IN-TEH
intehweb.com

Ant Intelligence in Robotic Soccer

R. Geetha Ramani*; P. Viswanath† and B. Arjun†
*Assistant Professor, Dept. Of CSE & IT, Pondicherry Engineering College
†Department of Computer Science, Pondicherry University
Corresponding author E-mail: rgeetha@yahoo.com, visucse@yahoo.com.

Abstract: *Robotic Soccer is a multi-agent test bed, which requires the designer to address most of the issues of multi-agent research. Social insect behaviors observed in nature when adopted to solve problems they are giving promissing results. The domains like computers, electronics, electrical, mechanical etc., are inspired in adopting these behaviors. This paper addresses the ant intelligence in robotic soccer to evolve the best team of players. The simulation team evolved (PUTeam) was tested with teams of soccerbots in teambots (a simulation tool for Robotic Soccer) and the experimental results clearly shows the performance of the evolved team against the opponent teams are more effective.*

Keywords: *Robotic Soccer, Social Insect Behaviors, Ant intelligence, Learning methods, Stigmergy, Self-organization.*

1. Introduction

Robotic Soccer is an interesting and emerging domain, which represents the problem of mobile autonomous robots playing the popular game of soccer. It is perceived as a multi-agent learning test bed, which is helpful in demonstrating the strengths of various multi-agent learning strategies. So soccer is a rich domain for the study of multi-agent learning issues. Teams of players must work together in order to put the ball in the opposing goal while at the same time defending their own. Learning is essential in this task since the dynamics of the system can change as the opponents' behavior change. The players must be able to adapt to new situations and behaviors of different opponents. Also they must learn to work together. By making the robots play this game, different developments in intelligent agents can be put into practice. These include developments in autonomous, cooperative, competitive, reasoning, learning, and revision systems. So Robotic soccer has become the new benchmark problem and Holy Grail in the field of Artificial Intelligence.

To foster the research in the field of multi-agent systems, an international robotic soccer competition called RoboCup was started in 1997. The Robot World Cup Initiative now known as RoboCup is proposed as a standard problem for research in the areas of AI and robotics, requiring the use of several technologies and research in a wide range of areas in AI and robotics. It can be seen as an international research and education initiative, which attempts to foster artificial intelligence and intelligent robotics research by providing a standard problem where wide range of technologies can be integrated and examined. As soccer game is chosen as a primary domain, the competition can also help to foster public awareness of the current level of development of intelligent machines. The various leagues of robocup are discussed in the following subsection.

1.1 RoboCup Soccer Leagues

The main focus of the RoboCup organization is competitive soccer. The competition has several leagues namely physical league, simulation league and rescue challenge league. In the case of physical league there are different sizes of physical robots and the issues of hardware and software arise as sensors and actuators must perform correctly and well to interact effectively with the software driving them, and that software must be designed to solve many problems such as cooperation in a dynamic environment. In simulation league, which uses the SoccerServer and client code, allowing software agents to compete. The league focuses more on the design of intelligent strategies for agents and teams. Rescue challenge leagues are conducted to evaluate the skills of the robots for their rescue actions in the hazardous environment and during disaster scenarios. The detailed information about the leagues are given below.

Small-size robot league: Small robots of no more than 18 cm in diameter play soccer with an orange golf ball in teams of up to 5 robots on a field with the size not bigger than a ping-pong table. Matches are having 10-minute halves. This league was introduced in 1997 RoboCup itself.

Middle-size robot league: Middle-sized robots of no more than 50 cm diameter play soccer in teams of up to 4 robots with an orange soccer ball on a field the size of 12x8 metres. Matches are divided in 10-minute halves. This league also was introduced in 1997 RoboCup.

Four-legged robot league: This was introduced in RoboCup 2000. Teams of 4 four-legged entertainment robots (eg Sony's aibo) play soccer on a 3 x 5 metre field. Here also matches have 10-minute halves.

Humanoid league: This league was introduced in 2002. In this league, biped autonomous humanoid robots play in 'penalty kick', and '1 vs. 1', ' 2 vs. 2' matches.

Simulation league: Independently moving software players (agents) play soccer on a virtual field inside a computer. Matches have 5-minute halves. This is one of the oldest fleet in RoboCupSoccer, which was introduced in the preRoboCup held in 1996.

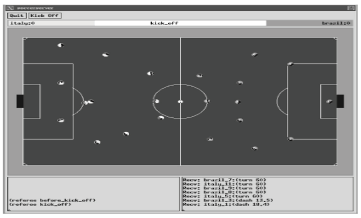
Apart from the soccer leagues, the RoboCup has another league called the RoboCup Rescue Challenge. This presents the participating agents with a realistic landscape with buildings, infrastructure, and other entities common in modern cities. Within this landscape, thousands of people are dispersed at various locations in a realistic fashion, when some sort of accident or catastrophic event occurs. This disaster event, e.g., a major earthquake, happens at time zero, when the simulation starts. The event quickly leads to consequences, according to the selected scenario, such as fires, explosions, gas clouds, collapsing buildings, etc. As there are different disaster events in different parts of the world, RoboCup Rescue can include several scenarios, which enables the researchers to investigate techniques that have applications pertaining to their own country.

A rescue team consists of different kinds of personnel or robotic rescuers, along with useful equipment. The rescue team must be controlled by agent programming techniques, i.e., at least autonomous and adaptive processes, and be efficiently used to save as many human lives as possible. Minimizing destruction of real estate, infrastructure, and other assets should also be rewarded. In order to save as many human lives as possible or minimizing destruction, the agents must be able to prioritize and make quick decisions.

1.2 RoboCup Simulation League

One among the main events of the RoboCup Simulation League is the simulated soccer matches. Both 2D and 3D simulation leagues are conducted. In the 2D Soccer Competition of the RoboCup Simulation League, teams of 11 autonomous software agents per side play each other using the RoboCup soccer server simulator. There are no actual robots in this league but spectators can watch the action on a large screen, which looks like a giant computer game. Each simulated robot player may have its own play strategy and characteristic and every simulated team actually consists of a collection of programmes.

Many computers are networked together in order for this competition to take place. The games last for about 10 minutes, with each half being 5 minutes duration. The Soccer Server allows autonomous software agents written in an arbitrary programming language to play soccer in a client/server-based style. The server simulates the playing field, communication, the environment and its dynamics, while the clients or the players are permitted to send their intended actions (e.g. a parameterized kick or dash command) once per simulation cycle to the server via UDP. Then, the server takes all agents' actions into account, computes the subsequent world state and provides all agents with (partial) information about their environment via appropriate messages over UDP. The course of action during a match can be visualized using an additional program, the Soccer Monitor. A screenshot of the soccer server is shown in Fig.1.



```
(referee before_kick_off)
(referee kick_off)
```

```
Recv: brazil_7:(turn 60)
Recv: italy_11:(turn 60)
Recv: brazil_9:(turn 60)
Recv: brazil_8:(turn 60)
Recv: italy_5:(turn 60)
Recv: brazil_3:(dash 13.5)
Recv: italy_1:(dash 18.4)
```

Fig. 1. Screen of the soccer server

Several research issues are involved in the development of real robots and software agents for RoboCup. One of the major reasons why RoboCup attracts so many researchers is that it requires the integration of a broad range of technologies into a team of complete agents, as opposed to a task-specific functional module. The following is a partial list of research areas, which RoboCup covers:

- Agent architecture in general;
- Combining reactive approaches and modeling/planning approaches;
- Real-time recognition, planning, and reasoning;
- Reasoning and action in a dynamic environment;
- Sensor fusion;
- Multi-agent systems in general;
- Behavior learning for complex tasks;
- Strategy acquisition;
- Cognitive modeling in general.

Currently, each league has its own architectural constraints, and therefore research issues are slightly different from each other. For the synthetic agent in the simulation league, the following issues are considered:

- Teamwork among agents, from low-level skills like passing the ball to a teammate, to higher-level skills involving execution of team strategies.
- Agent modeling, from primitive skills like recognizing agents' intentions to pass the ball, to complex plan recognition of high-level team strategies.
- Multi-agent learning, for on-line and off-line learning of simple soccer skills for passing and intercepting, as well as more complex strategy learning.

For the robotic agents in the real robot leagues, for both the small-and middle-size ones, the following issues are considered:

- Efficient real-time global or distributed perception possibly from different sensing sources.
- Individual mechanical skills of the physical robots, in particular target aim and ball control.
- Strategic navigation and action to allow for robotic teamwork, by passing, receiving and intercepting the ball, and shooting at the goal.

More strategic issues are dealt with in the simulation league and in the small-size real robot league while acquiring more primitive behaviors of each player is the main concern of the middle-size real robot league. Since the simulation league is well suited for testing the various multi-agent strategies without bothering about the hardware and the electrical and mechanical aspects, a number of multi-agent learning methods are applied to it. The next subsection gives a brief information of the multi-agent learning methods that are applied for developing the player strategies for robotic soccer simulation.

1.3 Learning methods

A team's success in robotic soccer will depend on how efficient it can react to the uncertain environment, which in turn depends on the learning ability of the agent. Thus learning methods play an important role in robotic soccer. In the pre RoboCup which was held in 1996, the participated teams were having fixed hand coded strategies. But in the following years the researchers found more and more efficient strategies for their teams by incorporating learning abilities to the soccer-playing agents. Some of the important methods among them are discussed in section 2. As part our work includes incorporation of social insect behaviors especially ant behavior, a brief introduction is given in next subsection.

1.4 Social insect Behaviors

Many people discovered the variety of the interesting insect or animal behaviors in the nature. A flock of birds sweeps across the sky. A group of ants forages for food. A school of fish swims, turns, flees together[1]. Hive's of bee communicates using dance language. In fact the honeybee dance language has been called one of the seven wonders of animal behaviors and is considered among the greatest discoveries of behavioral science[2]. Termites are small in size, completely blind and wingless - yet they have been known to build mounds 30 meters in diameter and several meters high[3]. We call this kind of aggregate motion "swarm behavior[4]". Recently biologists and computer scientists in the field of "artificial life" have studied how to model biological swarms to understand how such "social animals" interact, achieve goals, and evolve. Moreover, engineers are increasingly interested in this kind of swarm behavior since the resulting "swarm intelligence" can be applied in optimization, robotics, traffic patterns in transportation systems, and military applications etc. As the days passes many domains are influenced by these social insect behaviors in problem solving. The various domain influenced are computer science, electronics, electrical, aeronautical, mechanical, bio-informatics, defense, music.

Next section deals with learning methods in detail and the subsequent section is proposing idea of mapping social insect, especailly ant behaviors in robotic soccer and finally simulation results followed by conclusion and future work.

2. Learning Methods

Learning methods plays an important role in robotic soccer. Some of the important methods are discussed below

2.1 Reinforcement Learning

An agent situated in some environment interacts to the environment using its sensors and effectors. The actions of the agent bring about changes in the environment and the environment provides feedback that guides the learning algorithm as illustrated in Fig 2. These feedbacks can act as positive or negative reinforcements to the agent's action. The reinforcement learning algorithms were proven to be applicable to a variety of complex domains. It has been used widely in the robotic soccer domain also. Here the algorithm learns a policy of how to act given by observation of the world.

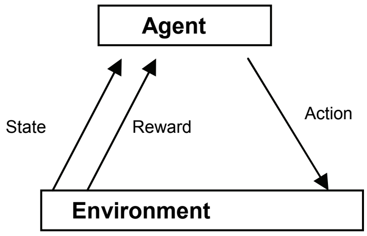


Fig. 2. The interaction of agent and environment in reinforcement learning

Reinforcement learning was applied to robotic soccer in various ways. Some of the approaches and variations of Reinforcement learning to Robotic Soccer is discussed below.

2.1.1 Observational Reinforcement Learning

Observational Reinforcement Learning was used for learning to update players' positions on the field based on where the ball has previously been located. This was used in the Andhill team, which was the runner up in the first RoboCup simulation league. With Observational Reinforcement Learning method, the learning agent evaluates inexperienced policies, which is evaluated as good from its observation, and reinforces it. In the RoboCup positioning problem, an agent can evaluate some positions as good just only from its observation. One example evaluation may be like: A place where the ball comes frequently will suit for positioning. In comparison with ordinary reinforcement learning, observational reinforcement learning was shown to be helpful for avoiding local optima.

A similar mechanism for reinforcement learning from teammates that operates in tandem with a method for modeling the ability of other agents was explored. This allows a learning agent to take advantage of teammates' reinforcements, while simultaneously attempting to differentiate the skill levels of the reinforcers. Each player maintains an ongoing reputation of the skills of each teammate in the form of a cumulative average of a reputation score based on episodes of good and bad play observed.

Modeling other agents like this helps in making wise choices when interacting with others during play. Through the use of such a modeling scheme, one can appropriately select good players to interact with, and also will be able to use this mechanism to differentiate reinforcement provided by good and poor players during play. From this, different methods of combining or weighting reinforcement may be explored in order to improve learning in such settings. The experiments showed that ability to identify poorly-skilled agents and filter their reinforcement was not of use when all of ones teammates were good. If surrounded by good agents, the learning agent is able to reliably learn to select actions like a good player in 8 out of the 9 possible different situations.

2.1.2 Clay

Clay was an evolutionary architecture for autonomous robots, which integrates motor schema-based control and reinforcement learning. Motor Schemas are primitive behaviors for accomplishing a task. For instance, important motor schemas for a navigational task may be 'avoid-obstacles' and move-to-goal'. If motor schema based control and reinforcement learning are integrated, robots using this system can benefit from the real-time performance of motor schemas in continuous and dynamic environments while taking advantage of adaptive reinforcement learning. Clay co-ordinates assemblages or groups of motor schemas using embedded reinforcement learning modules. Learning occurs as the robot chooses assemblages and then samples a reinforcement signal over time. Clay was used by Georgia Tech in the configuration of a soccer team for the RoboCup 97 simulator competition.

2.1.3 Team-Partitioned Opaque-Transition Reinforcement Learning (TPOT-RL)

A concept of using action-dependent features was introduced to generalize the state space, namely *Team-Partitioned Opaque-Transition Reinforcement Learning (TPOT-RL)*. The Domains in which there is a lack of control for single agents to fully achieve goals are called as team-partitioned. In opaque transition domains, the agents do not know in what state the world will be in, after an action is selected, since another possibly hidden agent will continue the path to the goal. Adversarial agents can also intercept and thwart the attempted goal achievement. Also, real world domains have far too many states to handle individually.

TPOT-RL constructs a smaller feature space V using action dependent feature functions. The expected reward Q (v, a) is then computed based on the state's corresponding entry in the feature space. This action-dependent feature space is used to allow a team of agents to learn to co-operate towards the achievement of a specific goal. TPOT-RL was used to train the passing and shooting patterns of a team of agents in fixed positions with no dribbling capabilities for the CMUnited teams.

2.1.4 Vision-based Reinforcement Learning

A Vision based reinforcement learning that acquires cooperative behaviors in a dynamic environment was applied on real soccer playing robots. In this method, each agent works with other team members to achieve a common goal against opponents. The relationships between a learner's behaviors and those of other agents in the environment are estimated through interactions (observations and actions). Next, reinforcement learning based on the estimated state vectors is performed to obtain the optimal behavior policy. While applying the method to Robotic Soccer, a robot firstly learnt to shoot the ball into a goal given the state space in terms of the size and the positions of both the ball and the goal in the image, then learnt the same task but with the presence of a goalkeeper. The proposed method, which was applied to a soccer-playing situation successfully models a rolling ball and other moving agents and acquires the learner's behaviors. It was also described how Reinforcement Learning can be used to obtain optimal behaviors, based on estimated state vectors in order to obtain the optimal behavior. The method can cope with a rolling ball.

2.1.5 Scoring Policy using Reinforcement Learning

Scoring behavior can be thought of as the most effective one in the result of the game. So, it is important to have a clear policy for scoring goal. UvATrilearn simulation team, which was the champion of the world in RoboCup 2003, had one of the best scoring techniques. In this technique, the best point of the goal and the probability of scoring at this point are calculated. If the probability of goal is greater than a threshold, agent shoots toward the goal point otherwise, the agent executes another action. Later Reinforcement learning was applied considering two additional parameters (the body and the neck angle of the goalkeeper) beside the probability to the policy of the UvA team. The results of applying Reinforcement learning shows that the scoring behavior improved compared to the previous approach.

The robotic soccer problem has been modeled as a multi-agent markov decision process. The moves to learn several basic behaviors were learned using reinforcement learning with neural nets as function approximators. The algorithm learns along the trajectories, which lead to a goal or to the loss of the ball. In the first case adds a positive reinforcement and the second a negative cost. The low-level skills such as kicking, ball interception, and dribbling, as well as the cooperative behavior of team members were learned using this learning method. Very promising results in learning of coordinated offensive behavior are reported. The team was the runner up of the simulation league of RoboCup 2000.

2.1.6 Q-learning

Q-learning is a form of Reinforcement Learning which is very suited for games against an unknown opponent. This does not need a model of its environment and can be used on-line. In Q-learning, the value of taking each possible action in each situation is represented as a utility function, Q(s, a) where *s* is the state or situation and *a* is a possible action. If the function is properly computed, an agent can act optimally simply by looking up the best valued action for any situation. The problem is to find the Q(s, a) s that provides an optimal policy. Then the agent can use this to select an action for each state.

A learning approach which is feasible for an agent running to the ball and dribbling the ball had been devised using the concept of Q-learning. Basic skills in the simulated robotic soccer, like learning to walk to the ball, or learning to shoot at goal are learned using the approach. Bayesian networks were used for modeling other agents in the environment. Decision trees and Bayesian networks helped to cut down the large state space due to incomplete information.

2.1.7 Modular Q-Learning Architecture

Modular Q-learning, which is one of the reinforcement learning schemes, is employed in assigning a proper action to an agent in the multi-agent system. A modular Q-learning architecture was applied to the robotic soccer domain to solve the action selection problem among robots. This specifically selects the robot that needs the least time to kick the ball and assign this task to it.

The architecture of modular Q-learning consists of learning modules and a mediator module. The learning modules amount to the number of agents involved in the task. Each agent in the learning module carries out Q-learning in the environment. The mediator module selects the most suitable action based on the Q-value received from each learning modules. The concept of the coupled agent was used to resolve a conflict in action selection among robots. The effectiveness of the scheme was demonstrated through real robot soccer experiments.

2.1.8 Q-Learning based behavior assignment

A market-driven multi-agent collaboration strategy with Q-Learning based behavior assignment mechanism was applied to the robot soccer domain in order to solve issues related to multi-agent coordination. Each team member calculates costs for its assigned tasks, including the cost of moving, aligning itself suitably for the task, and cost of object avoidance, then looks for another team member who can do this task for less cost by opening an auction on that task. With this, a Q learner added to replace the role assignment to make the approach more adaptive.

The learning implementation queries the action set and assigns the best action to the agent, thus enables multiple agents acting in the same role at the same time. This task assignment process is illustrated in Fig.3. It was shown experimentally that team with learned strategy performs better than the purely market-driven team since it has learned to assign behaviors adaptively. The main disadvantage of the approach for robotic soccer domain was the time requirement for the auctioning and utility calculation processes.

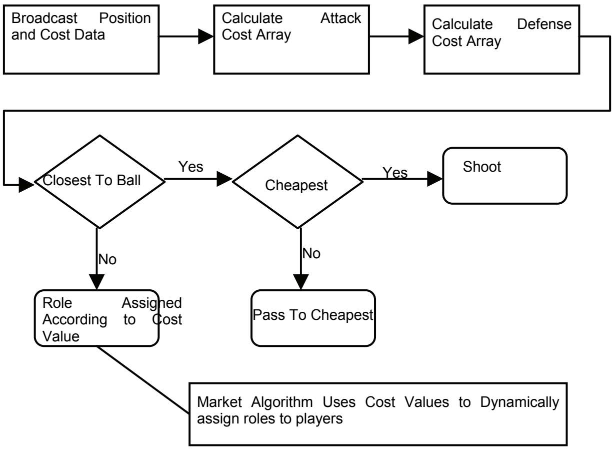

Broadcast Position and Cost Data
Calculate Attack Cost Array
Calculate Defense Cost Array
Closest To Ball
Yes
Cheapest
Yes
Shoot
No
No
Role Assigned According to Cost Value
Pass To Cheapest
Market Algorithm Uses Cost Values to Dynamically assign roles to players

Fig. 3. Flow chart for task assignment using Q learning

This method was improved by using reinforcement learning for role assignment by utilizing a reduced state vector. The state vector includes information about the agents and the ball. The improved state vector has information about Ball position, Ball possession, own role, Teammate positions and Opponent positions. The reinforcement measures are the goals scored by either our team or the opponent team. The team was tested against three teams using the teambots simulator, with three opponent teams SchemaNewHetero, AIKHomoG, RIYTeam and MarketTeam. The proposed team was able to defeat other opponents. The results showed that reinforcement learning is a good solution for role assignment problem in the robot soccer domain.

In addition to these works, the concept of reinforcement learning has been studied extensively and used to develop strategies for teams of soccer agents. It has been applied to a sub task of robotic soccer namely keepaway. It can be seen that the main advantage of reinforcement learning is that it provides a way of programming agents by reward and punishment without needing to specify how the task is to be achieved. The agent should choose actions that maximize the long-run sum of rewards.

2.2 Inductive Learning

Inductive learning is a machine-learning framework, which is based on generalization of examples. The concept of Inductive Logic Programming (ILP) has been used for soccer agents' inductive learning.

A framework for inductive learning soccer agents (ILSAs) had been proposed in which the agents acquire knowledge from their own past behavior and behave based on the acquired knowledge. The inductive learning soccer agents decides each action taken in the game to be good or bad according to the examples which are classified as positive or negative. Also the agents themselves classify their past states during the learning process. In the framework the agent is given an action strategy, ie, a state checker and an action-command translator. An inductive learning soccer agent acquires a rule from examples whose positive examples consist of states in which the agent failed an action, and uses the acquired rule as the state checker to avoid taking actions in states similar to the positive examples.

Based on this work, another agent architecture that adapts its own behavior by avoiding actions, which are predicted to be failure, is proposed in. The inductive learning agent used first-order formalism and inductive logic programming (ILP) to acquire rules to predict failures. First, the ILA collects examples of actions and classifies them. Then the prediction rules are formed using ILP and uses them for their behavior. This was implemented in soccer using parts of the RoboCup-1999 competition champion CMUnited-99 and an ILP system Progol. It was shown that agents could acquire prediction rules and could adapt their behavior using the rules. It was found that the agents used actions of CMUnited-99 more effectively after they acquired prediction rules.

Another research has been reported, which uses ILP systems for verifying and validating multi-agents for RoboCup. This concentrates on verification and validation of knowledge based system, not but prediction or discovery of new knowledge. Consequently, agents cannot adapt their own behavior using rules or knowledge acquired by ILP.

2.3 Memory Based Supervised Learning

A memory-based supervised learning strategy was introduced, which enables an agent to choose to pass or shoot in the presence of a defender. Learning how to adjust to an opponent's position can be critical to the success of having intelligent agents collaborating towards the achievement of specific tasks in unfriendly environments. Based on the position of an opponent indicated by a continuous-valued state attribute the agent learns to choose an action. A memory-based supervised learning strategy, which enables an agent to choose to pass or shoot in the presence of a defender, was attempted.

In the memory model, training examples affect neighboring generalized learned instances with different weights. Each soccer agent stores its experiences in an adaptive memory and is able to retrieve them in order to decide upon an action. It has been seen that using an appropriate memory size, the adaptive memory made it possible for the agent to learn both time-varying and non-deterministic concepts. Also short-term performance was shown to be better when acting with a memory.

2.4 Neural Networks

The Artificial Neural Network (ANN) is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The network is composed of a large number of highly interconnected processing elements or neurons working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task. An ANN is configured for a specific application through a learning process.

Neural networks had been successfully used for learning low-level behaviors of soccer agents. This learned behavior, namely shooting a moving ball, equips the clients with the skill necessary to learn higher-level collaborative and adversarial behaviors. The learned behavior enabled the agent to redirect a moving ball with varying speeds and trajectories into specific parts of the goal. By carefully choosing the input representation to the neural networks so that they would generalize as much as possible, the agent was able to use the learned behavior in all quadrants of the field even though it was trained in a single quadrant. In another work, neural networks were used to learn turn angles based on balls distance and angle as a part of a hierarchical layered learning approach.

Neuro-Evolution

A neuro-evolutionary algorithm, which was successfully used in simulated ice hockey, was employed to evolve a player who can execute a dribble the ball to the goal and score behavior in the environment of robot soccer. Both goal-only and composite fitness functions were tried. The evolved players developed rudimentary skills; however it appeared that considerably more computation time is required to get competent players. The fitness of the individuals increases with generations. Also, the goal-only fitness was found more likely to lead to success than the composite fitness function

Using this approach, some of the evolved players exhibited rudimentary dribble and score skills, but the overall results were not very good considering the lengths of the runs. More complex networks with more input variables may lead to the evolution of better players more quickly.

2.5 Layered Learning

Application of layered learning, to a problem consists of breaking the problem into a bottom up hierarchy of sub problems. Then these sub problems are solved in order where each previous sub problem's solution is input to the next layer. Proceeding this way, the original problem will eventually been solved. This approach will simplify the learning task by decomposing the problem to be addressed and will reduce the amount of computation required for learning. The common and basic behaviors can be put in the lower layers and the complex and specialized behaviors in the higher layers.

This technique has been applied to the evolution of client behaviors of robotic soccer players. The layered architecture, which allows machine learning at various levels, is shown in Fig.4. First, the clients learn a low level individual skill that allows them to control the ball effectively. Then, using this learned skill, they learn a higher-level skill that involves multiple players. In the lowest level a neural network was used for learning to intercept a moving ball. This was selected as the lowest layer because it is the prerequisite for executing more complex behaviors. In the second layer a decision tree was used to learn the likelihood that a given pass would succeed. Ie, the agent reasons whether a pass to a particular teammate would succeed. This learned decision tree was used to abstract a very high-dimensional state-space into one manageable for a multi-agent reinforcement learning technique. This was the winning strategy for the CMUnited teams, which has won the RoboCup simulation league more than once.

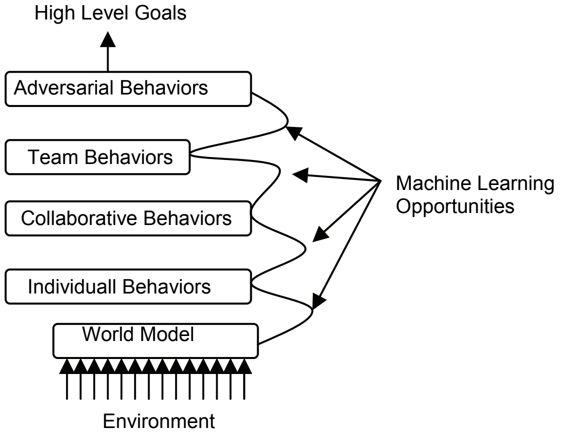


Fig.4. Overview of Layered Learning Framework

It has been proposed that while using layered learning, more layers can be employed for still higher-level behaviors for the soccer agent. It has been shown that layered learning is able to find solutions comparable to standard genetic programs more reliably and in a shorter number of evaluations. The benefits of layered learning over genetic programming were examined by using the evolution of goal scoring behavior in soccer as a test scenario. It was concluded that layered learning is on average able to develop goal-scoring behavior comparable to standard genetic programs more reliably and in a shorter time.

2.5.1 Layered Learning with Genetic Programming

In another attempt, the concept of layered learning was applied with genetic programming where GP is applied to sub problems sequentially, where the population in the last generation of a sub problem is used as the initial population of the next sub problem. By following this approach, multiple fitness functions may be used in each layer so as to evolve fitter individuals. The method has been applied to a sub problem of robotic soccer namely the keepaway, and the results showed that the layered learning GP outperforms standard GP by evolving a lower fitness faster and also an overall better fitness.

Knowledge Discovery in Databases (KDD) based architecture with genetic programming was proposed for strategy learning in robotic soccer. The KDD architecture uses supervised learning where the inductive algorithm is decision tree learning. A layered learning approach was intended, with the core learning method at each level being genetic programming. The proposed hierarchy, extending upon layered learning, uses three levels of learning, three tasks which build upon each other to learn one high level task. Each level will be learned separately using genetic programming in the KDD architecture. The lowest, or primitive, level is that of simply passing a ball to a fixed point. The second level is learning to pass to a player moving in a given direction with a given velocity. Acceleration was not accounted for at the time. The highest-level task was to learn to coordinate among multiple players, the task of moving to an acceptable position to accept or give a pass.

2.5.2 Concurrent Layered Learning

Another variation of layered learning namely concurrent layered learning was proposed in, which may be applied to situations in which the lower layers may be allowed to keep learning concurrently with the training of subsequent layers. Neuro-evolution was used to concurrently learn two layers of a layered learning approach to a simulated robotic soccer keep away task. It was proved that there exist situations where concurrent layered learning outperforms traditional layered learning. Thus it was concluded that concurrent training of layers could be an effective option.

2.6 Genetic Programming

Genetic programming (GP) is an automated method for creating a working computer program from a high-level statement of the problem. It uses evolutionary techniques to learn symbolic functions and algorithms, which operate in some domain environment. It starts from the statement of 'what needs to be done' and automatically creates a computer program to solve the problem. In this aspect it is unlike other learning methods. Most other learning strategies are designed not to develop algorithmic behaviors but to learn a nonlinear function over a discrete set of variables. In contrast with these methods, which are effective for learning low-level behaviors such as intercepting the ball or following it, GP can help to learn the emergent, high-level player coordination.

The evolutionary technique of genetic programming was used to evolve coordinated team behaviors and actions for soccer soft-bots in RoboCup-97. They entered the first international RoboCup competition with two of these teams and qualified to the third round. Also, the scientific challenge award for the best team strategy was awarded to this team in RoboCup 1997.

The problem addressed using the genetic programming was the action selection for the agents. One team was 'homogenous' and the other was 'pseudo-homogenous'. The homogenous team consisted of players with identical programs and the other team was made up of squads. Each squad was composed of three to four identical programs. A program consisted two sub-programs, a kick-tree and a move-tree. The kick-tree was executed when the ball was kickable whereas move-tree was executed otherwise. The soccer players learned to run after the ball and kick it towards the opponent's goal. They also learned some basic defensive abilities. It was the less complex homogenous team that performed the best. However, it was believed that the pseudo-homogenous team would outperform the homogenous team if it was given additional time to evolve. The individual fitness was calculated based on the number of goals only.

Taking inspiration from this work, another attempt was done to evolve soccer playing agents for real robot competitions. The real robots need sophisticated control strategies, which were hand-coded before. They used very low level genes like that for doing arithmetic in contrast to the more high level behaviors used in the above said work. Comparative to the previous work, the team thus developed had a poor quality ball following behavior.

A fitness function for genetic programming, based on the observed hierarchal behavior of human soccer players was proposed in. This fitness function rewarded players by taking into consideration their position, distance to ball, number of goals scored number of kicks and the out come of the game. Each of these has given different weights. Winning the team has been given highest weight since it represents the ultimate goal of the game.

Each team in the population follows the following schedule for fitness evaluation. First, the team is tested against an empty field. It passes this test if it scores within 30 seconds, and fails otherwise. Second, the team plays against a hand-coded team of kicking posts (players that simply stay in one spot, turn to face the ball, and kick it towards the opposite side of the field whenever it is close enough). This promotes teams that can either dribble or pass around obstacles. When a team scores against the kicking posts, it then plays the winning team from the 1997 RoboCup championship, the team from Humboldt University, Germany. Then, only if the team scores at least one goal, it is allowed to play three games in a tournament with other teams who have also made it through these three competition filters. One drawback was that the team thus evolved doesn't have the notions of positions other than that of a goalie. If one player evolves to play on the left side of the field, it will do this independent of whether a teammate is already in this space. This must be learned as part of the evolutionary process. But it has been seen that positioning doesn't evolve in the way that humans enforce it.

Three experiments in the use of genetic programming to create RoboCup players using genetic programming were described in. In the first experiment, the only actions available to the programs were those provided by the soccer server. The second experiment employed higher-level actions such as 'kicking the ball towards the goal' or 'passing to the closest team-mate'. These two experiments used a tournament fitness assignment while the third experiment was a slight modification of the first. In the third experiment, the terminals and low-level functions in experiment 1 were used along with some additional terminals and functions. The teams created by the first and third approaches performed poorly. The players from the second experiment were able to follow the ball and kick it around.

The work showed that a team, which was generated by evolving a player with the basic functions and making 11 copies of it, performs fairly well. Having a designated goalie also improves the team performance. It was concluded that the use of genetic programming enabled teams to perform well. Higher-level functions and better fitness measure may improve this method further.

2.7 Hybrid Approaches

There have been approaches, which combines various multi agent methods for developing agent strategies for soccer agents. One of the significant works, which combines several multi agent strategies in order to arrive at efficient soccer team, was the CMUnited teams which participated in the RoboCup simulation league form the first RoboCup. The CMUnited 97 team used layered learning approach with locker room agreement as their strategy. Improving upon that, the CMUnited-98 simulator team used the following multi-agent techniques to achieve adaptive coordination among team members.

Hierarchical machine learning (Layered learning): Three learned layers were linked together for layered learning. Neural networks were used by individual players to learn how to intercept a moving ball. With the receivers and opponents using this first learned behavior to try to receive or intercept passes, a decision tree was used to learn the likelihood that a given pass would succeed. This learned decision tree was used to abstract a very high-dimensional state-space into one manageable for the multi-agent reinforcement learning technique TPOT-RL.

Flexible, adaptive formations (Locker-room agreement): Locker-room agreement includes a flexible team structure that allows homogeneous agents to switch roles (positions such as defender or attacker) within a single formation

Single-channel, low-bandwidth communication: Use of single-channel, low-bandwidth communication, ensures that all agents must broadcast their messages on a single channel so that nearby agents on both teams can hear; there is a limited range of communication; and there is a limited hearing capacity so that message transmission is unreliable.

Predictive, locally optimal skills (PLOS): Predictive, Locally Optimal Skills was another significant improvement of the CMUnited-98 over the CMUnited-97 simulator teams. Locally optimal both in time and in space, PLOS was used to create sophisticate low-level behaviors, including dribbling the ball while keeping it away from opponents, fast ball interception, flexible kicking that trades off between power and speed of release based on opponent positions and desired eventual ball speed, a goaltender that decides when to hold its position and when to advance towards the ball based on opponent positions.

Strategic positioning using attraction and repulsion (SPAR): SPAR determines the optimal positioning as the solution to a linear-programming based optimization problem with a multiple-objective function subject to several constraints. The agent's positioning is based upon teammate and adversary locations, as well as the ball's and the attacking goal's location.

Team-Partitioned, Opaque Transition Reinforcement Learning (TPOT-RL): TPOT-RL allows a team of agents to learn to cooperate towards the achievement of a specific goal.

The team, which used this strategy, was the RoboCup 98 simulation league champion. Improvements upon these strategies were presented by CMUnited-99, which was the world champion of RoboCup 99. The low-level skills were improved and updated to deal with server changes, the use of opponent and teammate models was introduced, and some coordination procedures were improved, and a development paradigm called layered disclosure was introduced, by which autonomous agents include in their architecture the foundations necessary to allow a person to probe into the specific reasons for an agent's action.

It can be seen that hybrid approaches were effective in the soccer domain. Hybrid-methods may be able to capture the complexity of the soccer domain better because the problem itself demands solution to a combination of different multi-agent issues. Also the evolutionary approach genetic programming is also found to evolve good and effective strategies through automatically. This approach has the advantage of giving multiple strategies, which are equally good.

3. Ant Intelligence in Robotic Soccer

Even though ant is small creature it exhibits different behaviors which turned the world of computers to think about the behaviors of social insects and make them to adapt to solve the problems. The few behaviors of ant are given below.

3.1 Ant Intelligence

Foraging : Biologists have found some obvious distinctive features of real ants in the process of looking for food. Ants release some chemical substance called *pheromone* while moving. The released pheromone will lessen gradually along with the passing of time. An ant can detect the existence of intra-class pheromone trail within a given area; then it will move along the path on which pheromone trail is plentiful. Ant can find a shortest from nest to food source based on this collective pheromone-laying or pheromone-following behavior.

Nest protection : Ants fight in order to monopolize a food resource or to protect their nest. Fight includes aggression against other insects attempting to steal their food. Some ants move away if defeated. This behavior suggests that the ants prefer to keep their nests apart in order to live peacefully. During a fight, a poisonous fluid called formic acid is sprayed on the foe. Some ants spray formic acid from above by curving the abdomen upwards behind the body. Ant sprays the juice from below by curving the abdomen upwards in front of the body.

Stigmergy : This behavior is exhibited when there is large prey to carry out. For this they need cooperation and coordination of other ants to carry that large prey. In general based on the task difficulty the number of ant are recruited to solve the task.

3.2 Soccer players with ant intelligence

The ant intelligence discussed in the previous subsection are incorporated in newly evolved team (PUTeam).

The pheromone following technique is an indirect signalling to have a cooperation and coordination among the set of players. A player who moves towards the ball lays the pheromone as an indirect signal for its neighbouring forwarders to take the relative positions in accordance to the pheromone layed by it.

Nest protecting mechanism is adopted by the golie, as an indirect signal to its teammates that he is strong enough to block the goal of opponents.

The concept of stigmergy is exhibited for blocking the opponent. Inorder to defend the strong opponent there is a need for two or more players. In such situations the stigmergy technique is incorporated. A player who goes for blocking the stronger opponent will give an indirect signal (Technique of laying required quantities of pheromones) to its neighbouring defenders to come for the block zone.

4. Experimental Results

The experimental results were obtained with the multi-agent simulation tool (teambots). Soccerbots is the domain of teambots package which provides a support to test the newly evolved team (PUTeam) against the other available opponent teams. The results of this work are quite promissing. There are 20 teams in soccerbots which are listed below.

AIKHomog
DoogHomoG
*BrianDemo
*BasicTeam
*CDTeamHetero
*CommonTeam
*FemmHeteroG

DoogHeteroG
MattiHeteroG
*SchemaDemo
*BriSpec
*DaveHeteroG
*DTeam
*PermHomoG

*SchemanewHeteroG
*GotoBall
*Kechze
*SibheteroG
*JunTeamHeteroG
*LoneforwardTeamHomoG

Out of these 20 teams the Bio-Inspired PUTeam (our team) has won the match against 16 teams which are marked with * in the above list. We conducted four runs, each run of 50 matches. The following table 1 depicts the average of number of goals scored by PUTeam with the remaining 4 teams in SoccerBots and thereby assessing the success of PUTeam.

Runs	Average of 50 matches			
	DooghomoG Vs Puteam	DoogHeteroG Vs PUteam	MattiHeteroG Vs PUteam	AIKHomoG Vs PUteam
Run 1	0- 3(won)	3 -7 (won)	0 -3 (won)	1 -3 (won)
Run2	1- 4 (won)	0 -3 (won)	0 -2 (won)	1 -2 (won)
Run3	1-3 (draw)	6 -0 (lost)	0 -5 (won)	0 -3 (won)
Run4	2-2(draw)	1 -5 (won)	2 -0 (lost)	3 -6 (won)
Run5	3-3(draw)	1-1(draw)	0-2(won)	1-1(draw)
Run6	2-3(won)	2-5(won)	0-2(won)	0-3(won)
Run7	3-2(lost)	1-4(won)	1-1(draw)	0-1(won)
Run8	0-4(won)	2-4(won)	0-5(won)	3-2(lost)
Run9	0-3(won)	1-4(won)	1-2(won)	1-3(won)
Run10	0-2(won)	3-1(lost)	0-2(won)	0-2(won)

Table 1. Success results of PUTeam against top 4 teams of Teambots.

Thus the Percentage of win over the remaining top four teams has reached about 85% through the influence of ants behavior in player strategies. To achieve 100% success, further enhancements can be made by incorporating more bio-inspired behaviors.

5. Conclusion and Future work

The idea of mapping social insect behaviors in robotic soccer gave good insight to think in that direction and based on the experimental results it shows that it is giving promising results. As the part of futurework we are extending this by incorporating few more social insect behaviors.

6. References

Bonabeau, Dorigo & G. Theraulaz. Inspiration for Optimization from Social Insect Behaviour, *Nature* 406, (2000) 39 – 42.

E. Shaw, *The schooling of fishes*, Sci. Am., vol. 206, pp. 128-138, 1962.

E. Crist. Can an Insect Speak? The Case of the Honeybee Dance Language. *Social Studies of Science*. SSS and Sage Publications. 34(1), pp. 7-43.

http://biology.arizona.edu/sciconn/lessons2/shindelman/background.html

Jesper Bach Larsen. Specialization and division of labor in distributed autonomous agents. *Master's thesis*, Dept. of Computer Science, University of Aarhus, Denmark, 2001

Kiyohiko Hatturi, Ynshimasa Narita, Yoshiki Kaahimori and Takeahi Kambara, *Self-Organized critical behavior of fish schools and emergence of group intelligence,* IEEE 1999.

LAURA HELMUTH, *Spider Solidarity For ever, Social spiders create the communes of the arachnid world.* Vol 155 No. 19 p. 300, May 8, 1999.

Merke, A. and Riedmiller, M. Karlsruhe Brainstormers - a reinforcement learning way to robotic soccer. In *RoboCup 2001: Robot Soccer World Cup V*. Springer, Berlin, 2002

Minoru Asada, Hiroaki Kitano, Itsuki Noda, Manuela Veloso RoboCup: Today and tomorrow—What we have learned. *Artificial Intelligence* 110 (1999) 193–214

Parpinelli, R.S. , Lopes, H.S. and Freitas, A.A. An ant colony based system for data mining: applications to medical data. *Proc. 2001 Genetic and Evolutionary Computation Conf. (GECCO-2001)*, pp. 791-798. Morgan Kaufmann, 2001

Park, K. H., Y. J. Kim and J.-H. Kim, Modular Q-learning based Multi-agent Cooperation for Robot Soccer, *Robotics and Autonomous Systems*, vol. 35, no. 2, May. 2001, pp. 109-122.

Peter Stone and Manuela Veloso. A Layered Approach to Learning Client Behaviors in the RoboCup Soccer Server. *Applied Artificial Intelligence,* 12:165–188, 1998

Shimon Whiteson and Peter Stone. Concurrent Layered Learning. In *Second International Joint Conference on Autonomous Agents and Multi-agent Systems*, pp. 193–200, ACM Press, New York, NY, July 2003

Shu-Chuan Chu1, Pei-wei Tsai2, and Jeng-Shyang Pan2, *Cat Swarm Optimization,* PRICAI 2006, LNAI 4099, pp. 854 – 858, 2006.

Tatlidede, U., K. Kaplan, H. Kose, and H. L. Akın, Reinforcement Learning for Multi-Agent Coordination in Robot Soccer Domain, *AAMAS'05 Fifth European Workshop on Adaptive Agents and Multi-Agent Systems*, Paris, March 21-22 2005.

The official website of Robotic soccer http://www.robocup.org/

Uchibe E., Asada M., Noda S., Takahashi Y., Hosoda K., Vision-Based Reinforcement Learning for RoboCup: Towards Real Robot Competition. *Proc. of IROS 96 Workshop on RoboCup*, 1996.

Vic Ciesielski, Dylan Mawhinney, and Peter Wilson. Genetic programming for robot soccer. In Andreas Birk, Silvia Coradeschi, and Satoshi Tadokoro, editors, *Proceedings of the RoboCup 2001 International Symposium,* Lecture Notes in Artificial Intelligence 2377, pages 319-324. Springer, 2002.