A Hierarchical XCS for Long Path Environments

Dr Alwyn Barry
University of the West of England,
Coldharbour Lane, Bristol, BS16 1QY, UK

Email: alwyn.barry@uwe.ac.uk
Phone: (++44) 344 3135

Abstract

It has been noted (Lanzi, 1997, Butz et al, 2000) that XCS (Wilson, 1998) is unable to identify an adequate solution to the Maze14 problem (Cliff and Ross, 1994) without the introduction of alternative exploration strategies. The simple expedient of allowing exploration to start at any position in the Maze will allow XCS to learn in such 'difficult' environments (Barry, 2000b), and Lanzi (1997) has demonstrated that his 'teletransportation' mechanism achieves similar results. However, these approaches are in truth a re-formulation of the problem. In many 'real' robotic learning tasks there are no opportunities available to 'leapfrog' to a new state. This paper describes an initial investigation of the use of a pre-specified hierarchical XCS architecture. It is shown that the use of internal rewards allows XCS to learn optimal local routes to each internal reward, and that a higher-level XCS can select over internal sub-goal states to find the optimum route across sub-goals to a global reward. It is hypothesised that the method can be expanded to operate within larger environments, and that an emergent approach using similar techniques is also possible.

1 INTRODUCTION

Within their investigation of the introduction of a memory mechanism to ZCS, Cliff and Ross (1994) introduced the Maze 14 environment. This Markovian environment provides a length 18 corridor path constructed using the Woods-1 inputs, providing a non-linear action route to the reward position. The length of the pathway which the environment provides is itself a challenge to current Bucket-Brigade algorithms. For example, Riolo (1987) estimated that for CFSC within single action corridor environments the number of times payoff must pass through the rule-chain to achieve 90% of stable strength is $R = 22 + 11.9n$[1]. In such an environment using pure random exploration the probability of exploring state s_n from state $s_{n\text{-}1}$ is always P(1.0). However, within the Maze14 environment the probability is P(0.125). Clearly this probability remains the same within each successive state, and thus, the probability of moving directly from s_0 to s_{18} in 18 steps is P(5.55×10^{-17}). It is therefore highly improbable that ZCS will move from s_0 to s_{18} within exploration even where a large number of iterations [in bucket-brigade terms] is permitted in each trial. Furthermore, transitions from s_n to s_n will be explored with P(0.875), resulting in a disproportionate exploration of the early states within this environment.

The problem of exploration within the Maze14 environment can be overcome by a number of mechanisms. The first involves a change in the environment definition itself to allow all of states s_0 to s_{17} to be start states. This means that exploration starting in the later states will have a higher chance of reaching the reward state to feed back stable reward. This reward will be passed down the local states of the chain which when subsequently explored will further propagate these values. XCS (Wilson, 1998) allows prediction learning within exploitation so that the pathway to the reward state will rapidly become established through exploitation once discovered. Lanzi's 'teletransportation' mechanism (Lanzi, 1997) is an example of the use of this approach.

An alternative approach that does not require a change in the environment definition would be to dynamically modify the division between exploration and exploitation so that as transitions within the FSW are increasingly explored their probability of future exploration is decreased. This allows the LCS to advance progressively further through the environment to areas that require exploration. Lanzi (1997) notes that Wilson has used this approach within Maze-14, although no results have been published to date.

It is possible that a third approach to this problem exists if the LCS representation can be expanded to allow a hierarchy of classifiers to be utilised. This paper elaborates a hypothesis for the use of a hierarchy for the solution of such problems and investigates the use of the approach identified within a number of corridor environments. It provides a number of new contributions to LCS research. It demonstrates that a simple solution to the problem of learning to traverse long action chains

[1] Barry (2000b) has shown that XCS is able to learn the optimal solution to the GREF-1 environment much more rapidly than CFSC.

within simple progressive corridor environments exists. It then shows that the addition of hierarchical control will allow this solution to be applied to a more complex environment. This expansion represents the first application of hierarchy within XCS, and the methods are contrasted with those of Dorigo and Schnepf (1993) and Dorigo and Colombetti (1994) with ALECSYS. The approaches are also related to two methods for hierarchical learning within Reinforcement Learning, demonstrating that techniques within Reinforcement Learning can be utilised beneficially within LCS.

2 LOCALISATION OF REWARD

In resolving the difficulties involved with the learning of Maze 14 an understanding of the core problem of the learning task is essential. Section 1 highlighted the inability of exploration to move the animat controlled by XCS from the early states towards the later states in this environment due to the properties of the environment itself. As a result, there is little opportunity to identify the reward state or to feed the payoff back to earlier classifiers. Equally, the inability to distinguish between classifiers in earlier states due to non-availability of reward information prevents consolidation of reward feedback as a result of learning within exploitation.

This dilema can be partially solved if the LCS was able to introduce intermediate rewards in addition to that provided by the environment. Consider a state s_i which is i steps from the start state s_0. If a classifier leading to this state from s_{i-1} received a fixed 'internal' reward R_I, this reward could be fed back to preceeding classifiers. Thus, the LCS would be able to establish classifiers leading to state s_i even though the ultimate goal state had not yet been encountered. Now consider a set of states s such that $s \subset S \wedge \forall s_\mathrm{i} \in S \cdot \neg\exists s_\mathrm{j} \in S \cdot j = i + 1$ where $S = \{s_0 \ldots s_\mathrm{n}\}$). If each of the states within s was a state providing an internal reward, the states within s represent a chain of intermediate goals towards which XCS can learn a route.

The provision of these internal reward states is not in itself sufficient to enable XCS to find a path to the solution. XCS must, in addition, be able to identify which of the internal reward states to move towards next, and equally must be able to decide not to re-visit an internal state that has already been visited. It is possible to consider an XCS implementation in which the state space is subdivided and for each sub-division one of the internal 'sub-goal' states is advocated and the internal reward is paid out when the Animat controlled by XCS enters this state. For the situation where there is only ever a single sub-goal per environment subdivision the Optimality Hypothesis (Kovacs, 1996) implies that XCS will learn the optimal state × action × payoff mapping for each environmental subdivision. The problem of learning a route from the start state to the reward state is thus decomposed into the problem of moving from one internal reward state to another internal reward state.

Hypothesis 1

Using a prior identification of internal goal states and subdivision of the state-space in relation to the goal states, XCS is able to learn the optimum state × action × payoff mapping for each subdivision, and given a mechanism to determine the sequence of internal goals an optimum path to a global goal can be constructed.

Limiting the environment to a single sub-goal per environmental subdivision limits this mechanism to unidirectional corridor environments. Clearly this is an undesirable limitation. The limitation may be overcome by identifying more than one goal state within each subdivision. Providing the conditions for the classifiers within the XCS are constructed to identify both the current goal and the current local state, it is hypothesised that the Optimality Hypothesis can be extended so that the populations covering each state-space decomposition will be able to identify its optimal state × sub-goal × action × payoff mapping for that part of the state-space.

Hypothesis 2

Where more than one sub-goal state exists within a state-space subdivision and the desired sub-goal is made available through the input mechanism, XCS is able to learn the optimum state × sub-goal × action × payoff mapping for each subdivision, and given a mechanism to determine the sequence of internal goals a sequence of optimum local routes to a global goal can be constructed.

The mechanism for the selection of the current 'goal' states or the relevant XCS sub-population has not been discussed thus far. For this investigation it is proposed that the method of requiring the user to identify the state subdivisions and their "terminal states" used within many Reinforcement Learning approaches (such as Diettrich, 2000; Parr and Russell, 1997) is adopted. As such, the structures used are *fixed* rather than *emergent*. Given this input it is hypothesised that an additional high-level XCS can be added that operates over the space of internal states, treating the lower XCS sub-populations as "macro-actions" that move from the current state to the chosen sub-goal state. Given the current input (which will be one of the sub-goal states) the high-level XCS will select a new sub-goal state and a low-level XCS population to reach that sub-goal. Upon reaching the subgoal state this lower-level XCS will be rewarded the internal reward and will hand control back to the high-level XCS. When the environmental reward state is reached, the high-level XCS will receive the environment reward and through the normal payoff mechanism it is hypothesised that it will learn the optimal state × next sub-goal × payoff mapping.

Hypothesis 3

An XCS can be employed to learn the optimum sequence of sub-goals from a pre-defined set of sub-goal states to reach a reward state within an FSW by the invocation of low-level XCS populations each mapping a unique sub-division of the state-space.

This fixed hierarchical structure using pre-defined sub-goal states does not represent a fully emergent solution.

However, the demonstration of the ability of XCS to operate within these structures and retain the advantages inherent within XCS at each level will pave the way towards further work leading to truly emergent hierarchical XCS formulations. In addition, a demonstration of the validity of the hypotheses provides new solutions to the problem of learning within environments requiring long action chains, and opens the possibility of re-using learnt mappings within more than one area of the state space.

3 EXPERIMENTAL APPROACH

The XCS implementation used within this work is XCSC (Barry, 2000b). In the experimental investigation of the hypotheses the operation of the base implementation will be changed as little as possible to achieve the required structured XCS architectures. In order to maintain comparability with previous work on action chain length (Barry, 2000b, 2001) the parameterisation used within these experiments is as follows: N=400, p_1=10.0, ε_1=0.01, f_1=0.01, R=1000, γ=0.71, β=0.2, ε_0=0.01, α=0.1, θ=25, χ=0.8, μ=0.04, P(#)=0.33, s=20 (see Kovacs (1996) for a parameter glossary). The Finite State World (Riolo, 1987; Barry, 1999) environment used is depicted in figure 1.

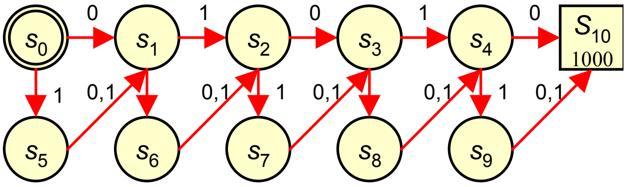

Figure 1 - An extensible corridor test environment suitable for testing action-chain learning within XCS.

This environment has the following useful features:

- It can be trivially extended by small or large increments as longer test action chains are required;
- It includes a choice of route at each state so that the ability of XCS to decide the optimal route as the action chain increases can be determined;
- The sub-optimal route does not prevent progress towards the reward state;
- The optimal route is always re-joined to limit the penalty of a sub-optimal choice;
- The stable payoff received for a sub-optimal choice will always be equivalent to the γ discount of the payoff received for the optimal choice;
- The alternation of actions prevents generalization from prematurely producing very general classifiers that cover much of the optimal path to reward;
- The small number of separate actions limits exploration complexity.
- The environment can be sub-divided into sections each of which has a single identifiable sub-goal state.

The unsuitability of some of the 'standard' XCS performance measures for multiple-step environments has been rehearsed elsewhere (Barry, 1999, 2000b). The System Relative Error (Barry, 1999) is therefore adopted as the standard performance metric and the coverage table (Barry 2000a) is used to identify the level of convergence on the optimal classifier for each action set. As hierarchical structures are introduced the validity of some of the performance measures previously used within XCS become strained. Where appropriate, therefore, these measures are then applied locally and reported separately for each sub-population, and other means are introduced to provide a measure of global performance. Such changes are identified alongside the experimental investigations as they are required.

4 SUB-DIVIDING THE POPULATION

In order to investigate Hypothesis 1 a simple structuring of the population space within XCS was devised. The approach taken is based on the methods used within HQ learning (Wiering and Schmidhuber, 1996), although greatly simplified. The developed XCS formulation is therefore known as a *Simple* H-XCS (SH-XCS),. The standard XCS implementation was modified so that an array of populations is maintained rather than a single population and a variable is added to reference the current population. The environmental interface is modified so that a set of states from the environment can be identified as internal reward states (to become the sub-goals) and operations to allow XCS to detect when an internal state has been reached are provided. The environment is also modified to allow the provision of an internal reward value for reaching an internal state, again with operations that allow XCS to obtain that reward value.

The operation of XCSC is modified so that at the start of each trial the first sub-goal is identified and the current population is set to the first population. XCS then runs as normal within this population until the environment identifies that an internal sub-goal or global goal has been reached. On reaching an internal sub-goal, the state is compared with the desired internal goal and if it is the same the internal reward is provided, the current population variable is moved on to the next sub-population, and the next sub-goal is identified. Upon reaching the global goal the same internal reward is provided to the current sub-population and the global reward is discarded. Thus far the modifications can all be related to features found within a HQ implementation. The simplification comes in the selection of the current sub-goal. Within HQ learning this is performed by an additional HQ table associated with each Q-table - the HQ table uses the current global goal to select a local sub-goal for the local table. The modifications to XCS are concerned with verifying Hypothesis 1, and this does not require a higher-level choice of sub-goal. Therefore the choice of the next sub-goal is deterministic and is simply the next sub-goal in the available sub-goal list. Thus each sub-population learns the optimal path to one sub-goal. SH-XCS was tested by running both XCS and SH-XCS starting with the same random seed in the length-10 (20 state) version of the test environment. It was found that SH-XCS with a single sub-population and a single sub-goal at the global goal produced identical performance plots to XCS for all the standard results collected, confirming that the modifications had not changed the normal operation of XCS.

4.1 INVESTIGATING HYPOTHESIS 1

SH-XCS was now applied to the length-10 environment using two sub-populations. The sub-goal states were s_5 and s_{20}, with the goal state also s_{20}. The condition size was set to 6 bits. A total population size of 800 was used, divided equally between the two sub-populations. The internal reward value was set to 600, a value chosen because of the known reduction in confusion that its discounted values cause when the main reward is 1000 (Barry, 1999b) (although not an issue at this stage). Ten runs of SH-XCS were performed with up to 100 iterations per trial. The performance of the whole learning system was captured using the System Relative Error metric. The population size measure was modified so that it captured the size of each sub-population rather than providing a single result. This gives a means of tracking the comparative rate of learning in terms of the focus of the sub-populations on their optimal sub-populations [O].

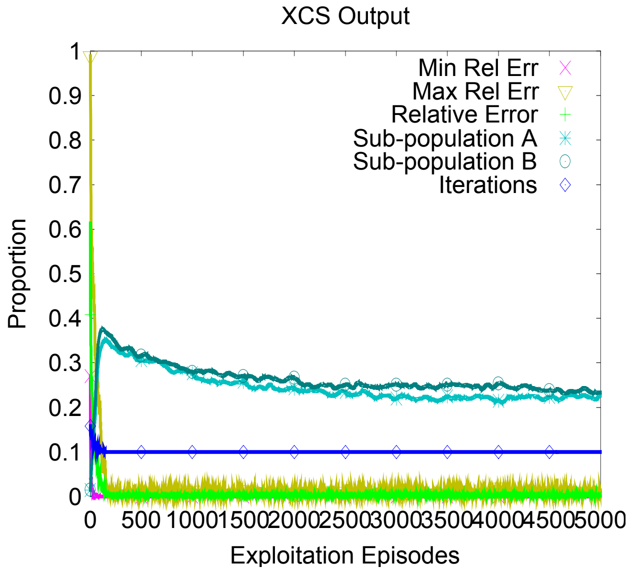

Figure 2 - The performance of the sub-populations of SH-XCS within a length 10 two-choice progressive corridor FSW

Figure 2 pictures the performance of SH-XCS in this experiment. SH-XCS rapidly converged on solutions for each of the sub-populations, and the System Relative Error of the two sub-populations was rapidly eliminated. The population curves show that each population has converged onto a solution and at 5000 exploitation episodes continue to consolidate on their respective [O]. A comparison of these results with those obtained from XCS in the same environment indicated that SH-XCS learns the optimal number of steps in which to traverse this environment in the same length of time as that taken by XCS within an equivalent environment of length 5. Thus, the two sub-populations operating within their own length 5 portion of the length 10 environment are able to establish a solution to their state-space in the same time as a single XCS in an equivalent length 5 environment. It is important to note that the environments tackled by XCS in the length 5 test and the environments tackled by XCS in the two length 5 sub-divisions of the state space within this test are not the same. The state encoding for the length 10 environment was not changed when it was sub-divided so that the advantages that might be gained by a reduced input space (Diettrich, 2000) result from the reduced size of the search space only. Thus, the generalisation task undertaken in each of the sub-populations was different and leads to different [O] within each sub-population.

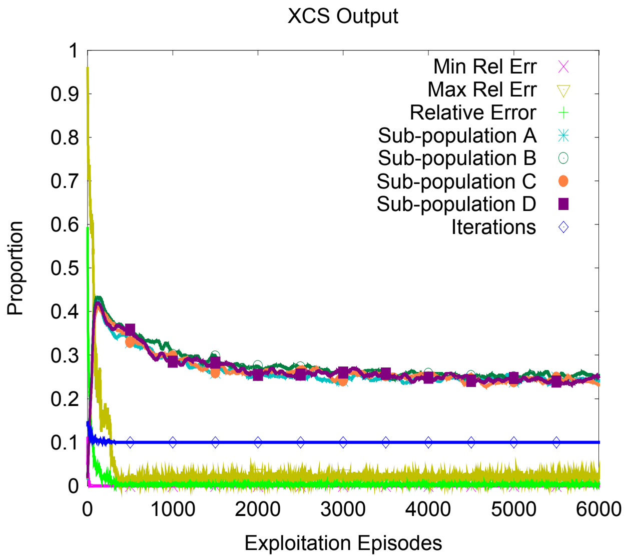

Figure 3 - The performance of four sub-populations of SH-XCS within a length 20 two-choice progressive corridor FSW.

The test environment was now extended to length 20, a length that Barry (2000b) demonstrated could not be adequately learnt using the standard XCS. Four sub-populations were provided, each of length 5 as in the previous experiment. Sub-goal states were 5, 10, 15 and 40, with 40 also providing the goal state. The parameterisation was kept constant apart from the total population size, which was increased to 1600 (divided between each sub-population). Figure 3 gives the averaged performance within the first 6000 exploitation trials from ten runs of 15000 exploitation trials. The similarity of the results presented with those in the length 10 environment is striking. Even though the bit length of the message was increased from six bits to seven and the distance between the decimal value of the messages from the 'optimal route' states and the messages from the 'sub-optimal route' states has increased the learning rate within each sub-population has changed little. This bears out Wilson's hypothesis (Wilson, 1998) that the difficulty experienced by XCS in finding [O] scales with generalisation difficulty rather than bit length. This is particularly relevant for the development of hierarchical approaches using XCS, since the requirement to physically reduce the input size for each sub-population in order to see beneficial performance improvements within Diettrich's MaxQ approach (Diettrich, 2000) may not apply to hierarchical XCS solutions in the same way. Other experiments (see Barry, 2000b) revealed that performance improvements can in fact be gained by utilising input optimisations. This is logical, since a reduction in the message size should require a smaller population to learn the generalisations and should therefore produce performance improvements.

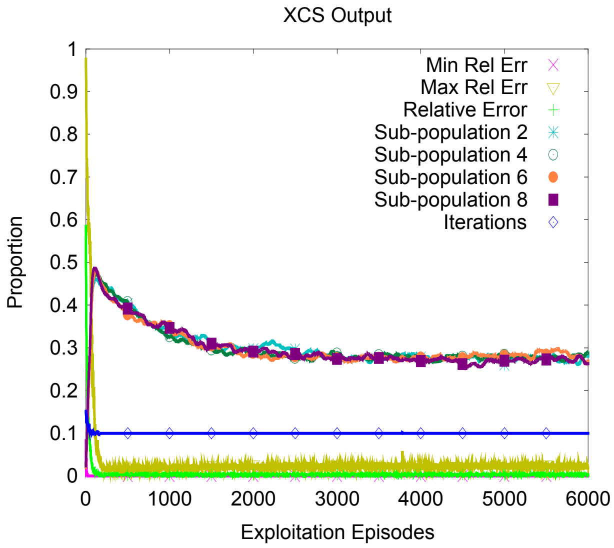

Figure 4 - The performance of eight sub-populations of SH-XCS within a length 40 two-choice progressive corridor FSW.

An analysis of the coverage tables produced from these runs revealed that each sub-population had learnt and proliferated [O] and that [O] was dominant to approximately the same degree as SH-XCS with two sub-populations within the length 10 environment. This is unsurprising, since the learning problem for each sub-population has only been modified in terms of the generalisations to be formed and not in terms of the size or structure of the underlying state-space. Given this finding, the SH-XCS approach would be expected to continue to scale to larger environments penalised only slightly by the additional number of bits required to encode the enlarged state space. To evaluate this claim the environment was increased once more to provide a length 40 action chain to the reward. Eight sub-populations were provided and the sub-goal states were 5, 10, 15, 20, 25, 30, 35, and 80. The condition size was 8 bits and the total population size was 3200.

As expected, the SH-XCS implementation continued to learn rapidly how to traverse this extended environment (figure 4), and once more each sub-population developed a dominant [O]. Given these results, it can be concluded that the use of a prior identification of internal goal states and the subdivision of the state-space in relation to the goal states allowed XCS to learn the optimum state × action × payoff mapping for each subdivision of the state space. Similarly it is concluded that this capability can be used to construct a set of an optimal local paths to a global goal. Therefore Hypothesis 1 is upheld.

4.2 HIERARCHICAL CONTROL

Whilst Hypothesis 2 could be investigated by extending SH-XCS to provide each population with a deterministic sub-goal identification mechanism, it was decided to examine hypotheses 2 and 3 using one mechanism. The Feudal Q-Learning approach to reinforcement learning (Dayan and Hinton, 1993) is a simple approach to hierarchy construction that requires a pre-identified sub-division of the state space into small Q-tables and a pre-selected hierarchy of Q-tables. A Q-table at level *n* in the hierarchy would learn the optimal choice of Q-table from the sub-division of Q-tables at the level *n + 1*. Thus, at the top of the hierarchy a single Q-table would exist, and an inverted tree of successive levels of hierarchy would be constructed until the lowest level of Q-tables operated over a choice of actions on the environment rather than a choice of Q-tables. Each Q-table in levels above the lowest acted like a feudal Lord - they had oversight of a distinct sub-space within the environment and they decided the sub-goal that a selected lower-level Q-table would have to seek to achieve.

The Feudal Hierarchy approach is very close to the form of hierarchical control that hypotheses 2 and 3 pre-suppose. It is therefore appropriate to seek to apply this form of hierarchy within XCS as a natural extension to the previous work with SH-XCS. Rather than implement this "*Feudal XCS*" as a hierarchy of populations a simpler implementation strategy was chosen. It was recognised that if an upper level *n* XCS selects a sub-population and chooses a sub-goal at the next level down (*n-1*) then the set of lower populations and their sub-goals can be seen as the environment that the level *n* XCS is operating upon. If the level *n-1* sub-populations were themselves instances of XCS, then the choice of a sub-population can be viewed as *invoking* a lower XCS to run an episode that seeks to reach the specified sub-goal.

The standard XCSC implementation was therefore modified so that the environment for any level XCS above the base level was an XCS instance. To invoke the lower XCS an upper XCS would write the selected sub-goal into the environment of the lower XCS and then invoke a trial of the lower XCS. Whilst all levels of the Feudal Hierarchy are given input from the current environmental state, levels lower than the uppermost level will also have the sub-goal state identified in their input message. It was recognised that a full specification of the sub-goal within the message would double the message size of the lower XCS and that only a small number of states covered by the lower level XCS would be used as potential sub-goals. Therefore the sub-goals within each environment subdivision were identified within a user-supplied table and the sub-goal choice and message are constructed from the index into that table.

Upon invocation, the lower level XCS will use its population to find the best route to the sub-goal selected by the upper XCS. If the current state is outside the area covered by the selected lower XCS then it will immediately return without any further action (or payoff) so that the discounted payoff mechanism identifies the selection of that sub-population as a "*null action*". Otherwise the XCS uses its population to identify (and learn) the optimal route to the chosen sub-goal. During operation of the lower XCS any action that would cause movement out of the state subdivision covered by that XCS is prevented so that each state space decomposition is treated as though it were the only state-space for that XCS. If the sub-goal is achieved within the number of steps allowed for a trial of the XCS an internal reward value is given to the XCS. If the sub-goal is not achieved no reward (or penalty) is given - the temporal difference update will identify the route as sub-optimal without penalty. At the end of a trial control is handed back to the upper level XCS without reward. The uppermost XCS is the only XCS to receive environmental reward, and will use temporal difference to learn the optimal choice of sub-populations and sub-goals from this payoff. Each trial of XCS at any level is an unaltered XCS trial, including normal induction algorithms. However, the explore-exploit choice is specified by the uppermost XCS.

The capture of integrated reports for even the simple two-level hierarchies used within this investigation is problematic - the learning rates of the two levels are different and invocation of each sub-population will occur at different rates within any non-trivial environment. Therefore each sub-population produces separate reports and these results are gathered for presentation as appropriate to the experiment.

4.2.1 Feudal XCS in a unidirectional Environment

The Feudal XCS was created and after appropriate testing was applied to the same length ten environment used within section 4.1 so that comparative performance data could be gained. The length 10 environment was subdivided into two length 5 environments to correspond to the decomposition within section 4.1. State 5 was designated as the sub-goal for the first subdivision and the reward state, state 20, was the sub-goal for the second. Since the aim of the Feudal XCS is to allow an upper level XCS to prescribe not only the sub-population to use but also the sub-goal to move towards, for each sub-division two sub-goals were specified although they both referenced the same sub-goal state. The message for the top-level XCS consisted of the current state, with its output specifying the lower level XCS to use (1 bit) and the sub-goal to select (1 bit). The message for the lower level XCS instances consisted of the current state and the sub-goal specified by the upper level (1 bit). The action consisted of the direct environmental action (1 bit). Experiments demonstrated that Feudal XCS was able to learn the optimal selection of the lower XCS within this environment, and so the experiment was extended to a length 20 environment requiring four lower level XCS instances. As figure 5 illustrates, Feudal XCS was able to concurrently learn the optimal choice of sub-population and the optimal route to the sub-goal state. The dominance of [O] was good in each XCS population.

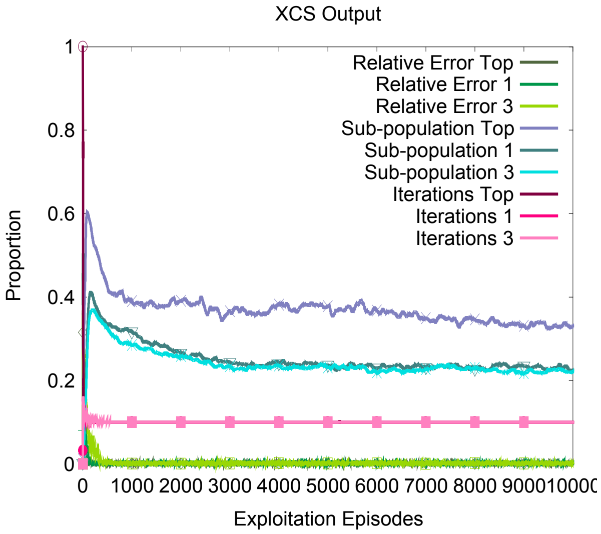

Figure 5 – Feudal XCS in length 20 unidirectional environment

4.2.2 Feudal XCS in a two subgoal environment

Having demonstrated that Feudal XCS is able to select the optimal sub-XCS and then find the optimal local pathway attention was turned to the ability of Feudal XCS to operate within an environment where each state-space sub-division identified two sub-goals at different locations. A suitable environment is pictured in figure 6.

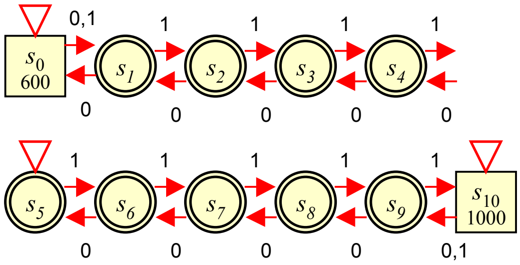

Figure 6 - A corridor environment with two sub-goals in each of two state-space sub-divisions

The state-space was divided into two, with states s_0 to s_5 within the first sub-division and states s_5 to s_{10} within the other. The sub-goals identified were states s_0 and s_5 in the first subdivision and states s_5 and s_{10} within the second. In this environment the upper XCS must learn both the optimal sub-goal and which lower level XCS to select from in any state, and the order of choice of the lower XCS instances and sub-goals required in order to maximise payoff from the two payoff sources. Through a number of pre-experimental runs it was found that the optimal population size for both the upper and lower XCS instances was 400. The condition size for the top population as set to four bits, with a two bit action (bit 0 = sub-population bit 1 = sub-goal). The condition size of the bottom populations was set to five bits - four for the current state and one for the desired sub-goal. The action size remained 1 bit for the selection of the environmental action. An examination of the performance of the sub-populations under exploration also revealed that the limit of 50 steps within a population led to the sub-populations on occasions not achieving their sub-goal. This had the effect of introducing fluctuating payoff to the upper XCS, preventing a full reduction in System Relative Error. This was rectified by allowing the lower XCS to continue until a subgoal state was discovered. After these modifications were made the experiment was run.

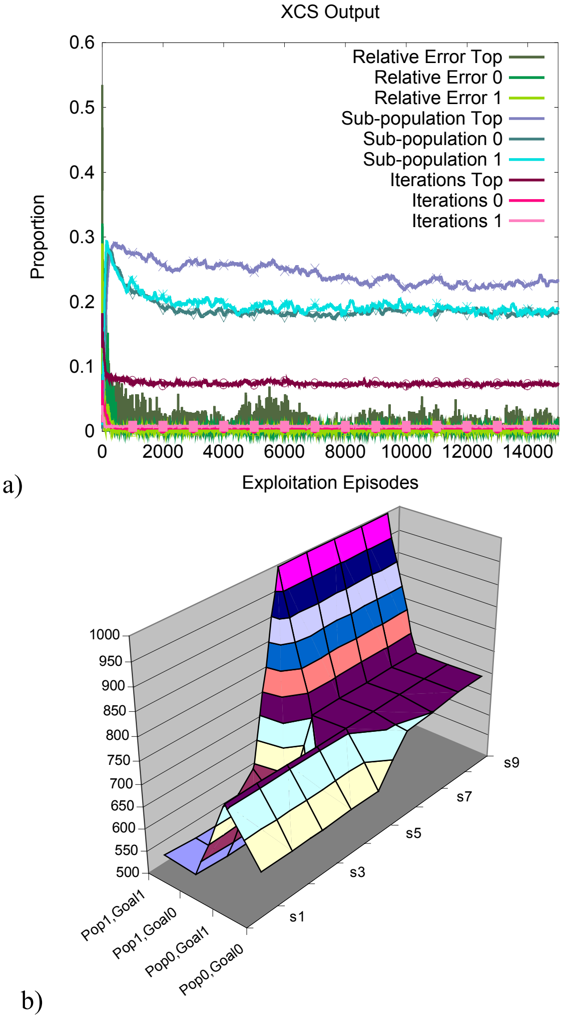

Figure 7 - a) The performance for Feudal XCS with two sub-populations in a two-goal length 11 corridor environment, and b) the coverage graph for the top-level population.

In Figure 7a the System Relative Error of the high-level XCS did not reduce as much as expected. It was hypothesised that this was due to the uneven nature of exploration - states s_0, s_5, and s_9 would be explored more regularly than the other states and similar problems had been seen in previous experiments (Barry, 2000a). This hypothesis was verified by reducing the start states to s_0, s_3, s_5, s_7, and s_9, where it was found that the dominance of [O] was normal and the System Relative Error was reduced to expected values. The dominance of [O] for the low-level XCS populations was high in all runs, demonstrating the ability of Feudal XCS to identify the optimal local state × sub-goal × action × payoff mappings, empirically verifying the first section of hypothesis 2. The second section of hypothesis 2 suggested that given a suitable policy these sub-populations could be used to provide a sequence of optimal local routes to achieve a global goal. This is demonstrated by the iterations plot for the top-level XCS in figure 7a. This plot reveals that the Feudal sub-population is able to achieve a global goal using the optimum one or two sub-population invocations (the line is plotted so that 0.1 on the scale represents the optimal two steps for the longest path).

A consideration of figure 7b reveals that the high-level XCS was able to identify the optimum pathway, using the lower level sub-goals and sub-populations, that will achieve the highest payoff from the environment. This demonstrates that the mapping created is the optimal global mapping of state × sub-population × sub-goal × payoff and thus hypothesis 3 is also upheld.

Whilst the Feudal XCS did acquire the capability to select between global payoffs, it should be noted that the global payoff chosen by XCS will not necessarily be that chosen by the normal XCS. For example, in the environment used for these experiments XCS will select a route to the state s_0 that provides the reward of 600 when starting in states s_1 to s_4 and the route to the state s_{10} that provides the reward of 1000 when starting in states s_5 to s_9. In Feudal XCS the reward of 1000 is a maximum of two 'macro-steps' away from any starting location, and therefore XCS will always prefer the sequence of sub-goals leading to s_{10}. Thus, the high level XCS population plans over sub-goals rather than individual states. As McGovern and Sutton (1998) note, this form of hierarchical approach produces routes to reward states that are optimal at the level of planning.

5 DISCUSSION

Previous work with Structured LCS is explored in more detail within Barry (2000b), but it is helpful to consider a number of relevent earlier approaches in the context of this work. Booker (1982) used multiple instances of his GOFER LCS implementation to differentiate between input and output mappings and enable the LCS to learn internal associations between input and output. This represents a different aim to that of the Feudal XCS which focuses on learning to plan over concurrently learnt subgoals and competences. Bull and Fogarty (1993) used a number of classifier populations that could switch each other on or off by messages to a shared message list.

These LCS populations were stimulus-response systems, although learning a long-term behaviour, and this work therefore has much in common with the work of Dorigo.

The main body of previous investigation into hierarchical forms of LCS was performed by Dorigo and collegues (e.g. Dorigo and Schnepf, 1993; Dorigo and Colombetti, 1994). Using ALECSYS they created fixed control hierarchies. Their work was characterised by the dependency upon direct environmental feedback for the reward of switching decisions made by the upper level LCS. Their bottom-up hierarchical approach required input to be divided between the low level populations. Each decided whether to propose an action, and the top-level LCS chose between the actions. In an alternative top-down approach a state memory was used to identify the current goal. Each lower level LCS learnt to use the state memory to identify which LCS should operate and a co-ordinator LCS learnt to control this memory switch. Although the learning environments were multiple-step environments, a regular payoff for each action was provided and training was performed separately.

In contrast, Feudal XCS is designed to learn within delayed-reward environments - the purpose of Feudal XCS is the decomposition of large action sequences into smaller units and the localisation of reward within those units. Secondly, the Feudal XCS selects lower level capabilities based on identified sub-goals, and uses these to plan at a higher level. Whilst ALECSYS did select between behavioural competences, it did not use the competences to identify sub-goals that established a route to a rewarding state. Finally, the Feudal XCS maintains all the capabilities of XCS to acquire accurate and optimally general mappings of each state-space partition and sub-goal space, which is not possible in ALECSYS.

Much further work remains to be done to assess the scalability and wider applicability of this approach. It must be applied to larger numbers of sub-divisions and scaled to operate with more than one level of decomposition. In particular, exploration of the potential for autonomous identification of subgoal states would lead to a truly emergent hierarchical approach. However, these results do provide encouragement and expand upon the available research results for hierarchical LCS forumulations.

References

Barry, A.M. (1999), Aliasing in XCS and the Consecutive State Problem: 1 - Problems. In Banzhaf et al, 1999.

Barry, A.M. (2000a), Specifying action persistence in XCS. In Whiteley et al, 2000.

Barry, A.M. (2000b), XCS performance and population structure in multiple-step environments, PhD Thesis, Queens University Belfast.

Barry, A.M. (2001), The stability of long action chains in XCS, to be published in Bull, L., Lanzi, P-L (eds), The Journal of Soft Computing, Sept 2001.

Banzhaf, W. et al. (eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99).* Morgan Kaufmann: San Francisco, CA, 1999.

Booker, L. B. (1982), *Intelligent Behaviour as an Adaptation to the Task Environment*, Ph.D. Dissertation, The University of Michigan.

Bull, L., Fogarty, T. C., (1993), Co-evolving Communicating Classifier Systems for Tracking, in Albrecht, R.F. et al (eds.), *Proc. Intl. Conf. on Artificial Neural Nets and Genetic Algorithms*, Springer-Verlag.

Butz, M. V., Stolzmann, W., Goldberg, D. E., (2000), Introducing a Genetic Generalisation Pressure to the Anticipatory Classifier System Part 2: Performance Analysis, In Whiteley et al, 2000.

Cliff, D., Ross, S., (1994), Adding Temporary Memory to ZCS, *Adaptive Behaviour*, 3(2), 101-150.

Dayan, P., Hinton, G. E. (1993), Feudal reinforcement learning, in Hanson, S. J. et al (eds.), *Neural Information Processing Systems 5*, Morgan Kaufmann.

Dietterich, T. G. (2000), *An Overview of MaxQ Reinforcement Learning,* Technical Report, Computer Science Department, University of Oregon.

Dorigo, M., Colombetti, M., (1994), Robot shaping: Developing autonomous agents through learning, *Artificial Intelligence*, 71 (2), 321-370, Elsevier Science.

Dorigo, M., Schnepf, U. (1993), Genetics-based Machine Learning and Behavior-Based Robotics: a new synthesis., *IEEE Trans. Systems, Man, and Cybernetics*, 23(1).

Kovacs, T., (1996), Evolving optimal populations with XCS classifier systems. Tech. Rep. CSR-96-17, School of Computer Science, University of Birmingham, UK.

Lanzi, P.L., (1997), Solving problems in partially observable environments with classifier systems, Tech. Rep. N.97.45, , Politecnico do Milano, IT.

M^{c}Govern, A., Sutton, R. S. (1998), *Macro-Actions in Reinforcement Learning: An Empirical Analysis*, Technical Report 98-70, Computer Science Department, University of Massachusetts, Amherst.

Parr, R., Russell, S. (1998), Reinforcement Learning with Hierarchies of Machines, in *Advances in Neural Information Processing Systems*, 10, MIT Press.

Riolo, R.L. (1987), Bucket Brigade performance: I. Long sequences of classifiers, in *Proc. Second Intl. Conf. on Genetic Algorithms and their Applications*, 184-195.

Whitely, D., Goldberg, D. E, Cantú-Paz, E., Spector, L., Parmee, I., Beyer, H-G., (eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, Morgan Kaufmann.

Wiering, M., Schmidhuber, J. (1996), HQ-Learning: Discovering Markovian Sub-Goals for Non-Markovian Reinforcement Learning, Technical Report IDSIA-95-96.

Wilson, S.W. (1998), Generalization in the XCS Classifier System, in *Proc. 3rd Ann. Genetic Prog. Conf.*