manual design capability.

Cariani identified the three current tracts of thought on emergence, calling them "computational", "thermodynamic" and "relative to a model" (Cariani, 1991). Computational emergence is related to the manifestation of new global forms, such as flocking behavior and chaos, from local interactions. Thermodynamic emergence is concerned with issues such as the origins of life, where some degree of order emerges from noise. The emergence relative to a model concept deals with situations where observers need to change their model in order to keep up with a system's behavior. This is close to Steels' concept of emergence, which refers to ongoing processes which produce results invoking vocabulary not previously involved in the description of a system's inner components – "new descriptive categories" (Steels, 1994, section 4.1).

Evolutionary emergence falls into the 'emergence relative to a model' category. An example will clarify the divisions. Consider a virtual world containing agents that can move and try to reproduce or kill according to rules which are sensitive to the presence of other agents and which evolve under natural selection. Should flocking manifest itself in this system, perhaps due to agent diversification into predators and prey, we could classify it as emergent in two senses: firstly in the 'computational' sense from the interaction of local rules, flocking being a collective behavior, and secondly in the 'relative to a model' sense through the evolution, the behavior being novel to the system. While the first is also relevant to our goal, in that complex adaptive systems will involve such emergence, the second sense is the key to understanding *evolutionary* emergence.

Artificial selection can only select for that which is specified. Therefore anything that emerges during evolution *must* result from another aspect of selection, which must in turn arise from the innate dynamics of the system – natural selection. In the context of evolutionary emergence, any artificial selection used constitutes just one of the parts of a system.

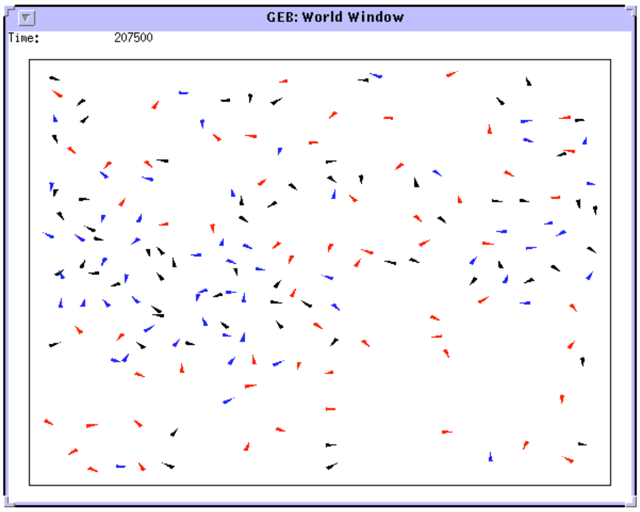



**Figure 1: The experimental world (Geb)**

## 3. Experimental model

A system believed to be suited to the incremental artificial evolution of socially intelligent agents by natural selection has been created. 'Geb' (named after the Egyptian god of the Earth) is a two-dimensional toroidal virtual world containing agents, each controlled by a neural network. The networks are produced from bit-string genotypes by a developmental process based on L-systems. Neural controllers were chosen because of their suitability for both incremental evolution (due to their graceful degradation) and the goal of intelligent behaviors. Evolution within Geb is strictly by ***Natural Selection***. There are no global system rules that delete agents; this is under their own control.

Geb's world (figure 1) is divided into a grid of 20 × 20 areas (squares). At most one individual may occupy an area at any one time. This effectively gives the agents a size within the world and puts a limit on their number. Individuals are otherwise free to move around the world, within and between areas. As well as a position within the world, each agent has a forward (facing) direction, set randomly at birth. Agents 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.

**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.

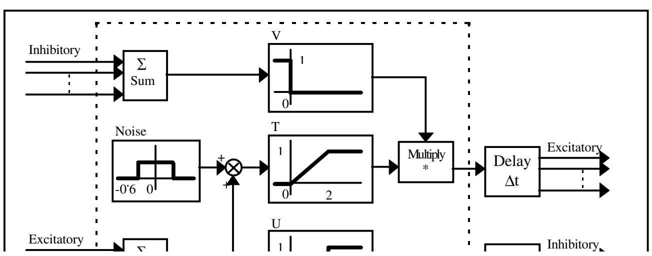
These are the steps involved for each individual:

1. Network inputs are updated from the outputs of neighboring agents.
2. Development – one iteration of the ontogenesis mechanism.
3. All neural activations, including network outputs, are updated.
4. Actions associated with certain network outputs are carried out according to those outputs.

Each agent is born with an axiom network (of just three neurons) that generates reproductive behavior. Each node has a bit-string 'character' assigned to it. This is used by the genetically determined developmental process and by the interaction system. The developmental process involves dividing and destroying nodes in ways that either preserve the links between surviving nodes or creates new ones. Each time a node is divided, its character is changed. A character-based method of matching up agents' inputs and outputs ensures that the addition or removal of an input/ output node at a later stage of development or evolution will not damage the relationships of previously adapted interactions.

### 3.1. The neurons

The neural networks used in Geb are recurrent networks of nodes as used successfully by Cliff, Harvey and Husbands in their evolutionary robotics work (figure 2).



PDF

+0'6

**Figure 2: Schematic of a neuron, from (Cliff, Harvey and Husbands, 1992)**

All links have unit weight; no lifetime learning is used. This is to avoid the criticism that lifetime learning may be the main factor.

## 3.2. Agent ↔ environment interactions

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

1. `01*` Try to *reproduce* with agent in front
2. `100*` *Fight*: Kill agent in front (if there is one)
3. `101*` *Turn anti-clockwise*
4. `110*` *Turn clockwise*
5. `111*` *Move forward* (if nothing in the way)

For example, if a network output node has the character 1101001, then the agent will turn clockwise by an angle proportional to the excitatory output of that node. If an action has more than one matching output node then the relevant network output is the sum of these nodes' excitatory outputs, bounded as within any node. If an action has no output node with a matching character, then the relevant network 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 agent reproduces with another in front of it, the child is placed in the square beyond the other individual if that square is empty. If the square is not, the child replaces the individual being mated with. An agent cannot reproduce with an individual that is fighting if this would involve replacing the fighting individual.

Reproduction involves crossover and mutation. Geb's crossover always offsets the cut point in the second individual by one gene (bit position), with equal probability either way. This is why the genotype lengths vary. Mutation at reproduction is a single gene-flip (bit-flip) on the child genotype.

An agent's network input nodes have their excitatory inputs set to the weighted sum of 'matching' network output nodes' excitatory outputs, from other individuals in the neighborhood. If the first bit of a network 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. A network input node 'matches' a network 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, a network input node with character 10011 matches (only) network output nodes with characters starting with 0011 in the networks of individuals to the right. The weights are inversely proportional to the Euclidean distances between individuals. Currently the input neighborhood is a $5 \times 5$ square area centered on the relevant agent.

Notice that the network output nodes with characters 0, 1, 10, 11 and all those starting with 00 produce no action. However, their excitatory values can still be input by other individuals. Thus there is the potential for data exchange not directly related to the actions.

## 3.3. The developmental system

A context-free L-system was designed for the evolution of networks of the neurons outlined above. Specific attention was paid to producing a system in which children's networks resemble aspects of their parents'.

Every node is processed once during each developmental step. The production rule that best matches the node's character is applied (if there is one). 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. Thus ever more specific rules can evolve from those that have already been successful.

The production rules have the following form:

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

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

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. Necessary limits on the number of nodes and links are imposed.

The 'replacement' successor (if it has a character) is just the old (predecessor) node, with the same links but a different character. The 'new' successor (if it has a character) 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. The character-based method of matching up network inputs and outputs ensures that the addition or removal of a network input/ output node at a later stage of development or evolution will not damage the relationships of previously adapted network inputs and outputs.

The axiom network consists of three nodes with two excitatory links. 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, and the hidden node's character neither matches nor is matched by the other nodes' or the action characters:

network input 001 $\longrightarrow$ 000 $\longrightarrow$ 01 network output

Details of the theory behind the choice of developmental system can be found in (Channon and Damper, 1998a).

## 3.4. The genetic decoding

The genetic decoding of production rules is loosely similar to that of Boers and 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. Results

The system has demonstrated non-trivial *incremental* species evolution and emergence. This alone is an advance on previous natural selection systems – see Channon and Damper (1998b). Further, some of the behaviors that have emerged demonstrate (very) rudimentary social intelligence, including both social cooperation and social competition. These cannot be directly explained in terms of artificial selection or any other aspect of the initial system specification (program).

Note that the evolutionary concept of species is being used, emphasizing the cladistic (branching) nature of speciation and allowing for interbreeding species. A cross between individuals of different species will probably produce either a very unfit child (which may not develop into a viable adult) or a new member of one of the existing species (the child inheriting only a few genes from one species that are not common to the other). This latter possibility is very much more likely within artificial systems with short genotypes and few genes than in nature. Other issues of species evolution in artificial systems have been addressed by Harvey's 'Species Adaptation Genetic Algorithm' theory (Harvey, 1992).

## 4.1. Emergent social behavior and Cooperation

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

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. Note the network outputs o101, o01 [x2] and o100 (turn anti-clockwise, reproduce and fight). Note also the large number of links necessary to pass from network inputs to outputs, and the network 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).

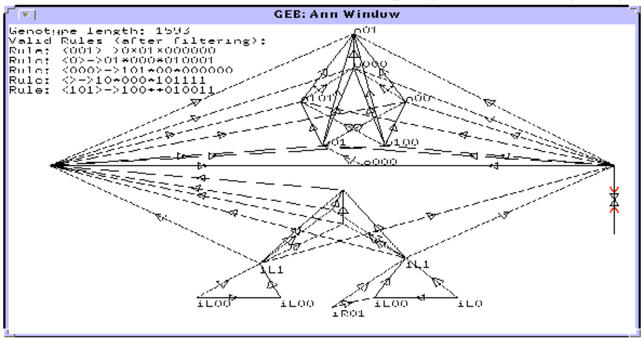



**Figure 3: A dominant agent's neural network**

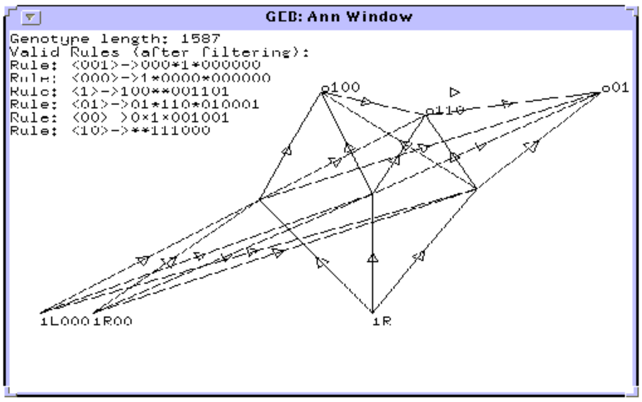



**Figure 4: A rebel agent's neural network**

Although a very simple strategy, watching it in action makes its success understandable. The individuals keep each other moving quickly, in tight circles. Any attacking agent would have to either get its 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.

## 4.2. Naturally arising coevolution and competition

Because of the success of this first dominant species, the world always has enough space for other agents to exist. Such agents tend not to last long; almost any movement will bring them into contact with one of the dominant agents, helping that species in its reproduction as much as themselves. Hence these agents 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. Note that most rebels have many more nodes and links; this one was picked for its clarity.

Further, 'running averages' of the number of agents reproducing and killing (figure 5) suggest that further species emerge, indicating perpetuating evolutionary emergence. However, agents 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.

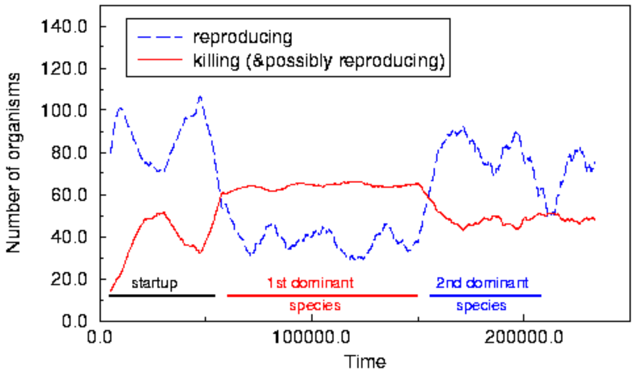



**Figure 5: Typical run (running averages of population sizes by actions)**

## 5. Conclusions

Rather than specifying behavioral descriptions, the approach used here acknowledges evolutionary emergence as the underlying force behind socially intelligent behaviors. While computational emergence can arise via artificial selection, evolutionary emergence requires natural selection. The logical progression or aim is the perpetuation of evolutionary emergence via naturally arising coevolution. However, this requires long-term incremental evolution and so what we evolve must be chosen accordingly. Neural networks are a clear choice, because of their graceful degradation (high degree of neutrality) and suitability for the goal of intelligent behaviors.

Social agent research should be leading the way, through the natural selection of neural controllers, towards the emergence of ever more intelligent behaviors. The work discussed in this paper has started down that route, with some success. Geb has proven to be suitable for long-term incremental artificial evolution. The behaviors identified are encouraging too, for the increases in complexity were in ways not specified by the design – evolutionary emergence, and some of the behaviors demonstrate (very) rudimentary social intelligence.

Whether or not emergence is continuing in Geb is hard to tell, for it soon becomes difficult to identify behaviors; evolved neural networks are hard to understand and so offer little help. Constructing systems such that behaviors will be more transparent is likely to be the most productive way forward.

A further problem is that specifying the available 'actions' constrains agents around these actions and so limits evolution. Despite showing the important new result of evolutionary emergent behaviors (not specified within the initial system) within a social system suited to long-term incremental evolution, all basic (inter-)actions were as specified within the initial system and not evolvable. At a later stage, alternatives in which the evolvable embodiment of an agent gives rise to its actions will need to be considered. Then the number of possible social interactions will be unlimited by design, encouraging the evolutionary emergence of socially intelligent agents.


# 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 (Channon, 1996) also supported by an award from the EPSRC (supervisor Inman Harvey, University of Sussex).

## References


**Boers, Egbert J. W. and Kuiper, Herman**. 1992. *Biological metaphors and the design of modular artificial neural networks.* Master's thesis, Departments of Computer Science and Experimental Psychology, Leiden University, The Netherlands.

**Bullock, Seth G.** 1995. *Co-evolutionary Design: Implications for Evolutionary Robotics.* Technical report CSRP384. University of Sussex School of Cognitive and Computing Sciences.

**Cariani, P.** 1991. Emergence and Artificial Life, from *Artificial Life II, Santa Fe Institute Studies in the Sciences of Complexity, Vol. X.*, ed. Langton, C. G. and Taylor, C. and Farmer, J. D. and Rasmussen, S. Addison-Wesley, Redwood City, CA. Pages 775–797.

**Channon, A. D.** 1996 (September). *The Evolutionary Emergence route to Artificial Intelligence.* Master's thesis, School of Cognitive and Computing Sciences, of Sussex. Revision October 1996. http://www.soton.ac.uk/~adc96r/.

**Channon, A. D. and Damper, R. I.** 1998a. Evolving Novel Behaviors via Natural Selection, from *Proceedings of ``Artificial Life VI'', Los Angeles, June 26-29, 1998.*, ed. Adami, C. and Belew, R. and Kitano, H. and Taylor, C. MIT Press. http://www.soton.ac.uk/~adc96r/.

**Channon, A. D. and Damper, R. I.** 1998b. Perpetuating evolutionary emergence, from *Proceedings of SAB98, the Fifth International Conference of the Society for Adaptive Behavior, Zurich, August 17-21, 1998.* MIT Press. http://www.soton.ac.uk/~adc96r/.

**Cliff, Dave and Harvey, Inman and Husbands, Phil**. 1992. *Incremental evolution of neural network architectures for adaptive behaviour.* Technical report CSRP256. University of Sussex School of Cognitive and Computing Sciences.

**Cliff, D. and Miller, G.** 1995. Tracking the Red Queen: Measurements of adaptive progress in co-evolutionary simulations, from *Advances in Artificial Life: Proceedings of the Third European Conference on Artificial Life.*, ed. Mor , F. and Moreno, A. and Merelo, J. J. and Chacon, P. Springer Verlag.

**Dawkins, R. and Krebs, J. R.** 1979. Arms races between and within species. *Proceedings of the Royal Society*, **B**(205), pages 489–511.

**Harvey, Inman**. 1992. Species Adaptation Genetic Algorithms: A basis for a continuing SAGA, from *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life.*, ed. Varela, F. J. and Bourgine, P. MIT Press/ Bradford Books, Cambridge, MA. Pages 346–354.

**Hillis, W. Daniel**. 1990. Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D*, **42**, pages 228–234.

**Reynolds, Craig W.** 1992. An Evolved, Vision-Based Behavioral Model of Coordinated Group Motion, from *From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior (SAB92).*, ed. Meyer and Roitblat and Wilson MIT Press, Cambridge, MA. Pages 384–392.

**Reynolds, Craig W.** 1994. Competition, Coevolution and the Game of Tag, from *Artificial Life IV.*, ed. Brooks, Rodney A. and Maes, Pattie MIT Press, Cambridge, MA. Pages 59–69.

**Ryle, G.** 1949. *The Concept of Mind.* Hutchinson.

**Steels, Luc**. 1994. The artificial life roots of artificial intelligence. *Artificial Life Journal*, **1**(1), pages 89–125.