Estimating Future Reward in Reinforcement Learning Animats using Associative Learning

George D. Konidaris **Gillian M. Hayes**
Institute of Perception, Action and Behaviour
School of Informatics, University of Edinburgh
James Clerk Maxwell Building, King's Buildings
Mayfield Road, Edinburgh EH9 3JZ
Scotland, UK

Abstract

We introduce a model of animat reinforcement learning where an associative learning element is placed between the animat's reinforcement learning component and its internal reinforcement functions. This element forms an impression of the sensory stimuli present near a goal and uses it to make an initial estimate of the value of newly discovered state-action pairs in tasks where reward is necessarily delayed.

We then describe the implementation of Peaches 'n Cream, a simulated robot that implements one version of the model where behaviour-based reinforcement learning is used in a puck foraging task. The results suggest that once given an initial simple task to learn from, the resulting associations significantly speed up learning in a later, more complex task, and provide further evidence that other learning methods may be used in conjunction with reinforcement learning to make it feasible for situated agents.

1. Introduction

Reinforcement learning provides an intuitively appealing animat learning model because it is well suited to the problems animats face (Sutton, 1990), and because it provides a principled way to build animats that function according to a set of internal drives.

However, when such drives are expressed as reward functions, they often lead to situations where a long sequence of unrewarded actions culminates in a single reward obtained by the satisfaction of one of the drives. Learning when reward is thus delayed is difficult both in terms of finding optimal solutions and in terms of finding any solution at all, because the reward signal is not sufficiently informative to guide the agent to a solution. This creates a problem for animats in domains where the agent does not have sufficient time to fully explore the problem space in search of a solution (e.g., where the animat's energy levels are low and it needs to find a battery) and is therefore (at least initially) concerned with quickly finding any solution at all.

One way to alleviate this problem is to reduce the state space by making it as task-relevant as possible (e.g., Mahadevan and Connell (1992)). However, some learning problems are intrinsically difficult, and require a more informative reward signal to guide the initial search for a first solution, after which rapid propagation of reward through the state space (either using traces or some form of offline learning) can guide the agent in optimising it.

In this paper we introduce the use of an associative learning element that acts between the animat's internal drives and its reinforcement learning component. This element learns to associate environmental stimuli with estimated reward when performing an initial, relatively easy task. In later tasks, when a new state is encountered, its value can then be estimated based on its similarity to prior reward states, and the animat has effectively learned a heuristic that it can use to guide search. Animats built with this model are therefore first exposed to a task where they can learn the correct associations, and thereafter may be exposed to more difficult tasks where the associative element can provide more information than a single reinforcement signal alone. The model is thus aimed at increasing the autonomy of animats that use reinforcement learning by freeing the designer from having to specify more explicit reinforcement functions.

We then present the implementation of a control system for a simulated robot that performs reinforcement learning in a simulated arena using one implementation of the associative element described here. By comparing the results obtained by Peaches 'n Cream in a complex environment after having learned in a more simple one to those obtained without the use of an associative learning mechanism, we show that such a mechanism is capable of speeding up learning in complex tasks.

2. Background

Exploration in reinforcement learning is usually performed using ϵ-greedy methods, where the animat makes a random move some fraction (ϵ) of the time. ϵ-greedy methods guarantee asymptotic coverage of the state space, and are usually combined with the assignment of optimistic initial values to states to encourage exploration. The agent thus starts with no knowledge of the state space, and is drawn to the unexplored parts of the space by their higher values (Sutton and Barto, 1998), making random choices every now and then. The optimistic estimates are gradually replaced as the animat learns the actual value of each state.

However, Konidaris and Hayes (2004) make the claim that ϵ-greedy methods (along with all other methods that emphasise asymptotic coverage) are not appropriate for situated agents that must act as best they can given the knowledge, time, and resources they have available. A hungry animat should not make random decisions every now and then, in the hope of stumbling across food. Rather, it should find the food it knows where to find, and explore when it isn't hungry. We therefore separate exploration in situated reinforcement learning into two separate concerns: first, how to find some initial solution path, and second, how to cover the state space in order to find better solutions than the first. This paper deals with the former problem; Konidaris and Hayes (2004) describe an exploration drive that deals with the latter.

When an animat is trying to find any solution at all, it is essentially performing something akin to a search through the state space where it may only keep one state "open" at a time. The use of optimistic initial values (the most common solution method to this problem) is thus essentially equivalent to a blind or depth-first search because all states estimations are equivalently optimistic.

Matarić (1994) suggests the use of "progress indicators" or some other form of more informative reinforcement signals to help guide search. The major drawback to this is that it requires extra engineering effort, and does not map well to the analogy of reinforcement functions as internal drives. The balance to be struck here is between a priori knowledge making learning easier and principled design making learning harder. However, there is a middle ground: the animat may be allowed to learn its own richer reward signal from experience without having to have it specified in advance.

3. Estimating Initial Values using Associative Learning

We propose that instead of having a very informative but difficult to engineer reinforcement signal, animats should be able to learn which sensory signals indicate the nearby presence of a reward state from experience. This sensory signature can then be used as a heuristic measure, with states that have similar signatures to states that have in the past contained a reward being given higher exploration values than states that do not.

An agent using this mechanism would start off without any knowledge, and therefore perform as badly as agents without any associations, but would learn them over its lifetime in perhaps many different related problem instances. Such a system would be an instance of layered learning (Stone and Veloso, 2000), where associative learning is used to make reinforcement learning more efficient.

A good example here is the use of smell by some agents to find food. It is not necessarily true that this knowledge is built in – humans did not evolve specifically to like the smell of coffee, for example, but have learned to associate it with the act of drinking coffee. Once this has been learned, office workers are more likely to expect to find a coffee dispensing machine in corridors that smell of coffee than in those that do not. This knowledge does not have to be obtained through inference; simple association will do.

Learning would thus proceed as follows. First, the agent is placed in an environment that is either relatively simple, or where it has sufficient time to properly explore. Whenever the agent encounters a goal state, it makes a note of the sensory stimuli that are especially strong there and thus learns to associate the presence of those stimuli with reward. Later, when encountering new states, the relevant stimuli levels are used to determine how similar the new state is to the expected goal state characteristics, and this is in turn used as a heuristic estimate of the value of the state.

There are three specific issues to be dealt with here: determining which stimuli are relevant, determining some form of relative weighting for them (since some are more important then others), and then using this weighting to estimate initial reward. The exact mechanism used to perform the associative learning will in general depend on the characteristics of the relevant stimuli and the properties of the task. In the following section, we describe an example task and outline the implementation of an associative learning mechanism that is appropriate to it.

4. Puck Foraging in a Simulated Arena

In order to test the model proposed above, we developed a working system where a plain reinforcement learning animat could be compared against one that made use of associative learning to estimate rewards.

4.1 The Task

The experimental task chosen was a simulated version (in Webots 4 (Webots, 2003)) of the puck-foraging task used to validate the behaviour-based reinforcement learning model introduced in Konidaris and Hayes (2004).

A mobile robot was placed in an arena constructed from orthogonal white walls, and was driven by three internal needs: the need to find food (represented by a green cylindrical puck), the need to explore, and the need to return to its home area. These needs were governed by a circadian cycle that started the robot off as hungry, then switched to exploration mode when food was found, and then switched to homing mode at the end of a predetermined time (the cycle length). If the robot failed to find food by the end of the cycle length, it proceeded directly to the homing phase without exploring. Upon returning home, the robot started the cycle again, and ran for ten cycles in total.

The robot was required to perform its task by following the walls of the arena. At each wall (either a left or a right wall – we omit corridors here for simplicity) the robot could at each end go straight or turn against the wall after it has ended, giving four possible actions for each wall. In addition, walls that were connected to other orthogonal walls to form internal corners did not permit turns. The robot thus also had to discover which turns were not permitted and avoid them.

The robot's aim was to maximize the reward corresponding to its currently active drive. For the homing and feeding phases, it received a reward of 1200 points for returning home and finding food, respectively, and lost 1 point every half second it failed to. In the exploration phase, the robot received a reward of between -1200 and 1200 points for each landmark-action pair, depending on the ratio of the number of times that pair had been taken previously against the average number of pair experiences, with rarer transitions receiving higher rewards.

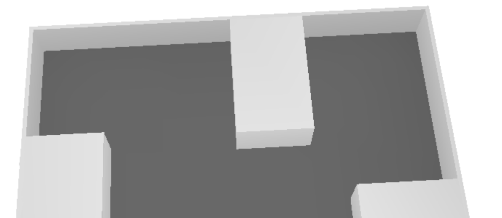

Figure 1: The First Arena

The two models were first tested in a relatively simple environment, shown in Figure 1, which aimed to provide a fairly simple training case for the associative learner. Peaches 'n Cream is in the bottom left corner, which serves as its home area, and the puck can be seen in the bottom right corner. Although this arena does not look imposing, it was nearly as complex (at 20 distinct walls) as the most complex arena used in Konidaris and Hayes (2004). However, in this work we allowed the robot fifteen minutes of simulated time per cycle as opposed to three minutes of real time[1].

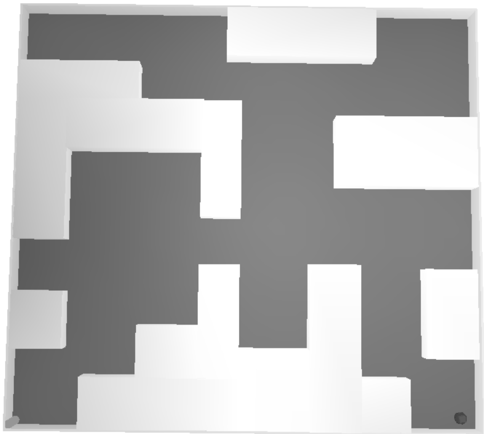

Figure 2: The Second Arena

The models were then tested in a much more difficult environment, shown in Figure 2, with 44 distinct walls. This environment was intended to provide a difficult problem instance where the associations learned in the first environment could guide the robot's exploration. In addition, the cycle length was reduced to ten simulated minutes, which significantly increased the difficulty of the problem since the most direct route from the robot's home area to the puck took approximately eight simulated minutes. Peaches 'n Cream started out in the bottom right corner of the second arena (within the box that serves as its home area) with the food puck in the bottom left.

4.2 The Robot

The robot used in the tests, Peaches 'n Cream, was a simulated standard khepera robot with a colour vision turret. Peaches used a ported version of the reinforcement learning control software written for Dangerous Beans, and detailed in Konidaris (2003) and Konidaris and Hayes (2004). We therefore provide only a high level outline of it here, along with a description of the associative learning modules added to it.

The architecture of the control system can be viewed as having three distinct control layers. The first layer

[1]Note that the simulated time and real time measures are not directly comparable. In our experience the model we used in simulation was approximately half as fast as the real robot described in Konidaris and Hayes (2004).

performs obstacle avoidance and wall-following using a standard set of behaviors. The second performs landmark detection and distributed map building. The distributed topological map is a graph where a *place behavior* or node is allocated to each distinct landmark the animat has encountered so far, and with connections between place nodes indicating that the animat can move directly between them with some simple (recorded) sequence of behaviors. Each place behavior is able to detect when it should be active, and the place behavior that is active following it, as well as the decisions made to move there, and form connections to them. Another behavior is responsible for merging place behaviors that represent the same state and creating new ones when a wall is encountered for which no suitable behavior exists. This layer is equivalent to the distributed map-building system developed by Matarić and Brooks (1990). An example map visualisation taken from the first arena is shown in Figure 3.

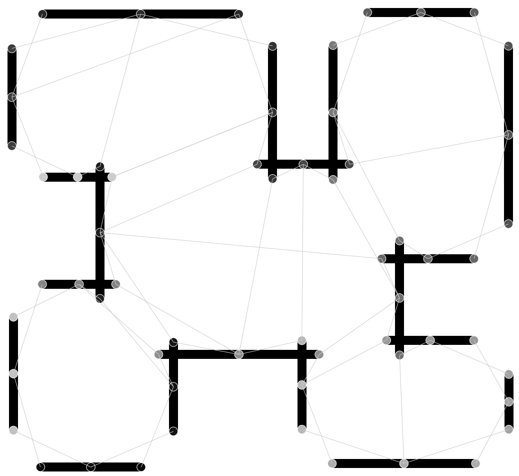

Figure 3: A Sample Topological Map of the First Arena

The final control layer performs reinforcement learning over the topological map, with nodes in the map acting as states and edges as actions in the reinforcement learning problem. Each place behaviour maintains a set of action values, and updates them using the TD(0) update rule and a full backup (Sutton and Barto, 1998). Updates occur all the time, at all behaviors, but the currently active place behavior is able to request the action with the highest learned action value for execution. We refer to the resulting structure as a *reinforcement learning complex* rather than a reinforcement value table because of its dynamic structure. Note that there are actually three distinct complexes (one for each drive) embedded in the same map.

In the standard reinforcement learning model, actions that have not yet been executed are assigned an estimated value of 300 for the homing and reward drives, and an exact value of 1200 for the exploration drive. This optimistic initial value encourages the agent to try these transitions, but it does so uniformly, resulting in an unguided search of the state space until a reward point has been found.

The version of Peaches 'n Cream with the associative learning model was given a set of five extra sensors, each returning a real number between 0 and 1, and each subject to 10% uniformly distributed noise. The first sensor returned a randomly generated value, while the third and fifth returned a constant and cycling value, respectively. The second and fourth sensors returned a measure that represented the straight line distance to the homing and puck points, respectively. The sensor readings were scaled so that they read as 1 at the goal point (the middle of the homing area or the puck), and 0 at a distance of about 700 units away (with the first and second arenas measuring 500 and 1100 units in both directions, respectively). They thus acted as smell sensors with a signal emanating from the goal but decreasing with distance, although they could be interpreted as any abstract distance heuristic.

The associative learner collected samples of each sensor reading every half second, and every time a puck or homing reward was obtained. These samples were used to estimate the mean sensor reading and variance for each sensor in puck reward states, home reward states, and over the arena as a whole, with the intention of using sensors that show a markedly different distribution near goal states as indicators. In order to determine which sensors were markedly different, a t-test was performed, measuring the difference between the reward state distributions and the overall arena distribution. Using the resulting t_i value for each sensor s_i, a weight w_i was calculated as follows:

$$w_i = \begin{cases} 0 & \text{if } t_i < 2.5 \\ \frac{t_i - 2.5}{t_{tot}} & \text{otherwise} \end{cases}$$

where t_{tot} was the sum of all of the t_i values of at least 2.5. The weights thus sum to 1, and weight each factor in proportion to its difference from the overall arena distribution. The cutoff value of $t = 2.5$ was chosen as a cutoff that gives a low probability of mistaken significance ($p < 0.05$) for a relatively small sample size. The similarity factor k for a particular state was then estimated as:

$$k = \sum_i w_i d(s_i, e_i)$$

where s_i is the reading of the ith sensor and e_i is its expected reward mean. The function $d(s_i, e_i)$ simply returns the inverse of the fraction of e_i that s_i is off by – for example, $d(s_i, e_i)$ returns 0.75 if e_i was 0.8 and s_i was either 0.6 or 1.0. The initial transition value was then estimated by the reinforcement learning system as:

$$E_R = 300 + 500k$$

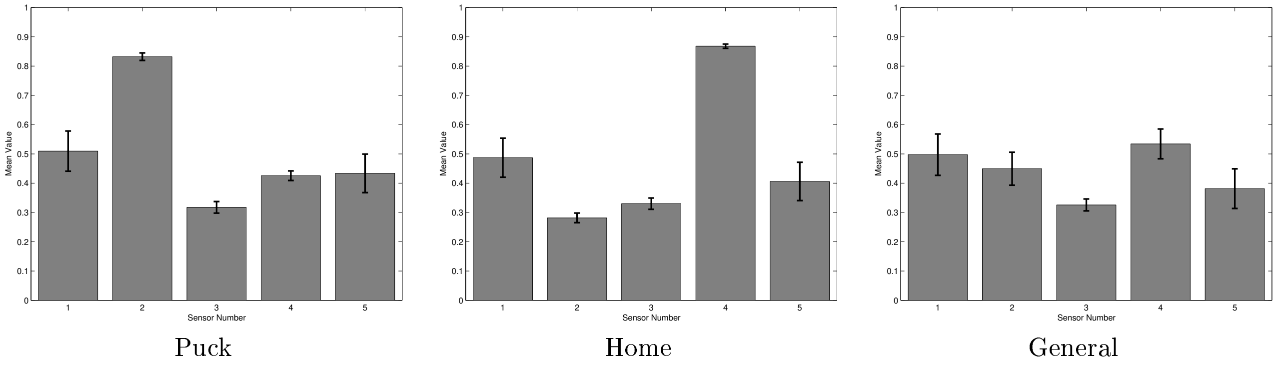

Figure 4: Averaged Sensor Means Learned from the First Arena

This guarantees every transition an optimistic initial value of at least 300 (because k is always positive) but allows some (those that are similar to goal states) to receive estimates that are proportionally higher, although never quite as high as an actual reward state[2].

5. Results

In this section we present the results obtained in the task described above. We first cover the associations learned by the associative learning version of Peaches 'n Cream in the first arena, and then compare the performance of associative robots using these learned values to that obtained by reinforcement learning robots without association.

5.1 The First Arena: Associations

The means of the sensor readings obtained by ten robots at the puck, the homing area, and the first arena generally are shown in Figure 4. In the puck goal case the sensor corresponding to the puck distance sensor (sensor 2) was consistently very high, and similarly in the homing goal case the homing distance sensor was consistently very high. The other sensors do not differ significantly from their values in the rest of the arena.

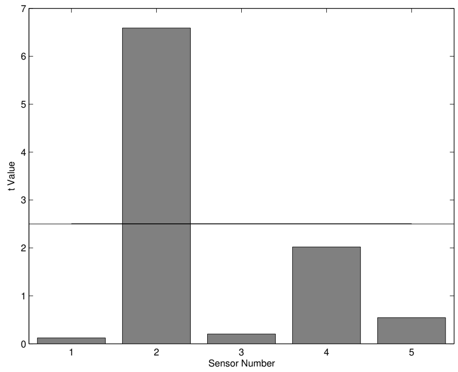

Figure 5: Puck t Values

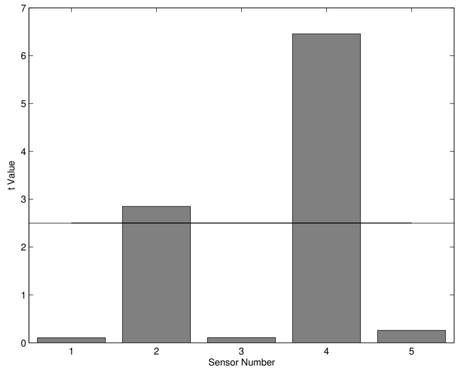

Figure 6: Home t Values

Figure 5 shows the mean t-values for the puck goal. Here, only the puck distance sensor has a sufficiently high t value to be above the cutoff, although in some of the individual robots the homing distance sensor with a low mean value was just above the cutoff of 2.5. Figure 6 shows the t values for the homing case, where both the homing distance sensor and the puck distance sensor are above the threshold and thus considered significant, although again in some individual robots the puck distance sensor fell just below the significance threshold.

Both of these extra sensor significances make sense since the goals were fixed relative to each other and it is therefore reasonable for the robots to associate fixed (in this case low) values from these sensors with reaching the goal. Note that the puck goal association with the homing distance sensor is correctly lower than that of the homing goal association with the puck distance sensor because the two places where a puck reward could be obtained were further apart than the two places where the robot could enter the home arena, and were therefore likely to vary more widely.

Because each weight is proportional to the sensor's t value's height above the threshold, the alternate sen-

[2]Note that all of the constants presented in this section were determined by trial and error, guided by the desire to have a puck reward just too far for the robot to reach in time to have a value of zero.

sors were given low weightings by the associative learner and thus did not adversely affect performance. We can therefore conclude that the associative learning element was able to correctly and reliably learn the associative relationship between the two distance sensors and their goals, but that in general several learning experiences may be required so that the animat can avoid learning coincidental sensory relationships.

5.2 The Second Arena: Performance

Figure 7 shows the average puck reward against cycle number for both models over ten test runs in Arena 2. The associative learning robots were given the means and variances outlined in the previous section as a priori knowledge.

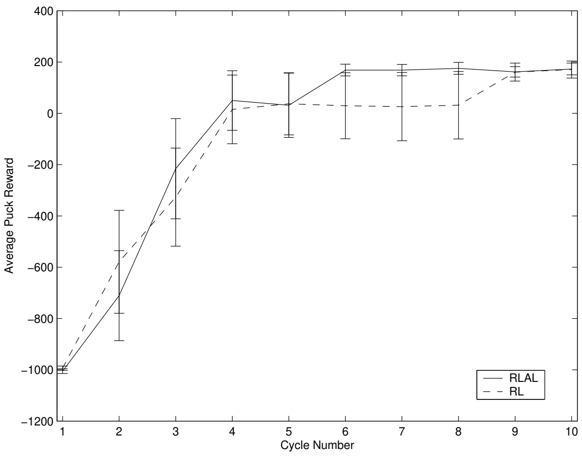

Figure 7: Average Puck Reward over Time

In Figure 7 the associative learning animats appear to be able to find the puck initially slightly faster and more consistently than the standard reinforcement learning agents. However, the difference here is hard to discern because often the standard reinforcement learning robots would explore deep into the maze, and when returning home would go through such long paths that they stumbled across the puck on the way. This is the way that most of the pucks through the runs were discovered, and it skews the puck reward data.

This is illustrated by Figures 8 and 9, which show two representative robot paths, with the position of the robot starting off black and fading over time. Figure 8 shows the route an associative learning robot took back home once it had reached the end of a cycle length, but before it had discovered a robust way home. Note that it starts out in the bottom corner near the puck and proceeds to slowly but definitely move toward the home area in the bottom right corner.

In Figure 9, however, it is clear that the robot is essentially making random choices without repetitions until it arrives back at the home arena, thus covering virtually

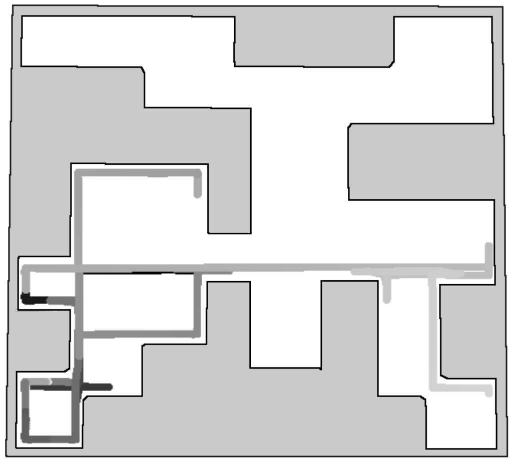

Figure 8: An Associative Robot Returning Home

the entire arena and discovering the puck. In particular, it comes very close to returning home fairly early on, but then turns around and goes the other way, before eventually wandering back home.

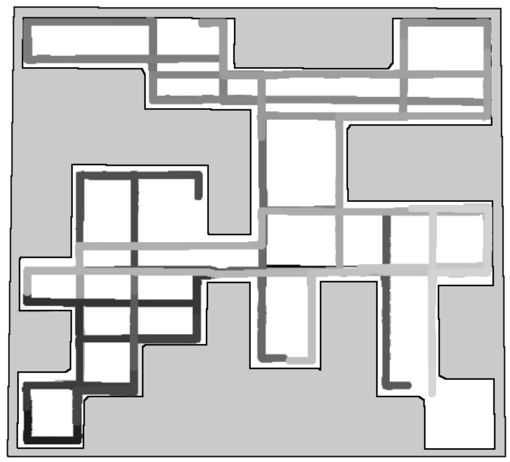

Figure 9: A Non-Associative Robot Returning Home

Figure 10, showing the average home reward against cycle number over the same ten runs, shows the difference between the two models clearly. The associative learners are able to return home much more quickly than the standard reinforcement learners, at least initially. The associative learners have higher average reward values and smaller error bars up until the 4th cycle, when the two graphs level out. Thus the associative element was capable of speeding up learning initially even though eventually the systematic but unguided standard reinforcement learning agent caught up.

Although the differences between the mean home reward values for each cycle as shown in Figure 10 are fairly large, they cannot be used directly to determine

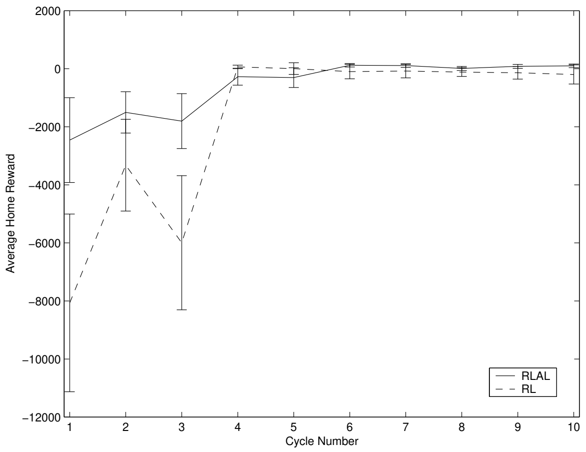

Figure 10: Average Homing Reward over Time

the significance of the results because the structure of the task means that the animat's performance in one cycle is heavily biased by its performance in the cycles before it. Instead, we use the average total homing rewards over time for comparison, which are shown in Figure 11. A t-test indicates that this difference is statistically significant ($p < 0.05$), and this, combined with the significant qualitative differences in search behavior shown in Figures 8 and 9, demonstrates clearly that the associative learning model performs better than the standard reinforcement learning model.

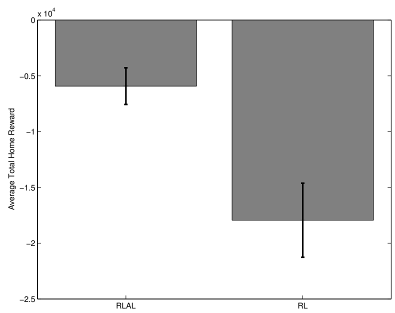

Figure 11: Average Total Home Reward

6. Discussion

6.1 Significance

The results and model presented in this paper are significant for two reasons. The first is that they suggest that reinforcement learning animats may be able to use their own experiences to learn their own, more informative, initial reinforcement learning estimates without having to have them built in. This has the potential to increase the autonomy of animats built using reinforcement learning because it removes the need for highly informative but hand-crafted reinforcement signals.

This is superior to simply adding the signal to the reinforcement learning state space for two reasons. First, even an impoverished representation of such a signal would significantly increase the size of the state space – the addition of only one bit of extra information would double it – and thereby impede the learning capabilities the agent. Second, such a signal simply does not belong in the state space, because it is not a degree of freedom in the state space – rather, its presence at certain states is highly correlated with a high return at those states. As such, many combinations of state and signal simply could never be experienced by the agent, and the ones that could would carry no more information as states than those without the signal. In other words, the signal adds no information to the state space, but it instead adds information to the reward landscape embedded in the state space. Thus, the system preserves the advantages of using reinforcement learning and provides a principled way to build value driven agents without losing the ability to use extra sensory cues to improve the agent's performance.

The second significant aspect of the research presented here is that it provides evidence for the claim made in Konidaris and Hayes (2004) that additional learning models may be added to an already layered reinforcement learning system in order to speed learning. Although there are several ways this could be done, the existence of one such model and the results given here prove that it is feasible in at least one interesting case, and strengthen the argument that layered learning can bring substantial benefits to situated agents.

6.2 Limitations

There are two primary limitations with the work presented here. The first is that it reports the results of the use of the model given here for only one task (puck foraging and homing in a simulated arena) and in only one trial instance (the second arena). In particular, a task more suited to highlighting the difference between an associative and standard reinforcement learning agent should provide clearer and more decisive results. There are also several aspects of the work presented here that could be expanded upon or improved. For example, the arena used does not contain a local maximum "trap" for the associative learning agent. Such a trap would likely be completely explored by the agent before it moved on, thus slowing it down. This problem should not occur in real smell sensors because they work through air flow rather than distance, but in general no heuristic method will be without flaws.

Another potentially useful aspect of the model that has not been tested here is the notion that it should work with a discrete sensor or one that is not strictly monotonic with respect to distance or gradient based, provided the sensor is relatively consistent and has some unusual value near the goal state.

However, although we cannot therefore claim that in general this methodology will always work, these results do show that it is capable of speeding learning in a fairly difficult problem and that the approach is promising.

The second primary limitation is that the task was carried out in a simulator, rather than on a real robot. In this case, the simulator allowed for the use of virtual sensors for association, and rapid experiments in larger arenas than would have been feasible had the arenas required physical construction. We feel that this justifies its use given the fact that the reinforcement learning model and task have been previously validated on a real robot (Konidaris and Hayes, 2004), and because smell sensors similar to those that we require already exist (e.g., Duckett *et al.* (2001)).

6.3 Implications

The results given here imply that informative and carefully structured reinforcement signals may not be necessary when designing animats to handle complex tasks, provided those animats are allowed to go through a training stage first. This is loosely analogous to the process of shaping (Dorigo and Colombetti, 1998).

One of the negative implications of this is that if such an animat first encounters a poorly or even maliciously constructed environment, it may learn incorrect associations which could hinder its ability to solve later tasks. This may not be too serious since the animat may be able to correct its associations over time, but in some cases it may perform badly in subsequent tasks.

The use of this model also provides further evidence for the suggestion that the form and function of representation is tied to its use in learning models, and that the layering of learning models may create complex representational structures. In the model given here, the associative weights are meaningless outside of the associative learner, and similarly, the associative learner's role is difficult to explain outside of the context of the reinforcement learning complex.

7. Conclusion

This paper has introduced a model of animat reinforcement learning that includes the use of an associative learning element. The associative element learns to estimate the initial value of a state based on its similarity to the sensory characteristics experienced at previous goal states, and thereby provides a heuristic which guides the animat's search for an initial solution. The results of an experimental scenario outlined here show that this model is capable of speeding up reinforcement learning in a complex task, once the animat has been exposed to a simpler training task. These results go some way toward relieving animat designers of the difficult task of handcrafted informative reinforcement signals for problems in which reward is necessarily delayed.

Acknowledgments

We would like to thank the Institute for Perception, Action and Behaviour at the University of Edinburgh for the use of its resources, and two anonymous referees for their useful comments. George Konidaris is employed under the HYDRA project (EU grant IST 2001 33060).

References

Dorigo, M. and Colombetti, M. (1998). *Robot Shaping: An Experiment in Behavior Engineering*. MIT Press/Bradford Books.

Duckett, T., Axelsson, M., and Saffiotti, A. (2001). Learning to locate an odour source with a mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2001)*, Seoul, Korea.

Konidaris, G. (2003). *Behaviour-Based Reinforcement Learning*. Master's thesis, School of Informatics, University of Edinburgh.

Konidaris, G. and Hayes, G. (2004). Behavior-Based Reinforcement Learning. Submitted to *Adaptive Behavior*.

Mahadevan, S. and Connell, J. (1992). Automatic programming of behavior-based robots using reinforcement learning. *Artificial Intelligence*, **55**(2–3).

Matarić, M. (1994). Reward functions for accelerated learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 181–189.

Matarić, M. and Brooks, R. (1990). Learning a distributed map representation based on navigation behaviors. In R. Brooks, editor, *Cambrian Intelligence: the Early History of the New AI*. The MIT Press, Cambridge, Massachusetts.

Stone, P. and Veloso, M. (2000). Layered learning. In *Proceedings of the 11th European Conference on Machine Learning*, pages 369–381, Barcelona, Spain. Springer, Berlin.

Sutton, R. (1990). Reinforcement learning architectures for animats. In J. Meyer and S. Wilson, editors, *From Animals to Animats: Proceedings of the International Conference on Simulation of Adaptive Behavior*. MIT Press.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Webots (2003). http://www.cyberbotics.com. Commercial Mobile Robot Simulation Software.