Is XCS Suitable For Problems with Temporal Rewards?

Kai Wing TANG, Ray A. JARVIS
Intelligent Robotics Research Centre
Department of Electrical and Computer Systems Engineering
Monash University, Victoria 3800, Australia
{Kai.Tang,ray.jarvis}@eng.monash.edu.au

Abstract

XCS [1], the accuracy-based classifier system, provides a very brilliant way to merge genetic algorithmic (GA) rule learning and reinforcement learning (RL) methodologies together. This makes it suitable for a wide range of applications where generalisation over decision making states is desirable. Also, its Q-learning-oriented prediction update scheme enables it to handle multi-step problems adequately. This paper reports how the intertwined spirals problem, initially a popular benchmark in classification, was modified by the authors to verify XCS's suitability for behavioural design of robotic systems. When the results obtained were not as expected, investigations were continued until a rather surprising conclusion was drawn: XCS cannot handle very simple problems if the rewards are temporally-oriented, even if the reward is extremely short-delayed.

1 Introduction

The Learning Classifier System (LCS) proposed by Holland [2] is a rule-based machine learning paradigm that, by mapping the input stimuli to output actions, and through an evolutionary mechanism, an agent can learn to adapt to new or changing environments automatically. However, LCS has its limitations. The first is its inability to produce optimally generalised rule sets [1]. The second is its bucket-brigade algorithm [3] cannot handle a very long-delayed reward. Alternatively, the XCS introduced by Wilson [1] provides a dynamic niching technique and an accuracy-based fitness which can produce optimally general co-operative rule sets. Moreover, Wilson [1] demonstrated, by using the Woods2 environment, that XCS can handle multi-step problems adequately. In summary, the major advantages of XCS are:

- XCS executes the genetic algorithm, i.e. new rule generation and weak rule deletion, in niches defined by the match sets. Accordingly, the best rule in each niche will be found without being influenced by selection pressures imposed by the other un-related rules.
- XCS can evolve classifiers that are optimally general subject to an accuracy criterion.
- XCS updates the predictions of a set of classifiers with a Q-learning-like [4] mechanism that the complete mapping of State by Action to Payoff (X x A $\Rightarrow$ P) will converge to stable Q-values. This enables XCS to handle multi-step problems effectively.

The authors' research interest is in applying the genetic algorithmic paradigm to evolve behaviours of a swarm of simple robots to collectively perform a task with a certain complexity, e.g. a team of robots are gathered at a disaster site. They are required to run a search and rescue operation, i.e. look for any entrapped victims and bring all found victims back to a refuge. Each of these robots is specialised in one type of action only. In other words, a robot can either be a rescuer or an explorer. We want to take a rule-based behavioural approach, i.e. for each type of robots, there is a rule set that controls their actions. The format of a rule is:

if {certain conditions are met} *then* {perform an action}

Naturally, XCS is a very attractive option. However, we are well-aware that any multiple agent system (MAS) can become a complex system and multiple robot systems are no exception [6]. For a complex system, the non-linearity introduced by actions amongst agents in a system will make the collective effects very difficult to analyse. Furthermore, XCS is an evolutionary approach which requires substantial computational resources and long time to construct solid results. If after running for a long period and no good result is realised, it is near to impossible to identify where the problem(s) come from. Consequently, we put the features of MAS aside first, set up an experiment to test XCS's performance about the formation of a rule set which has to be optimal under a preset criterion.

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 IEEE

This paper is a report about how the experiment was set up, how to further test XCS's capability in handling multi-step problems, what the results were and why the results were different from the other papers in the literature.

This paper consists of five other sections. Section 2 describes the setup of the first experiment and its results; section 3 is about the enhanced multi-step experiment. Section 4 explains the results and why further experiments were constructed to double verify our explanation. Section 5 is an investigation of the cause of differences from those of Wilson [1] and Barry [5]. Section 6 deals with the discussion and conclusions.

2 Testing of Niche and Generalisation

The criteria of the experiment used for testing the XCS's features of niched GA and generalisation are:

- The problem should be very simple, preferably with a very small number of states and actions, as well as a narrow range of payoff.
- The results can be easily visualised.
- There is a well-defined measurement scheme to evaluate the results.
- The effectiveness of the resultant rule set has to be measured by their collective outcome, instead of the effects of individual rules.

Owing to the second and third criterion, we decided to employ a popular benchmark problem originally designed for testing supervised classification algorithms as our experiment. It is the intertwined spirals problem [7]. This problem consists of two classes of points placed in two interlocking spirals that go around a central point as shown in fig.1.

There were a total of 194 points, the ratio of the two classes were 1:1. The aim of this experiment was to find a set of 2D sectors (fig. 2), each of which covered a number of points. Each sector had to correctly identify the class of its covered points. The criteria were:

- All points had to be covered by at least one sector. This required a cooperative result.
- The classification of a sector had to be optimally general and correct, i.e. a sector should not cover misclassified points and should cover as many points as possible. While a sector was represented by a classifier, this criterion required the classifiers to compete within some niches, instead of panmictically. Also, an effective generalisation method to enable every classifier to cover the possibly maximum number of points was required.

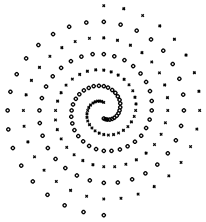

Figure 1. The pattern of intertwined spirals. Two classes of points are represented by 'x' and 'o', respectively.

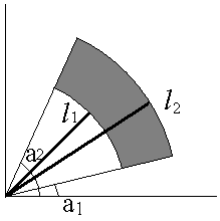

Figure 2. A sector is defined by four polar coordinates, i.e. angular difference $a_2 - a_1$, length difference $l_2 - l_1$.

2.1 Encoding of a Classifier

In order to ensure that a sector at least covered a point, the condition of a classifier was encoded in five alleles (fig. 3).

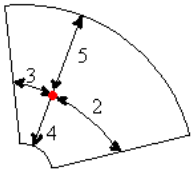

Figure 3. Five alleles of the condition of a classifier: a point and its four distances relative to the edges of the sector in the polar coordination system.

1. point: the index of one of the 194 spiral points.
2. angle-before: the angular difference between the point and the lower-angle edge of the sector.
3. angle-after: the angular difference between the point and the higher-angle edge of the sector.

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 IEEE

4. length-before: the distance between the point and the arc closer to the origin.

5. length-after: the distance between the point and the arc further to the origin.

All these alleles were encoded in integers. The value of the point was in the range of 1..194. The total angular difference of a full circle, i.e. 2 PI, was divided into 64 units and the difference between radii of the outer- and innermost spiral points was divided into 128 units. As a result, both angle-before and angle-after were in the range of 1..64; whereas the range of length-before and length-after was 1..128.

The action of a classifier was to nominate a binary state. It represented the act of a classification: declared in what class a state was, i.e. either 'x' or 'o'.

The crossover operation was one-point crossover; run at a randomly selected allele position. The mutation was run as replacing alleles of a condition, or an action, with another valid value.

2.2 State and Reward

A state was one of the 194 points. A matched set was the group of classifiers (sectors) which covered the state. The speciality of a classifier was the count of points covered by that classifier. Classifier A was more general than classifier B if A covered all points which were covered by B, and A's speciality was larger than B's. Two classifiers were equal if they covered the same set of points.

The rewards were in a balanced form. If a state was correctly classified, the reward would be +1.0. On the other hand, a reward of -1.0 would be returned if the classification was wrong.

New states were generated randomly, i.e. a series of random numbers in the range of 1..194 were serially passed to the XCS. The corresponding rewards were used to update the prediction, the prediction error and the fitness of every classifier.

2.3 Implementation of the Experiment

To run this experiment, we used the implementation of XCS developed by Butz[8]. The subroutines about rule matching etc. were modified to handle the format of classifier described earlier. All the parameters were retained as in the original package. A maximum of three hundred (300) classifiers were used. In the beginning, half of them were generated randomly. The number of classifiers grew from 150 to 300 gradually as the XCS sought for new, better classifiers to cover all 194 states. The prediction and the fitness of new classifiers were initialised to 10.0 and the prediction error to 0.0. The exploration - exploitation ratio was 1:1.

2.4 Results

The experiment was run 5 times and the results were consistent. The plot of average fitness and average prediction error versus spiral point classifications is as shown in fig. 4. It is obvious that both quantities converged steadily to a narrow range.

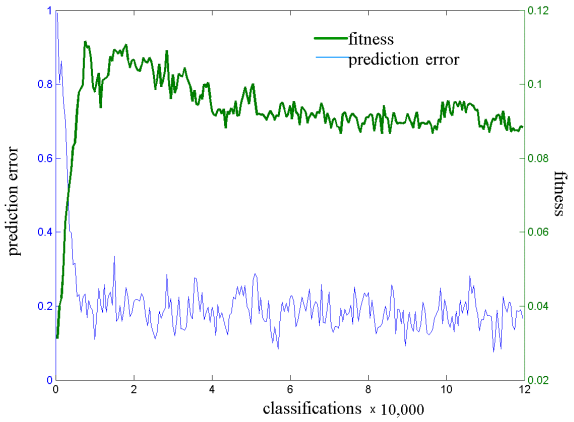

Figure 4. Convergence of average fitness and average prediction error.

After 120 thousand spiral point classifications, amongst the 300 classifiers, all of the macroclassifiers with a numerosity [1] higher than 4 were picked out and showed in fig. 5. We found that the two sets of classifiers, 'x' and 'o', complimentary covered nearly all the points. No misclassified points were included in any sectors. These results proved that the XCS can be used to effectively evolve a correct and general enough rule set.

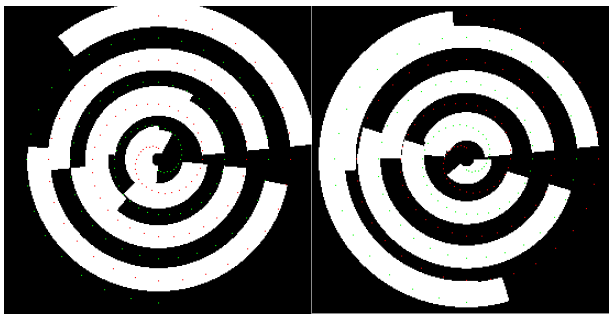

Figure 5. Two sets of classifiers which nearly covered all the points.

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 IEEE

3 Testing of Multi-Step Problems

For most co-operations executed by a multiple robot system, the performance is measured in terms of the collective result of the team. Once again, taking the search and rescue operation as an example, the objective of the collaborative operation is to search the whole disaster site and bring back all found victim(s) within a shortest period. The performance will then be evaluated based on how fast all found victims are rescued and how soon the whole environment is searched. It cannot be measured by summing up the size of areas searched by each explorer and how many victims are saved by each rescuer. In other words, within a complex system the total is not equal to the sum. As a result, one and only one environmental reward will be available after the operation is completed; and this is a very long-delayed reward.

The experiment described in the above section was a single-step problem because a reward was returned immediately after a spiral point was classified. A minor change was sufficient to convert it into a multi-step problem: the return of reward was deferred until a fixed count of spiral points were classified, then the sum of rewards was passed to the XCS.

Intuitively, we should defer the return of reward until all 194 points were classified. However, we preferred the experiments be run in a progressive mode. Consequently, we deferred the reward for one step only. That is, after two spiral points were classified, a reward in the range between -2.0 and 2.0 was returned, dependent on the results of the two classifications.

The plot of average prediction error versus spiral point classifications is as shown in fig. 6. This time the prediction error alternating continuously, failed to converge.

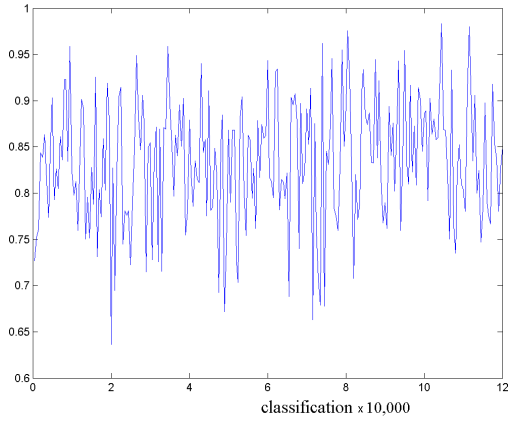

Figure 6. The average prediction error failed to converge.

That was a surprise to us. At that moment, we suspected that the state space formed by the 194 points was too huge for the Q-learning-based mechanism of XCS to handle. Consequently, we reduced the number of points by half and re-tried the experiment. Disappointingly, the same effects were resulted. We continued to reduce the number of points down to 7, i.e. 4 'x' and 3 'o', but yet no improvement was realised.

We studied the prediction updating method of XCS again and found the reason. It was due to the aliasing state problem [5].

The prediction of j-th classifier, p_j, in the action set of the previous time step is updated by the formula:

$$p_j \leftarrow p_j + \beta(P - p_j) \qquad \text{where } (0 < \beta \leq 1)$$

If there is no external reward, P is the maximum prediction of the previous time step's action set, P(a_i), multiplying by a factor $\gamma(0 < \gamma \leq 1)$. If there is an external reward, then the external reward is added to P before the predictions of classifiers are updated.

For this experiment, the reward obtained by classifications of point A followed by point B was the same as point B followed by point A. For the former case, the action set of point A was updated by $\hat{Q}$ values, whereas the action set of point B was update by a genuine reward value. On the contrary, the latter case was the opposite. Since external rewards and action sets of both cases were the same, but the updated values were drastically different; these materialised in a form of the aliasing state problem: the existence of more than one identical state, which requires the same action but returns with different rewards.

4 Testing the Existence of Aliasing

In order to further confirm the reason of this failure to converge, another experiment was set up to study the prediction error of some particular classifiers, instead of the average of the whole population. Since the effect of averaging had to be minimised, the number of classifiers in the population had to be reduced as well. Accordingly, this experiment was the simplest classification problem whose details were as follows:

There were only three points in two classes:

A='x', B='x', C='o'

Fourteen classifiers were sufficient to represent all combinations of groupings. The first seven were as shown in table 1, we named them as cl-1 to cl-7. Their action was 'x'. Similarly, the action of cl-8 to cl-14 would be 'o'.

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 IEEE

condition	name
A	cl-1
B	cl-2
C	cl-3
A,B	cl-4
A,C	cl-5
B,C	cl-6
A,B,C	cl-7

Table 1. Classifiers cl-1 to cl-7

In order to emphasise the effect of alternate point classification, the sequence of points passed to the XCS was a repeated cycle of:

(A,B),(B,A),(A,C),(C,A),(B,C),(C,B)

For this experiment, only fourteen classifiers were in the population and the features of action subsumption and rule discovery of XCS were switched off. Similarly, the prediction and the fitness of these classifiers were initialised to 10.0 and the prediction error to 0.0.

4.1 Results

As a control, we first ran the experiment as a single-step problem and plotted the convergence of prediction error of cl-4 and cl-7, i.e. the most correct and the most general, classifiers (fig. 7). The curves showed that prediction error of cl-4 dropped down to zero in less than 100 classifications.

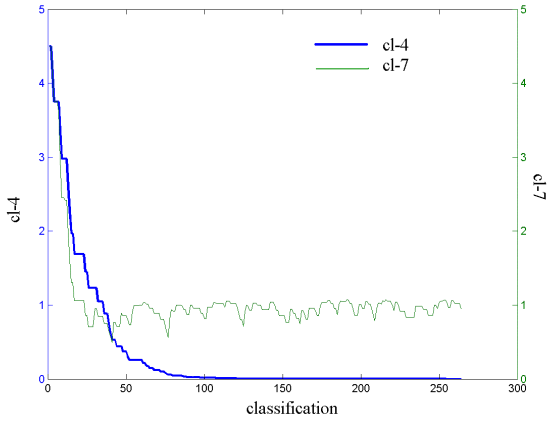

Figure 7. The convergence of prediction errors of cl-4 and cl-7 in a single-step problem.

Then we ran the experiment as a double-step problem and plotted the prediction error of cl-4 versus number of classifications (fig. 8). This time it showed that the prediction error was fluctuating within the range of 0.0 to 1.0.

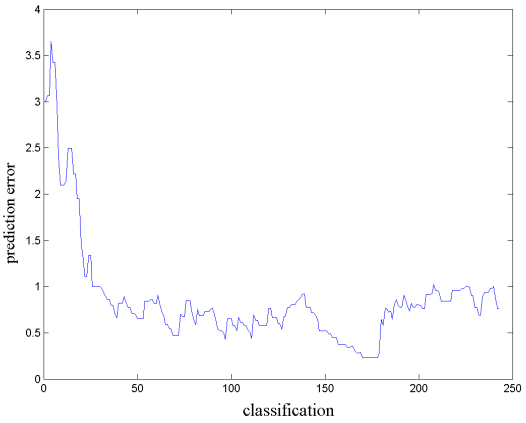

Figure 8. The prediction error of cl-4 was fluctuating in a double-step problem.

With the aim to clarify whether the learning rate, β, and the discount factor, γ, had any effect on this result, these two parameters were varied and more trials were run. The results confirmed that changes of these parameters could not achieve convergence.

5 Comparisons to Other Experiments

The above results demonstrated that XCS cannot handle some very simple multi-step problems. The question now becomes: why Woods2 of Wilson [1] was so successful and Corridor of Barry [5] could show that XCS can handle delays up to ten steps?

After a comparison between our problem and these two other problems, we found that their ways of returning rewards were drastically different. The rewards of both Woods2 and Corridor were spatially dependent, while our classification problem was temporally dependent (figs. 9, 10 & 11).

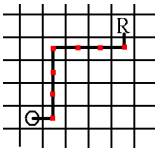

Figure 9. Woods2 problem: a reward will be given if the animat (circle) reaches the food (R).

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 IEEE

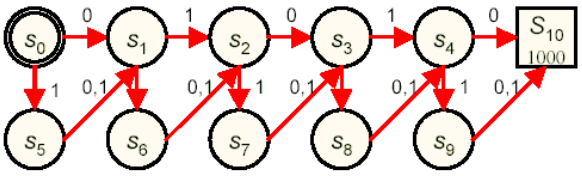

Figure 10. Corridor problem: a reward will be given if S_{10} is reached.

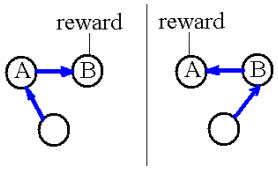

Figure 11. Our problem: a reward will be given whenever two actions are executed.

XCS employs a mechanism similar to Q-learning to estimate the payoff of steps in-between two external rewards. This method presumes that the sequences of states traversed are distinctively unique; therefore, just by updating the $\hat{Q}$-value of the previous action, the effect will be backwardly propagated and eventually cause convergence to the true Q-value. However, once this presumption cannot be guaranteed, this method will become totally dysfunctional.

As shown in fig. 11, predictions of either state A or B change alternately in the left- and right-handed scenarios. Therefore, we can conclude that aliasing will be present hand-in-hand with temporal rewards.

6 Discussion and Conclusions

The last questions before the conclusions of this paper will be: is the problem quoted in this paper a very artificial one and, for most of the real world multi-step problems, is XCS still highly applicable? Unfortunately, the answer is no. Temporal reward is a very common payoff scheme. In the robotics field, just use a single robot exploration task as an example: a robot which can move in eight different directions is placed in an unknown environment. It is required to plot a map of the environment. First of all, the task is to seek for any significant geographic markers which can be used as landmarks. During this operation, the robot may have to decide whether continue to moving ahead, or turn to another direction, to explore. This decision will base on how long since the current direction has been taken. Consequently, the rewards will be time dependent. In the natural world, a honey bee uses the comparison of energy consumed versus nectar collected to assess the possibility to re-try a route. Energy consumption also is a temporally related quantity.

This can be fixed by adding internal memory to XCS [9]. However, if the reward is very long-delayed, the convergent time as well as the number of interim states will increase geometrically.

Finally, the conclusions are: The features of niching and generalisation of XCS are proven to be very powerful for our classification problem. However, its reinforcement learning mechanism makes it unsuitable for problems where temporal rewards are present. For those problems whose nature is not fully understood, i.e. without solid knowledge about whether the rewards are spatial- or temporal-oriented, applications of XCS may bear the risk of failure of convergence.

References

[1] S.W. Wilson. ”Classifier fitness based on accuracy”, Evolutionary Computation, Vol.3, 1995, pp. 149-175.

[2] L.B. Booker, D.E. Goldberg & J.H. Holland. ”Classifier Systems and Genetic Algorithms”, Artificial Intelligence, Vol. 40, Nos. 1-3, 1989, pp. 235-282.

[3] D.E. Goldberg, ”Genetic Algorithms in Search, Optimization, and Machine Learning”, Addison-Wesley, 1989.

[4] C. Watkins & P. Dayan, ”Technical note: Q-Learning”, Machine Learning, Vol.8, 1992, pp. 279-292.

[5] A. Barry, ”Limits in Long Path Learning with XCS”, Proc. GECCO 2003, Genetic and Evolutionary Computation Conference, 2003, pp. 1832-1843.

[6] R. Beckers, O.E. Holland, and J.L. Deneubourg, ”From Local Actions to Global Tasks : Stigmergy and Collective Robotics”, Proc. Artificial Life IV, MIT Press, 1994, pp. 181 - 189.

[7] K.J. Lang & M.J. Witbrock, ”Learning to Tell Two Spirals Apart”, Proc. 1988 Connectionist Summer School. Morgan Kaufman, 1988.

[8] M. Butz, C-XCS: An implementation of the XCS in C. (http://www.cs.bath.ac.uk/amb/LCSWEB/computer.htm), 1999.

[9] P.L. Lanzi, ”Solving problems in partially observable environments with classifier systems”, Tech. Rep. 97.45, Dipartimento di Elettronica e Informazione, Politecnico do Milano, IT.

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 IEEE