Learning in a State of Confusion: Perceptual Aliasing in Grid World Navigation

Paul A. Crook and Gillian Hayes
Institute of Perception, Action and Behaviour,
School of Informatics, University of Edinburgh,
5 Forrest Hill, Edinburgh. EH1 2QL, UK
{paulc, gmh}@dai.ed.ac.uk

Abstract

Due to the unavoidable fact that a robot's sensors will be limited in some manner, it is entirely possible that it can find itself unable to distinguish between differing states of the world. This confounding of states, also referred to as perceptual aliasing, has serious effects on the ability of reinforcement learning algorithms to learn stable policies. Using simple grid world navigation problems we demonstrate experimentally these effects. Although 1-step backup reinforcement learning algorithms performed surprisingly better than expected, our results confirm that algorithms using eligibility traces should be preferred.

1. Introduction

Consider a robot learning to navigate its way to a goal point from anywhere in a building. Whatever the design of the robot it can only be equipped with a finite number of sensors and will have limited computational resources with which to interpret this sensory information. Due to these limitations there is always a chance that multiple states of the world, for example two T-junctions or two long corridors in the building, will be indistinguishable to the robot. This problem was identified in active vision systems by Whitehead and Ballard (1991) who coined the phrase *perceptual aliasing*. Although with the current pace of technological advance it is always possible to augment both sensory information and the processing, it would be better to have a range of techniques that the robot can use to deal with these situations. In addition, perceptual aliasing if controlled correctly could form a useful tool. If the mapping between the external world and the agent's internal state is chosen correctly state distinctions that are irrelevant to a task could be ignored, reducing the state space that has to be explored.

Perceptual aliasing is especially problematic when using reinforcement learning algorithms. Reinforcement learning algorithms learn to associate rewards and actions with specific states. Perceptual aliasing, which results in the confounding of the true state of the world, therefore, makes it difficult if not impossible for algorithms to learn stable policies.

Systems that contain perceptual aliasing are examples of *partially observable* environments. Work looking at partially observable environments has shown that reinforcement learning, when augmented with memory (Lanzi, 2000) or the ability to create models of its world (Chrisman, 1992; McCallum, 1993), can solve tasks which contain perceptually aliased states. Our long term goal is to test whether the use of active perception can provide an effective alternative to these two approaches. However, at this stage we wish to better understand the problems caused by perceptual aliasing. To study the fundamentals of the problem we consider simulated agents with fixed perception moving around deterministic grid worlds, such as Sutton's Grid World (figure 1). Depending on the sensory input we allow the agent, it faces similar problems to those that could be encountered by a real mobile robot.

This paper presents results of applying various reinforcement learning algorithms, that are commonly used in robotics, to two grid world navigation problems. These two problems can be rendered partially observable by selecting the observations that make up the agent's state. The aim of these experiments is to test the hypotheses that:

- 1-step reinforcement learning algorithms are not able to learn policies which are both stable and optimal, when the task involves perceptually aliased states;
- Reinforcement algorithms that use eligibility traces can however learn *optimal memoryless*[1] and stable policies for the same task.

.

We confirm results observed by Loch and Singh (1998) that in partially observable environments, reinforcement learning algorithms which use eligibility traces are

[1] see section 2. for an explanation of this term

In Proceedings of Towards Intelligent Mobile Robots (TIMR 2003)
4th British Conference on (Mobile) Robotics, UWE, Bristol 2003

preferable to those that use 1-step backup. Furthermore, by running multiple trials we gather some useful insights on the performance of the algorithms tested.

2. Background

Whitehead and Ballard (1991) considered reinforcement learning of active perceptual systems, specifically active vision systems. To simulate the problems involved they considered a blocks world problem where a robotic arm has to uncover and grasp a green block from among randomly sorted piles of blocks. Their system could select two blocks to be the focus of its attention and the characteristics of these two blocks formed the state of the world as seen by the learning algorithm. The reinforcement learning algorithm had to learn to coordinate which objects were the focus of attention as well as the actions of the robotic arm. Whitehead and Ballard (1991) found that 1-step backup Q-learning failed to learn the optimal policy, only performing slightly better than selecting actions at random. This failure, they concluded, was due to the inability of Q-learning (or any reinforcement learning algorithm using 1-step backup) to learn stable policies in the presence of perceptual aliasing, the perceptual aliasing in their case being caused by the design of the active perception system.

Littman (1994) considered learning state-action policies without memory in partially observable environments, *i.e.* environments containing perceptual aliasing. He introduced the useful concepts of *satisficing* and *optimal memoryless policies*. A memoryless policy returns an action based solely on the current sensation. Standard reinforcement learning algorithms, such as SARSA and Q-learning (Sutton and Barto, 1998, p146,p149), work on exactly this basis. A policy is satisficing if independent of its initial state an agent following this policy is guaranteed to reach the goal. The performance of a policy is measured using the *total steps* that the agent takes to reach the goal from all possible initial states. An optimal policy is one that achieves the minimum total steps to goal. Therefore, an optimal memoryless policy is a policy that achieves the minimum number of total steps which can be achieved by a memoryless policy.

Using hill climbing and branch and bound techniques Littman (1994) showed that it is possible to find optimal memoryless policies for various grid world navigation problems including the variation on Sutton's grid world[2] presented here. Loch and Singh (1998) then showed that reinforcement learning using eligibility traces could also find optimal memoryless policies in grid world navigation problems.

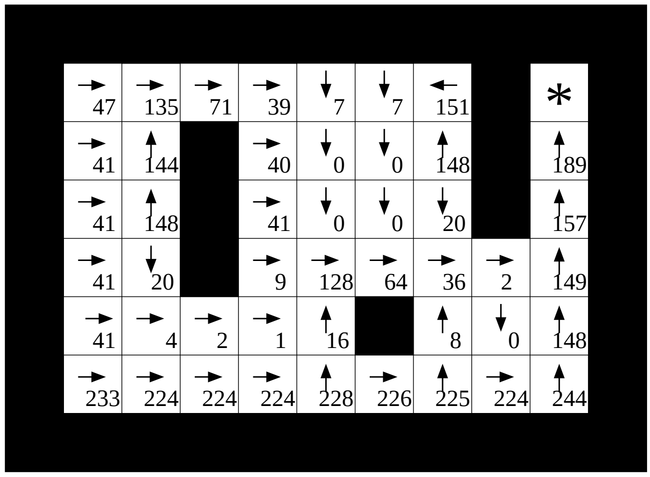

Figure 1: Sutton's grid world. Values indicate observations obtained by an agent with fixed perception sampling the eight surrounding squares (8 Adjacent Squares Agent). Arrows show an example optimal memoryless policy (from Loch and Singh (1998)). Filled black squares are obstacles or walls, and the goal is indicated by an asterisk (*).

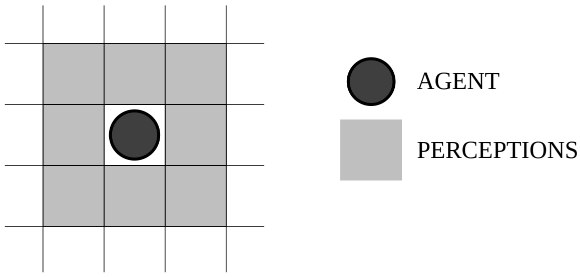

Figure 2: 8 Adjacent Squares Agent — fixed perception sampling the eight surrounding squares.

2.1 *Effects of Perceptual Aliasing*

Perceptual aliasing causes two distinct effects labelled by Whitehead (1992, (chp.5)) as local and global impairment.

Local impairment — an agent that is unable to distinguish between several states of the world will sometimes select actions that are inconsistent with the true underlying state. An example can be seen in figure 1. The states in this figure are labelled to indicate the observations obtained by an agent who can only observe the eight squares adjacent to itself, such as the agent illustrated in figure 2. Such an agent believes it is in state 148 in three distinct locations. These are: four squares directly below the goal; near the middle of the obstacle which is to the left of the goal; and near the middle of the obstacle on the left hand side of the board. In one

[2] The original problem was presented in Sutton (1990)

of these three states the optimal action is to move north (the state directly below the goal), in the second case the optimal action is to move south (the state just to the left of the goal), in the third state it does not matter if the agent moves north or south. As can be seen in figure 1, the agent, unable to distinguish between these three locations, decides it is best served by moving north. Although appropriate for two of the three occurrences of 148, it is not optimal for the state just to the left of the goal. It is because of such situations that optimal state-action policies learnt in the absence of memory, *i.e.* the optimal memoryless policy, are arbitrarily worse than the optimal policy that can be achieved in the absence of perceptual aliasing (Singh et al., 1994).

Global Impairment — given the bucket-brigade update employed by 1-step backup reinforcement learning algorithms, inaccurate estimates of state values that occur with perceptually aliased states can lead to errors in the state values of non-perceptually aliased states. This problem is best illustrated by considering figure 3. This world is deliberately designed such that an agent who can only observe the eight squares adjacent to itself cannot distinguish between states 2 and 5. It is able to uniquely identify all of the remaining states. In all states the optimal action is to move right. The agent when updating state values does not regard states 2 and 5 as separate states, thus it stores only one state value to represent both states and their updates are averaged. This averaging results in state 2 having a value greater than its true value. If an agent is in state 3 it might mistakenly select the action move left, towards state 2, believing state 2 to be nearer the goal than state 4. Updating state 3 on the basis of the state value of state 2 then propagates this error potentially causing other states, such as state 4, to select the action move left and further propagating the inconsistent state values. These errors in state values can end up affecting the whole of the agent's policy.

3. Experiments

We conduct our experiments using two grid world navigation problems:

(i) Sutton's Grid World;

(ii) A simple 1-D example devised by Whitehead (1992, pp73–78) to illustrate the problems perceptual aliasing causes to 1-step Q learning.

And two types of agent:

(i) An agent whose state representation is its location in the grid world given in Cartesian coordinates, the *Absolute Position Agent*.

(ii) An agent whose state representation is formed by observing the 8 squares adjacent to its current location, the *8 Adjacent Squares Agent*, see figure 2.

The importance of these two agents is not in the detail of what they can observe but that the Absolute Position Agent has a unique state representation for every location in either of the two grid worlds, while the latter, the 8 Adjacent Squares Agent, has the same state representation for multiple states in each of the two worlds.

3.1 Sutton's Grid World

Sutton's grid world is shown in figure 1. It consists of a 9×6 grid containing various obstacles and a goal in the top right hand corner (indicated by an asterisk). An agent in this world can choose between four physical actions; move north, south, east and west. State transitions are deterministic and each action moves it one square in the appropriate direction. If an agent tries to move towards an obstacle or wall it is not allowed to move, *i.e.* location and state remain unaltered, although it receives the same reward as if the action had succeeded. The agent receives a reward of -1 for each action that does not move it directly to the goal state and a reward of 0 for moving directly to the goal state. When the agent reaches the goal state it is relocated to a uniformly random start state.

For the 8 Adjacent Squares Agent there are multiple locations that appear to be the same state, e.g. state 148 (figure 1) as discussed in section 2.1 above.

3.2 Simple 1-D Example

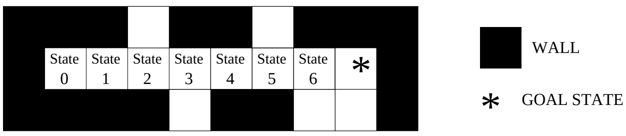

Figure 3: Simple 1-D example world to illustrate the problems caused by perceptual aliasing (Whitehead, 1992, pp73–78).

The simple 1-D example world consists of a 1×8 grid as shown in figure 3, with the goal at the far right hand side. An agent in this world can select between two physical actions; move east or move west. State transitions in this world are deterministic and the two actions move the agent one square in the appropriate direction. The agent is not allowed to move past either the far left hand end or far right hand end of the world; if it tries to do this its location and state remain unaltered. On reaching the goal state the agent receives a reward of 5000. Non goal states yield zero reward.

The arrangement of the wall and gaps above and below play no part in the actions the agent is allowed to execute. They do however encode the state as seen by the 8 Adjacent Squares Agent. For this agent each state appears unique except states labelled State 2 and State

5 (figure 3) which to it appear to be one and the same.

3.3 Learning Algorithms & Action Selection

A selection of learning and action selection algorithms were used: random action selection, SARSA, Q-learning, SARSA(λ) with replacement traces, Watkins's Q(λ) with accumulating traces. For details of the learning algorithms see Sutton and Barto (1998, p146, p149, p181, p184) respectively.

The random action selection algorithm, as its name suggest, selects uniformly between the available actions and provides a baseline for comparison with the other methods.

All of the learning algorithms continuously update their policies. Actions are selected greedily using the current policy with a probability of $(1-\epsilon)$. In cases where actions have the same value ties were broken at random. In the remaining ϵ cases the action executed was select randomly between all the available actions. In both cases above the random selection was uniform across all possibilities.

The following values were used for the learning algorithms:

- learning rate $\alpha = 0.1$,
- discount rate $\gamma = 0.9$,
- probability of random action ϵ started at 20% and decayed linearly reaching zero at the $200,000^{th}$ action-learning step. Thereafter it remained at zero.
- A range of values were tried for the eligibility trace decay rate λ from 0.001 to 0.9.
- The state-action values for all the learning algorithms were initiated at zero.

3.4 Evaluation

We adopted the same evaluation method as used by Loch and Singh (1998). After every 1000 learning steps the policy is evaluated greedily to determine the total number of steps required to reach the goal from every possible starting state. The agent is limited to a maximum of 1000 steps to reach the goal from each starting state. Thus if a policy is evaluated in a world with N starting states and fails to reach the goal from all of them it would have a maximum total steps of $N \times 1000$.

Each run consists of a million action steps, with evaluation of the current policy occurring every 1000 steps. Each combination of agent, world, learning algorithm and value of λ was repeated 100 times giving 100 samples per data point.

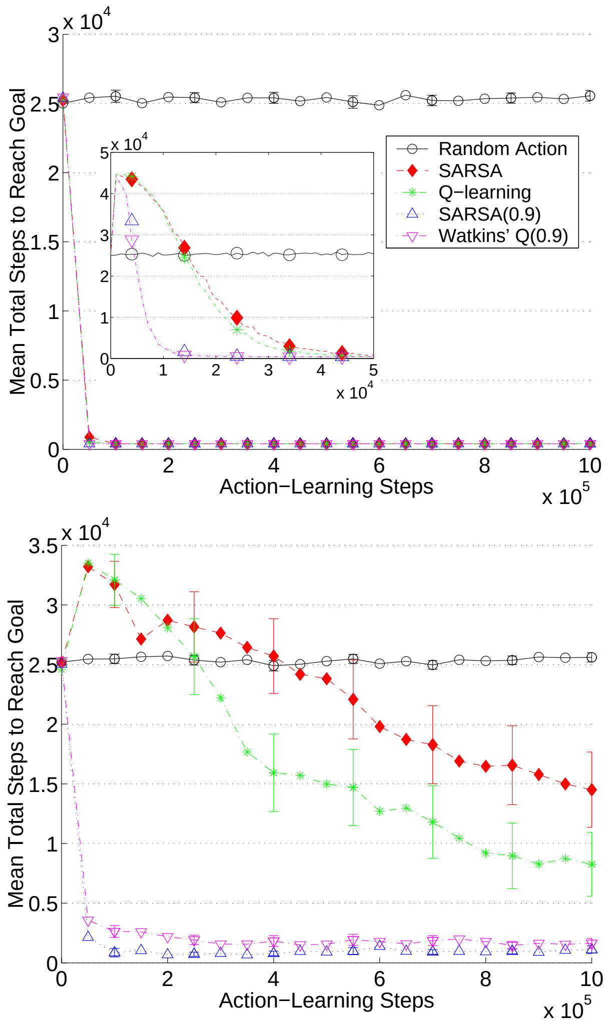

Figure 4: Plot for Sutton's Grid World of mean total steps found when policies were evaluated versus action-learning steps. Bars indicate 95% confidence intervals. To simplify plots data points are only shown for every 50,000 action-learning steps. Top graph shows results for Absolution Position Agent which suffers no perceptual aliasing. Insert shows enlargement of first 50,000 steps which would otherwise not be visible (data plotted for every 1,000 step). Lower plot shows results for 8 Adjacent Squares Agent which aliases multiple locations.

4. Results

4.1 Sutton's Grid World

Figure 4 shows the mean total steps for all four learning algorithms. The top plot shows results with the Absolute Position Agent which experiences no perceptual aliasing. All four learning algorithms quickly converge on the optimal solution in around fifty thousand action-learning steps. This indicates that all four learning algorithms have no problem in learning this task if there is no per-

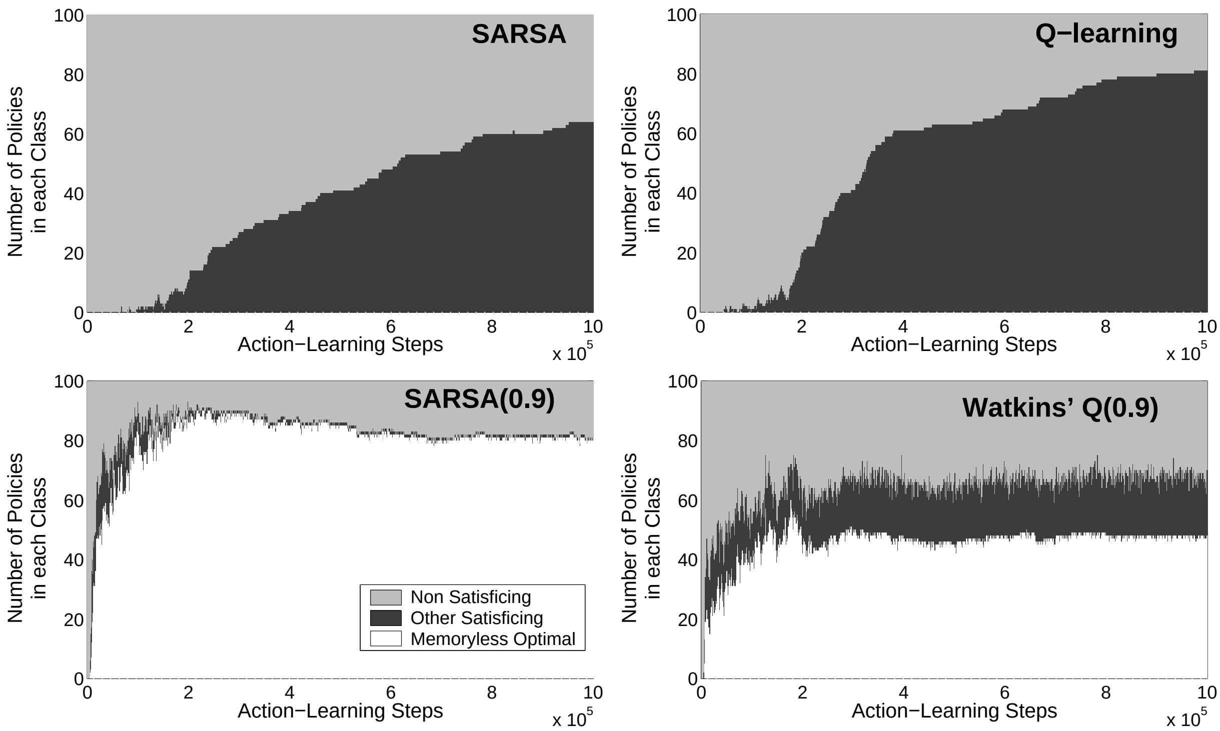

Figure 5: Plots show categorisation of policies versus action-learning steps for the four learning algorithms. All are learning to solve Sutton's grid world using the 8 Adjacent Squares Agent.

ceptual aliasing.

The lower plot shows results for the 8 Adjacent Squares Agent which aliases multiple locations in the world. The mean total steps for SARSA(λ) and Watkins's Q(λ) with $\lambda = 0.9$ rapidly approach the optimal memoryless solution, with SARSA(λ) reaching convergence in less than one hundred thousand action-learning steps, and Watkins's Q(λ) in around three hundred thousand action-learning steps. The other values of λ tried were 0.001, 0.005, 0.01, 0.05, 0.1, 0.5 and 0.75 (none of which are shown) and all converged to a similar number of mean total steps as $\lambda = 0.9$. As would be expected lower values of λ take longer to converge, with the lowest value, $\lambda = 0.001$, converging after eight hundred thousand action-learning steps.

The mean total steps of the policies learnt by by the 1-step backup algorithms, Q-learning and SARSA, appear to be gradually converging towards a level where the majority of policies will be satisficing. This convergence is, however, extremely slow compared to SARSA(λ) and Watkins's Q(λ). As indicated by the 95% confidence intervals there is a significant variation in the policies learnt by Q-learning and SARSA.

To obtain a better idea of the quality of the policies that are being learnt we have identified five policy categories and tracked the number of policies that fall into each category over time, see Figure 5. The five policy categories are optimal, better than memoryless optimal, memoryless optimal, other satisficing and non-satisficing. We will define these categories specifically for Sutton's grid world in terms of the total steps measured when the policy is evaluated. The optimal policy is defined as that which takes the minimum possible total steps to reach the goal from all starting positions. For Sutton's grid world this is 404 steps. Littman (1994) showed that the optimal memoryless solution for Sutton's grid world is 416 steps. Littman (1994)'s definition of satisficing is a policy that reaches the goal from all possible start states. Our measure of satisficing is stricter than this requirement as the agent is limited to 1,000 actions from each start state, after which the run is truncated. Accordingly, any policy who fails to reach the goal state from any start location in under 1,000 steps is classed as non-satisficing irrespective of the total steps for that policy. The remaining policies who succeed in reaching the goal from all start states are classified as: optimal if their total steps equals 404; better than memoryless optimal if their total steps lies between 404 and 416 (exclusive); memoryless optimal if the total steps equals 416; and other satisficing if the total steps exceeds

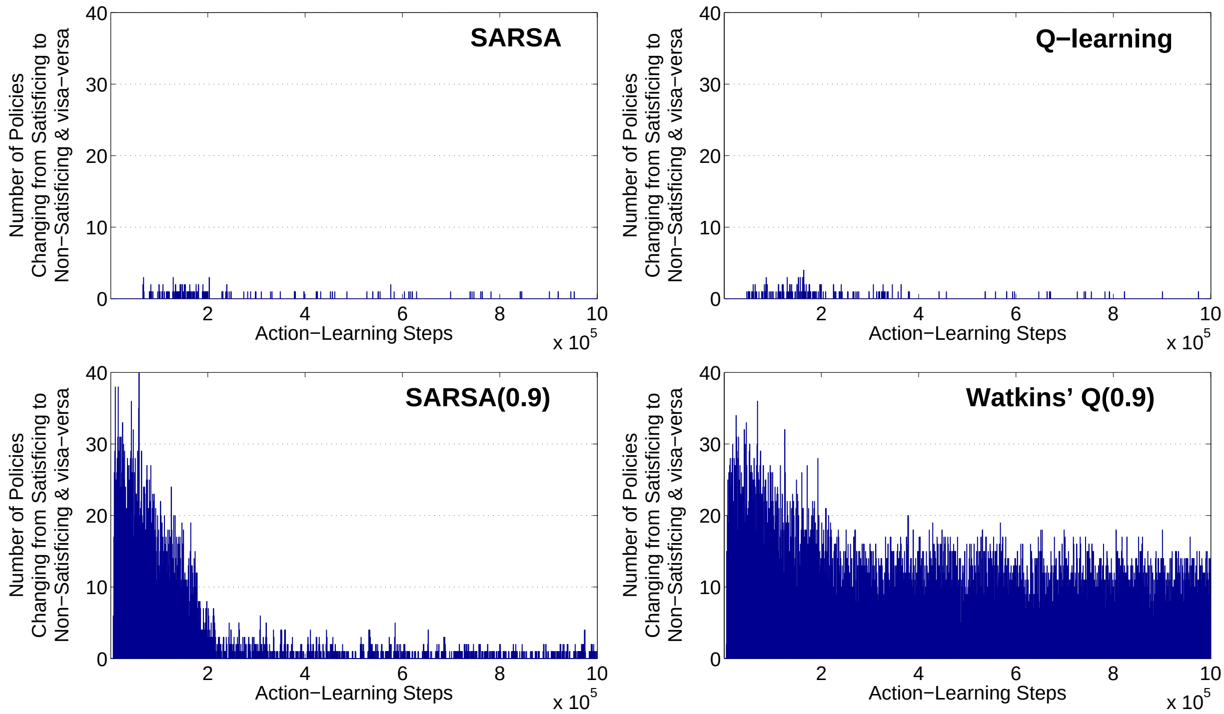

Figure 6: Plots of stability of policy classification versus action-learning steps for the four learning algorithms. Height of bars indicate the number of policies that changed from being satisficing to non-satisficing and visa-versa since the previous policy evaluation. All are learning to solve Sutton's grid world using the 8 Adjacent Squares Agent.

Goal Reached From All Start States	Total Physical Actions	Policy Category
yes	404	Optimal
yes	405 − 415	Better Than Memoryless Optimal
yes	416	Memoryless Optimal
yes	$>$ 416	Other Satisficing
no	-	Non-Satisficing

Table 1: Policy categories for Sutton's Grid World

416. These five categories are summarised in table 1.

Figure 5 shows the variation in types of policies learnt against the number of action-learning steps that have been executed. Each combination of parameters and learning algorithm was repeated 100 times. The height of the shaded areas on these plots indicate the number of policies falling into each policy category as measured after a particular number of action-learning steps. Examining the top left hand plot, which shows the policies learnt using SARSA, initially all one hundred policies are non-satisficing (grey shading). At around one hundred thousand action-learning steps a small number of policies become satisficing, but their total steps exceeds 416 so they are classified as other satisficing policies (black shading). The number of policies classified as other satisficing gradually increases until after one million action learning steps 64 of the policies are other satisficing and 36 are non-satisficing. The results for Q-learning (top right) are similar with final tallies of 81 other satisficing policies and 19 non-satisficing. Neither learnt any policies that were better than other satisficing at any stage. However, the other satisficing policies found were reasonable with a mean total steps of 457 for SARSA and 487 for Q-learning.

By comparison, SARSA(λ) and Watkins's Q(λ) had in less than six thousand action-learning steps, learnt a small number of policies that were classified as memoryless optimal (white shaded areas in the lower two plots of figure 5). By the end of a million action-learning steps the distribution of policies for SARSA(λ) were: 80 memoryless optimal; 0 other satisficing, and 20 non-satisficing. Similarly, after a million action-learning steps, the distribution of policies for Watkins's Q(λ) was: 48 memoryless optimal; 22 other satisficing, and 30 non-satisficing

There existed the possibility for all of the learning algorithms that although a given proportion of the population of policies were continually categorised as satisficing, individual policies were not stable, switching back and forth between satisficing and non-satisficing solutions. With this possibility in mind we examined the stability of the policies learnt by the four learning algorithms

Plots for stability are shown in figure 6. These are derived by counting the number of individual policies that change classification between consecutive policy evaluations. A change in classification is counted when a policy changes from being non-satisficing to any of the other satisficing classifications, or visa-versa. From figure 6 both SARSA and Q-learning are very stable with no more than three policies changing classification at any one time. Much large changes in classification are seen initially for SARSA(λ) and Watkins's Q(λ), as we would expect. The number of changes then steadies at a fairly low level for SARSA(λ), but remains relatively high for Watkins's Q(λ).

A related observation is that Watkins's Q(λ) generates a much large number of other satisficing policies than SARSA(λ). SARSA(λ) learns either memoryless optimal or non-satisficing polices, and virtually zero other satisficing policies. A quick investigation reveals that the memoryless optimal policies generated by Watkins's Q(λ) are reasonably stable, *i.e.* the number of changes between memoryless optimal policies and any other classification are comparable to the figures for SARSA(λ). However, a large number of polices, on average 11.6, flip between other satisficing and any other classification. This accounts for most of the changes reported on the Watkins's Q(λ) plot in figure 6. It thus appears that the policy update rule used by Watkins's Q(λ) learns a significant proportion of unstable, non memoryless optimal policies.

4.2 Simple 1-D Example

Results for the simple 1-D example world are shown in figure 7. The top two plots show mean total steps with 95% confidence intervals for the four learning algorithms and also for random action selection. The left hand plot shows results for the Absolute Position Agent which does not experience any perceptual aliasing. In the absence of any state aliasing all four learning algorithms learn the optimal solution to this world.

The 8 Adjacent Squares Agent (top right hand plot) aliases State 2 and State 5 (figure 3) in this simple 1-D world. Using this agent both SARSA and Q-learning perform worse than an agent selecting actions at random, though the large confidence intervals suggest that it is worth investigating what is occurring with the individual policies. SARSA(λ) and Watkins's Q(λ), $\lambda = 0.9$, both learn the optimal solution in less than 50,000 action-learning steps.

The total steps for an optimal solution to this problem is 28. Due to the very simple nature of this world the total steps for an optimal memoryless policy is also 28. Because these two types of policy are identical we reduce the number of categories used to just three, see table 2. Even though there appears to be only one solution, move east in all states, and policies are evaluated greedily, other satisficing policies could still exist. Ties where actions have the same value are broke at random. Thus it is possible to image a policy which, in one or more states, has no preference between moving east or moving west, such that an agent following this policy performs a limited random walk before ultimately reaching the goal. Such a policy, if it reached the goal in less than 1,000 steps from each starting state, would still be satisficing but would exceed 28 total steps. In practice this never occurred and only two categories are shown on the plots in figure 7.

Goal Reached From All Start States	Total Physical Actions	Policy Category
yes	28	Optimal
yes	> 28	Other Satisficing
no	-	Non-Satisficing

Table 2: Policy categories for Simple 1-D Example

The middle two plots (figure 7) show the number of policies falling into each category for SARSA and Q-learning. We see that both SARSA and Q-learning reach a plateau with just over 65% of the policies learnt being classified as optimal after just 300,000 action-learning steps. For these two learning algorithms we again plot the change in classification of the policies to test that the policies are stable. The two lower plots in figure 7 suggest that the optimal policies learnt are indeed stable.

Plots of the categorisation of policies learn by SARSA(λ) and Watkins's Q(λ) are not shown as all the policies were classified as optimal after 1000 action-learning steps and there is very little variation from this initial level for the remainder of the one million action-learning steps.

5. Discussion & Conclusions

The results successfully replicates those of Loch and Singh (1998) showing that SARSA(λ) can find optimal memoryless solutions to tasks containing perceptual aliasing. In fact this result generalises to Watkins's Q(λ) suggesting that any method that uses eligibility traces can find optimal memoryless solutions.

The surprise result was that SARSA and Q-learning could learn satisficing policies to Sutton's Grid World, and optimal policies for the simple 1-D example. The latter is even more remarkable as Whitehead (1992) presented the example in order to illustrate the extent to which perceptual aliasing can interfere with Q-learning and claimed that "1-step Q-learning cannot learn the optimal policy for this task" (p.73). In both instances the policies learnt appear to be stable.

Further examination of this issue indicates that ex-

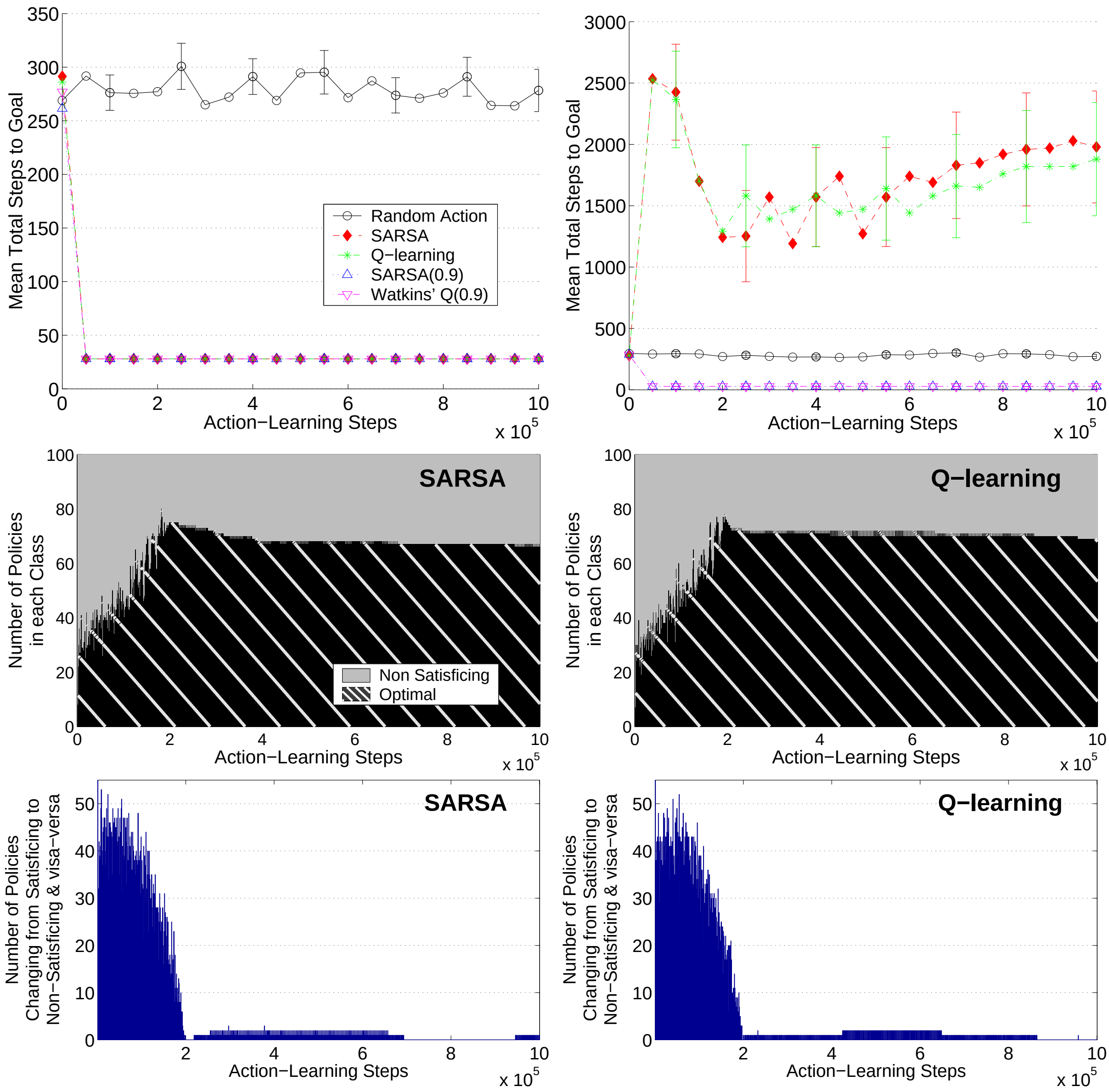

Figure 7: Top plots are mean total steps found when policies were evaluated versus action-learning steps. To simplify plots data points are only shown for every 50,000 action-learning steps. Left hand graph shows results for Absolution Position Agent which suffers no perceptual aliasing. Right hand plot shows results for 8 Adjacent Squares Agent which aliases states 2 and 5. Middle plots show categorisation of policies versus action-learning steps for SARSA and Q-learning. Bottom plots show stability of policy classification versus action-learning steps for SARSA and Q-learning. Height of bars indicate the number of policies that changed from being optimal to non-satisficing and visa-versa since the previous policy evaluation. All plots are for the Simple 1-D Example grid world and (with the exception of top left) the 8 Adjacent Squares Agent.

ploration is important in determining whether reinforcement learning algorithms which use 1-step backup, such as SARSA and Q-learning, can learn policies that are both stable and satisficing in partially observable environments. In the experiments presented above the probability of selecting an exploratory action (ϵ) starts at 20% but reaches zero after two hundred thousand action-learning steps. For the remaining eight hundred thousand steps the agent always follows the current policy without trying any exploratory actions. The lack of exploration appears to avoid the destructive effects of global impairment allowing policies to achieve stable solutions. The effect of exploration is nicely illustrated by figure 8 which shows the categorisation of policies learnt for the Simple 1-D Example world and 8 Adjacent Squares Agent, using Q-learning with ϵ fixed at 0.01. With a fixed value for ϵ the policies are not stable, and continuous oscillations are seen in the number of optimal policies. This is in contrast to the plateau seen in figure 7. A secondary point of note is that in figure 7 the number of optimal policies ramps up slowly as ϵ decreases from 0.2 to zero. This contrasts with figure 8 where, with a fixed, but initially lower, value of ϵ, the number of optimal policies learnt increases more rapidly. The observed oscillations reinforce Whitehead (1992, p.78)'s argument that Q-learning (or any 1-step backup algorithm) is unable to converge on stable solutions in partially observable environments, provided there is some possibility of selecting an exploratory action. However, once exploration has ceased, it is possible for 1-step backup algorithms to converge on satisficing policies.

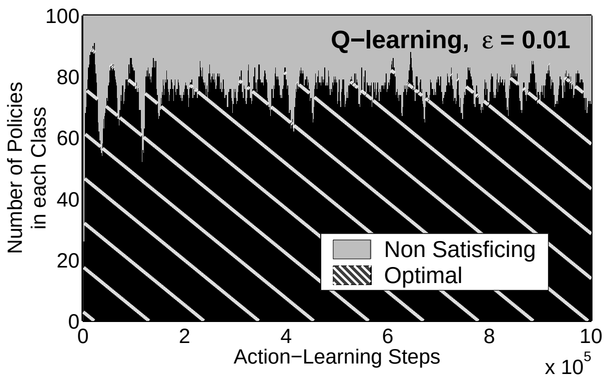

Figure 8: Plot shows categorisation of policies versus action-learning steps for Q-learning with ϵ fixed at 0.01, for the Simple 1-D Example world and 8 Adjacent Squares Agent.

Of our initial hypotheses we have confirmed the second, that reinforcement algorithms that use eligibility traces can learn optimal memoryless policies, though a question can be raised as to the stability of the solutions given that around 10% of the policies learnt by Watkins's Q(λ) appear to be flipping between satisficing and non-satisficing, see figure 6.

Our first hypothesis "1-step reinforcement learning algorithms are not able to learn policies which are both stable and optimal, when the task involves perceptually aliased states," needs to be modified in light of the above discussion to reflect the importance of selecting exploratory actions. Although this is an interesting result, it is apparent that policies learnt using SARSA and Q-learning converge on satisficing solutions very slowly.

The main aim of the experiments presented above is to illuminate the problems that occur when applying reinforcement learning to partially observable environments. We are interested in doing this in order to clear the ground before moving on to look at whether active perception can be used to address these issues. These results are, however, of interest in their own right as reinforcement learning is often used in robotics, and real limited sensor arrays certainly create the possibility of perceptual aliasing of states. An important observation, therefore, is that if there exists the possibility of state aliasing, then it either needs to be designed out of the task, or careful selection should be made of the learning algorithm. For example, it is probably worth avoiding reinforcement learning algorithms that do 1-step backup. In fact any reinforcement learning algorithm that uses truncated returns will be subject to some detrimental effects of global impairment (Whitehead, 1992, p.80). However, as demonstrated by the above results, reinforcement learning algorithms that use eligibility traces can quickly learn reasonable solutions.

6. Future Work

The main focus of our future work is to test the conjecture that active perception can allow reinforcement learning algorithms which are not enhanced using memory or internal world models, to find optimal solutions to navigation problems which involve perceptual aliasing. This we plan to investigate initially by equipping grid world agents with some form of self directed, perceptual system. Ultimately, we would like to prove our approach using a mobile robot navigating a building, the robot's input state being formed from the images captured by an on board camera, which can pan, tilt and zoom, and over which the robot's learning algorithm has direct control.

Acknowledgements

Paul Crook would like to thank EPSRC without whose funding this work would not be possible. We would also like to thank the reviewers for their useful comments.

References

Chrisman, L. (1992). Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. In *Tenth National Conference on Artificial Intelligence*, pages 183–188. AAAI/MIT Press.

Lanzi, P. L. (2000). Adaptive agents with reinforcement learning and internal memory. In Meyer, J.-A. et al., (Eds.), *From Animals to Animats 6: Proceedings of the Sixth International Conference on the Simulation of Adaptive Behavior (SAB'2000)*, pages 333–342. The MIT Press, Cambridge, MA.

Littman, M. L. (1994). Memoryless policies: Theoretical limitations and practical results. In Cliff, D. et al., (Eds.), *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior (SAB'94)*, pages 238–245. The MIT Press, Cambridge, MA.

Loch, J. and Singh, S. (1998). Using eligibility traces to find the best memoryless policy in partially observable Markov decision processes. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML'98)*, pages 323–331. Morgan Kaufmann, San Francisco, CA.

McCallum, A. K. (1993). Overcoming incomplete perception with utile distinction memory. In *Proceedings of the Tenth International Conference on Machine Learning (ICML'93)*, pages 190–196. Morgan Kaufmann, San Francisco, CA.

Singh, S. P., Jaakkola, T., and Jordan, M. I. (1994). Learning without state-estimation in partially observable Markovian decision processes. In *Proceedings of the Eleventh International Conference on Machine Learning (ICML'94)*, pages 284–292. Morgan Kaufmann, San Francisco, CA.

Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the Seventh International Conference on Machine Learning (ICML'90)*, pages 216–224. Morgan Kaufmann, San Francisco, CA.

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

Whitehead, S. D. (1992). *Reinforcement Learning for the Adaptive Control of Perception and Action.* PhD thesis, University of Rochester, Department of Computer Science, Rochester, New York.

Whitehead, S. D. and Ballard, D. H. (1991). Learning to perceive and act by trial and error. *Machine Learning*, 7(1):45–83.