Specifying Action Persistence within XCS

Alwyn Barry

Faculty of Computer Studies and Mathematics,
University of the West of England,
Coldharbour Lane, Bristol, BS16 1QY, UK

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

Abstract

In investigating the Consecutive State Problem within XCS (Barry, 1999) it was suggested that a possible solution lay in allowing the XCS to persist with a single action over the aliased states. It was shown that this technique was sufficient to overcome the Consecutive State Problem as long as mechanisms were also provided which prevented the persistent application of 'Null Actions'. An alternative solution based on the work of Cobb and Grefenstette (1991) was discussed which sought to extend the action of each classifier so that each classifier could specify the duration that the action should be applied for. It was noted that this was an inadequate solution for the Consecutive State Problem because XCS would still explore the possibility of an action which persisted into but not beyond the aliased states. This work now applies these ideas to a number of non-aliased multiple step environments. It demonstrates that, given a suitable exploration strategy, action persistence can be utilised within XCS to enable the selection of a pathway to a reward state which entails the minimum number of different *actions*. It is also shown that a modification to the learning mechanism restores the ability of XCS to select the pathway to a reward state with the minimum number of *steps* whilst minimising the number of actions used.

1 INTRODUCTION

Wilson (1995) introduced a novel form of Learning Classifier System based upon his simple ZCS (Wilson, 1994) implementation. This new LCS, termed XCS, uses Accuracy as its Fitness criteria, combined with a niching mechanism derived from that of Booker (1989). In Wilson's *Generalisation Hypothesis* it was suggested that the mechanisms of XCS will introduce, identify and proliferate general accurate classifiers within the classifier population. Kovacs (1996) investigated this claim, and demonstrated further that XCS will find and maintain the sub-population of optimally general classifiers, and hypothesised that XCS would always produce this sub-population - the Optimisation Hypothesis. Lanzi and Colombetti (1999) have demonstrated that XCS is even able to satisfy this hypothesis within environments with moderate amounts of uniform random noise in their feedback, demonstrating a surprising degree of robustness within XCS. Kovacs (1999) reasoned that only LCS implementations which utilise the kind of accuracy measure adopted by XCS will be able to overcome the problem of correctly identifying and deleting the so-called 'strong over-general' classifiers - those classifiers which match in too many states but nonetheless retain a high payoff value from those states in which the actions they propose are correct. XCS thus represents a large step forward in the performance and reliability of Learning Classifier Systems.

Although initial investigations into the use of XCS within environments which required a number of steps to be taken before a reward is delivered were fairly limited, Lanzi (1997, 1998) has recently considerably extended knowledge in this area. Barry (1999) continued Lanzi's work into the problem of learning over states generating aliased messages as part of wider research into the emergence of hierarchical invocation of classifier sequences. This work identified a form of the Aliasing Problem where the aliasing states occur in consecutive states. It was shown that, since the Consecutive State Problem is a sub-problem of the Aliasing Problem, it is possible to devise an alternative solution to this sub-problem which is simpler to implement than a solution to the Aliasing Problem as a whole. The solution involved modifying XCS so that an action in state S_{t-1} would persist in S_t if the environmental input I_t in S_t was the same as that in S_{t-1} ($I_t = I_{t-1}$). The payoff and rule induction mechanisms were therefore effectively suspended until a future state S_n was reached in which $I_n \neq I_{t-1}$. A generalisation of this solution can be conceived in which each classifier is able to specify in its action how long the action should persist. This form of action persistence was first demonstrated by Cobb and Grefenstette (1991) in their SAMUEL LCS implementation in which it was shown that the LCS was able to identify a number of time steps over which each action should persist within a 'missile avoidance' environment. Barry (1999) presented some initial results from the application of this technique to XCS as a solution to the Consecutive State Problem, but argued that this technique was not an adequate solution to the Consecutive State Problem. However, this does not preclude the mechanism from a more general utilisation within XCS. In this paper the application of action persistence identification to XCS within non-aliased general multiple step environments will be investigated in an attempt to establish more general results on the use of this technique within XCS.

2 XCS STRUCTURE AND OPERATION

The XCS Learning Classifier System (Wilson, 1995, 1998) is, on an initial inspection, similar to traditional Learning Classifier Systems. Detectors interact with an 'environment' to produce a binary encoded message which becomes the input to the XCS. This is *matched* against a population of classifiers, each consisting of a ternary coded condition and an encoded action, in order to identify those classifiers which are relevant to the current input condition. Those classifiers which *match* the message are used to create the *Match Set* [M]. [M] is a set of records where each record identifies a distinct action, the set of classifiers which have been matched that propose the action, and the predicted payoff that will be received upon performing the action. The payoff prediction is the weighted sum of the payoff prediction of each classifier in the record. A record from [M] is chosen to be the *Action Set* [A] that performs an action. This is chosen arbitrarily if *exploring* to enhance the classifier representation or chosen by selecting the highest predicted payoff if seeking to *exploit* the learnt classifier representation. The action advocated by [A] is performed in the environment by decoding the action representation through an effector interface. If a reward R is received from the environment the goal is considered to have been reached and R is used to update the predictions of all classifiers in [A] using the modified Widrow-Hoff update mechanism known as MAM (Venturini, 1994). If no reward is received, and the environment is potentially a multi-step environment, the action is considered to be one action en route to the goal and payment is taken from the maximum prediction of [M] in the next iteration discounted by a discount factor γ ($0 < \gamma < 1$). Thus, any accurate classifiers in an [A] which leads directly to a reward R can be expected to converge to a prediction of R, and those i steps before the reward will converge to $\gamma^i R$. The speed of convergence is controlled by the learning rate parameter β ($0 < \beta \leq 1$) within the Widrow-Hoff update equations.

Within XCS each classifier carries with it a *Prediction* measure - the prediction of the average payoff it receives when invoked. Unlike the 'Strength' measure within traditional LCS implementations, XCS only uses this measure for Action Selection. For selection as a partner for breeding within the GA, XCS maintains the *Error* and *Accuracy* measures which provide two viewpoints on the accuracy of the prediction measure. A classifier can be inaccurate because its prediction has not yet been updated sufficiently to make it accurate or because it has an over-general condition which involves the classifier in too many [M]. Inaccurate classifiers could nonetheless have a high prediction and therefore it is important to remove them in favour of accurate classifiers. The classifier *Fitness*, used in the GA selection for crossover, is the accuracy of a classifier relative to other classifiers in the [A] the classifier occurs within. Thus, the GA will favour accurate classifiers over inaccurate and will, over time, replace inaccurate classifiers with accurate versions. Furthermore, the fitness is used to weight the contribution of the classifier's prediction within [A] so that accurate classifiers contribute more and drive the System Prediction towards higher accuracy whilst increasing the calculated error within the inaccurate classifiers. Since an accurate general classifier occurs in more [A] than an accurate but more specific classifier and because the invocation of the GA is tied to occurrences within [A], the more specific classifiers are also driven out of the population. The classifier deletion mechanism, used when the population becomes full, deletes classifiers based on the average number of classifiers that exist within the [A] each classifier occupies. This dynamically adjusts the population composition to provide sufficient population niches (Booker, 1989) for all the accurate optimally general classifiers (given sufficient population space). The Optimality Hypothesis (Kovacs, 1996) suggests that XCS is thus capable of identifying and maintaining the accurate optimally general population (termed [O]), and this has been demonstrated for a number of small problems (e.g. Kovacs, 1996; Saxon and Barry, 1999).

XCS identifies duplicate classifiers on creation and increases a *Numerosity* count held by the classifier to represent the number of duplicates. This facility increases the speed of operation of the XCS without changing the XCS execution (Kovacs, 1996). Such classifiers are termed *MacroClassifiers*, in contrast to the usual *MicroClassifier* representation. During insertion after the GA a process known as *Subsumption Deletion* (Wilson, 1998) is used. This looks for experienced general classifiers within [A] which will match more messages than the new classifier. If such a classifier is found, the new classifier is discarded and the numerosity of the *subsuming* classifier is incremented instead. This process provides a further pressure towards optimum generalisation.

XCS thus introduces a swathe of new features which together provide XCS with a unique ability in the field of LCS research to find the optimally general accurate classifiers that represent the complete State × Action × Payoff mapping of an environment.

3 ACTION PERSISTANCE

In Reinforcement Learning many test environments are formulated using world representations which are divided into discrete 'states'. The Finite State World (Grefenstette, 1987; Riolo, 1987) representation and Grid Worlds such as the Woods environments (e.g. Wilson, 1995) are examples of such discrete Markovian environments. Bonarini, Bonacina and Matteucci (1999) provide an excellent exposition of the benefits of such environments and the price that must be paid in regard to the general applicability of results from such environments.

In these environments it is common for an Animat to take a single action in each state, which will either lead to the Animat staying in that state or moving to a new state. The Animat will then invoke the deliberative process of the Learning System to decide on which action is best from the new state. 'Time' is synonymous with this 'detect-decide-act' process; one unit of time equivalent to one such 'step'. Barry (1999) hypothesised that a solution to the Consecutive State Problem existed if XCS was allowed to continue with the same action whilst the input to the XCS remained constant, provided that 'Null Actions' were disallowed. This hypothesis was validated using a suitably modified XCS. This solution is a specific solution to the problem of consecutive aliased states, but shows some similarities with a more general feature proposed for LCS implementations by Cobb and Grefenstette (1991). In their SAMUEL implementation the action of a classifier not only specifies an action, but also specifies the duration over which the action may continue to operate, with the duration specified in terms of environmental 'steps'.

This mechanism is an appropriate mechanism for application to XCS only if it is possible for XCS to determine the duration for the application of an action which will produce the highest accurate payoff. Consider a hypothetical XCS implementation that allows persistence, termed PXCS. This implementation provides as a payoff the value of the final state entered for a persistent action which is a legal action (is a label of an edge from the current node within a FSW) for all the steps specified in the duration specification part of the action. If the action becomes illegal (is not a valid label of an edge from the current node) at any time during the persistence, a payoff of 0 is given. Now consider a finite state world consisting of a chain of sparsely connected nodes, as shown in Figure 1a.

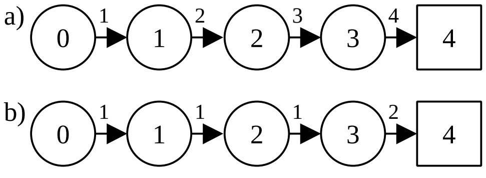

Figure 1. FSW without and with consecutive actions

For a standard XCS three classifiers are required to traverse this FSW : 0→1, 1→2, 3→4. In this FSW any classifier within PXCS which specifies an action duration greater than 1 cannot succeed because there are no set of consecutive transitions which have the same action label. Where a specification of an action persistence cannot succeed for the duration specified by the classifier an immediate non-environmental payoff R=0 will be given. In this case, any classifier specifying an action duration of greater than one step will become accurate and be preserved within the population, but will eventually have a prediction of 0 and thus not be selected during exploitation cycles. Thus XCS is capable of detecting when an action duration of greater than one step is not required.

Now consider the environment depicted in Figure 1b. In this environment a classifier which seeks to perform action 1 in state s_0 for one step will tend to receive a payoff of $\gamma^3 R$ if single step actions exist for all following states. However, a classifier that seeks to perform the action 1 in state s_0 for two steps will tend to receive a payoff of $\gamma^2 R$ under the same conditions. Both classifiers will be maintained within the XCS population as accurate classifiers, but the classifier with an action persistence of 2 will have a higher stable prediction than that with a persistence of 1 and therefore be selected in exploitation. This argument can be trivially extended to any number of consecutive states which will admit to the same action and ultimately lead towards a fixed point reward. As in the case of Figure 1a, classifiers which propose a duration greater than can be usefully employed will receive a payoff of 0 and therefore not be selected during exploitation. Notice that 'null actions' - those actions which lead immediately back to the same state, will receive the discounted payoff from the state and therefore will also not be selected during exploitation.

This argument leads to the first hypothesis:

Hypothesis 1 - PXCS is able to identify, maintain, and optimally utilise the classifier in each state which will allow the longest persistence in action on any action chain which leads to a stable environment reward.

The rationale behind Hypothesis 1, although not the Hypothesis itself, is based on the assumption that only single step actions exist for all following steps. This will not be the case for any construction of the classifier system that has more than two consecutive states, and thus will not be the case for almost all cases of useful persistence. Consider Figure 1b. Assume that a classifier c_3 receives a reward R from the environment for moving from s_3 into the reward state s_4. In this case, a classifier c_2 will receive a fixed payoff of γR for moving from s_2 to s_3. Similarly a classifier c_1 should receive a fixed payoff of $\gamma^2 R$ for moving from s_1 to s_2 and c_0 should receive a fixed payoff of $\gamma^3 R$ for moving from s_0 to s_1. However, if a classifier c_{1p} exists which moves with duration 2 from s_1 to s_3 it will receive the payoff of γR. This will become the highest prediction in the match set [M$_1$] for s_1 and will therefore be the payoff value for preceding states. Similarly if a classifier c_{0p} exists which moves with duration 3 from s_0 to s_3 it will receive the payoff of γR and this will become the highest prediction in the match set [M$_0$] for s_0. Since all classifiers in any action set within any state will receive a payoff based on the highest action set prediction within the match set for that state, all classifiers covering states s_0 to s_2 will receive the payoff

$\gamma^2 R$ apart from the classifier which moves with the appropriate duration to s_3. Thus, the property of temporal difference is disrupted over the states in which persistence of action can occur. This does not invalidate Hypothesis 1 since the longest duration action that remains legal throughout its invocation will continue to hold the highest prediction, and will therefore be selected in exploitation. However, there will be no quantitative measure of the utility of any other action × duration pair that does not lead to a higher payoff value.

This problem cannot be solved by a re-definition of PXCS in which the payoff given to an Action Set is $\gamma^\delta P$, where P is the payoff from the Action Set in time t+1 or the environmental reward and δ is the duration of an action successfully performed for the whole of its duration. This formulation would result in classifiers which specify a duration obtaining the same stable prediction as a classifier specifying a single step with the same action from the chain of steps which ultimately reach the same high payoff state. PXCS would no longer be able to select the highest duration classifier since it would receive the same payoff at the point of invocation as classifiers of a lower duration. If the discount rate γ was replaced by a lower discount rate $\Gamma < \gamma$ for actions with a duration higher than 1 it would be possible to favour actions with a duration but it will still not be possible to identify the longest correct duration action.

Ultimately the Action Set prediction score represents the maximum payoff that can be expected when that action set is chosen, and therefore it is appropriate that the mechanism chosen for PXCS is not changed. In effect PXCS chooses the highest payoff achievable in the lowest number of *different actions*, and therefore represents an alternative form of learning system than XCS, which chooses the highest payoff achievable in the lowest number of *steps*. Consider Figure 2. If the start state is s_0 and R=1000 is given in the reward states s_4 and s_7 then XCS would move to s_5, s_6, and s_7 whereas PXCS would choose s_1 through to s_4. Furthermore, with γ=0.71, R from s_4 would have to be greater than 1408 in order for PXCS to choose the route to s_4.

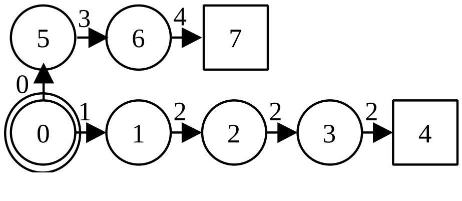

Figure 2. A FSW not optimally selected by PXCS

Hypothesis 2 - In exploitation PXCS will select the highest payoff achievable with the lowest number of separate actions.

A least impact solution to this problem exists by utilising a [relatively small] proportion p of the discounted payoff removed from the payoff in inverse proportion to the duration: $\gamma^{\delta-1}P$-(1.0-δ/Δ)p$\gamma^{\delta-1}P$ (where Δ is the maximum possible duration). In the example FSW of Figure 2, if R=1000.0, γ=0.71, Δ=4, and p=0.1, then in s_1 the stable prediction for a duration 3 move to s_4 would be 0.71^2x1000.0 - (1 - 3/4)x0.1x(0.71^2x1000.0) = 491.49 and thus in s_0 the stable prediction for a move to s_1 will be 348.96. The stable prediction for a move from s_0 to s_5 will be 431.32 and thus this proposal will allow XCS to continue to select the closest equal rewarding state. Now, if a state s_8 was imposed between s_6 and s_7 the stable prediction for a move to s_5 would be 283.27 and XCS would chose the action leading to s_1 in preference to the action leading to s_5. This illustrates that the addition of the small fixed payment can ensure that the modified PXCS chooses the path with the fewest different actions where two equal length paths lead to the same reward, and leads to the third hypothesis:

Hypothesis 3 - Re-instatement of step-based discounting of the payoff with the addition of a small step based additive component to the payoff will allow PXCS to preserve the Temporal Difference properties of XCS whilst selecting the path with the lowest number of separate actions where equidistant paths to equal rewards exist.

4 EXPERIMENTAL INVESTIGATION

In order to investigate the hypotheses a number of FSW were constructed. FSW are appropriate for this investigation because of the control they provide over the number of actions available within any state, the number of states which can be entered from within a state, and the message which is produced to identify a state. The base parameterisation of the XCS or PXCS was set 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, X=0.8, μ=0.04, P(#)=0.33, s=20 (see Kovacs (1996) for a parameter glossary), and the maximum trial length was set to 50. These parameters were chosen for consistency with previous work, but appear appropriate given the level of complexity of the tests used. Any variation in the parameterisation for particular experiments is stated alongside the experimental results.

4.1 PROVIDING PERSISTANCE

To investigate whether PXCS is able to find classifiers which identify the optimal time over which an action should persist a simple two action ten state FSW with no null actions was created (figure 3). In order to create an implementation of PXCS a working XCS implementation was modified so that an action posted to the environment with a duration greater than one would cause the environment to continue to utilise the action, decrementing a duration counter each time, until the duration counter became zero or the action was inappropriate for the environmental state. No environmental message was sent back to the XCS during the operation of the action. Once the action has been performed the normal XCS cycle resumes. The calculation of payoff was also modified so that if the previous action operated over a duration the payoff was only given if the full specified duration was completed (the duration counter for that action was reduced to zero). No other changes to the XCS implementation were required.[1]

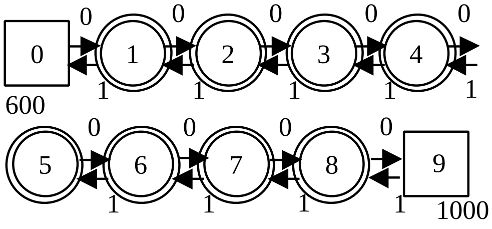

Figure 3. An FSW to test persistence

The PXCS implementation was tested by providing the environment shown in Figure 3, setting the action size to 4 to allow 3 bits for the specification of the duration and one bit for the action, introducing a population of fully specific classifiers covering all state × action × duration combinations, and running PXCS with all the induction algorithms turned off. The resulting population was examined to determine whether the XCS had learnt the optimal duration from each state.

Table 1. Selected classifiers from a pre-loaded PXCS

Classifier	Pred	Exp	Classifier	Pred	Exp
00010→0000	710.0	207	00010→1000	710.0	208
00010→0001	710.0	188	00010→1001	600.0	210
00010→0010	710.0	206	00010→1010	0.0	198
00010→0011	710.0	244	00010→1011	0.0	201
00010→0100	710.0	218	00010→1100	0.0	197
00010→0101	710.0	186	00010→1101	0.0	196
00010→0110	1000.0	188	00010→1110	0.0	212
00010→0111	0.0	217	00010→1111	0.0	204

Table 1, which gives a selection of classifiers from state s_2, illustrates that the optimal duration from each state was correctly identified by PXCS. Any duration that was too long achieved a stable prediction of 0 and any duration that was too short achieved a stable prediction of 710.0. As predicted, all classifiers which do not lead directly to the reward state indicate that they are only one step from the reward state because there will always exist a classifier in the resulting state which specifies the correct duration to reach the reward state directly. Thus, as predicted, PXCS reflects the number of different actions required to move to the reward, not the number of steps.

The ability of PXCS to learn optimal durations through the induction mechanism was now investigated. To provide baseline results, the standard XCS was applied to this two-reward environment. It was found that the absence of an input to XCS of a '00000' message allowed some states to be represented by competing generalisations (two different conditions with equal generality). The state messages were therefore re-organised so that s_n produced a message corresponding to the binary representation of n-1. XCS was able to learn the 16 classifiers of the optimal population [O] for this environment with the following averaged coverage table (from 10 runs):

Table 2. The coverage table from an XCS in a 2 reward state corridor environment

000	001	010	011	100	101	110	111
302	215.9	182.5	253.8	356.8	501.2	708.6	999.2
598.4	424.8	301.7	214.7	181.0	253.3	356.6	502.4

Before PXCS was introduced, the ability of XCS to learn given the same number of action bits as would be required by PXCS was ascertained. The actions were extended to four bits with only bit 0 interpreted. In our experience, 10-12 micro-classifiers are required for each member of [O] to be established without threat from competing generalisations. Since the predicted [O] would now increase from 16 to 128 classifiers, the population was increased to 2000. The number of learning episodes [where a learning episode is a exploration episode followed immediately by an exploitation episode] was increased to 15000. Space precludes inclusion of the resultant [O] or coverage table, but XCS was able to establish [O] within 3000 exploration episodes with each action set highly converged on the optimal classifier.

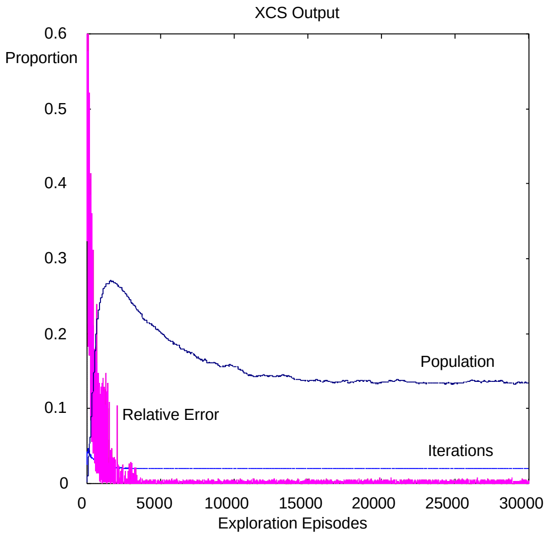

Figure 4. Performance of PXCS within two-reward single chain FSW

PXCS was now introduced, reducing the population to 1500 to allow for predicted generalisations and running each of the 10 test runs for 30000 learning episodes to allow for the increased complexity of learning these generalisations. On examining the resultant performance

[1] This implementation is potentially limited. It would be preferable for XCS to be able to interrupt an action in a changing environment where a message indicates that an alternative action is desirable.

graph, the population had stabilised by 15000 episodes, with the system relative error (Barry, 1999) reduced to close to zero by 4000 episodes, as shown in Figure 4. Table 3 shows the averaged coverage table for the first three states of the environment. The rows of this table represent the durations (1-8) for action 0, whilst the columns reflect the averaged prediction, number of macro classifiers in the action set for the state, total numerosity of these classifiers, and maximum numerosity of the most represented classifier.

Table 3. An extract from the coverage table of PXCS in a 2 reward state corridor environment

000	m	N	>N	001	m	N	>N	010	M	N	>N
709.0	5	64	25	709.3	5	55	28	708.9	5	41	35
709.4	5	47	24	709.4	4	54	28	710.9	5	42	35
709.2	5	44	24	711.1	5	36	26	706.3	5	41	35
712.8	6	34	20	708.5	4	31	26	708.7	4	40	35
709.3	5	46	22	712.6	5	35	25	713.3	6	27	17
709.6	5	34	21	703.4	4	31	25	998.7	6	52	43
709.7	6	34	19	999.2	6	55	45	4.1	6	33	24
999.9	6	53	42	3.2	6	30	22	5.6	6	37	24

When the population was inspected it was found that, although [O] was fully formed, more specific classifiers continued to exist within the population. An detailed examination of the populations revealed that the additional classifiers were all younger and yet more specific than the optimally general classifier within their action set. The PXCS was modified to examine the operation of the LCS during induction. It was found that on occasion the GA mutation operator will create classifiers outside the currently selected action set due to mutation of the action. Classifiers lying outside the current action set will not be subsumed by the existing optimally general classifier within the current action set, and Wilson(1996) does not provide for population-wide subsumption. Within XCS these are rapidly deleted due to the wider GA opportunities of the optimal classifier if the classifier is fit but over-specific, or due to the low fitness if the classifier is over-general. Within PXCS it was found that the exploration of the state × action × duration × payoff map was much less even than within XCS. It was therefore hypothesised that mutation by the GA introduced over-specific classifiers which were not eradicated within the PXCS.

This hypothesis was tested initially by reducing the population size from 1500 to 1000, and 800 to put more pressure on the general classifiers to make optimal use of the population space. Although this did eradicate the problem by a population size of 800, it also compromised the formation of [O]. The hypothesis was therefore further tested by modifying XCS to provide population-wide subsumption after a failure of normal action-set subsumption from the GA (although in general such a technique could delay the removal of over-general classifiers).

The modified PXCS was re-run for 10 runs within the same environment and it was found that the populations were strongly converged with no over-specific classifiers. A single factor anova test of the average number of macro-classifiers within each action set revealed a significant change in the number of macro-classifiers within each action set between this run and that of the standard PXCS (P(0.01), F=109.062, F_{crit}=6.73572). Together with the evidence that the over-specific classifiers were not seen within the standard XCS with four action bits, we therefore conclude that our hypothesis was upheld.

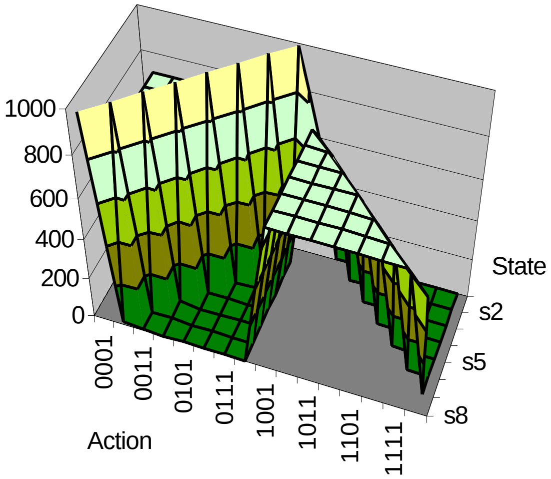

Figure 5. Predictions of PXCS for each action in each state in the two reward corridor FSW.

In order to verify Hypothesis 1, each match set within the coverage table for this PXCS was examined. Figure 5 is the plot of predictions for actions against state, showing that PXCS selects duration 8-n-1 for state s_n predicting a reward of 1000 for these actions. Thus, in all states the action leading to the highest available reward regardless of duration is selected. PXCS has also identified all action × duration combinations which are too long (prediction 0) and not long enough (prediction 710). Although more difficult to identify from the plot, PXCS has also identified that the action × duration combinations which lead to a reward of 600. This demonstrates that PXCS is able to identify, maintain, and optimally utilise the classifier in each state which will allow the longest persistence in action on any action chain which leads to a stable environment reward, confirming hypothesis 1.

4.2 SELECTION OF DURATION

When figure 5 is examined, it is apparent that PXCS will always select the action that leads to state s_9 even when in s_1. This is in contrast to XCS, which would trade-off the size of reward and the distance to the reward to choose movement to s_0 from s_1. Whilst this behaviour verifies hypothesis 2, it does not verify the behaviour of PXCS in an environment that does not provide a single [persistent] action direct to reward route. To further test PXCS, a new FSW based upon a Benes Switch (used in computer network switching to create low contention switching from simple crossbar switches). This FSW requires a four-step solution for XCS, but the reward state can be reached using a two step solution with PXCS, and there are competing solutions requiring 3 or 4 steps. No single step solution to a non-zero reward exists.

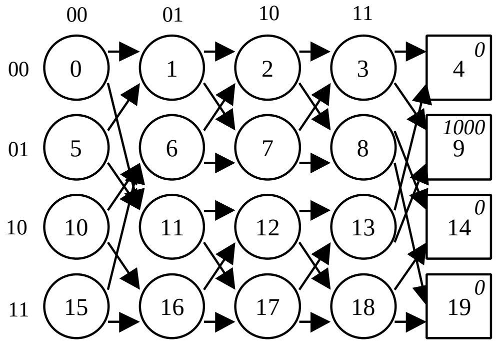

Figure 6. A FSW derived from the Benes Switch

To help XCS utilise any potential generalisation across columns or rows, the non-reward states were labelled for the creation of input messages by concatenating the row bits and column bits identified in figure 6. Initially the start states were states s_0, s_5, s_{10} and s_{15}. A base-line XCS learning experiment was conducted using a population limit of 300 micro-classifiers. The following is a typical [O] found by one of the ten runs used:

Classifier	Pred	Error	Fitness	Acc	N	[A]
###00→0	357.74	0.0002	0.9745	1.00	35	36.94
###01→0	504.03	0.0003	0.9913	1.00	33	33.36
###10→0	710.00	0.0000	1.0000	1.00	35	37.15
###11→0	0.00	0.0000	1.0000	1.00	30	32.30
###00→1	357.56	0.0004	1.0000	1.00	34	37.72
###01→1	500.52	0.0024	1.0000	1.00	37	39.81
###10→1	3.86	0.0023	0.7320	1.00	25	38.53
##011→1	1000.0	0.0000	1.0000	1.00	31	34.00
##11#→1	2.06	0.0022	0.6643	1.00	22	37.49

When PXCS was run within this environment (with the population set to 1000, and 30000 learning episodes used in each of the 10 runs) although it was able to find an [O] some populations were unable to sustain all members of [O] at high numerosity and a small number of over-general classifiers continued to exist within the population. An examination of the relative experience of the classifiers revealed a highly irregular exploration pattern. Although this was also the case within XCS, the use of persistence meant that classifiers covering states within rows 1 and 2 were inadequately explored. The disruptive effects of inadequate exploration had been noticed by Lanzi (1997) in another context, but rather than employ his 'teletransportation' mechanism, the situation was remedied by allowing all non-reward states to be start states. The presence of additional classifiers generated by mutation remained a problem, and so population subsumption for child classifiers not subsumed by the action set was applied.

In all 10 runs with PXCS, [O] was obtained. The following is an example of an accurate sub-population:

Classifier	Pred	Classifier	Pred
###0#→0000	503.57	###0#→1000	503.31
###10→0000	707.84	##0#1→1000	999.96
###11→0000	0.00	##010→1000	6.32
###00→0001	503.67	##011→1000	1000.0
###01→0001	708.38	##11#→1000	0.27
###1#→0001	0.00	####1→1001	0.17
####1→0010	0.00	###00→1001	503.03
###00→0010	709.60	###1#→1001	0.00
###1#→0010	0.00	#####→1010	0.12
#####→0011	0.00	#####→1011	0.00
#####→0100	0.00	#####→1100	0.00
#####→0101	0.00	#####→1101	0.00
#####→0110	0.00	#####→1110	0.00
#####→0111	0.00	#####→1111	0.00

PXCS has learnt to apply a three step duration from any of states s_0, s_5, s_{10} and s_{15}. Once in s_3, PXCS selects a one step action into the reward state s_9. Although other duration actions are available, in exploitation PXCS will select the highest payoff achievable with the lowest number of separate actions, as stated in hypothesis 2.

4.3 RE-INSTATING TEMPORAL DIFFERENCE

It has been shown that PXCS is able to identify the lowest number of distinct actions, but at the cost of possibly ignoring nearer low value rewards. Thus, PXCS does not provide the Temporal Difference learning of XCS. It was suggested that PXCS could be modified to discount the reward or payoff received so that it was equivalent to that received for a succession of single steps to the reward. Although this would restore the TD properties of XCS, it would not allow PXCS to favour a higher reward obtained by initiating a persistent action. However, if the payoff was then further modified by a small amount so that longer duration actions were favoured, account for the duration of the action could be taken.

This technique was implemented within PXCS, creating the *discounting* PXCS (dPXCS). Whenever a reward was received the reward allocated was $\gamma^{\delta-1}R$ (where R is the reward and δ is the duration just applied), whilst payoffs were discounted as $\gamma^{\delta}P$-(1.0-δ/Δ)p$\gamma^{\delta}P$ (where P is the maximum prediction from the match sets in the next PXCS iteration, p=0.2 is a constant representing the amount of the payoff we will further adjust, and Δ is the maximum persistence possible). The two-reward corridor environment was used as a test environment for dPXCS, allowing comparison of the results gained in the previous experiments. The experiment was run for 10 runs of 30000 learning episodes and a population size of 2000.

When the final populations were examined, it was found that dPXCS was able to learn the separate payoffs for each state × duration × action combination, and formed [O] in all runs. The coverage table was examined and, as Figure 7 illustrates, a graduation of the state × action × duration × payoff mapping was found. Thus, in state s_4, dPXCS will select a duration 5 forward action to s_{10}, whereas in state s_3 it will select a duration 3 backward action to s_0, restoring the Temporal Difference property.

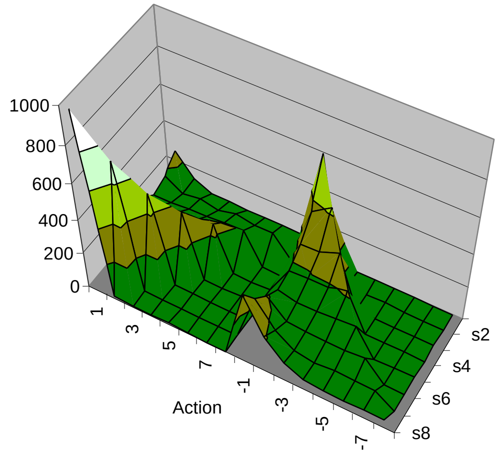

Figure 7. Predictions of PXCS for each action in each state in the two reward corridor FSW.

5 CONCLUSIONS

In Barry (1999) it was noted that the use of the Greffenstette and Cob (1991) prediction mechanism within XCS was inadequate for the solution of the Consecutive State aliasing problem. It has now been shown within two small Finite State Worlds that XCS is able to learn the optimal duration over which to apply an action, both when leading directly to a reward and when choices on the pathways to an ultimate reward are made. More importantly, it has been shown that an XCS modified to provide duration learning is able to establish and maintain accurate and optimally general maps of the state × action × duration × payoff mapping of these environments. This demonstrates that the Generalisation Hypothesis can be extended to learning over durations.

However, it has been shown that a naïve approach to persistence within XCS can lead to the removal of some of the Temporal Difference properties of the XCS. This means that rather than selecting the path to a reward based on a function of the reward magnitude and the distance to the reward, only the reward magnitude is taken into account. Nevertheless, it was shown that it is possible to restore the function so that the original operation of XCS is preserved whilst providing selection over durations by introducing a discounted reward and payoff mechanism.

These facilities build further the developing toolset of techniques which can be applied to XCS. However, the true benefit will possibly only be seen when they are applied successfully within the domain of mobile robotics, which remains as work to be done.

References

Barry, A.M. (1999) Aliasing in XCS and the Consecutive State Problem: 2 -- Solutions. In Banzhaf et al, 27-34.

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

Bonarini, A., Bonacini, C., Mattiucci, M. (1999), Fuzzy and Crisp representation of real-valued input for Learning Classifier Systems, in Wu, A. (ed.), *Proceedings of the 1999 Genetic and Evolutionary Computation Conference Workshop Program*, 228-235.

Booker, L.B. (1989), Triggered Rule Discovery in Classifier Systems, in Schaffer, J.D. (ed.), *Proc. Third Intl. Conf. on Genetic Algorithms*, 265-274.

Cobb, H.G., Grefenstette, J.J. (1991), Learning the persistence of actions in reactive control rules. In *Proceedings 8th International Machine Learning Workshop*, pages 293-297. Morgan Kaufmann, 1991

Grefenstette, J.J. (1987), Multilevel Credit Assignment in a Genetic Learning System, in *Proc. Second Intl. Conf. On Genetic Algorithms and their Applications*, 202-209.

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

Kovacs, T., (1999), Strength or Accuracy? A comparison of two approaches to fitness calculation in Learning Classifier Systems, in Wu, A. (ed.), *Proceedings of the 1999 Genetic and Evolutionary Computation Conference Workshop Program*, 258-265.

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

Lanzi, P.L., (1998), Generalization in Wilson's XCS. In A. E. Eiben, T. Bäck, M. Shoenauer, and H.-P Schwefel, (eds.), *Proceedings of the Fifth International Conference on Parallel Problem Solving From Nature -- PPSN V*, number 1498 in LNCS. Springer Verlag, 1998.

Lanzi, P.L. Colombetti, M., (1999) An Extension to the XCS Classifier System for Stochastic Environments. In Banzhaf et al, 353-360.

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.

Saxon, S., Barry, A.M., (1999), XCS and the Monks Problems, in *Proc. Second Intl. Workshop on Learning Classifier Systems*, 272-281.

Venturini, G. (1994), *Apprentissage Adaptatif et Apprentisage Supervisé par Algorithme Génétique*. PhD Thesis, Université de Paris-Sud.

Wilson, S.W. (1994), ZCS, a zeroth level classifier system, *Evolutionary Computation 1(2)*, 1-18

Wilson, S.W. (1995), Classifier fitness based on accuracy, *Evolutionary Computation 3(2)*, 149-175

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