Learning and Evolution

Brett Calcott

Introduction

The role of learning in animal evolution is typically restricted its selective advantage, much the same as any other phenotypical trait. Having a longer neck allows you to reach more foliage; having an ability to learn where food is, or has been, allows you to spend less time searching. However, any suggestion that a particular learned habit could be passed on to offspring is tarred with the Lamarckian brush, and promptly sent packing. Humans, of course, are treated differently - we have culture and traditions. Our advanced cognitive abilities give us an alternative inheritance system, over and above that of genes, and the relative rapidity of cultural inheritance means that it has, at least recently, played a huge part in human evolution. A recent book by Avital and Jablonka (Jablonka and Avital 2000) questions this exclusive application of behavioural inheritance to humans and suggests that learning has been a driving force in evolution amongst animals as well.

Now, it seems that in some way this exclusive application *is* warranted, as the depth of culture in human society, and our ability to accumulate knowledge seems far more evident than in any other animal, even our close primates relatives. This difference can be explained once we recognise that there are different methods of transmitting learned behaviour - and these methods range in cognitive requirements from very simplistic to extremely complex. At the complex end of scale, we have the purposeful pedagogic training of offspring, and imitative learning -- where the learner attempts to replicate the actions and the goal of the teacher – a behaviour that seems to be exhibited only in humans (Tomasello 1999). Toward the more simplistic end of the scale, we can simply increase the probability that the behaviour will be transmitted by offspring simply being exposed to the same stimuli as the parent. Avital and Jablonka suggest that even the simplest type of transmission can be an effective driving force in evolution.

The introduction to their book (and the inspiration for this simulation) asks us to imagine a world of creatures they dub "tarbutnicks"; creatures that reproduce, but with no genetic variation. They are all effectively clones of one another, although family and other social relationships are maintained[1]. However, tarbutnicks can learn -- if a parent tarbutnick has acquired some information about the environment, it is more likely that their offspring will acquire this same information. Here is one simple example of how evolution could continue without any genetic modification: Suppose that a single tarbutnick, perhaps by chance, acquires the ability to eat a new type of food. If this new type of food is particularly nutritious, then the tarbutnick may well out-perform its conspecifics. Additionally, this ability will be transmitted with an increased probability to its offspring -- creating a lineage with this ability. Avital and Jablonka continue with this thought experiment to show how it could cumulatively affect developmental results, and mating preference -- resulting in further changes. All this without genetic differentiation in the individuals.

Although Avital and Jablonka suggest many ways in which learning can affect the evolutionary process, I am going to restrict this simulation to the inheritance of a single learned behaviour, that of food preference. There is good evidence that a parent's food choice can affect the food choice of their offspring. For example, young mice prefer food that was evident in the milk they were suckled on. Genetic influence plays little part here as the food preference persists even when the mouse pups are exchanged with different mothers. Food preference may also be affected by liquids that pass across the mother’s placenta, and by the practice of coprophagy (Jablonka and Avital 2000, pp109-114). One particular example on which I have, in part, based this simulation is the case of the Israeli black rat. This species of

[1] Of course, this is somewhat unrealistic; if the entire population was clonal these relationships would break down. However, that misses the point -- they want to show how evolutionary change can proceed without genetic variation. Locking it down in their thought experiment is meant to achieve *only* this.

rat has recently enlarged its habitat into pine forests by extending its behavioral repertoire to include the ability to open pinecones. It appears that only black rat pups, rather than adults can learn this behaviour, and they do so by practicing on partially stripped pinecones provided by their mother. Again, this behaviour has little to do with any genetic change, as cross-fostering of pups shows that the behaviour appears in the pups raised by the foster-mother who had this particular ability, rather than the genetic mother that had the ability.

The simulation that I propose then, is one that depends only on a very limited cognitive ability. As in the case of the black rat, rather than the offspring requiring the ability to do some complex imitative behaviour, the situation requires that the parent provides what I'll call an *aided context*. The essence of this is that a parent lowers the initial experience threshold required to efficiently extract calories from food source by partially completing the task for their offspring. To make the model at least somewhat realistic will have to build in some decision-making procedures for how and when to explore, reproduce, and eat. It turns out that all of this requires a fairly large chunk of underlying architecture to get it working.

The Simulation

The simulation consists of two-dimensional surface on which different types of food are distributed, and a number of the agents forage and reproduce. The surface is rectangular, with top and bottom edges, and left and right edges wrapping; producing a toroidal surface. Each cycle, or time step, of the simulation consists of several phases. One of these phases permits each agent a time slice in which it can interact with the environment. The other phases consist of constructing a local environment for each agent, and dealing with such housekeeping issues as births, deaths, and collisions between agents. The system is not pre-emptive – each agents' processing return controls when it has completely finished its tasks; thus the implementation of the agent must ensure that actions can be broken down into simple components that execute in a reasonable time span. The interaction phase randomly shuffles the order in which the agents are called to prevent any agent from gaining preferential access to resources.

The behaviour of the agents in the system can be broken down into two sections. Firstly, the adoption of a high level control system; and secondly, the details involving the adaptive foraging strategy.

POSH Behaviour Control System

It is important that we select the correct level of detail at which to encode the behaviour of the agent. Selecting a very low level encoding, such as a neural net, would lead to an excessive time in getting even the simplest functionality operating. Alternatively, deliberately encoding the actions at too higher level would result in a system that lacks adaptive flexibility. For the simulation, I chose to base my implementation on a flexible control system developed by Joanna Bryson which she calls parallel-rooted, ordered, slip-stack, hierarchical reactive plans, or POSH for short (Bryson 2001). This system has proved to be effective in a similar, though more constrained, two-dimensional agent-based world (Bryson 2000). In addition, the scheme proposed by Bryson is specifically designed to be relatively easy to engineer by hand.

The control system divides the behaviour up into **triggers**, **actions**, and **competences** (Bryson's competences are an instance of what is more generally referred to as "basic reactive plans"). A trigger, or releasing action, is simply a Boolean function such as "am I hungry?" or "can I see some food?". Actions are the simple execution of modular behaviour such as "turn left", or "bite food". Triggers and actions can be combined into lists, and a competence is an ordered list of trigger(s)-action(s) pairs. When a particular competence is active, the first trigger in the list that fires causes the associated action to be executed. If the first trigger in the list fires, then the competence has reached its goal. A simple example of Bryson's follows; it assumes a world consisting of stacks of colored blocks; the object is to "hold a blue block".

"holding block" & "block blue" ➔ *goal*
"holding block" ➔ "drop held", "lose fixation"
"fixated on blue" ➔ "grasp top of stack"
"blue in scene" ➔ "fixate blue"

As Bryson points out, this type of reactive plan takes advantage of the current state it is in when activated and, additionally, copes with any failures of the actions in the competence (such as dropping a block). I have omitted one feature up until now -- rather than the pair containing list of actions, it may in fact point to another competence. This recurrent containment is what allows us to construct the hierarchy of behaviour. It's important to note that any of the actions may be autonomous -- once initiated a may continue functioning in parallel with the control system. For example, in this particular simulation once an action initiates movement of the agent, the agent continues to move in parallel with other behaviors that are initiated by the control system until another behaviour directs it to stop.

To complete the control system one other feature is introduced -- a **drive**. A drive-collection, or list of drives, forms the base of the behaviour hierarchy. It resembles a competence in that it consists of an ordered list of trigger-action pairs, however it differs in the way that is used. At each time-step we iterate over the drive collection stopping at the first trigger that fires. The corresponding competence or action is initiated. Here is where the notion of what Bryson calls a slip-stack comes into play. In the case of competence the search for the first firing trigger will result in another corresponding action or competence. This action or competence replaces the original competence at the root of that particular drive. When the drive is activated in a subsequent time-step, it is this competence that is immediately activated. The drive reverts to its original root item only when the competence reaches its goal or one of the actions fails. Notice that this lack of a "regular" stack means that competences can contain recursive references. This allows a particular competence further down the hierarchy to be active in successive cycles without it having to reassess *why* it is active. Apart from the obvious implementation feature of increased speed, this ability also allows an agent to concentrate on completing a task without constantly reassessing whether it is the right task to do. This confers a realistic type of ‘attention span’ and avoids behaviour such as continously reselecting better looking food sources as you head toward the current target food source – with the result of never actually eating anything!

The POSH architecture supplied by Bryson satisfies the requirements for the simulation on two different levels. Firstly, the actions of a complex agent require the ability to switch their attention from one task to another and effectively deal with multiple constraints. Bryson's control system does this by partitioning autonomous behaviour between multiple drives with each drive maintaining its own state. Secondly, because the system requires that we break down complex behaviour into component actions it fits elegantly within the non-preemptive simulation model I have chosen. All actions execute in a small amount of time and the system maintains its own internal state of the current execution status.

Two final notes on this subject. Firstly, I don't make full use of the system’s ability to deal with multiple constraints. Originally, I had thought to include both mating and predator avoidance along with foraging behaviour. As it turns out the foraging behaviour in itself was complex enough. Secondly, I should comment on a slight difference between Bryson's implementation and my own. Bryson includes ability for triggers to habituate by limiting the maximum number of failures of the associated actions. This prevents an agent from insistently banging its head against a brick wall when an action repeatedly fails. Rather than implicitly encode this habituation into the control system, I have relegated the habituation to the implementation of each trigger. This allows for more flexible approach, as the habituation is maintained outside the control system, and can therefore be under the control of complex interactions between other triggers and actions.

Adaptive Foraging

Although the POSH architecture gives us a framework for decomposing the agent's behaviour we still require a model for how the agent memorises and adapts its foraging behaviour. There is a wealth of material describing optimal foraging, and comparing it with actual animal behaviour (Krebs and Davies 1987; Roughgarden 1998). Typically, selecting food to maximise calorie intake requires meeting multiple constraints depending on the relative search time, frequency, processing time, and calorie content of the different food sources. Once we know these values, some simple calculations allow us select the best choice for the agent. For example, is it better to attempt to consume a food source that has been located, or is it preferable to continue to search for another food source that is larger, easier to process, or contains more calories? For most part, it seems that animals are very good at making the correct choice, as the behaviour seems to approximate what we can mathematically show is the best choice. However, these calculations are *descriptive* of the agent's behaviour. It is doubtful that the actual agent reaches its decisions by anything like these kinds of calculations. As Roughgarden puts it:

> "How can an animal know how to forage optimally? Of course, a very intelligent animal could read this book and compute for itself what the optimal strategy is. It's more likely thought, that an animal find out what's based for itself through progressive discovery, and if so, we need to investigate how an animal can learn its optimal foraging strategy is it goes along." (Roughgarden 1998, p35)

Roughgarden provides only a partial solution to the problem however. His calculations for food choice are adaptive only for the changing frequency of the food types available. The assumption is that the animal already knows the processing time and caloric content of the food types. Although an animal may be able to assess the caloric content by smell, it's not clear how they could know in advance the processing time for a particular food type. It is also likely that this processing time may change as the agent gains more experience. Moreover, none of this takes into account what might induce an agent to try a novel food source. One explanation is that animals are more likely to try a novel food source, or be less selective about maximising calorie intake when made a closer to starvation (Krebs and Davies 1987, p65). I have attempted to incorporate all of these factors into a relatively simple adaptive foraging strategy.

Each agent keeps track of two energy values that are used to make decisions about food sources. Firstly, a value that represents their current energy resource -- when this value reaches zero the agent is dead. Secondly, a value representing the expected energy that they will receive per time-step. This value is a decaying average of the total energy lost or gained in a single time-step.

```
UPDATE ENERGY
begin time-step
    record energy-resource
    gain or loss of energy (moving, eating food, reproducing etc.)
    calculate Äenergy-resource
    current-expected-energy = (energy-decay * expected-energy)
        + (1-energy-decay * Äenergy-resource)
end time-step
```

The energy-decay parameter takes a value close to one (e.g. I am using 0.99). The closer to value is to one, the more the agent will value past experience. The expected-energy value automatically takes into account searching time, processing time, and caloric value of the recently eaten food types.

As an agent wanders through the two-dimensional world, it selects food that is nearby, and assesses whether it wishes to eat it. It does this by comparing its current expected energy with the expected energy it would gain from consuming this food source. However, to do this, it must make an estimate of the expected energy it will gain from this particular food type, and this estimate can only be gained from experience. For each food type, the agent maintains an expected energy to be gained per time-step whilst attempting to consume it. The expected

energy associated with the food type is updated in the same fashion as the agent's expected energy -- using a decay parameter. As in the previous case, modelling expected energy incorporates both the calorie content and the processing time of the food type. The decision to attempt to eat a particular food source proceeds as follows:

```
ASSESS FOOD SOURCE
total-expected-reward = amount-of-food * caloric-content // from "smell"
expected-processing-time = total-expected-reward / expected-energy-for-food-type
total-expected-reward -= expected-processing-time * cost-of-eating
if (not-touching-food)
    expected-processing-time += time-to-get-to-food
    total-expected-reward -= time-to-get-to-food * cost-of-moving
choose-food-expected-energy = total-expected-reward / expected-processing-time
if (choose-food-expected-energy > current-expected-energy)
    accept-food
else
    reject-food
```

Once an agent decides to eat a particular food source, it moves towards it until it is touching the food source and then attempts to eat it, periodically reassessing the status of the food source. (Note that touching a positively assessed food source corresponds to the "salivation" state in the POSH system). At each time step biting the particular food source selected may or may not result in gaining some food energy; however, biting always incurs a cost. Whether or not the bite is successful depends on the food type and the experience of the agent with this particular food type. An agent's experience is simply the number of bites they have attempted with this food type. This is taken into account using four parameters that control the food types response to experience. Two of these values are a minimum and maximum probability of success, corresponding to no experience and infinite experience. The other two are parameters to a sigmoidal function controlling the threshold and gradient. This allows for food types that require and all-or-nothing insight, or those that gradually respond to experience.

```
BITE FOOD (all values are for a particular food type)
experience-factor = 1/(1+exp((threshold-param – experience) / gradient-param))
P(successful-bite) = min-prob + ((max-prob – min-prob) * experience-factor)
if (random-choice(P(successful-bite)))
    energy-gained = food-type-calories
else
    energy-gained = 0
```

Notice that the results of this bite are used to update the expected energy from this food type. So as experience with this food type increases, the agent's assessment of its expected energy will also increase.

An agent must also periodically reassess the food that it is trying to eat. A food source is limited; so after a number of successful bites it will be depleted. The frequency with which it reassesses the food source is controlled by a simple fixed probability.

Notice that the strategy is already quite adaptive. If an agent spends a long time searching for food it is likely that its expected energy will decrease below zero (all of the recent Äexpected-energy will be made up of only the cost of moving, and no positive food benefit), and this makes it more likely that it works it a food type it supplies least caloric value. However, one major problem still exists. The "perfect" calculations make no allowance for experimenting with new food types, and perhaps gaining enough experience with them for it to be profitable to frequently eat them. An accurate assessment of a food type can only be gained by interacting with that food type. The basic decision making mechanism for a food source relies on calculating the expected energy difference between accepting and rejecting the food. Currently, we always reject the food if the expected energy is less. A simple way of building in some degree of explorative behaviour is to probabilistically accept a lower expected energy. Additionally, we can make this probability higher if the agent is hungrier (has less energy-resource). Our agent's acceptance of food types now resembles simulated

annealing with the temperature controlled by hunger. We can extend the food-assessing algorithm in the following manner:

```
ACCEPT LOWER EXPECTED ENERGY
energy-difference = choose-food-expected-energy - current-expected-energy // always negative!
P(accept-lower) = exp(energy-difference * suspicious-of-novelty-factor * energy-resource)
if (random-choice(P(accept-lower)))
    accept-food
else
    reject-food
```

Note: there are a couple of other options I would like to try here. Firstly, the current energy-resource could dynamically update the energy decay parameter. This has the realistic appeal that a hungry agent pays less attention to past experience, and more to recent events. Secondly, it seems that hunger should affect the probability of reassessing the food source as well. I would guess that a hungry agent might be more persistent, and less likely to reassess its options.

Nurturing Behaviour

Up until now I have only discussed the foraging details. How does reproduction take place, and how is it that particular food preferences are inherited? Importantly, how can inheritance happen *without* an agent's offspring simply appropriating their parent's food type experience in a Lamarckian manner?

Reproduction in the simulation currently takes place asexually. Once the parent's energy-resource reaches a predetermined threshold, another agent is spawned at the same position as them. Half of the parent's energy is redistributed to the child. The child receives no information from the parent – it is born *tabula rasa*. However, the two do form a special relationship during the nurturing period, a predetermined number of time steps that is set by a parameter for each tarbutnick. During this nurture period the parent is responsible for locating food and directing the child to it. Food choice during the nurture period is totally controlled by the parent. However, as the child bites the food it gains experience with that food, and begins to approximate the expected energy from this food type. If the child's energy resource falls below that of the parent, then energy from any successful bites that the parent makes are passed along to the child. The child gains no experience from this, however it does update its expected energy for this food type. This type of behaviour corresponds to a common occurrence in animals -- that of aiding their offspring again initial experience in processing a particular food type. Three things to note here. Firstly, this type of behaviour by the parent will cause the initial estimates of expected energy by the child to be over optimistic. Secondly, as mentioned in the introduction, this type of behaviour requires very little cognitive sophistication. No complex mimicking is taking place. The parent is merely making its own food choice, and providing a consistent aided context in which the child calculates its own estimates of the food type's utility. Lastly, having a child is costly, as the resources that you locate are partitioned between the parent and child and, in addition, when the child is still inexperienced the parent pays the cost of getting the food but receives no benefit.

When the parent currently has no positively assessed food source, it continues to wander looking for other food sources. Under the circumstances, the child simply attempts to stay close to its mother by limiting the distance between the two -- a simple following behaviour results.

Other Parameters

As can be seen, there are a number of parameters that are fixed, and are currently tuned by hand. A few others are worth mentioning.

- Food types have a linear re-growth rate.

- The way in which and agent moves whilst searching for food is controlled by wandering algorithm that provides smooth transitions between tight circular movements and a relatively straight lines. This parameter can be adjusted, but currently isn't. The source for this algorithm is (Reynolds 1999)
- Whilst wandering, the simulation architecture provides the agent with a list of local food sources within the particular radius. It chooses only one of these to assess per time step. The algorithm for doing this begins with the closest food source, and accepts it for assessment with some fixed probability. If it is not accepted, then it proceeds to the next closest and applies the same probability. Additionally, any rejected foods, both those that have been eaten and those that were rejected outright, are placed in a short-term memory for a fixed number of time steps and are always rejected. This prevents an agent from dwelling on any one food source.
- As well as dying from lack of energy, each tarbutnick has a fixed probability of dying each time-step from 'other causes'. This gives tarbutnicks a limited lifetime – something that becomes important when we start evolving parameters.

Results

The adoption of the POSH control system and the complexities of the foraging model were both implemented with the one thing in mind: the creation of an organism capable of adaptive foraging. My first task is to show that this has been achieved. I will populate a world with two types of food -- the first has a relatively high yield that can be achieved with minimal experience; the second, a low yield with a higher degree of experience required. Notably, the second food type yields absolutely nothing at first. Figure 1 shows the probability that each of these food types will yield energy upon being bitten depending on the amount of experience. I will refer to these as the **easy** and **hard** food from now on.

Figure 1. Differing yield profiles of 2 food types

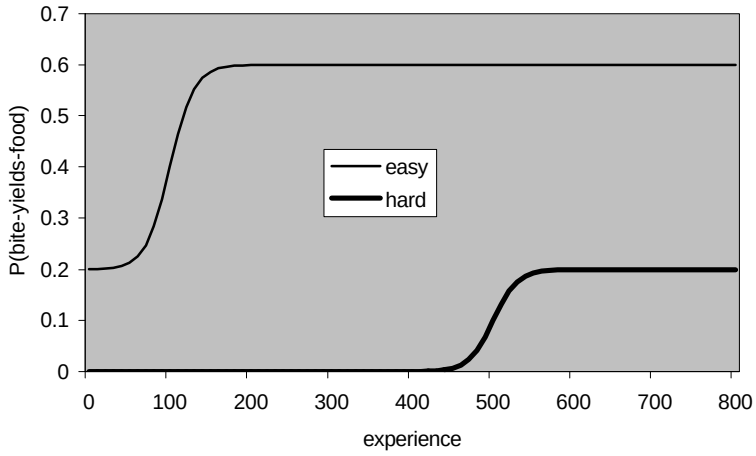
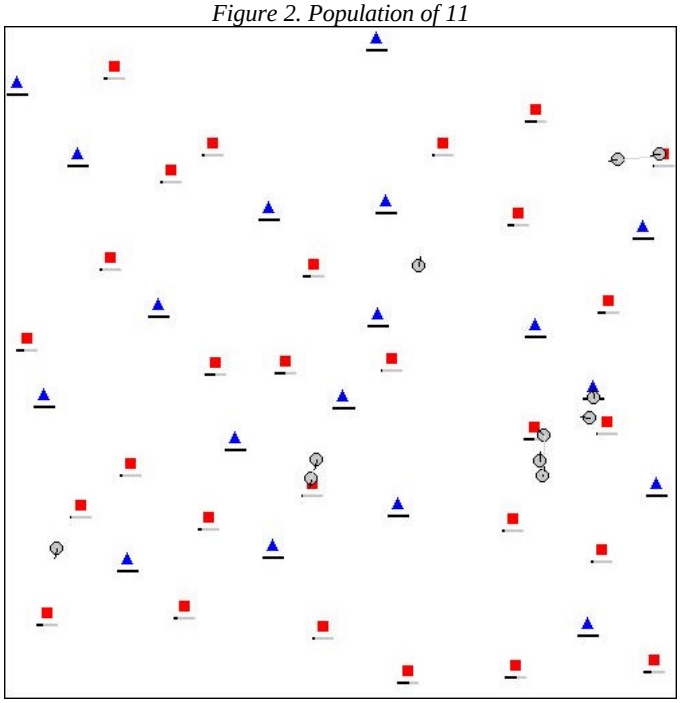


Figure 2 shows the world after a small amount of time. The hard food is represented by blue triangles, the easy food by red rectangles. The small bar underneath the food represents much food remains at that particular food position. When a tarbutnick successfully bites the food the bar reduces in length, and when left alone the bar eventually increases to the maximum number of energy parcels allowed for that food type. Initially I placed a single tarbutnick in the world, the others have appeared through reproduction. Examining the bars beneath different food types we find that, at this stage, the hard food (blue triangles) has not been successfully eaten -- all of the bars show them to be still completely full. On the other hand, many of the patches of easy food (red squares) show that they have been successfully

consumed. One of the tarbutnicks can be seen attempting to eat the hard food. Although they periodically do this, they quickly give up, as there is plenty of easy food to be had.

Figure 2. Population of 11



In Figure 3 the population has now increased to 15 and, as can be seen, some of the hard food has begun to be eaten. This happens because as the population increases the easy food becomes less and less available. As the tarbutnicks becomes hungrier, they are more likely to attempt to eat the less appealing food. After a significant amount of unsuccessful attempts have been made a tarbutnick finally gains enough experience to begin profiting from the hard food. Figure 4, whether population has reached 18, finally shows a situation where the majority of the population are consuming both the hard and the easy foods.

Figure 3. Population of 15

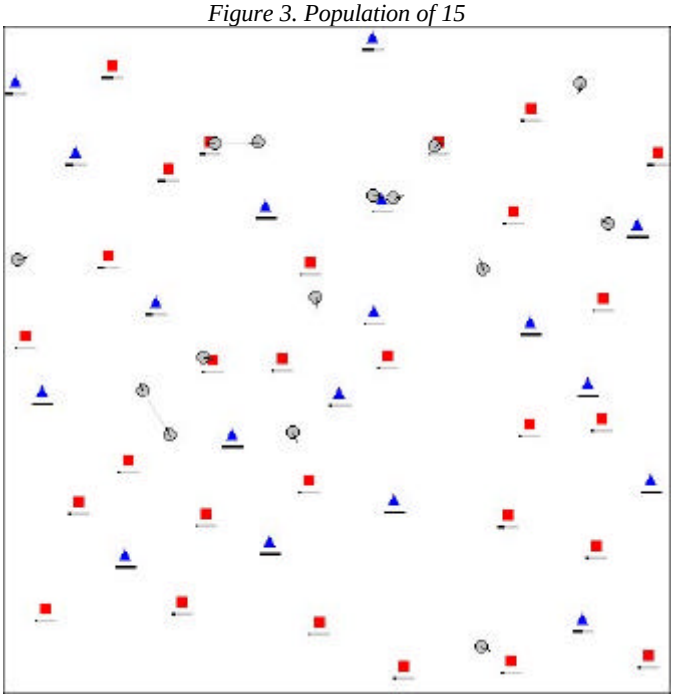
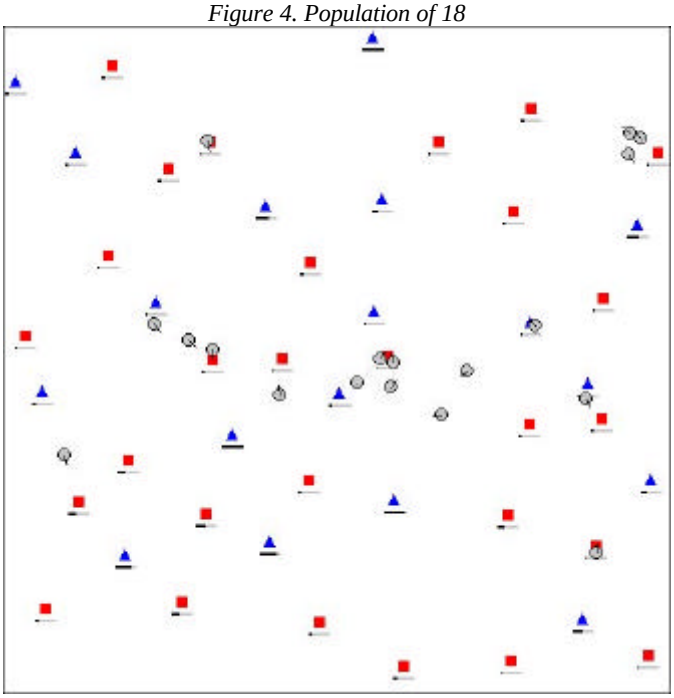


Figure 4. Population of 18



We have established that our tarbutnick behaviour at least meets the minimum requirements for being adaptive to different food types. How does an aided context, in the form of a nurturing period for offspring, affect the evolution of a population? Figure 5 shows the increase in population in two separate simulations; one with that no nurturing period, and the other with the nurture period of 3000 time steps. These simulations consisted of a world containing only sparsely distributed easy food types and a single tarbutnick. It is clear from the graph that nurturing your offspring places a burden on the parent; the population increase of the simulation with a nurturing period has a significantly lower gradient. This burden can be attributed to two factors. Firstly, when a tarbutnick is nurturing its offspring it cannot reproduce again. In a situation where food is plentiful, the tarbutnick may well possess enough energy to reproduce again before it has weaned its current child. Secondly, whilst a tarbutnick is nurturing it's offspring it effectively has to forage for two. This slows the rate at which she can increase its own energy and thus prepare itself to reproduce again.

Both population increases appear to show the kind of sigmoidal increase that is explicitly modelled in population dynamics. Even though the simulation with a nurturing period has a slower population increase both simulations are constrained by the same carrying capacity of the food and the world and therefore both of them eventually stabilise into a random pattern of fluctuations around this value. Again, this carrying capacity is not explicitly modelled but, like the sigmoidal increase, is an emergent property of the low-level interactions of the parameters controlling the food and the agents.

Figure 5. Population with only easy-food..

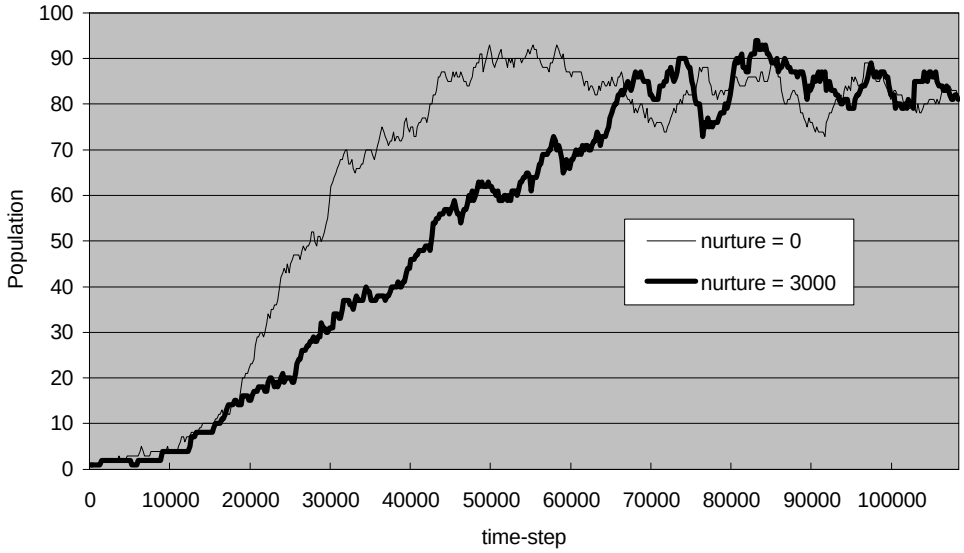
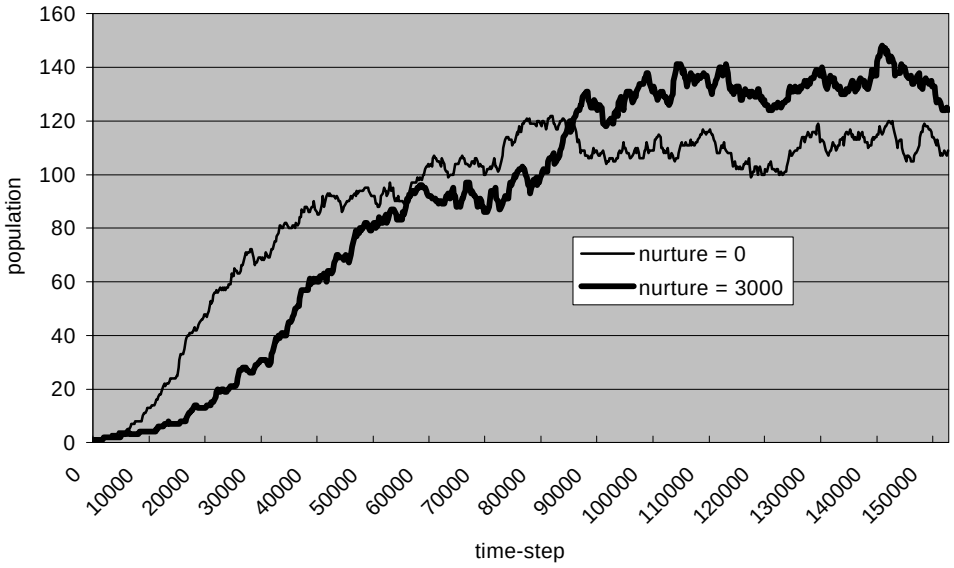


Figure 6 shows two simulations that differ from those in Figure 5 in only one way: I have added some widely dispersed hard food types to the world. Note that the hard food types are in addition to the already existing easy food types, so we should expect the carrying capacity of this world to be higher than the previous one.

The most obvious difference is that the simulation with a nurturing period, although having a slower population increase like the previous simulation, it eventually levels out at a significantly higher value than the simulation with no nurturing period. It seems that although tarbutnicks with no nurturing period show a slight increase in population (in comparison to the previous world) which reflects the increased carrying capacity, they are not able to take full advantage of the newly introduced food type. The problem with the tarbutnicks that have no nurturing period is that each tarbutnick must relearn from scratch an ability to deal with the hard food. In the case of the tarbutnicks with a nurturing period, as the parents are already successfully eating and hard food whilst the nurture the offspring, their offspring started their own foraging with an initial boost experience.

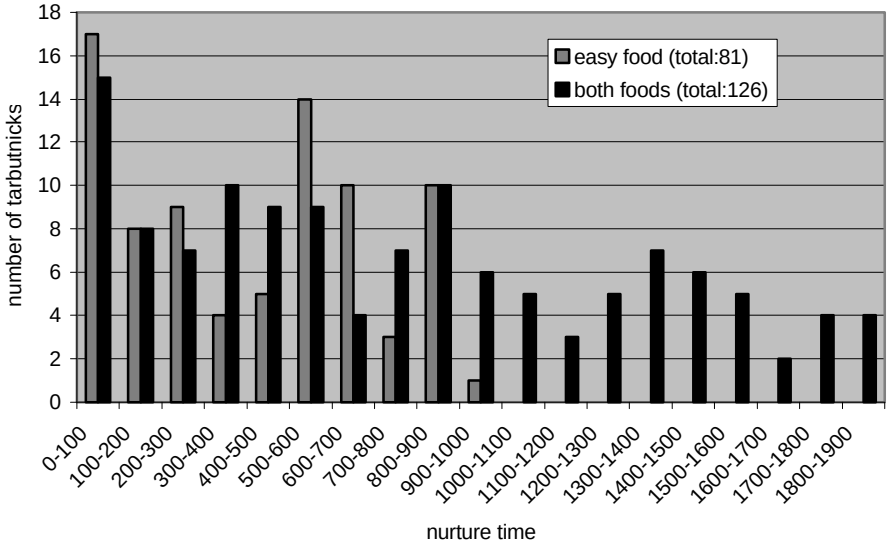
Notice that the simulation with a nurture period appears to undergo a second vaguely sigmoidal increase in population after they reach a first peak roughly equivalent to the carrying capacity of the easy food (by comparison with the previous simulation). The second increase is due to the differential success of tarbutnicks whose parents have learnt how to harvest the hard food -- the slow spread through the population of an acquired ability. In other words, the increased carrying capacity of the population is a direct effect of the tarbutnicks learning an ability from its parent. Note that the spread is purely due to replacement -- tarbutnicks cannot learn by watching one other (i.e. inheritance, in this case, is purely vertical).

Figure 6. population growth with both food types.



We have seen that although nurturing places a burden on the parent, in cases where the environment provides challenges that require some experience to overcome, and the parent can aid in lowering the threshold for success, nurturing can be advantageous. I was interested to see what effect allowing the nurture period to 'genetically' evolve would have. Using the same two worlds as in previous simulations (one with the easy food, the other with both), I started to new simulations off with a single tarbutnick having a nurture period of 1000 time steps. Each time a tarbutnick reproduced, the child inherited its parents nurture period with the addition of a random amount drawn from a normal distribution with an s.d. of 200. Figure 7 shows the resulting distribution of nurture period values after an approximate 150,000 time steps in each simulation. As expected, the total population of the simulation with both foods is higher. The results are far from conclusive largely, I think, due to the speed at which I tried to evolve the system – a s.d. of 200 is a little large, but I wanted some results without waiting a week. However, it does appear that and the case of environment containing only easy food, the nurture period *only* decreased. When the more difficult food was available, and subsequently learning was useful, the simulation shows at least a significant proportion of the surviving tarbutnick lineages have increased the nurture period.

Figure 7. Evolution of Nurturing parameter



Summary

I began by explaining a cognitively unsophisticated way in which the learned abilities of the parent may influence their offspring -- a very simple case of Lamarckian inheritance. I proceeded to construct a model that demonstrated the observed characteristics of typical animal foraging and included this simplistic ability for inheritance. The final model demonstrates that, at least under some circumstances, the learned abilities of the parent being transferred to the offspring can have a significant effect on the population's use of resources. I am not sure whether this is the enough for them to be considered significant and long-term evolutionary terms, but it is clear that we should not discount them.

There are ways that I think the model could be enhanced to strengthen the case for the significance of learning in evolution. As the case of the black rat, it would be interesting to see if the ability to inherit learned behaviour was significant and enable the successful invasion of new habitats. It would seem from the current evidence showing an increased ability to take advantage of more difficult food sources that this would be possible. I have attempted some initial experimentation with the current model, however it was difficult to draw any conclusions mainly due to the fact that I had no good instrumentation to track what is happening. A way of combining lineages of the tarbutnicks with their individual foraging zones would provide a clear picture of the spatial evolution of a population. This would require a fair bit of work! Additionally, I would be interested in seeing if distinct food habitats could enable food type specialisation in the lineages of tarbutnicks -- a kind of speciation via learning. This kind of thing could be enhanced with the introduction of sexual reproduction and mate preference, as Avital and Jablonka show that mate selection in some cases may be affected by food preference. Lastly, or advantage could be taken of the POSH system by introducing more complexity to the system including predator avoidance, the seeking of shelter, territory defence and, as I have mentioned sexual reproduction. The addition of a simplistic learning system feeding back amongst these multiple processes may have some interesting effects.

Bibliography

Bryson, J. (2000). Hierarchy and Sequence vs. Full Parallelism in Action Selection, (http://www.ai.mit.edu/people/joanna/publications.html).

Bryson, J. (2001). Intelligence by Design, (Phd.), (http://www.ai.mit.edu/people/joanna/publications.html).

Jablonka, E. and E. Avital (2000). Animal Traditions, Cambridge Unversity Press.

Krebs, J. R. and N. B. Davies (1987). An Introduction To Behavioural Ecology, Blackwell Scientific Publications.

Reynolds, C. W. (1999). Steering Behaviors for Autonomous Characters, (http://www.red.com/cwr/steer/gdc99/index.html).

Roughgarden (1998). Primer of Ecological Theory, Prentice-Hall.

Tomasello, M. (1999). The Cultural Origins of Human Cognition, Harvard University Press.

Appendix 1. POSH Control System for the Tarbutnick Agents

The following is the C++ code that constructs the POSH control system. Tracing back from the definition of the drives you can get some idea of the order in which tarbutnick controls and executes its behaviour.

```
ADD_COMPETENCE
  (locate_food,
  conditional(TRIGGER(watching_food), ACTION(assess_smell)) +
  conditional(ACTION(wander) + ACTION(sniff_for_food))
  );

ADD_COMPETENCE
  (goto_food,
  conditional(TRIGGER(touching_food)) +
  conditional(TRIGGER(facing_food), ACTION(move_toward_food)) +
  conditional(ACTION(turn_to_food))
  );

ADD_COMPETENCE
  (forage,
  conditional(TRIGGER(touching_food) & TRIGGER(food_smells_good), ACTION(salivate)) +
  conditional(TRIGGER(food_smells_good), COMPETENCE(goto_food)) +
  conditional(COMPETENCE(locate_food))
  );

ADD_COMPETENCE
  (stay_close_to_mum,
  conditional(!TRIGGER(mum_is_far)) +
  conditional(TRIGGER(facing_mum), ACTION(move_toward_mum)) +
  conditional(ACTION(turn_to_mum))
  );

ADD_COMPETENCE
  (do_what_mum_says,
  conditional(TRIGGER(touching_food), ACTION(eat_food)) +
  conditional(TRIGGER(watching_food), COMPETENCE(goto_food)) +
  conditional(TRIGGER(mum_has_food), ACTION(take_mums_food)) +
  conditional(COMPETENCE(stay_close_to_mum))
  );

m_drives =
  DRIVE(juvenile, TRIGGER(still_a_baby), COMPETENCE(do_what_mum_says)) +
  DRIVE(reproduction, TRIGGER(can_reproduce), ACTION(make_baby)) +
  DRIVE(maternal, TRIGGER(need_to_feed_baby), ACTION(share_food_with_baby)) +
  DRIVE(salivation, TRIGGER(salivating), ACTION(eat_food)) +
  DRIVE(hunger, TRIGGER(hungry), COMPETENCE(forage));
```