UNIVERSITY OF SOUTHAMPTON

Evolutionary Emergence

The Struggle for Existence in Artificial Biota

by

Alastair Channon

A thesis submitted for the degree of

Doctor of Philosophy

in the

Faculty of Engineering and Applied Science

Department of Electronics and Computer Science

November 2001

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE

DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

EVOLUTIONARY EMERGENCE: THE STRUGGLE FOR EXISTENCE IN ARTIFICIAL BIOTA

by Alastair Channon

The generation of complex entities with advantageous behaviours beyond our manual design capability requires long-term incremental evolution with continuing emergence. This thesis presents the argument that artificial selection models, such as traditional genetic algorithms, are fundamentally inadequate for this goal. Existing natural selection systems are evaluated, revealing both significant achievements and pitfalls. Thus, some requirements for the perpetuation of evolutionary emergence are established. An (artificial) environment containing simple virtual autonomous organisms with neural controllers has been created to satisfy these requirements and to aid in the development of an accompanying theory of evolutionary emergence. Resulting behaviours are reported alongside their neural correlates. In one example, the collective behaviour of one species provides a selective force which is overcome by another species, demonstrating the incremental evolutionary emergence of advantageous behaviours via naturally-arising coevolution. Further behavioural or neural analysis is infeasible in this environment, so evolutionary statistical methods are employed and extended in order to classify the evolutionary dynamics. This qualitative analysis indicates that evolution is unbounded in the system. As well as validating the theory behind it, work with the system has provided some useful lessons and directions towards the evolution of increasingly complex advantageous behaviours.

Contents

List of Figures

Acknowledgements

This work was supported by an award from the United Kingdom's Engineering and Physical Sciences Research Council (supervisor Bob Damper). It is a continuation of previous work (Channon, 1996) also supported by an award from the EPSRC (supervisor Inman Harvey, University of Sussex). My thanks to both supervisors for their encouragement in the area, comments on the work and guidance through the postgraduate process. I would like to offer special thanks to Bob Damper for giving me the chance to follow my own interests and so to work in this exciting area. Without such trust this research would not have been possible.

Thanks to Mark Bedau for emails and discussions on my evolutionary statistics work, and of course for developing the basis for them. Thanks also to the many other friends and colleagues who have provided so much relevant discussion and enjoyment at conferences: Seth Bullock, Dave Cliff, Ezequiel Di Paolo, Steve Grand, Inman Harvey, Phil Husbands, Jason Noble, Tom Ray, Anil Seth, Tim Taylor, Adrian Thompson, Richard Vaughan, Richard Watson, Michael Wheeler and many more. Special thanks to Tim Taylor and Richard Watson: conferences are yet more worthwhile when colleagues are also very good friends.

My partner has had to share me with my work for a long time now. So thank you, Jacqui, for putting up with the late nights and working weekends. Thank you also for your support during the final stages. Last, but not least, I would like to thank my parents for ensuring that I received a good education as a child, with the opportunity and encouragement to learn.

Chapter 1

Introduction

1.1 Background: Darwinism

Darwinism is the most powerful and unifying theory in biology. Underlying it is the principle of ‘descent with modification’ (evolutionary change), that species are not fixed: that new species arise from those which already exist, some of which become extinct. The theory can be summarised as four statements of fact and an inference (cf. Gould, 1996, chapter 12):

1. All organisms tend to produce more offspring than survive.
2. There are variations between offspring.
3. Some at least of this variance is heritable.
4. Some heritable variations increase the probabilities of survival and reproduction, or rather of further inheritance to subsequent generations, within the context of the current environment.
5. Hence, such variations will tend to be maintained and accumulate over the generations.

The theory is one of local change and adaptation, not of optimisation along an absolute scale of fitness. “Selection” (the accepted yet unfortunate choice of word) is an abstraction of statement 4. It is better viewed in terms of probability of persistence through inheritance, instead of its usual treatment as an active selection mechanism favouring fit organisms. Indeed, Darwin (1859) introduced natural selection by reference and contrast to artificial selection, as carried out in mankind's selective breeding of animals and plants, which is an active selection mechanism.

To explain the huge increases in organism complexity which have clearly occurred over the course of evolution, Darwin divided the "struggle for existence" (p. 62) into the struggle between organisms for limited resources (*biotic* competition) and the struggle against features such as drought, of the non-living physical environment (*abiotic* competition). Biotic competition, he argued, has been the cause of evolutionary progress. For in Darwin's model, consistent with statement 1 above, ecologies are always full (or at least quickly become so) and any new species must therefore displace others to some extent. "The inhabitants of each successive period in the world's history have beaten their predecessors in the race for life, and are, in so far, higher in the scale of nature" (p. 345). Gould (1996) reminds us that lower-complexity organisms have not been displaced: the world abounds with bacteria. He describes the distribution of organism complexity as having spread out from, and always including, a lower bound (that of bacteria), and becoming increasingly skewed in favour of simpler organisms. However, Darwin's model still holds at any single order of physical magnitude of an ecology, and the highest organism complexity has increased over the course of evolution.

1.2 Research motivation

1.2.1 Scientific motivation

Neo-Darwinism, which adds Mendelian heredity and post-Mendelian genetic theory, has clarified the nature and origins of species to the extent that we can now carry out evolutionary experiments within artificial systems such as computer simulations and robotics. However, within the fields of artificial evolution (including evolutionary/genetic algorithms/programming, and sub-fields of artificial life, adaptive behaviour and digital biota), work to date uses artificial selection, with very few exceptions. Most of Darwin's theory would seem to have been ignored. Where natural selection has been used (in the systems examined later in the thesis), serious problems of evolvability can be identified. This thesis puts the natural selection of artificial organisms, preferably worded as "The Struggle for Existence in Artificial Biota" (thesis sub-title), at its centre. It also aims towards the creation of an open-ended such system, within which increasing complexity can emerge, so addressing the evolvability issues.

This brings us to the issue of emergence. As explained in the previous section, there is an emphasis on biotic competition because this is theorised to be the cause of increasing complexity. *Evolutionary emergence*, which I specify more exactly in section 3.1, is probably the only process capable of producing open-ended increases in complexity, and so holds the key to our understanding 'life, the universe and everything'. This point is developed, to some degree, through the course of the thesis. Further, computer-based experiments into the struggle for existence in artificial biota provide an excellent method of investigating evolutionary emergence. Holland (1997) presents similar opinions:

> "Our understanding of the universe will be severely limited until we have a more definitive view of how much of *life* and *consciousness* can be explained as emergent phenomena." (p. 248)
>
> "To build a competent theory one needs deep insights ... In the case of emergence, these insights can only be gained with the help of computer-based *exploration*." (p. 240)

1.2.2 Engineering motivation

This work aims towards the generation of systems within which increasingly complex advantageous behaviours can emerge. The adjective *advantageous* is used rather than *adapted* because the concern is the emergence of increasingly complex behaviours which mould a dynamical system of artificial entities, rather than just fit into an environment. This presents a dilemma: we do not understand such behaviours well enough to program them into a machine.

So we must either increase our understanding until we can, or create a system which outperforms the specifications we give it.

The first possibility includes the traditional top-down methodology, which appears as inappropriate here as it has so far proved to be for (symbolic) artificial intelligence. It also includes manual incremental (bottom-up) construction of autonomous systems with the aim of increasing our understanding and ability to model complex behaviours. The goal here is to build increasingly impressive systems, retaining functional validity by testing them within their destination environments (e.g. Wilson, 1991). However, by the very nature of complexity, it is unlikely that human designers will be capable of manually producing complex *advantageous* behaviours beyond a rudimentary level.

The second option is to create systems which outperform the specifications given them and which are open to producing increasingly complex advantageous behaviours. Evolution in nature has no (explicit) evaluation function. Through organism-environment interactions, including interactions between similarly-capable organisms, certain behaviours persist while others die off. This is how the non-random cumulative selection operates without any long-term goal; it is why novel structures and behaviours emerge. Whereas work on adapted behaviour focuses on fitness in the present, research on advantageous behaviour must shift the focus to the future, where what it is to be fit may have changed because of the moulding of the environment. However, this does not prevent us from evolving advantageous behaviours and discovering which ones are fit with respect to the resulting evolving environment – that is, which ones persist. Further, one can expect many behaviours that evolve and persist to have done so because they are (or at least have been) advantageous, in that the behaviours contribute to the persistence of their host system (organism, species or such).

1.3 Research objectives

In light of the above, the research objectives can be stated simply:

1. To add to the list of known requirements for the generation of evolutionary systems within which increasingly complex advantageous behaviours can emerge.
2. To do this by creating computer-based "struggles for existence in artificial biota" aimed at this goal and learning from the achievements and shortcomings of these systems.

1.4 Thesis overview

This thesis presents a detailed argument for the use of natural selection systems as a means of generating evolutionary emergence, before describing experimental results which further develop a methodology for constructing such systems. Accordingly, the remainder of the thesis is structured as follows. Chapter 2 provides an overview of the field of incremental artificial evolution, necessary for long-term evolutionary emergence. Chapter 3 presents the bulk of the theory. Section 3.1 specifies what evolutionary emergence is, and how it differs from other types of emergence. Section 3.2 makes the case that artificial selection cannot generate evolutionary emergence (by my definitions), which must therefore be the product of natural selection. Section 3.3 refines the argument and concludes that the important distinction is not between artificial and natural selection, but between abiotic and biotic selection. Section 3.4 evaluates existing biotic selection systems. This section is more than a tutorial: it offers important and novel insights into these systems, revealing not only their significant achievements but also crucial pitfalls. Thus, it provides the first major contribution documented in this thesis. Section 3.5 addresses the issue of *what* to evolve, the focus being on principles of neutrality and neural developmental modularity that enable the incremental evolution of complex behaviours. The experimental system, which satisfies the requirements identified thus far, is described in chapter 4. The purposes of this system are to verify and extend the theory of evolutionary emergent system generation.

This system itself is the second major contribution of my work. Its results (chapter 5) demonstrate that this is a uniquely successful platform for studying biotic selection and emergence. Current methods of evolutionary statistical analysis are applied and criticised in chapter 6. These methods are extended in chapter 7, resulting in a more confident qualification of unbounded evolution. These extended methods constitute the third major contribution in this thesis. Chapter 8 provides the fourth by summing up what has been learned from the system.

1.5 Published work

The majority of the theoretical basis (chapter 3) was published in (Channon and Damper, 1998c), which specified what evolutionary emergence is and evaluated existing artificial and natural selection systems. This paper includes my critique of Tierra (section 3.4.1). The remainder of the theoretical basis appeared in (Channon and Damper, 1998b), which focused on the developmental requirements. These papers include an overview of the experimental system, and (Channon and Damper, 2000) provides a more detailed description, similar to that found in chapter 4. Between them, these papers also cover the results reported in chapter 5. Population-level behaviours were discussed in terms of rudimentary social-intelligence in (Channon and Damper, 1998a). Evolvability requirements for open-ended evolution were outlined in (Channon, 2000), based on conclusions that can be found in chapter 8. The evolutionary statistical analysis from chapter 6 was published in (Channon, 2001). I plan to have the analysis from chapter 7 published in a suitable journal as soon as possible.

Chapter 2

Incremental artificial evolution

Genetic algorithms (GAs) are biologically inspired search procedures initially developed by Holland (1962, 1975, 1992) in the early 1960s – although see also Fraser (1957), Fogel (1962) and Fogel, Owens and Walsh (1966) for other evolutionary algorithms' roots. GAs evolve an initial random population of genomes (codings for solutions to the problem in hand) by selecting which individuals are reproduced and which are replaced. This is done by evaluating each solution's fitness via some function relevant to the problem and favouring the fitter solutions. Reproduction typically involves both crossover, whereby parent genomes are split into sections at common (randomly chosen) cut points and the new genome inherits corresponding sections from one parent or the other, and mutation, which involves randomly altering a small proportion of the new genome. Mutation increases diversity and crossover combines beneficial discoveries. There are many variations on the typical GA, but most share this base description.

2.1 Harvey's SAGA theory

Although most GAs work on populations of solutions with a fixed size and structure, the evolution of increasingly complex entities requires us to evolve variably sized subjects over many generations. Harvey's (1993b) Species Adaptation Genetic Algorithm (SAGA) theory provides a framework for incremental artificial evolution. In this paradigm, a population (with possibly just a few tens of members) evolves as nearly-converged species, for thousands or millions of generations.

2.1.1 The need for correlated fitness landscapes

The increases in complexity must therefore result from the evolution itself. This is in contrast to the common use of the genetic programming (GP) paradigm (Koza 1990, 1992), for example, where a population of millions may be evolved for less than a hundred generations (Harvey, 1997b, section 5). In the GP case, recombination effectively mixes the random initial population, exhausting variation in few generations. Because genetic codings of computer program instructions result in rugged (uncorrelated) fitness landscapes (i.e. mutating a bit in the genotype of a fit program will almost certainly produce a very unfit program), there can be little further evolution of this converged population. Here we see one of the requirements of SAGA: a sufficiently correlated fitness landscape (actual or implicit). Mutation must be possible, at a low rate, without dispersing the species in genotype space or hindering the assimilation by crossover of beneficial mutations into the species.

2.1.2 Variable length genotypes and converged populations

The open-ended evolution of increasing complexity cannot, of course, be achieved with fixed-length genotypes. Harvey (1992) states the case for gradual changes in genotype length (sections 2–6). First, he reports some theory due to Kauffman and Levin (1987). In an adaptive walk on a completely uncorrelated landscape, with "fitness achieved" defined as the highest encountered, each step up the fitness rank will (be expected to) take twice as long as the previous step, there being only half the (expected) number of fitter neighbours. This result still holds if in each time step a large population (of fixed number) samples different mutants, with the population moving as a whole to the fittest. Harvey (1992) makes the point that the result holds on correlated landscapes for a “long jump … defined to be the equivalent of several *simultaneous* mutations, long enough to jump beyond the correlation lengths in the landscape” (section 5). Thus, the argument proceeds, such long jumps will play less and less of a beneficial role as evolution progresses. After an initial period of fluctuation, only small jumps (such as individual mutations) will be beneficial[1]. Harvey notes that this is not connected to the punctuated equilibria controversy; only a single large step is ruled out, not a cascade of small steps that could be rapid in geological time.

Harvey (1993a, section 6.7) provides an accurate tail of this argument. A change in genotype length which causes the information content of the genotype expressed in the phenotype (GIP) to change can also be considered as a mutation in the above argument. Therefore, such changes would only be beneficial if made in small jumps. Because detrimental mutation must not be so high as to disperse members away from existing fit genotypes, the argument is now complete for a low rate of change in the information content of the GIP, both by direct genetic mutation and indirectly through changes in genotype length.

If this rule – of a low rate of direct genetic mutation and a low rate of change in genotype length with respect to effect on information content of GIP – is followed, then the population will evolve as nearly-converged species, convergence being in terms of GIP information content. If, in addition, a direct genetic coding (i.e. a bijection between genotype and phenotype) is used, then the population’s members will have an almost uniform genotype length that increases in small steps. Even if this is not the case, the information content of the GIP will increase (at most) gradually.

As made explicit by Harvey (1992, figure 3), the convergence of the population need not be around a single species. The possibilities of splitting into separate species and of extinction exist. Variation in number of species is not engineered in, but rather is a result of this theory. A new species arises (emerges) when a progenitorial species splits into separate ones. A species becomes extinct once all its members have died.

[1] There is some similarity between this and Fisher’s argument (Fisher, 1930, chapter 2) that probability of improved adaptation drops off rapidly as magnitude of change [mutation] increases.

Note also that the ‘problem of premature convergence’ (onto a local optimum) from traditional GA theory is now irrelevant.

2.1.3 Maintaining convergence by balancing mutation and selection

Selection and recombination serve to concentrate a population. A low rate of mutation will cause the converged population to hill-climb to a local optimum, with no further search after that (although see 2.2 below); a higher rate will enable the population to search the neighbourhood against a fitness gradient, but too high a rate will disperse the population completely. So a balance should be maintained between mutation and selection. (Recombination is assumed to be at a fixed rate.) This is the problem of “Muller’s ratchet” (Maynard Smith, 1978).

Harvey (1993b, section 6) states a result from molecular evolution theory, due to Eigen and Schuster (1979): that for an infinite population, assumed selective forces are just balanced by a mutation rate per base of

$$m = \frac{\ln \sigma}{l}$$

where l is the genotype length and σ is the *superiority* parameter — the factor by which selection of the master sequence, the local optimum around which the population is converged, exceeds the selection of the rest of the population. Thus for typical values of σ between 2 and 20, this upper limit for mutation is between $0.7/l$ and $3/l$.

Using the work of Nowak and Schuster (1989) on this upper limit or “error threshold” for finite populations, Harvey (1993b, section 6) reasons that “for genotypes of length order 100, and populations of size order 100, the error threshold will be extremely close to that for an infinite population”. Further, later in that section he reports that “in all simulations of hill-crawling where different mutation rates are tried, a typical U-shaped curve is found, giving the shortest time needed to reach another hill at around the mutation rates suggested by theory”. This despite the fact that tournament selection was used — significantly different to the assumptions of the molecular evolution theory.

In summary, the best mutation rate lies within a constant range *per genotype*, (nearly) independent of genotype length. However, this should be multiplied by the level of redundancy in the genotype.

2.2 Neutral networks and genetic drift

In his recent work, Harvey (1997a) has emphasised that ‘neutral networks’ in genotype space (connected networks of genotypes with only fitness-neutral differences) often play an important role in evolution. Where there is a very-many to one mapping from genotypes to phenotypes, genotype space can be percolated by neutral networks of genotypes which can be traversed through single mutations yet code for the same phenotype (Huynen, Stadler and Fontana, 1996). This can be useful when some parts of the genotype are ‘junk’ in some contexts but not in others. For example, when evolving neural networks, subnetworks may be masked or eliminated by the presence or absence of other connections. Hence, Harvey (1997a) argues that much of evolution is spent searching around the neutral network associated with the fittest phenotype present, with negligible possibility of becoming trapped in a local optimum. When a fitter neutral network is reached, the population will ‘jump’ onto it, in a form of punctuated equilibrium.

2.3 Analysis of an example experiment

Harvey (1997a, section 8) provides a relevant summary analysis of the evolutionary pathway of one of Thompson’s (1996) hardware evolution experiments. Thompson had evolved the connections of a reconfigurable chip being evaluated at a signal-recognition task. Principal component analysis of the genotype centroid of the population shows a process of random drift, clearly demonstrating that evolution used the neutral networks resulting from the massive redundancy of the genotype to phenotype mapping.

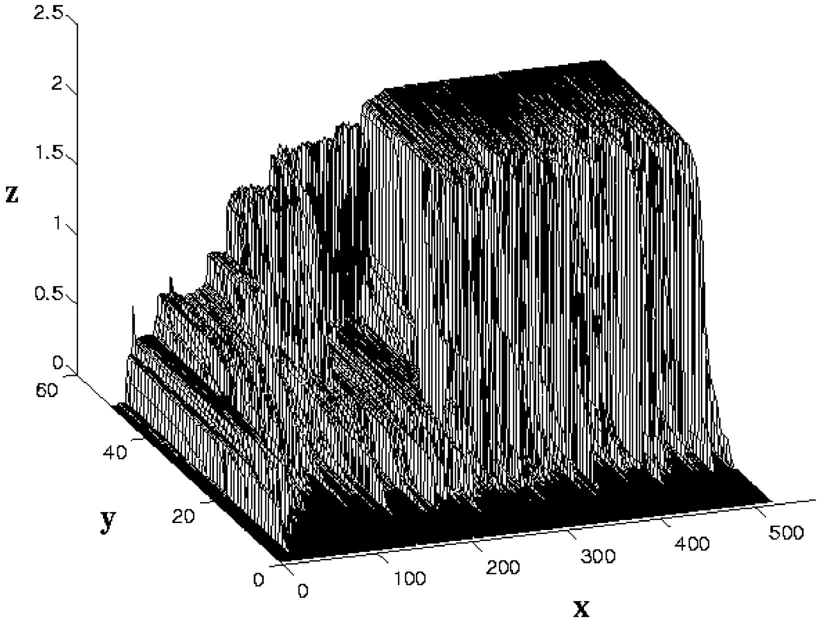


Figure 2.1: Fitness values (z-axis) for each generation (x-axis: 1 unit = 1 snapshot, taken every 10 generations), sorted along the y-axis, from Thompson's hardware evolution experiment. From Harvey (1997a, figure 14).

Figure 2.1 shows how the fitness values, with noise, of the population's members evolved. Harvey points out the (high) plateau after generation 2660, which indicates that a high level of neutral mutation is possible at this late stage. He then draws attention to the narrow ridge leading up to this plateau, with most of the population much less fit. This period matches the decrease in genetic convergence between generations 2000 and 2660, shown in figure 2.2. He states that "This period corresponds to the increase in genetic diversity in the population" (Harvey, 1997a, section 8). I take his use of the word *diversity* as opposed to *divergence* (which is clear from figure 2.2) to imply the assumption that the population remains centred in a single region of genotype space. I would like to offer a different analysis.

As shown in figure 2.3 and stated in Harvey (1997a, section 7), 60–70% of the final genotype was redundant. Harvey (1997a, section 7) reasons that therefore "the

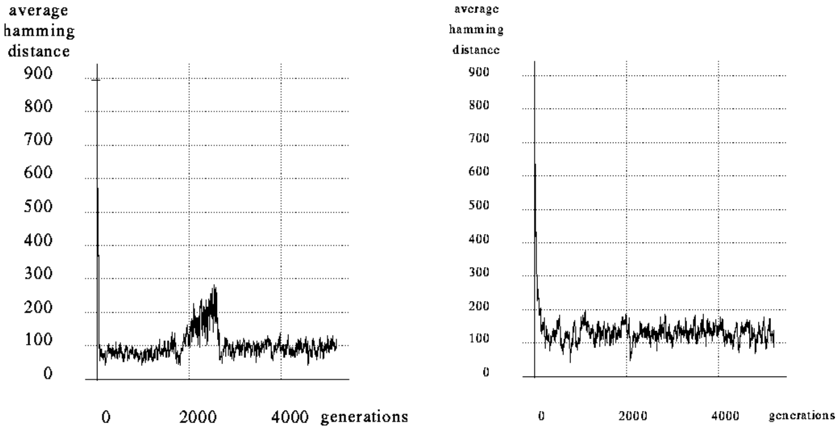


Figure 2.2: Genetic convergence (average Hamming distance between pairs of genotypes). Left: from Thompson's hardware evolution experiment. Right: with fitness randomly allocated. From Harvey (1997a, figure 12).

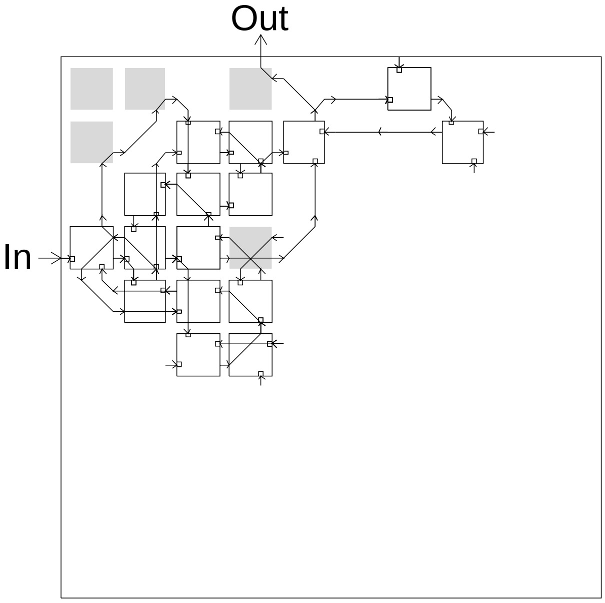


Figure 2.3: The functional subset of the genetically specified field programmable gate array (FPGA) after 5000 generations of Thompson's hardware evolution experiment. From Harvey (1997a, figure 10, right).

mutation rate used of some 2.7 mutations per whole genotype corresponds to about one mutation per *functional, non-junk* proportion of the genotype". This would put the mutation rate in line with the theory of section 2.1.3 above. However, as he also states, "Some areas of the chip may be unused in the successful configuration shown, but in earlier 'ancestor configurations' they would have had an effect on the chip behaviour" (1997a, section 7); thus the mutation rate used could have been too high earlier on in evolution, as I shall argue.

In light of the above considerations, I believe that figures 2.1 and 2.2 tell the following account. The initial series of short (along the generation [*x*-]axis) plateaus shows the existence of neutral networks, with the population phenotypically converged. The width (along the population [*y*-]axis) of these plateaus shows the level of neutral mutation possible, corresponding to the level of redundancy in the genotypes. These widths narrow as the population's fitness becomes less plateau-like, until generation 2000 when hardly any neutral mutation is possible, indicating that little of the genotype is redundant. Note that up to this point, the genetic convergence has remained approximately constant (after initial convergence – see figure 2.2).

Around generation 2000, the mutation rate (2.7) is clearly too high and passes through the 'upper limit for mutation' discussed in section 2.1.3; most of the population is dispersed over genotype space, then quickly converges to a new point in the same way that the population quickly converged at the start of evolution (and to a similar fitness plateau). However, because the strategy of elitism is used, the ridge between generations 2000 and 2660 exists. Once the non-elite bulk of the population has converged (just after 2000 generations), initially to a genotype with a low Hamming distance from the ridge genotypes, figure 2.1 shows that it evolves along a neutral network (until generation 2660), drifting away from the ridge genotypes.

Indeed, if we assume that there are at most two converged groups (one kept by elitism and the other a SAGA species), with the rest of the population being crossovers between them, then we can show that the distance between them must be high.

Let *h* be the Hamming distance between the two converged groups and assume that the divergence of each of these groups stays below a maximum average Hamming distance b (such as 150) – see figure 2.2 outside the generation range (1900,2700) for justification. We will read the average Hamming distance of the population to be h_{POP}, from figure 2.2. Then the worst case scenario for h (that is the case producing the lowest value) would be for all the crossovers to be at the extremes: R (say) members at the ridge group and $(50-R)$ at the other.

In such a case,

$$\begin{aligned} h_{POP} &< \frac{1}{50^2}\left[\left(R^2+(50-R)^2\right)b+2R(50-R)(h+b)\right] \\ &= b+\frac{2R(50-R)}{50^2}h \end{aligned}$$

$$\text{So,} \qquad \begin{aligned} h &> \frac{50^2}{2R(50-R)}(h_{POP}-b) \\ &> 2(h_{POP}-b) \end{aligned}$$

From figure 2.2 we can read that $h_{POP} > 275$ at generation ≈ 2600 and $b < 150$. Thus, given the above assumptions, we have a lower bound of 250 for the distance between the ridge group being kept by elitism and the converged SAGA species.

There is one issue from this experiment that remains to be resolved. How did the jump from the ridge and low plateau to the high plateau after generation 2660 occur? While only a more detailed analysis of the data could answer this, by providing a possibility that fits the given information, I can show that the above ideas need not contradict the data. Figure 2.3 shows that the final functional circuit is in the upper left corner of the FPGA, where the input and output are. To human designers, this seems logical because one would start in the region of the input and output and move out to use more of the chip as the design increased in size, producing a better design with each step. However, evolution was operating on the whole chip and was not trying to improve each part as time progressed; only the fitness of the whole chip was ever evaluated. Thus, though perhaps a little odd to us at first, it is possible that the rest of the chip could have had a negative effect on the fitness, which was neutralised either by a mutation in a ridge member or, more likely, by a crossover between a ridge member and one from the species drifting along a neutral network. For example, such a crossover could have broken a crucial connection at the bottom of the functional circuit shown in figure 2.3, or the non-ridge species could have drifted far enough from the ridge genotypes that the offending area of the circuit was no longer detrimental when crossed with a ridge member. Thus the final form of the circuit would have arisen – the fittest in the population and with huge genotypic redundancy, producing the high plateau.

2.4 Freeing evolution

Whichever of the above two accounts is true, a crucial tenet is being strengthened. As Harvey (1997a, section 9) puts it, "There is a lot more to evolution than meets the eye, and naive models and metaphors may lead to poor decisions in the design of evolutionary algorithms, or prejudice against reasonable decisions". I believe that a necessary development for incremental artificial evolution is an increase in evolution's freedom, rather than constraining it to naive models as GA work often does. For example, perhaps allowing mutation rates to evolve with their subjects would be a better route than trying to formulate how they should change; if the problem specified by the fitness function of the above example experiment had not had such a small highly fit solution, then it is unlikely that such a fitness could have been achieved with the mutation rate used. And most importantly, freeing evolution from necessarily simple fitness formulae could allow the evolutionary emergence of truly complex systems.

2.5 Related work

Although the most solid foundation for incremental artificial evolution, SAGA theory is not on its own in the field. Many of the ideas stem from work on natural evolution, and a few artificial evolutionists have independently reached similar conclusions. However, I am not aware of any such literature that pre-dates Harvey's.

2.5.1 An “anti-adaptationist” approach

Juric (1994) reasons for what he calls an “anti-adaptationist” approach, which is essentially a re-think of traditional GAs such that they are used not as optimisers but rather as “satisficers” (1994, section 3.1), operating over a long time-frame. He also recognises many of the other issues which surround SAGA theory and its application, including: the value of ‘junk’ DNA and random genetic drift (compare section 2.2 above), robustness (compare Jakobi 1997; Thompson 1997) and application to evolution in dynamic and interactive environments (compare Cliff, Husbands and Harvey 1993), overcoming the lack of scalability of traditional GAs (compare Channon 1996), and changes in the functional roles of genes (compare Channon 1996). However, Juric (1994) neither argues for the evolution of a nearly-converged population nor reports any actual evolutionary work.

2.5.2 Tracking multiple adaptive peaks

Gattiker and Wilson (1996) developed a genetic algorithm capable of distributing its population on a number of non-adjacent phenotypic peaks and, through genetic drift and the population’s own distribution, capable of tracking an environment where the adaptive peaks are changed in a recurrent manner.

A many-to-one genotype to phenotype mapping was randomly allocated at the start of each run and a subset of the phenotypes allocated as fitness peaks. This results in a highly epistatic genotype to fitness mapping, for “there is no information to the system about what a change in a genotype will do to the fitness. The only information is that a change, large or small, is as likely to encounter a peak as the ratio of peaks to phenotypes” (Gattiker and Wilson, 1996, section 2.4). Fitness scaling, in which the fitness of a member at a phenotypic peak is proportional to the number of members at that peak, was used to prevent the population from converging to a single peak. Monte Carlo (roulette wheel) selection with a non-zero minimum ‘background’ fitness was used to allow genetic drift. This constant background fitness was set such that it would be significant if the population was very unevenly distributed over the peaks but relatively low for an even distribution.

A first set of experiments showed that the system was capable of distributing the population over a number of non-adjacent phenotypic peaks. Analysis of these experiments showed that they were achieving this through the use of "Lossy-crossover-closed sets" (LCCSs) (Gattiker and Wilson, 1996, section 3.1). An LCCS is a genotype set within which most, but not all, crossovers generate members of the same set and which is stable within the specific GA. Final populations in these first experiments were LCCSs that contained genotypes with phenotypes evenly distributed over the peaks. Whenever crossover produced a genotype with just background fitness, it would be selected out as this fitness is relatively low for an even distribution. Leading up to the final population, when the phenotypes were not so evenly distributed over the peaks, the background fitness would have been significant, allowing genetic drift to redistribute the population more evenly over the peaks.

In a second set of experiments, the phenotype peaks were swapped in a recurrent (periodic) manner. The GA was able to evolve an adaptable population capable of tracking the peaks. Analysis showed that this was achieved by an LCCS population containing all the possible peaks. This population is stable at any stage but also " 'sends out' offspring to non-peak areas due to the action of crossover ... 'directly sensing' the peaks, or generating non-viable offspring" (Gattiker and Wilson, 1996, section 5.1). Such offspring would have just a background fitness except just after the peaks are swapped, when they provide the new base for a fit population. The population's distribution itself contains a "memory ... about past aspects of the environment" (Gattiker and Wilson, 1996, section 6).

How are these ideas relevant to incremental artificial evolution? Clearly the issues of convergence and genetic drift have been addressed, although sufficient conditions were artificially imposed on the system. The main relevance is that the work deals with the evolution of multiple phenotypic convergences – multiple species and species with multiple phenotypes. Although most work to date relating to SAGA-like species has only been concerned with the evolution of one species, my work necessarily involves the divergence of species – see chapter 5. Further, theory concerning the ‘memory’ inherent in populations’ genetic distributions is thin on the ground but clearly relevant to long term evolution, linking in with the issues of neutral networks and ‘redundancy’ of genetic information. Gattiker and Wilson’s (1996) paper adds to the debate. However, it should be noted that their experiments solved only fairly trivial problems, using genotypes of just 10 bits each. As noted in section 2.4, we would be well advised to found our concepts in actual evolution rather than premature theory.

2.5.3 A suitable genetic decoding technique

Boers and colleagues (Boers and Kuiper, 1992; Boers, Kuiper, Happel and Sprinkhuizen-Kuyper, 1993) based their genetic decoding method on the decoding of natural DNA into amino acids. The key feature that makes this suitable for incremental evolution is that developmental rules are read from each base position on the genotype. For open incremental evolution, genotype lengths must be able to increase. By reading the developmental rules (or other basic blocks) from each base position, we leave our systems free to insert genetic information without destroying the relationship between the existing genes and the phenotype (except perhaps when the insertion splits an active genetic string). Note that this is not to say that we would want to insert arbitrary lengths; there is a good case for the gradual increase in lengths so that crossover will not lose genetic information. This is the basis of the decoding technique I have used in my work.

2.5.4 Functional validity

One further issue worth clarifying is that of functional validity with respect to the destination environment, or “situation within a world”. Brooks (1991a, 1991b) argues that incremental development (including evolution) must take place within the environment that the objects inhabit. This is to avoid problems (common in traditional artificial intelligence) caused by a divide between a system and the real world. So, for example, some researchers (e.g. Harnad 1993) argue that robots intended to inhabit the real world must be evolved (or at least frequently evaluated) in it. However, if organisms are only ever to inhabit an ‘artificial’ environment then there should be no concern about them being evolved in that environment. Their ‘world’ is not a simulation and so the approach suffers none of the problems that occur when trying to use a simulation to evolve robots for the real world. Where the artificial environment differs from our world (however greatly), there is no problematic error. There is simply a difference.

Chapter 3

Theoretical basis

3.1 Evolutionary emergence

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

Cariani identified the three current tracts of thought on emergence[1], calling them "computational", "thermodynamic" and "relative to a model". Computational emergence is related to the manifestation of new global forms, such as flocking behaviour and chaos, from local interactions. Thermodynamic emergence is concerned with issues such as the origins of life, where 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 behaviour. This is close to Steels' (1994) concept of emergence, which refers to ongoing processes which produce results invoking vocabulary not previously involved in the description of the system's inner components – "new descriptive categories" (section 4.1).

[1] See (Damper, 2000) for a more recent overview and history of the range of thought on emergence.

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

Langton (1989) gave a simple, compatible method of ascribing emergence: "The essential features of computer-based Artificial Life models are: … There are no rules in the system that dictate global behavior. Any behavior at levels higher than the individual programs is therefore emergent" (pp. 3–4). Note that this can be used for both the computational and relative to a model senses of emergence. He also stressed (p. 41) the importance of nonlinear systems – those which do not obey the superposition principle (i.e. which cannot be understood in terms of independent constituent parts), where it is necessary to understand the interactions between the parts. Thus, new descriptive categories cannot be invoked from a system which obeys the superposition principle.

Having specified what is meant by evolutionary emergence, I will now explore the two types of selection which might be used to bring about such emergence. Packard (1989) referred to these as "*extrinsic* adaptation, where evolution is governed by a specified fitness function, and *intrinsic* adaptation, where evolution occurs 'automatically' as a result of the dynamics of a system caused by the evolution of many interacting subsystems" (p. 141). I will use the terms artificial and natural selection respectively, because the first involves the imposition of an artifice crafted for some cause external to a system beneath it, while the second relies solely on the innate dynamics of a system. Ray (1998, section 2.1) is one of the better known personalities trying to bring an awareness of the difference between artificial and natural selection to the many practitioners in the artificial evolution field who claim to be using natural selection when they are in fact using artificial selection.

3.2 Artificial selection in artificial systems

Holland did not originally envisage GAs as functional optimisers, but rather as processes similar to natural adaptive systems. In the natural world, organisms interact in complex ways and so coevolve with their environment, which includes other organisms. However, GAs proved suitable for a range of optimisation tasks and this has grown to be their most widespread application (Goldberg, 1989).

3.2.1 The state of artificial selection work

Within the artificial evolution field, variants of the optimisation paradigm have proven fruitful. Even where the concepts of SAGA theory (section 2.1) are dominant, practice still holds to the use of fitness functions. But as the complexity of behaviours under consideration increases, flaws in the artificial selection approach are appearing. Zaera, Cliff and Bruten's (1996) failed attempts at evolving schooling behaviour in artificial fish provide an account of the difficulties faced:

> "The problem appears to be due to the difficulty of formulating an evaluation function which captures what schooling is. We argue that formulating an effective fitness evaluation function for use in evolving controllers can be at least as difficult as hand-crafting an effective controller design. Although our paper concentrates on schooling, we believe that this is likely to be a general issue, and is a serious problem which can be expected to be experienced over a variety of problem domains."

Zaera et al. considered possible reasons for their failure. The argument which most convinced them was that real schooling arises through complex interactions, and that their simulations lacked sufficient complexity (their section 5). They cited two promising works: Reynolds' (1992) evolution of coordinated group motion in prey animats pursued by a hard-wired predator, and Rucker's (1993) ecosystem model in which Boid-like animat controllers (or rather their parameters) were evolved. Both of these are moves towards more intrinsic, automatic evolution.

The use of coevolutionary models has become a popular approach in the adaptive behaviour field. This is essentially a response to the problems encountered when trying to use artificial selection to evolve complex behaviours. 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 and Dawkins and Krebs 1979), an early example of which is Hillis' (1990) 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 optimisation procedure continue to take us closer to real biological systems" (p. 233).

As with Hillis' paper, the reason 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 (1994) of Boids fame worked towards more automatic evolution by coevolving simulated mobile agent controllers which competed with each other in games of Tag. This eliminated the need to design a controller in order to evolve a controller, as in his previous work (Reynolds, 1992) mentioned above.

3.2.2 (No) Emergence via artificial selection

From the above discussion, one might assume my argument to be developing toward the extreme statement that evolutionary emergence is not possible in a system using artificial selection. This is not quite so, although I do argue that artificial selection is not sufficient. I shall now give an example of emergence from a genetic algorithm which Ray (1998) would classify as using "partial natural selection" (section 2.8), in that the interactions between artificial entities play a role commensurate with the artificial fitness function aspect of selection.

Sannier and Goodman (1987) used a distributed GA to evolve genomes within a two-dimensional toroidal grid containing "food" which is placed into the environment according to some pattern. An individual's "strength" (fitness), which is deducted from its parents' strengths at birth, increases on consumption of food and decreases in each time step (and upon reproduction). An individual is reproduced if its strength is above a threshold, and killed if its strength drops below a lower threshold. A genome encodes rules which allow it to move in eight directions (N, NE, E, SE, . . .) with program branching conditional on the presence of food in the eight neighbouring locations. Thus, the individuals can interact (only) by moving around and consuming food, so affecting each other's program branching.

In the experiment reported, food was restricted to two farm areas, spaced apart in the toroidal world. The level of food introduced into the farms was varied periodically. When one farm was having its summer, the other would be having its winter. A farm's potential was set lower the more it was either over-consumed or neglected (under-consumed) during the previous period.

Two classes of individual emerged: *farmers* who stayed put in one of the farms, their populations rising and falling with the seasons, and *nomads* who circled the world, moving in such a way that they passed through both farms during their summers. The nomad population increased as it went through a farm and decreased as it moved through the area without food. Notice the new descriptive categories: farmer and nomad. Groups of individuals from each category were extracted from the total population and tested in the absence of the other category. While farmers could survive without nomads, it was found that nomads needed farmers so that the farms would not be neglected between visits.

The important feature is the emergence of the two classes of individual. Never was it specified that they should arise. Evolution produced them because they perform better than other genomes within the environment. The previous paragraph demonstrates the need to update our model to include the new descriptive categories.

It would, of course, be erroneous to claim that this is an example of emergence via artificial selection. In this partial natural selection system, the artificial component of selection is incidental to the emergence, the source of which is the natural component of selection arising from interactions of the system's parts. The statement that "artificial selection is not sufficient [for evolutionary emergence]" (from the first paragraph of this subsection) does not imply a necessity for (or even benefit of) artificial selection. *In the context of evolutionary emergence, any artificial selection used constitutes just one of the parts of a system.*

In summary, artificial selection can only select for that which is specified. Therefore anything that emerges during evolution (in the evolutionary emergence sense) *must* result from another aspect of selection, which must in turn arise from the innate dynamics of the system – natural selection.

Artificial selection can result in evolved solutions that an experimenter had not anticipated. For example, a highly fit solution might only use a fraction of the genotype that had been made available and thought necessary, or a solution might exploit properties of the phenotype that had previously been unknown. However, examples such as the former do not require an extension to the observer's model, and examples such as the latter require an extension that is the product not of evolution, but of the observer's lack of knowledge. Neither qualifies as evolutionary emergence by my definition.

3.3 Biotic and abiotic selection

The distinctions between artificial, abiotic-natural and biotic-natural selection are clear in the real world. Artificial selection is induced by conscious human selection (although both the affected and the direction of change need not be as intended), natural selection is not. Biotic selection is induced by biota (living organisms), abiotic selection is not. However, the Sannier and Goodman example demonstrates that these differentiations require clarification for artificial evolutionary systems, which have their construction entirely specified by humans, and in which the distinction between biota and abiota may not be immediately clear.

First let us revisit the term ‘selection’. As covered in section 1.1, selection is an abstraction of the statement “Some heritable variations increase the probabilities of survival and reproduction, or rather of further inheritance to subsequent generations, within the context of the current environment”. So the set of factors that effect these probabilities constitute selection, and each can be considered a selection pressure on the entity in question. A selection pressure can be either a factor of the entity itself, or a result of potential interactions with the environment, which includes other entities.

Now consider these entities, the primary ‘units of selection’. Dawkins (1976, 1983) argues that we should think of genes as the units of selection. This departs from Darwin’s view of organisms as the units of selection, and is much disputed. My own reasoning is that selection governs the probability that an organism will pass on its heritable variations, which include the variations of genes that it has inherited. This indirectly governs the probability that a gene will pass on its heritable variations: the variation of nucleotide sequence that it has inherited. For example, the removal of a gene requires that the number of organisms with that gene drop to zero. Arguments for higher-level structures, such as species, as possible units of selection can be similarly dismissed: the removal of a species requires that the number of organisms within that species drop to zero. So, in line with Darwin’s argument, organisms are the units of selection, not genes or species.

In an artificial system, it may be that not all ‘organisms’ are subject to evolution. For example some might have only a fixed, pre-specified behaviour. Some might be programmed to survive forever. As life is a product of evolution, it makes sense to term such individuals ‘abiotic organisms’, and reserve the classification of biota for organisms that are subject to evolution. Extending this argument, in an artificial system, even a biotic organism can have some fixed (inter-)actions. So we can divide organisms' actions into abiotic (fixed) actions and biotic (evolvable) actions. This division is consistent with Darwin's argument that biotic competition is the cause of any (sustained) evolutionary progress, through the cycle of competitive displacement (see section 1.1).

In artificial systems, and arguably also in the real world, all artificial selection is abiotic: the selection is not induced by other biota. Indeed, in artificial systems, the distinction between (abiotic-)artificial selection and abiotic-natural selection is somewhat arbitrary, depending only on whether or not the selection was consciously intended by the system's designer. It makes sense to simply group all such selection into a single classification: abiotic selection. For example, selection induced by Reynolds' hard-wired predator (see section 3.2.1) could be considered to be artificial selection, on the grounds that it was consciously intended by the system's designer, and/or abiotic-natural selection, on the grounds that it is induced by a part of the system that is not subject to evolution. So the important division is not, in fact, between artificial and natural selection, but between abiotic and biotic selection. Further, using the above definitions, this division is now well defined for artificial systems. *Any selection resulting from a part of the system (for example a segment of code) carrying out a biotic (inter-)action is biotic selection. All other selection is abiotic.*

Note that selection can be caused by an abiotic interaction that uses the results of biotic interactions, such as Reynolds' fixed fitness function (abiotic interaction) which uses calculations from games of Tag that involve biotic interaction. I consider such selection to be divisible into abiotic selection, from the fitness function based segments of code, and biotic selection, from the game of Tag segments of code. Note that in this, and most other existing coevolutionary artificial systems, abiotic selection has been used to cause biotic interactions (which could be run without causing a selection pressure) to give rise to biotic selection. Such systems run the risk that abiotic selection dominates (suppresses) biotic selection. This issue is revisited later in this chapter.

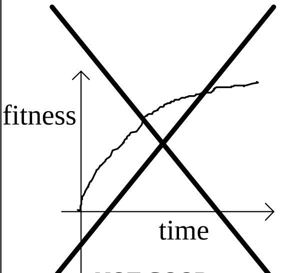
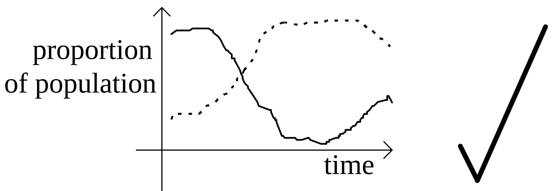
GAs for function optimisation	Natural Evolution
 NOT GOOD FOR EMERGENCE	Example: if one of two species which share resources improves sufficiently, the other either improves or becomes extinct  GOOD FOR EMERGENCE (DO NOT HAVE TO SPECIFY WHAT IS GOOD)

Figure 3.1: The difference between abiotic and biotic selection.

3.4 Biotic selection in artificial systems

Biotic selection retains the fitter variants of biota (those that persist through survival and reproduction) without any explicit specification of what is required to be fit, which depends on the biota and so changes as the system evolves. This feedback in the generation of selection pressures is the vital factor missing from purely abiotic selection systems (figure 3.1).

As noted in section 2, genetic codings of computer program instructions result in rugged fitness landscapes and this makes them unsuitable for incremental evolution by artificial selection. One would expect this argument to carry through to biotic selection systems, where fitness is an abstract concept or external measure. However, the approach of most biotic selection work to date has been to evolve program code, following the initial success of Tierra (Ray, 1991) which demonstrated incremental evolution over millions of reproduction cycles. Despite a lack of continuing emergence, this early success needs to be explained against the argument that computer programs are not suitable for incremental evolution.

3.4.1 The evolution of program code via biotic selection

Tierra is a system of self-replicating machine code programs. As an evolutionary biologist, Ray was interested in comparing artificial evolution with that in the real world. To make evolution possible, a certain rate of random bit-flipping was imposed on the memory within which the population of running code resided. Each program was allowed to write to the block of memory it occupied but not outside that block. However, programs could read and execute instructions from any part of memory.

The population was initialised as a single manually-designed, self-replicating program. This program first examined itself to determine its length in memory, then issued an instruction to allocate free memory for a child, copied its code byte by byte to this free memory and finally issued an instruction to treat the child as an independent process.

A degree of abiotic selection was imposed by the system itself deleting the oldest programs in order to free memory when it was full beyond a certain threshold, with an added bias against programs that generated error conditions. As emphasised at the end of section 3.2, this artificial selection constitutes just one of the parts of the system. It does not (necessarily) prevent biotic selection.

Tierra was implemented as a virtual computer, allowing Ray to design a machine language with some properties suiting it to evolution. One aspect of this language was that it contained no numeric constants. This was to reduce the brittleness of the language by decreasing the size of the "real" instruction set, in which `add 1` and `add 2` are considered to be distinct instructions. Thus, direct memory addressing was not possible, in either a relative or absolute form. Instead, the manually-designed program began and ended with consecutive `NOP` (No-OPeration) instructions which acted as templates that could be found by certain machine code instructions which search memory backwards or forwards in a single step. This addressing by templates is how the program examined itself to determine the points at which to begin and end copying and so also its length.

Computational errors were introduced at random. For example, a left bit-shift instruction would sometimes shift a register's bits two positions, sometimes not at all. A copy instruction would occasionally copy to a neighbour of the correct location. Such errors could lead to genetic changes by affecting replication.

When Tierra was run, various classes of programs evolved. "Parasites" had shed almost half of their code, allowing them to replicate almost twice as fast; they replicated by executing the copy loop from neighbouring organisms, which could easily be found by template matching instructions as before. Because the parasites depended on their "hosts", they could not displace them and the host and parasite populations entered into Lotka-Volterra population cycles, characteristic of predator–prey and parasite–host systems (Lotka, 1925; Volterra, 1926). Ray reported that coevolution occurred as the hosts became immune to the parasites, which overcame these defences, and so on. "Hyper-parasite" hosts emerged containing instructions that caused a parasite to copy the host rather than the parasite; this could lead to the rapid elimination of the latter. Ray also reported cooperation (symbiosis) in replication followed by "cheaters" (social parasites) which took advantage of cooperators.

The above are examples of ecological adaptations, which involved interactions between the programs. Another class of adaptations found was optimisations, where individual programs replicated faster than their ancestors. For example, non-parasitic replicators almost a quarter the length of the initial replicator were found, as were programs with unrolled copy loops which copied two or three bytes per loop, reducing the overhead of looping. By adding split and join instructions, which allowed a program to split into a multi-threaded process and join back into a single one, the evolution of efficient parallel-processing replicators was later achieved.

While the results of Tierra are impressive, the system is not truly open-ended. There have been no new reports of emergent phenomena during the last few years and it is generally accepted that not much more will occur unless further alterations are made to the system, as with the addition of the split and join instructions. Indeed, Ray established a "biodiversity reserve for digital organisms" (1998, section 5.2) based on a networked version of Tierra, in an attempt to generate more complex organisms. His hope was that the increased scale would hold an ecological community of many species, with the network model providing initial selective forces resulting from its temporal and spatial complexity. “Once a significant impulse in the direction of complexity has occurred, the hope is that selective forces arising from interactions among the digital organisms can lead to an auto-catalytic increase in complexity” (1998, section 5.2.2). However, this system has not yet produced any such results – see (Ray, 1997) and (Ray and Hart, 1998).

I believe that the evolvability of the code stems largely from the template matching system, and that this could account for all of the ecological adaptations reported but would be of little use for much other than replication. To justify this statement, I considered (in Channon and Damper, 1998c) the following pseudocode of the initial, manually designed program:

```
T1111 a=address(T1110)+1 ; b=address(T1111) ; c=a-b
T1101 allocate memory for child, length c =>a=start
      call T1100 ; divide from child ;jump to T1101
T1100 push a,b,c onto stack             }
T1010 memory(a)=memory(b)               }   copy
      c-- ; if c=0 then jump to T1011   } procedure
      a++ ; b++ ; jump to T1010         }
T1011 pop c,b,a off stack ; return      }
T1110
```

This pseudocode is based on the initial program as listed in Ray (1992, appendix C). The `T????s` on the left denote four-bit templates (with the same bits as the real templates), which can be thought of as labels; the difference is that a jump or call instruction will search outwards in memory to find the nearest matching template. Now, as reported in Ray (1992, section 3.1.1), a parasite can be obtained by simply mutating one bit of the `T1100` template to produce `T1110`; this would then be the same as the end template, reducing the length to be copied (`c`), and the `call T1100` statement would search outwards in memory until it found a host containing the copy loop. Further, a host that is immune to such a parasite can also be produced by a single-bit template mutation of the initial program. To be more exact, a single-bit mutation in the template-comparison `1011` in the `if c=0 then jump to T1011` statement of the copy loop achieves this; by mutating it to `1111`, the program will re-evaluate its address and length after every reproduction. Thus, should a parasite try to use this program's copy routine, it will be copied just once but ever after that it will be copying the host; this host is also a hyper-parasite as defined above. In just two template bit-flips, we have reproduced the most impressive ecological results of Tierra! While the actual evolution might have taken a slightly different route, to slightly different programs, these phenomena clearly result from the flexibility of the template matching. It is also clear that the same argument applies to the other ecological adaptations reported.

Although this demonstrates the flexibility of template matching at the four-bit level, it also shows the ecological results to be somewhat less dramatic than we might have first hoped. For we must hold to our previous argument that programs are not suitable for incremental evolution; complex programs would necessarily contain many more templates (labels) and so these templates would have to be significantly longer. Thus we would pass far beyond the trivial four-bit templates that random search can operate on, to a stage where evolution is impractical. The above argument that programs are not suitable for (long-term) incremental evolution still stands. Note that I am not dismissing Ray's work as trivial or unimportant; I consider Tierra to be a significant milestone in the artificial evolution field – the first intentional example of *naturally arising* coevolution in an artificial evolutionary system.

There is still the issue of the optimisation adaptations to address. If programs are too brittle for (long-term) incremental evolution, then how was it that these adaptations evolved? Two of the results, unrolled copy loops and efficient replication by parallel processing, can be easily explained. All evolution needed to do was insert (probably by the action of the random computational errors) repetitions of neighbouring code; the local functionality was not changed and the efficiencies are from more-of-the-same solutions. These examples should not alter our perception of the brittleness in mutating code or inserting *different* instructions. The final results to be explained are the non-parasitic replicators almost a quarter the length of the initial replicator. Comparing the shortest self-replicating program (Ray, 1992, appendix D) with the initial program (appendix C) shows that the transition can be made by simply deleting instructions (mostly NOPs – cutting out the redundancy in the templates) and just six mutations, three of which are completely unnecessary. So apart from three mutations, this is a less-of-the-unnecessary solution – certainly not sufficient to challenge our argument.

Computer Zoo (Skipper, 1992), inspired by Ray, shared most of the features of Tierra and demonstrated very similar results. Skipper noted that remote execution is essential to the evolution of parasites and hyper-parasites. There have been many other Tierra-inspired systems, such as Avida (Adami and Brown, 1994) and Cosmos (Taylor and Hallam, 1997), all of which failed to produce significantly more impressive results. There are other possible explanations for this, including an inability on our part to detect further evolutionary emergence, and/or a deficiency in the language that lies not in its brittleness but in its lack of expressive ability within the system. However, I believe that the discussion presented here eclipses such possibilities, in that it provides a sufficient explanation.

One piece of work often cited as an example of evolving programs by natural selection is Koza (1993). I do not include a full discussion of it here because, although biotic selection is possible within the presented framework, the improvements reported result from artificial selection involving evaluation at a task. The only example of "emergence" reported was the existence of 604 self-replicating programs in a set of 12,500,000 randomly generated programs. There was no emergence resulting from evolution.

3.4.2 The evolution of more suitable entities via biotic selection

Although the approach of most biotic selection work to date has been to evolve program code, there are two notable exceptions. Both involve the evolution of neural networks, which are well suited to incremental artificial evolution because of their graceful degradation (high degree of neutrality). The second (chronologically) is the work described in this thesis, which was conceived (Channon, 1996) independently of and initially created in ignorance of both Tierra (including its derivatives) and the first exception: PolyWorld (Yaeger, 1993). Yaeger stated three motivations for his work, including the production of emergent complex behaviours and the exploration of artificial life as a route to artificial intelligence. These are (very) similar to the motivations behind the work presented here. However, his other motivation is what sets PolyWorld's design apart; it was "to create *artificial* life that is as close as possible to *real* life, by combining as many critical components of *real* life as possible in an *artificial* system" (Yaeger, 1993, p. 264). Hence PolyWorld simulates many aspects of the real world including energy conservation, food, movement (on a 2D plane), vision, neural networks and learning.

PolyWorld organisms have seven pre-programmed behaviours: eating, mating, fighting, moving, turning, focusing and lighting. These are expressed according to the activation levels of pre-specified neurons. An organism's chromosome determines its physiology (size, strength, maximum speed, green coloration, mutation rate, number of crossover points and life span) and some basic characteristics of its neural network. These characteristics include the number of neurons devoted to vision (in three groups: red, green and blue), the number of internal neuronal groups, a connectivity density (between groups) matrix and Hebbian learning rates. The neural networks are constructed stochastically.

PolyWorld is initially seeded with random genomes and run as a steady-state GA using an *ad hoc* fitness function until a population that maintains its number through mating is achieved. This occurred in the seed population of some runs, yet not at all in others. Yaeger reported a variety of emergent behaviours, including some trivial ones but also "fleeing", "fighting back", "grazing", "foraging" and "following". He also claimed that a drifting group of organisms and "one example of a few organisms [apparently] 'chasing' each other were even suggestive of simple 'flocking' behaviours"

(Yaeger, 1993, p. 285), although this seems somewhat speculative.

One criticism of PolyWorld, in the context of perpetual evolutionary emergence, is that (Hebbian) learning appears to be overwhelmingly responsible for the results. There is little evidence of significant *genetic* evolution; the genomes had very limited control over the small number of neural groups. It is conceivable that if comparison PolyWorld experiments were run with birth networks specified by random parameters (or perhaps constant parameters which result in large enough neural networks), then the same results might emerge. New organisms would either learn such that they become adapted to the evolving environment or die, and so evolution would occur without genetic change. Although this is valid evolutionary emergence, it is not sufficient for perpetuating evolutionary emergence. Unless learning is made evolvable, or what is learned can be passed on, a maximal level will be reached at which organisms are not capable of learning more in their lifetimes.

I consider PolyWorld to be an example of biotic selection work because, like Tierra, biotic selection has not been completely suppressed by abiotic selection. Both Tierra and PolyWorld include some abiotic selection (deletion of oldest programs in Tierra, the barely-implicit “energy” fitness function in PolyWorld) but in each case the results show that this is at a low enough level for biotic selection still to effect the population. PolyWorld is not alone in being an evolving artificial ecology (which I define as an artificial evolutionary system in which biotic selection is technically possible[2]) not based on the evolution of program code. However, in all other such artificial ecologies that I have been able to find (many of which are unpublished works), abiotic selection is such a strong force that we should expect (as a null hypothesis) it to block biotic selection’s feedback, preventing open-ended emergence. None of the results that have been reported from these systems contradict this. See for example (Axelrod, 1984; Hillis, 1990; Holland, 1994; Hraber and Milne, 1997; Kerce, 1993; Packard, 1989; Reynolds, 1994; Rucker, 1993; Sannier and Goodman, 1987; Ventrella, 1996; Ward, Gobet and Kendall, 2001). This is much like the way that selective breeding of animals

[2] Note that this definition excludes single-agent ‘ecologies’, such as that used in (Nolfi and Parisi, 1995).

blocks natural selection in the real world. If horses are selected for mating according to their race performance, then the result is (at 'best') a population of horses that can race fast. Biotic competition between the horses, which would normally result in significant selection pressures, is suppressed. Note that it is possible that abiotic selection does finally limit biotic selection in PolyWorld. However, significant emergent behaviours that appear to be due to biotic selection were reported (see above).

My own experimental system (detailed here in full but see also Channon, 1996; Channon and Damper, 1998b, 1998c for the development of the ideas behind it) shares many features with PolyWorld, despite having been created without knowledge of Yaeger's work. However, because it does not attempt to simulate aspects of the real world, it is considerably simpler. There is no learning in the neural networks, purposely to avoid the first criticism above. Further, it uses biotic selection, with no (or negligible) abiotic selection. There is no obvious implicit fitness function, such as the "energy" in PolyWorld, that might dominate selection. This is what is special about the system.

3.5 Developmental requirements

Biotic selection is necessary for evolutionary emergence but does not guarantee the evolution of evermore novel emergent phenomena. The question "what class of objects can/should we evolve?" needs to be answered with that in mind, along with the central aim: increasingly complex advantageous behaviours. Neural networks are the clear choice because of their graceful degradation (as noted in section 3.4.2) and suitability for this aim. But how should the network structure be specified?

The evolutionary emergence of novel behaviours requires new neural structures, or "modules". We can expect most to be descended from neural structures which once had different functions (Mayr, 1960). There are many known examples of neural structures that serve a purpose different from a previous use, for example Stork, Jackson and Walker (1991).

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 neighbours.

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 behaviours, a modular developmental process is called for. For the sake of research validity (regarding long-term goals), this should be included from the outset.

Most artificial neural networks (ANNs) that have been manually designed are layered feed-forward networks. However, recurrent networks can have internal state sustained over time and demonstrate rich intrinsic dynamics. This makes them attractive for use in behaviour-based work. Evidence from neuroscience provides further support, as biological neural networks are frequently recurrent. Although recurrent ANNs can be very hard to study (Boers and Kuiper, 1992, p. 40), artificial evolution should have no problem using them. Indeed, there seems to be little reason to constrain the evolution to feed-forward networks, especially when aiming for organisms that are to act as complex dynamical systems working within a time frame.

3.5.1 Gruau's cellular encoding

Gruau (1996) used genetic programming techniques (Koza, 1992) to evolve his cellular programming language code 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 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 modules.

So, with GP techniques, we are reduced to gene duplication for all but simple iterative structures. What is required is a rule of the sort `follow (rules X)` where `(rules X)` is a marker for (pointer to) rules encoded elsewhere on the genotype. But this would be difficult to incorporate into GP. A better route is to use a system capable of such rules.

3.5.2 Cellular automata

Many investigators have used cellular automata (CA) for the construction of neural networks, for example Gers and de Garis (1996) and Lee and Sim (1998). However, such work is more often at the level of neuron growth than the development of whole operational (rather than just large) networks. The working networks developed to date have been only basic. Although CA *rules* are suited to the evolution of network development in principle, the amount of work remaining makes this a major research hurdle.

3.5.3 Diffusion models

Although there are examples of work involving the evolution of neural networks whose development is determined by diffusion along concentration gradients, for example Vaario and Shimohara (1997), 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.

3.5.4 Lindenmayer systems

As mentioned above, developmental biology shows that genes provide a recipe for each cell to follow, and that the activation of relevant genes is determined by a cell's immediate environment. All cells use the same set of rules, derived from the genes.

Lindenmayer systems (L-systems) were developed to model the biological growth of plants (Lindenmayer, 1968). They are a class of fractals which apply *production rules* in parallel to the cells of their subject. A specified *axiom* subject (typically one or two cells) develops by repeated re-application of these rules. Each step in a cell's development can be determined by its immediate environment, including itself. In general, the most specific production rule that matches a cell's situation is applied.

Kitano (1990) used an L-system with context-free rules to evolve connectivity matrices. The number of rules in the genotype was variable. After each developmental step, the matrix would have doubled in both width and height. Kitano demonstrated better results than direct encoding when evolving simple ANNs (such as XOR and simple encoders) using training by error back-propagation. He also showed that the number of rules could be small.

Boers and Kuiper (1992) used an L-system with context-sensitive rules to evolve modular feed-forward network architectures. A fixed-length alphabet was used for the rules, restricting the possible network architectures but still producing some good results. The evolution of production rules used a conventional genetic algorithm, with fixed-length genomes initially randomised. A limit of six rewrite passes over the network string was imposed.

Both these works used back-propagation 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 back-propagation more than evolution. However, these works demonstrate the suitability of L-systems to non-iterative modular network development.

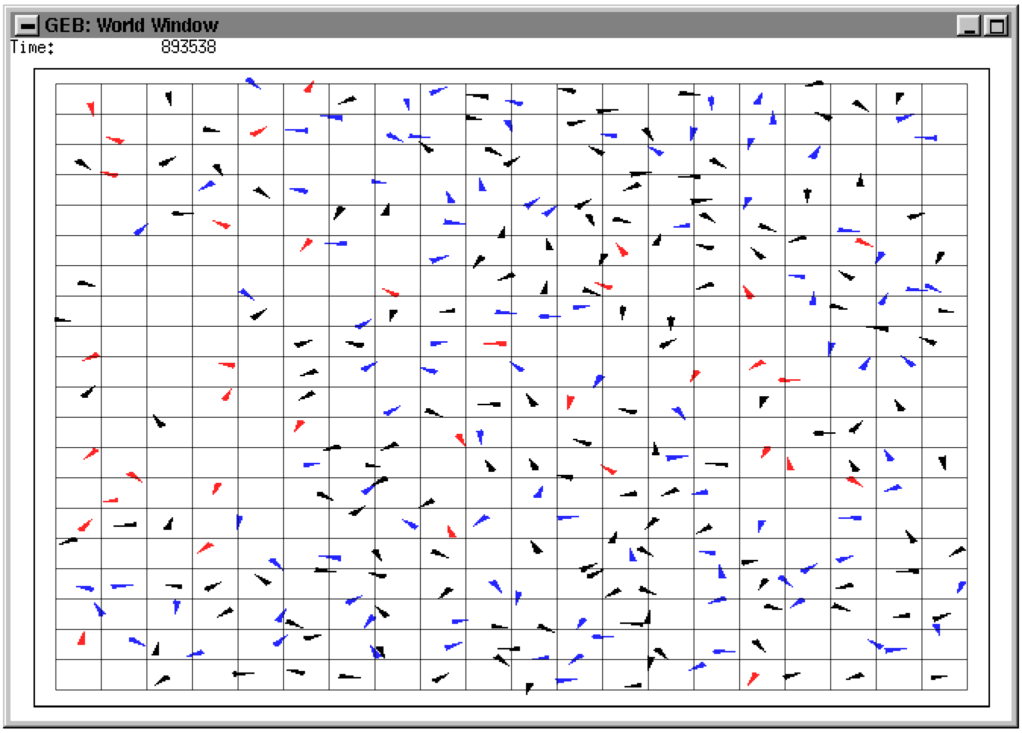
Chapter 4

Experimental system definition

So far in this thesis I have examined existing artificial evolutionary systems in order to uncover requirements for the emergence of increasingly complex advantageous behaviours. As discussed in the previous chapter, not one of these systems satisfies all of the requirements identified. In order to both verify and extend this list of requirements, I have constructed a system that does.

To satisfy these requirements the new system, named Geb (after the Egyptian god of the Earth), uses a correlated genotype to phenotype mapping, so that long-term incremental evolution is possible, and has biotic selection as the dominant selection force, unsuppressed by abiotic selection. Chapters 2 and 3 presented the arguments that the perpetuation of evolutionary emergence requires these properties.

A correlated genotype to phenotype mapping is achieved through the use of neural networks as the organisms' control architectures. Neural networks are a suitable choice because of their well known 'graceful degradation' property, which translates into genetic neutrality (and near-neutrality). Insufficient genetic neutrality is a pitfall in the evolution of computer program instructions, where no new genes (subroutines, at the pseudocode/descriptive level) have yet evolved. Neural networks also satisfy the requirement of being suitable for complex advantageous behaviours. The networks are produced from bit-string genotypes by a developmental process, chosen with both genetic neutrality and the developmental requirements of section 3.5 in mind.



Time: 893538

Figure 4.1: The experimental world (Geb).

Evolution within Geb is by *biotic selection*, with no (or negligible) abiotic selection and no lifetime learning. There are no global system rules that delete organisms; this is under their own control. So biotic selection can operate, without being suppressed by abiotic selection. And there can be no doubt that all results reported are due to biotic selection.

Geb is a two-dimensional toroidal virtual world (figure 4.1) containing autonomous organisms, each controlled by a neural network. The world is divided into a grid of squares: 20 × 20 of them in most runs. No two individuals can occupy the same square at any one time – no square in figure 4.1 contains more than one individual's centre. This effectively gives the organisms a size within the world and puts a limit on their number. Individuals are otherwise free to move around the world, within and between squares. As well as a position within the world, 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:

Initialisation

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. See section 4.2.
2. Development – an iteration of the ontogenesis mechanism. See section 4.3.
3. All neural activations, including network outputs, are updated. See section 4.1.
4. 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. See section 4.2.

4.1 Geb's neural networks

The artificial neural networks used in Geb are recurrent networks of nodes as used successfully by Cliff, Harvey and Husbands in their evolutionary robotics work (Cliff, Harvey and Husbands, 1992; Harvey, Husbands and Cliff, 1992; Husbands, Harvey and Cliff, 1993). The neural model (figure 4.2) is based on McCulloch and Pitts' (1943) original proposal, which includes a distinct inhibitory mechanism (rather than the more prosaic positive-or-negative synaptic weights as typically used in parallel distributed processing systems). Cliff et al. evolved recurrent networks of these nodes for visual navigation tasks in simple environments. Such networks were chosen (rather than alternative neural network designs such as feed-forward perceptron-based networks) for use in Geb because of their suitability for adaptive behaviour work, as demonstrated by Cliff at al.'s experiments.

The level of noise here (0.6 – see figure 4.2) is significantly higher than that used by Cliff et al. (0.1). This is because noise is the only source of activation in Geb

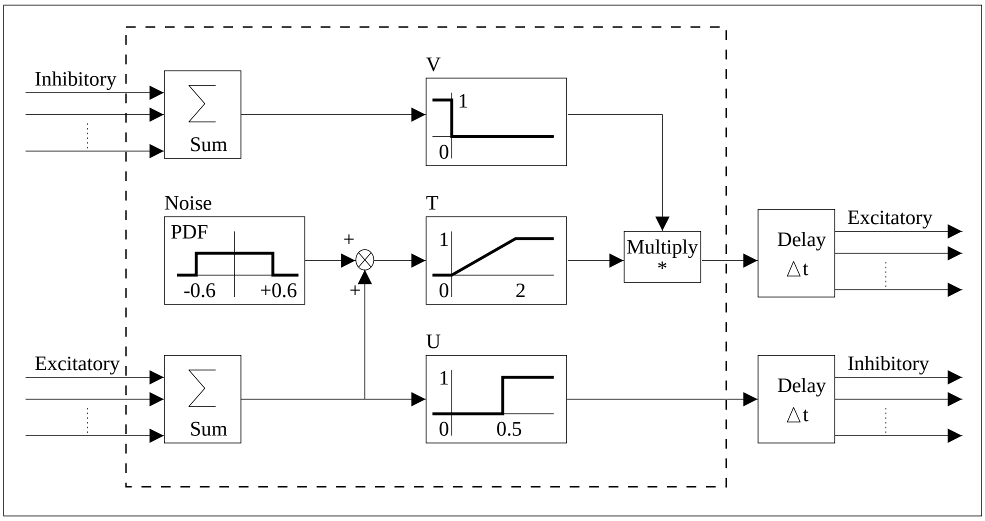


Figure 4.2: Schematic of a neuron, from Cliff, Harvey and Husbands (1992) .

and, with the developmental method outlined below, it is easy for evolution to produce *generator units* (Husbands, Harvey and Cliff, 1993), which are sources of high output. A corresponding (high) decision threshold for organisms' binary (yes/no) actions, such as reproduction, is used. Thus, full control is available (via inhibition and generator units), early random binary actions are at a sensible level and early random multi-valued actions (such as moving forwards by a distance) can be at a reasonably high level without having to be scaled such that the maximum possible is unreasonably high. The neurons' veto threshold (0.5 – see function U in figure 4.2) is equal to the decision threshold for organisms' binary actions. All links have unit weight; no lifetime learning is used. This is to avoid the criticism that lifetime learning may be the main factor, as levelled at PolyWorld in section 3.4.2.

Each node has a bit-string 'character' (label) attached to it, used to match organisms' network inputs, outputs and actions, and to determine the node's development during the individual's lifetime. These characters may be of any non-zero length. A node may be a network input, a network output, or neither. This is determined by the developmental process.

4.2 Organism—environment interactions

There are five built-in actions available to each organism. Reproduce and fight (kill organism in front if there is one) were chosen to enable purely biotic selection: either is a sufficient action for biotic selection, but both are required if there is to be no abiotic selection. The remaining three actions allow basic locomotion, intended to enable complex behaviours that could be interpreted by Geb's user. Each action is associated with network output nodes whose characters start with a particular bit-string:

1. `01...` Try to *reproduce* with organism in front
2. `100...` *Fight*: Kill organism 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 organism will turn clockwise by an angle proportional to the excitatory output of that node. If an action has more than one matching network output node, then the relevant network output is the sum of these nodes' excitatory outputs, bounded by unity as within any node. If an action has no network 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 organism reproduces with another in front of it, the child is placed in the square beyond the other individual if that square is empty. If not, the child replaces the individual being mated with. An organism 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. Also, crossover is strict, always using genes from both parents; the cut point cannot be at the very end of either genotype.

```
Parent 1’s genotype: 1 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 1 1 1
                                ^--- cut point in parent 1
Parent 2’s genotype: 1 0 1 1 0 0 1 1 1 1 0 0 0 1 0 0 0 1
                                  ^--- cut point in parent 2
------------------------------------------------------------
Child’s    genotype: 1 1 0 0 0 1 1 1 1 0 0 0 1 0 0 0 1
                       ^-- mutation point
```

Figure 4.3: Example of crossover and mutation.

This provides significant initial pressure for length increase until genotypes are long enough to produce developmental rules. Mutation at reproduction is a single gene-flip (bit-flip) on the child genotype. Figure 4.3 gives a simple illustrative example of the crossover and mutation used (although note that genotype lengths in the thousands would be more representative).

An organism’s network input nodes have their excitatory inputs set to the weighted sum of the excitatory outputs from *matching* network output nodes’ of other individuals in the neighbourhood. 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 neighbourhood is a 5×5 square area centred on the relevant organism.

Notice that the network output nodes with characters `0`, `1`, `10`, `11` and all those starting with `00` do not produce any 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.

4.3 Developmental system

A class of L-systems with context-free production rules was designed for the evolution of networks of the neurons outlined above. Design of the developmental system was carried out with both genetic neutrality and the developmental requirements of section 3.5 in mind. Specific attention was paid to producing a system in which children's networks resemble aspects of their parents'. A genotype determines the L-system's production rules which determine the organism's neural development. Thus the production rules evolve.

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. So an empty (zero-length) predecessor matches any node's character and a predecessor cannot match a node's character that is shorter than it. 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:

$$P \rightarrow S_r, S_n \; ; b_1, b_2, b_3, b_4, b_5, b_6$$

where:

P	Predecessor (initial bits of node's character)
S_r	Successor 1: *replacement* node's character
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) inherited links on S_n
(b_3, b_4)	(inhibitory, excitatory) link from S_r to S_n
(b_5, b_6)	(inhibitory, excitatory) link from S_n to S_r

The *successors* (1 and 2) are characters for the node(s) that replace the old node. 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* (successor 1, if it has a character) is just the old (predecessor) node, with the same links but a different character. The *new successor* (successor 2, if it has a character) 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 input nodes are (only) produced from input nodes and new output nodes are (only) produced from output nodes. The character-based method of matching up 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 inputs and outputs.

The axiom network consists of three nodes with two excitatory links. The network output node's character (`01`) is matched by the reproduction character. The network input node's character (left input `01`) matches this output without matching or being matched by the other nodes, and 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:

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

Development takes place throughout the individual's life. Figure 4.4 provides an example of the development of a very simple network. It shows the axiom (age 0) network growing as, at every time step, each node has the most specific rule applied to it. Only the first two steps are shown.

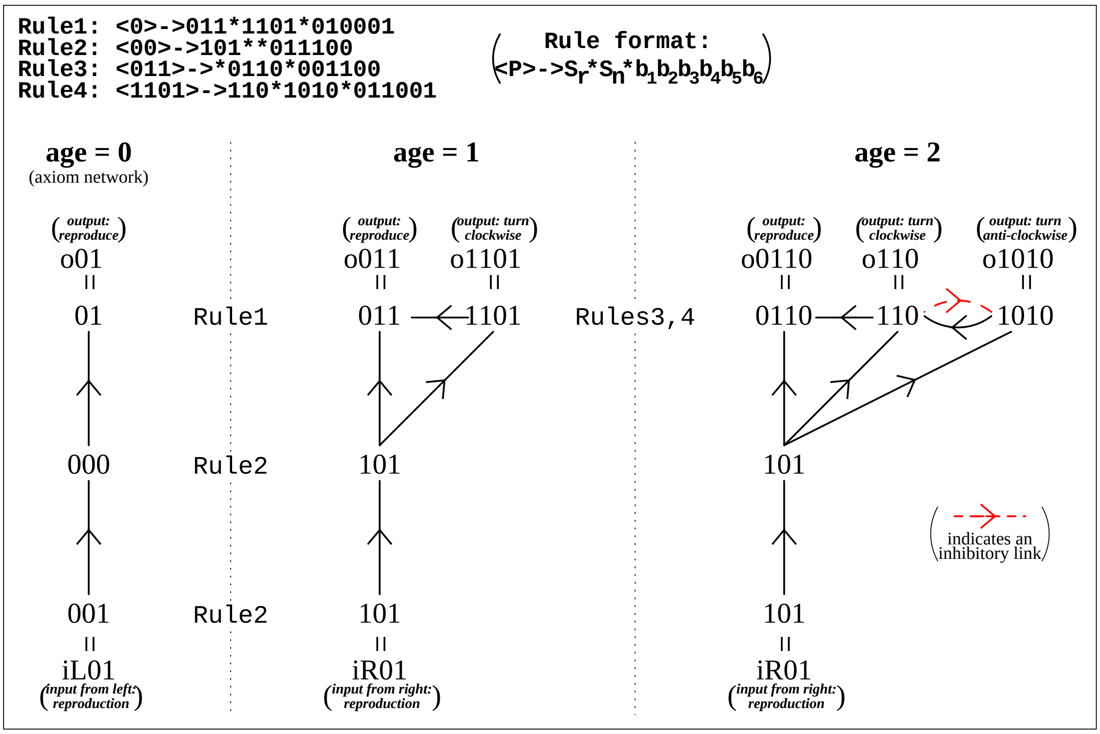

Rule1: <0>->011*1101*010001
Rule2: <00>->101**011100
Rule3: <011>->*0110*001100
Rule4: <1101>->110*1010*011001
Rule format: <P>->S_r*S_n*$b_1b_2b_3b_4b_5b_6$
age = 0
(axiom network)
age = 1
age = 2
output: reproduce
output: turn clockwise
output: turn anti-clockwise
o01
o011
o1101
o0110
o110
o1010
01
011
1101
0110
110
1010
Rule1
Rules3,4
000
Rule2
101
101
indicates an inhibitory link
001
Rule2
101
101
iL01
iR01
iR01
input from left: reproduction
input from right: reproduction
input from right: reproduction

Figure 4.4: Example neural network development.

4.4 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. This method is well suited to a system in which genotype length changes through the addition or removal of sections (here just one bit position per reproduction – see section 4.2) from within the genotype.

To read a rule, the system uses the concept 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 no information bits).

The full procedure to (attempt to) read a rule begins with reading a segment for

```
Genotype:  1 1 1 0 1 1 0 0 1 0 1 1 1 0 0 0 0 0
Rules:  i)   <>->  1 *   0 * 0 1 1 1 0 0
       ii)     <   1>->  0 *   1 * 1 0 0 0 0 0
Format:  < P >-> Sr *  Sn * link bits
```

Figure 4.5: Example rule generation.

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. If this is achieved before the end of the genotype then a rule is created.

Figure 4.5 shows an example. As the first two bits are `11`, an attempt is made to read a rule from the third bit onwards. The segment-reading procedure above is carried out for the predecessor, which is empty because the third genotype bit is a `1`; for the first successor, which has one bit `1` from the fifth genotype bit because the fourth genotype bit is a `0`, but only one bit because the sixth genotype bit is a `1`; and for the second successor, which has one bit `0` from the eighth genotype bit because the seventh genotype bit is a `0`, but only one bit because the ninth genotype bit is a `1`. Having read these three characters, the six link-details bits are read, in this case from the tenth genotype bit onwards. As the last link-details bit can be read, that is reading has not gone beyond the end of the genotype, a rule has been successfully read and is added to the list of decoded rules. This rule-reading procedure is now carried out again, starting from the second genotype bit. As the second and third genotype bits are `11`, an attempt is made to read a rule from the fourth bit onwards. This attempt is again successful, as the sixth link bit can be read without going beyond the end of the genotype. As the third and fourth genotype bits are not `11`, and likewise for the fourth and fifth genotype bits, no attempt to read a rule will be made for these positions. The fifth and sixth genotype bits are `11`, so an attempt is made to read a rule from the seventh bit onwards. However this and subsequent attempts fail because the procedure runs out of genotype before completing a successful read.

After reading all possible rules from a new-born's genotype, Geb filters the rules. It starts with rules whose predecessors best match a node in the axiom network, and then repeatedly adds in the best matching new rules if possible and as required, matching predecessors to the successors of rules already picked. Rules that have not been picked when this process stops (because no new rules can be added under the criteria) have predecessors that could never match a node during development, at least not as well as another rule. In this way the redundant rules, which constitute the vast majority of decoded rules from long genotypes, are filtered out, much reducing memory required by Geb.

During this process, a further criterion must be met for a rule to be added: the gene-segment the rule was decoded from must not overlap with a gene-segment of any rule already picked. This prevents the otherwise common situation of a rule $P{\rightarrow}R{,}N$;`bits` producing successors R and N which can then be subject to rules $R{\rightarrow}N{,}B{;}C$ and $N{\rightarrow}B{,}C{;}D$ (and so on) as would be the case whenever P ends in `1` or R ends in `1`. So, without this criterion, certain rules (such as those where P ends in `1`) would not be possible independently; they would automatically produce rules (such as $R{\rightarrow}N{,}B{;}C$) which interfere with their successors.

Chapter 5

Results

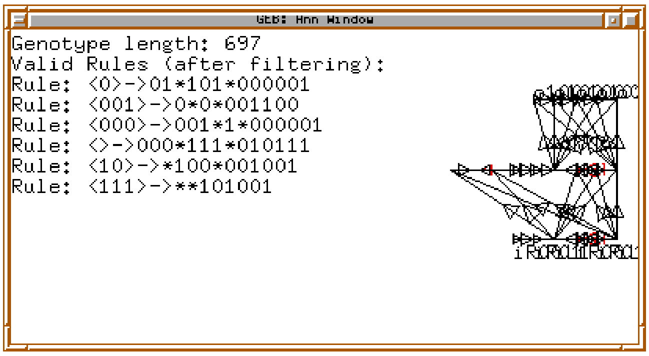
The system has consistently produced some important macro-level behaviours, although obviously the details of its evolution are different every time. This makes it difficult to describe the behaviour except by focusing on some typical and interesting observations.

5.1 Kin similarity and convergence

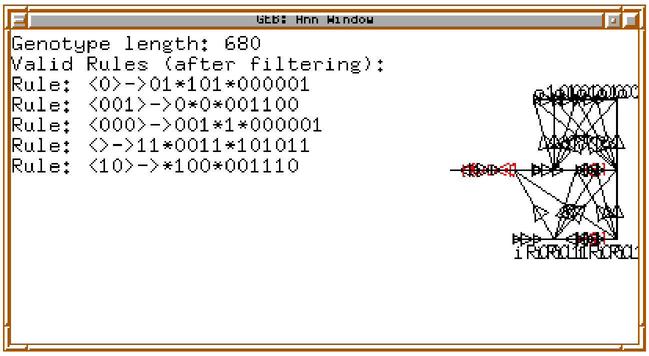
When two Geb organisms (with networks developed from more than just a couple of production rules each) reproduce, the child's network almost always resembles a combination of the two parents' networks. This has been seen many times; Figure 5.1 is a typical example. This figure shows the form in which Geb displays neural networks on request, with all nodes crowded into three rows (network inputs, hidden nodes and network outputs). The 'pulling apart' and detailed examination of networks is left to be carried out by the user through the dragging and dropping of nodes.

Examination of a larger number of networks from Geb's population, at any time, shows similarities between many of the networks. 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. The remaining results are the proof

Parents:

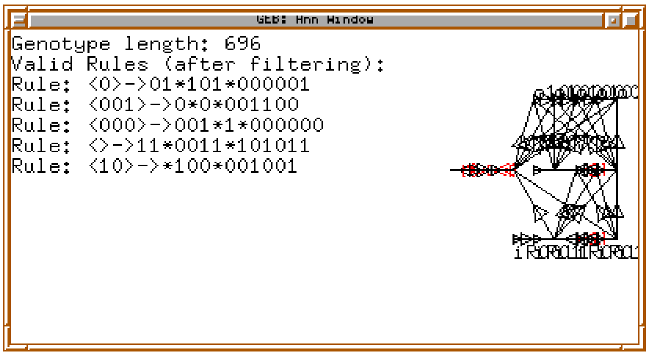


```
Genotype length: 697
Valid Rules (after filtering):
Rule: <0>->01*101*000001
Rule: <001>->0*0*001100
Rule: <000>->001*1*000001
Rule: <>->000*111*010111
Rule: <10>->*100*001001
Rule: <111>->**101001
```



```
Genotype length: 680
Valid Rules (after filtering):
Rule: <0>->01*101*000001
Rule: <001>->0*0*001100
Rule: <000>->001*1*000001
Rule: <>->11*0011*101011
Rule: <10>->*100*001110
```

Child:



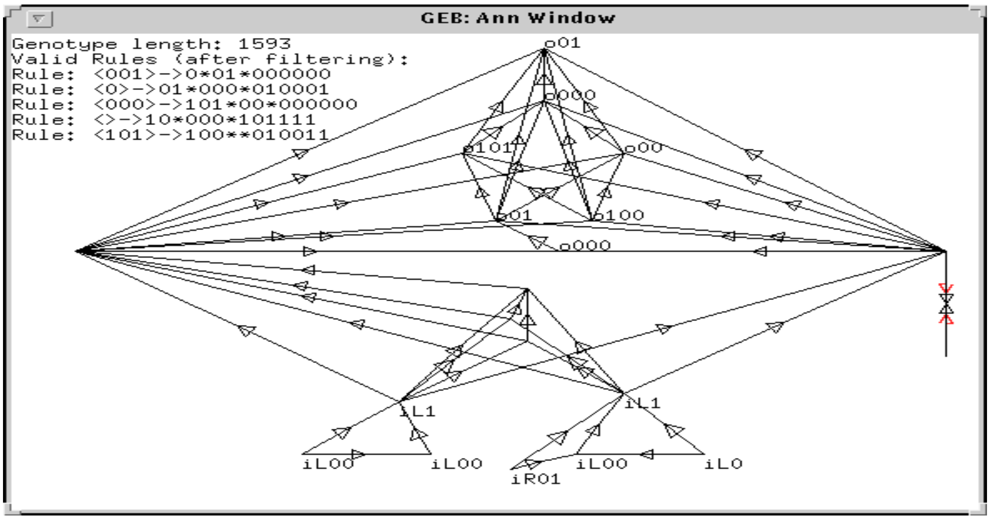
```
Genotype length: 696
Valid Rules (after filtering):
Rule: <0>->01*101*000001
Rule: <001>->0*0*001100
Rule: <000>->001*1*000000
Rule: <>->11*0011*101011
Rule: <10>->*100*001001
```

Figure 5.1: The unexpanded neural networks of two parents and their child.

of this suitability and so justify the claim that the use of neural networks can result in sufficiently correlated landscapes and further that Geb achieves this within a modular development system.

5.2 Emergent collective behaviour

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 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 Geb's world window.

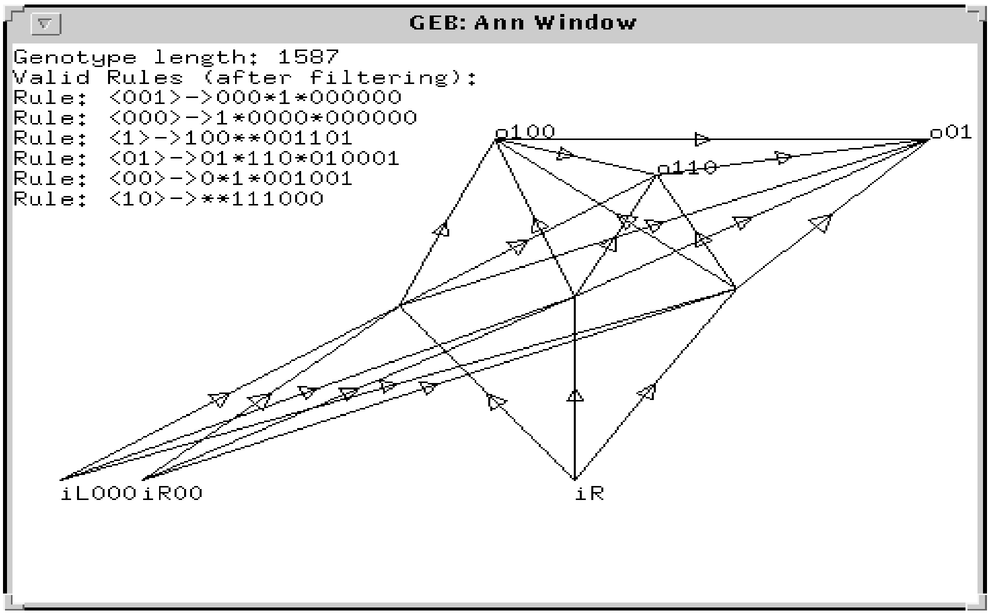


```
Genotype length: 1593
Valid Rules (after filtering):
Rule: <001>->0*01*000000
Rule: <0>->01*000*010001
Rule: <000>->101*00*000000
Rule: <>->10*000*101111
Rule: <101>->100**010011
```

Figure 5.2: A dominant organism's neural network.

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 5.2 shows an example of such an individual. Network outputs are prefixed with `o`, inputs with `i`. Input characters are shown with their first bit translated from `0`,`1` to `L`,`R` (left,right). 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).

Although a very simple strategy, watching it in action makes it clear why this is so advantageous. The individuals keep each other moving quickly, in tight circles. Any attacking organism would have to either get its timing exactly right or approach in a faster spiral – both relatively advanced strategies. These dominant individuals also mate just before killing. The offspring (normally) appear beyond the individual being killed, away from the killer's path.



```
Genotype length: 1587
Valid Rules (after filtering):
Rule: <001>->000*1*000000
Rule: <000>->1*0000*000000
Rule: <1>->100**001101
Rule: <01>->01*110*010001
Rule: <00>->0*1*001001
Rule: <10>->**111000
```

Figure 5.3: A rebel organism's neural network.

5.3 Naturally arising coevolution

Because of the success of this first dominant species (especially their success at killing other organisms), the world always has enough space for other organisms to exist. Such other organisms tend not to last long; almost any movement will bring them into contact with one of the dominant organisms, helping that species in its reproduction as much as themselves. However, they can make some progress. Individuals have emerged that are successful at turning to face members of 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 5.2) is shown in figure 5.3. Note that most rebels have many more nodes and links; this one was picked for its clarity. The main points to note from this figure are the network inputs `iL000` (left `000`) and `iR00` (right `00`) which match the very non-action output characters that members of the dominant species use to support each other’s activations (`o000` [x2], `o00`). The network outputs are `o100` (fight), `o01` (reproduce) and `o110` (turn clockwise). By turning clockwise, a rebel will turn towards its enemy fastest when the enemy is to its right, which is where most of the rebel’s input is taken from (via the input `iR`), and hence the side where it responds best. Most rebels have more complicated networks, which are very difficult to comprehend. Indeed, many of these have proved unassailably difficult to understand in detail.

5.4 Ongoing coevolution

Figures 5.4 and 5.5 show *running averages* of the number of organisms reproducing and killing, from two typical experimental runs. Each point is the average of the raw data (number of appropriate organisms) over a window moving along the time axis. This filters out Lotka-Volterra population cycles and short term random variations, revealing long term shifts. These figures suggest that further species emerge, indicating ongoing evolutionary emergence. However, organisms have proved difficult to analyse beyond the above, even at the behavioural level. All that can currently be said is that they share characteristics of the previous species but are different.

5.5 The need for behavioural transparency

The infeasibility of further behavioural analysis provides one of the most important lessons to be learned from Geb. When designing the system, my thinking was that by simply including locomotion in the set of agent’s actions, most (or at least a good

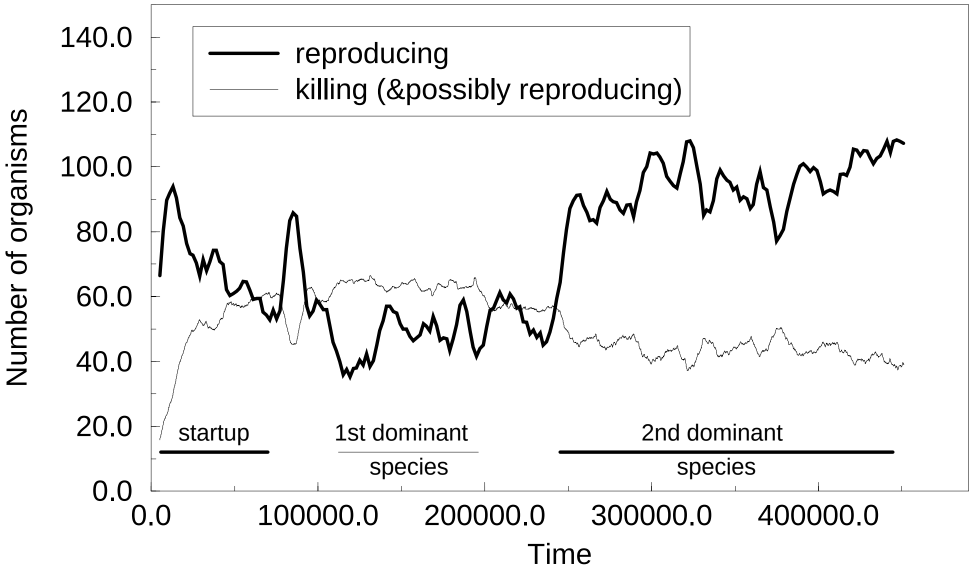


Figure 5.4: Typical run 1 of Geb (running averages of population sizes by actions).

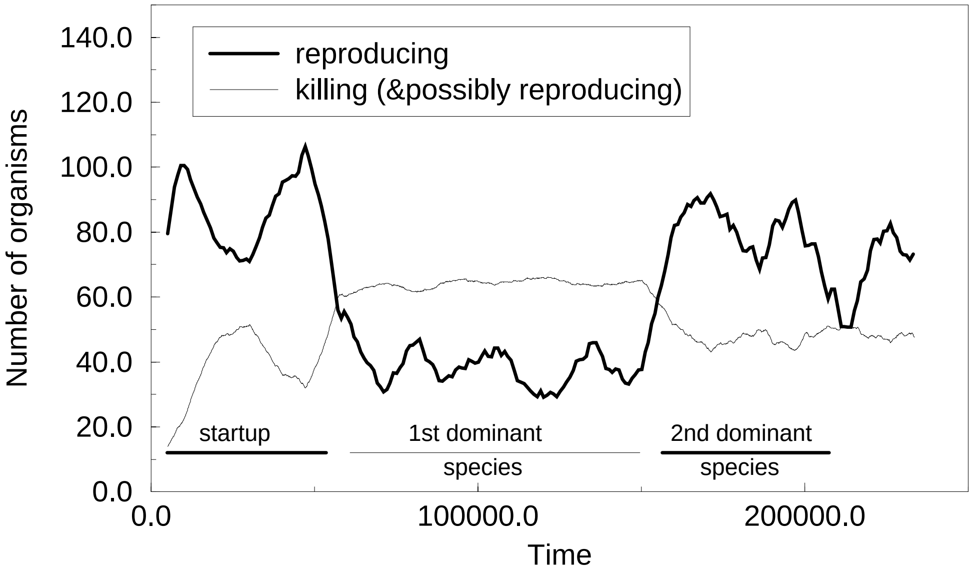


Figure 5.5: Typical run 2 of Geb (running averages of population sizes by actions).

proportion of) emergent behaviours would be open to behavioural interpretation – see section 4.2. I had in mind behaviours such as pursuit, evasion and Boid-like flocking, at the very least. However, further thought (easy to do retrospectively) reveals why this should not be expected in Geb.

In Darwinian evolution, ecologies are always full (or at least quickly become so) – see section 1.1. In Geb, an always-full ecology translates (approximately) to between a quarter and three quarters of grid squares being occupied in most timesteps. This leaves little room for locomotion behaviours over long distances, such as flocking. And even pursuit and evasion (for example) would be difficult to observe over short distances. In large part this difficulty arises because all a user can observe in Geb's main window are coloured filled arcs (pie slices) moving around over small distances: a very limited supply of data on which to form interpretations of emergent behaviours, or even to identify when a novel behaviour has emerged. If the organisms' bodies were less trivial and formed an embedded part of the system, then it might be possible to interpret behaviours from the motions of body parts. Sims' (1994a, 1994b) evolution (by *abiotic* selection) of virtual creatures in a three dimensional physically simulated environment provides an excellent example of this. These experiments produced behaviours which are easy to identify and describe, operate over short distances but many time-steps in a non-simplistic way, and are largely the result of the evolved embodiment of the organisms. Incorporating such features into a system such as Geb would certainly be a significant step forward.

Chapter 6

Evolutionary statistical analysis I

Because analysis at the level of individual organisms (either neural network based or behavioural) is not feasible beyond that in the previous chapter, a more statistical approach is required in order to test for ongoing evolution. Fortunately a new method for classifying evolutionary systems has recently been developed.

6.1 Bedau and Packard's evolutionary statistics

As the discipline of Artificial Life has matured into a science, so has the need for quantifiable measures of success. Bedau and Packard (Bedau and Packard, 1991; Bedau, Snyder, Brown and Packard, 1997; Bedau, Snyder and Packard, 1998; Bedau and Brown, 1999) have developed not only elegantly simple statistical measures for long-term evolution, but also a test for unbounded evolution. The test is so adaptable that it can be applied to any evolving system with an available record of its components' existence times, such as the biosphere's fossil record. Any artificial system can be tested, and those that have been include Tierra-like systems (Adami and Brown, 1994; Taylor and Hallam, 1998), Echo (Holland, 1975), Bugs (Packard, 1989), Lindgren's model of evolving strategies in the iterated prisoner's dilemma (Lindgren, 1991), Arthur's "Bar

Problem" (Arthur, 1994) and Ecolab (Standish, 2000). Previously only taxonomic families in the fossil record have passed the test [1].

The motivation behind the statistics is to measure the level at which a system generates and then persistently uses adaptive innovations. By continually measuring both the generation of and the persistent use of adaptive innovations, these statistics are able to determine unbounded evolutionary activity, which requires both be ongoing.

Bedau and Packard's test is based on the following statistics (from Bedau, Snyder and Brown, 1997; Bedau, Snyder and Packard, 1998), which are calculated from the record of components' existence times.

Activity increment (by presence).

$$\Delta_i(t) = \begin{cases} 1 & \text{if component } i \text{ exists at } t \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

A component's activity is intended as a measure of its level of usage so far in evolution. An activity increment of one unit in each time step that the component exists satisfies this provided a component is only considered to exist if it is being used. So, for example, a gene that is present in a population but not currently expressed should not be considered to exist in the population of in-use gene components.

This is not the only activity increment that Bedau et al. have used, but it is the best for comparison across systems because it can be calculated for any system with a record of components' existence times.

[1] Maley (1999) makes the claim that two of his models, 'Urmodel 3' and 'Urmodel 4', exhibit unbounded evolutionary activity. However, Urmodel 3 shows less new activity than its shadow (with no reason to think that it would become greater), Urmodel 4 shows a lower mean activity than its shadow and both are only examined during their initial growth stages, so these claims are not valid. To my knowledge, there have not yet been any other claims of unbounded evolutionary activity in an autonomous artificial system.

Evolutionary Activity of a component.

$$a_i(t) = \begin{cases} \sum_{\tau=0}^{t} \Delta_i(\tau) & \text{if component } i \text{ exists at } t \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

Diversity (the number of components present, in use).

$$D(t) = \#\{i : a_i(t) > 0\} \tag{6.3}$$

Total cumulative evolutionary activity ('total activity').

$$A_{\text{cum}}(t) = \sum_i a_i(t) \tag{6.4}$$

Mean cumulative evolutionary activity ('mean activity').

$$\bar{A}_{\text{cum}}(t) = \frac{A_{\text{cum}}(t)}{D(t)} \tag{6.5}$$

Total activity provides a measure of the extent of evolutionary activity: how much activity has accumulated and remained in the system through the persistent use of components. It can be unbounded due to unbounded mean cumulative evolutionary activity, unbounded diversity, or both.

New evolutionary activity per component ('new activity').

$$A_{\text{new}}(t) = \frac{1}{D(t)} \sum_{i:a_i(t)\in[a_0,a_1]} a_i(t) \tag{6.6}$$

New evolutionary activity (per component) provides a measure of the intensity of evolutionary activity: the level at which the system is generating adaptive innovations.

		STATISTICAL SIGNATURE		
CLASS	EVOLUTIONARY DYNAMICS	D	A_{new}	$\bar{A}_{\text{cum}}$
1	none	bounded	zero	zero
2	bounded	bounded	positive	bounded
3a	unbounded (D)	unbounded	positive	bounded
3b	unbounded ($\bar{A}_{\text{cum}}$)	bounded	positive	unbounded
3c	unbounded (D & $\bar{A}_{\text{cum}}$)	unbounded	positive	unbounded

Table 6.1: Classes of evolutionary dynamics and their statistical signatures, based on table 1 from Bedau, Snyder and Packard (1998a). Rows 3b and 3c have been added to class 3.

Unbounded evolutionary activity requires both ongoing intensity and increasing extent of evolutionary activity: both positive new activity and unbounded total activity.

For A_{new} to be a good measure of new activity, the range $[a_0, a_1]$ should be chosen such that component activities within it can be considered both adaptively significant (so a_0 should be high enough to screen out most non-adaptive activity) and not amongst the highest activities (so a_1 should be low enough that a good proportion of activities lie above it).

For artificial systems, a “shadow” should be run, mirroring the real run in every detail except that whenever selection (artificial or natural) operates in the real system, random selection should be employed in the shadow. The statistics from this shadow run can then be used to determine a_0 and levels of total and mean activity that can be considered adaptively significant.

6.2 Classification of evolutionary dynamics

After determining long-term trends in these statistics, the system being examined can be classified according to table 6.1. The hallmark of class 3 (unbounded evolutionary dynamics) is unbounded total cumulative evolutionary activity in combination with positive new evolutionary activity per component.

Other possibilities exist with zero A_{new}, but these belong in class 1 (no evolutionary activity) because such cases have no significant new components. Table 1 in (Bedau, Snyder and Packard, 1998) only shows the first row (3a) for class 3, but footnote 1 in (Bedau, Snyder and Packard, 1998) mentions the other rows (3b and 3c). Note that table 6.1 includes all possibilities for positive A_{new}, because zero $\bar{A}_{\text{cum}}$ implies zero A_{new}. So any system with unbounded evolutionary dynamics will belong to class 3 (one of 3a, 3b and 3c).

Figure 6.1, from Bedau, Snyder and Brown (1997, figure 1), shows evolutionary statistics for the biosphere. The raw existence data sets for this were drawn from fossil records showing the first and last appearances of taxonomic families (Benton, 1993; Sepkoski, 1992). Clearly it is not possible to run a shadow model for the biosphere. Bedau et al. consider "normalization to be accomplished *de facto* by the fossil record itself. In our view, the mere fact that a family appears in the fossil record is good evidence that its persistence reflects its adaptive significance. Significantly maladaptive taxonomic families would likely go extinct before leaving a trace in the fossil record." (Bedau, Snyder and Packard, 1998, p. 229). Figure 6.1 shows that cumulative evolutionary activity is unbounded in the biosphere, as a result of unbounded diversity. Of course new activity is positive in the biosphere (as shown in Bedau, Snyder and Packard, 1998, figure 2). So the biosphere is a class 3 system, exhibiting unbounded evolutionary dynamics.

Figure 6.2, from Bedau, Snyder and Brown (1997, figure 2), shows evolutionary statistics from a run of Evita, a Tierra-like systems, and its shadow. Evita's programs exist in a two dimensional grid, with at most one program in each square. Once a program has made a copy of itself, the new program is placed in another square. If an empty square exists nearby, then that is used. Otherwise the system overwrites one of the oldest neighbours. A shadow run can be constructed from a recording of the number of mutations and reproductions in each timestep from a real run. The shadow run is then executed using these parameters, with random selection of "programs" for reproduction. Figure 6.2 shows that both cumulative activity and diversity are bounded

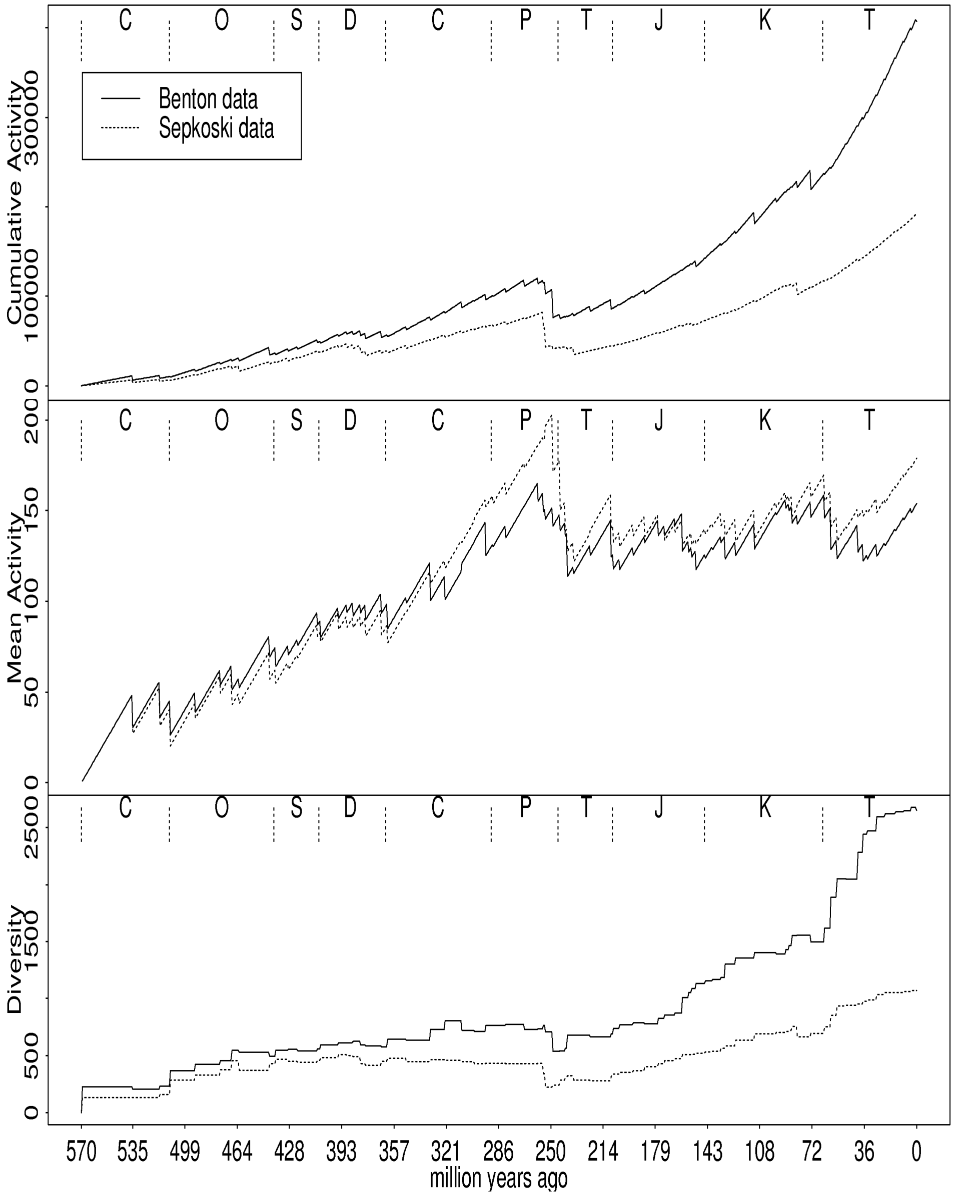


Figure 6.1: Cumulative activity (top), mean activity (middle) and diversity (bottom) in the fossil data of Benton and Sepkoski. The labels at the top of each graph show the boundaries between the standard geological periods, thus: Cambrian, Ordovician, Silurian, Devonian, Carboniferous, Permian, Triassic, Jurassic, Cretaceous, Tertiary. Figure and caption from Bedau, Snyder and Brown (1997, figure 1).

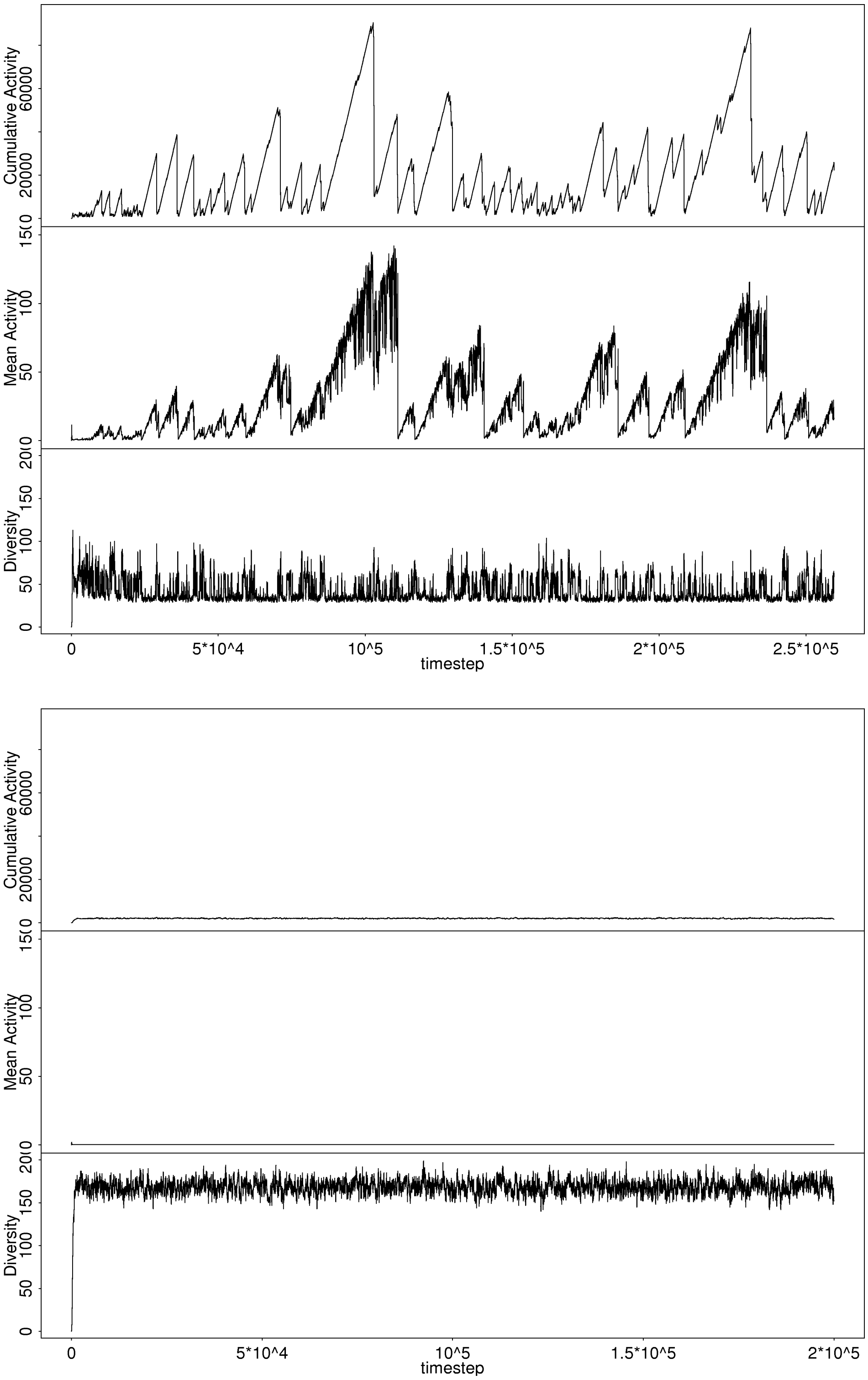


Figure 6.2: Above: cumulative activity (top), mean activity (middle) and diversity (bottom) in an Evita simulation. Below: the same statistics for a neutral analogue of the Evita simulation above. Figure and caption from Bedau, Snyder and Brown (1997, figure 2).

in the real run, so these results cannot fall into class 3. It should be noted that code parasitism is not possible in Evita, as it is in Tierra. However, because the number of evolved programs in Tierra was found to be bounded (see section 3.4.1), we know that new activity drops to zero. So Tierra's results cannot fall into class 3 either.

The same is true for any conventional optimisation-based genetic algorithm lacking neutrality. As the population converges around the final solution(s), new activity drops to zero. Diversity would be bounded and mean (and so also cumulative) activity would grow without bound. A genetic algorithm with (ongoing) search along neutral networks would demonstrate positive new activity but bounded mean (and cumulative) activity, because each component's activity is lost from the system once that component is lost from the system. This is also what we see in figure 6.2 from the Evita run. None of these systems demonstrate both an ongoing intensity and an increasing extent of evolution.

The following quote from their discussion section summarises Bedau, Snyder and Packard's conclusion.

> "Other natural evolving systems probably show class 3 dynamics. Class 3 dynamics might even be detectable in systems like the global economy or Internet traffic. We also suspect that no existing artificial evolving system has class 3 dynamics. In our opinion, creating such a system is among the very highest priorities of the field of artificial life. From one perspective, this is a negative result: Echo, and perhaps all other existing artificial evolutionary systems, apparently lack some important characteristics of the biosphere – whatever is responsible for its unbounded growth of adaptive activity. But at the same time this conclusion calls attention to the important constructive and creative challenge of devising an artificial model that succeeds where all others have failed." (Bedau, Snyder and Packard, 1998, p. 236)

6.3 Implementing the statistics in Geb

In order to apply Bedau and Packard's test to an evolutionary system, the biggest decision to make is what the class of components should be. As Geb's genotypes both change length and contain a high degree of neutrality[2] the genotype is not a good choice of component class. Production rules, the alleles from a genotype, are a much better choice. It can be expected that if a production rule has an adaptive advantage, then it will persist. So the production rule would be a suitable choice of component. Better still is the choice "production rules that survive the filtering process at birth", for these are the rules that are actually used in the developmental process (section 4.4); the idea behind activity statistics is to measure the degree to which components both *persist* and are *used*.

When mutation causes a component to not be expressed (currently present), the activity count of the original component is no longer included in the total activity of the system, even if the mutation is functionally neutral. At first I implemented the activity statistics on production rules directly. But there is often a high degree of neutrality in a production rule, especially when its 'successors' relate to neurons that are over-specified (have excess bits at the end of their characters) or development-terminal (not matched by any production rule). The predecessor and link-bits sections of production rules are more plastic. If a predecessor bit is mutated, then the rule will most likely either fail to match or be less specific to its target neuron than another rule. If a link detail bit is mutated, then the result will more often than not be a damaged network, and organisms with that production rule active (not filtered out) will be driven to extinction. So the choice of component used here is 'predecessor plus link details' $(P, b_1, b_2, b_3, b_4, b_5, b_6)$. This can be thought of as a disjoint grouping of alleles, with each group being a component. Which individual a component is from is irrelevant: two identical production rules in two different organisms result in two instances of the same component.

[2] In order to avoid confusion, I only use the term *neutral* to refer to genetic variations that are phenotypically equivalent, and not in relation to shadow runs.

This grouping does not completely remove the neutrality problem. As successor lengths increase, neural character lengths increase, and so the number of predecessors that can potentially match a typical neural character increases. If two rules have the same successors and link-details (or neutral variants), then it makes no difference to development which one is used. So, as component lengths increase, we can expect the level of neutrality to increase.

Having chosen this component class, there is a clear consequence for the possible classifications of evolutionary dynamics. Because the number of neurons that an organism can have is limited (for practical reasons), the number of production rules that can survive filtering is limited. And because the population size is small (a maximum of four hundred organisms), there is little room for more than a couple of species at a time. So diversity of these components will certainly be bounded, and we can rule out class 3a and 3c dynamics.

6.3.1 Implementation details

Geb's shadow mirrors the real run in every detail except that selection is random. Whenever a real organism is killed, a randomly chosen shadow organism is also killed. Whenever a real organism is born (as the product of two real organisms), a new shadow organism is born as the product of two randomly chosen shadow organisms, using the same reproduction procedure with the same rates of crossover and mutation. For each initial real organism born with single-bit genotype ‘0’, an initial shadow organism is also born with single-bit genotype ‘0’.

Gathering evolutionary activity statistics, or rather the component existence record from which they are calculated, is (processor-)time consuming. It is not feasible to gather the statistics at every timestep. So snapshot existence records are taken at regular intervals and the evolutionary statistics are calculated from these. In the results reported here, snapshots were taken every one thousand timesteps. To put this in context, the run reported lasted six million timesteps, during which time there were over five hundred and eighty million organism reproductions. In each timestep, every organism is updated. Because activity is intended as a measure of how much a component both is used (already covered above) and *persists*, I screen out (in each of the real and shadow populations) isolated occurrences: when a component occurs in the current snapshot but not the previous one.

In the previous chapter, total extinction (population size dropping to one individual) was not mentioned because it had not been encountered. However, there was no mechanism in place to prevent it and, during the long trial runs undertaken when experimenting with evolutionary statistics, I encountered occasional runs in which total extinction occurred. So for the set of runs from which the example reported here is taken, I set a minimum number of organisms to twenty. The fact that total extinction is so rare despite the population size being so small (a maximum of four hundred organisms at any one time) indicates that there is no serious problem here. Once population sizes can feasibly be increased, the problem should in practice disappear rapidly.

6.4 Results and Discussion

This section contains the results from a typical run, drawn from the full set of twenty runs. Atypical variations are discussed at the end of this section.

6.4.1 Activity waves

In order to gain an understanding of the dynamics behind the higher level evolutionary statistics, it is a good idea to look first at the activity wave diagrams, which simply show all components' activities plotted against time. Figure 6.3 shows the activity wave diagrams for the real and shadow populations.

The most obvious feature of the real run's activity waves in figure 6.3 is that many of them keep increasing. This would also be true in a similar analysis of genes from the biosphere's evolution. Genes that are beneficial to life tend to remain in the population of genes and be used by many species: humans have a significant proportion of genes in

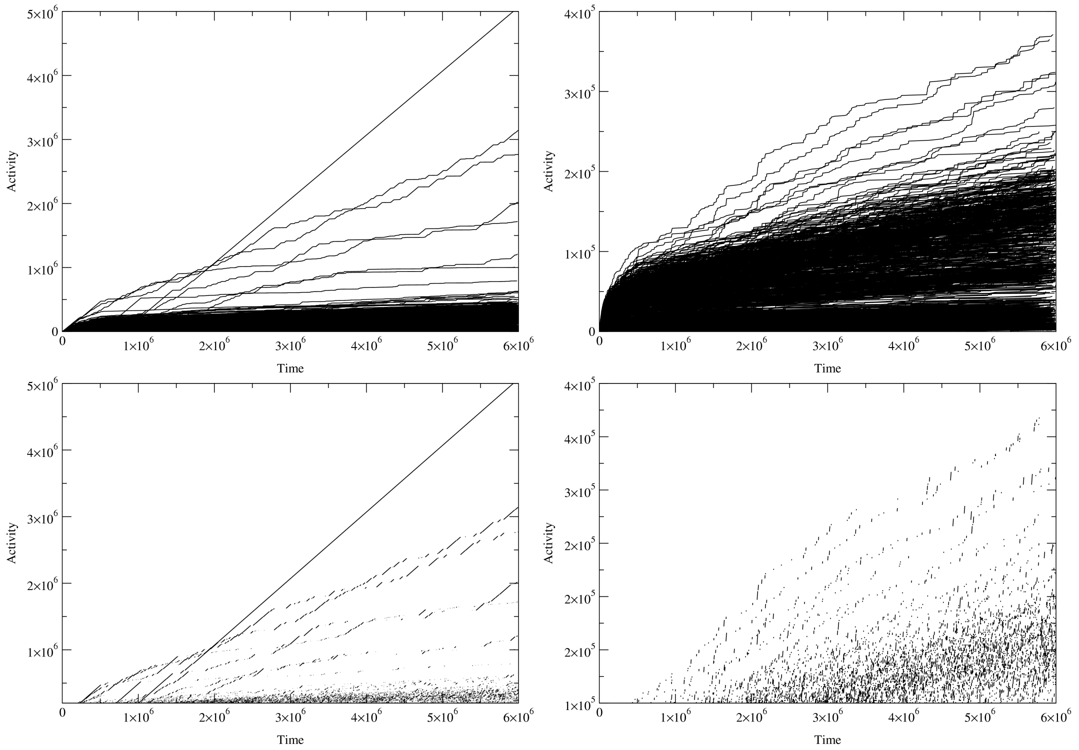


Figure 6.3: Activity wave diagrams for the real (left) and shadow (right) runs. The diagrams on the bottom have had all horizontal (no-increase) lines removed. Note the different scales for real and shadow.

common with mice, flies and even plants. So because the components here are (groups of) genes, not whole genotypes, this feature does not imply a quasi-stable ecosystem.

In systems without neutrality new components initiate an activity (by presence) wave that increases with a constant slope and then stops when the component goes extinct. Here however, that increase is often shared between two or (perhaps many) more phenotypically equivalent components, with interchanging presence. If the population size was larger, then there would be greater scope for more than one of these neutral variations to be expressed in the population at any time. But with a population of less than four hundred, and short lifetimes, genetic variation spreads quickly through the population so wave transitions between neutral variants show up almost as either-or events.

We can also observe that as time goes by, the level of neutrality increases: the

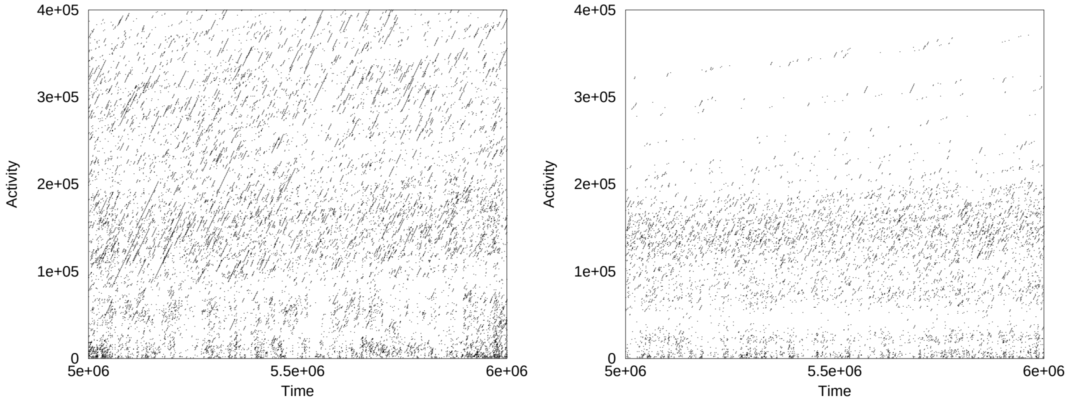


Figure 6.4: Activity point-plots for the real (left) and shadow (right) runs in the last million timesteps, within the shadow's activity range.

average rate of increase for a new component decreases and the number of components in each neutrality-group increases. This is consistent with the expectation from section 6.3 that as component lengths increase, we would see the level of neutrality increase. Were it possible to automatically sort these neutral-variants into groups of phenotypically equivalent components, the resulting waves would be straight lines, as is usual in systems with no neutrality.

Because of this increase in neutral-group size and decrease in average component activity rate of growth, most activity falls within the solid black regions at the bottom of each graph. So it is instructive to look in more detail at the bottom-right corner of the activity wave diagrams. Figure 6.4 shows the activity waves in the last one million timesteps, with just a point for each recorded activity value. Its scale covers the shadow run's full range only, so that the real and shadow data can be easily compared. Notice the long runs of consecutively increasing activity in the real run, and the lack of them in the shadow.

6.4.2 Determining the new-activity range

In order to measure new activity (A_{new}), we must first determine the range $[a_0, a_1]$ of component activity values that should be considered both adaptively significant (so a_0 should be high enough to screen out most non-adaptive activity) and not amongst the

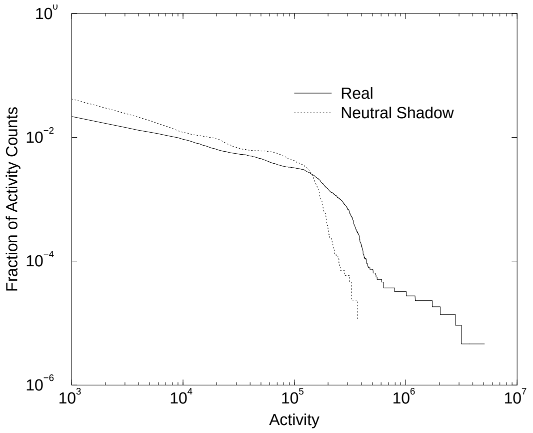


Figure 6.5: Log-log plot of the component activity distributions.

highest activities (so a_1 should be low enough that a good proportion of activities lie above it). The method given in (Bedau, Snyder and Packard, 1998) involves finding the activity value which is equally likely to have occurred in the real run as in the shadow run, and setting $[a_0, a_1]$ to be a narrow band that surrounds it. Figure 6.5 shows the component activity distributions for the run reported here. These cross at an activity of approximately 1.42×10^5. However, it is clear from figure 6.3 that the component activity distributions are far from constant over the run, and that this value increases during the course of the run. Looking again at figure 6.4 shows that by the end of the run, activities of approximately 1.42×10^5 are common in the shadow.

The best approach is probably to recalculate the range $[a_0, a_1]$ in each timestep (and perhaps apply a running average to smooth it), because we are working with a system with increasing component activity values; this would also produce the most impressive results for A_{new}. However, in order to avoid the criticism that I have changed the method, I have chosen here to use a fixed range that screens out the majority of the shadow activity in the final million timesteps of the run. This results in artificially low values for A_{new} early on in the run, but the results are still positive despite this. Looking at figure 6.4 we can see that most of the shadow's activities are below 3×10^5. In fact even in the last million timesteps, less than 3.5% of the shadow activity is above 2.8×10^5. Further, it is also clear from figure 6.4 that an activity of 3×10^5 is far from uncommon in the real run. In fact approximately 27% of the real activities are above

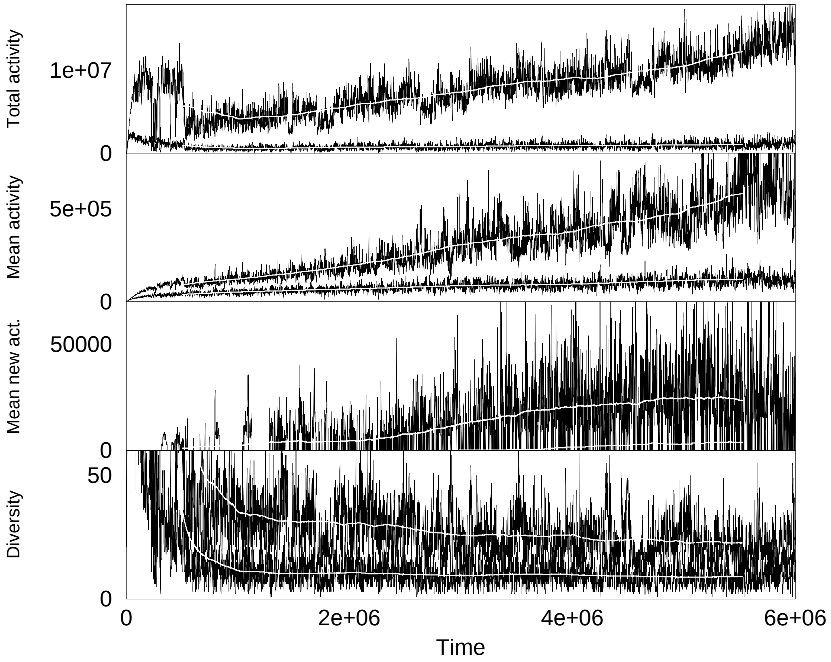


Figure 6.6: Total activity, mean activity, new activity and diversity from a typical Geb run and its shadow. Running averages are shown in white.

3.2×10^5 in the last million timesteps. So the results that follow were calculated using a new-activity range of $[2.8 \times 10^5, 3.2 \times 10^5]$.

6.4.3 Evolutionary statistics and classification

Figure 6.6 shows both total and mean activity increasing rapidly in the real run, and much slower in the shadow run. New activity is positive in the real run, and much higher than in the shadow, which exhibits only occasional blips of new activity. Diversity is bounded in both the real and shadow, as expected (see section 6.3).

Figure 6.7 shows the activity difference[3] between the real and shadow statistics, for both total and mean activity. According to this classification system, these results clearly fall into class 3b: unbounded evolutionary activity.

[3] (Channon, 2001) includes graphs for “excess activity”, defined in (Rechtsteiner and Bedau, 1999) as activity difference divided by shadow activity. However, after further thought I was sceptical about this measure in this context. Mark Bedau later confirmed that its sole purpose is to produce a scale-neutral statistic, and that its use is therefore not appropriate where shadow activity is increasing.

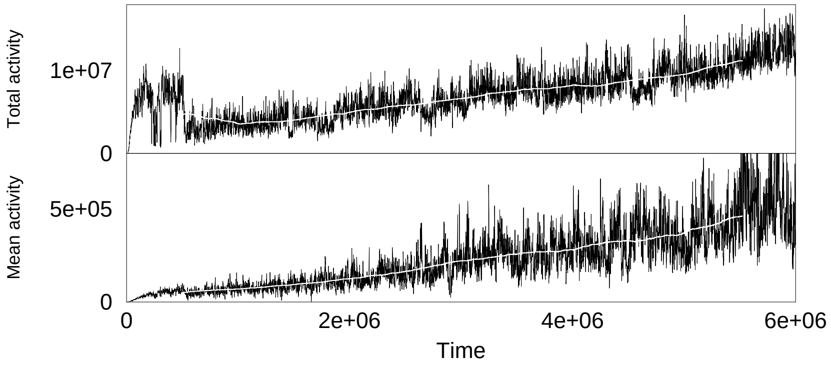


Figure 6.7: Normalised total and mean activity. Running averages are shown in white.

6.4.4 Atypical runs

These results are typical of the twenty runs that were carried out for this set of experiments. However, three of the runs encountered problems, causing their results to be atypical. Two of these effectively met total extinction. In section 6.3.1 I mentioned that I imposed a minimum limit on the number of organisms, in an attempt to avoid total extinction. However, if population size hits this limit and does not increase rapidly, then many reproductions may occur with selection effectively random. This causes evolutionary activity to plummet as adaptive traits are lost. Once lost, this activity cannot be regained, except by the evolution of new adaptive components. These results should not be a cause for concern, for the same reasons mentioned in section 6.3.1: once population sizes can feasibly be increased, the problem should in practice disappear rapidly.

In the third atypical run, it appears that a freak mutation has caused the only existing species to take on a behaviour of never reproducing or moving forward and always turning and trying to kill. Of course this would ordinarily be a very poor strategy. It is easy to imagine how the bad gene (production rule) could have spread through a population of just one species as fit individuals reproduced with the new unfit ones, causing their children to pick up the dominant bad gene. However, one would not expect this to pose a threat to a different species. This is easily verified: introducing just a few organisms from any of the other evolved populations (from the other runs), causes the old organisms to be rapidly displaced by the newcomers. So this result is also not a cause for concern, for the same reason: it is due to the small population size, which cannot support more than one or two species at a time.

6.5 Criticisms and Conclusions

Geb has demonstrated class 3 behaviour, and so passed the test. Does this mean that Geb truly exhibits unbounded evolution? Possibly, for it was designed to verify and extend theories of evolutionary emergent systems generation and so a number of potential pitfalls have been avoided. However, having passed the test the most prudent course of action is to look for weaknesses in the test.

The main concern that I have at this time is that the test relies on normalisation (or validation) from a shadow that can drift away from core aspects of the real run that it is intended to shadow. For example, the components that exist in the real population at any one time (well into evolution) are almost certainly more densely clustered than those in the shadow. So the mutation of a real component is more likely to produce another high-activity component than the mutation of a shadow component. Once the real and shadow populations have been allowed to evolve, we are no longer comparing the real run with a true shadow. One way around this problem would be to develop a method of comparing the real run with a shadow that is regularly reset (both components and activity history) to be identical to the real run but which evolves using random selection between resets. Normalised activity increment between resets could then be determined by comparing real and shadow increments.

My other criticism of the test as it stands is in its use of mean activity when looking for unbounded activity growth, especially when classifying a system as belonging to class 3b. When diversity is bounded, the retention (forever) of a single component results in unbounded mean activity. The test should not be so influenced by such components, and should rather look for trends in typical components. So it is median activity, not mean activity, that should be measured, and required to be unbounded for a

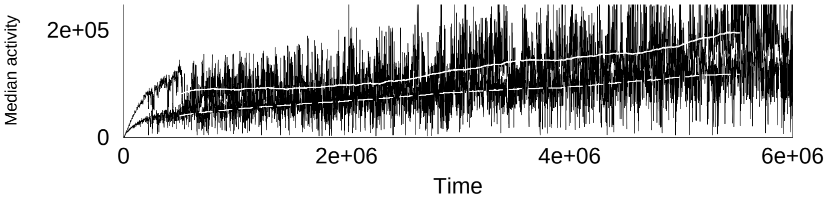


Figure 6.8: Median activity from a typical Geb run and its shadow. Running averages are shown in white: solid for the real run, dashed for its shadow.

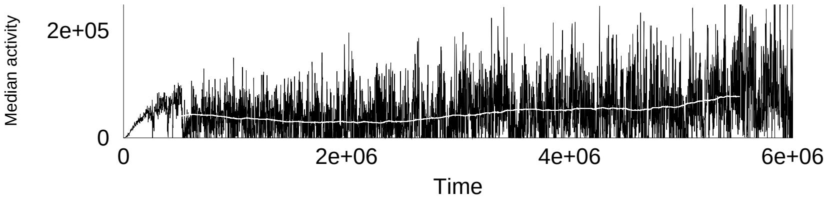


Figure 6.9: Normalised median activity. A running average is shown in white.

system to be classified as within class 3b. The activity waves from Geb's runs indicate that median activity is also unbounded. However, when median activity is measured in both real runs and their shadows, it shows up as unbounded in both (figure 6.8), and it is not possible to make any firm conclusion (bounded/unbounded) about normalised median activity (figure 6.9). However, in light of my main concern above, just as the positive results cannot be trusted, this should not be seen as a cause for concern or fuel for further investigation along this path. The correct course of action is to proceed as outlined above, by developing a shadowing method that regularly resets the shadow state to the real state, and then look at the results again, including median activity.

Chapter 7

Evolutionary statistical analysis II

In the previous chapter I concluded that because there is reason to doubt a method of normalising (or validating) evolutionary statistics that relies on a shadow that can drift away from core aspects of its real run, a new method should be developed that regularly resets the shadow (both components and activity history) to be identical to the real run. This chapter details the development of such a method and reports on its results when applied to Geb.

The basic idea is that immediately after each snapshot (when an entry is made in the component existence record), the shadow run has its components reset to those of the real run, so that we can compare changes in activity in the real run with the changes

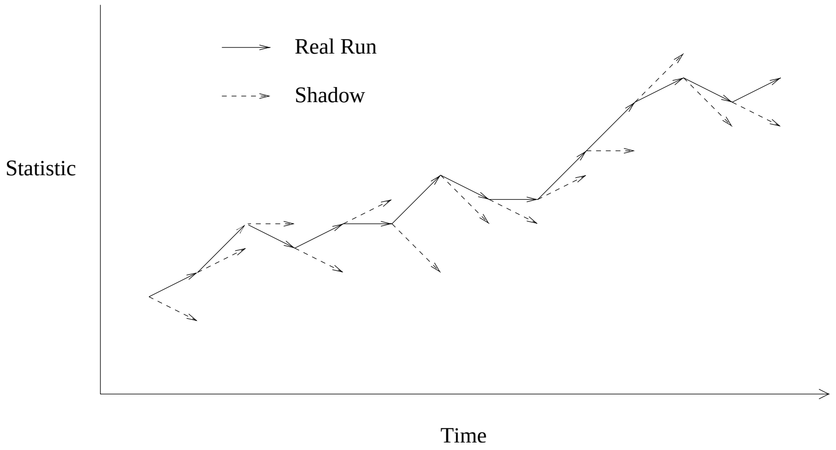


Figure 7.1: Illustration of the shadow-resetting method.

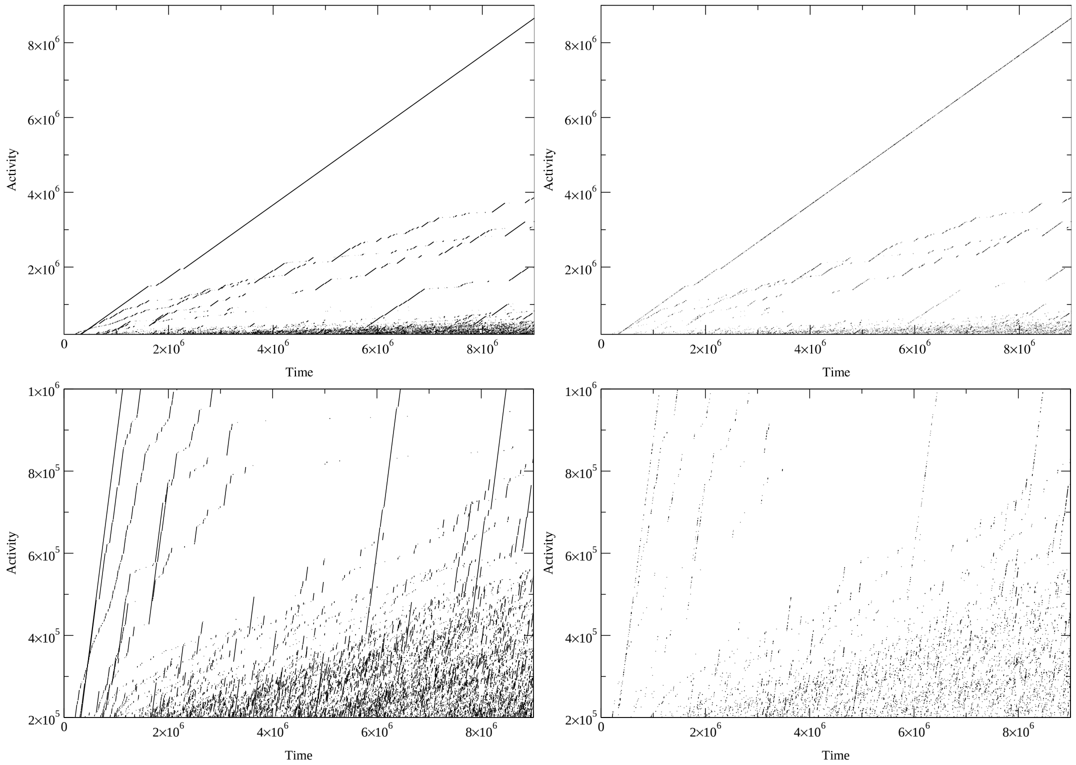


Figure 7.2: Activity wave diagrams for the real (left) and shadow (right) runs, with all horizontal (no-increase) lines removed. The diagrams on the bottom show a magnified view of the activity range below 1 million.

we would expect from random selection. So when calculating evolutionary statistics (and indeed when recording component numbers), the shadow's history is taken to be that of the real run - see figure 7.1.

Most of the results in this chapter are from a typical run, drawn from a set of twenty carried out using this procedure. Atypical variations found within this set are also reported and discussed. The component class used (production rule predecessor plus link details) is the same as in the previous chapter, as are the run implementation details (snapshots every thousand timesteps, isolated component occurrences screened out, minimum number of organisms set to twenty). Figure 7.2 shows the raw real and shadow activity wave diagrams from the typical run. Shadow waves follow the real waves, because the shadow is reset after each snapshot. The shadow loses components between snapshots far more frequently than the real run does. This is especially true of the lower-activity components, as we should expect. Adaptively significant production rules have many redundant copies on a typical genome such that should mutation break the rule at one point, it will still be decoded from elsewhere on the genome. Such components can survive even sustained periods of random selection. Yet even the highest activity components are frequently lost in the shadow and this provides verification that the snapshot interval (one thousand timesteps) is sufficient for comparing activity by presence. Employing a much higher reset frequency would require the use of an activity increment function which is sensitive to the number of occurrences of a component at any one time[1], for example $\Delta_i(t) =$ fraction of organisms that have the i^{th} component at time t. However, because significantly more frequent snapshots are not feasible with the computational resources currently available, comparing activity by presence is still the best option.

Of course it would not make sense to calculate activity statistics based solely on the shadow's component existence record. Perhaps the most obvious course of action (and the one I took at first) is to calculate activity statistics for the shadow on the basis that at each timestep t we use the real run's component record for timesteps before t, and the shadow's component record at t. Figure 7.3 shows the resulting statistics from the shadow, alongside the real run's statistics. Activity (total, mean and median) is unbounded in the shadow only because the shadow has its components reset to those of the real run after each snapshot. On average both total activity and diversity drop sharply in the shadow over each short (1000 timesteps) interval after it has been reset to the real run's state. The shadow's mean and median activity statistics show that (on average) it is the higher activity components that remain in the shadow, in agreement with the discussion (above) of the activity wave diagrams. Do not be confused by the fact that mean and median activity increase in the shadow over each inter-snapshot interval. This is due to the loss of lower activity components, not the result of any increase in component activity.

[1] Thanks to Mark Bedau for bringing this to my attention.

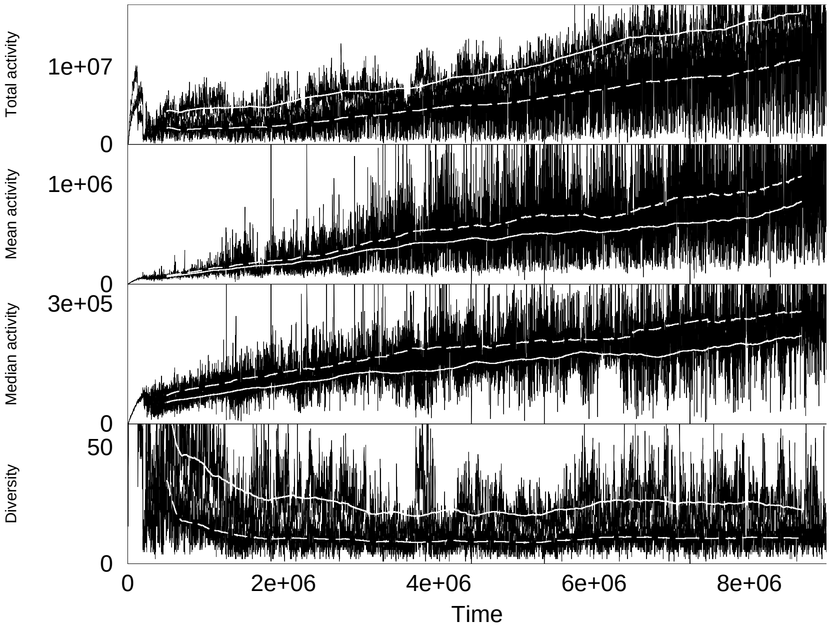


Figure 7.3: Total activity, mean activity, median activity and diversity from a typical Geb run and its regularly-reset shadow. Running averages are shown in white: solid for the real run, dashed for its shadow.

These results are encouraging, but they provide no route to normalising the real run's statistics in order to demonstrate a presence or lack of unbounded growth in, say, median activity. The method so far also provides no sound way of measuring new activity. So naively calculating the shadow's activity statistics leads us to a dead-end. The idea of resetting the shadow run's state to match the real run's state just after each snapshot is a good one, but how can it be used to normalise the real run's statistics?

7.1 Component activity normalisation

The solution is to normalise at the lower level of individual components' activities, rather than at the component-population level. This is done by subtracting the shadow's component activity increment from the real run's component activity increment, for each component. So when calculating activity by presence, a component's normalised activity is incremented if and only if it persists (and is used) in the real run but not in

the shadow, and is decremented if and only if it persists (and is used) in the shadow but not in the real run. Here are the revised statistics:
Real run's component activity increment (by presence).

$$\Delta_i^{\mathrm{R}}(t) = \begin{cases} 1 & \text{if component } i \text{ exists in the real run at } t \\ 0 & \text{otherwise} \end{cases} \tag{7.1}$$

Shadow's component activity increment (by presence).

$$\Delta_i^{\mathrm{S}}(t) = \begin{cases} 1 & \text{if component } i \text{ exists in the shadow at } t \\ 0 & \text{otherwise} \end{cases} \tag{7.2}$$

Normalised component activity increment.

$$\Delta_i^{\mathrm{N}}(t) = \Delta_i^{\mathrm{R}}(t) - \Delta_i^{\mathrm{S}}(t) \tag{7.3}$$

Normalised component activity.

$$a_i^{\mathrm{N}}(t) = \begin{cases} \sum_{\tau=0}^{t} \Delta_i^{\mathrm{N}}(\tau) & \text{if component } i \text{ exists in the real run at } t \\ 0 & \text{otherwise} \end{cases} \tag{7.4}$$

Normalised diversity.

$$D^{\mathrm{N}}(t) = \#\{i : a_i^{\mathrm{N}}(t) > a_0^{\mathrm{N}}\} \tag{7.5}$$

Note that this formula for D^{N} is only a suggestion for how diversity could be normalised when investigating systems with unbounded diversity, by counting the number of components whose normalised activity has passed the threshold at which we consider them to be adaptively significant (see below). This method of normalising diversity is debatable. However, because Geb does not exhibit unbounded diversity I safely ignore that debate here, and do not calculate D^{N} for Geb. This is valid because no claim of unbounded diversity is being made, and because D^{R} (not D^{N}) is the relevant value to use when calculating $\overline{A}^{\mathrm{N}}_{\mathrm{cum}}$, $\widetilde{A}^{\mathrm{N}}_{\mathrm{cum}}$ and $A^{\mathrm{N}}_{\mathrm{new}}$, because D^{R} is the number of components that contribute to $A^{\mathrm{N}}_{\mathrm{cum}}$.

Normalised total cumulative evolutionary activity.

$$A^{\mathrm{N}}_{\mathrm{cum}}(t) = \sum_{i:\substack{\text{component i exists}\\ \text{in the real run at t}}} a^{\mathrm{N}}_i(t) \tag{7.6}$$

Normalised mean cumulative evolutionary activity.

$$\overline{A}^{\mathrm{N}}_{\mathrm{cum}}(t) = \frac{A^{\mathrm{N}}_{\mathrm{cum}}(t)}{D^{\mathrm{R}}(t)} \tag{7.7}$$

Normalised median cumulative evolutionary activity.

$$\widetilde{A}^{\mathrm{N}}_{\mathrm{cum}}(t) = \operatorname*{Median}_{i:\substack{\text{component i exists}\\ \text{in the real run at t}}} a^{\mathrm{N}}_i(t) \tag{7.8}$$

Normalised new activity per component.

$$A^{\mathrm{N}}_{\mathrm{new}}(t) = \frac{1}{D^{\mathrm{R}}(t)} \sum_{i:\text{component i `new'}} a^{\mathrm{N}}_i(t) \tag{7.9}$$

See below for the details of calculating normalised new activity per component.

This is clearly the better approach, for it produces normalised component activities that measure how much each component's activity has increased above the increase that would have occurred had selection been random. So a component's normalised activity is a direct measure of the degree to which adaptive selection in the real run is causing the component to persist (and be used).

7.1.1 Determining the normalised new-activity criteria

The final requirement, before these statistics can be used to classify evolutionary dynamics, is a method of determining when a component is newly adaptively significant. This involves finding the (normalised) activity level a_0^{N} at which a component can be considered adaptively significant, and a procedure for dropping a component from the list of new components. For the second of these concerns, a simple upper bound cannot be used, because normalised activity can both increase and decrease, so a component could potentially be considered ‘new’ forever. The simplest (and adequate) solution is to consider a component to be ‘new’ (newly adaptively significant) in the snapshot at which its activity reaches a_0^{N}, and never after that. So each component is considered new at most once. This leaves the issue of determining a_0^{N}.

If the presence or absence of a component confers no adaptive advantage or disadvantage, then the two systems (real and shadow) will be equivalent for this component. Further, which is used as the reset-to system (after each snapshot) does not matter to the activity of the component. So the (normalised) activity distribution for this class of components will be symmetric about the origin. Therefore, provided we can make the assumption that the most negative normalised activity encountered during a run is from such a component, we can negate this value to find a level at which normalised activity can be considered adaptively significant. Even if this assumption does not hold, negating the most negative activity encountered provides a value above which activity can be considered adaptively significant, even if this bound is higher than necessary.

This method can be expected to work well when activity is calculated by presence (as it is in Bedau and Packard’s test and so also here), where changes in component activity (Δ_i) are small when compared with the activities of non-adaptive components. However, we should not expect it to provide a good bound when calculating activity by, for example, usage ($\Delta_i =$ #components i at t), where the most negative activities arise from neutral mutations of high usage components, some of which a shadow will encounter before its real run does.

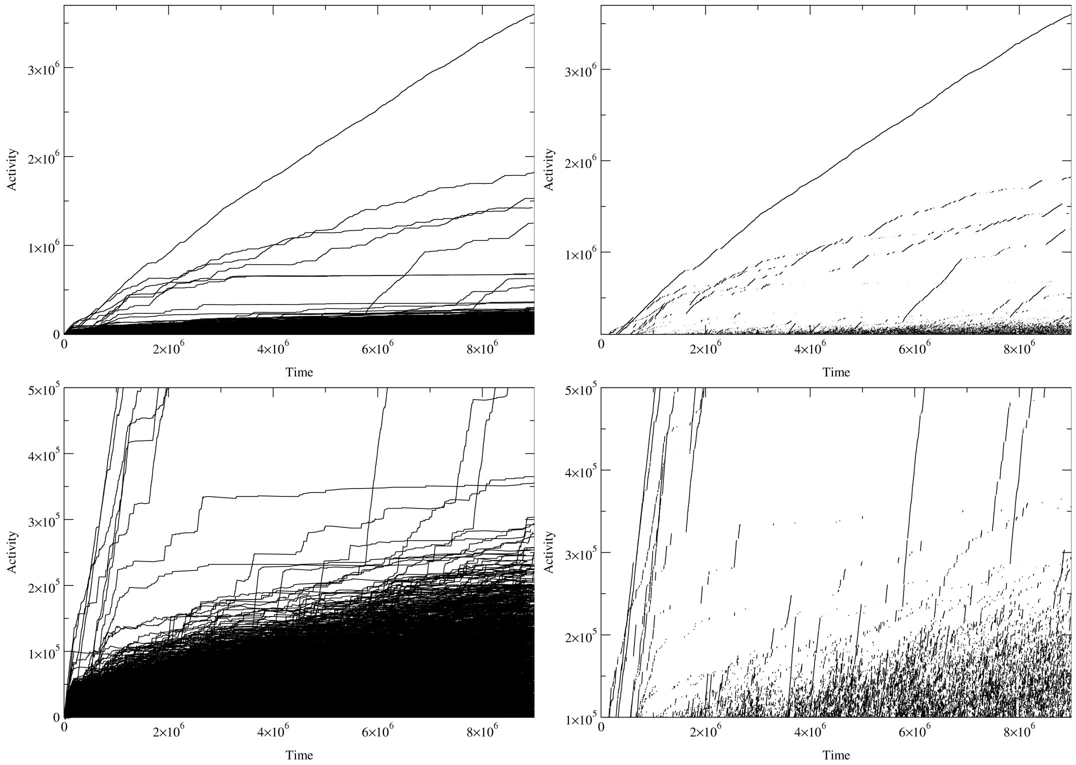


Figure 7.4: Normalised activity wave diagrams. The diagrams on the right have had all horizontal (no-increase) lines removed. The diagrams on the bottom show a magnified view of the activity range below 0.5 million.

7.2 Results and Discussion

This section contains the results from the typical run, already discussed above, drawn from the full set of twenty runs. Atypical variations are discussed at the end of this section.

Figure 7.4 shows the resulting normalised activity waves. Notice that the activity values are significantly lower than before normalisation - see figure 7.2.

In each of the twenty runs, the lowest normalised activity encountered was greater than -30, with -10 being a more typical value. For simplicity when calculating the evolutionary statistics for these runs, I used a new-activity threshold (a_0^{N}) of 30 on all runs. Figure 7.5 shows the resulting statistics for the typical run. Normalised median activity is unbounded (as are normalised total activity and normalised mean

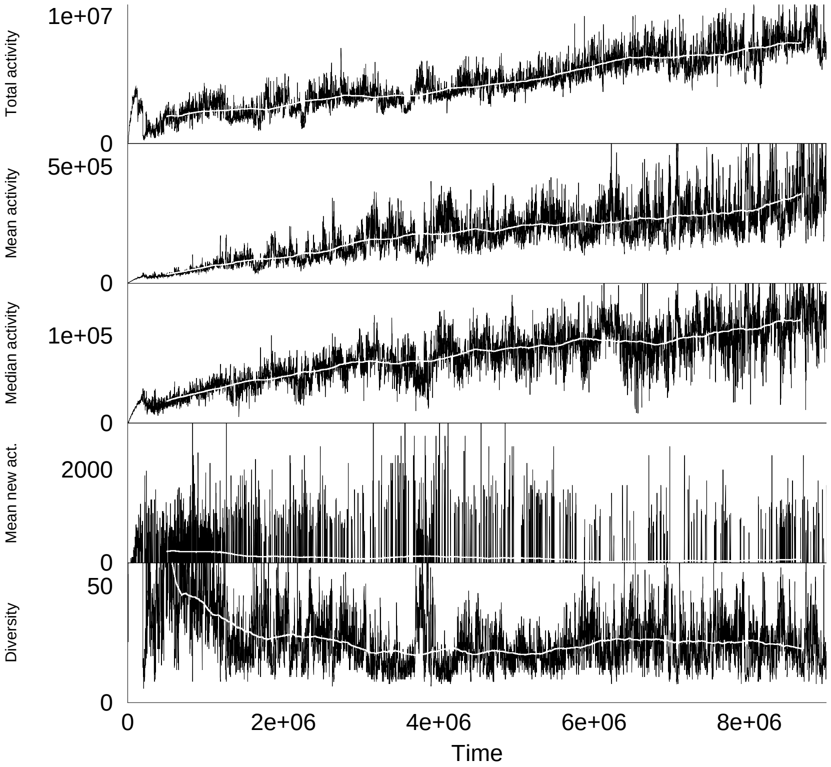


Figure 7.5: Normalised total activity, normalised mean activity, normalised median activity, normalised new activity, and real diversity. Running averages are shown in white.

activity) and normalised new activity is positive. These results clearly fall into class 3b (according to this classification system): unbounded evolutionary activity.

These results are typical of the twenty runs that were carried out for this set of experiments. However, three of the runs effectively met total extinction, and two of the runs stagnated when the only existing species stopped reproducing. Both of these possibilities were encountered in the previous chapter. As discussed there, they should not be a cause for concern.

It is easy to demonstrate that unbounded growth (or rather unbounded non-monotonic but *directed* increase) in activity, with positive new activity, is not a trivial consequence of unbounded genotype length. Consider the analogous system Geb^R, in which selection is random but all other details are as in Geb. Whenever a (randomly chosen) real organism is killed in Geb^R, a randomly chosen organism is also killed in its shadow. Whenever a real organism is born in Geb^R (as the product of two randomly chosen real organisms), a new shadow organism is born as the product of two randomly chosen shadow organisms. Of course running either real or shadow system from the same snapshot more than once would produce different results on each run, because of the stochastic nature of the systems. So normalised activity would unfold as a random walk, with 'step' probability distribution changing at each snapshot but always symmetric about zero.

Could unbounded growth in activity, with positive new activity, be a trivial consequence of unbounded genotype length in a biotic selection system? Certainly not, for new activity drops to zero in systems such as Tierra (see chapter 6). Could unbounded growth in activity be a trivial consequence of unbounded genotype length in a biotic selection system that exhibits unending positive new activity? No, because the requirement remains that activity be retained, so that it can accumulate. For example, a (diversity-bounded) biotic selection system that continually generates new components only by mutation along (phenotypically) neutral networks would only be able to *use* a finite number of neutral variants at any one time. It would lose activity whenever a component is lost from (ceases being used in) the system.

7.3 Conclusions

Both of the concerns of the previous chapter have been addressed. The new shadowing method used here ensures that the normalisation of statistics is through a shadow that remains true to its real run, and median rather than (or rather as well as) mean activity has been used in the classification.

Geb has demonstrated class 3 behaviour, and so passed the test. And this time we can have a greater degree of confidence in the results. However, this is a new variant of a previous test, and it is not beyond possibility that it could be improved upon. Certainty in these results can only come about through the application of the test to a range of evolutionary systems. That may take some time, since there are no other known artificial systems that even pass the original test. So for now I must be content with the conclusion that there is reason to believe that Geb exhibits unbounded evolution.

If evolution is unbounded in Geb then, following the reasoning of chapter 3, it is through biotic competition, the source of evolutionary emergence. While the caution of the previous paragraph is warranted, it is at least possible to say with certainty that these results qualitatively surpass those from previous artificial evolutionary systems. No previous biotic selection artificial evolutionary system has demonstrated unbounded evolutionary activity with positive new activity. As such these results provide validation of the theory behind Geb's design: a design constructed to satisfy the set of requirements for an evolutionary system within which increasingly complex advantageous behaviours can emerge, as uncovered by evaluating previous artificial systems.

Chapter 8

Conclusions and future research

This thesis has so far presented three major contributions. First, an evaluation of existing artificial selection and natural selection systems was used to uncover a set of known requirements for the generation of evolutionary systems within which increasingly complex advantageous behaviours can emerge. The most important of these requirements are biotic selection and a sufficiently correlated genotype to phenotype mapping. Second a system was constructed that satisfied these requirements, providing a unique platform (as demonstrated by its results). Third evolutionary statistical methods were extended to better analyse candidate open-ended artificial evolutionary systems. However, it is the fourth major contribution that moves us on a step, ready to start work on the next generation of systems. By learning from the successes and failures of Geb, we can add to our list of requirements.

In this chapter I first look back over the requirements uncovered by evaluating previous systems, and assess the extent to which Geb is successful in satisfying these requirements. Then I consider the failures of Geb and the new requirements that these reveal.

8.1 Initial requirements

The two main requirements uncovered by evaluating previous systems, and that Geb was designed to satisfy, are the use of biotic selection and a sufficiently correlated genotype to phenotype mapping.

8.1.1 Biotic selection

The emergence of increasingly complex advantageous behaviours requires the perpetuation of evolutionary emergence. While computational emergence can arise via abiotic selection, evolutionary emergence requires biotic selection (by my definitions). The logical progression or aim is the perpetuation of evolutionary emergence via naturally arising coevolution.

Geb uses biotic selection, with no (or negligible) abiotic selection and no lifetime learning. So there can be no doubt that all results reported are due to biotic selection.

8.1.2 Correlated genotype to phenotype mapping

The perpetuation of evolutionary emergence also requires long-term incremental evolution and so what we evolve and how we evolve it must be chosen accordingly. The initial groundwork on “how” has already been covered by SAGA theory – by using low enough mutation rates that the population evolves as nearly-converged species, with crossover assimilating beneficial mutations into the species. As for what class of entities to (attempt to) evolve, computer program instructions are too brittle. Even the use of template matching cannot overcome that fact. Neural networks are a clear choice because of their graceful degradation.

Biotic selection research has an opportunity to lead the field, through the evolution of neural controllers within virtual environments, towards the emergence of increasingly complex advantageous behaviours. The work presented in this thesis has started down that route, with some success. In work involving biotic 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 recognising how the epistasis (lack of correlation) of fitness landscape issue transfers across. Certain ontogenetic (developmental) and interaction systems can result in individuals with similar genotypes but very different phenotypes. Geb organisms satisfy this criterion, because offspring resemble their parents (but are not identical). Geb's results prove it to be suited to long-term incremental artificial evolution. This alone is a significant result for a modular developmental system. The behaviours identified are encouraging too, for the increases in complexity were clearly advantageous and in ways not specified by the design – evolutionary emergence.

8.2 New requirements

Two new requirements have been uncovered from the failures of Geb. Note that while these may seen obvious in retrospect, so do the requirements above and yet I know of only one previous system (PolyWorld) that even came close to satisfying both of those.

8.2.1 Transparency of behaviours

We want to be able to understand (or at least identify) novel behaviours as they emerge. With relatively simple agents, we can analyse their controllers (programs, neural networks, or other) directly to determine the resulting behaviours. But as the complexity of evolved controllers increases, this becomes infeasible, especially when using neural networks or other PDP controllers. So, until the agents themselves can help us understand their behaviours (a very distant prospect), we can only observe the resulting behaviours and attempt to identify innovation. An embedded presence, and more so a body, can provide us with the ability to observe the resulting movements of our agents. The logical aim is therefore to develop future systems such that behavioural descriptions are as easy to generate as possible, probably by constructing the systems such that behaviours will be transparent to human observers. Brain-body coevolution could be used to aid in the observation (identification and analysis) of emergent behaviours.

8.2.2 An open range of lowest-level actions

Specifying the available lowest-level actions constrains organisms around these actions and so limits evolution. Despite showing the important new result of evolutionary emergent advantageous behaviours (not specified within the initial system) from a system suited to long-term incremental evolution, all basic (inter-)actions in Geb were as specified within the initial system and not evolvable.

If only the 'brains' of agents can evolve, then the range of resulting behaviours is limited by body design. This is especially true at the current (early) stage of research into AI-Life, where we should expect most evolved behaviours to be motor-based: following, fleeing, circling, flocking, running, jumping, etc.. Allowing bodies to coevolve with their brains could provide for a far greater range of behaviours, no longer restricted to sequences of predefined actions.

So brain-body coevolution surfaces as one possible route to satisfying both of the new requirements. Sims' (1994a, 1994b) evolution (by *abiotic* selection) of virtual creatures produced behaviours which are easy to identify and describe, operate over many time-steps in a non-simplistic way, and are largely the result of the evolved embodiment of the organisms. So Sims' abiotic selection of "blockies" might provide a good starting model for this route. However, because we are interested in long-term evolution, the computational cost of such physics simulation is a significant problem. Largely as a result of increases in readily-available computational power, but also thanks in part to the availability of satisfactory Physics Engine packages, there has recently been a small number of attempts to replicate Sims' work (Taylor and Massey, 2001), but none of these have succeeded in producing such impressive results.

8.3 Future work

My research objective remains the same as specified in the introduction to this thesis: to add to the list of known requirements for the generation of evolutionary systems within which increasingly complex advantageous behaviours can emerge. This is an iterative approach, and the next iteration involves designing a system that satisfies all the requirements identified so far, and investigating its successes and failures.

By continuing this exploration, we can progress towards artificial systems with both unbounded evolutionary activity and behaviours that we can understand.

Appendix A

Source code and experimental data

The CD that accompanies this thesis contains the source code for Geb and data from all forty of the runs discussed in chapters 6 and 7.

The source code is in the gzipped tar file `geb_v07c.src.tar.gz`.

The data for chapter 6's runs is in the directory `RUNS/no_resets`.

The data for chapter 7's runs is in the directory `RUNS/resets`.

The data for each run is contained in a tar file that has been compressed with bzip2. To view evolutionary statistics from a run you will also need to copy the directory `RUNS/ANALYSIS_FILES`, which contains the file `makefile_rs` and six graphing configuration files, all copied from `geb_v07c.src.tar.gz`. These files are linked to by each run (once expanded from its tar file). Please see `makefile_rs` for instructions on how to run the evolutionary statistics calculations for a new run of Geb, and on how to display the results from an existing run.

All analysis was carried out under GNU-Linux and `makefile_rs` requires a GNU system. For graphing you will also need xmgr and its successor xmgrace. GNU-Linux, xmgr and xmgrace are all freely available on the internet.

References

Adami, C. and C. T. Brown (1994). Evolutionary learning in the 2D artificial life system ‘Avida’ . In R. A. Brooks and P. Maes (Eds.), *Proceedings of Artificial Life IV*, Cambridge, MA, pp. 377–381. MIT Press.

Arthur, W. B. (1994). Inductive reasoning and bounded rationality. *American Economic Review (Papers and Proceedings) 84*, 406–411.

Axelrod, R. (1984). *The Evolution of Cooperation*. New York: Basic Books.

Bedau, M. A. and C. T. Brown (1999). Visualizing evolutionary activity of genotypes. *Artificial Life 5*(1), 17–35.

Bedau, M. A. and N. H. Packard (1991). Measurement of evolutionary activity, teleology, and life. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen (Eds.), *Proceedings of Artificial Life II*, Redwood City, CA, pp. 431–461. Addison-Wesley.

Bedau, M. A., E. Snyder, C. T. Brown, and N. H. Packard (1997). A comparison of evolutionary activity in artificial evolving systems and the biosphere. In P. Husbands and I. Harvey (Eds.), *Proceedings of the Fourth European Conference on Artificial Life (ECAL97), Brighton*, Cambridge, MA, pp. 125–134. MIT Press.

Bedau, M. A., E. Snyder, and N. H. Packard (1998). A classification of long-term evolutionary dynamics. In C. Adami, R. Belew, H. Kitano, and C. Taylor (Eds.), *Proceedings of Artificial Life VI, Los Angeles*, Cambridge, MA, pp. 228–237. MIT Press.

Benton, M. J. (Ed.) (1993). *The Fossil Record 2*. London: Chapman and Hall.

Boers, E. J. W. and H. Kuiper (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.

Boers, E. J. W., H. Kuiper, B. L. M. Happel, and I. G. Sprinkhuizen-Kuyper (1993). Designing modular artificial neural networks. Technical Report 93-24, Leiden University.

Brooks, R. A. (1991a). Intelligence without reason. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, San Mateo, CA, pp. 569–595. Morgan Kauffman.

Brooks, R. A. (1991b). Intelligence without representation. *Artificial Intelligence 47*, 139–159.

Bullock, S. 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. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen (Eds.), *Proceedings of Artificial Life II*, Redwood City, CA, pp. 775–797. Addison-Wesley.

Channon, A. D. (1996). The evolutionary emergence route to artificial intelligence. Master's thesis, School of Cognitive and Computing Sciences, University of Sussex.

Channon, A. D. (2000). Three evolvability requirements for open-ended evolution. In C. C. Maley and E. Boudreau (Eds.), *Artificial Life VII Workshop Proceedings*, pp. 39–40.

Channon, A. D. (2001). Passing the ALife test: Activity statistics classify evolution in Geb as unbounded. In J. Kelemen and P. Sosik (Eds.), *Advances in Artificial Life: Proceedings of the Sixth European Conference on Artificial Life (ECAL2001), Prague*, Heidelberg, pp. 417–426. Springer-Verlag.

Channon, A. D. and R. I. Damper (1998a). The evolutionary emergence of socially intelligent agents. In B. Edmonds and K. Dautenhahn (Eds.), *Socially Situated Intelligence: a workshop held at SAB'98, University of Zurich Technical Report*, Zurich, pp. 41–49.

Channon, A. D. and R. I. Damper (1998b). Evolving novel behaviors via natural selection. In C. Adami, R. Belew, H. Kitano, and C. Taylor (Eds.), *Proceedings of Artificial Life VI, Los Angeles*, Cambridge, MA, pp. 384–388. MIT Press.

Channon, A. D. and R. I. Damper (1998c). Perpetuating evolutionary emergence. In R. Pfeifer, B. Blumberg, J.-A. Meyer, and S. Wilson (Eds.), *From Animals to Animats 5: Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior (SAB98), Zurich*, Cambridge, MA, pp. 534–539. MIT Press.

Channon, A. D. and R. I. Damper (2000). Towards the evolutionary emergence of increasingly complex advantageous behaviours. *International Journal of Systems Science 31*(7), 843–860.

Cliff, D., I. Harvey, and P. Husbands (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 G. Miller (1995). Tracking the red queen: Measurements of adaptive progress in co-evolutionary simulations. In F. Morán, A. Moreno, J. J. Merelo, and P. Chacón (Eds.), *Advances in Artificial Life: Proceedings of the Third European Conference on Artificial Life (ECAL95), Granada*, New York, pp. 200–218. Springer Verlag.

Cliff, D. T., P. Husbands, and I. Harvey (1993). Evolving recurrent dynamical networks for robot control. In R. Albrecht, C. Reeves, and N. Steele (Eds.), *Proceedings of ANNGA93, the International Conference on Neural Networks and Genetic Algorithms*, pp. 428–435. Springer-Verlag.

Damper, R. I. (2000). Emergence and levels of abstraction. *International Journal of Systems Science 31*(7), 811–818.

Darwin, C. R. (1859). *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. London: Murray.

Dawkins, R. (1976). *The Selfish Gene*. Oxford: Oxford University Press.

Dawkins, R. (1983). *The Extended Phenotype: The Gene as a Unit of Selection*. Oxford: Oxford University Press.

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

Eigen, M. and P. Schuster (1979). *The Hypercycle: A Principle of Natural Self-Organization*. Springer-Verlag.

Fisher, R. A. (1930). *The Genetical Theory of Natural Selection*. Oxford: Oxford University Press.

Fogel, L. J. (1962). Autonomous automata. *Industrial Research 4*, 14–19.

Fogel, L. J., A. J. Owens, and M. J. Walsh (1966). *Artificial Intelligence through Simulated Evolution*. New York: John Wiley.

Fraser, A. S. (1957). Simulation of genetic systems by automatic digital computers. *Australian Journal of Biological Sciences 10*, 484–491.

Gattiker, J. R. and D. S. Wilson (1996). Population dynamics in a radically epistatic genetic system. In *Proceedings of INBS95: First International Symposium on Intelligence in Neural and Biological Systems*, pp. 218–224. IEEE Computer Society Press.

Gers, F. and H. de Garis (1996). CAM-Brain: A new model for ATR's cellular automata based artificial brain project. In *Proceedings of ICES'96: International Conference on Evolvable Systems*.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley.

Gould, S. J. (1996). *Life's Grandeur: The Spread of Excellence from Plato to Darwin*. London: Jonathan Cape. Also publised by Harmony Books, United States, 1996, as "Full House".

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

Harnad, S. (1993). Artificial life: Synthetic vs. virtual. In C. G. Langton (Ed.), *Proceedings of Artificial Life III*, Reading, MA, pp. 539–552. Addison-Wesley.

Harvey, I. (1992). Species adaptation genetic algorithms: A basis for a continuing SAGA. In F. J. Varela and P. Bourgine (Eds.), *Towards a Practice of Autonomous*

Systems: Proceedings of the First European Conference on Artificial Life, Cambridge, MA, pp. 346–354. MIT Press/Bradford Books. Also available as report CSRP221, School of Cognitive and Computing Sciences, University of Sussex.

Harvey, I. (1993a). *The Artificial Evolution of Adaptive Behaviour*. Ph.D. thesis, School of Cognitive and Computing Sciences, University of Sussex.

Harvey, I. (1993b). Evolutionary robotics and SAGA: the case for hill crawling and tournament selection. In C. G. Langton (Ed.), *Proceedings of Artificial Life III*, Reading, MA, pp. 299–326. Addison-Wesley. Also available as report CSRP222, School of Cognitive and Computing Sciences, University of Sussex.

Harvey, I. (1997a). Artificial evolution for real problems. In T. Gomi (Ed.), *Evolutionary Robotics: From Intelligent Robots to Artificial Life (ER'97). Proceedings of the 5th International Symposium on Evolutionary Robotics, Tokyo*, Ontario Canada, pp. 127–149. AAI Books.

Harvey, I. (1997b). Cognition is not computation: Evolution is not optimisation. In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud (Eds.), *Proceedings of International Conference on Artificial Neural Networks (ICANN 97), Special Session on Adaptive Autonomous Agents at ICANN97, Lausanne, Switzerland*, Berlin, pp. 685–690. Springer-Verlag.

Harvey, I., P. Husbands, and D. Cliff (1992). Issues in evolutionary robotics. In J. A. Meyer, H. Roitblat, and S. Wilson (Eds.), *From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior (SAB92)*, Cambridge, MA, pp. 364–373. MIT Press/Bradford Books. Also available as report CSRP219, School of Cognitive and Computing Sciences, University of Sussex.

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

Holland, J. H. (1962). Outline for a logical theory of adaptive systems. *Journal of the ACM 9*(3), 297–314.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press. 2nd edition: MIT Press, 1992.

Holland, J. H. (1992). Genetic algorithms. *Scientific American 267*(1), 44–50.

Holland, J. H. (1994). Echoing emergence: Objectives, rough definitions, and speculations for Echo-class models. In G. A. Cowan, D. Pines, and D. Meltzer (Eds.), *Santa Fe Institute Studies in the Sciences of Complexity XIX*. Addison-Wesley.

Holland, J. H. (1997). *Emergence: from chaos to order*. Reading, MA: Addison-Wesley.

Hraber, P. T. and B. T. Milne (1997). Community assembly in a model ecosystem. *Ecological Modelling 103*, 267–285.

Husbands, P., I. Harvey, and D. Cliff (1993). Analysing recurrent dynamical networks evolved for robot control. In *Proceedings of the Third IEE International Conference on Artificial Neural Networks (IEE-ANN93)*, London, pp. 158–162. IEE Press.

Huynen, M. A., P. F. Stadler, and W. Fontana (1996). Smoothness within ruggedness: The role of neutrality in adaptation. *Proceedings of the National Academy of Science USA 93*, 397–401.

Jakobi, N. (1997). Half-baked, ad-hoc and noisy: Minimal simulations for evolutionary robotics. In P. Husbands and I. Harvey (Eds.), *Proceedings of the Fourth European Conference on Artificial Life (ECAL97), Brighton*, Cambridge, MA, pp. 348–357. MIT Press.

Juric, M. (1994). An anti-adaptationist approach to genetic algorithms. In *Proceedings of IEEE Conference on Evolutionary Computation*, pp. 619–623. IEEE Press. Vol. II.

Kauffman, S. and S. Levin (1987). Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology 128*, 11–45.

Kerce, K. F. (1993). Evolution of artificial neural networks grounded in virtual ecosystems. Master's thesis, Department of Computer Science, Florida State University College of Arts and Sciences.

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

Koza, J. R. (1990). Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems. Technical Report STAN-CS-90-1314, Department of Computer Science, Stanford University, Palo Alto, CA.

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

Koza, J. R. (1993). Artificial life: Spontaneous emergence of self-replicating and evolutionary self-improving computer programs. In C. G. Langton (Ed.), *Proceedings of Artificial Life III*, Reading, MA, pp. 225–262. Addison-Wesley.

Langton, C. G. (1989). Artificial life. In C. G. Langton (Ed.), *Santa Fe Institute Studies in the Sciences of Complexity, Vol VI: Proceedings of the Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (Artificial Life I)*, Redwood City, CA, pp. 1–47. Addison-Wesley.

Lee, D. W. and K. B. Sim (1998). Evolving cellular automata neural systems 1 (ECANS1). In *Proceedings of The Third Asian Fuzzy System Symposium, Masan, Korea*, pp. 158–163.

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

Lindgren, K. (1991). Evolutionary phenomena in simple dynamics. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen (Eds.), *Proceedings of Artificial Life II*, Redwood City, CA, pp. 295–312. Addison-Wesley.

Lotka, A. J. (1925). *Elements of Physical Biology*. Baltimore, OH: Williams and Wilkins. Reprinted as *Elements of Mathematical Biology*, Dover Press, 1956.

Maley, C. C. (1999). Four steps toward open-ended evolution. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO99)*, San Francisco, CA, pp. 1336–1343. Morgan Kaufmann. Volume 2.

Maynard Smith, J. (1978). *The Evolution of Sex*. Cambridge University Press.

Mayr, E. (1960). The emergence of evolutionary novelties. In S. Tax (Ed.), *Evolution after Darwin, The University of Chicago Centennial, Vol. I: The Evolution of Life*, pp. 349–380. Chicago, IL: University of Chicago Press.

McCulloch, W. S. and W. Pitts (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics 5*, 115–133.

Nolfi, S. and D. Parisi (1995). Learning to adapt to changing environments in evolving neural networks. Technical Report 95-15, Department of Neural Systems and Artificial Life, Institute of Psychology, National Research Council, Rome, Italy.

Nowak, M. and P. Schuster (1989). Error thresholds of replication in finite populations, mutation frequencies and the onset of Muller's ratchet. *Journal of Theoretical Biology 137*, 375–395.

Packard, N. H. (1989). Intrinsic adaptation in a simple model for evolution. In C. G. Langton (Ed.), *Santa Fe Institute Studies in the Sciences of Complexity, Vol VI: Proceedings of the Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (Artificial Life I)*, Redwood City, CA, pp. 141–155. Addison-Wesley.

Ray, T. S. (1991). An approach to the synthesis of life. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen (Eds.), *Proceedings of Artificial Life II*, Redwood City, CA, pp. 371–408. Addison-Wesley.

Ray, T. S. (1992). Evolution, ecology and optimization of digital organisms. Technical Report 92-08-042, Santa Fe Institute, Santa Fe, NM.

Ray, T. S. (1997). Selecting naturally for differentiation. In J. R. Koza, K. D. Deb, M. Dorigo, D. B. Fogel, M. Garzon, H. Iba, and R. L. Riolo (Eds.), *Genetic Programming 1997: Proceedings of the Second Annual Conference*, San Francisco, CA, pp. 414–419. Morgan Kaufmann.

Ray, T. S. (1998). La vita artificiale. In W. Gilbert and G. T. Valentini (Eds.), *Frontiere Della Vita, Estratto Dal Volume I*, pp. 109–125. Istituto della Enciclopedia Italiana, Fondata da Giovanni Treccani.

Ray, T. S. and J. Hart (1998). Evolution of differentiated multi-threaded digital organisms. In C. Adami, R. Belew, H. Kitano, and C. Taylor (Eds.), *Proceedings of Artificial Life VI, Los Angeles*, Cambridge, MA, pp. 295–304. MIT Press.

Rechtsteiner, A. and M. A. Bedau (1999). A generic neutral model for measuring excess evolutionary activity of genotypes. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO99)*, San Francisco, CA, pp. 1366–1373. Morgan Kaufmann. Volume 2.

Reynolds, C. W. (1992). An evolved, vision-based behavioral model of coordinated group motion. In J. A. Meyer, H. Roitblat, and S. Wilson (Eds.), *From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior (SAB92)*, Cambridge, MA, pp. 384–392. MIT Press.

Reynolds, C. W. (1994). Competition, coevolution and the game of Tag. In R. A. Brooks and P. Maes (Eds.), *Proceedings of Artificial Life IV*, Cambridge, MA, pp. 59–69. MIT Press.

Rucker, R. (1993). *Artificial Life Lab*. Corte Madera, CA: Waite Group Press.

Sannier, A. V. and E. D. Goodman (1987). Genetic learning procedures in distributed environments. Technical report, A.H. Case Center for Computer-Aided Engineering and Manufacturing, Michigan State University, East Lansing, MI.

Sepkoski, J. J., Jr.. (1992). A compendium of fossil marine animal families, 2nd edition. Milwaukee Public Museum Contributions in Biology and Geology, vol. 83.

Sims, K. (1994a). Evolving 3d morphology and behavior by competition. In R. A. Brooks and P. Maes (Eds.), *Proceedings of Artificial Life IV*, Cambridge, MA, pp. 28–39. MIT Press.

Sims, K. (1994b). Evolving virtual creatures. In *Proceedings of Computer Graphics (Siggraph '94) Annual Conference Proceedings*, New York, pp. 15–22. ACM Siggraph.

Skipper, J. (1992). The computer zoo – evolution in a box. In F. J. Varela and P. Bourgine (Eds.), *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, Cambridge, MA, pp. 355–364. MIT Press/Bradford Books.

Standish, R. K. (2000). An ecolab perspective on the Bedau evolutionary statistics. In M. A. Bedau, J. S. McCaskill, N. H. Packard, and S. Rasmussen (Eds.), *Proceedings of Artificial Life VII*, Cambridge, MA, pp. 238–242. MIT Press.

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

Stork, D. G., B. Jackson, and S. Walker (1991). 'Non-optimality' via pre-adaptation in simple neural systems. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen (Eds.), *Proceedings of Artificial Life II*, Redwood City, CA, pp. 409–429. Addison-Wesley.

Taylor, T. and J. Hallam (1997). Studying evolution with self-replicating computer programs. In P. Husbands and I. Harvey (Eds.), *Proceedings of the Fourth European Conference on Artificial Life (ECAL97), Brighton*, Cambridge, MA, pp. 550–559. MIT Press.

Taylor, T. and J. Hallam (1998). Replaying the tape: An investigation into the role of contingency in evolution. In C. Adami, R. Belew, H. Kitano, and C. Taylor (Eds.), *Proceedings of Artificial Life VI, Los Angeles*, Cambridge, MA, pp. 256–265. MIT Press.

Taylor, T. and C. Massey (2001). Recent developments in the evolution of morphologies and controllers for physically simulated creatures. *Artificial Life 7*(1), 77–87.

Thompson, A. (1996). An evolved circuit, intrinsic in silicon, entwined with physics. In T. Higuchi (Ed.), *Proceedings of the First International Conference on Evolvable Systems: From Biology to Hardware (ICES96).* Springer-Verlag.

Thompson, A. (1997). Temperature in natural and artificial systems. In P. Husbands and I. Harvey (Eds.), *Proceedings of the Fourth European Conference on Artificial Life (ECAL97), Brighton*, Cambridge, MA, pp. 388–397. MIT Press.

Vaario, J. and K. Shimohara (1997). Synthesis of developmental and evolutionary modeling of adaptive autonomous agents. In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud (Eds.), *Proceedings of International Conference on Artificial Neural Networks (ICANN 97), Special Session on Adaptive Autonomous Agents at ICANN97, Lausanne, Switzerland*, Berlin, pp. 721–725. Springer-Verlag.

Ventrella, J. (1996). Sexual swimmers: Emergent morphology and locomotion without a fitness function. In P. Maes, M. Mataric, J.-A. Meyer, J. Pollack, and S. Wilson (Eds.), *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB96)*, Cambridge, MA, pp. 484–493. MIT Press/Bradford Books.

Volterra, V. (1926). Variations and fluctuations of the number of individuals in animal species living together. In R. N. Chapman (Ed.), *Animal Ecology*, pp. 409–448. New York: McGraw-Hill.

Ward, C. R., F. Gobet, and G. Kendall (2001). Evolving collective behavior in an artificial ecology. *Artificial Life 7*(2), 191–209.

Wilson, S. W. (1991). The animat path to AI. In S. W. Wilson and J.-A. Meyer (Eds.), *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior (SAB)*, Cambridge, MA, pp. 15–21. Bradford Books/MIT Press.

Yaeger, L. (1993). Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or PolyWorld: Life in a new context. In C. G. Langton (Ed.), *Proceedings of Artificial Life III*, Reading, MA, pp. 263–298. Addison-Wesley.

Zaera, N., D. Cliff, and J. Bruten (1996). (Not) evolving collective behaviours in synthetic fish. In P. Maes, M. Mataric, J.-A. Meyer, J. Pollack, and S. Wilson (Eds.), *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB96)*, Cambridge, MA, pp. 635–644. MIT Press/Bradford Books.