Evaluating The XCS Learning Classifier System In Competitive Simultaneous Learning Environments

Neera P Sood
The Mitre Corporation
7515 Colshire Drive
McLean, VA 22102-7508
1-703-983-7515
nsood@mitre.org

Ashley G. Williams
The Mitre Corporation
7515 Colshire Drive
McLean, VA 22102-7508
1-703-983-6113
Ashley@mitre.org

Kenneth A. De Jong
George Mason University
4400 University Drive
Fairfax, Virginia 22030
1-703-993-1553
kdejong@gmu.edu

ABSTRACT
We would like to evaluate the XCS [1] Learning Classifier System (LCS [2]) to see if it can be applied to a specific aviation industry problem. We are interested in seeing whether it can offer an accessible representation model and evolve feasible strategies to predict future demand patterns endogenously, and in parallel with the supply side simulation.

Categories and Subject Descriptors
D.3.3 [**Distributed Artificial Intelligence**]: – *Multi-Agent Systems, Machine Learning.*

General Terms
Algorithms, Economics, Experimentation.

Keywords
Machine Learning, XCS, LCS, Reinforcement Learning, Prediction accuracy.

1. INTRODUCTION
Agents selling goods or services in a competitive market need to adapt to the environment if they are to be successful. In a dynamic market, agents may appear and disappear and individual bidding strategies may change, so it is difficult to know which strategy to apply in a particular situation which would maximize profit. Hence agents need to learn their opponents' strategies and adapt to the ever-changing market. In a competitive market, agents often face each other in encounters in which the simultaneous actions of a set of agents lead to different utility payoffs for all the participants. For example, a set of agents might submit their bids in an auction and, depending on the outcome of the auction, each agent may experience a utility gain or loss. Agents learn from the results of the auction and over time begin to improve their performance.

GECCO'05, June 25–29, 2005, Washington, DC, USA.

Accurate classifiers are classifiers that give an accurate prediction of the payoff that the agent should expect when using that particular classifier. In order to build an effective, accurate learning classifier system, various combinations of learning and multi-strategy learning algorithms need to be explored. For example, we would like to investigate whether agents learn to cooperate to forgo short term gain and increase mutual long term rewards. These strategies in the current state of the environment can then be input into a Learning Classifier System (XCS LCS) [1] [2] which would improve upon them, evolve them and return them to the environment to be used in the next state. The motivation for this study is to determine the applicability of these techniques to a specific aviation industry problem. The complexity of this problem is so great with all users making multiple, interdependent decisions simultaneously, that a Learning Classifier System appears to be the ideal solution for it. Similar work has been done by others such as, Schulenburg et al. [3] in the stock market environment. We would like to extend the work performed by them, as, their assumptions that the individual decision does not affect the overall output of the system does not apply to the complex world of the aviation industry. In this study, we are interested in examining not only different agent learning strategies, such as reinforcement learning strategies but also the effectiveness of accurate classifiers in combination with them using the XCS [1] framework. We would then like to test this framework with other market strategies, such as the Cournot and Bertrand solutions and finally our own

2. PROBLEM
At many large airports, demand far exceeds capacity resulting in costly delays and unpredictable operations. As an interim solution, until additional facilities can be built, the Federal Aviation Administration (FAA) has recently begun investigating new resource allocation policies. To assure Congress and the flying public that any new policy will be fair and efficient, the FAA needs quantitative results to back up their proposal with reasonable certainty that it will have the intended effects without substantial side effects. In current practice, analysts typically make assumptions about how the market will respond to the new policy. The relevant metrics are then computed to inform

decision makers about the relative merits of various policies. However, because any new resource allocation policy constitutes a significant deviation from the current economic and operational environment, simply extrapolating the market, or "demand" response from existing demand patterns would be a gross simplification of the problem and would most probably result in misleading conclusions. In fact, any solution that requires an assumption about the future demand pattern would most likely suffer from the analyst's bias.

The MITRE Corporation's Center for Advanced Aviation System Development (CAASD) is building the MarketFX™ model to predict future demand patterns endogenously, and in parallel with the supply side simulation. By modeling the market-driven behavior of all direct aviation facility users (the airlines), the new environment in which they must operate and the passenger responses to airline schedule changes, the MarketFX™ model produces a high-fidelity simulation of the profit maximizing behavior of the airline industry while making very few high level assumptions.

This is a particularly difficult problem because of the structure of the airline industry. The industry consists of many airlines with unique schedule networks, business models, and cost structures. As a result, they typically value the same resources differently and, consequently, respond to new policies and capacity changes differently. If we add to that the fact that all users are making multiple, interdependent decisions simultaneously, it soon becomes apparent that the profit maximization problem is simply intractable using traditional optimization techniques. The MarketFX™ model solves this problem by coupling agent-based simulation with the latest machine learning and numerical techniques to model the airline planning process.

Currently, MarketFX™ uses an adaptive regression algorithm to model the decision makers at every level within an airline. The agents require feedback from the environment in the same form as the decisions being made. In other words, if an agent makes a decision regarding fare, it will need to know what the "best" fare really was in the previous state; if it was a departure time agent, and it will need the "best" departure time; etc. The obvious shortcoming for this domain is that we don't really know what the best decisions should have been. To get around this limitation in the latest design, we have transformed all decision variables into a profit prediction – a known quantity in every situation. However, we have really had to bend the tool quite a bit to get it to do so. Therefore, we would like to explore an alternative machine learning technique that uses reinforcement learning (directly) so that our airline agents can be simple profit maximizers.

Another feature of the domain is that the search space is enormous, if not infinite. Therefore, we need (1) a means of transforming states into various classifications and (2) an efficient means of finding feasible strategies. Learning classifier systems satisfy both of these requirements and additionally offer an accessible representation so results can be better understood (which is useful since validation will be difficult).

3. BACKGROUND

Reinforcement Learning (RL) [4] techniques and Genetic Algorithms have been applied successfully in many agent-based systems for learning the policy of an agent in uncertain environments. Reinforcement learning can be applied to estimate the quality value for each state-action pair. Genetic algorithms help create populations of possible solutions out of which the fittest are selected [5]. Genetic algorithms may be combined with reinforcement learning. The incoming reward is exploited to guide the evolution of the agent's behavior which, in learning classifier systems, is represented by a set of update rules, the classifiers. The idea is to estimate the goodness/fitness of classifiers and use genetic algorithms to favor the reproduction and recombination of better classifiers, better suggestions of actions, that should be taken or better strategies that should be followed.

Using these techniques, although many successful results have been achieved in the past in a Markov environment, composed of a finite set of states and actions, we do not find many applications of these techniques in a non-Markov environment, such as the Market structure, possibly composed of many different states. Some evidence of their success in abruptly changing environments such as the world of economics may be found in [6] and [7]. To effectively use these techniques to solve real world practical problems we must also extend them to non-Markov environments.

A detailed description of the XCS and the algorithms it uses may be found in [1] and [8]. We would like to use the XCS [1] for the following reasons:

- It is an LCS that uses Reinforcement Learning
- It evolves optimal and accurate representations or condition/action rules
- It has the ability to generalize condition/action rules

As described in [9],

"learning classifier systems evolve a set of condition-action rules by measuring the performance of individual rules and then periodically using crossover and mutation to breed new rules from old".

Why use the XCS [1] in particular? We recognize that the XCS [1] was intended to be used in a Markov environment. However, others, such as [10] and [11] have applied it in a non-Markov environment,

"Corporate classifier systems develop chains of classifiers which potentially can be exploited for tackling non Markov problems and to anticipate the consequence of the classifier action" [11].

Given the fact that our problem is a non-Markovian problem, we would also like to test its effectiveness in dealing with a non-Markov environment. Another reason for using the XCS may be found in [8],

"I suggest that XCS systems can evolve optimal populations (representations); populations which accurately map all input/action pairs to payoff predictions using the smallest possible set of non-overlapping classifiers".

Yet another significant reason is that,

"in a competitive market, agents are primarily driven by the need to maximize profit on a daily basis, but in the real world more complex objectives such as maximizing market share may also affect agent behavior [12].

As also mentioned in [9],

"in early learning classifier systems, rules occasionally did an action that earned external reward, and this contributed to the

rule's fitness and to the fitness of those that enabled it to fire. Earned rewards were spread by the so-called `bucket brigade algorithm' (effectively a trickle-down economy) or `profit-sharing plan' (essentially a communal reward-sharing) or other such algorithm. However, in those early systems, a rule's fitness was a measure of the reward it might earn (when considering what rule to fire) and also a measure of the reward it had earned (when selecting rules for breeding). This caused various problems, notably that rules which fired very rarely but were crucial when they did would tend to be squeezed out of the population by the evolutionary competition long before they could demonstrate their true value. XCS [20] largely fixed this by instead valuing a rule for the accuracy rather than the size of its prediction of reward. For this reason - because, in our application, there might be heuristics which were rarely used but crucial - we chose to use XCS" [9].

And finally, according to [7],

"The particular implementation we choose (XCS) has another interesting property: their favor the accuracy of strategies instead of their brute strength. This enhances the expectation dimension of LCSs. This is a very important dimension for modeling learning because many evolutionary models neglect the formation of expectations by the firms and their influence on decisions (see Oltra & Yildizoglu [1999])..... Our results show that the use of XCS pays in terms of the efficiency of the learning process even if they are quite demanding in computational power. Industries formed by XCSFirms are more efficient at the level of technology dynamics, as well as of social welfare."

We hope to conduct experiments to see if they result in the creation of a number of classifiers each suited for a particular target/problem/environment. By comparing and evaluating these classifiers or combinations of the different learning strategies under the same real-world conditions, we hope to create a single "best" robust classifier or learning system.

4. METHODOLOGY

As the above-mentioned evidence shows, this area is certainly not new. Others have already tested it. Our work borrows ideas from their experience in this area. However, before we can reach the stage they have already arrived at and before we can apply the above mentioned technology to our current work, we need a better understanding of how agents develop, evolve and work. How does learning take place in competitive environments? Which experimentation strategy produces most effective learning [5]? We plan to use a system in which learning takes place not only in the external environment but also in the XCS. The external environment in our system is a heterogeneous environment in which agents with different strategies, whether they are reinforcement learning strategies, such as Q-Learning or Temporal Difference Learning or other economic strategies such as the Cournot and Bertrand simultaneous matrix game solutions [Becker] interact with the XCS which helps them improve their current strategies.

Given the complex and competitive situation that we are dealing with in the airline industry, the application of Game Theory appeared to be most suitable for us. We have already experimented and still are experimenting with competitive game theory composed of simultaneous moves by agents, with a view to applying it to and exploiting "market competition". Research literature [6], [13], [14], [15] and [16] shows that The Iterated Prisoner's Dilemma (IPD) problem has established itself as a central model in competitive multi-agent environments. The prisoner's dilemma is a classic problem of conflict and cooperation. In its simplest form each of two players has a choice of cooperating with the other or defecting. Depending on the two players' decision, each receives payoff according to a payoff matrix. The problem is made more interesting by playing it repeatedly with the same group of players, thereby permitting partial time histories of behaviour to guide future decisions [6]. Here, we emphasize that our goal is to continue the work performed by others, since it is based on their work already done in this field and their advice, and use their insight into this complex problem by trying to make it successful in a truly multi-agent environment. We would like to investigate whether agents learn to cooperate to forgo short term gain and increase mutual long term rewards. We hope that the potential for cooperation exists in certain environments with the characteristics of games such as the Prisoners' Dilemma. Can multiple adaptive agents learn to behave in ways analogous to real world bidding strategies? How do changes in the market structure affect agent behavior? This is what we would like to find out.

We started by first experimenting with different learning strategies, since we are interested in agents evolving optimal strategies. Since we did not want to build a whole new system, we downloaded and adapted the GENEIPD system [14], [15] to the XCS [1] in order to build our experimental framework of a heterogeneous competitive market-like environment. The GENEIPD system already has the IPD agents built into it and therefore provides us with a convenient multi-agent framework for our experiments. Our heterogeneous agents play in this market, each using its own strategy and then periodically, after a certain number of games, new strategies are created by invoking the XCS [1] GA from the GENEIPD system. The results of the experiments conducted so far are presented below.

5. EXPERIMENTS

Since we are concentrating on agents learning good machine learning strategies we first started by experimenting with different strategies. We have conducted experiments with different reinforcement learning techniques as well as economic profit game theory strategies in order to ascertain which ones produce effective results in a competitive environment.

The next step was to experiment with The Iterated Prisoner's Dilemma strategies, with many agents. As mentioned above, we believe that this has potential application in a non-Markov environment in which the agents make simultaneous moves in order to outdo each other. The idea was to test whether the XCS could truly evolve better strategies to produce a more cooperative agent behavior. As mentioned above, we are using the GENEIPD system [14], [15] integrated with the XCS [1]. We have experimented with the ALLC (all players cooperate), ALLD (all players defect), TFT (Tit-For-Tat players) and ATFT (All Tit–For-Tat players). Each agent suggests an action based on its past experience and feedback. The genetic algorithm, i.e. the GA as a higher level agent then picks between them. The function of the

GA is to find the value of the update rule (x) which maximizes f(x). The results of our experiments are presented below.

6. Experiments with Reinforcement Learning Methods

These experiments were performed in a Markov environment using the simple Game of NIM [17]. NIM is an old, simple game in which two agents take turns in picking up tokens from a pile. There are a number of versions of NIM. We use the version in which the player who has to pick up the last token loses. If you are in a winning state, you get a 1.0 reward. If you are in a losing state, you get a -1.0 (negative) reward. Otherwise the reward is 0.0. We explored some of the basic issues in machine learning. How agents learn when using Q-Learning [18] and Temporal Difference (TD) [18] methods. The Q-Learner and the TD-Learner play the game of NIM against each other each using its own strategy. Figures 1 and 2 show that the Q-Learner is a stable learner, whereas the Temporal Difference learner can behave erratically depending on the learning rates.

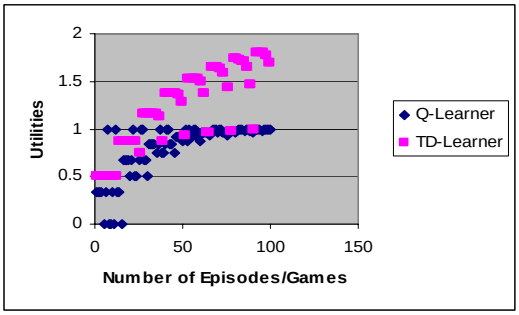

Figure 1 - Agents learning utilities over time - Alpha = 0.5

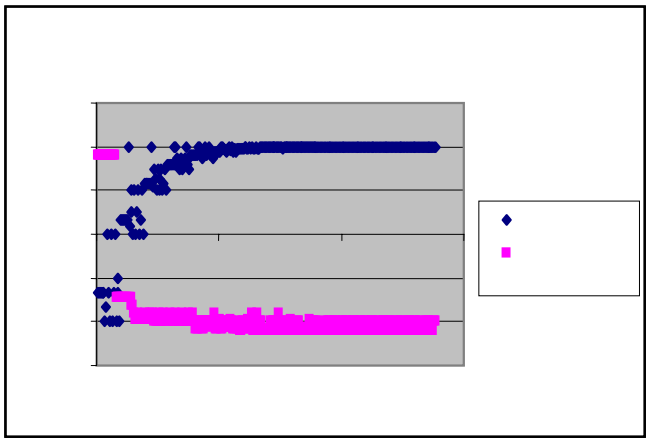

Figure 2 - Agents learning utilities over time - Alpha = 0.9

7. Experiments with Market games (Beckman) using the optimal policy

These experiments are based on matrix choice games illustrating monopoly, shared monopoly, Cournot and Bertrand behavior and price wars. The games use a profit table given the choices of two players. One player selects the column, the other the row, and the table gives the profit of the row chooser.

The following results were obtained by actually making the agents go through a learning phase in which they choose an optimal policy in actual market conditions. We use Beckman's market games with Cournot and Bertrand solutions. Figure 3 shows the results of agents playing Profit Games using the Cournot solution. It is based on the "best response" policy. A simple demand function, **P = (Q1+Q2)**, where Q1 and Q2 are the output choices of two agents. The profits reported are just **P*Q1**. This policy eliminates dominated strategies.

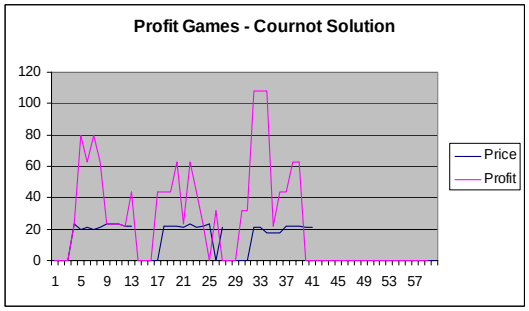

Figure 3 - Agent Price and Profit

Figures 4, 5, 6 and 7 show the Bertrand solutions.

The idea here is that whichever firm produces the least determines sales. The price of the combined bundle is given by: **P = 12 – min(Q1,Q2)**. The price is given by: **P = 12 – min(Q1,Q2)** and the profit is **P *min(Q1,Q2).**

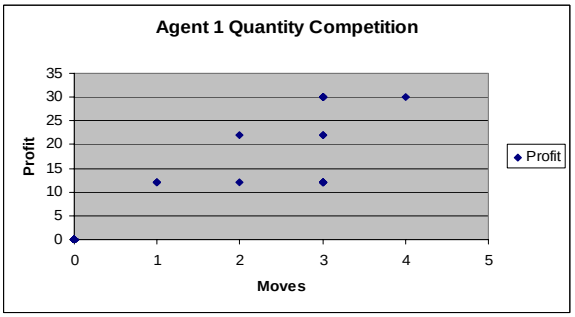

Figure 4 - Profit vs. Moves

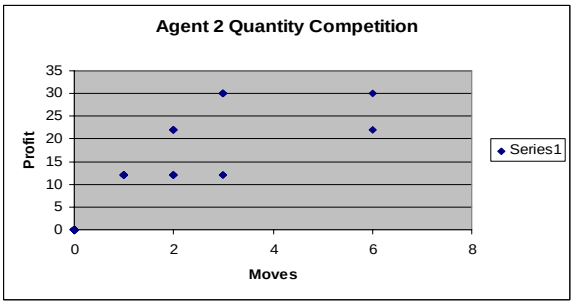

Figure 5 - Profit vs. Moves

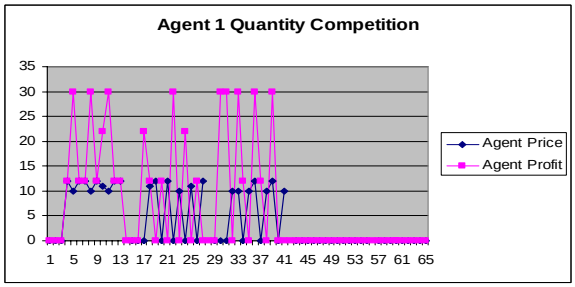

Figure 6 - Profit vs. Price

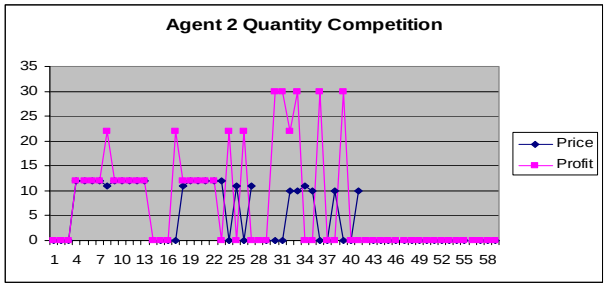

Figure 7 - Profit vs. Price

8. Experiments with the Iterated Prisoner's Dilemma strategies

The aim of this experiment was to see whether it was possible to change a dominant strategy environment into a more cooperative one using the XCS. Before we go on to describe our experiments and the results that we have achieved so far, we would like to describe the representation of our problem. Suppose the memory of each player is one previous turn. There are four possibilities for the previous turn:

CC (case 1)

CD (case 2)

DC (case 3)

DD (case 4)

Table 1. Strategy Evolution – Tit For Tat

Actions	Strategy
CC	C
CD	D
DC	C
DD	D

If CC (case 1), then C.

If CD (case 2), then D.

If DC (case 3), then C.

If DD (case 4), then D.

This results in the following strategy: CDCD

Axel rod's [6] tournaments involved strategies that remembered three previous turns. Thus a strategy can be encoded by a 4x4x4 = 64-bit string, e.g., CDCCCDDCCCDD [6], [16]

We first ran the original system, GENEIPD [14], [15], the way it is originally set up with the five IPD rules. Each agent is run in the current GENEIPD environment and uses the strategy defined by its chromosome to play an IPD with other strategies. This produces results that show that Defector agents always win in competitive situations. We then modified the system to use the XCS [1]. An examination of the *chromosomes* shows that the XCS produces improved *chromosomes* favoring the Cooperation agents. An example of the results are presented below, where:

For 0 = C = Cooperate

and 1 = D = Defect

After a game/competition: One parent chromosome before going into the XCS system:

IPD sending chromosone (current state) to XCS: 0010101111100000110

Before the next game/competition: One child chromosome after the process of crossover and mutation from the XCS system to the IPD system:

XCS sending modified chromo = 0000101110100000000

Figures 8, 9, 10 and 11 show the results for the defector and cooperator agents.

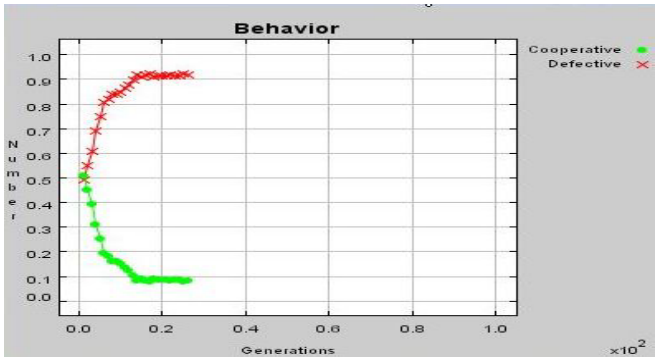

Figure 8 - IPD Agents before XCS processing

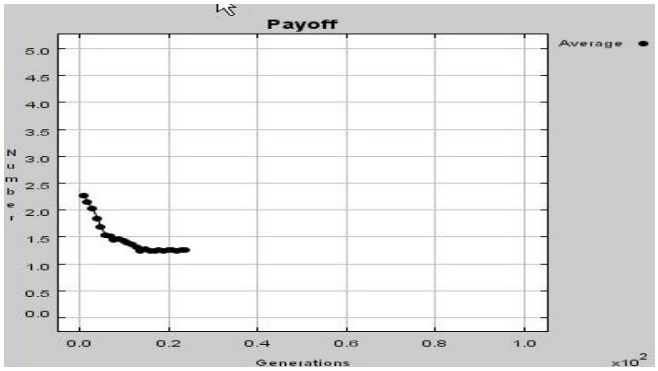

Figure 9 - Payoff before XCS processing

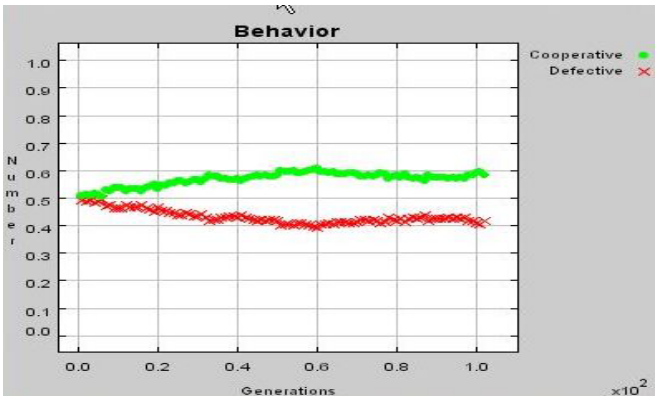

Figure 10 - IPD Agents after XCS processing

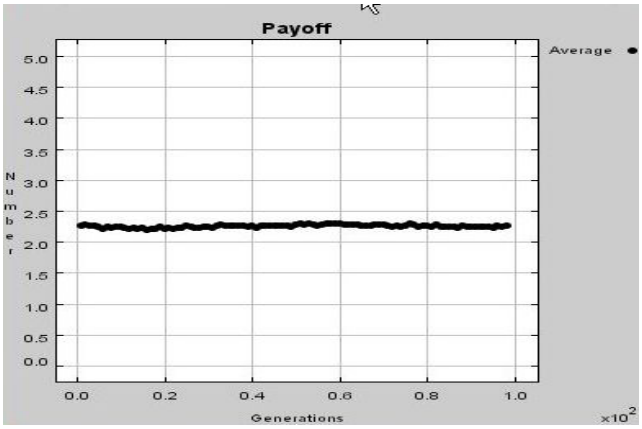

Figure 11 - Payoff after XCS processing

9. CONCLUSION

We have shown that because of its prediction accuracy and the ability to generalize condition/action rules, it is possible for the XCS to generate improved strategies. We have presented here the first set of our experiments. We now plan to experiment with other strategies, such as the different reinforcement learning strategies and other profit game strategies. We are currently in the process of encoding these for input into the XCS. Our work is by no means yet complete. However, we are optimistic about these first results. They show that it is possible for the XCS [1] to come up with better rules in competitive environments. Our future plans are to test the XCS system with our own rules.

10. ACKNOWLEDGMENTS

We gratefully acknowledge the assistance and support provided by The MITRE Corporation for this research. We are especially grateful to Ashley G. Williams of The Mitre Corporation for providing a domain application for our experiments. We are also grateful to Dr. Gary L. Klein of The Mitre Corporation, Dr. Michael C. Tanner of The Mitre Corporation, David B. Smith of The Mitre Corporation and Dr. Kenneth B. Samuel of The Mitre Corporation for providing an insight into the complexity we are trying to resolve. Finally, we are grateful to Dr. Kenneth De Jong from George Mason University for his guidance during the experiments performed and continue to be performed in the area of Reinforcement Learning and Genetic Algorithms.

11. REFERENCES

[1] Martin V. Butz, Stuart W. Wilson, *An Algorithmic Description of XCS*, Institute for Psychology III & Department of Computer Science butz@psychologie.uni-wuerzburg.de, University of Illinois at Urbana-Champaign, Prediction Dynamics, Concord, MA 01742 wilson@prediction-dynamics.com.

[2] Holland, *LCS*.

[3] Sonia Schulenburg and Peter Ross *An Adaptive Agent Based Economic Model*, – Artificial Intelligence Application Institute, Division of Informatics, University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, Scotland {sonias,peter}@dai.ed.ac.uk.

[4] Sutton & Barto, 1998.

[5] Tom M. Mitchell, *Machine Learning*

[6] Axelrod, *The Iterated Prisoner's Dilemma*.

[7] Murat Yıldızo˘glu *Modeling Adaptive Learning: R&D Strategies in the Model of Nelson & Winter (1982)* IFREDE-E3i Universit´e Montesquieu Bordeaux IV Avenue L´eon Duguit Fax. 335 56 84 86 47 e-mail: yildi@montesquieu.u-bordeaux.fr http://yildizoglu.montesquieu.u bordeaux.fr/May 2001 – Preliminary version v.0.5.

[8] Tim Kovacs, *The XCS Classifier System Reliably Evolves Accurate, Complete, and Minimal Representations for Boolean Functions* School of Computer Science, University of Birmingham Birmingham U.K. B15 2TT T.Kovacs@cs.bham.ac.uk

[9] Peter Ross, Sonia Schulenburg, *Hyper-heuristics: learning to combine simple heuristics in bin-packing problem* Peter Ross School of Computing Napier University Edinburgh EH10 5DT peter@dcs.napier.ac.uk Sonia Schulenburg School of Computing Napier University Edinburgh EH10 5DT s.schulenburg@napier.ac.uk Javier G. Mar__n-Bl_azquez Division of Informatics, The University of Edinburgh Edinburgh EH1 1HN, UK javierg@dai.ed.ac.uk Emma Hart School of Computing Napier.

[10] Alwyn Barry, *Aliasing in XCS and the Consecutive State Problem.*

[11] Pier Luca Lanzi, *An Introduction to Learning Classifier Systems* Artificial Intelligence and Robotics Laboratory Dipartimento di Elettronica e Informazione Politecnico di Milano.

[12] A. J. Bagnall, *A Multi-Agent Model of the the UK Market in Electricity Generation* School of Computing Sciences University of East Anglia Norwich, NR47TJ Englandajb@sys.uea.ac.uk.

[13] Dan Ashlocky, Mark D. Smuckerx, E. Ann Stanleyyz, and Leigh Tesfatsion *Preferential Partner Selection in an Evolutionary Study of Prisoner's Dilemma.*

[14] Luc Girardin, Prof. Lars-Erik Cederman, *Evolutionary Agents*, Luc Girardin Center for Comparative and International Studies (CIS) Seilergraben 49, Room G.1, girardin@icr.gess.ethz.ch Prof. Lars-Erik Cederman, CIS Room G.2, lcederman@ethz.ch http://www.icr.ethz.ch/teaching/compmodels Lecture, June 8, 2004.

[15] Prof. Lars-Erik Cederman, *Repast Tutorial*, Prof. Lars-Erik Cederman Center for Comparative and International Studies (CIS) Seilergraben 49, Room G.2, lcederman@ethz.ch Nils Weidmann, CIS Room E.3, weidmann@icr.gess.ethz.ch http://www.icr.ethz.ch/teaching/compmodels Lecture, December 14, 2004.

[16] JENNIFER GOLBECK, *Evolving Strategies for the Prisoner's Dilemma* Computer Science Department University of Maryland, College Park College Park, MD USA golbeck@cs.umd.edu http://www.cs.umd.edu/~golbeck.

[17] Charles Leonard Bouton, *"Nim, a game with a complete mathematical theory"* Ann. Math. Princeton, Series 2, Vol. 3, pp. 35-39 (1902).

[18] Stuart Russell, Peter Norvig *Artificial Intelligence, A Modern Approach.*

[19] Steven R. Beckman, *Cournot and Bertrand Games.*