Beyond Individualism: Modeling Team Playing Behavior in Robot Soccer Through Case-Based Reasoning*

Raquel Ros[1], **Manuela Veloso**[2], **Ramon López de Mántaras**[1], **Carles Sierra**[1] and **Josep Lluis Arcos**[1]

[1] IIIA - Artificial Intelligence Research Institute
CSIC - Spanish Council for Scientific Research
Campus UAB, 08193 Barcelona, Spain

[2] Computer Science Department
Carnegie Mellon University
Pittsburgh PA 15213, USA

Abstract

We propose a Case-Based Reasoning approach for action selection in the robot soccer domain presented in the 8th European Conference on Case-Based Reasoning (2006). Based on the current state of a game, the robots retrieve the most similar past situation and then the team reproduces the sequence of actions performed in that occasion. In this domain we have to deal with all the difficulties that a real environment involves.

Introduction

This paper summarizes previous work presented in (Ros *et al.* 2006; 2007) where we proposed a Case-Based Reasoning (CBR) approach for action selection in the robot soccer domain (Four-Legged League). In CBR new problems are solved by reusing and if necessary adapting the solutions to similar problems that were solved in the past (López de Màntaras *et al.* 2006). Action selection in robotics is a challenging task: the robot has to reason about its world beliefs, i.e. the state of the environment, and rationally act in consequence in order to complete a task (typically divided in subtasks). Moreover, in the case of a robot team, robots must agree on the decisions made (who and what to do to complete the subtasks), jointly execute the actions, and coordinate among them to successfully perform the task. Working with real robots has additional difficulties that must be considered. Thus, the reasoning engine must be capable of dealing with high uncertainty in the robot's perception (incoming information of the world), and be robust in case of failure, since the outcomes of the actions performed are unpredictable. Not to mention that decision must be made in real time and in our case, with limited computational resources. The work we present here tries to solve these problems using CBR techniques, and moreover, we allow the robots to perform explicit passes guided through cases. To the best of our knowledge, in the Four-Legged League passes occur by chance, i.e. they are not planned through any explicit mechanism. The paper is organized as follows: first we analyze the difficulties in the action selection problem and the related work. Next we briefly present the methodology we use to solve the problem described and most significant results obtained so far. Finally, we present some conclusions.

The problem

Having an agent autonomously decide which actions to perform given the current state of the environment is a hard task addressed by researchers in different fields of AI. The degree of difficulty varies depending on the kind of environment and the type and number of agents involved in the task.

Trying to design completely controllable environments in the real world is unfeasible since unpredictable situations always occur. Creating highly controlled scenarios decreases the difficulty of the task, but it also results in less realistic scenarios. In this kind of environments, dynamic and unpredictable, the agent must be capable of detecting if the actions selected for a given state of the environment are still applicable when the state evolves. If they are, then the agent continues with the initial plan. Otherwise, it must either correct the selected actions or re-plan.

The problem becomes even harder if besides not having a completely controllable scenario the agent is an autonomous robot. In this case, we also have to deal with imprecision, not only in the incoming information from the world (perception), but also, given the physical body of the robot and the imprecision of its mechanics, on the outcome of the actions the robot performs. Moreover, if instead of having a single robot we have a team of robots the task becomes even more challenging. Some of the questions that arise under these circumstances are: who decides what to do?, i.e. a single agent decides which actions to perform or all agents discuss the selected actions; who does what?, i.e. one agent is selected to perform the complete task or each agent may perform part of the task or subtasks; who monitors the task execution?, i.e. one agent receives all the information from the rest of the agents and decides by itself or each agent has its own "beliefs" of the world and reacts accordingly.

Within this framework we introduce a problem domain which fulfills most of the characteristics presented above: robot soccer. In this domain we face a dynamic and unpre-

*Partial funding by the Spanish Ministry of Education and Science project MID-CBR (TIN2006-15140-C03-01) and partly sponsored by BBNT Solutions, LLC under contract no. FA8760-04-C-0002. Raquel Ros holds a scholarship from the Generalitat de Catalunya Government. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution or any other entity.

dictable environment where a team of robots selects the actions to perform in order to reach its goal: to win a game. More precisely, we focus our work on the Four-Legged League, one of the leagues in the RoboCup competition.

In the past years researchers have presented different approaches to solve the problem of action selection in the Simulated League. (Riedmiller *et al.* 2000) proposed the use of Reinforcement Learning techniques. Although the state space of this problem is very large (the number of different states is huge due the nature of the environment), they divide the problem in two levels: a low level focused on single moves (such as learning how to kick) and a tactical level focused on game strategies. (Lattner *et al.* 2005) proposed a behavior prediction approach based on pattern recognition. A pattern is a qualitative representation of the state of the world at a given time. From a set of past game logs they generate a list of patterns to eventually deduce prediction rules describing what (future) robot actions or situations might occur with some probability if certain preconditions are satisfied. They use these rules during a game in order to predict the next state of the game and therefore, select the most appropriate action for a robot in the current state. (Lam, Esfandiari, & Tudino 2006) focused their research on learning from observation. The aim of this technique is to model agents that learn from observing other agents and imitating their behavior. As in Case-Based Reasoning, the learning agent selects the most similar past observed situation with respect to the current problem and then reproduces the solution performed at that time. The main difference between these approaches is that the learning agent is not able to improve the observed agent since there is no feedback in the model. They presented preliminary work with one robot and single action selection.

Case-Based Reasoning has been used in different occasions to solve planning problems as summarized in (Cox, Muñoz-Avila, & Bergmann 2005). Action selection can also be seen as planning, where a plan is the sequence of actions the agent must perform. Hence, researchers have opted for applying this technique in the robot soccer domain as well. Some examples are (Wendler & Lenz 1998) in the Simulation League, (Karol *et al.* 2003) in the Four Legged League, and (Marling *et al.* 2003) in the Small Size League. Although the different leagues in the Robocup competition have a common objective (agents that play soccer), the difficulties among them vary. Hence, the Simulated league has the advantage of having a simulated environment where unpredictable situations do not occur, but there is a large number of parameters to model the agents behaviors. In the Small Sized league, an off-field PC processes the vision information sent by an overhead camera and typically performs most, if not all, of the processing required for coordination and control of the robots. The main challenge in this league is that the game speed is very high, and therefore, the reasoning has to be fast. Finally, in the Four-Legged league the robots are fully autonomous, i.e. each robot perceives, reasons and actuates independently. The challenge here is to have the robots deciding the appropriate actions by themselves in a real environment with limited computational resources and in a cooperative way.

In the Four-Legged League teams consist of four Sony AIBO robots. There are two goals, cyan and yellow, and four colored markers the robots use to localize themselves in the field. The robots can communicate with each other by wireless. A game consists of two parts of 10 minutes each.

The methodology

A case represents a snapshot of the environment at a given time from a single robot point of view. We call this robot the *reference* robot, since the information in the case is based on its perception and internal state (its beliefs). The case definition is composed of three parts: the problem description, which corresponds to the state of the game; the knowledge description, which contains additional information used to retrieve the case; and finally, the solution description, which indicates the sequence of actions the robots should perform to solve the problem. We formally define a case as a 3-tuple:

$$case = ((R, B, G, Tm, Opp, t, S), K, A)$$

where:

1. *R*: relative position wrt the ball and heading of the *reference* robot.
2. *B*: ball's global position.
3. *G*: defending goal.
4. *Tm*: teammates' relative positions wrt the ball.
5. *Opp*: opponents' relative positions wrt the ball.
6. *t*: timing of the match.
7. *S*: difference between the goals scored.
8. *K*: scope of the case defined as the regions of the field within which the ball and the opponents should be positioned in order to retrieve that case. With this representation we can easily handle imprecision since we refer to regions instead of exact locations in the field.
9. *A*: sequence of actions (gameplays) each robot performs.

The first step in CBR is the retrieval of past similar cases in order to reuse the solution of one of the retrieved cases. We evaluate similarity along two important measures: the similarity between the problem and the case, and the cost of adapting the problem to the case. Thus, we separate the features in the problem description into two sets: *controllable* indices and *non-controllable* indices. The former refers to the *reference* robot's and teammates' positions (since they can move to more appropriate positions), while the latter refers to the ball's and opponents' position, the defending goal, time and score (which we cannot directly modify). The idea of separating the features is that a case can be retrieved if we can modify part of the current problem description in order to adapt it to the description of the case.

Similarity function: This measure indicates how similar the non-controllable features are between the problem and the case. We define different functions for each domain of features and we then compute the overall similarity using the harmonic mean of the individual similarities.

Cost function: This measure computes the cost of modifying the controllable features, i.e. the cost of adapting the problem to the case. We define it as the sum of the distances between the positions of the robots in the problem and the adapted positions specified in the case after obtaining their correspondences. The adapted positions correspond to the global locations where the robots should be positioned in order to execute the solution of the case.

Cases are manually created and stored in a file. When the system loads them, for each case the system automatically generates three more cases through spatial transformations taking into account the symmetry of the field. Since we are working in a real time domain and because of computational limitations in the robots, it is essential to minimize the time invested during the retrieval process. To speed up the search we use an indexed list to store the cases. Thus, given a new problem we can easily access the subset of cases (CB^s) we are interested in by indexing the case base using the value of the defending goal (yellow or cyan) and the number of opponents involved in each case.

After computing the similarities and costs between the problem and the cases in CB^s, we obtain a list of potential cases. To select the retrieved case we consider a compromise between the similarity degree between the problem and the case and the cost of adapting the problem to the case. Moreover, since we are working in a multi-robot domain (teams of robots), we are also interested in stimulating cooperation between them as much as possible. Therefore, the retrieval process orders the list of potential cases such that we maximize the similarity and number of players involved in the solution of the problem, while minizing the cost.

Our multi-robot system is composed of n robots. All robots interact with the environment and among themselves, i.e. they perceive the world, they perform actions and they send messages to each other to coordinate and to exchange information about their internal states. Each robot has a copy of the same case base so they can gather the information needed during the case reuse.

Given a new state of the environment the first step is to select the robot responsible for the retrieval process and for the coordination of the robots during the case reuse. We refer to this robot as the *coordinator*. We base the selection on the distance between the robots and the ball. The further a robot is from an object, the higher the imprecision about the object's information. Therefore, the *coordinator* corresponds to the one closer to the ball. Next, the *coordinator* retrieves a case according to the process described before and informs the rest of the robots which case to reuse.

At this point the case execution begins. Firstly, all robots that take part of the solution of the case move to their adapted positions. Once they reach them, they send a message to the *coordinator* in order to synchronize the beginning of the gameplay execution with the rest of the robots. Next, they all execute their actions until ending their sequences. Finally, they report the *coordinator* that they finished the execution and wait for the rest of the robots to end. When the *coordinator* receives all messages, it informs the robots so they all go back to the initial state of the process, i.e. selecting a new *coordinator*, retrieving a case and executing its solution. The robots may abort the execution of a case at any moment if any of the robots either detects that the retrieved case is not applicable anymore or an expected message does not arrive. In either case, the robot sends an aborting message to the rest of the robots so they all stop executing their actions and restart the process.

Results

We next describe two types of experiments. The goal of the first set of experiments is to test the correctness of the retrieval stage, i.e. to verify if given a set of problems to solve the retrieval process obtains the expected cases. We focused the second set on evaluating the resulting behavior of the team having the robots retrieve and reuse cases. We performed the first set of experiments in simulation, and with real robots for the second.

Retrieval Performance

At this first stage of the work, we empirically obtained the values for the thresholds used during the retrieval process. We also evaluated four aggregation functions to compute the overall similarity: the mean, the weighted mean, the minimum and the harmonic mean. We opted for the harmonic mean due to its property of taking into account all values as much as possible but highlighting the lower ones.

We manually defined 90 cases with one player, i.e. no teammates, varying the number of opponents, the time and the score difference. We then randomly created 50 new problems and then manually labeled them. The retrieval process correctly classified all the problems, i.e. always retrieved the case indicated in the labeled problem. It also computed the adapted position the robot should take and the actions to perform from that point on.

Multi-robot Performance

To evaluate the performance of our case-based approach, we have compared it with a behavior-based approach. This second approach consists in defining high level behaviors (state-based behaviors) the robot executes based on the state of the environment. For example, a robot defending its goal should get the ball and clear it from the defense region. To prevent collisions between robots, when a robot decides to go after the ball it informs its teammates so they try to move away from its trajectory and eventually, the possible trajectory of the ball to avoid intercepting it.

The goal of this experiment is to prove that the resulting behavior of the robot team using our approach is more cooperative than a robot team using the behavior-based approach. In other words, our approach results in a collective or "team playing" behavior (participation of more than one robot of the same team during the execution of a task) with explicit passes, as opposed to individual behavior (only one robot executing the task) where passes occur only by chance.

A trial consists in positioning the robots and the ball on the field and the robots' task is to move the ball until reaching the penalty area (rectangular box in front the attacking goal). We designed two sets of experiments, each composed of 15 trials. In the first set we had two robots from the same team positioned on the right middle side of the field. The second set was more complex, since besides the two robots positioned in the left middle back side of the field, we also included a fixed opponent (the goalie). We tested each in both sets of experiments.

During the experiments with the behavior-based approach, we observed that due to its individualistic nature, in general only one robot was involved in the execution of the task. From the 30 trials (15 for each scenario), 4 times the ball went out of field failing the experiment. Although the remaining trials were completed, a single robot always pursued the ball while the second robot remained behind it to avoid intercepting either the first robot or the ball. Hence, for an external observer, the performance lacked of cooperation although the robot was actually avoiding to cross the path of the first robot.

Regarding the case-based approach, in both scenarios the first retrieved cases were always the same for each layout since the initial positions are fixed. From that point on, because of the non-deterministic nature of the environment (the ball's trajectory is not exactly the same, a robot may lose the ball when attempting to grab it, the kick strength can be stronger or weaker, etc.) based on the events occurred during the execution, the next retrieved case may vary. In any case, the robots always made a good decision and performed the task successfully and in a cooperative way having passes between them to increase the control of the ball. Figure 1 illustrates an example of two executions, one for each scenario. From a total of 65 cases, 57 were correctly retrieved and successfully executed. The 8 remaining were initially incorrectly retrieved from an observer point of view. However, due to localization errors, from the robots' point of view the cases matched the state of the environment at the retrieving stage. From the moment they correctly localized themselves in the field, they realized that the cases did not match the state of the environment and aborted the execution. Afterwards, the robots retrieved the correct case and eventually, completed the task.

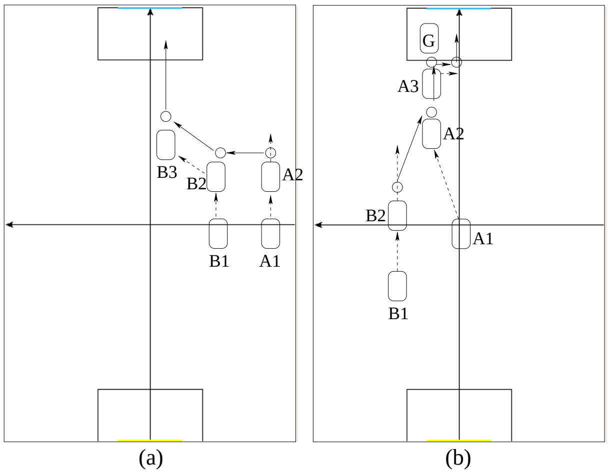

Figure 1: The letters correspond to the robots (A, B and G), the numbers to the time step of the execution (1, 2 and 3) and the arrows represent the ball's and robots' movement, solid and dashed respectively. (a) Scenario 1 using the case-based approach: "multiple right side" case followed by "single right middle" case. (b) Scenario 2 using the case-based approach: "multiple left middle" case followed by "single goalie front" case.

Conclusions

In this paper we have addressed the problem of action selection in the robot soccer domain. This domain has the characteristic of dealing with an unpredictable, dynamic and imprecise environment that requires a real time response. We have addressed this problem with a case-based approach. The results show that the CBR approach produces a collective "team playing" behavior where passes between players are explicitly indicated in the cases as opposed to and individual behavior. We believe that the approach we have presented might be applicable to domains other than robot soccer with similar characteristics.

References

Cox, M. T.; Muñoz-Avila, H.; and Bergmann, R. 2005. Case-base planning. *The Knowledge Engineering Review* 20:283–287.

Karol, A.; Nebel, B.; Stanton, C.; and Williams, M. 2003. Case Based Game Play in the RoboCup Four-Legged League Part I: The Theoretical Model. In *RoboCup*, volume 3020 of *LNCS*, 739–747.

Lam, K.; Esfandiari, B.; and Tudino, D. 2006. A Scene-based Imitation Framework for RoboCup Clients. In *MOO - Modeling Other Agents from Observations*.

Lattner, A.; Miene, A.; Visser, U.; and Herzog, O. 2005. Sequential Pattern Mining for Situation and Behavior Prediction in Simulated Robotic Soccer. In *RoboCup*, volume 4020 of *LNCS*, 118–129.

López de Màntaras, R.; McSherry, D.; Bridge, D.; Leake, D.; Smyth, B.; Craw, S.; Faltings, B.; Maher, M.-L.; Cox, M.; Forbus, K.; Keane, M.; and Watson, I. 2006. Retrieval, Reuse, Revise, and Retention in CBR. *The Knowledge Engineering Review* 20(3):215–240.

Marling, C.; Tomko, M.; Gillen, M.; Alexander, D.; and Chelberg, D. 2003. Case-Based Reasoning for Planning and World Modeling in the RoboCup Small Size League. In *IJCAI Workshop on Issues in Designing Physical Agents for Dynamic Real-Time Environments*.

Riedmiller, M.; Merke, A.; Meier, D.; Hoffmann, A.; Sinner, A.; Thate, O.; and Ehrmann, R. 2000. Karlsruhe Brainstormers - A Reinforcement Learning Approach to Robotic Soccer. In *RoboCup*, volume 2019 of *LNCS*, 367–372.

Ros, R.; Veloso, M.; López de Màntaras, R.; Sierra, C.; and Arcos, J. 2006. Retrieving and Reusing Game Plays for Robot Soccer. In *8th European Conf. on Case-Based Reasoning*, volume 4106 of *LNAI*, 47–61. Best Paper Award.

Ros, R.; Veloso, M.; López de Màntaras, R.; Sierra, C.; and Arcos, J. 2007. Team Playing Behavior in Robot Soccer: A Case-Based Approach. In *International Conf. on Case-Based Reasoning*. To appear.

Wendler, J., and Lenz, M. 1998. CBR for Dynamic Situation Assessment in an Agent-Oriented Setting. In *Proc. of AAAI-98 Workshop on CBR Integrations*.