David Ackley and Michael Littman
Cognitive Science Research Group, Bellcore, 445 South Street, Morristown, NJ 07960

Interactions Between Learning and Evolution

A program of research into weakly supervised learning algorithms led us to ask if learning could occur given only natural selection as feedback. We developed an algorithm that combined evolution and learning, and tested it in an artificial environment populated with adaptive and non-adaptive organisms. We found that learning and evolution together were more successful than either alone in producing adaptive populations that survived to the end of our simulation. In a case study testing long-term stability, we simulated one well-adapted population far beyond the original time limit. The story of that population's success and ultimate demise involves both familiar and novel effects in evolutionary biology and learning algorithms.

1. EVOLUTION, LEARNING, ARTIFICIAL LIFE

The processes of life involve change at many scales of space and time, from the small, fast biochemical cycles of cell energetics, to the growth and aging of an organism, to the rise and fall of entire populations, entire species, entire orders. Choosing a spatiotemporal scale emphasizes the changes at that scale, rendering smaller scales essentially as noise, larger scales essentially as constant. Many useful investigations

Artificial Life II, SFI Studies in the Sciences of Complexity, vol. X, edited by
C. G. Langton, C. Taylor, J. D. Farmer, & S. Rasmussen, Addison-Wesley, 1991

can be performed within a given scale of life—there are striking mixtures of order and complexity almost everywhere one looks—but, of course, such investigations will be fundamentally limited by the assumptions of the rendering. Smaller scales are not always insignificant, sometimes they become decisive; larger scales are not always constant, sometimes they become cataclysmic.

Such limitations due to choice of scale can be partially eliminated by devising models that explicitly address multiple spatial and temporal scales (or, more generally, by increasing the *spatiotemporal bandwidth* covered by a model). In this chapter, we study interactions between adaptive processes on two adjacent levels—individuals and populations. Learning is a process at the individual level whereby an organism becomes optimized for its environment; evolution operates similarly at the level of populations or species. The two scales are evident: An entire lifetime of learning is but one tick of the clock for evolution. One trade-off between the two processes is readily apparent: Learning is facilitated by long individual lifetimes, whereas evolution benefits from rapidly passing generations. How else do they interact?

Such multiple time-scale questions are generally very difficult to answer in the natural world. The depths of evolutionary history can be probed via the fossil record and molecular genetics, but such techniques provide only hints about the day-to-day histories of long-passed organisms. Similarly, learning abilities can be studied in live organisms, but feasible length experiments can observe an evolutionarily significant number of generations only with relatively short-lived and learning-limited species.

As the available computational power grows, the "artificial life" experimental approach—based on computer simulations of systems modeling selected aspects of the natural world—becomes more and more feasible. The rich diversity of material in this book demonstrates some of the ways in which this power can be exploited. For present purposes, it is the power to create artificial organisms that combine reasonably long *simulated* lives—allowing for substantial learning—with reasonably short *real-time* lives—allowing us to perform experiments that span many generations. Given the power of a computer workstation, an artificial creature can live a simulated lifetime encompassing thousands of learning opportunities in only seconds of elapsed time, and small populations of such organisms can be tracked over thousands of generations in only days.

This chapter introduces, demonstrates, and studies an adaptation strategy called *evolutionary reinforcement learning* (ERL), which combines genetic evolution with neural network learning, and an artificial life "ecosystem" called AL, within which populations of ERL-driven adaptive "agents" struggle for survival. Although ERL and AL are tremendously impoverished models—possessing only a few stereotypical properties selected from the richness and depth of adaptation and the natural world—they give rise to a broad range of behaviors and phenomena. AL should be distinguished from single-level population-size models of species interactions, such as Lotka-Volterra or Rosenstein-McArthur-Zweig.[25] Instead of modeling an ecosystem in terms of *a priori* birth, interaction, and death rates, AL is a moment-to-moment simulation of each organism's lifetime in the ecosystem. Quantities such as birth and death rates are not input parameters; instead they are

observables whose values reflect the interacting consequences of each organism's decisions.

Section 2 presents ERL.[2] Section 3 presents AL and summarizes a comparative study that supports the basic hypothesis that evolution and learning can mutually aid each other. Section 4 presents an in-depth historical study of one successful population, seeking an account of the population's longevity, and an account of its eventual extinction. A phenomenon known as the "Baldwin effect"[7,24] plays a role in the former case, and a phenomenon that we call *shielding* plays a role in the latter. Section 5 contains discussion, and Section 6 concludes the chapter.

2. EVOLUTIONARY REINFORCEMENT LEARNING

Learning algorithms require some sort of feedback to function, but different approaches vary widely in the amount and nature of the feedback required. One fundamental question is: How limited can the feedback be? *Supervised* paradigms[21,22] supply immediate detailed correct answers as feedback; the system must learn to produce them on demand. *Reinforcement* paradigms[8,27] supply less—only judgments of right or wrong—so the system must first discover and then remember the correct responses. Viewed as a learning algorithm, the paradigm of natural selection[11] supplies still less—only birth and death. How can an organism learn in such circumstances, where the only unarguable sign of failure is the organism's own death, and the reproduction process preserves only the genetic information, which is unaffected by any learning performed during the organism's life?

"Evolutionary Reinforcement Learning"[2] (ERL) provides one answer to this question. In ERL, we allow evolution to specify not only inherited *behaviors*, but also inherited *goals* that are used to guide learning. We do this by constructing a genetic code that specifies two major components. The first component is a set of initial values for the weights of an "action network" that maps from sensory input to behavior. These weights represent an innate set of behaviors that the individual inherits directly from its parents.

The second component is an "evaluation network" that maps from sensory input to a scalar value representing the "goodness" of the current situation. By learning to move from "bad" situations to "better" situations—modifying its action network weights in the process—an individual achieves the goals of learning passed down from its predecessors. Whether those inherited goals are actually sensible or not is, of course, a separate issue; insofar as learning is a factor, each organism stakes its life on the *assumption* that its inherited evaluation function is reliable.

Figure 1 depicts the three central structures possessed by an individual. The *genetic code* is a string of bits which the individual receives at birth. It is unchanged by learning and is passed from parents to offspring modified by *crossover* (genetic recombination[17]) and *mutation*. (Asexual reproduction is employed if no mate can be located; in such cases only mutation applies.)

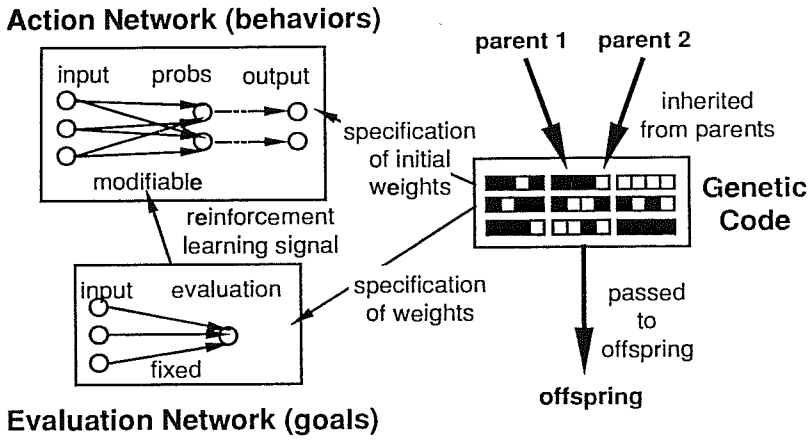

FIGURE 1 Overview of ERL.

The *evaluation network* is a feed-forward neural network that maps the organism's sensory input to a real-valued scalar. The weights of this network are determined solely by the genetic code and they do not change during the lifetime of the individual.

The *action network* is a feed-forward neural network that maps sensory input to behavioral output. The initial weights of this network are specified genetically. However, they are adjusted over time by a reinforcement learning algorithm that rewards behaviors that lead to an increase in the evaluation and punish those that lead to a decline.

To limit the computational costs, in the simulations presented here we used single-layer networks for both evaluations and actions. By design, however, all aspects of ERL carry over to the multi-layer case.

2.1. THE ERL ALGORITHM

The details of ERL are summarized in Figure 2. The first procedure is an implementation of evolution; the second, an implementation of learning. A few comments may help clarify the more and less critical aspects of our approach.

In steps B1 and B2, a spatial distance measure is presumed to exist for purposes of mate selection. In world AL (see Section 3) such a metric is readily available, but one is not required for the algorithm to make sense. In principle, mates could be chosen at random, or by some more elaborate mating ritual, and many interesting questions arise in such situations. We chose this very simple method to avoid the procedural complexities attending more realistic courting behavior.

ERL: Evolutionary reinforcement learning

At Birth

Given: A parent agent A and an offspring O to be initialized.

B1 *Clone.* Copy A's genetic code to O. If there are one or more other agents within a prespecified distance of A, pick the closest such agent B and go to B2, otherwise go to B3.

B2 *Crossover.* Modify O's genetic code by crossing with B's using two random crossover points.

B3 *Mutate.* With low probability mutate O's genetic code by flipping random bits.

B4 *Elaborate.* Translate O's genetic code into weights for O's evaluation network, and and initial weights for O's action network.

Living at time t:

Given: A living agent A, and a new current input vector I_t.

L1 *Evaluation.* Propagate I_t through the evaluation network producing a scalar evaluation E_t.

L2 *Learning.* If this is A's day of birth, go to L3. Otherwise, produce a reinforcement signal by comparison with the previous evaluation: $R_t = E_t - E_{t-1}$. Use the CRBP learning algorithm to update the action net with respect to the previous action X_{t-1} and previous input I_{t-1}.

L3 *Behave.* Use the CRBP performance algorithm to generate a new action X_t based on I_t. Perform the chosen action.

FIGURE 2 Summary of ERL.

Although we use a particular reinforcement learning algorithm called CRBP (Section 2.2) in steps L1–L3, in principle any associative reinforcement learning algorithm supporting multiple output bits could be employed. Regardless of the specific choice of algorithm, a *reinforcement function* is required for learning to proceed. From a computational point of view, the primary novel contribution embodied in ERL is the inheritable evaluation function that converts long-time-scale feedback (lifetime-to-lifetime natural selection) into short-time-scale feedback (moment-to-moment reinforcement signals). As we shall see in Section 4, there are both benefits and risks inherent in this approach.

It is important to recognize that natural selection, when viewed as a computational paradigm for search and learning, places severe restrictions on possible adaptation strategies. There are only two circumstances in which a strategy has decisions to make. The first situation—concerned with learning—is the choice of behavior for a given agent at a given time step, and the second situation—concerned with evolution—is the passage of genetic information to the offspring when a birth occurs. Everything else is determined by the "laws of nature" of the world at hand.

For example, death requires no action on the part of the strategy. Also, in contrast to conventional genetic algorithms,[13,17] a strategy is not free to specify the existence and maintenance of any particular population size, nor who lives, who dies, and who reproduces. The strategy influences such decisions only indirectly, via the interactions between the (static and dynamic) properties of the world and the behavior of the agents governed by the strategy.

2.2. CRBP

The ERL algorithm description in Figure 2 refers to a reinforcement learning algorithm called CRBP—*complementary reinforcement back-propagation*[3]—in the implementation of the action network. Although there is neither space nor pressing need to enter into an extensive discussion of CRBP here, an algorithm summary and a few brief comments may be useful for the interested reader.

Figure 3 summarizes CRBP as used in ERL to implement steps L1–L3. Compared to previous presentations of CRBP,[3] this version performs the action function and the learning function backwards, to reinforce the action at time t based on the input at time $t+1$. Thus, this version of CRBP is a simple *temporal* reinforcement learning algorithm.[26] Exploring the effects of incorporating more sophisticated *temporal difference* algorithms[27] would be an interesting extension to this work.

CRBP: Complementary reinforcement backpropagation (ERL version)

Given: A backpropagation network with input dimensionality n and output dimensionality m, and a reinforcement function $f(\Re^n, \Re^n) \rightarrow r$. Let $t = 0$.

1. Receive vector $i_t \in \Re^n$. If $t = 0$ go to 6. Otherwise compute reinforcement $r = f(i_t, i_{t-1})$.
2. Generate output errors e_j. If $r > 0$, let $e_j = (o_j - s_j)s_j(1 - s_j)$, otherwise let $e_j = (1 - o_j - s_j)s_j(1 - s_j)$.
3. Backpropagate errors.
4. Update weights. $\Delta w_{jk} = \eta e_k s_j$, using $\eta = \eta_+$ if $r \geq 0$, and $\eta = \eta_-$ otherwise, with parameters $\eta_+, \eta_- > 0$.
5. Forward propagate again to produce new s_j's. Generate temporary output vector o^*. If ($r > 0$ and $o^* \neq o$) or ($r < 0$ and $o^* = o$), go to 2.
6. Set network input to i_t. Forward propagate to produce s_j's.
7. Generate a binary output vector o. Given a uniform random variable $\xi \in [0, 1]$ and parameter $0 < \nu \leq 1$,

$$o_j = \begin{cases} 1, & \text{if } (s_j - \frac{1}{2})/\nu + \frac{1}{2} \geq \xi; \\ 0, & \text{otherwise.} \end{cases}$$

8. Perform the action associated with o. Let $t = t + 1$. Go to 1.

FIGURE 3 Summary of CRBP as used in ERL.

CRBP extends the *back-propagation* neural network learning algorithm[22] to reinforcement learning.[6,28] Back-propagation by itself is a supervised learning algorithm in which the desired outputs corresponding to given inputs are provided externally. By contrast, in reinforcement learning the network itself is given the task of *discovering* desired outputs—i.e., those that produce a positive reinforcement signal. This search task is implemented by step 7, where weighted random numbers are used to generate an output vector. Thus, the ERL action network has the job of specifying output *probabilities* conditional on the current inputs.

When the reinforcement signal is positive, the learning task is fairly clear: The generated output vector should be made *more probable* given the same input vector. If the output vector is taken as the *desired target*, back-propagation learning will do exactly that. Negative reinforcement, however, only says that the generated output vector is wrong, without suggesting which other output vector would be right. What should the desired target be?

Different reinforcement learning algorithms emerge depending on how that question is answered. One strategy[6,28] sidesteps the issue by taking the generated output vector as the *undesired target*—in essence, simply flipping the signs of the errors produced in the positive reinforcement case. CRBP embodies a somewhat stronger heuristic—that the desired output on negative reinforcement is the *complement* of the generated output. Without *a priori* knowledge of the reinforcement function, all treatments of negative reinforcement are fundamentally heuristic, but fortunately, since search is an integral part of reinforcement learning,[1] the occasional failure of the assumption need not be catastrophic.

The loop introduced in step 5 implements a simple form of "mental rehearsal." On positive reinforcement, the reward continues until another stochastic output generation produces the same result as the initial success, and on negative reinforcement, the punishment continues until another output generation produces something different than the initial failure. In empirical studies of CRBP,[3] this loop improved learning speed substantially without much overhead.

3. WORLD AL

We needed a source of natural selection to illustrate and evaluate ERL, so we constructed an artificial life world we called "AL." In doing so, we had to balance the desire for richness and complex interactions against the need for compactness and computational tractability. The result has much in common with other artificial life worlds.[5,18,20,29] AL is a two-dimensional 100×100 array of cells populated by adaptive ERL agents and non-adaptive carnivores, plants, trees, and walls. The world is summarized in Figure 4 and can also be seen in a video demonstration.[4] The various rates and thresholds that determine the artificial physics of AL are all fixed at constant values, as are the biological and physiological properties of

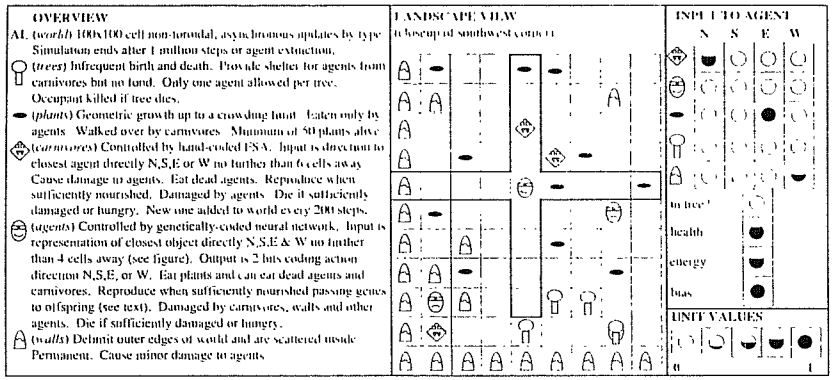

FIGURE 4 Summary of World AL.

AL agent and carnivore action semantics			
Contents of Target Cell	*Cell Visual Appearance*	*Effect of Agent action*	*Effect of Carnivore action*
Empty	Empty	Enter	Enter
Plant	Plant	Eat all	Enter
Empty Tree	Tree	Climb	No effect†
Agent* in Tree	Tree	No effect	No effect†
Wall	Wall	Damage self	Damage self†
Living carnivore‡	Carnivore	Damage other	Damage other†
Dead carnivore‡	Carnivore	Eat some	Eat some†
Living agent	Agent	Damage other	Damage other
Dead agent	Agent	Eat some	Eat some

* Living or dead.

† Carnivores as programmed will not choose these moves.

‡ Perhaps accompanied by a plant.

FIGURE 5 Effects of agent and carnivore actions.

all inhabitants. All that must be supplied is an algorithm for agent learning and evolution, and the name of the game is *maximize agent population survival time.*

The agents receive as input the visual appearance of the closest object not further than four cells away in each of the four compass directions. Carnivores can see objects six cells distant. All visual inputs in a given direction take value zero if only empty cells are visible, and otherwise the input corresponding to the visual appearance of the occupied cell takes a value from 0.5 to 1.0 proportional to the closeness of the cell. An additional binary input indicates whether an agent is currently on the ground or in a tree. Agents also have "proprioceptors" indicating the amount of energy and health they possess. Agents must produce as output a 2-bit pattern indicating the compass direction to their choice of *target cell.* Although it seems likely that the specific details of the action interpretations in AL are not critical to our basic results, for the sake of concreteness and as an aid to the reader's intuition, Figure 5 presents the complete "semantics" of agent and carnivore actions.

Agents reproduce by accumulating enough energy from food and die by running low on energy or health. Injured but alive agents and carnivores recover spontaneously over time. Dead agents and carnivores are eaten or simply decay until their energy is gone. Carnivores reproduce by eating enough agents and die by starvation (almost always), or agent-inflicted damage (very rarely—in a slugfest between a healthy agent and a healthy carnivore, the agent always loses). At regular intervals a carnivore is created in a random empty cell. Also, though they have not been observed to be necessary, procedures are included to reseed plants and trees if their numbers fall perilously low. Agents, of course, as the objects of our population longevity study, receive no such safety nets.

Simulations of ERL in AL display phenomena at several time scales.[4] Observing at highest resolution, agents are seen moving about or collecting in corners, feeding or starving, encountering carnivores and escaping or not, and so on. AL is not an overly kind world: Most initial agent populations die out quite quickly. Observing summary statistics at the $\times 100$ time scale, in those populations that survive the most apparent features are irregular predator-prey oscillations involving plants, agents, and carnivores, interspersed with periods of stable or slowly changing population sizes. In the simulation considered in Section 4, agent population sizes were oscillating in the 30–60 range when one million steps were reached (see the $\times 1,000$ view in Figure 6).

3.1. A COMPARATIVE STUDY

Our evaluation of the algorithm consisted of the following test: we ran the simulation on 100 random initial agent populations and recorded the time at which they eventually went extinct (up to 1 million time steps). We then used just the evolution component of the algorithm (E), just the learning component (L), and neither (F— Fixed random action networks), and ran the same test. As a baseline, we tested Brownian agents (B), who simply wander the world at random, ignoring their inputs.

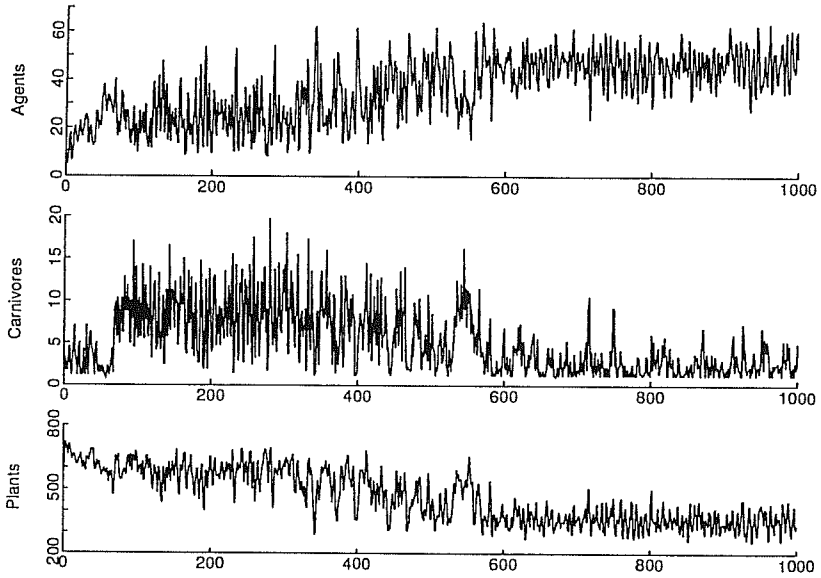

FIGURE 6 ERL in AL: Species population sizes vs. time (×1000) for a long-term successful agent population.

Figure 7 displays the percentage of initial populations that survived to various times for each of the five variations. No more than 18% of all populations reached 10,000 time steps and only 1.8% reached the 1 million time-step simulation limit.

There is a clear distinction in the first half-million time steps or so between the two algorithms that included learning (ERL, L) and their non-learning counterparts (E, F). The latter two algorithms even did poorly compared to the "brainless" B agents. Learning appears to contribute towards keeping the agents alive during this period.

Above about half a million time steps, ERL begins to pull away from learning-only (L), suggesting that evolution has an impact at this timescale. ERL finally goes on to produce seven populations that last to the 1 million time-step limit.

We were surprised that evolution without learning did so poorly, and that learning without evolution did so well. The former was surprising since evolution without learning is a common approach to artificial life, and the latter was surprising since, without evolution to improve the evaluation functions, strategy L can never move beyond on the randomly generated evaluation functions found in the initial populations.

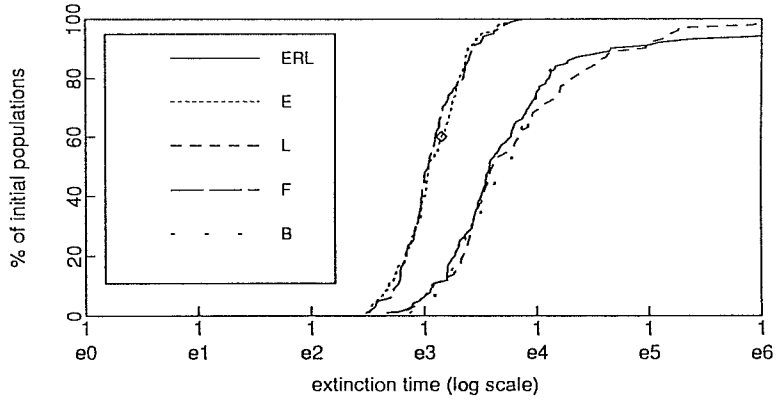

FIGURE 7 Cumulative plots showing the distributions of population lifetimes generated by the five strategies. The point marked with a diamond, for example, indicates that 60% of the strategy E initial populations were extinct by about 1500 time steps.

We hypothesize that evolution alone has difficulty because survival in AL is no trivial matter: Most agents with randomly generated action networks die quickly (*viz.* the strategy F results). This puts evolution at a disadvantage: Either the population dies out completely before there is time to evolve, or a single survivor becomes the ancestor of all subsequent agents. In the latter case, all the agents are close genetic kin, leaving little diversity for evolution to operate upon.

The success of learning alone was noteworthy. It is easy enough to conclude merely that the space of genetic codes for action networks is more difficult to search than the code space for action-plus-evaluation networks, so that strategy L could simply "luck into" good initial populations often enough to make the difference. However, that cannot be the whole story. After all, the code space for L is thirty orders of magnitude larger than that for E, so one might expect it to be harder to search. Our explanation is that *it is easier to generate a good evaluation function than a good action function.*

Notice, for example, that there are two output units in the action network, but only one in the evaluation network. To specify an action in response to a particular input requires specifying two weights, but to specify that a particular input is "good" requires only one weight. Furthermore, if the evaluation function specifies that the energy level input is positively valued, then there is pressure towards making "eating moves" more probable regardless of the direction of the food source. Thus, *one* evolutionarily specified learning weight can have the effect of specifying the *eight* action weights involved in response to plants. The insight that strategy L

highlights is that it can be much easier to specify *goals* than *implementations*—assuming, of course, the existence of a search and learning process adequate to fill in the details.

ERL, which combines evolution and learning into a single system, is better at producing long-lasting populations than either alone. The interaction of these components can result in successful adaptations and stable populations.

4. A LONGITUDINAL STUDY

One advantage of artificial life studies over natural world studies, as we have seen, is the fact that experimental conditions can be so easily controlled, precisely repeated, and systematically varied. Another advantage is the abundance of data that such experiments provide, at whatever granularity we choose.

The object of our study was the population depicted in Figure 6 and on video.[4] It was doing well as it reached its 1-million-step birthday. Even in the depths of population declines, dozens of agents survived. Its very long-term prospects—on the multimillion-step timescale—looked good. We reset the simulator to that population's initial seed, let it loose with no upper time limit, and went about other business. By the next morning it had regained the million-step milestone and pushed into new territory.

Days went by—more millions of steps—and the population survived. Checking in on the simulation, it was clear that matters were more complex than they appeared at 1-million steps. There were periods of very large agent populations, and dangerous agent population crashes. Eventually, after about a week, almost at the 9 million mark, the sole remaining agent, a member of the 3,216th generation, died. Figure 8 displays the population sizes over the entire run. What happened?

Each agent's genetic sequence consists of 336 bits—84 weights total at 4 bits per redundantly encoded weight. On average there were 40 agents alive at any one time and each agent lived approximately 4,000 time steps. Almost 100,000 agents were born during the entire run, yielding slightly over 4 megabytes of genetic information from start to finish. Genealogical and census data added several more megabytes.

How does one go about reducing all this data? Averaged population size changes indicated substantial dynamics in the multimillion step regime, but suggested little in the way of explanation. We hoped to identify the relative importance of learning and evolution in the survival of the agents and perhaps even detect changes in the importance over time. If we were lucky, we hoped to find a genetic explanation for the instability that lead to the population's eventual extinction.

The biological literature suggested an approach we found fruitful: *functional constraints*. By looking at the changes in given sites on the genome over the millenia, biologists argue that one can assess those sites' relative importance to survival.

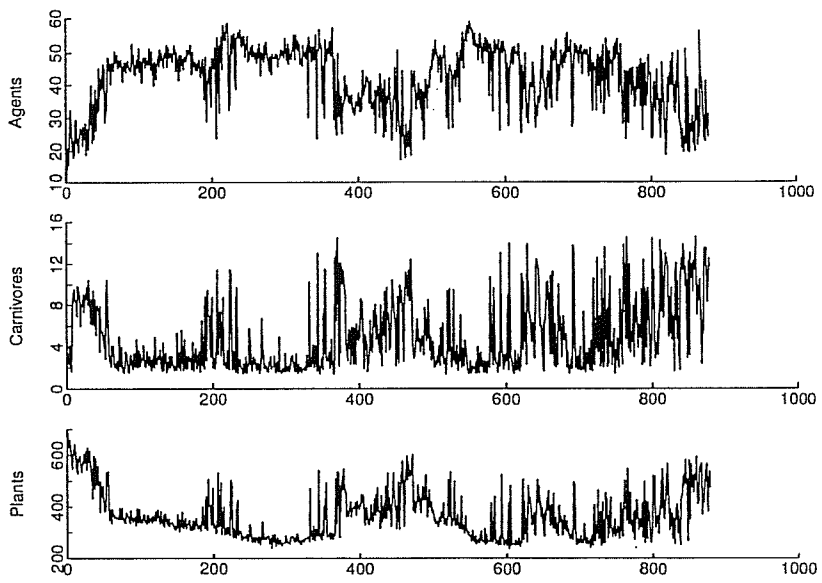

FIGURE 8 Population sizes vs. time (×10,000) for full run.

We came across functional constraints in an article by Gould,[14] who describes a lovely application of the idea, recounting the work of Hendriks, Leunissen, Nevo, Bloemendal, and de Jong[15] on the blind mole rat *spalax ehrenbergi*. The argument runs as follows: at the molecular level, one site is as likely as any other to mutate during reproduction. On the one hand, some mutations will occur in irrelevant portions of the genome (for instance, the so-called *pseudogenes* which are evidently non-expressed "commented out" portions of DNA). Such changes will have no effect on the probability of survival of the offspring, and will, therefore, tend to accumulate in the population over time.

On the other hand, some mutations will disturb genetic sites that contain information crucial to the survival of the organism. These changes will tend to disrupt the functioning of the organism and will tend not to be passed down to later generations. Thus, the *lack* of observed mutations in a gene sequence, over time, suggests that sequence is "functionally constrained" by natural selection.

In the case of *S. ehrenbergi*, the researchers looked at the genes coding for the protein αA-crystallin, which plays a role in the lens of vertebrates.[15] Figure 9 presents some of their data, which they gleaned from a painstaking mix of fossil data analysis and comparative biochemistry. A base mutation rate is computed from observed pseudogene mutation rates. In sighted rodent species, the gene for

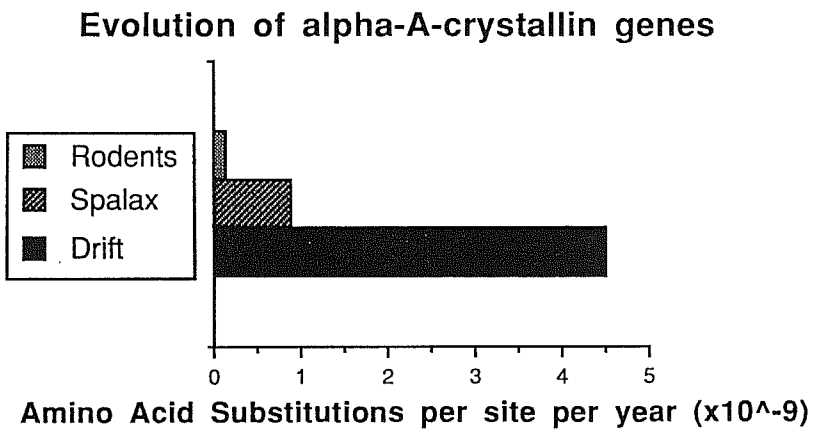

FIGURE 9 Relative mutation rates for various DNA sequences in *spalax ehrenbergi*. (Based on data from Hendriks et al.[15]).

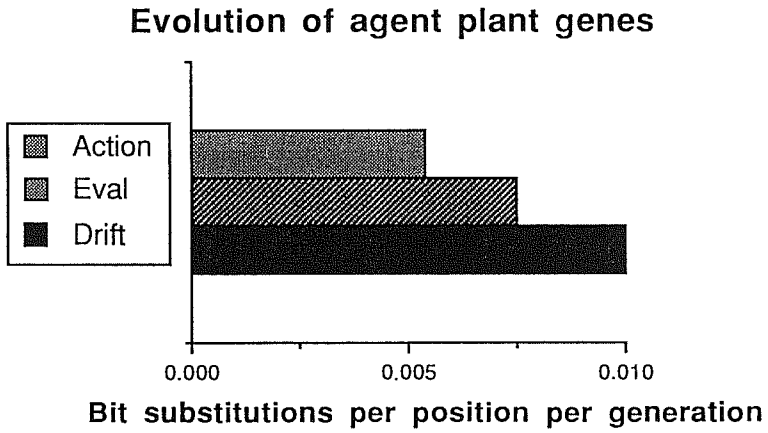

FIGURE 10 Relative mutation rates for various gene sequences in AL agents.

αA-crystallin mutates very slowly compared to the base rate. In *Spalax*, they find that the same gene is changing more rapidly than in sighted rodents but less fast than pseudogenes. This supports the inference that this sight-related protein has *some* survival value to the blind mole rat, but not as much as in sighted rodents. The functional constraints on a gene are inversely related to its observed rate of mutation.

We applied this technique to the agents in AL. Figure 10 displays the observed rates of change of three types of genes: the action-related genes associated with plants, the learning-related genes associated with plants, and a set of genes that happened to code for nothing in our simulation.

As in the natural world example, we found that the non-coding genes changed much more quickly over the generations than did the plant-related genes. We can therefore infer that the plant-related genes were functionally constrained and had a positive impact on fitness over the lifetime of the population.

Unlike our biological counterparts, with our densely sampled data, we can easily perform more detailed data analyses. By partitioning the data into pre-600,000 and post-600,000 time-step periods, we see that the relative importance of learning and evolution changes (Figure 11). During the first 600,000 time steps, there is little change in the genes related to plant evaluation. Therefore, changes in learning goals are being selected against—learning is very important to survival. After 600,000, however, it is the genes controlling the initial action towards plants which are conserved more. Therefore, inherited behaviors are more significant during this time.

One effect suggested by the above data is that in the initial periods, the successful behaviors are being represented in the evaluation network and that somehow, later, these behaviors have showed up in the action networks. In other words, it appears that in the beginning, agents have learning-related genes that state "Plants are good"[1] and from this, learn to approach plants. Later in the simulation, however, their action-related genes recommend approaching plants right from birth.

One might be tempted view this as a Lamarckian effect, with changes during an organism's lifetime somehow being transmitted genetically, but the mechanisms of ERL make direct transmission of acquired characteristics impossible. Fortunately, there is a Darwinian explanation, an effect first suggested in the biological literature around the turn of the century by J. M. Baldwin (and several others) that has come to be referred to as the "Baldwin Effect."[7,24] (Baldwin's own term for the phenomenon—*organic selection*—did not persist, but recently it has been proposed as a more general term for a variety of effects including Baldwin's.[23])

With ERL in AL, the Baldwin Effect often appears this way: In the beginning era of successful populations, agents possess (mostly by luck) learning genes telling

[1] "Plants are good" refers to both the antecedent— closeness to plants—and the consequent— increase in energy— of eating. Similarly "carnivores are bad" refers to both closeness to carnivores and decrease in health.

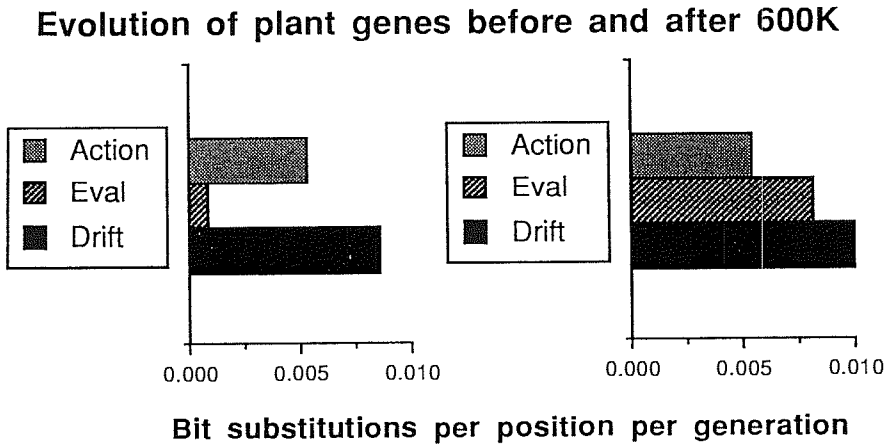

FIGURE 11 Relative importance of learning and evolution as observed by mutation rates, partitioned into pre- and post-600,000 time-step periods.

them plants are good. This is a big benefit for survival since the agents learn to eat, leading to energy increases, and eventually offspring. From time to time, action-related mutations occur that cause agents to approach plants instinctively. These changes are favored by natural selection because they avoid the shortcoming of each new agent having to rediscover that plants' goodness means it should approach them. Agents begin to eat at birth and are better able to survive.

The Baldwin Effect shows how inherited characteristics can mimic acquired characteristics in a population using only conventional evolutionary mechanisms. Though support from biologists for the concept been spotty, the phenomenon has been previously demonstrated in a computational evolutionary simulation by Hinton and Nowlan[9,16] (see also Section 5).

To investigate further, we devised hypothetical *fitness models* for various sets of agent genes, based on our sense of world AL. In Figure 12, the two graphs depict the values over time of fitness models for four sets of genes: the plant-evaluation, plant-action, carnivore-evaluation, and carnivore-action. Maximum plant-evaluation fitness implies a positive view of energy and positive responses to plants in all directions. Plant-action fitness is related to the probability that a plant will be approached in every direction. Carnivore-evaluation fitness incorporates a positive attitude toward health with negative attitudes towards carnivores in all directions. Carnivore-action fitness is *inversely* related to the probability that a carnivore will be approached in every direction.

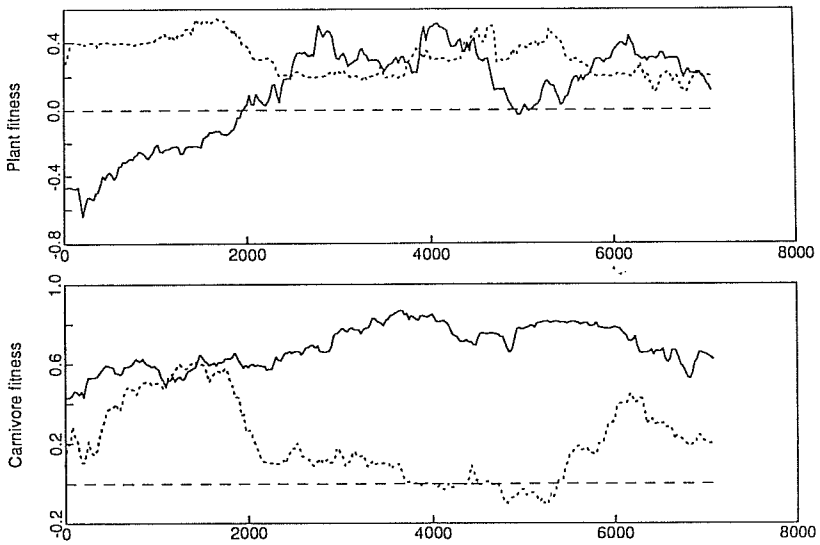

FIGURE 12 Genetic changes over time ($\times 10,000$) relative to hypothetical fitness models for plant-related genes (top) and carnivore-related genes (bottom). In each graph, solid lines represent values of *action*-related genes, and dashed lines represent values of *learning*-related genes.

It must be stressed that these models, though plausible, are only *hypothetical*. Lacking a closed form solution, only AL itself can model fitness exactly. For example, although the handcrafted agents mentioned in Ackley and Littman[4] receive a perfect score in plant-action fitness, their *actual* fitness, overall, is unclear.

Given these models to estimate genetic fitness, we can see in Figure 12 that in the plant domain, the Baldwin Effect is evident. Although plant-evaluation fitness is relatively high throughout the run, plant-action fitness rises steadily to supplant it. What once had to be acquired, was now inherited by around the 3 million time-step mark.

What leverage do these models give us in understanding the population's eventual demise? We believe a crucial destabilizing factor is visible in the plot of carnivore-evaluation fitness. It begins fairly low, rises quickly and levels off, then suddenly plummets and even goes negative for a time. For more than a million time steps, agents actually *liked* the sight of carnivores. It is distinctly possible that this maladaptation contributed to the population's eventual extinction.

How could this have come to be? Why would natural selection have permitted such unfit organisms to proliferate? The answer appears to be that the organisms were not actually unfit even though they possessed this "obvious" flaw. Although

the carnivore-evaluation fitness was quite poor during the latter half of the simulation, the carnivore-action fitness was high and gradually increasing. The well-adapted action network apparently *shielded* the maladapted learning network from the fitness function. With an inborn skill at evading carnivores, the ability to learn the skill is unnecessary.

Our data does not make a strong case for the Baldwin Effect at work in the carnivore-related genes. In fact, based on preliminary analyses of other successful runs, it appears that carnivore-action fitness is so important to survival that if the initially created agents are not at least partially able to evade carnivores instinctively, the entire population dies out quickly. AL carnivores are dangerous beasts; without some innate tendency to avoid them, an agent is likely to die before it can learn to dodge.

Although our data does not clearly demonstrate both the Baldwin Effect and shielding on the same set of genes, it is easy to foresee the possibility of the combination, and the peculiar effect—which we call *goal regression*—that would be a likely consequence. A successful inherited ability acquired via the Baldwin Effect will not shield all learning genes equally—it will preferentially affect the learning genes that were responsible for the Baldwin Effect to begin with! The very goals that were known to be adaptive—since they aided population survival initially—are exactly those that become most shielded and subject to genetic deterioration—since the inherited abilities supplanted the need for those goals.

When natural selection is the only source of feedback, shielding and goal regression are potential hazards wherever the Baldwin Effect is a potential benefit.

5. DISCUSSION

The Baldwin Effect depends upon the stability of both a problem and its solution over evolutionary time. On the one hand, if the solution changes, the genetic acquisition of a specific solution is a liability. On the other hand, if the problem simply vanishes, any added fitness for possessing the solution vanishes with it. The effect can only persist through extended evolutionary time if, somehow, improvements can continually be added to the solution without ever *really* solving the problem. Shielding and goal regression are possible consequences if the Baldwin Effect undermines itself by solving the problem too well.

We can summarize the effects and their relationships this way:

- In an environment that poses a problem for survival, a population arises that survives because it possesses learning ability and an inborn set of goals that happens to be best satisfied when the problem is *anticipated and avoided.*
- *Baldwin Effect:* As generations pass, ways of anticipating and avoiding the problem become incorporated and instinctive.

- A successful inherited ability to avoid a problem—whether a consequence of the Baldwin Effect or not—means that learning to avoid the problem confers little advantage.
- *Shielding:* Genetic information related to learning the ability is less constrained functionally; mutations can accumulate without affecting fitness.
- *Goal regression:* If the inherited ability did arise via the Baldwin Effect, shielding will preferentially affect the original learning ability that the Baldwin Effect relied upon.

In effect, the adaptations selected to avoid a problem tend to flatten the fitness subspace related to learning to solve that problem. Goals can arise whose achievement would actually aggravate the problem, but they can be shielded from a fitness penalty because the inherited ability tends to avoid situations in which the *possibility of achieving* the goals can be discovered.

Given the above descriptions, and scrutinizing Figure 12, a characterization of the significant evolutionary time-scale events in that simulation would include the Baldwin Effect at work in the plant-related genes, and shielding in the carnivore-related genes. Although those seem to be major effects, there are clearly other contributing factors at work in the simulation. Predation as the challenge to survival is one such factor, because the size of the predator population, and thus the severity of the problem, interacts nonlinearly with the ability of the prey to evade. The reduction in the size of the carnivore population, in effect, increases the size of the "flat area" in fitness space, because the base rate of predator-prey interactions is reduced. The effect of shielding is exaggerated, and the agent population risks disastrous "population implosions" when the carnivore population begins to rebound, giving the maladapted learning networks more opportunities for mischief.

Predation can be contrasted with a less reactive problem—such as, say, an ice age—which is largely unaffected by a population's success at keeping warm (ignoring possible long-term issues such as interactions between fire-making and greenhouse gases). In such a case, opportunities for unlearning are hard to avoid completely, so shielding and goal regression tend to incur a fitness penalty more quickly.

It may seem strange that we have claimed such a close coupling between the Baldwin Effect, shielding, and goal regression, given that the simulation of Hinton and Nowlan[16] displays only the first one. In this context, the critical distinction between their approach and ERL is that they have assumed the *a priori* existence of a *criterion of success.* Their organisms are presumed to possess an ability to recognize when the problem has been solved and to therefore stop learning, but they leave open the question of how this ability could be acquired via natural selection. ERL provides an explanation—its evaluation network is an *evolvable criterion of success*—but as we have seen, the price of a mutable evaluation function includes a long-term risk of goal regression.

Goal regression may also give pause to theorists such as Schull[23] who argue that a species as a whole can fruitfully be viewed as an intelligent entity. The Baldwin Effect is a central component of that viewpoint, which likens evolution

to species-level learning, and likens individual organism's learning experiences to species-level "hypothetical thoughts." Such viewpoints are not strictly amenable to proof or refutation, but goal regression raises the possibility that sometimes a species may fruitfully be viewed as a fairly stupid entity.[12]

The architectural split between actions and goals in ERL challenges Lloyd's[19] assertion that "organic selection, to the extent that it produces evolutionary change, *eliminates* phenotypic plasticity and replaces it with the genes that produce the local optimal phenotype" (pg. 79, original emphasis). On the one hand, Lloyd's claim is clearly true of the Hinton and Nowlan[16] simulation. In that case, adaptive sites and inherited sites are drawn from the same pool, so more inherited sites necessarily means fewer adaptive sites. On the other hand, the picture is rather different with ERL. One must carefully separate the *source* of individual plasticity—the learning algorithm and the evaluation network—from any *specific goal* that may be represented in the network. Via the Baldwin Effect, a specific goal may cease to be relevant due to shielding and drift away, but some other goal will necessarily replace it. The evaluation network is still there, and the learning algorithm; In ERL, plasticity is not eliminated; at most it is redirected.

6. CONCLUSIONS

We identified two main interaction effects of learning and evolution in our system. The first, known as the *Baldwin Effect*, made it possible for organisms to use learning to stay alive while waiting for successful behaviors to be incorporated directly into the genetic code. We feel this accounts for the superiority of ERL over the systems we studied that used evolution and learning in isolation.

The second interaction effect we encountered had not previously been studied in a computational context. Here we saw that successful inherited behaviors *shielded* (or reduced functional constraints on) the inherited preferences which control learned behavior. This effect appears to account for the long-term instability of the population presented in Ackley and Littman[4] and Section 4.

In addition, the combination of the Baldwin Effect and shielding can lead to the phenomenon of *goal regression*, in which the specific goals that initially facilitated survival are preferentially eroded.

Functional constraints are powerful tools for inferring the relative impacts of learning and evolution. The richness of artificial life datasets allows statistics and fitness models to be analyzed as detailed functions of evolutionary time—rather than as scattered sample points—revealing the dynamical behavior of such systems.

Although in our simulations shielding and goal regression seemed to be liabilities, it is worth noting that they have potential benefits as well. The shielded learning genes might happen upon goals even better the original ones. For example, although it is beyond the capability of the simple agents we simulated, the potential exists in AL for sophisticated agents with shielded plant-learning genes to

discover agriculture! A rough calculation shows that world AL could easily support several hundred agents continuously if they controlled the plant population and dispersal with selective feeding. Such a development could follow from just the sort of goal—e.g., to walk away from some meals—that could be deadly at the outset.

There are many obstacles to building and analyzing multiple-scale models. Size and duration acquire multiple interpretations, introducing fundamental ambiguity into such basic concepts as equilibrium and stability. Nonetheless, multiple-scale research efforts such as this one offer hope of uniting the sometimes fractious research groups that are separated only by a scale change. Caughley[10] expressed our sentiments well, while pondering how group selection could possibly occur if natural selection operates solely at the individual level:

> "A resolution to this dilemma must wait for population geneticists to grow weary of their pivotal assumption that a population has no dynamics, and for population dynamicists to abandon the belief that a population has no genetics." (pg. 113)

Computers, like microscopes, are instruments of empirical science. Multiple-scale simulation models offer a way of casting light on elusive phenomena that hide in the cracks between levels due to scale-crossing interaction effects. Between evolutionary theory and population biology, group selection may be such a phenomenon. Between cognitive science and neuroscience, the emergence of mind from brain may be another. The power of the computational microscope is growing by leaps and bounds, and we are just beginning to learn how to use it.