16.412 Cognitive Robotics

Cooperative Q Learning

Steve Block, joint with Lars Blackmore

May 10th 2004

Abstract

Q-learning is an algorithm for conducting reinforcement learning, which uses the reinforcement signals received by an agent to learn the optimal policy for acting in a non-deterministic environment defined by a Markov Decision Process.

Recent work has extended Q-learning to allow cooperation between multiple agents by sharing Q-values. The concept of *expertness* has been introduced as a means of assesing the relative worth of an agent's Q-values and the concept of a *specialized* agent allows this assessment to be done over subsections of the state space. These advances define *expertness based cooperative Q-learning with specialized agents* and limited results for this algorithm have been presented for simulations of mobile robots learning in grid world.

This work validates, refines and extends the current capabilities of the algorithm in the following ways. Firslty, investigations were carried out into a variety of implementation issues with the current algorithm. The results presented in previous papers were replicated and simulations were carried out to broaden the conclusions made in previous work and to hightlight the limitations of the current algorithm. Secondly, an assessment was made of the performance of the algorithm in dynamic environments. Thirdly, *discounted expertness* was proposed, implemented and tested as a means of overcoming the possible decrease in performance seen in dynamic environments.

Contents

1 Introduction

This paper builds on previous work by Tan, Ahmadabadi, Asadpour and Ashgh, the result of which is an algorithm for *expertness based cooperative Q-learning with specialized agents*. This work aims to validate, refine and extend the current capabilities of this algorithm.

Firstly, simulated results previously presented for this algorithm are reproduced. In doing so, a number of implementation details are investigated and discussed. Further simulations are conducted and the results are used to define more precisely the capabilities and limitations of the algorithm.

Secondly, the algorithm is tested in dynamic environments and its performance is assessed. In particular, the effects of cooperation in such an environment will be tested, as it is important to determine whether cooperation ever produces a decrease in performance relative to individual learning.

Thirdly, the concept of *discounted expertness* is proposed as a means of avoiding some of the problems encountered in dynamic environments. This concept is then implemented and tested in simulation.

2 Previous Work

2.1 Cooperative Q-Learning

The concept of cooperation between agents involved in Q-learning was first proposed by Tan[1] and Whitehead[2]. Tan offered an introductory discussion of ways in which this problem could be addressed and proposed that agents cooperate through the sharing of information. He suggested three possibilities for information that could be shared: sensory information, experiences and Q-values.

Sharing sensor information increases an agent's ability to perceive its environment, so is equivalent to expanding the size of the observed state space. Sharing experiences, by means of observation, action, reward triples, allows an agent to learn from the data acquired by another agent, without undertaking the actions itself. Both of these are valid techniques, and while the implementation is straightforward, the benefits are limited. Sharing Q-values, however, allows the agents' policy to be shared directly and offers the greatest potential for improving the performance of a group of collaborating agents.

Tan used *simple averaging* to share Q-values between agents. For any agent i, the value of $Q_i(s,a)$ after cooperation is simply equal to the average of the values held by the population of agents before cooperation. The cooperation update equation is as follows, where $Q(s,a)$ is the Q-value for taking action a from state s.

$$Q_i(s,a) \leftarrow \frac{\Sigma_{j=1}^{n} Q_j(s,a)}{n}$$

Ahmadabadi and Asadpour[3] reported the disadvantages of simple averaging. Firstly, the technique is non-optimal when the agents have different levels of experience. This is because the Q-values held by each agent are weighted equally during sharing and no preference is given to values which may represent a superior policy. Secondly, in general, when compared to individual learning, *simple averaging* reduces the rate at which an agent's Q-values converge towards an optimal policy.

Ahmadabadi and Asadpour presented simulation results for a hunter-prey scenario which demonstrated that *simple averaging* gives a decrease in performance relative to individual learning. Eshgh and Ahmadabadi[4] presented simulation results for mobile robots learning in a segmented maze world and these too showed a significant decrease in performance.

2.2 Expertness Based Cooperative Q-Learning

Ahmadabadi and Asadpour introduced the concept of *expertness* as a measure of how expert an agent is at obtaining rewards. This value can then be used to weight the Q-values offered by other agents during cooperation. This gives the following cooperation update equation, where $W_{i,j}$ is the weighting that agent i assigns to Q-values held by agent j.

$$Q_i(s,a) \leftarrow \Sigma_{j=1}^{n} W_{i,j} Q_j(s,a)$$

Note that for *simple averaging*, the weighting is a constant for all i,j pairs.

$$W_{i,j} = \frac{1}{n}$$

Ahmadabadi et. al.[3][4] proposed a number of definitions for this expertness measure, the most successful of which were based on the reinforcement signals received by the agent. These included the *absolute* expertness

measure, which is a sum of the absolute values of the reinforcement signals received by an agent over its lifetime and reflects the fact that both positive and negative reinforcements contribute to improving an agent's policy. Although other reinforcement-based expertness measures were shown to be optimal in certain circumstances, the absolute measure is the superior choice in the general case.

$$e_i = \Sigma_t |R_t|$$

Ahmadabadi and Asadpour also suggested a number of methods for calculating the weighting $W_{i,j}$ from the expertness values e_i and e_j of the two agents. One method, *learning from experts*, defines a scheme where agents only consider Q-values provided by agents more expert than themselves. This gives the following expression for the weighting function $W_{i,j}$, where k indexes over the m agents where $e_k > e_i$.

$$W_{i,j} = \begin{cases} 1 & i = j, e_i = max_i\left(e_i\right) \\ 1 - \alpha_i & i = j, e_i \neq max_i\left(e_i\right) \\ \alpha_i \frac{e_j - e_i}{\Sigma_{k=1}^{m}(e_k - e_i)} & e_j > e_i \\ 0 & otherwise \end{cases}$$

2.3 Expertness Based Cooperative Q-Learning with Specialized Agents

Eshgh and Ahmadabadi[4] extended their definition of expertness based cooperative Q-learning to allow an agent's expertness to be a function of state. The state space was split into a number of zones and the agents are assigned an expertness value for each. Note that this expertness value is a measure of confidence in the Q-values corresponding to all state, action pairs for all of the states contained in the zone. The following three possibilities were suggested for the definition of the expertness zones.

global The entire world, as was the case previously

local A small number of zones based on the topography of the environment

state An expertness zone is established for every state

The expression for the weighting assigned by agent i to agent j for a Q-value corresponding to a state, action pair in zone z is then as follows.

$$W_{i,j,z} = \begin{cases} 1 & i = j, e_{i,z} = max_i\left(e_{i,z}\right) \\ 1 - \alpha_i & i = j, e_{i,z} \neq max_i\left(e_{i,z}\right) \\ \alpha_i \frac{e_{j,z} - e_{i,z}}{\Sigma_{k=1}^{m}(e_{k,z} - e_{i,z})} & e_{j,z} > e_{i,z} \\ 0 & otherwise \end{cases}$$

2.4 Results

Eshgh and Ahmadabadi carried out simulations of three mobile robots learning in a maze world. Obstacles are arranged in the world such that it is approximately segmented into three regions, with a goal located in each. Although movement from one segment to another is possible, the topology of the world made this unlikely. A single large reward is received for finding the goal, a punishment for colliding with walls and a small punishment for all other movements.

Initially, each robot is assigned to a region of the world and conducts a certain number of trials. Each robot begins a trial at a random start location within its region and conducts Q-learning without any form of

cooperation until it finds any of the three goals. Each agent retains its Q values from one trial to the next, so learning is continuous throughout this phase.

Once all trials are complete, the agents share their Q-values as described above, using the *absolute* expertness measure and the *learning from experts* weighting strategy.

Following cooperation, all three agents conduct trials from random start locations spread throughout the entire world. Eshgh and Ahmadabadi do not state explicitly whether or not the agents continue to learn during this phase and whether or nor they continue to share their Q-values at regular intervals in time.

Simulations were conducted using *global*, *local* and *state* expertness zones, where the *local* zones are the regions into which the world is segmented. Their results showed that cooperation produced a decrease in the average number of steps required to reach the goal of over 50% for either *local* or *state* expertness. The average number of steps to reach the goal increased slightly when global expertness was used.

2.5 Conclusions

Previous work has suggested a number of different ways in which agents involved in Q-learning can cooperate to improve their policy. The method chosen for further development is sharing Q-values, and the most refined algorithm is *expertness based cooperative Q-learning with specialized agents*. This has been shown to offer the greatest improvement in performance relative to individual learning for a segmented world if either *local* or *state* expertness is used.

3 Expertness Zones

Eshgh and Ahmadabadi close their paper by stating that 'finding proper methods to identify the area of expertise in Q-tables automatically is the next goal of this research'.

Intuitively, one would expect state expertness to be the best choice, as it distinguishes between the agents' expertness as the finest level. The disadvantage of this method is the space required to store the many expertness values and the added computation required to compute the weightings. At the other end of the scale, global expertness requires storage of only a single value and allows simple computation of the weightings, but offers the worst performance. At first glance, therefore, a trade-off is required.

In order to implement expertness zones for an arbitrary assignment of zones, we must store the parent zone for each state and the expertness value for each zone. This means that the storage requirements are linear in the number of the states and the number of zones. Given this storage structure, the computational expense of looking up the zone to which a state belongs and then looking up the expertness assigned to that zone is constant with respect to both the number of states and the number of zones. However, if we simply store an expertness value for each state directly, both the storage requirement and computational expense is reduced. Therefore, we conclude that state expertness is in general the optimum zone assignment.

There are, however, possible exceptions to this conclusion. Firstly, there may be ways in which the assignment of states to zones can be parameterized such we need not store the member zone for every state, thus reducing storage requirements. Secondly, there may be certain cases in which state expertness is not the best method for assigning weightings. Continuity between the Q-values in adjacent states can be lost during cooperation when no single agent is significantly more expert than the others. This is accentuated by the use of state expertness and it may be the case that assessing expertness over groups of adjacent states would improve performance.

In practice, no simple parameterization of the states to form zones was found. Also, the tests conducted in Section 8 suggest that Q-learning will rapidly overcome any problems caused by discontinuities through its ability to conduct local repair. Therefore, although this area has not been investigated fully, it seems that state expertness is likely to be the optimal choice and it was therefore used in all of the simulations presented in this report.

4 Mobile Robot Simulation

All of the results presented in this report are for the simulation of mobile robots operating in a grid world. The scheme is similar to that used by Eshgh and Ahmadabadi, but has been generalized to allow a number of additional parameters to be varied. Each simulation consists of three successive phases; *individual learning*, *cooperation* and *testing*.

During individual learning, each agent of the n agents conducts Q-learning without any form of cooperation. Agent i starts from state s_i and conducts N_i trials, the first of which is at time t_i, where a trial is used as the unit of time. A trial ends when the agent finds any of the goals or when the number of steps reaches 1000, and Q-values are retained between trials.

Cooperation occurs at time $t = 0$ and is when agents may first share their Q-values. If sharing does take place, the *absolute* expertness measure and the *learning from experts* weighting strategy are used unless explicitly stated otherwise.

During the testing phase, each agent conducts 100 trials from each of the members of the set S of start locations. Set S is the union over all i of the start locations s_i used by the agents during individual learning. During this phase, each agent may or may not learn during a single trial and any changes made to its Q-values during that trial may or may not be retained for the next trial. In addition, the agents may or may not share their Q-values after each trial. As for individual learning, a trial ends when the agent reaches any of the goal states or when the number of steps reaches 1000. Once the trials for a given start location are complete, any changes made to the Q-values since the initial sharing during the cooperation phase are discarded. This is because the test trials from each start location are intended to be independent, so learnt policy should not be retained from one to the next.

The four grid worlds used for in the simulations presented in this report are shown in Figures 1 to 4. Black checked squares represent impassable obstacles, red checked squares represent the start locations s_i and are numbered from left to right, as are the goal locations, which are denoted by blue checked squares. In Sections 8 and 9, the *doors* and *contrived* worlds are used with dynamic obstacles, and these are discussed when they are introduced.

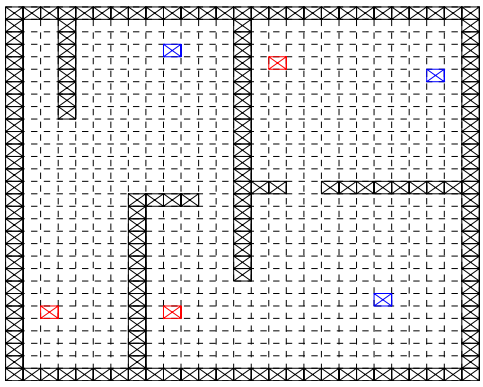

Figure 1: The *segmented* grid world, with three start and goal locations

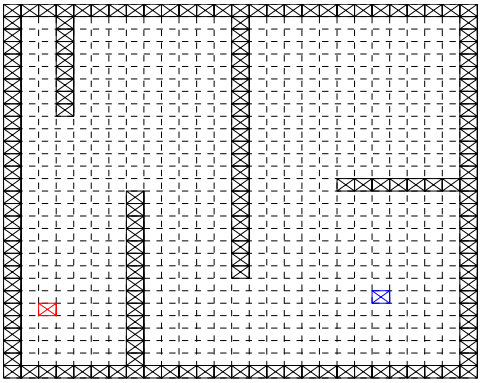

Figure 2: The *maze* grid world, with a single start and goal location

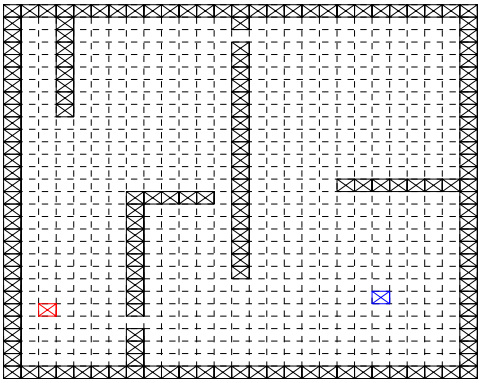

Figure 3: The *doors* grid world, with a single start and goal location

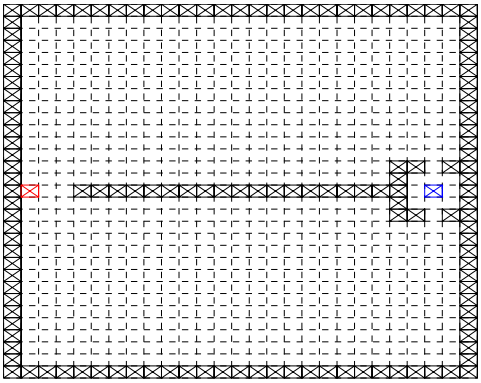

Figure 4: The *contrived* grid world, with a single start and goal location

5 Presentation of Results

All of the simulation results presented in this report are displayed using the following two types of figure.

5.1 Q-Fields

An agent's Q-values determine its motion policy at that point in time. A graphical representation of this policy was found to be very useful in gaining insight into the progress of the agents learning and the effects of sharing. Figure 5 shows the *segmented* grid world with tick-marks plotted in each state. The tail of the tick-mark points in the direction of the action corresponding to the maximum Q-value outgoing from that state. The tick-marks therefore form a vector field, or *Q-field*, which defines the agent's policy throughout the world.

Furthermore, the color of the tick-mark represents the magnitude of the Q-value, with red, green, blue and black representing ranges of values in order of decreasing magnitude. The range of values are chosen such that red corresponds to regions to which the reward received at the goal has been propagated through successive exploration. At the other extreme, black corresponds to regions that the agent has not visited, so the Q-values are unperturbed.

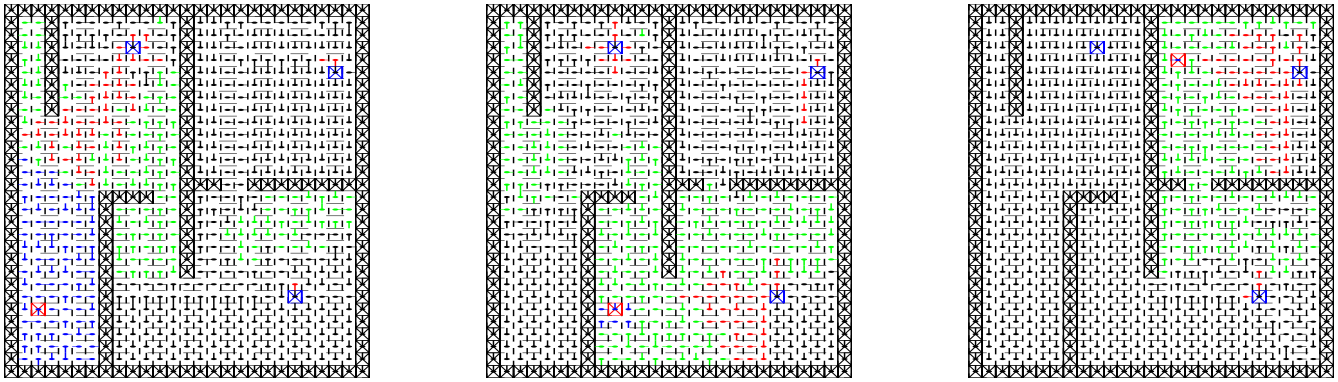

Figure 5: Sample Q-field plot

5.2 Performance Plots

The measure used to assess the performance of an agent is the number of steps required to reach a goal and this is plotted on the y-axis of the sample performance plot in Figure 6. The x-axis represents time, measured in number of trials. This approach is adopted to represent the fact that agents learn not only for different numbers of trials during individual learning, but also at different times. Recall that time $t = 0$ is the time of cooperation, when agents may first share their Q-values. Different colored lines are plotted for each of the n agents. Recall also that during the testing phase, all agents complete 100 trials from each of the possible start locations, so a performance plot is presented for each of these cases.

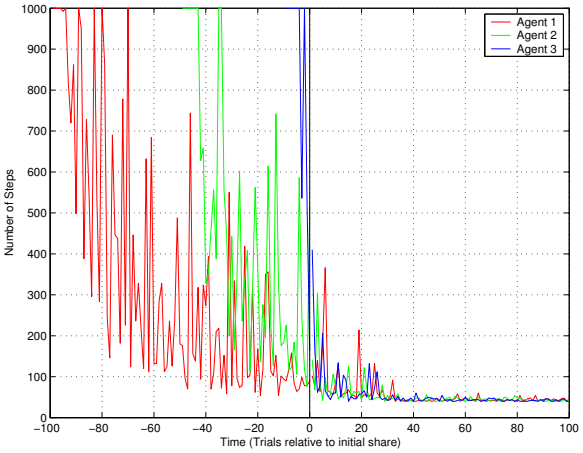

Figure 6: Sample performance plot

6 Initial Findings

Before progressing to the main objectives of the investigation, certain issues of implementation were addressed.

6.1 Initializing Q-Values

The array of Q-values held by each agent is updated incrementally during learning so must be initialized to some value at the beginning of each simulation. Eshgh and Ahmadabadi assigned values randomly between the maximum and minimum reinforcement signals that exist in the world and the other obvious choice is to use zero for all values. Tests showed that initializing the Q values to zero gives a slight increase in performance relative to using random initial values, but more importantly, this choice allows increased insight into the progress of the algorithm. In particular, initializing to zero allows meaningful use of the Q-field plots discussed in Section 5.1, so this choice was used throughout.

6.2 Learning Parameters

The Q-learning parameters used by Eshgh and Ahmadabadi were learning rate $\alpha = 0.8$ and discount factor $\gamma = 0.9$. Other constant values used were action selection temperature $T = 1.5$ and impressibility $\alpha_i = 0.8$ for all agents. The results presented here use identical values for the Q-learning parameters α and γ, but it was decided to use different values for T and α_i.

The action selection temperature T determines the probability with which the action suggested by the agent's policy is selected and represents the trade-off between exploitation and exploration. The probability of following policy increases as the temperature is lowered and reaches unity at $T = 0$. It was decided to set $T = 0$ to reduce noise in performance data and to aid comprehension of experimental results.

The impressibility α_i is the proportion of the total weighting that agent i allocates to other agents during cooperation. If $1 - \alpha$ is greater than zero, then an agent will incorporate its own Q-value when summing over the contribution of all agents irrespective of its own lack of expertness. This became a problem in the dynamic environments with expertness discounting discussed in Section 9, where an agent may posses well-converged, and therefore large, Q-values, but have a very low expertness due to the fact that this information is out-of-date. In this situation, even small values of $1 - \alpha$ will correspond to a significant contribution from the large Q-values during cooperation, and this disrupts the intended purpose of the sharing procedure. Therefore, a value of $\alpha = 1$ was used throughout. Note that $\alpha = 1$ will not lead to homogeneity of the agents Q-values, since the *learning from experts* weighting scheme is dependent on an agent's own expertness.

6.3 Random Start Locations

Eshgh and Ahmadabadi used random start locations for both the individual learning and testing phases. This obviously presents the agents with a more difficult task than using a small number of possible start locations and consequently increases the average number of steps per trial and the number of trials required to reach a given level of convergence. In addition, random start locations mean that the *optimum* path length to the goal is now a random function of the trial number. This is manifested as additional noise in the performance plots and makes interpretation of the results more difficult. As a result, it was decided to use only a small number of possible start locations and to test the agents from each location individually.

6.4 Testing without Learning

Eshgh and Ahmadabadi are unclear as to whether the agents continue Q-learning during the testing phase. For reasons of increased execution speed, initial simulations were conducted without learning during the testing phase, but this proved to be very unsuccessful.

After completing the individual learning phase, there is no guarantee that the Q-field will not contain 'loops' where the agent's policy directs it around a closed curve, thus preventing it from reaching the goal. In practice, these were found to be fairly common, and the mixing of different agents' policies caused by cooperation tends to increase their frequency. Without continual Q-learning, an agent never modifies its Q-values and hence will remain 'stuck' in these loops until either it randomly selects to ignore follow policy (for $T > 0$), or the non-deterministic nature of the world produces an unexpected result for the chosen action. In practice, this requires a significant number of steps and performance is very poor as a result. An agent involved in Q-learning, however, will not get stuck because the Q-value updates made after executing each action will redirect the Q-field and hence break the loop. In practice, this was found to happen within a small number of steps; typically within the first cycle. The conclusion, therefore, is that agents should continue Q-learning in the testing phase and this was used in all of the results presented in here.

This leaves open the question of whether an agent should retain the modifications made to its Q-values during each testing trial, or whether they should be discarded. During individual learning, when the agent's policy is relatively poorly converged, the performance plots show a great deal of noise in the number of steps taken to reach the goal. This is due to the fact that the Q-values are continually being updated and whilst the Q-learning algorithm guarantees only that the policy will converge in the limit of infinite experience, each incremental update may produce either an increase or decrease in performance. If the modifications made to the Q-values in each trial of the testing phase are discarded, then the agent starts each trial with an identical policy and this removes much of the noise. However, this also means that the agent's performance during testing is heavily dependent on the performance in the final trial of individual learning and this is itself subject to significant noise.

If the updated Q-values are retained between trials, each agent conducts standard Q-learning during the testing phase, so whilst performance is noisy, it will improve as more trials are conducted and the dependence upon the final trial of individual learning is lessened significantly. This approach was therefore adopted for all of the simulations described in this report.

6.5 Repeated Cooperation

Eshgh and Ahmadabadi are also unclear as to whether or not agents share their Q-values at regular intervals during the testing phase. Initial results showed that whilst cooperation is helpful between agents of different experience and therefore performance, it is not helpful between agents of similar experience and performance. After the initial share, the agents' performances become very similar and we would therefore expect that further cooperation will only be helpful if agents subsequently learn individually in such a way that their experiences diverge significantly. This does not occur for the worlds investigated in this report within the 100 trials of the testing phase, so we would expect continual sharing to be of little benefit and tests confirmed this result. Also, the time required to share the Q-values is typically an order of magnitude greater than the time required for a single trial. As a result, it was decided to disable repeated sharing during the testing phase in all of the simulations described in this report.

6.6 Assessment of Performance

It is important to use an appropriate measure of performance when comparing results obtained from groups of agents learning with and without cooperation. Tan remarks that

> *the more practical study is to compare the performance of n independent agents with the one of n cooperative agents*

However, the method used to asses the population of agents depends heavily on the task under consideration. For example, in a goal-seeking scenario, performance may determined by the number of steps required for *any* agent to reach a goal, or for *all* agents to reach the goal. In the first case, only the performance of the most capable agent matters and that of the least agent is unimportant, but the reverse is true in the second case.

In this report, the performance of each agent is measured individually. This allows conclusions valid for both of these assessment methods to be drawn from the results presented.

7 Static Environments

In the case of a static environment, simulations were conducted to achieve the aims listed below.

- Duplicate the results presented by Eshgh and Ahmadabadi
- Assess the performance of the algorithm in a more general maze world
- Investigate relevant implementation details

7.1 Segmented World

Eshgh and Ahmadabadi presented results for mobile robots in a segmented world and concluded that co-operation provides an increase in performance relative to agents learning without cooperation. In order to replicate these findings, the *segmented* world shown in Figure 1 was used, with three agents. During individual learning, each agent completed 50 trials starting at time $t = -50$. Firstly, the individual learning phase was simulated and the agents' Q-values saved. This data set was then used as the starting point for simulations of the testing phase both with and without prior sharing of Q-values in the cooperation phase.

7.1.1 Results

The Q-fields for the three agents are shown before and after cooperation in Figures 7 and 8 respectively. The performance plots for the simulation without cooperation are shown in Figures 9, 10 and 11, where the testing phase is conducted from start locations 1, 2 and 3 respectively. The corresponding information for the simulation with cooperation is shown in Figures 12, 13 and 14.

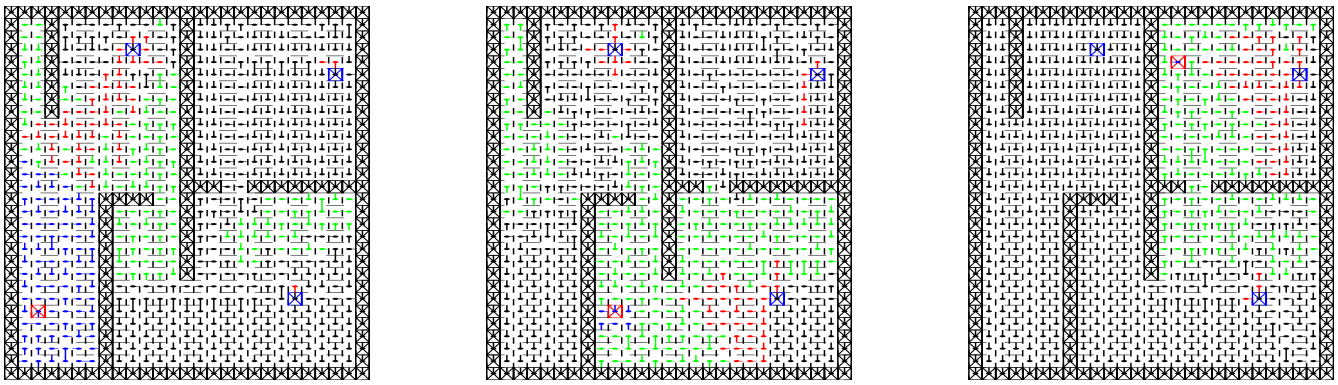

Figure 7: Segmented world, Q-fields after individual learning phase

7.1.2 Discussion

The plot of the Q-fields after individual learning shown in Figure 7 shows that each agent has learnt a reasonably converged policy from its start location to the nearest goal, as indicated by the large red area of the field. Correspondingly, the performance in the individual learning phase shows a steady decrease in the number of steps taken for each agent to reach a goal, representing the convergence of the policy. The

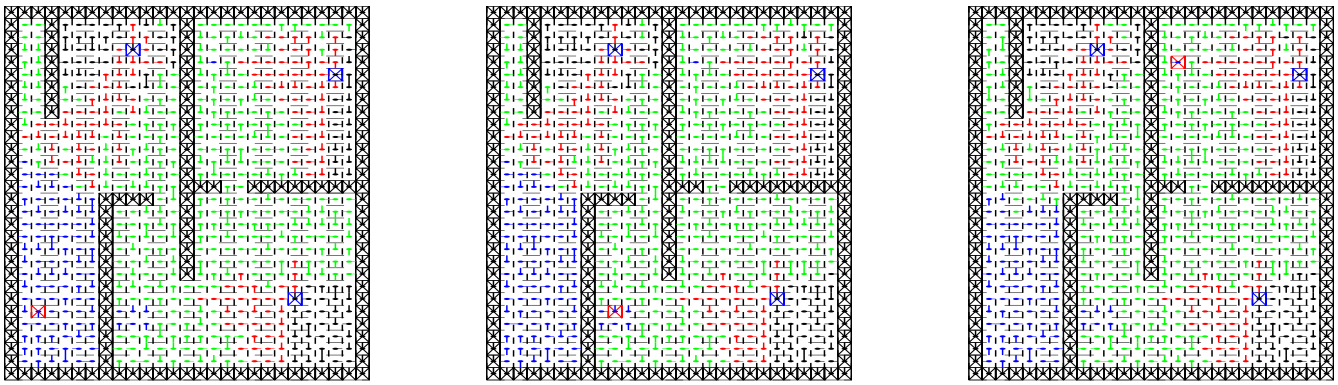

Figure 8: Segmented world, Q-fields after cooperation phase

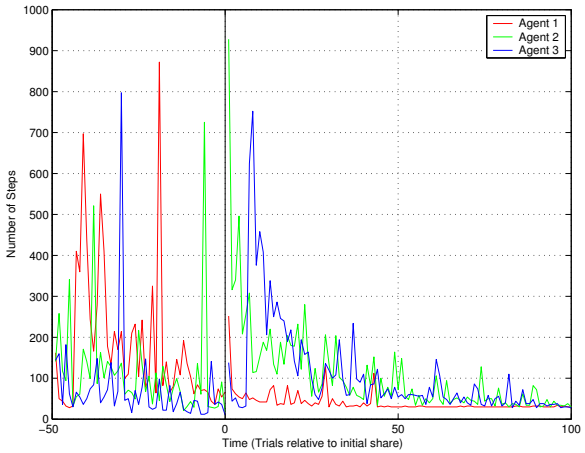

Figure 9: Segmented world, performance without cooperation, for testing phase trials from start location 1

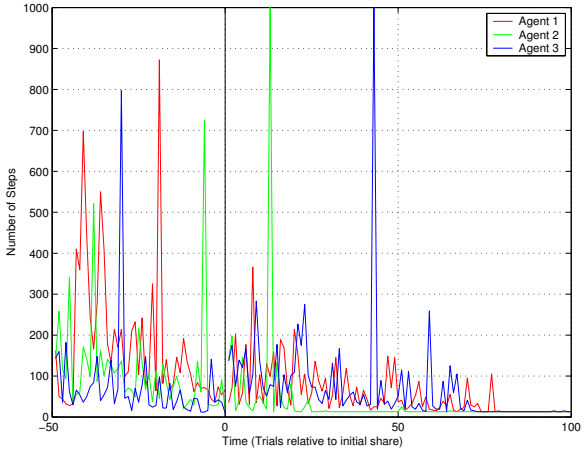

Figure 10: Segmented world, performance without cooperation, for testing phase trials from start location 2

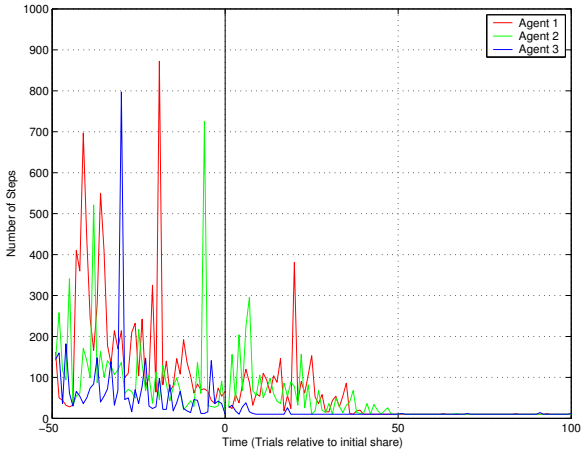

Figure 11: Segmented world, performance without cooperation, for testing phase trials from start location 3

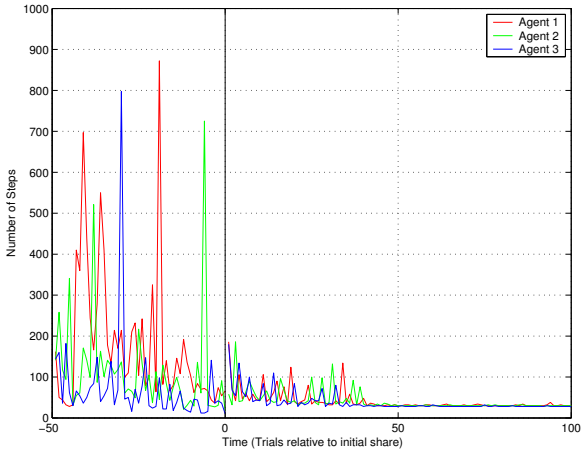

Figure 12: Segmented world, performance with cooperation, for testing phase trials from start location 1

red areas around the other two more distant goals, however, are relatively small, showing that there is little converged policy in these areas. In general, the Q-field in both regions not containing the agent's own start location is mostly black, representing a lack of experience and very poor policy.

In the case of no cooperation, Figures 9 to 11 show that for each start location, at the start of the testing phase, the performance of one agent continues to converge, whereas the performances of the other two becomes suddenly significantly worse. This is because for each start location, one agent is being tested at the location from which it began its initial trials, whereas the other two agents are being tested from 'foreign' locations, so they have very little experience of the local area and a correspondingly poor policy. Over time, all three agents show an improvement in performance in the testing phase, due to the action of their own learning and their policies begin to converge.

By sharing Q-values, each agent benefits from the policy derived from the experiences of its peers, despite having no personal experience in that region. This is shown vividly in Figure 8, where each agent now has a relatively well converged policy between each pair of start and goal locations, demonstrated by the red areas

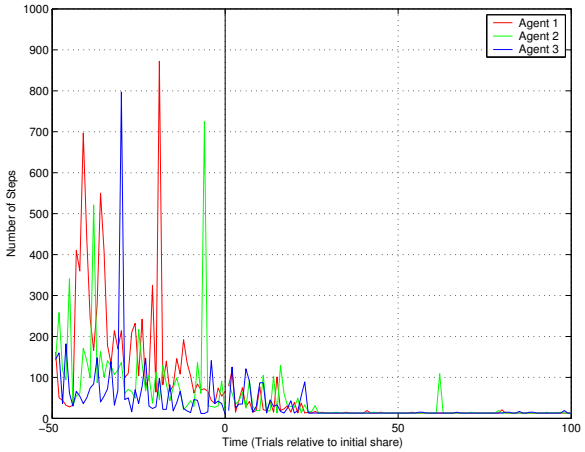

Figure 13: Segmented world, performance with cooperation, for testing phase trials from start location 2

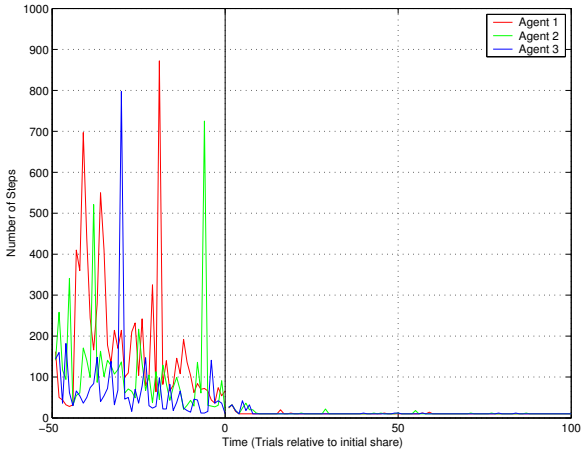

Figure 14: Segmented world, performance with cooperation, for testing phase trials from start location 3

of the Q-field. Furthermore, all three agents show little black in their Q-fields, representing the fact that they now have at least some policy in almost all parts of the world. Therefore, as a result of cooperation, all three agents display extremely good performance in the testing phase, as shown in Figures 12 to 14. During testing, continued learning causes this policy to converge further.

It has been shown, therefore, that in a segmented world, cooperation significantly improves the performance of an agent when tested in a region in which it has little or no experience. Conversely, however, cooperation is of no benefit to an agent which already possesses experience in the region in which the test is conducted. Hence the conclusion presented by Eshgh and Ahmadabadi has been confirmed.

7.2 General Maze World, Equal Experience Case

The *maze* map shown in Figure 2 was used to test cooperative Q-learning in a more general goal-seeking scenario. Three agents were used, each conducting 50 trials from time $t = -50$ in the individual learning

phase from a single start location. Although each agent will have slightly different expertness values in each state, they have completed an equal number of trials so their overall levels of *experience* will be approximately equal. As before, the Q-values were saved after completion of the individual learning phase and subsequently used as the starting point for simulations of the testing phase both with and without prior sharing of Q-values in the cooperation phase.

7.2.1 Results

The Q-fields for the three agents are shown before and after cooperation in Figures 15 and 16 respectively. Performance plots are shown in Figures 17 and 18 for the cases without and with cooperation respectively.

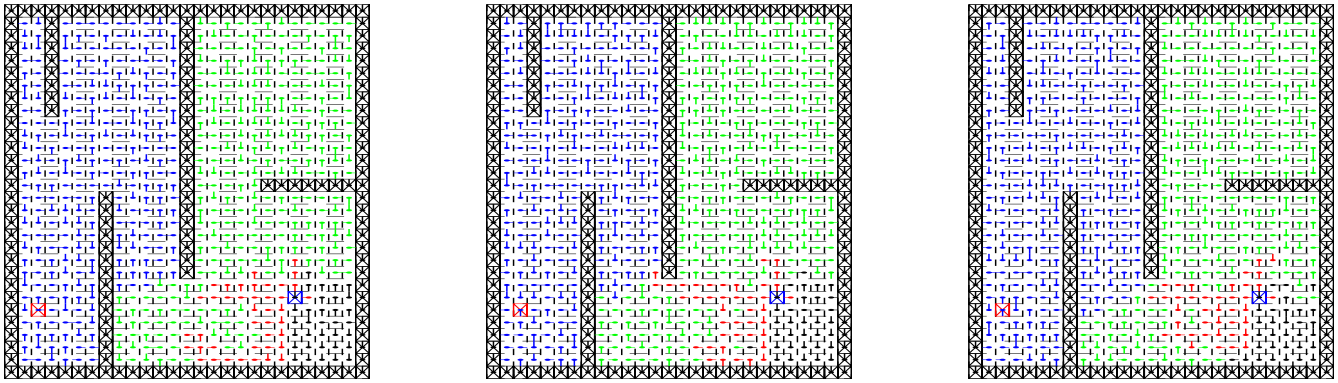

Figure 15: Maze world with equal individual experiences, Q-fields after individual learning phase

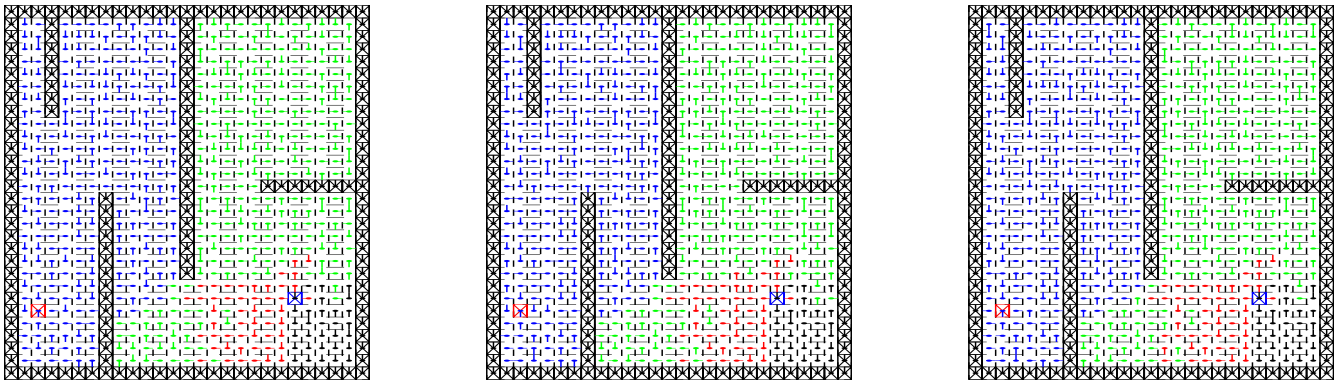

Figure 16: Maze world with equal individual experiences, Q-fields after cooperation phase

7.2.2 Discussion

It is immediately obvious from the performance plots in Figures 17 and 18 that cooperation does not improve performance relative to individual learning in this situation. In both cases, the plots show steady convergence over time, but there is no noticeable change in behavior when the agents cooperate at time $t = 0$ in Figure 18.

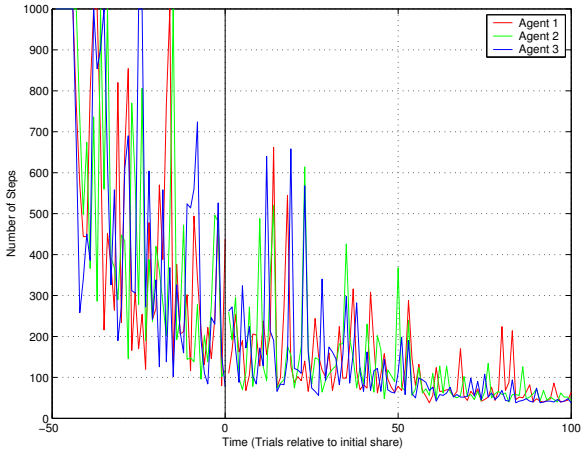

Figure 17: Maze world with equal individual experience, performance without cooperation

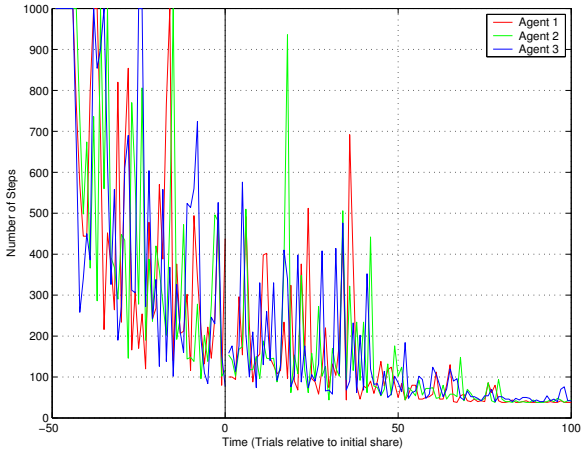

Figure 18: Maze world with equal individual experiences, performance with cooperation

This behavior can be explained with reference to the plots of the Q-fields. In Figure 15, immediately before cooperation, it is clear that each agent has achieved a converged policy in a very similar area around the goal. Therefore, when the agents share their Q-values, they have little new information to offer each other and the Q-fields are almost unchanged, as shown in Figure 16. In the areas where the Q-field appears blue an agent's policy is clearly lacking and this is where cooperation could be of the most benefit, but none of the other agents have a well converged policy in these areas and so have nothing to offer. Indeed, sharing non-converged Q-values causes a slight decrease in performance, as the continuity present within the values held by a single agent is lost, as discussed in Section 6.4.

7.3 General Maze World, Different Experience Case

This simulation addresses the problem where each agent has a very different level of experience. Again, the *maze* world is used, with three agents and a single start and goal location and learning with and without

cooperation is considered. However, in this case, agents 1, 2 and 3 conduct 100, 50 and 10 individual trials respectively, each timed such that the trials are completed at time $t = 0$.

7.3.1 Results

The Q-fields for the three agents are shown before and after cooperation in Figures 19 and 20 respectively. Performance plots are shown in Figures 21 and 22 for the cases without and with cooperation respectively.

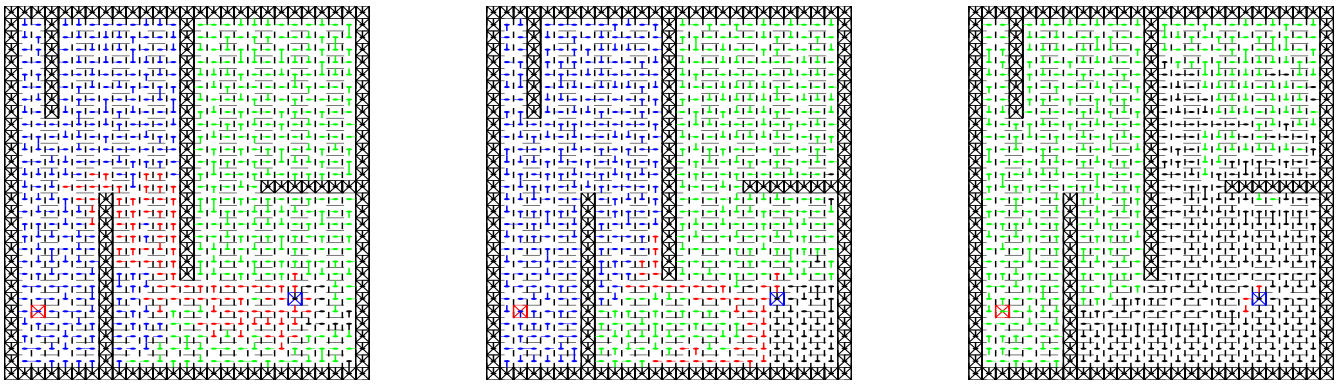

Figure 19: Maze world with different individual experiences, Q-fields after individual learning phase

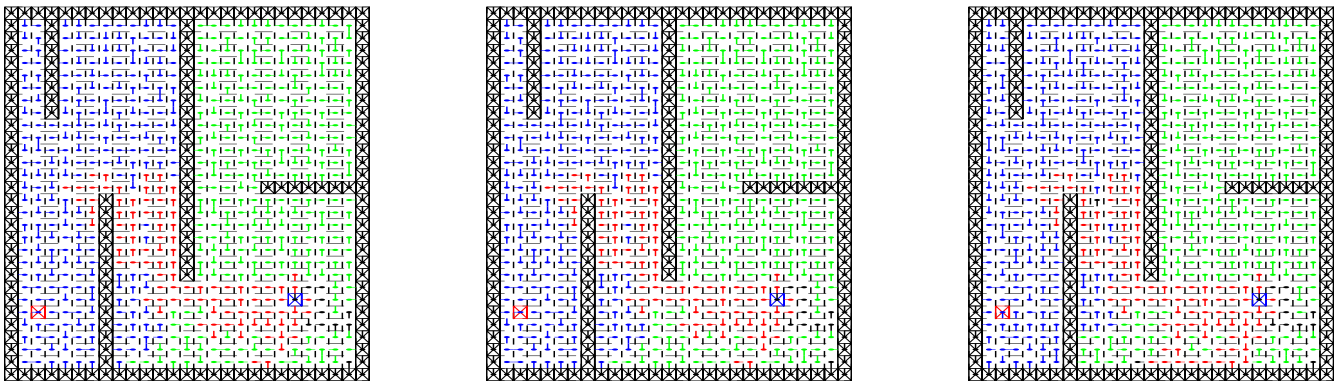

Figure 20: Maze world with different individual experiences, Q-fields after cooperation phase

7.3.2 Discussion

Figure 21 shows that without cooperation, the performance of each agent steadily improves over time as its Q-values converge towards an optimal policy, as we would expect. However, Figure 22 shows that when the agents cooperate at time $t = 0$, agents 2 and 3, which have the least experience at any given point in time and hence the poorest performance, display a sudden increase in performance. Their performance jumps to that of agent 1, the most experienced agent, after which it continues to converge over time. Conversely, the performance of agent 1 shows no noticeable change due to cooperation.

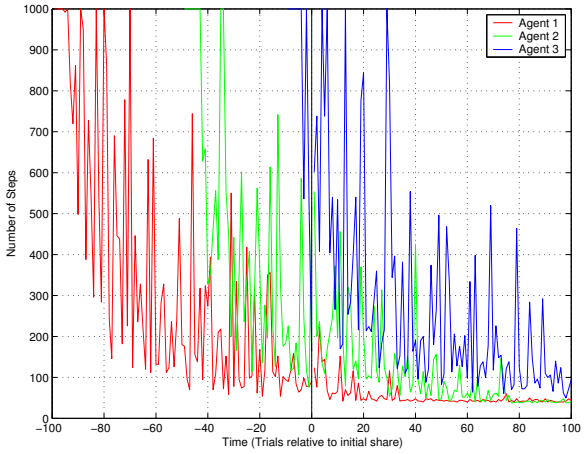

Figure 21: Maze world with different individual experiences, performance without cooperation

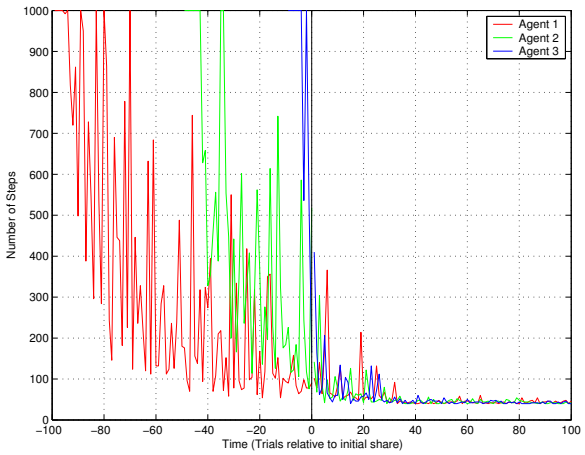

Figure 22: Maze world with different individual experiences, performance with cooperation

Once again, these results can be explained in terms of the Q-fields. In Figure 19, immediately before sharing, the extent of the red zone in each Q-field shows that agent 1 has learnt a converged policy to the goal that extends most of the way back to the start location. The extent of the this converged region is significantly less for agents 2 and 3, simply because they have conducted fewer trials. As a result, agent 1 has a greater expertness value in almost all states in the world and all three agents will weight its Q-values heavily during cooperation. This produces the similar Q-fields after cooperation shown in Figure 20 and means that the three agents will show very similar performance in the testing phase. This performance is very similar to that shown by agent 1 in Figure 21, where cooperation was not used.

7.4 Learning from the Most Expert Agent

Section 7.1 confirmed Eshgh and Ahmadabadi's result that cooperation is beneficial in segmented environments and Sections 7.2 and 7.3 demonstrated that in the case of a more general maze world, cooperation is

only beneficial when the agents have significantly different levels of initial experiences. In both of these cases, in any given state, one agent will be significantly more expert than all of the others, and the expertness could be equally well described using a binary variable. This means that we would expect the choice of weighting assignment mechanism to be largely irrelevant.

To test this intuition, the simulation in the *segmented* world with cooperation presented in Section 7.1 was repeated using a different weighting assignment strategy. This strategy is referred to as *most expert* and is described by the following expression, where a weighting of unity is applied to the data supplied by the agent most expert in the current zone and a weighting of zero to all other agents.

$$W_{i,j,z} = \begin{cases} 1 & e_j = e_{max} \\ 0 & otherwise \end{cases}$$

7.4.1 Results

The Q-fields for the three agents before cooperation are identical to those shown in Figure 7 and the fields after cooperation are shown in Figure 23. The performance plots for the three start locations show very similar information, so for reasons of brevity, only the results for start location 1 are presented and this is shown in Figure 24.

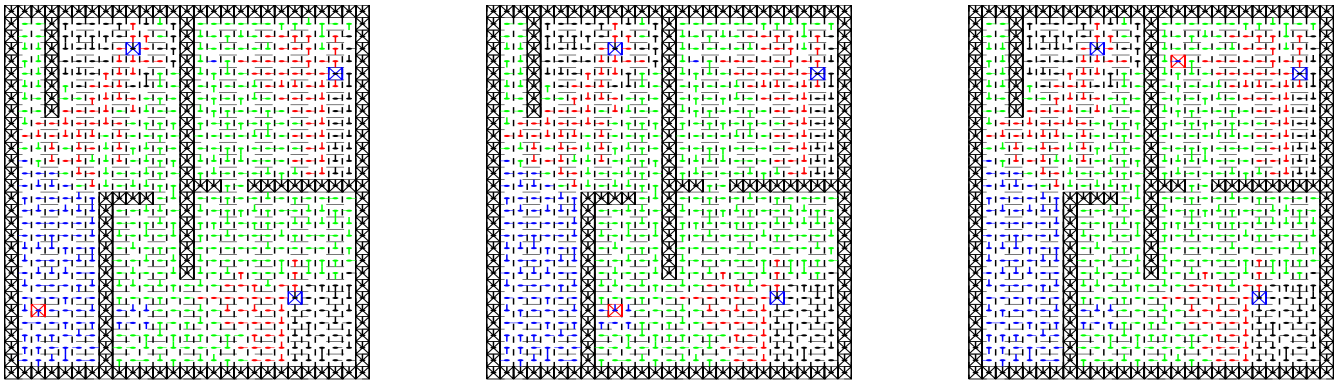

Figure 23: Segmented world with *most expert* weighting strategy, Q-fields after individual learning phase

7.4.2 Discussion

As expected, Figure 24 shows that the performance of the *most expert* strategy is almost identical to that of the *learning from experts* strategy shown in Figure 12. Correspondingly, the Q-field in Figure 23 is very similar to that in Figure 7.

Although the *most expert* method is simpler than *learning from experts*, it leads to homogeneous Q-values amongst the agents after cooperation and this limits the ability of the population of agents as a whole to adapt to changes in the environment[4]. For this reason, it was decided to retain the *learning from experts* method for all future simulations.

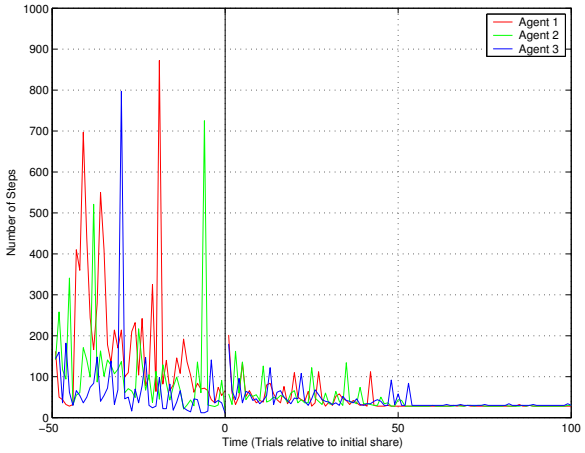

Figure 24: Segmented world with *most expert* weighting strategy, performance from start location 1

8 Dynamic Environments

The results obtained so far indicate that whilst cooperation is not always advantageous, it is never significantly detrimental to the agents' performances, relative to individual learning. One of the objectives of this work is to investigate the effect of dynamic environments on the cooperative Q-learning algorithm and to determine whether this statement remains valid. This results of investigation are presented in the following sections.

8.1 Random Dynamic Maze

The environment used in this investigation was initially identical to the *maze* world shown in Figure 2, but after each trial, every non-obstacle state transitions into an obstacle with probability p. The relevant values of the transition function are updated accordingly and the modified MDP is used in the following trial. The simulation used a single start and goal location and three agents, each of which conducted 50 individual trials starting at time $t = -50$. Cooperation was conducted at time $t = 0$ and the simulation was run for both $p = \frac{1}{10N}$ and $p = \frac{1}{N}$, where N is the number of states in the world, such that on average, a new obstacle appears every 10 and 1 trials respectively.

8.1.1 Results

Performance plots are shown for the two simulations in Figures 25 and 26.

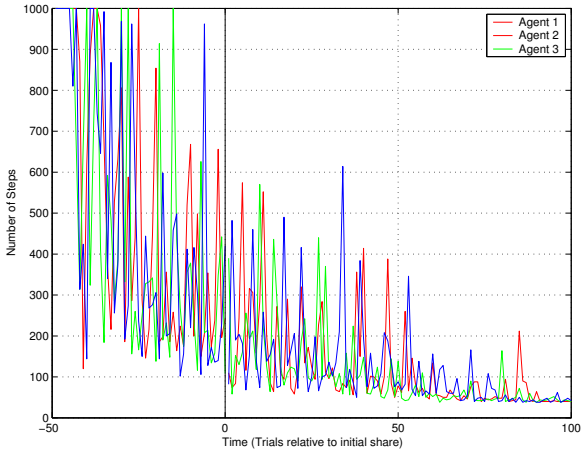

Figure 25: Maze world with random obstacles, performance with $p = \frac{1}{10N}$

8.1.2 Discussion

Figure 25 shows that in the case of an environment which is only modestly dynamic, the performance of the Q-learning algorithm is almost unaffected compared to the case of a static case, shown in Figure 18. Furthermore, it appears that cooperation at time $t = 0$ does not cause a decrease in performance, so cooperation can safely be used in this case. In cases where the environment is significantly dynamic, Figure 26

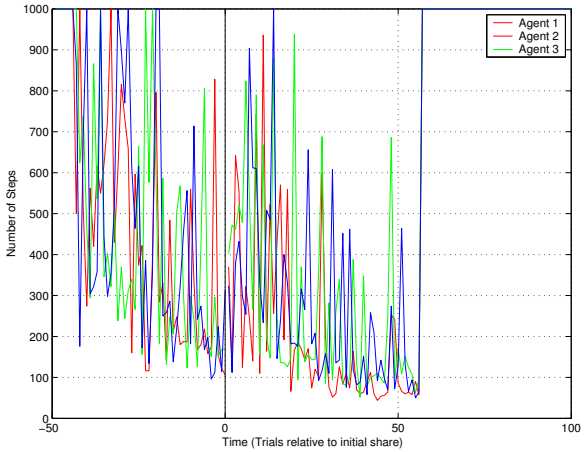

Figure 26: Maze world with random obstacles, performance with $p = \frac{1}{N}$

shows that for $t < 50$, the performance of Q-learning does decrease slightly compared to the static case in Figure 18, but the effect is still not major.

At $t = 55$, the performance of all three agents suddenly drops and the number of steps to reach the goal jumps to the maximum value of 1000. This is because at this point in time, the positions of the new obstacles are such that a feasible path from the start to the goal no longer exists. Considering this, it is remarkable that the performance of all three agents is affected so slightly at trials immediately prior to $t = 55$.

The reason for this robustness to dynamic environments is that Q-learning is well suited to local repair of the Q-field. A policy is learnt for all states in the world, not just those that lie on current best path from start to goal. This means that if an agent encounters an unexpected obstacle, it need only update a very small number of Q-values before the field is re-directed around the obstacle and the agent can continue following the policy dictated by the Q-values of neighboring states. These modifications to the Q-field are made within the first few trials after the obstacle has appeared and hence the algorithm quickly recovers its previous good performance.

8.2 Doors World

The results of Section 8.1 show that Q-learning is in general very robust to dynamic environments. However, it was decided to attempt to simulate a scenario in which a dynamic environment causes a significant drop in performance, with the intent of using cooperation to improve the situation. Such a scenario was constructed using the *doors* world, which is shown in Figure 3 and requires the agent to pass through one of the three constrictions or 'doors' in order to reach the goal from the start. The dynamic aspect of the world is manifested in the fact that obstacles can be added and or removed at each of the constrictions at arbitrary points in time, effectively closing or opening the doors.

For this simulation, each agent conducted 200 individual trials, but these were staggered in time, such that agent 1 starts at $t = -600$, agent 2 at $t = -400$ and agent 3 at $t = -200$. Initially, all three doors are open, but door 1 closes at $t = -400$ and door 2 at $t = -200$, where the doors are numbered from left to right. This means that agent 1 learns while all three doors are open, agent 2 while doors 2 and 3 are open, and agent 3 while only door 3 is open. Furthermore, only door 3 is open during the testing phase.

Under this scheme, each agent conducts its individual learning phase in an effectively static world, and the large number of trials means that each agent will learn an approximately optimal policy for the world at the time at which it was learning. This means that agent 1 will begin the testing phase with a well converged Q-field that directs it through door 1, a policy which is completely misleading in the current world. Similarly, agent 2 will hold a well-converged policy that will direct it through door 2 but this is equally invalid: only the policy learnt by agent 3 will be appropriate during the testing phase.

In regions of the world in which a given agent has relatively little experience, it will incorporate the converged policy of the appropriate peer. In regions of the world common to the three pathways, the three agents will have approximately equal expertness values, since they have the same overall level of experience. This means that their Q-values will be shared with approximately equal weightings and it was postulated that the resulting combined Q-field would lead to very poor performance for all three agents. This includes agent 3, whose valid policy will be corrupted by the information offered by agents 1 and 2.

8.2.1 Results

The Q-fields for the three agents are shown before and after cooperation in Figures 27 and 28 respectively and the performance plot is shown in Figure 29.

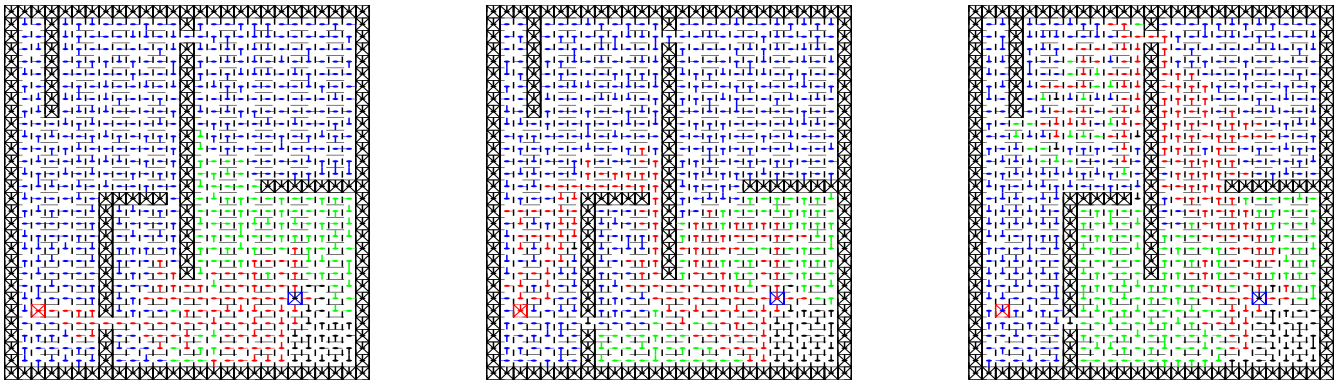

Figure 27: Doors world, Q-fields after individual learning phase

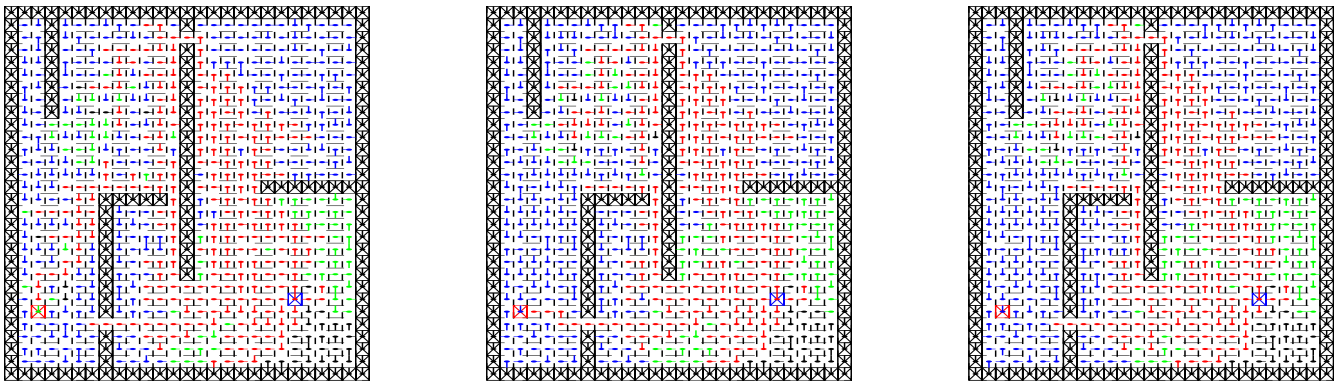

Figure 28: Doors world, Q-fields after cooperation phase

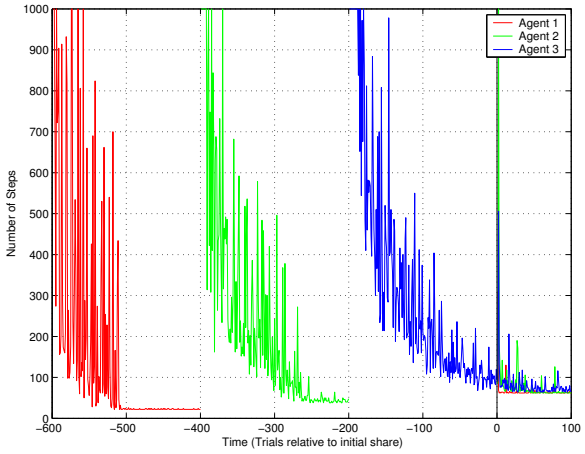

Figure 29: Doors world, performance

8.2.2 Discussion

The Q-fields in Figure 27 show exactly the pattern we would expect, where the red pathways between the start and goal mark the different converged policies learnt by the three agents. The result of sharing is shown in Figure 28 and each agent has clearly incorporated the policy learnt by other agents in areas of the world in which it is relatively inexpert. It is also apparent that in areas of the world in which all three agents have significant experience, such as around the start location, the Q-values have been combined in such a way that converged policy has been lost. This is particularly apparent in the disappearance of the red Q-field around the start held by agents 1 and 2 prior to sharing.

However, the performance plot in Figure 29 shows that after a brief, albeit drastic, decrease in performance after sharing, the performance of the three agents converges extremely rapidly to a very good policy. This is a remarkable result, as agents 1 and 2 enter the cooperation phase with entirely misleading policies leading them through doors 1 and 2 respectively, yet within the first ten trials of the testing phase, they both converged to a near optimal path through door 3.

The reason is exactly that stated in Section 8.1: Q-learning is very well suited to local repair. During the initial trials of the testing phase, each agent will encounter unexpected obstructions at doors 1 and 2, as its corrupted policy will likely lead it in these directions and this explains the spike in the number of trials at this point in time. However, within a small number of trials, the agents have updated their Q-values around doors 1 and 2 and are able to follow the policy supplied by agent 3 that leads them on a near optimal path through door 3 to the goal.

In conclusion, even in this rather artificial environment, Q-learning is capable of such efficient local repair that any decrease in performance due to unforeseen changes in the world is very brief and policy quickly re-converges.

8.3 Contrived World

In both the random and *doors* environments, only a small number of Q-values must be updated on detection of an unknown obstacle before the agent is able to follow well converged policy. Here, a scenario is developed in which the effects of the new obstacle must be propagated through a large number of states before the

agent can follow an alternative route to the goal. The scenario uses the *contrived* world shown in Figure 4, with doors located at both points of entry to the goal region.

Only two agents are used, each of which conducts 100 individual trials, starting at $t = -200$ and $t = -100$. The doors are configured so that only the top door is open during the individual learning phase for agent 1 and only the bottom door is open for agent 2. The bottom door only remains open for the testing phase, such that agent 1 will begin this phase with a well converged, but entirely misleading policy, while agent 2 will hold the correct policy.

8.3.1 Results

The Q-fields for the two agents are shown before and after cooperation in Figures 30 and 31 respectively and the performance plot is shown in Figure 32 (dots rather than lines are used in this case for clarity).

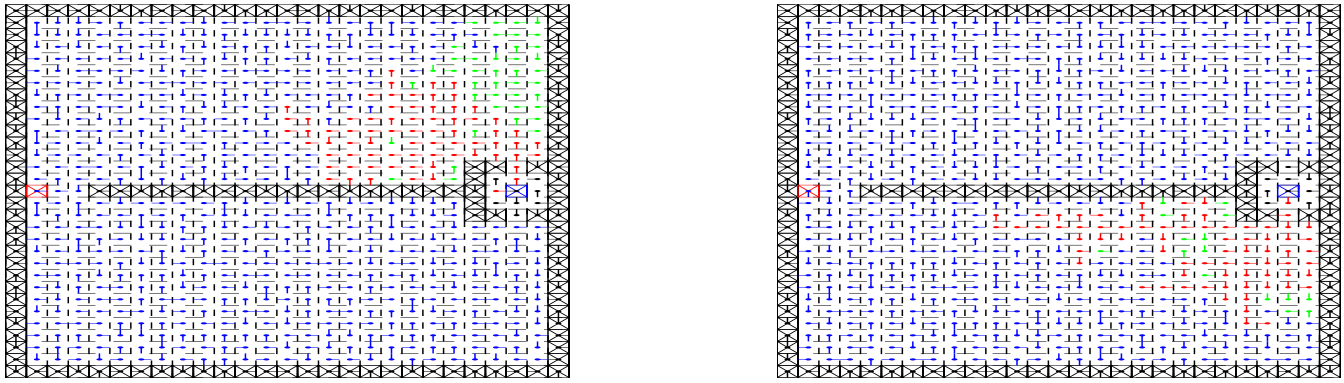

Figure 30: Contrived world, Q-fields after individual learning phase

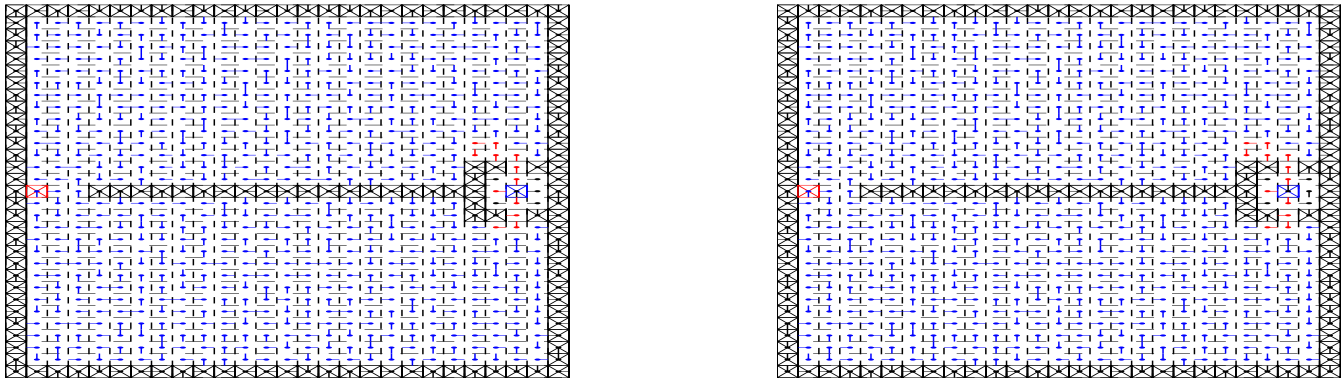

Figure 31: Contrived world, Q-fields after cooperation phase

8.3.2 Discussion

Figure 30 shows exactly the expected patterns in the Q-fields: agent 1 has learnt a fairly well converged policy that leads it through the top door, while agent 2 has learnt a fairly well converged policy that leads

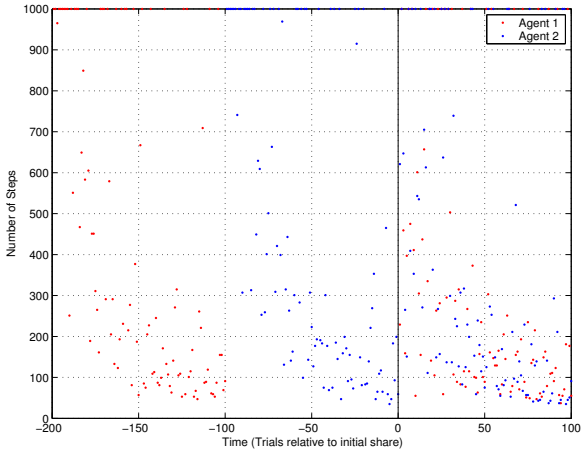

Figure 32: Contrived world, performance

it through the bottom door. After cooperation, however, both agents loose much of this policy information, as shown in Figure 31. The reason is that whilst both agents have learnt a good policy in one half of the world, they have each spent a great deal of time 'lost' in the other half of the world. In doing so, they have accumulated a great deal of negative reward in these states and under the *absolute* expertness measure, this contributes to their expertness, so that each agent has similar expertness in both halves of the world. Therefore, when the Q-values are shared, each agent's data is weighted approximately equally and in each half of the world the good policy provided by one agent is disrupted by the poorly converged Q-values provided by the other.

As we would expect, this means that the performance in the testing phase is very poor. Although Figure 32 shows some improvement in performance over time in this phase, the proportion of trials in which the number of steps reach the upper limit of 1000 is very high. It is important to note that both agents perform equally poorly and while we would expect agent 1 to perform poorly without cooperation, it is clear that cooperation has worsened the performance of agent 2.

In conclusion, in this highly contrived environment, cooperation between the two agents not only failed to improve performance, but significantly worsened the performance of one of the agents compared to its individual performance.

9 Discounted Expertness

The *contrived* world discussed in Section 8.3 caused cooperative Q-learning to fail because, in the states important for forming an optimal policy, both agents had similar expertness values, yet the data held by agent 1 was completely misleading. Although both agents had learnt a reasonably well converged policy for the world in which they had conducted their individual trials, only the policy learnt by agent 2 remained relevant to the world in which the agents were tested after cooperation.

To overcome this problem, the concept of *discounted expertness* was introduced to model the intuitive notion that in a dynamic envionment, information gained from recent observations is more valuable than that gained in the distant past. This is achieved by introducing a time decay with time constant T in each agent's expertness. Considering a trial as the unit of time, coopeartive Q-learning is a discrete-time process with updates occuring at times t_k, so the decay is modeled by the following recursive expression. Note that this update is applied even when the agent is not currently active, to simulate the continual passage of time.

$$e_{i,z}\left(t_{k+1}\right) = e_{i,z}\left(t_k\right) e^{-\frac{1}{T}}$$

The simulation software was modified to include discounted expertness with a time constant $T = 10$ and was used to obtain the following results.

9.1 Doors World

9.1.1 Results

The *doors* scenario presented in Section 8.2 was repeated with discounted expertness. The Q-fields after individual learning and after cooperation are shown in Figures 33 and 34 respectively and the performance plot is shown in Figure 35.

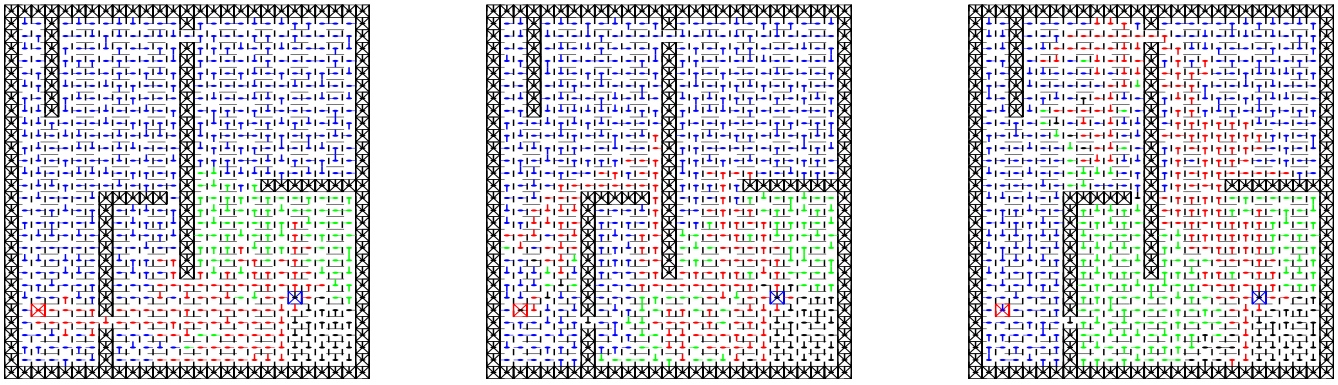

Figure 33: Doors world, Q-fields after individual learning phase with discounted expertness

9.1.2 Discussion

Prior to sharing, the Q-fields shown in Figure 33 clearly demonstrate the different policies learnt by the three agents, corresponding to the paths through each of the three doors, and as would be expected, these

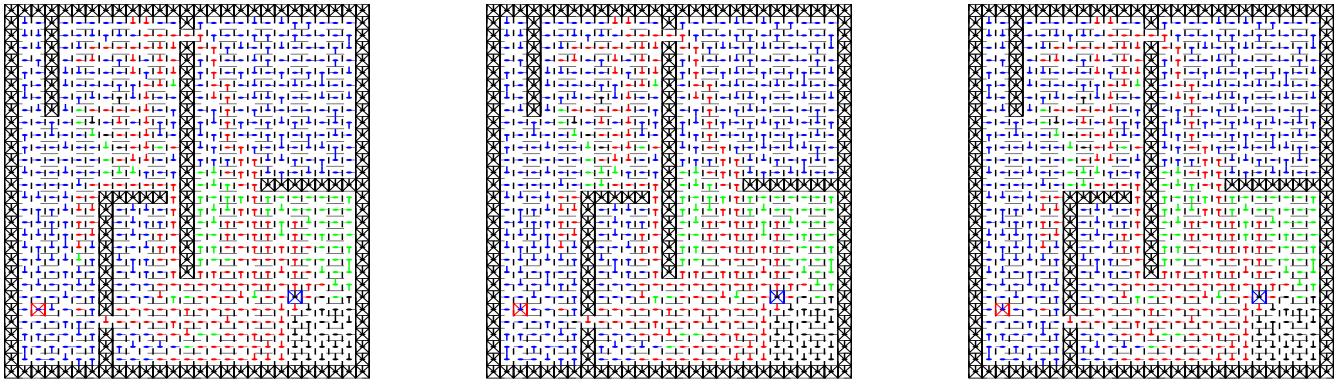

Figure 34: Doors world, Q-fields after cooperation phase with discounted expertness

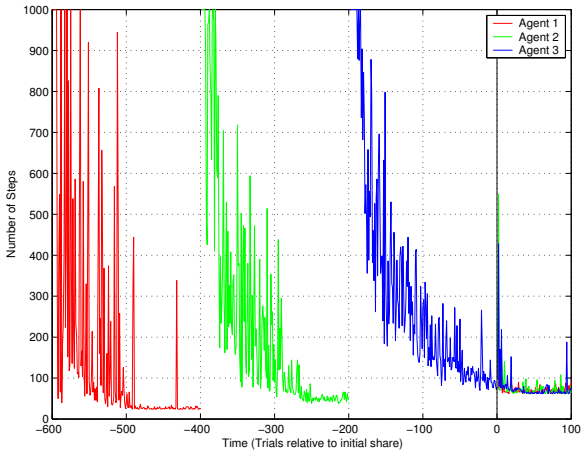

Figure 35: Doors world, performance with discounted expertness

appear very similar to those learnt without discounting shown in Figure 27. Sharing produces the obvious amalgamation of policies shown in Figure 34 and it is not apparent that there are any significant difference between these Q-fields and those shown in Figure 28, where discounting is not used. The performance plot in Figure 35 also appears similar to that in Figure 29 for the simulation without discounting, but there is a subtle difference. Without discounting, all three agents showed a brief but large decrease in performance immediatley after cooperation, but the addition of discounting has reduced the size of these spikes significantly. Agents 1 and 2, for whom the policy learnt from their individual trials is largely irrelevant in the environment used in the testing phase, show the largest decrease in the size of this spike, but agent 3 *also* shows this trend.

In conclusion, the use of discounting has improved slightly the performance of all three agents immediately after cooperation in this particular world. Most importantly, discounting has reduced the extent to which the performance of the most well-informed agent is impaired by cooperation with agents which have learnt a policy which is no longer valid. In all cases however, the period of time over which these improvements take place is very brief and longterm effecrts are negligible.

9.2 Contrived World

9.2.1 Results

The *contrived* scenario of Section 8.3 was also repeated with discounted expertness. The Q-fields after individual learning and after cooperation are shown in Figures 36 and 37 respectively and the performance plot is shown in Figure 38. The blue lines on the superimposed on the Q-fields are expertness contours for the agent.

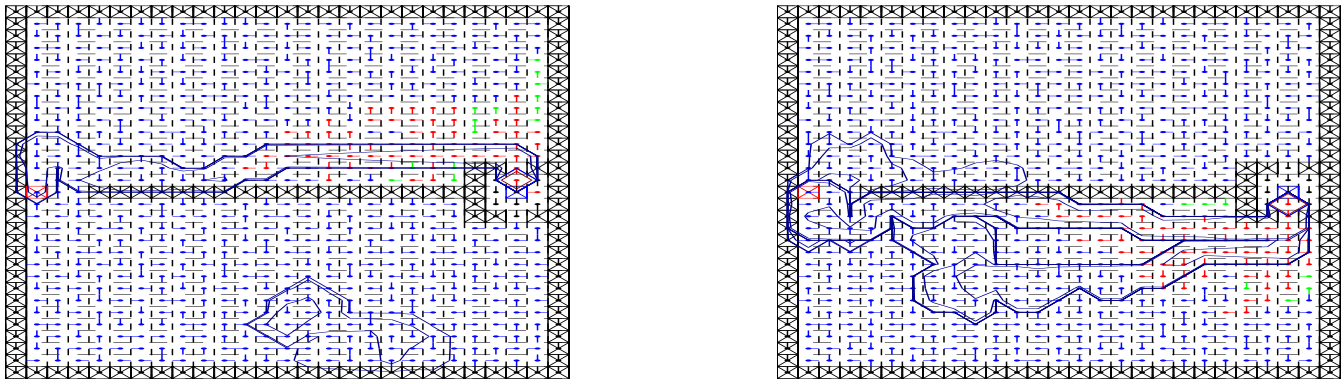

Figure 36: Contrived world, Q-fields after individual learning phase with discounted expertness

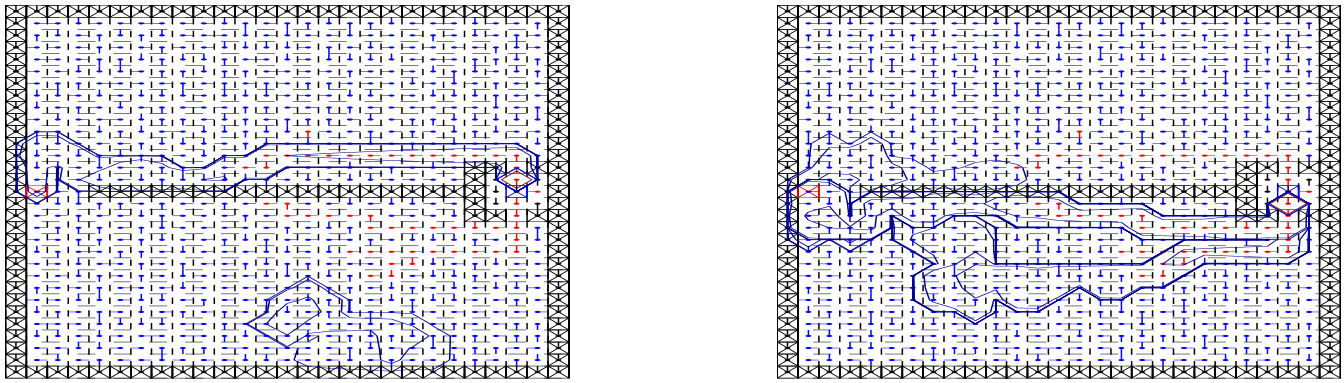

Figure 37: Contrived world, Q-fields after cooperation phase with discounted expertness

9.2.2 Discussion

The Q-fields in Figure 36 show that each agent has learnt a policy in only one side of the world and the patterns are are very similar to those in Figure 30 where discounting is not used, as we would expect. Without discounting, cooperation was shown to obliterate much of the converged policy information learnt by each agent, as represented by the lack of red Q-field in Figure 31. With discounting, however, Figure 37 shows that a much larger proportion of the policy information has been retained.

It is perhaps surprising that policy is retained in both sides of the world, but this can be explained as follows. Expertness for a given zone is calculated as the sum of the absolute magnitudes of the reinforcement signals

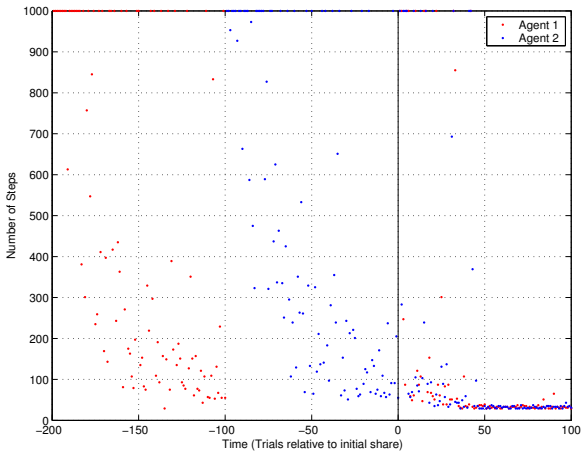

Figure 38: Contrived world, performance with discounted expertness

received in that zone over the lifetime of the agent. Once an agant has learnt a well converged policy, most of its subsequent time will be spent following this policy. In the case of discounted expertness, rewards obtained towards the end of an agent's individual learning phase count most towards the expertness, as these contributions are discounted least in the time between them being received and cooperation at time $t = 0$. Therefore, we would expect an agent to retain significant expertness only in the zones corresponding to its converged policy and this is exactly what is shown by the expertness contours in Figure 36. During cooperation, therefore, each agent is most expert in the region surrounding it's optimal policy and even though agent 1 conducted its individual trials earlier in time than agent 2, it is still the most expert agent in the zones corresponding to its optimal policy. For this reason, the conveged policy of both agents is preserved during cooperation.

During the testing phase, each agent is presented with two policies, one of which is misleading. Although the agent must repair these Q-vlaues in order to avoid taking the incorrect route from the start location, the correct path is also mapped out, so the overall time to reach a converged policy is much reduced compared to the case without discounting. This is confirmed by comparing Figures 38 and 32, which show the performance with and without discounting, respectively. With expertness discounting the rate of convergence is signifiacntly improved and the proportion of trials which reach the maximum number of steps is greatly reduced.

In conclusion, discounted expertness has been shown to offer a solution to a particularly contrived situation in which cooperative Q-learning performs very poorly.

10 Conclusions

Firstly, it should be noted that the nature of Q-learning is such that peformance is often highly problem specific and experiemental results can never be used to prove a result in general. However, a sufficient number of simulations were conducted, using a sufficent variety of paramemter values and problem descriptions, to ensure the conclusions stated in this report are reasonably robust to change.

Expertness based cooperative Q-learning with specialized agents, as presented by Eshgh and Ahmadabadi, gives an improvement in performance relative to an agent acting individually, in only two special cases. The first is when the agents have gained experience through individual learning in different areas of the state space and are then tested in areas in which they have little or no expereince. The second is when the agents have significantly different levels of experience.

In both cases, cooperation takes place between agents with very limited experience of a particular subset of the state space and agents with significantly more expereince in this region. It is therefore rather obvious that cooperation will improve performance in these situations. As would be expected, the less experienced agents are able to make use of the policy obtained by the more experienced agents to improve their performance, thus pulling the performance of the entire popoulation up to the level of the most experienced agents.

Tests showed that cooperation between agents of similar levels of experience is of no benefit and that cooperation never increases the performance of an agent beyond the that of the most experienced agent before cooperation.

In the two cases where the algorithm is effective, the large differences in the agents' expertness values mean that the choice of weighting assignment strategy is largely irrelevant. It was postulated that equally good performance could be obtained by usign a simpler strategy, *most expert*, and simulations confirmed this.

Simulations in dynamic environments showed that Q-learning is remarkably robust to changes in the world, due to its inherent ability to conduct very efficient local repair of the policy encoded in the Q-values. Only in contrived cases was the algorithm adversely affected by dynamic environments, but in these cases, cooperation was shown to reduce the performance of some of the agents.

The concept of *discounted expertness* was introduced to overcome this problem, whereby an agent's expertness decays over time, representing the reduced value of outdated information. This was implemented and shown to avoid the drop in performance in the contrived cases mentioned above. Although discounted expertness does not in general improve performance in dynamic environments, this is nevertheless an important result because it makes the cooperative Q-learning algorithm robust to a wider variety of problems, thereby increasing the confidence with which it can be used in new problems.

Finally, the investigations presented in this paper highlight an issue peculiar to goal-seeking scenarios. In a grid world, an agent is able to move to any adjacent state, so the choice of possible actions is identical at every step (excluding obstacles, which tend to be relatively sparsely distributed). This is in contrast to more complex problems, where only a limited subset of all possible actions is available in most states. Whilst an agent in a complex world may require a complex sequence of actions to return to its current state, thereby following a large loop in the phase space, an agent in a grid world can double-back on itself, so is able to follow loops of zero size. This means that if an agent in a grid world encounters *any* positive reward, its optimal policy will be to continually repeat the action which supplied the reward for all future time. More compex environments prevent this behaviour because of the lengthy series of actions required to return to the current state and repeat the rewarded action. Therefore, goal-seeking tasks must terminate when an agent reaches a goal state and no other positive rewards can be included in the world.

In the case of a goal-seeking problem, optimal policy is learnt by gradually propogating this single reward backwards towards the start location, but the rewards associated with the actions along this path are small

and negative. Therefore, an expertness measure based only on reinforcement signals assigns no greater value to the execution of these actions than it does to the execution of any of the other nominal actions in the world. Consequently, during sharing, no greater emphasis is placed on well converged policy propagated from the goal than is placed on any other actions which have been repeated a similar number of times. It would clearly be of benefit to use the magnitude of the Q-value as an indication of the importance of the local policy, in much the same way as the red areas of the Q-fields were used to track the debelopment of converged policy in the experimental results presented in this paper. This could lead to significant performance improvements and is a possible avenue for future work.

References

[1] *Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents*
Tan, M. 1993.

[2] *A Complexity Analysis of Cooperative Mechanisms in Reinforcement Learning*
Whitehead, S. D.
Proceedings of AAAI-91. 1991.

[3] *Expertness Based Cooperative Q-Learning*
Ahmadabadi, M. N. and Asadpour, M.
IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics, Vol. 32, No. 1 February 2002

[4] *An Extension of Weighted Strategy Sharing in Cooperative Q-Learning for Specialized Agents*
Eshgh, A. M. and Ahmadabadi, M. N.
Proceedings of the 9th International Conference on Neural Information Processing (ICONIP02), Vol. 1. 2002.