Limits in Long Path Learning with XCS

Alwyn Barry

Department of Computer Science
University of Bath
Claverton Down
Bath, BA2 7AY, UK
A.M.Barry@bath.ac.uk

Abstract. The development of the XCS Learning Classifier System [26] has produced a stable implementation, able to consistently identify the accurate and optimally general population of classifiers mapping a given reward landscape [15,16,29]. XCS is particularly powerful within direct-reward environments, and notably within problems suitable for commercial application [3]. The application of XCS within delayed reward environments has also shown promise, although early investigations were within enviroments with a comparatively short delay to reward (e.g. [28, 19]). Subsequent systematic investigation [19,20,1,2] have suggested that XCS has difficulty accurately mapping and exploiting even simple environments with moderate reward delays. This paper summarises these results and presents new results that identify some limits and their implications. A modification to the error computation within XCS is introduced that allows the minimum error parameter to be applied relative to the magnitude of the payoff to each classifier. First results demonstrate that this modification enables XCS to successfully map longer delayed-reward enviroments.

1 Background

Learning Classifier Systems ('*LCS*') are a class of machine learning techniques that utilise evolutionary computation to provide the main knowledge induction algorithm. They are characterised by the representation of knowledge in terms of a population of simplified production rules ('*classifiers*') in which the conditions are able to cover one or more inputs. The *Michigan* LCS [13] maintain a single population of production rules with a Genetic Algorithm ('*GA*') operating within the population ... each rule maintains its own fitness estimate. LCS are general machine learners, primarily limited by the constraints in the representation adopted for the production rules (see, for example, [29]) and by the complexity of solutions that can be maintained under the action of the GA [10].

LCS have been successfully applied to many application areas – most notably for Data Mining [22,14,3] but also in more complex control problems (e.g. [12,24])[1]. Although LCS performance has been competitive with the most effective machine learning techniques, it is notable that many of these cases (and all

[1] see [21] for further details of LCS applications

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2724, pp. 1832–1843, 2003.
© Springer-Verlag Berlin Heidelberg 2003

of those cited earlier in this paragraph) have been with LCS implementations that use direct reward allocation. In contrast, the application of LCS to complex delayed reward tasks has, until recently, been more problematic[2]. Recently a number of new LCS implementations appear to have overcome some of the instabilities of LCS when learning in delayed reward situations. ZCS [25] is a simplification of earlier LCS implementations and was designed for academic study, but [4,6] have shown that ZCS can *perform* optimally with appropriate parameter settings. [16] has argued persuasively that strength-based LCS implementations will inevitably lead to over-generalisation in tasks with biased reward functions. XCS [26] is an accuracy-based LCS derived from ZCS which overcomes many of these limitations, and the *Optimality Hypothesis:* [15] suggests that XCS will always identify a sub-population of accurate optimally general classifiers that occupy a larger proportion of the population than other classifiers. Bull argues that the fitness sharing mechanism of ZCS acts as a mechanism to prevent over-generalisation within ZCS, making ZCS a competitive strength-based LCS [5].

The effectiveness of XCS in its application to direct reward environments has been empirically demonstrated by many workers (for example, [26,15,29,3, 11]). Research into the performance of XCS within delayed reward environments has been more limited. [26,27] provided a proof-of-concept demonstration of the operation of XCS within the Woods2 environment. [17] identified that within certain Woods-like environments XCS was unable to identify optimum generalisations. This was attributed to two major factors: an inequality in exploration of all states in the environment allowing over-general classifiers to appear accurate, and an input encoding which meant that certain generalisations were not explored as often as others. [18] sought to apply these lessons to more complex Woods-based environments and discovered that XCS was additionally unable to establish a solution to the long chain Woods-14 problem [9]. This was due in part to the number of possible alternatives to explore in each state that prevented XCS from attributing equal exploration time to later states within the chain. It has been shown that XCS is able to learn this environment using an alternative explore-exploit strategy [6].

Whilst other work has investigated some of the more complex problems that delayed reward environments present, such as perceptual aliasing, there had been little investigation of the comparative performance of traditional LCS implementations and XCS in delayed reward environments beyond the simple Woods environments. Much more importantly, no work had attempted to identify the limits of XCS learning with increasing environment complexity or length when applied to delayed reward environments. [2] presented the results of an investigation of the abililty of XCS to form the population of optimally general classifiers mapping the payoff of environments of incrementally increasing length. It was shown that XCS performance was very good within the GREF-1 environment (100% performance within 1000 explorations – CFS-C [23] achieved 90% performance in 10,000 explorations). It was also shown that XCS can reliably learn the optimal route in a corridor environment (an extension of Fig. 1) of length

[2] There are many reasons for this . . . for a review of the issues see [1].

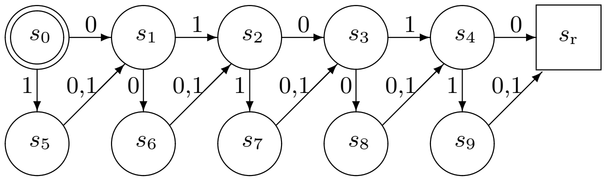

Fig. 1. A length 5 corridor Finite State World for delayed reward experiments

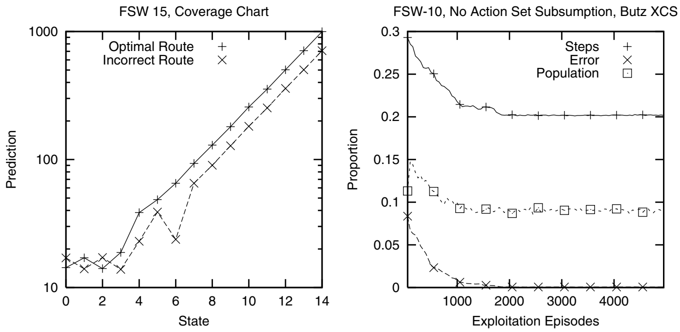

Fig. 2. (a) State × Prediction (log scale) coverage of the populations from 10 runs of Barry's XCS in a length 15 corridor environment after 15000 iterations; (b) Average of 10 runs of Butz's XCS in a length 10 corridor environment showing errors in the 'steps' plot

40 where generalisation was not used. However, once XCS was required to learn the optimal *generalisations* XCS was unable to reliably select the optimal route even within the length 10 environment. Learning performance deteriorated sufficiently thereafter to make investigation of performance in corridors longer than 15 steps superfluous (see Fig. 2a – the payoff prediction values of classifiers in the early states are highly confused). Subsequent investigations using an alternative XCS implementation [7] have confirmed these findings (see the perturbation in the 'steps' plot in Fig. 2b[3]).

Analysis of the population coverage of the environment indicated that XCS was unable to learn appropriate *generalisations* for early states in the environment. It was hypothesised that this was partially due to the reduction in payoff prediction values in classifiers covering these early states, making the prediction values sufficiently similar that XCS can identify a few over-general classifiers to cover the early states (see Fig. 2a). This conclusion caused some other workers to suggest that the problem might be resolved through the use of Wilson's modified accuracy measure or the use of an alternative subsumption algorithm (both introduced within [8]). These suggestions, though ill-founded, led to fur-

[3] XCS should rapidly identify the optimum number of steps to the reward, but instead often chooses at least one sub-optimal action in each corridor traversal

ther investigations into the cause of the over-generalisation within these early states.

2 XCS Learning in Delayed-Reward Environments

The *match set* ('$[M]$') is the set of all classifiers whose conditions *match* the current environmental input. XCS selects one of the actions proposed by $[M]$ and all classifiers in $[M]$ which advocate that action become members of the *action set* ('$[A]$') in the current iteration. When reward R is obtained by XCS in a delayed reward environment the reward is allocated to all classifiers in $[A]$. Clearly the reward could only have been reached as a result of earlier actions. So that the optimal path to the reward can be established XCS allocates payoff to classifiers within earlier action sets. In each exploration iteration the maximum action set prediction of [M] discounted by a discount factor γ is used as the payoff P in the update of the predictions p of the classifiers in the previous action set ($[A]_{t-1}$) using the following update scheme ($0.0 < \beta \leq 1.0$):

$$p' = p + \beta\,(P - p) \quad . \tag{1}$$

Over time all action sets will converge upon an estimate of the discounted payoff that will be received if the action was to be taken. The discount allows the estimate to take account of distance to reward as well as the magnitude of the reward so that a trade-off of effort to reward is inherent in the action selection. Unfortunately the discount also means that the payoff prediction becomes much smaller with distance from the reward. With the 'standard' XCS discount ($\gamma = 0.71$) the payoff reduces from the reward of 1000 to less than 10 within 14 steps ... the payoff in state n of an N state path to reward R is:

$$p_n = R\gamma^{N-n} \quad . \tag{2}$$

XCS is an accuracy-based LCS. Accuracy is a steep logarithmic function[4] of the error in the classifier's prediction of payoff:

$$\kappa = \begin{cases} \ln(\alpha)\,\frac{\varepsilon - \varepsilon_0}{\varepsilon_0} m & (\varepsilon > \varepsilon_0) \\ 1.0 & (\text{otherwise}) \end{cases} \quad . \tag{3}$$

(α and m are constants), where the error ε is calculated as:

$$\varepsilon' = \varepsilon + \beta\,(\varepsilon_{\text{abs}} - \varepsilon) \tag{4}$$

$$\text{where } \varepsilon_{\text{abs}} = \frac{|P - p|}{R_{\max} - R_{\min}} \quad . \tag{5}$$

($R_{\max} - R_{\min}$ is the reward range). Fitness is based on the *relative accuracy* of the classifiers appearing within each $[A]$.

[4] [8] introduce an alternative accuracy function which is calculated using a power function.

A cut-off parameter in the accuracy function, ε_0 (typically 0.01), is used to identify when the accuracy (κ) of a classifier is sufficiently close enough to 1.0 to be considered fully accurate (see Eq. 3). The conclusions in [2] failed to take into account the effect of this cut-off calculation of accuracy on the generalisation behaviour of XCS. Once the error in payoff prediction falls below $\varepsilon_0 R$ all predictions will be considered accurate. This is normally useful, filtering out noise in the payoff algorithm. However, in the early states the payoff prediction is sufficently small that ε_0 is a large proportion of the prediction. This will allow classifiers which advocate sub-optimal actions with a payoff prediction that is variant from the payoff by less than $\varepsilon_0 R$ to be considered accurate. When the difference in stable payoff values for the same action in two neighbouring states falls below this threshold it is possible to identify a single classifier to cover that action in both states. This classifier will be considered accurate even though there is a fluctuation in the payoff to the classifier. XCS will use this false accuracy to proliferate classifiers that generalise over successive states, so producing accurate over-general classifiers. That this is the case can be seen within Fig. 2a. In this environment the optimal route alternates between action 0 and 1. A single classifier is covering action 0 and another single classifier covers action 1 in states s_0 to s_3 so that the optimal action is not selected in states s_1 and s_3.

Although this might suggest that a solution to the problem would be to reduce ε_0, this is problematic because the noise created whilst seeking to identify appropriate generalisations will also prevent the identification of accurate classifiers. An alternative is to change the value of γ to reduce the amount of discount and keep the difference in neighbouring payoffs above $\varepsilon_0 R$. This may increase the length of paths that can be learnt, but as the level of discount is reduced the difference between neighbouring prediction values will decrease. It has been noted that where prediction values are close and there is an area of the environment where over-generals are encouraged, it is possible for large numerosity over-general classifiers to develop [2]. These classifiers use their numerosity to dominate the action-sets, reducing the payoff of the action-sets to a value similar to their prediction, thereby making themselves more accurate and giving themselves more breeding opportunities. It is therefore hypothesised that:

Hypothesis *Reducing the discount level will increase the number of steps over which XCS can learn the optimal state × action × payoff mapping, but there will be a point beyond which further reduction will cause the mapping to be disrupted by over–general classifiers.*

3 Experimental Approach

[1] introduced a test environment designed for testing the ability to learn an optimal policy as the distance to reward increases, whilst controlling other potentially confounding variables. This environment, depicted in Fig. 1 has many useful properties for these experiments – see [1]. In the experiments that follow the environment used will be labelled 'FSW-N', where N is the length of the optimal path from the start to the reward state.

Unless stated otherwise, each experiment uses the following XCS parameters: $\beta = 0.2, \theta = 25, \varepsilon = 0.01, \alpha = 0.1, \chi = 0.8, \mu = 0.04, p_{\mathrm{r}} = 0.5, P_{\#} = 0.33, p_{\mathrm{i}} = 10.0, \varepsilon_{\mathrm{i}} = 0.0, f_{\mathrm{i}} = 0.01, m = 0.1, s = 20$ (see [26] for a parameter glossary). Each experiment was repeated 30 times and the results presented are the average of 30 runs unless otherwise stated.

4 Investigating Length Limits

It was argued in §2 that the reason for the failure to learn the optimal solution in FSW-10 and FSW-15 was due to the small difference in action-set payoff as a result of the high discount value (γ). To demonstrate that this is the case XCS was run within the FSW-15 environment, changing γ to 0.75, 0.80, 0.85, 0.90, 0.95 in each successive batch of 30 experiments. For these experiments the maximum population size (N) was 1200 and the message size was 7 bits (for comparability with [2]). For each exploitation iteration the System Relative Error [1] was calculated, in addition to the number of steps taken in the environment and the population size. The results of each batch of 30 runs were averaged and the averages compared. Figures 3 and 4 show the results and coverage chart at discount values 0.80 and 0.95.

From 0.75 through to 0.90 the maximum System Relative Error reduces and the number of steps taken to achieve the reward moves towards the optimal route, with $\gamma = 0.95$ allowing the optimal route to be reliably selected. The identification that XCS can learn the optimal route in the FSW-15 environment given an appropriate discount factor provides an initial verification of the hypothesis. However, it is useful to question why a discount of 0.8 or 0.85 was not effective. The answer is partially revealed by an examination of Table 1. The difference between the predictions in s_0 and s_1 for discount 0.8 is 11 and the difference in payoff between the optimal and sub-optimal route in s_0 is 8.8 – below the ε_0 error boundary and therefore a candidate for generalisation. This is reflected in the results for $\gamma = 0.8$ – the 'iterations' plot shows one incorrect decision is taken in each episode (see Fig. 3). At $\gamma = 0.85$ the difference between

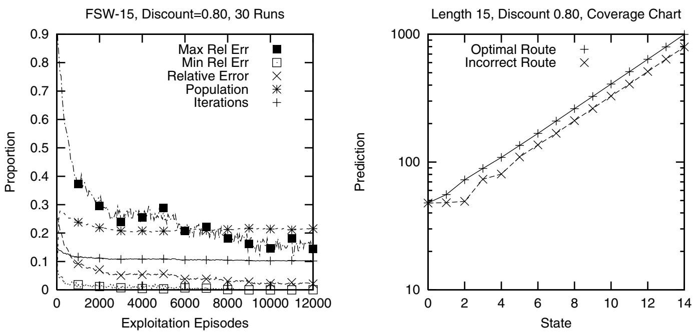

Fig. 3. Averaged results from 30 runs of XCS in FSW-15 at $\gamma = 0.80$

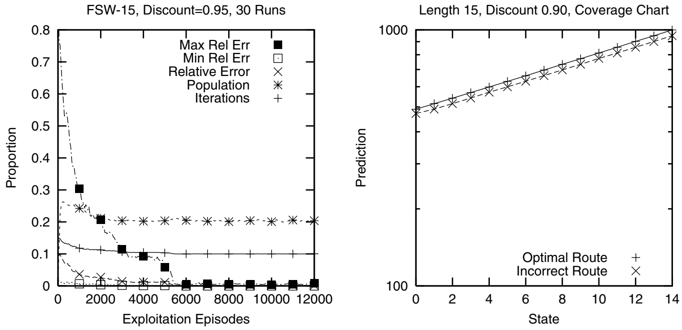

Fig. 4. Averaged results from 30 runs of XCS in FSW-15 at $\gamma = 0.95$

Table 1. Payout predictions in states within FSW-15 with different values of γ

γ	s_{0err}	s_0	s_1	s_2	s_3	s_4	s_5
0.71	5.87	8.27	11.65	16.41	23.11	32.55	45.85
0.75	13.36	17.82	23.76	31.68	42.24	56.31	75.08
0.8	35.18	43.98	54.98	68.72	85.90	107.37	134.21
0.85	87.35	102.77	120.91	142.24	167.34	196.87	231.62
0.9	205.89	228.77	254.19	282.43	313.81	348.68	387.42
0.95	463.29	487.68	513.34	540.36	568.80	598.74	630.25

the predicted payoff in s_0 and s_1 is increased. Unfortunately the coverage graph (not shown) indicates that difference of 15 between the optimal and non-optimal routes in s_0 is still sufficiently small to adversely influence the coverage of the sub-optimal route in s_0.

Now that it is clear that $\gamma = 0.95$ allows XCS to learn the optimal coverage of FSW-15, the second part of the hypothesis must be tested. This suggests that as the discount becomes small over-general classifiers will develop. It is worth noting that the proximity of the payoff values produced by $\gamma = 0.95$ were assumed by the author prior to the investigation to be sufficient to start to produce this generalisation. To investigate further, γ was systematically reduced to 0.99 in steps of 0.01, and then from 0.99 to 0.999 in steps of 0.001.

When the results were analysed, a clear pattern was evident. As γ was increased towards 0.99 the System Relative Error increased and more errors were evident in the number of steps taken to s_r. However, as it drew near to 0.99 and thereafter, the System Relative Error reduced even though the number of steps taken gradually moved towards that expected for a random selection of action in each state. An examination of the populations revealed that at 0.995 fully general classifiers dominated 7 of the 30 populations (see Fig. 5a) and at 0.999 all populations maintained high numerosity fully general classifiers. This is no surprise – the proximity of the predictions at 0.999 causes the difference in prediction to be well below ε_0 and the range of predictions is sufficiently small to allow a fully general classifier to easily gain sufficient numerosity to suppress the

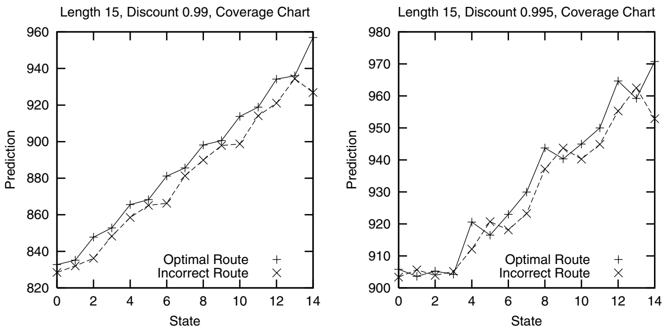

Fig. 5. Coverage charts for FSW-15 at $\gamma = 0.99$ and $\gamma = 0.995$

predictions and always be classed as accurate. What was surprising was the revelation that XCS was able to maintain such a reasonable coverage at a discount level of $\gamma = 0.99$.

Having demonstrated that XCS can perform optimally within FSW-15, it was appropriate to investigate how much the environment could be extended before XCS would once more perform sub-optimally. XCS was run with $\gamma = 0.95$ in FSW of length 20, 25, 30 and 40. If errors in coverage are guaranteed when the difference between payoff in s_n and s_{n+1} is 10.0 then a simple mathematical exercise would suggest that the practical limit is an environment of length 30. However it is now known that a difference of 15 was sufficient to prevent appropriate early state coverage, and this magnitude difference is seen between optimal and sub-optimal routes only 22 states from s_r. When the experiments were completed and the results analysed, is was found that XCS was able to find an optimal solution to FSW-20 at $\gamma = 0.95$. In FSW-25 the coverage chart remained almost optimal, and a careful analysis of each run showed that a single erroneous step was taken in 2.3% of the episodes after episode 3000. Within FSW-30 up to three additional steps were taken in each episode after episode 3000, and the coverage chart indicated generalisation over states up to s_6 (23 states from s_r).

These limitations present some difficulties. XCS has shown itself to be powerful in application to direct-reward problems, and yet apparently fundamentally limited in relatively small delayed-reward environments. However, it is important to understand that the problems arise not because of the basic mechanisms of XCS but due to the use of an *absolute* measure of error. As equation 4 indicates, error is computed relative to the reward range without taking account of the discount mechanism. A way to tackle this problem may be to identify an error measure that is independent of the position of the classifier in the action chain to the reward. A first attempt at such a solution was devised. This involved a simple modification to XCS to retain an estimate of the distance d from the reward within each classifier. This was used to calculate an action set estimate of distance to the nearest reward and to calculate the error as a proportion of

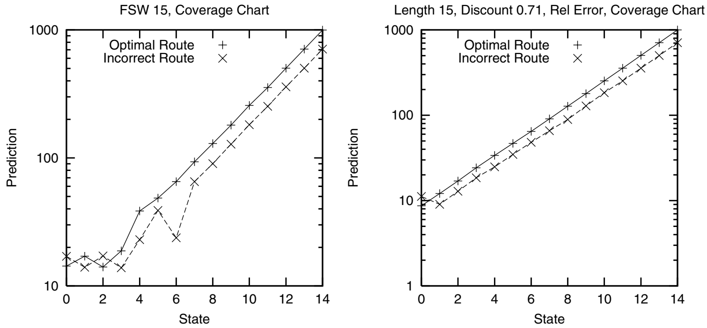

Fig. 6. Comparison of coverage charts for absolute error calculation and relative error calculation in FSW-15 at $\gamma = 0.71$

$\gamma^{N-d}R$. Although this technique resulted in a reduction in the System Relative Error, considerable error remained within the calculations, due to the magnification of errors in the estimate of d by the calculation. It is possible that reducing d to an integer value within this expression may mask these errors.

An alternative approach replaced the absolute calculation of error with:

$$\varepsilon = \begin{cases} \frac{|P-p|}{\max(P,p)} & (\max(P,p) > 0) \\ 0.0 & (\text{otherwise}) \end{cases} . \tag{6}$$

As the prediction becomes more accurate, the relative difference between P and p will reduce and so the error will become small independently of the magnitude of the payoff P. This, and an alternative error calculation: $\varepsilon = \frac{|P-p|}{p}$ (capped to 1.0 if $\varepsilon > 1.0$), have been the subject of recent investigations. Figure 6 shows that the use of the relative error update method reduces the over-generalisation in states above the state s_0. In FSW-15 at $\gamma = 0.71$ the two relative error expressions appear to produce similar results, although the second should provide less variance in the initial updates.

The results of applying a relative error calculation are encouraging, though further investigations are now required to identify any penalties that may be present in the formulation of the optimal sub-population of accurate classifiers. It would appear that the use of relative error makes the identification of accurate classifiers more problematic leading to greater divergence in the population, although this does not appear to dramatically affect the time taken for XCS to identify the optimal route. The use of a more focused GA selection technique, such as Tournament Selection, may resolve this problem.

5 Discussion

The limitations on the length of delayed reward environments are highly constraining. Delayed reward environments are commonly of much greater size within the laboratory, let alone within 'real-world' applications. In defence of XCS it should be noted that the corridor environment used within these tests is highly artificial. Richer environments may provide more distinct payoff gradients which will enable XCS to work over a larger range. It should also be noted that changes in parameters alongside modification to γ may affect the performance. For example, in experiments conducted alongside this investigation it was found that using the within-niche mutation scheme of [8] produced a much weaker performance because it encouraged an early decision on the most accurate generalisation and so aided the formation of over-general classifiers. Mutation schemes that encourage more diversity will act as a pressure against generalisation, and so enable XCS to map the environment using more specific classifiers. This is hardly a desirable solution, however.

The results are interesting in the light of [16]. This identifies that all non-trivial delayed reward environments are environments which encourage the development of over-general classifiers. Whilst XCS, as an accuracy-based LCS, has some protection against over-general classifiers, it is clear that the formation of over-generals will be encouraged as soon as there is a failure of the accuracy function to distinguish between payoff boundaries. The lack of provision for discount within the error calculation leads to an inequity of accuracy computation that encourages such a failure.

Whilst the results presented on the use of relative error are promising, more work is required in order to identify the dynamics of the calculated error in various parts of the action chain, and to identify the effect of the measure on the ability of XCS to satisfy the Optimality Hypothesis. The use of a relative measure of error should allow the length of paths that can be optimally mapped to be extended, but new limits must be established.

It is recognised that any discounted payoff scheme will cause hard limits in the length of path learning. Therefore work towards autonomous problem-space subdivision, the autonomous identification of sub-goals and their use in hierarchical planning remain important research objectives. Many other Reinforcement Learning methods face related problems in long path learning, and lessons can be drawn from these areas. However, the limits this paper has sought to address are those generated by the requirement to identify the minimum set of input generalisations that produce an accurate condition $\times$ action $\times$ payoff prediction mapping. The combination of accuracy-based learning and the requirement for the production of an optimally compact and accurate mapping is unique to XCS.

References

1. Alwyn Barry. *XCS Performance and Population Structure within Multiple-Step Environments*. PhD thesis, Queens University Belfast, 2000.
2. Alwyn M. Barry. The stability of long action chains in xcs. *Journal of Soft Computing*, 6(3–4):183–199, 2002.

3. Ester Bernadó, Xavier Llorà, and Josep M. Garrell. XCS and GALE: a Comparative Study of Two Learning Classifier Systems with Six Other Learning Algorithms on Classification Tasks. In *Proceedings of the 4th International Workshop on Learning Classifier Systems (IWLCS-2001)*, pages 337–341, 2001. Short version publishe in Genetic and Evolutionary Compution Conference (GECCO2001).
4. Larry Bull. On using ZCS in a Simulated Continuous Double-Auction Market. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, pages 83–90. Morgan Kaufmann, 1999.
5. Larry Bull. On accuracy-based fitness. *Journal of Soft Computing*, 6(3–4):154–161, 2002.
6. Larry Bull, Jacob Hurst, and Andy Tomlinson. Mutation in Classifier System Controllers. In J.A. Meyer et al., editor, *From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, pages 460–467, 2000.
7. Martin V. Butz. An Implementation of the XCS classifier system in C. Technical Report 99021, The Illinois Genetic Algorithms Laboratory, 1999.
8. Martin V. Butz and Stewart W. Wilson. An Algorithmic Description of XCS. Technical Report 2000017, Illinois Genetic Algorithms Laboratory, 2000.
9. Dave Cliff and Susi Ross. Adding Temporary Memory to ZCS. *Adaptive Behavior*, 3(2):101–150, 1994. Also technical report: ftp://ftp.cogs.susx.ac.uk/pub/reports/csrp/csrp347.ps.Z.
10. M. Compiani, D. Montanari, R. Serra, and P. Simonini. Learning and Bucket Brigade Dynamics in Classifier Systems. *Special issue of Physica D (Vol. 42)*, 42:202–212, 1990.
11. Lawrence Davis, Chunsheng Fu, and Stewart W. Wilson. An incremental multiplexer problem and its uses in classifier system research. In Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson, editors, *Advances in Learning Classifier Systems*, volume 2321 of *LNAI*, pages 23–31. Springer-Verlag, Berlin, 2002.
12. Marco Dorigo and Marco Colombetti. *Robot Shaping: An Experiment in Behavior Engineering.* MIT Press/Bradford Books, 1998.
13. John H. Holland, Keith J. Holyoak, Richard E. Nisbett, and P.R. Thagard. *Induction: Processes of Inference, Learning, and Discovery.* MIT Press, Cambridge, 1986.
14. John H. Holmes. A genetics-based machine learning approach to knowledge discovery in clinical data. *Journal of the American Medical Informatics Association Supplement*, 1996.
15. Tim Kovacs. Evolving Optimal Populations with XCS Classifier Systems. Master's thesis, School of Computer Science, University of Birmingham, Birmingham, U.K., 1996.
16. Tim Kovacs. *A Comparison and Strength and Accuracy-based Fitness in Learning Classifier Systems.* PhD thesis, University of Birmingham, 2002.
17. Pier Luca Lanzi. A Study of the Generalization Capabilities of XCS. In Thomas Bäck, editor, *Proceedings of the 7th International Conference on Genetic Algorithms (ICGA97)*, pages 418–425. Morgan Kaufmann, 1997.
18. Pier Luca Lanzi. Solving Problems in Partially Observable Environments with Classifier Systems (Experiments on Adding Memory to XCS). Technical Report 97.45, Politecnico di Milano. Department of Electronic Engineering and Information Sciences, 1997.

19. Pier Luca Lanzi. Generalization in Wilson's XCS. In A.E. Eiben, T.Bäck, M. Shoenauer, and H.-P. Schwefel, editors, *Proceedings of the Fifth International Conference on Parallel Problem Solving From Nature – PPSN V*, number 1498 in LNCS. Springer Verlag, 1998.
20. Pier Luca Lanzi. An Analysis of Generalization in the XCS Classifier System. *Evolutionary Computation*, 7(2):125–149, 1999.
21. Pier Luca Lanzi and Rick L. Riolo. A Roadmap to the Last Decade of Learning Classifier System Research (from 1989 to 1999). In Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson, editors, *Learning Classifier Systems. From Foundations to Applications*, volume 1813 of *LNAI*, pages 33–62, Berlin, 2000. Springer-Verlag.
22. Alexandre Parodi and P. Bonelli. The Animat and the Physician. In J.A. Meyer and S.W. Wilson, editors, *From Animals to Animats 1. Proceedings of the First International Conference on Simulation of Adaptive Behavior (SAB90)*, pages 50–57. A Bradford Book. MIT Press, 1990.
23. Rick L. Riolo. The Emergence of Coupled Sequences of Classifiers. In J. David Schaffer, editor, *Proceedings of the 3rd International Conference on Genetic Algorithms (ICGA89)*, pages 256–264, George Mason University, June 1989. Morgan Kaufmann.
24. Robert E. Smith, B.A. Dike, R.K. Mehra, B. Ravichandran, and A. El-Fallah. Classifier Systems in Combat: Two-sided Learning of Maneuvers for Advanced Fighter Aircraft. In *Computer Methods in Applied Mechanics and Engineering.* Elsevier, 1999.
25. Stewart W. Wilson. ZCS: A zeroth level classifier system. *Evolutionary Computation*, 2(1):1–18, 1994.
26. Stewart W. Wilson. Classifier Fitness Based on Accuracy. *Evolutionary Computation*, 3(2):149–175, 1995.
27. Stewart W. Wilson. Generalization in evolutionary learning. Presented at the Fourth European Conference on Artificial Life (ECAL97), Brighton, UK, July 27-31., 1997.
28. Stewart W. Wilson. Generalization in the XCS classifier system. In John R. Koza, Wolfgang Banzhaf, Kumar Chellapilla, Kalyanmoy Deb, Marco Dorigo, David B. Fogel, Max H. Garzon, David E. Goldberg, Hitoshi Iba, and Rick Riolo, editors, *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 665–674. Morgan Kaufmann, 1998.
29. Stewart W. Wilson. Mining Oblique Data with XCS. Technical Report 2000028, University of Illinois at Urbana-Champaign, 2000.