Chimps: An Evolutionary Reinforcement Learning Approach for Soccer Agents*

Carlos Castillo
Grupo de Inteligencia Artificial
Universidad Simón Bolívar
Caracas 1080-A, Venezuela
carlos@gia.usb.ve

Miguel Lurgi
Grupo de Inteligencia Artificial
Universidad Simón Bolívar
Caracas 1080-A, Venezuela
miguel@gia.usb.ve

Ivette Martínez
Departamento de Computación
Universidad Simón Bolívar
Caracas 1080-A, Venezuela
martinez@ldc.usb.ve

Abstract – *In non-deterministic and dynamic environments, such as the RoboCup simulation league, it is necesary to simplify the search space to manage action selection in real time. In this work, we present* Chimps, *a team for RoboCup simulation league that uses an accuracy-based evolutionary reinforcement learning mechanism, called* XCS *to achieve this simplification.* XCS *is a Genetic Classifier System, with generalization capacities; we use them for the evolution of individual behavior's rules. We modified an existing team, 11Monkeys, that used static rules for individual action selection, adding an XCS to learn in real time over the outcome of individual actions. We found that our extension enhanced the team's performance.*

Keywords: Robocup, Evolutionary Reinforcement Learning, Genetic Classifier Systems, XCS

1 Introduction

As proposed by Kitano *et al* [5], RoboCup has become a standard problem for the artificial intelligence, intelligent robotics, and multi-agent systems comunities. In particular, the simulation league provides a more realistic soccer environment that allows to deal with teamwork and offensive and defensive strategies in an effective way.

The RoboCup simulation soccer league domain could be caracterized as a fully distributed, multiagent, partially observable domain. The agents also have noisy sensors and actuators (i.e, perceptions and actions are non-deterministic). Additionally, perceptions and actions cycles are asyncronous, comunication is limited, and agent's action selection must be done in real-time. These characteristics made RoboCup simulation soccer league a challenging domain. *Perhaps the most pressing challenge in RoboCup simulated soccer is the large state space, which requires some kind of general function approximation*[11].

Usually the large state-space problem is managed through generalization techniques such as neural networks and other function approximators; *which allow compact storage of learned information and transfer of knowledge between "similar" states and actions* [3].

To manage the real-time action selection in this large state space, through generalization, we propose the use of an accuracy-based evolutionary reinforcement learning mechanism, called XCS [14].

Evolutionary reinforcement learning (ERL) is a new approach to reinforcement learning that takes advantage of Darwin's theory of evolution. Evolutionary algorithms allow a fast search over policy spaces. *Methods from genetic algorithms, evolutionary programming, genetic programming, and evolutionary strategies could all be used in this framework to form effective decision making agents* [7].

ERL methods represent policies as chromosomes or as distributed representations based on rules. XCS uses the second one. XCS main distinguishing features are the basing of classifier fitness on the accuracy of classifier reward prediction instead of prediction itself and the use of a niche genetic algorithm. These changes gave it generalization capacities and allow to make complete mappings of <*condition, action*> pairs over predictions [14].

For the learning soccer agents, we used evolutionary technics, replacing the static rules of the team 11Monkeys [4]. This selection was made because: the 11Mon-

*0-7803-7952-7/03/$17.00 © 2003 IEEE.

keys team was successful, it had an appropriate separation between the action selection mechanism and the rest of the functionality, and its code was freely available.

In the next section we present an overview of the XCS mechanisms. In Section 3 we describe 11Monkeys, the team that we took as base; in Section 4 we present detailed Chimps design and implementation, Section 5 contains experimental results. Conclusions and future work are in Section 6.

2 XCS: Accuracy based Learning Classifier Systems

Initially, a player's action selection decision, given an environment state, will be made by means of a *XCS*, genetic classifier systems based on accuracy [14].

The XCS classifier systems, as well as Holland's Learning Classifier Systems (LCS)[2], are domain independent adaptive learning systems. Their main distinguishing features are the basing of classifier fitness on the accuracy of classifier reward prediction instead of on the prediction itself, and the use of a niche genetic algorithm, i.e., a GA that operates on a subset of the classifier population.

The structure of *XCS* rules' conditions are the translation of the conditional part of the logical rules. Rules' actions are binary strings that represent motion actions.

A classifier is a compact representation of a complex set of environment states. Rules have the form $< condition > \rightarrow < action >$. Conditions are strings of length l in the alphabet $\{0, 1, *\}$, $*$ represents a don't care condition. A classifier's condition satisfies a message if its condition matches the input message. A condition c matches message m if and only if:

$$\forall i, (1 \leq i \leq l) \rightarrow \Pi_i(c) = \Pi_i(m) \vee \Pi_i(c) = '*'$$

. Actions are fixed length strings in the alphabet $\{0, 1\}$. XCS are composed of three subsystems:

- A performance system,
- an evaluation system, and
- a rule discovery system

The performance system takes an input from the environment, selects an action and transforms it into an output message. Structurally, a performance system consists of a finite population $[P]$ of classifiers, also called ruleset. The basic execution cycle of the performance system is as follows (a graphic representation of it is showed in figure 1):

1. Obtain a single input string from the environment.
2. Form the match set $[M]$ of classifiers whose condition matches the input string.
3. Select an action based on the prediction of the classifiers in $[M]$.
4. Form the action set $[A]$ of classifiers in $[M]$ which advocated the action selected in 3.
5. Translate the selected action into the output of the system.

The learning system takes feedback signals from the environment and updates the values of the four parameters that replaces the traditional fitness of LCS: Prediction, prediction error, accuracy, and fitness. This change allows a more complete *State* × *Actions* → *Prediction* mapping than traditional LCS.

The rule discovery system uses a genetic algorithm in order to create new rules. XCS's rule discovery system has two operations: niche genetic algorithm and covering. The niche genetic algorithm acts over the Action Set $[A]$, choosing random parents in proportion to the rules' fitness. Offprints are copies of the parents, modified by crossover and mutation. Covering is triggered when the matches set is empty or its media prediction is a small fraction of the population [P] average prediction. Covering creates a new classifier whose condition matches the current input message; its action is generated randomly.

2.1 XCS as a Technique of Reinforcent Learning (RL)

In the Reinforcement Learning (RL) research, two different approaches have been stated. These two approaches are known as: searching in value function space, and searching in policy space. In the first approach, RL algorithms try to find the optimum function value for the problem. Then, to find the optimal policy given the optimal function values, is immediate. The second approach is to search an optimal policy directly over the space of the policies. For this purpose, evolutionary algorithms are frequently used [7].

This RL approach based on evolutionary algorithms is called Evolutionary Reinforcement Learning (ERL). The ERL algorithms vary in terms of:

- the policies representation method
- the method for the fitness evaluation of the individual policies

The two methods of policies's representation are: a chromosome representation and distributed rules-based representations. Holland's Learning Classifier Systems (LCSs) [2] as well as XCS are examples of a rules-based ERL.

The advantage of XCS from the ERL point of view is its generalization capacity. The don't care conditions allow a more compact representation than a simple table, i.e., a single classifier could represent a group of

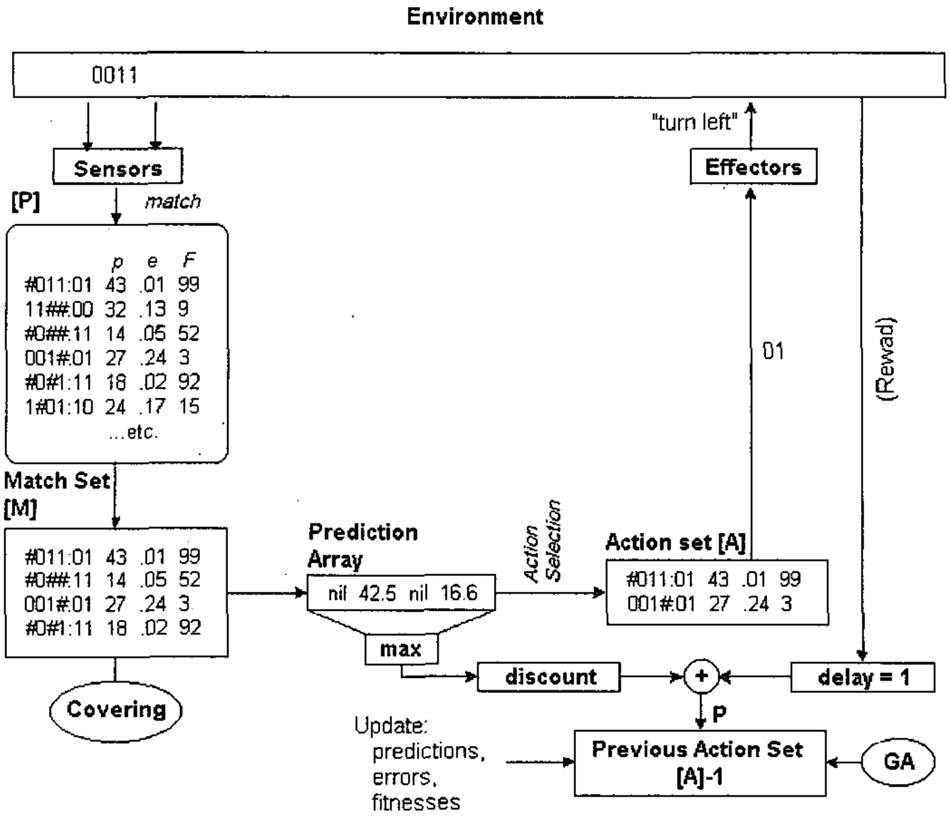

Figure 1: Schematic structure and functioning of XCS, taken from Wilson [14].

states. For this reason, the XCS is able to scale to more complex problems, in contrast with the RL traditional algorithms. [13].

3 11Monkeys

The original team that we use as base is the 11Monkeys [4], winner of the world cup RoboCup in 1998. The 11Monkeys team presents an elegant architecture and a well-done codification.

11Monkeys is based on an hierarchical architecture with selection of static rules, i.e. a logic-deductive architecture. The rules set is well defined and is composed of a set of pairs <*condition,action*>, where each condition represents a possibe environment's state and each action represents the response that the player (agent) will give to this particular state.

The architecture in this team is esentially logic-deductive, as mentioned, so one could think that in exactly the same situation or environment state at different times, the player will act the same way.

One of the goals of this work is to introduce the original team rules, in a XCS Genetic Classifiers System, by translating each rule of the form <*condition,action*> into a genetic classifier. These classifiers will evolve into new ones that will generate new or diverse answers for some environment's characteristics, i.e., new rules will be created based on some particular criteria, for example, the performance of the new rule.

4 Design and Implementation

The work is based esentially in the improvement of the 11Monkeys team with the goal of having learning ability. With this new feature the team will be able to have more victories than the original 11Monkeys team.

The work is based only in the offensive part of the team because the defensive part is more complicated and should be managed carefully, little changes produce a lot of different and ambiguous behaviors.

Chimps is a new team, able to learn behaviors based on the XCS genetic classifiers mechanisms.

4.1 Team's Architecture

Each agent has a hybrid architecture, with a reactive part and a deliberative one (also known as means-end). For each action, there is an agent that acts as a dynamic planner (in case of an offensive play, it is the player that has the ball, and in case of a defensive one, it is the nearest player to the ball) and some assistants that are continuously communicating with the planner player.

The layers of the architecture, inherited from 11Monkeys, are described here:

- *Strategy layer:* this layer includes all the players of the team. It also includes the static roles assignation that is performed at the beginning of each game, and how the team is positioned in the field.

- *Group Layer:* it includes the two or three players that in some given state are nearby the ball. There is a reactive and a deliverative part in this layer.

- *Individual layer:* includes the individual action selection; in this case there is either action planning or reactive support. In this layer the agents have two sets of rules: state dependent rules, based on current state (e.g., "has ball"); and position specific rules, that depends on the initial team formation (e.g., 4-4-2). A reevaluation of the plan is done, ideally, each time unit.

We choose to change only the 11Monkeys state dependent rules because XCS needs a continous interaction with the environment. Moreover, we just deal with the offensive set of rules since the defensive ones are more sensible to small changes, as we mentioned.

The environment where these agents act is nondeterministic, episodic, and very realistic. It is important to note here that each player has stamina, which determines the energy of the player. This variable determines indirectly the probability of the player to make a good or a bad play.

The rules choice is completely isolated from the communication or the planification parts of the architecture.

4.2 Rules and Learning

The initial set of rules was fixed based on extensions of the 11Monkeys original team rules and the intention was to translate these rules, into binary code with the goal of treating them as genetic classifiers that can evolve.

Each classifier condition is a string of 30 characters in the alphabet: $\Delta = \{0, 1, \#\}$, and the action is represented by binary string of 13 characters. The # symbol represents "don't care" conditions. This population can evolve by genetic rules inside the XCS.

An example of a rule translation into a binary chromosome is shown below:

- Original logic-deductive rule:
  Condition: (BallNotKickable)
  Action: (LookForBall)

- Translation of this rule into a binary chromosome:
  Condition:{0*****************************}
  Action:{1000000000000}

There are 30 high level inputs to the classifier (conditions), and 16 high level actions. The actions we propose are subsets of this universe of possible high level actions considered by 11Monkeys.

After several simplifications, the size of the space considered is of 2^{30} states. These simplifications include transforming the distance to the nearest opponent or the ball (a real number) to a boolean that represents if the opponent or the ball is within a key radius. We only consider 2 possible general positions (which are WingBack and DefensiveHalf) out of 12 general positions considered by 11Monkeys.

These rules as chromosomes are elitists, in the sense that they cannot be eliminated from the original population of classifiers. The strength element of the classifier is updated by the interactions with the environment. This strength is decremented in the classifiers which propose a bad action, while in the classifiers which propose good actions, the strength is incremented.

The solution schema formulated for the real-time learning for Chimps is described below:

- We selected the 16 attack rules from the original team which were translated into classifiers, and the XCS population was initialized with these 16 classifiers. These classifiers are an elite (they are never removed from the population).

- From the 16 initial rules, 9 more rules are generated by mutation and crossover to obtain a base set of 25 classifiers.

- In each iteration each agent perceives the environment's state through its sensors, translates it into a binary string, and gives it to the XCS. The XCS selects the rules that match this condition, picks the action set to be executed. Depending on the results of this action set the XCS updates strength and accuracy values in the classifiers that proposed the selected action set. Also, the classifier with the lowest strenght in the population is eliminated.

- When the population size is lower than 25 classifiers, a new one is generated from the others, maintaining a population of 25 classifiers; i.e, we trigger a covering operation.

5 Experimentation and Results

We ran the experiments using the Robocup Soccer Simulation version 9.3.7, running on Debian GNU/Linux on a Pentium III 1GHz machine with 256MB of RAM. All of our experiments were run in two halves of 3000 iterations (300 seconds) each.

We experimented with several rulesets, including a randomly generated one; and ended up using the one that comes with 11Monkeys initially and evolved from there. The level of play is strongly dependent on the initial rules.

We also tested playing series of games with other teams including the original 11Monkeys (which we extended).

In the following section, we describe the most interesting results from the experimentation with our team.

We played the Chimps against several other teams including 11Monkeys (source code version) [4], ATT-United [9] (2002 binary as it ran in Fukuoka), BS2K [8] (2002 binary), and current world champions TsinghuAeolus (2002 binary) [16, 15].

Our main result is a 70-12-18 record against the original 11Monkeys by changing only the action selection rules from static to dynamic and applying real-time learning over the outcome of the actions as described in the implementation. A detail of these results is presented in Table 1

We also won three series of 5 games (all of them 3-0) against ATTUnited. More details on one of these three series is presented in Table 2

Team	W	L	D	P	GF	GA
Chimps	70	12	18	228	141	34
11Monkeys	12	70	18	54	34	141

Table 1: Results of 100 games between 11Monkeys and Chimps. Our results show a dominance of real-time learning in Chimps to static rules in 11Monkeys

Team	W	L	D	P	GF	GA
Chimps	5	0	0	15	9	0
ATTUnited	0	5	0	0	0	9

Table 2: Results of a 5-game series between ATTUnited and Chimps. Two more 5-game series were played obtaining almost identical results.

We did not do well against more modern teams such as BS2K and TsinghuAeolus, both of which swept us series of five games. TsinghuAeolus uses adversarial planning and Q-learning which we believe is equivalent to what we're proposing.

Our implementation and preliminary results constitute a "proof of concept." Our approach can be readily ported to other logic-deductive architectures and provides natural integration with existing action selection schemes.

Modern machine learning techniques have been applied to specific portions of the soccer domain, see for example [10]. Evolutionary methods have also been applied in [1] and in [6]. However, ERL methods have not been studied in this domain, this work constitutes a step in this direction.

6 Conclusions and Future Work

We have presented a successful technique for real-time learning for action selection based on XCS. Our experiments show that this change increased level of play of an existing team considerably.

We played against several teams, including 11Monkeys. Against it we won 70 of 100 games, usually by more than 2 goals. These results show that our approach is an improvement over the original architecture. Our experience illustrates how real-time evolutionary learning improves static rules.

XCS provides an elegant and extensible reinforcement learning mechanism that can be used in the RoboCup domain. Additionaly XCS naturally extends logic deductive architectures with a powerful reinforcement learning mechanism. XCS constructs generalizations over the state space that allow to handle a large search space in a very effective way.

The method described here can also be applied (with some extensions) to defensive action selection. Other ERL methods can be tested both for offensive and defensive action selection. Our next step is to try with a XCS variation called AXCS[12] (average reward XCS) in order to evolve defensive rules.

Currently only some teams apply evolutionary reinforcement learning to this domain, but we expect this to change in the near future. Different ERL methods need to be thoroughly explored.

References

[1] D. Andre and A. Teller. Evolving Team Darwin United. In M. Asada and H. Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*, volume 1604 of *LNCS*, pages 346–351, Paris, France. Springer Verlag, 1999.

[2] J. H. Holland. *Escaping Brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems.* Mitchell, Michalski and Carbonell (Editors), 1986.

[3] L. P. Kaelbling, M. L. Littman, and A. P. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[4] Y. Kinoshita. Team 11monkeys description, 1999.

[5] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. RoboCup: The robot world cup initiative. In W. Lewis Johnson and Barbara Hayes-Roth, editors, *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pages 340–347, New York. ACM Press, 1997.

[6] S. Luke, C. Hohn, J. Farris, G. Jackson, and J. Hendler. Co-evolving soccer softbot team coordination with genetic programming. In *Proceedings*

of the First International Workshop on RoboCup, at the International Joint Conference on Artificial Intelligence, Nagoya, Japan, 1997.

[7] D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette. Evolutionary Algorithms for Reinforcement Learning. *Journal of Artificial Intelligence Research*, 11:199–229, 1999.

[8] M. Riedmiller. Karlruhe brainstormers - design principles, 1999.

[9] P. Stone, P. Riley, and M. Veloso. The CMUnited-99 champion simulator team. In M. Veloso, E. Pagello, and H. Kitano, editors, *RoboCup-99: Robot Soccer World Cup III*, Berlin. Springer Verlag, 2000.

[10] P. Stone and R. Sutton. Keepaway soccer: a machine learning testbed, 2002.

[11] P. Stone and R. S. Sutton. Scaling reinforcement learning toward RoboCup soccer. In *Proc. 18th International Conf. on Machine Learning*, pages 537–544. Morgan Kaufmann, San Francisco, CA, 2001.

[12] K. Tharakunnel and D. E. Goldberg. XCS with average reward criterion in multi-step environment, 2002.

[13] S. W. Wilson. Generalization in the XCS classifier system. In 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, editors, *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 665–674, University of Wisconsin, Madison, Wisconsin, USA. Morgan Kaufmann, 1998.

[14] S. W. Wilson. Classifier fitness based on accuracy. *Evolutionary Computation*, 3(2):149–175, 1995.

[15] J. Yao, J. Chen, and Z. Sun. An application in robocup combining q-learning with adversarial planning, 2002.

[16] J. Yao, C. Yiang, C. Yunpeng, and L. Shi. Architecture of tsinghuaeolus, 2002.