Proceedings of the 2007 IEEE Symposium on
Computational Intelligence and Games (CIG 2007)

Concept Accessibility as Basis for Evolutionary Reinforcement Learning of Dots and Boxes

Anthony Knittel
Centre for the Mind
Main Quadrangle (A14)
University of Sydney
NSW 2006
Australia
Phone: +61 2 9351 5816
Fax: +61 2 9351 8534
email: anthony@centreforthemind.com

Terry Bossomaier
School of Information Technology
Charles Sturt University and
Visiting Fellow, Centre for the Mind
email: tbossomaier@csu.edu.au

Allan Snyder
Centre for the Mind
University of Sydney
email: allan@centreforthemind.com

Abstract— **The challenge of creating teams of agents, which evolve or learn, to solve complex problems is addressed in the combinatorially complex game of dots and boxes (strings and coins). Previous Evolutionary Reinforcement Learning (ERL) systems approaching this task based on dynamic agent populations have shown some degree of success in game play, however are sensitive to conditions and suffer from unstable agent populations under difficult play and poor development against an easier opponent. A novel technique for preserving stability and allowing balance of specialised and generalised rules in an ERL system is presented, motivated by accessibility of concepts in human cognition, as opposed to natural selection through population survivability common to ERL systems. Reinforcement learning in dynamic teams of mutable agents enables play comparable to hand-crafted artificial players. Performance and stability of development is enhanced when a measure of the frequency of reinforcement is separated from the quality measure of rules.**

Keywords: Concept Accessibility, Evolutionary Reinforcement Learning, Dots and Boxes

I. Introduction

Micro and nanotechnology are driving the need to develop software solutions using teams of intelligent adaptive agents. Such teams may be large and entirely self-taught. Evolvable systems are often robust, compared to explicit procedures, albeit at the cost of some efficiency.

Games provide a very useful test framework, since they can have relatively simple rule systems with simple easily measurable rewards yet very complex dynamics as discussed recently by Sato et al [1] for games such as Rocks Paper Scissors. Conway and Berlekamp have studied a wide range of seemingly simple games, showing that they can have very deep strategies [2]. The childhood game of dots and boxes, or its duel, strings and coins, is such a game. The game dots and boxes is played on a grid of dots, players take turns to complete an edge between two adjacent dots on the board, receiving a point and an additional move if the edge completes a box. When all edges have been filled the player with the most points wins.

Previous systems have been developed using dots and boxes as a test bed for exploring the development of rules that capture features and principles of the game environment, and use those rules to produce effective play [3] [4]. The main goal of this research is not simply to develop the most effective game playing system, but rather to explore techniques to efficiently capture structure in the observed environment and use that structure to produce effective behaviour. Previous systems have been able to produce effective play against a basic artificial player in a restricted domain, playing as the second player on a three by three size board, however with some degree of instability or sensitivity to parameterisation [3], [4]. An alternative approach is presented that produces more stable behaviour and better performance, by avoiding instabilities inherent in a performance-sustained agent population. This approach no longer follows the analogy of a growing/shrinking population common to Evolutionary Reinforcement Learning, as the resulting behaviour of the analogous system is no longer appropriate for the environment being explored. An alternative technique inspired by cognitive behaviour, based on principles of limited but stable cognitive resources is presented.

II. Game Playing System

The system used to play the game is an Evolutionary Multi-agent System, using a form of Reinforcement Learning. Agents in the system represent rules, and are developed using an evolutionary algorithm based on mutation, based on the systems described in [3] and [4].

Strategies for playing the game effectively have been described in detail by Berlekamp [5]. These strategies are based on features of the game state and require recognition of specific concepts, rather than detailed examination of quality values of states as is common in artificial game learning. The number of possible states and actions produced in game play makes it impractical for a basic state-based reinforcement learning approach, but regardless this research focuses on examining the automated development of conceptual struc-

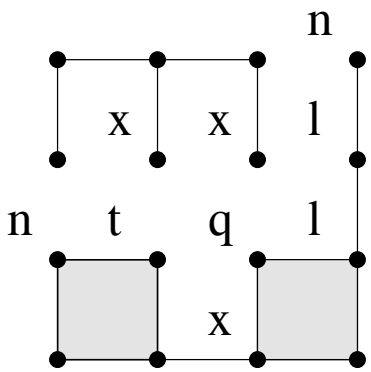

Fig. 1. A state of the game of dots and boxes, with degree labels added to squares on the board that are incomplete.

tures and rules to govern play, ideally based on rules related to those described by Berlekamp.

A. Rule Encoding

In the strings and coins view of dots and boxes, strategies can be represented in terms of connections of strings and coins, which effectively represent states in the game, or generalisations of states. Evaluating strategies to be used requires matching template (move) graphs, which may also contain generalisation symbols or other operators, against the current state graph of the game being played. The recent surge of interest in understanding networks [6], [7] and the characterising of common network fragments [8], [9] makes it of considerable interest to be able to evolve rule systems on graphs. Basic rules such as leaving chains unfinished to retain control may be represented using graph structures, although more sophisticated rules would require a broader rule encoding scheme.

B. Encoding Rule Graphs

In the following discussion we consider only the strings and coins representation (which is in fact a superset of the dots and boxes game [5]).

The pattern an agent uses to determine if it can make a move is a set of one or more unconnected graph fragments. Nodes in the graph represent boxes in the game state, labelled with the symbols q, t, l, x, n to represent nodes of degree 4-1 respectively, and "null nodes", used to represent connections at the edge of the board. An example of a game state with matching identifying symbols for each box is shown in Figure 1. Connected sequences of nodes of degree 2 are labelled in the rule graph with a single node, l, with an additional property describing the length of the chain. This does assist the learner in recognising *a priori* that chains are useful features to identify, and learning progresses from this assumption. Chain symbols in rule graphs also use an optional flag to allow the chain to match other chains of equal or greater length. An additional node symbol is used in rule graphs, w, which is a wildcard symbol to allow the rule graph to match part of a larger graph, as restrictions on the form of rule graphs enforce graphs to be complete- for example that a node with symbol q is connected to exactly four neighbours in the rule graph.

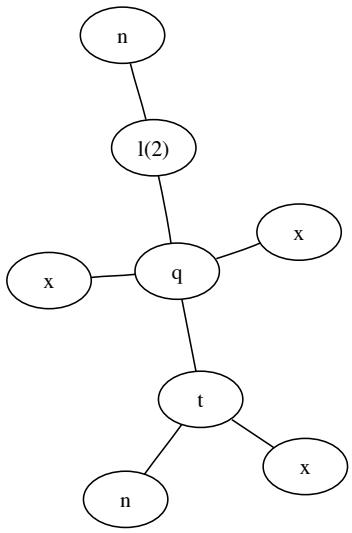

Fig. 2. A state graph produced from the game state in figure 1.

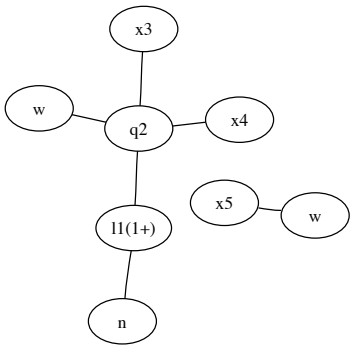

Fig. 3. Rule described by a number of graph fragments.

Rules also define an action to be conducted if the rule is activated, represented using a pair of nodes in the rule graph to be used as the action edge for the next move.

Figures 2 and 3 show a rule defined by a number of graph fragments, and a matching state graph.

Rules can also be described in textual format using an adjacency list representation. In this representation each node in the graph is listed along with all the nodes that node is adjacent to, for example the rule in Figure 3 can be described as:

$$q_2 l_1(1+) x_3 x_4 w : l_1(1+) q_2 n : x_3 q_2 : x_4 q_2 : x_5 w => q_2 - x_3$$

with the additional information of the edge to be completed if this rule is activated (between q_2 and x_3). Further details on the representation used are given in [3] and [4].

C. Mutations

New rules are created by copying the state graph from an existing state of the game, and child rules are created from existing rules using a series of randomly chosen mutation operations on the graph, including adding nodes, removing nodes, changing the properties of chain nodes, joining and separating graph fragments, and replacing symbols with wildcards.

III. Existing Systems

Previous studies have examined the use of an artificial economy to govern rule use and development [3], and reinforcement learning in a biological paradigm [4], where agents receive rewards for successful behaviour and die off if insufficient rewards are received. These techniques have shown good performance in the restricted game of playing always as the second player to move in the game, and adequate performance with the starting position being interchanged, winning up to 30% of games against a given artificial opponent. A number of limitations in the system used to maintain the rule population can lead to instability in on-line play against a difficult opponent, and over specialisation and niching (described following), such that rules that play well in a limited range of situations are preserved, and other more challenging game states are not addressed by the system, or receive limited development. These systems are sensitive to the degree of difficulty of the opponent- if the system is receiving limited rewards as a result of limited success against the opponent, the rule population is not sustained and the best rules that have been developed can be lost, and if the opponent is too easy and the system receives plentiful rewards, rules that are not very good are maintained and the system does not improve.

The instability of the agent population that has been shown is due to the use of an analogy with biological evolutionary systems to guide agent development. Populations grow according to input to the system and die off under poor conditions, such behaviour is inherent in the analogy used and practical problems that result from this show limitations of the applicability of the analogy for the problem being addressed, which is to develop concepts appropriate for playing a game.

Comparable systems developed in the field of Learning Classifier Systems (LCS) [10] and other Genetic Algorithms [11] handle population dynamics by maintaining a fixed population size, a useful abstraction of evolutionary development without placing too much consideration on the life span of individual units. This may be better for preserving population stability, however can still result in difficulties with specialisation and generalisation of rules [12]–[14]. Finding a balance between specialisation and generalisation has been addressed in a number of different ways and is a common problem in similar evolutionary systems, typically addressed through mechanisms such as selective deletion or reproduction [15], rather than an inherent point of stability of the underlying system.

A. Niching

A related problem, coverage versus niching, is influenced by the generalisation / specialisation properties of the system, an explanation of the terms used is presented for clarification.

a) Generalisation: refers to how reusable rules are, a highly generalised system uses a limited number of rules to cover a wide range of behaviour.

b) Specialisation: refers to rules with limited applicability. This allows specific behaviour to be captured that may be more appropriate for specific cases, but is not applicable to a wide range of cases.

c) Coverage: refers to how much of the problem domain is able to be handled by the system.

d) Niching: is the convergence of a system or components of a system towards a limited domain, typically such that behaviour within this domain is effective.

The problem of niching is inherent in the use of a population analogy where survival is based on a particular input. Such systems are guided simply by population survival, and in systems such as LCS where the domain of operation of components is limited, effective survival can be produced by restricting the domain of operation such that behaviour within this domain is effective, leading to niching. The problem with niching is that while components of the system are able to function effectively, there may be elements of the domain not effectively addressed by the system, such that the behaviour of the overall system observed externally is poor.

Highly generalised systems tend to produce good coverage and specialised systems tend to produce niching, however distinction between the concepts is important as the principles are independent. A high degree of coverage is essential for effective play in a game playing system, as a system that is drawn towards specific niches may operate effectively within those niches but poorly overall.

B. Specialisation of Rules

The problem of degree of specialisation of rules is important for the game of dots and boxes. It is possible to develop highly specific rules that recognise a particular action is useful or optimal in a particular situation, however due to practical (or design) limitations, and requirements of reusability in new situations, generalised rules are also necessary.

Specialised rules can be useful for capturing useful actions that occur infrequently, and special cases that are not appropriately captured by general rules. For example it is generally a good idea to take boxes whenever they are available, using a rule such as[1]:

$$x_0 w => x_0 - w$$

(described in adjacency list representation). However in specific cases, such as situations where it is beneficial to give up boxes to the opponent in order to retain control, a more specific rule, to play a different edge, can take precedent over

[1]This rule describes one open box connected to something else (wildcard), the right hand side describes a pair of connected nodes in the graph specifying the edge that should be completed as the action for the rule.

the rule to complete an available box, for example the rule fragment:

$$l_0(1+)x_1n => l_0(1+)[0] - n$$

This rule describes a long chain symbol (l_0) of length 1 or greater ($1+$), connected to a box and the edge of the board (x_1 and n), and states the action to play is the edge between l_0 and n, at the end of the chain ($[0]$ moves into the chain). Further refinements would be needed to define the conditions where the specific rule should take precedence.

1) Precedence: In order to implement an effective player it is necessary for precedence between the rules to be maintained, such that if a situation may be addressed by a general and a specialised rule, the specialised rule takes precedence. If the quality of rules is maintained using a Temporal Difference learning (or similar) procedure and the Q-value of the rule adequately represents the expected reward for using a rule, use of this value for selecting acting rules is sufficient for defining precedence.

Maintaining the population according to Q-values of rules, or expected reward, is problematic as it places preference on highly specialised rules which may occur infrequently, placing emphasis on the learner to remember specific moves in rarely occurring board states at the expense of general, reusable rules.

An alternative technique is presented that takes into account the frequency of use of rules, to encourage a degree of generalisation.

C. Reinforcement Algorithm

Existing evolutionary frameworks are commonly based on an analogy with genetic systems or living entities that survive according the presence of a given resource. The algorithm we present is based on a related but independent analogy, of the use and activation of concepts in the brain. Practical differences for the purposes of this study are limited, however the motivations of the design principles used are different.

Each agent in the system maintains a Q-value representing the expected reward when the rule is used, similar to the standard TD(0) learning procedure [16], [17]. An additional variable is maintained, f, representing the accessibility of a rule, or frequency of use. The purpose of this value is to discriminate rules such that only a limited number need to be assessed at a given time, even though new rules are being created regularly.

This value is maintained independently of the quality value, and reinforced according to how often a rule is accessed, regardless of its quality. Every time a rule is used its accessibility value is increased, and with time gradually decreases. The total amount of accessibility value in the system will stabilise, distributed amongst rules in a manner representative of the distribution of activity niches experienced by the system. This approach biases selection, and subsequently reinforcement, of rules according to the frequency of occurrence of states, providing a mechanism to enforce wide coverage of the system, and to avoid niching (section III-A).

Maintenance of available rules for assessment is conducted using the f-value, and selection using the Q-value. In this manner the Q-value can be used to independently represent the expected reward when a rule is used, allowing quality to be preserved independent of frequency of occurrence, and allowing precedence between rules to be developed as described in section III-B.1.

IV. Methods

A. Agent System

The game playing system is considered an Evolutionary Multi-Agent System, as it is composed of a number of independent units that perform actions (agents), which are developed using an evolutionary process. The agent system acts as a single player in the game, and the agents in the system are rules that suggest actions at various stages of play. This is a different structure to systems described in existing literature on Multi-Agent Reinforcement Learning, which commonly refer to the dynamics of a game being played by multiple agent players, each using an independent state-based reinforcement process [18], [19].

Reinforcement is performed on active rules using a procedure similar to TD(0) [16], for all rules that have acted during a game, according to the following formula where r is the reward according to the result of the game, and Q_n is the Q-value of agent n.

$$Q_n = \alpha r + (1 - \alpha)Q_n \tag{1}$$

The accessibility measure of each rule is updated according to:

$$f_n = d + (1 - \alpha)f_n \tag{2}$$

where $d = 1$ if the rule has been used, and $d = 0$ otherwise.

The agent population is limited to 1000 agents, restricted according to the f value of each.

For each state in the game, each rule in the system is tested if it matches the given state. From the matching rules the best ten are selected, for the purpose of maintaining a consistent selection probability according to the Q-value of rules, which can otherwise be distorted by large numbers of rules. The acting rule is selected probabilistically according to the Boltzmann distribution:

$$P_n = \frac{e^{\beta Q_n}}{\sum_{j=1}^{J} e^{\beta Q_j}} \tag{3}$$

B. Refinements

To assist the search process in covering the range of states experienced, new rules are added to the system probabilistically when few or no rules match a given game state. The new rule is added as a mapping of the game state to a random action. Generalisation and variation of the rule can occur through mutations in offspring of the rule, according to mutation rules that modify the graph describing the rule,

by adding, removing or reconnecting elements in the graph. Further details of the mutation rules are described in [3] and [4].

New rules are added to the system with Q-value equal to the average reward value experienced by the system. As a result rules which improve performance gain precedence easily, and the initial value does not need to be determined arbitrarily.

Reproduction is performed every 10 games, 4 agents are selected from the available population according to the same probabilistic distribution used for action selection described previously, new agents are produced as mutations of these agents. The rate of reproduction has been inherited from previous systems, where it was chosen to preserve an acceptable population size, maintaining a similar value is beneficial for comparisons between the systems.

Reward values are defined as the difference of scores at the end of the game, such that games lost give a negative reward. As selection of Q-values is determined using an exponential function, winning rules easily take precedence over rules that on average result in a loss.

C. Experimentation

Experiments are run by playing matches between the agent system and an artificial player, using a board of 3x3 boxes. The artificial player is a hand coded system following non-trivial rules such as the chain-rule, and plays at a good amateur level [20]. The starting player is determined randomly for each game, runs progress with no initial training and an initially empty set of rules. 10^6 games are played in an on-line mode, selecting rules probabilistically using various settings for the β parameter, and subsequently 10^4 games are played with a greedy method to assess the best performance of the existing rule set.

V. Results

Initial runs were tested using population selection according to the Q parameter of each rule, producing selection according to quality only. This caused the system to be highly niched, where each rule maintained by the system was highly specialised and referred to highly selective cases. This produced minimal overall performance ($< 1\%$ win rate) and showed activity on a very small percentage of game states (10%). Additional runs were performed using a combined parameter, where $Q_{n+1} = d\alpha r + (1 - \alpha)Q_n$, and $d = 1$ if the rule has acted, otherwise $d = 0$. This is identical to the metric described by Sato et al [1], although implemented in a different context. In this situation specialised rules that occurred infrequently were not maintained, as the Q value decreased over time when the rule was not being used. The system acted on most game states, however performance was limited at a consistent value of 10% [2]. Subsequent results describe the system operating using the parameters described in section IV, which allow a balance between specialised and generalised rules.

Using a constant value of $\beta = 10$, the system quickly and smoothly increased performance over time, to a maximum average value of about 0.4, indicating it won against the artificial opponent 40% of games played, as shown in Figure 4. Using a fixed schedule to set the β parameter, starting with $\beta = 0.1$ and increasing by increments of 0.2 every 10^4 games to a maximum value of 20, the system developed more slowly but reached a higher average performance value of 0.5, with a distribution of values between 0.45 and 0.55. Using a schedule of values starting at $\beta = 0.1$ and ending with $\beta = 10$, the system develops more slowly again and reaches a maximum performance value close to 0.35.

The activity rate of the system increased to above 99% after 1000 games, and remained close to 100% for the rest of the run. The activity rate is a measure of the number of unguided actions performed by the system, indicating how well the rules used by the agent cover the range of situations faced, showing that the rule set contained rules applicable for almost every state experienced by the system.

Examination of the rules used shows a diverse range of rules, some specific, some more general. The most common rules are generic rules such as completing a box:

$$x_0 n => x_0 - n$$

and playing into a long chain:

$$x_{45} l_{48}(2+) : l_{48}(2+) n x_{45} => x_{45} - l_{48}(2+)$$

A number of rules are also present in the system relating to specific moves to play in particular states, defined by rules with a range of degrees of specificity. An example of a highly specific rule is:

$$t_4 l_9 l_7 t_1 : l_7(2) t_4 n : l_9(4) t_1 t_4 : l_8(1) t_1 n : t_1 l_9 t_4 l_8 => t_1 - t_4$$

This rule[3] has a Q-value of 8.99, indicating that if this rule is able to act it will most likely lead to a large win. Other specific rules such as double-crossing the end of a long chain [5] were not seen in samples analysed, suggesting either alternative strategies have been used, such rules were not able to be supported with the given language, or those rules were simply not found.

Stable development was shown in every observed run, typically as a monotonically increasing level of performance, this was different to observations of previous systems which often collapsed to near-zero performance under difficult conditions.

VI. Conclusions

A varied reinforcement technique has been presented that is based on an analogy of accessibility of concepts in the brain, instead of genetic selection according to a fitness parameter. This reinforcement method has shown reliable development of a population of rules that has a balance of generality and specialisation. Instabilities in previous systems according to the difficulty of the environment, in this case

[2] Note that not all other conditions were the same in these experiments, these initial results are provided as a guide only.

[3] Parenthesised length-description elements have been removed from some symbols for clarity

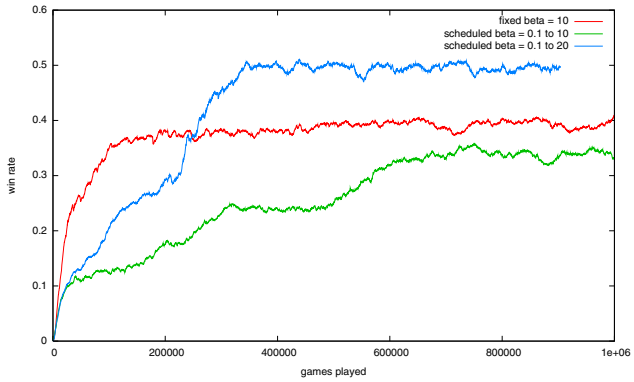

Fig. 4. Average performance of system with various schedules for β.

the difficulty of the opposing player or current rate of success against the player, are addressed, such that the best performing rules are preserved regardless of difficulty.

The language used to represent rules allows effective play against a basic opponent and versatile, reusable rules, although development of more sophisticated strategies will require a more versatile rule representation. Overall the system was able to develop a level of performance equal to the artificial system over a period of more than 2 times 10^5 games, being played on-line, showing the system has developed a strategy and level of play as good as the opponent it is trained against.

The most significant result is that the system produces stable and consistent development of rules, showing the Concept Accessibility technique to be more effective than previous implementations using an Artificial Economy or Evolutionary Reinforcement Learning with a dynamic agent population. The performance of the system is restricted by the capacity of the rule language, and reaching equal performance with the artificial opponent may be optimal with the given language. The Concept Accessibility technique provides a means to balance generalisation and specialisation and provide good coverage using streamlined principles, and may be valuable to other domains.

ACKNOWLEDGMENTS

This work was carried out under Australian Research Council Discovery Grant DP0560207 Bossomaier et al., under which Anthony Knittel has been funded.

REFERENCES

[1] Y. Sato, E. Akiyama, and J. Crutchfield, "Stability and diversity in collective adaptation," *http://www.arxiv.org/pdf/cond-mat/9907176*, 1999.

[2] E. Berlekamp, J. H. Conway, and R. K. Guy, *Winning Ways for Your Mathematical Plays*. London: Academic Press, 1982.

[3] T. Bossomaier and A. Knittel, "An evolutionary agent approach to dots and boxes," in *Proceedings of IASTED International Conference on Software Engineering and Applications*. IASTED, 2006.

[4] A. Knittel, T. Bossomaier, M. Harre, and A. Snyder, "Stochastic reinforcement in evolutionary multi-agent game playing of dots and boxes," in *International Conference on Computational Intelligence for Modelling, Control and Automation*. CIMCA, 2006.

[5] E. R. Berlekamp, *The Dots-and-Boxes Game: Sophisticated Child's Play*. AK Peters, Ltd., 2000.

[6] A.-L. Barabási, *Linked*. Perseus, Massachusetts, 2002.

[7] D. Watts, *Small Worlds*. Princeton University Press, 1999.

[8] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon, "Superfamilies of Evolved and Designed Networks," *Science*, vol. 303, no. 5663, pp. 1538–1542, 2004. [Online]. Available: http://www.sciencemag.org/cgi/content/abstract/303/5663/1538

[9] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, and U. Alon, "Response to Comment on "Network Motifs: Simple Building Blocks of Complex Networks" and "Superfamilies of Evolved and Designed Networks"," *Science*, vol. 305, no. 5687, pp. 1107d–, 2004. [Online]. Available: http://www.sciencemag.org

[10] L. BULL, "Learning classifier systems: A brief introduction," in *Applications of Learning Classifier Systems*, L. Bull, Ed. Berlin: Springer, 2004, pp. 3 – 14.

[11] L. B. Booker, D. E. Goldberg, and J. H. Holland, "Classifier systems and genetic algorithms," *Artif. Intell.*, vol. 40, no. 1-3, pp. 235–282, 1989.

[12] S. W. Wilson, "Generalization in the XCS classifier system," in *Genetic Programming 1998: Proceedings of the Third Annual Conference*, J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba, and R. Riolo, Eds. University of Wisconsin, Madison, Wisconsin, USA: Morgan Kaufmann, 22-25 1998, pp. 665–674. [Online]. Available: citeseer.ist.psu.edu/wilson98generalization.html

[13] D. Cliff and S. Ross, "Adding temporary memory to zcs," *Adapt. Behav.*, vol. 3, no. 2, pp. 101–150, 1994.

[14] M. V. Butz, T. Kovacs, P. L. Lanzi, and S. W. Wilson, "How XCS evolves accurate classifiers," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, Eds. San Francisco, California, USA: Morgan Kaufmann, 7-11 2001, pp. 927–934. [Online]. Available: citeseer.ist.psu.edu/article/butz01how.html

[15] M. Butz and S. W. Wilson, "An algorithmic description of xcs," in *IWLCS '00: Revised Papers from the Third International Workshop on Advances in Learning Classifier Systems*. London, UK: Springer-Verlag, 2001, pp. 253–272.

[16] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998. [Online]. Available: citeseer.ist.psu.edu/sutton98reinforcement.html

[17] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, 1988.

[18] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," in *AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1998, pp. 746–752.

[19] J. Hu and M. P. Wellman, "Multiagent reinforcement learning: Theoretical framework and an algorithm," in *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 242–250.

[20] A. Politi, "Nonie's dots and boxes, http://dsl.ee.unsw.edu.au/dsl-cdrom/unsw/projects/dots," (Resource of artificial player used as training opponent).