Decision Tree Based Task Allocation Algorithm For Soccer Domain

Dr.R.Subramanian[1], Ms.R.Geetharamani[2], Ms.P.Prathima[3], Ms.M.Sangeetha[3]

[1]Reader, Head In-Charge, Department of Computer Science, Pondicherry University, Pondicherry – 605 014

[2]Lecturer, Department of Computer Science, Pondicherry University, Pondicherry – 605 014
rgeetha@yahoo.com

[3] Department of Computer Science, Pondicherry University, Pondicherry – 605 014
mother@satyam.net.in, {palani_cikki, sangsasi}@yahoo.com

Phone No.: 91- 0413 - 253698

Abstract. The RoboCup is considered as a vehicle to promote robotics and AI research by providing a common task namely Robotic Soccer (Robots playing soccer). The ultimate goal of RoboCup is to ``develop a robot soccer team which beats human world champion team.'' This goal is to be one of the grand challenges shared by robotics and AI community for the next 50 years. The RoboCup competitions include different leagues, namely simulation, small-sized robots, medium-sized robots, and Sony legged robots etc. Researchers in each of these leagues investigate different aspects of the robotic soccer technical challenges.

RoboCup challenge offers research in a friendly environment that is dynamic, real-time and multi-agent involving a lot of teamwork. Such collaboration can be achieved only if the decision mechanism is quick and dynamic adopting the changes in the world around. This paper discusses about decision trees, which are considered to be a useful tool for the representation of decision-making.

In this paper two behaviors of soccer agents have been considered namely attack and defense. The agent's perception of the world has been represented as attributes of the dynamic world. The tasks that an agent has to perform in a given scenario depend on the boolean values taken by these attributes and are formulated as decision trees based on the roles that an agent can assume namely defender, forwarder and goalkeeper. Based upon the attributes' boolean and heuristic values applied on to the leaf nodes of the decision trees, an ultimate task is invoked for an agent as the final decision. This decision mechanism is used in our test bed software developed for simulation league.

Keywords. Artificial intelligence, Robotic soccer, Multi-agent system, Decision trees, Heuristic values.

1. Introduction

Research in Artificial Intelligence and Robotics requires a specific but very attractive real time multi-agent environment and for this was chosen the domain of soccer game, which led to the RoboCup [1], [11] initiative. Although RoboCup's primary objective is a soccer game with real robots, RoboCup also offers an opportunity for research on the software aspects of AI and robotics. The software robot league, also called the simulator league, which uses software to play soccer games on an official soccer server over the network, enables more researchers to take part in this event. It also promotes research on network-based multi-agent interactions, computer graphics, and physically realistic animations.

Of the challenges offered by Robotic soccer [1], [8], teamwork challenge and opponent modeling challenge have to be mentioned. Teamwork challenge addresses issues of real time planning, re-planning and execution of multi-agent teamwork in a dynamic adversarial environment. Team plan execution during the game is the determining factor in the performance of the team. Opponent modeling involves individual players' real-time dynamic tracking of the opponents' goals and intentions based on the observations of their actions and to react appropriately. As is obvious from the challenges, multi-agent soccer team must include good decision-making.

In the Soccer domain the agents have to be provided with suitable task decisions for responding to a particular situation. These task decisions involve several agents working together to manage an objective. The main aim of this paper is to present a simple and intuitive description and specification of these decisions. The decision trees are promising candidates for the representation of the decision mechanism of the agents.

2. Robotic Soccer Agents

A RoboCup agent is driven by a sequence of desires, i.e. objectives the agent wants or has to achieve in order to fulfill its global task. In general the agent has several candidate plans for achieving a objective at its disposal. It will choose one of them to execute based upon its belief about the current state of the world. The chosen plan is called the intended plan. In this discussion beliefs are referred to as attributes, desires as tasks and intended plan as the final task that is assigned to the agent.

Before the desires, beliefs and intentions of the agents [4] can be formulated, it is necessary to understand cooperative behavior of agents' team, which depends on whether the team has or does not have the ball. Hence, the behaviors of the agents are discussed in section 2.1, which is followed by the identification of attributes and tasks of the agents and the encoding of the behaviors into decision trees in sections 2.2 and 2.3 respectively.

2.1 Agent Behavior

In Robotic Soccer an agent can be in one of the two behaviors [2], [12] namely attack and defense. An agent is in attack behavior when the ball is with it or with its team. An agent takes up defense when the opponent team has the ball.

The *attack* behavior can be symbolically shown as:

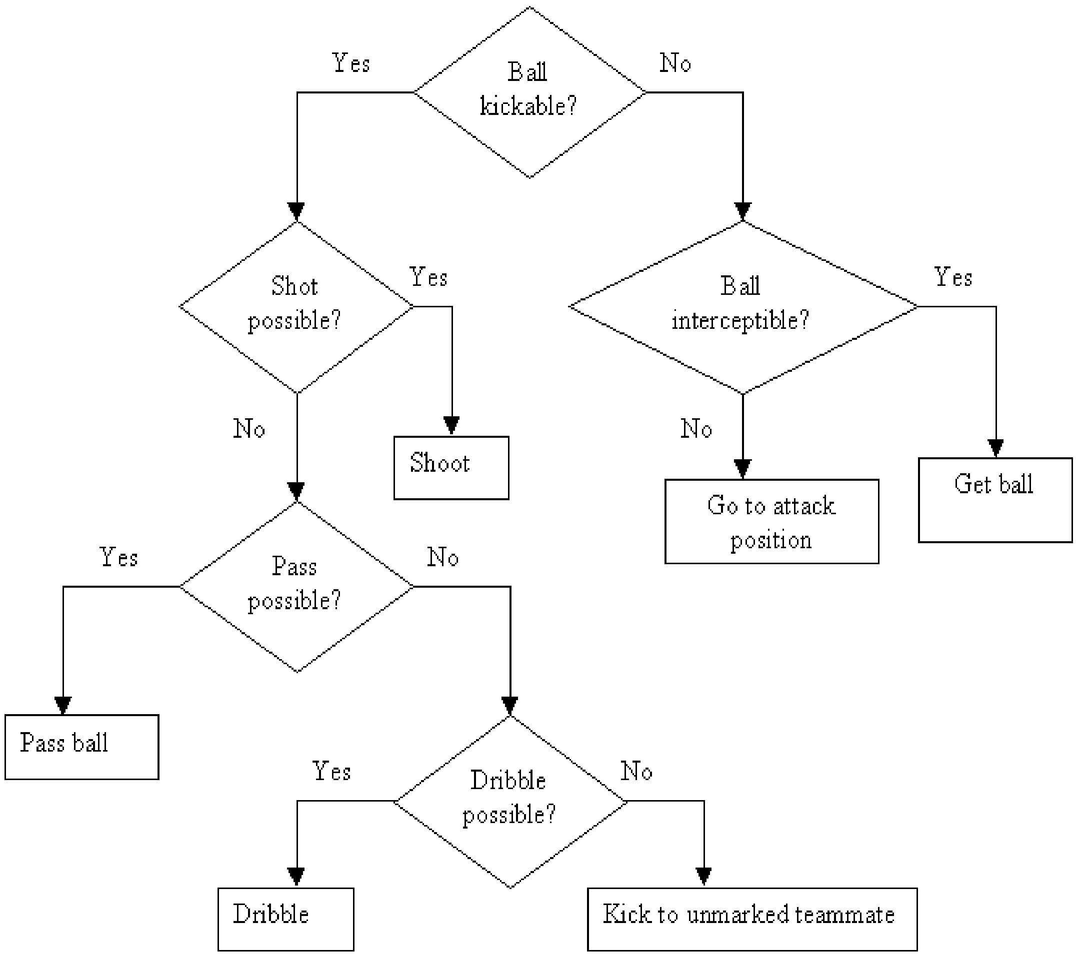

Fig.1. Attack behavior of an agent

In *defense* behavior, the agent does one of the tasks shown below

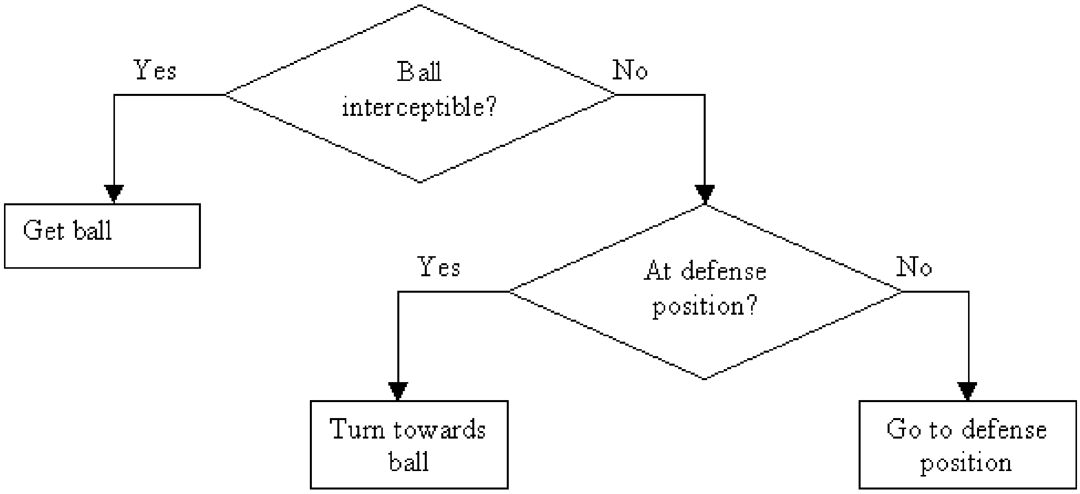

Fig.2. Defense behavior of an agent

2.2 Decision Trees

As cited earlier, the behaviors of the agents have been represented symbolically as a sequence of task decision that they have to make [4], [6]. These decisions have to be made based on the current state of the world. Therefore the attributes representing the belief of the agent about the world have been identified and the various desires of an agent have been framed based on the boolean values (true/false) of these attributes. The attributes making up the belief and the tasks making up the desires of an agent are listed in the tables 1 and 2.

Table 1. Attribute table

Attribute	Description	Attribute	Description
A1	Ball in view	A27	Goalkeeper with ball
A2	Ball in kicking range	A28	Teammate in view
A3	Ball in catching distance	A29	Goalpost in view
A4	Ball in risk distance	A30	Opponent goal post in view
A5	Ball close to goal post	A31	Teammate close to ball
A6	Ball close to opponent's goal post	A32	Opponent close to ball
A7	Ball close to quarter	A33	Opponent close to teammate
A8	Ball close to half	A34	Unmarked teammate close to half
A9	Ball close to opponent's quarter	A35	Unmarked teammate close to opponent's quarter
A10	Ball is kicked off	A36	Opponent in kicking range
A11	Ball placed for kicked off	A37	Unmarked defender in passing distance
A12	Ball is free	A38	Unmarked forwarder in passing distance
A13	Defender close by	A39	Teammate close to nearest opponent
A14	Forwarder close by	A40	Reached stopping distance
A15	Goal keeper in passing distance	A41	Teammate dribbling with ball
A16	Teammate close by	A42	Unmarked teammate in kicking distance
A17	Opponent close by	A43	Teammate in audible range
A18	Teammate close to goal post	A44	Player in audible range
A19	Opponent close to goal post	A45	Ball in intercepting range

A20	Opponent goalkeeper close to goal post	A46	Ball passed off
A21	Opponent close to his goal post (defend)	A47	Teammate in throwing distance
A22	Unmarked teammate close to opponent's goal post	A48	Ball in goalkeeper's hand
A23	Teammate goalkeeper close to goal post	A49	Unmarked teammate in *ball intercepting range*
A24	Defender close to goal post	A50	Player is the closest to *opponent with ball*
A25	Teammate with ball	A51	Unmarked teammate close to quarter
A26	Opponent with ball	A52	Opponent in intercepting range

Table 2. Task table

Task	Description
T1	Turn towards ball
T2	Move towards ball
T3	Move towards teammate
T4	Move towards opponent
T5	Move towards goal post
T6	Move towards opponent's goalpost
T7	Kick towards opponent goalpost
T8	Kick towards unmarked teammate
T9	Pass to defender
T10	Pass to forwarder
T11	Pass to goal keeper
T12	Dribble
T13	Catch the ball
T14	Stop moving
T15	Stand idle
T16	Mark opponent
T17	Receive ball
T18	Throw ball
T19	Indirect pass
T20	Throw to teammate
T21	Kick the ball out

2.3 Decision Tree Structure

AND/OR graph structure is used to represent the decision trees [3], [10]. The leaf nodes represent the attributes. A task is represented as AND, OR & NOT combinations of these attributes in the next higher levels of nodes. For e.g. the defender passes the ball to the forwarder only when the following condition is satisfied.

Pass to forwarder <= (defender NOT close by) AND
(Ball in kicking range) AND
(Opponent close by) AND
(Unmarked forwarder in passing distance)

The decision trees for defender, forwarder and goalkeeper have been formulated and represented in the predicate format in table 3. The decision tree of defender is shown in figure 3.

The decision tree is processed when there is a change in the world. The tasks enabled in the decision trees are the ones that the agent will consider to perform. The agent then selects the task to actually execute using the procedure *Select_Task* ( ).

In the decision tree a task node T_i where $1 \leq i \leq 21$ is said to be *solvable* if the boolean values of the attributes of the leaf nodes associated with it makes the task's boolean value true.

Table 3. Decision trees' representation in predicate format

GOALKEEPER	DEFENDER	FORWARDER
$T_1 = \neg A_1$ $T_2 = A_1 \wedge ((A_{17} \vee A_5 \vee (\neg A_{13})) \vee ((\neg A_2) \wedge A_{12} \wedge A_{11}) \vee (A_{12} \wedge A_{17} \wedge A_4))$ $T_4 = (A_{26} \vee A_{32}) \wedge A_4 \wedge (\neg A_{13})$ $T_5 = (A_4 \vee A_5) \wedge A_{19} \wedge (\neg A_{18})$ $T_8 = (A_{34} \vee A_{35} \vee A_{51} \vee A_{22}) \wedge A_{42} \wedge A_2 \wedge (A_4 \vee (\neg A_{52}))$ $T_9 = (\neg A_4) \wedge A_{37} \wedge A_7 \wedge A_2$ $T_{10} = A_{38} \wedge A_2$ $T_{12} = (\neg A_{16}) \wedge (\neg A_{36}) \wedge A_{27} \wedge A_{17}$ $T_{13} = A_3 \wedge A_{10}$ $T_{14} = A_{40}$ $T_{17} = (A_{10} \vee A_{46}) \wedge A_{45}$ $T_{18} = (\neg A_{47}) \wedge A_{48} \wedge (\neg A_{52})$ $T_{19} = A_{49} \wedge A_{17} \wedge A_2$ $T_{20} = A_{47} \wedge A_{48} \wedge (\neg A_{52})$	$T_1 = \neg A_1$ $T_2 = A_1 \wedge ((A_4 \vee A_7 \vee A_8 \vee A_5) \vee (\neg A_2) \wedge ((A_{26} \wedge A_{50}) \vee A_{11})) \vee (A_{12} \wedge (\neg A_{31})))$ $T_3 = (A_7 \vee A_5 \vee A_8) \wedge A_{33} \wedge (A_{41} \vee A_{31})$ $T_4 = (A_7 \vee A_8 \vee A_5) \wedge A_{36} \wedge (\neg A_{31}) \wedge (A_{32} \vee A_{26})$ $T_5 = A_{29} \wedge A_{19} \wedge A_5$ $T_6 = A_{30} \wedge A_5 \wedge (\neg A_8)$ $T_7 = (A_{34} \vee A_{35} \vee A_{22}) \wedge A_{17} \wedge A_{30} \wedge A_2$ $T_8 = (A_{34} \vee A_{35} \vee A_{51} \vee A_{22}) \wedge A_{36} \wedge A_{28} \wedge A_{42} \wedge A_2 \wedge (A_4 \vee (\neg A_{52}))$ $T_9 = (A_7 \vee A_5) \wedge A_{37} \wedge A_{17} \wedge A_2$ $T_{10} = (\neg A_{13}) \wedge A_{38} \wedge A_{17} \wedge (\neg A_{52}) \wedge A_2$ $T_{11} = (\neg A_{13}) \wedge A_{15} \wedge (\neg A_{14}) \wedge (\neg A_{52}) \wedge A_2 \wedge A_4$ $T_{12} = A_2 \wedge A_{17} \wedge (\neg A_{16}) \wedge (\neg A_{36})$ $T_{14} = A_{40}$ $T_{16} = (A_7 \vee A_8 \vee A_5) \wedge A_{17} \wedge (\neg A_{39})$ $T_{17} = (A_{10} \vee A_{46}) \wedge A_{45}$ $T_{18} = (\neg A_{47}) \wedge A_{48} \wedge (\neg A_{52})$ $T_{19} = A_{49} \wedge A_{17} \wedge A_2$ $T_{20} = A_{47} \wedge A_{48} \wedge (\neg A_{52})$ $T_{21} = (A_4 \vee A_8 \vee A_5) \wedge A_{32} \wedge (\neg A_{42}) \wedge A_2$	$T_1 = \neg A_1$ $T_2 = A_1 \wedge ((A_9 \vee A_6 \vee A_8) \vee ((\neg A_2) \wedge ((A_{26} \wedge A_{50}) \vee A_{11})) \vee (A_{12} \wedge (\neg A_{31})))$ $T_3 = (A_9 \vee A_6 \vee A_8) \wedge A_{33} \wedge (A_{41} \vee A_{31})$ $T_4 = (A_6 \vee A_8 \vee A_9) \wedge A_{36} \wedge (\neg A_{31}) \wedge (A_{32} \vee A_{26})$ $T_5 = A_{29} \wedge (\neg A_{23}) \wedge (\neg A_{24}) \wedge A_5$ $T_6 = (A_9 \vee A_6 \vee A_8) \wedge A_{30} \wedge (A_{26} \wedge A_{49}) \vee (A_{25} \wedge A_2)$ $T_7 = (A_{11} \vee A_{30}) \wedge (\neg A_{52}) \wedge (\neg A_{20}) \wedge (\neg A_{21}) \wedge A_2$ $T_8 = (A_{34} \vee A_{35} \vee A_{51} \vee A_{22}) \wedge A_{36} \wedge A_{28} \wedge A_{42} \wedge A_2 \wedge (A_4 \vee (\neg A_{52}))$ $T_9 = (A_7 \vee A_5) \wedge A_{37} \wedge A_{17} \wedge A_2 \wedge (\neg A_{52}) \wedge (\neg A_{38})$ $T_{10} = A_{38} \wedge A_{17} \wedge A_2$ $T_{11} = (\neg A_{38}) \wedge A_{15} \wedge A_{17} \wedge (\neg A_{37}) \wedge A_2 \wedge (A_4 \vee A_5)$ $T_{12} = A_2 \wedge A_{17} \wedge (\neg A_{16}) \wedge (\neg A_{36})$ $T_{14} = A_{40}$ $T_{16} = (A_9 \vee A_8 \vee A_6) \wedge A_{17} \wedge (\neg A_{39})$ $T_{17} = (A_{10} \vee A_{46}) \wedge A_{45}$ $T_{18} = (\neg A_{47}) \wedge A_{48} \wedge (\neg A_{52})$ $T_{19} = A_{49} \wedge A_{17} \wedge A_2$ $T_{20} = A_{47} \wedge A_{48} \wedge (\neg A_{52})$

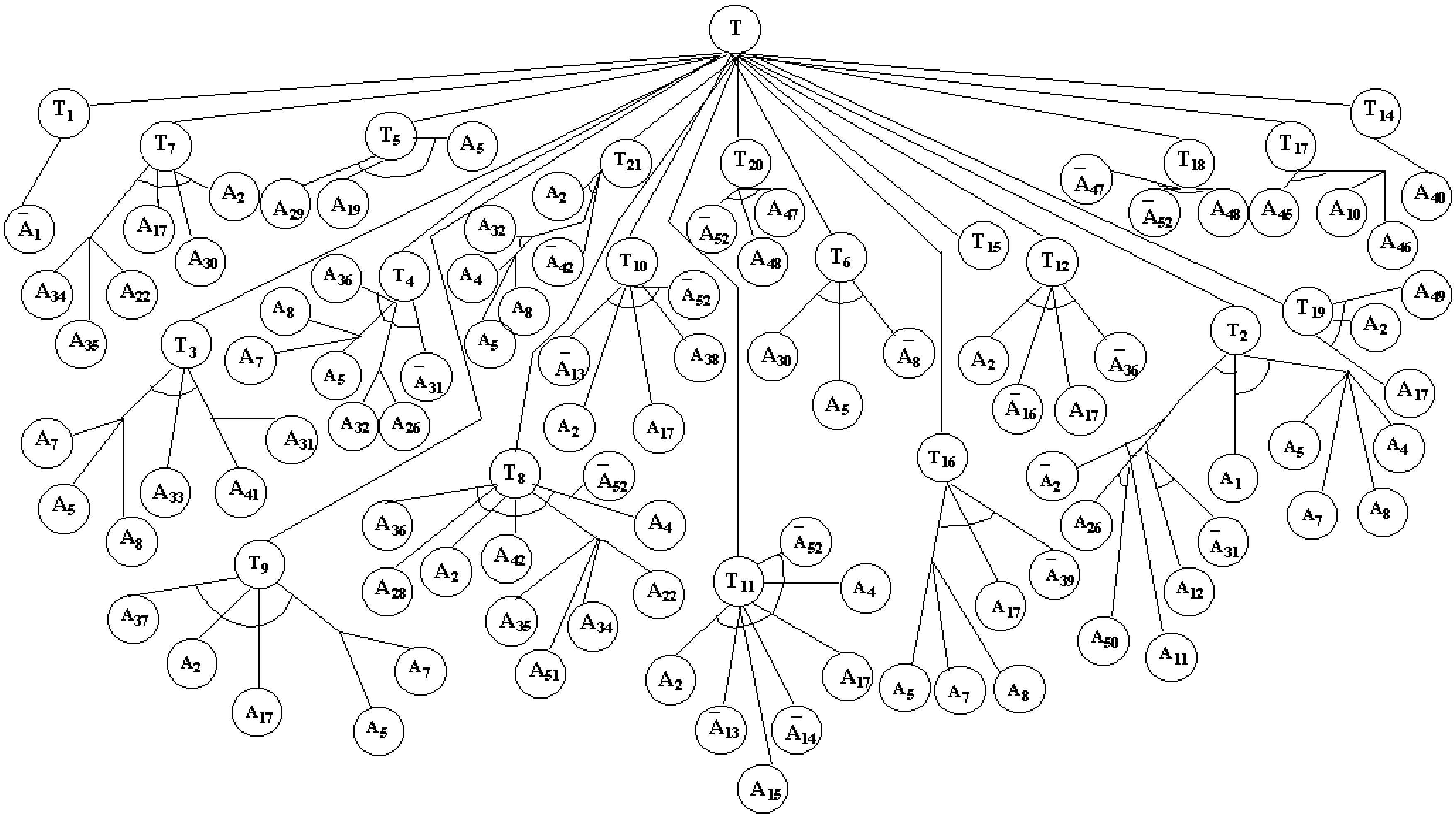

Fig.3. Decision tree of defender

Table 4. Heuristic values based on the attributes' importance in various tasks. The heuristic value of attribute A_i of task T at position (i,j) is H_j where $1 \leq i \leq 52$ and $1 \leq j \leq 10$.

	$H_1=1$	$H_2=2$	$H_3=3$	$H_4=4$	$H_5=5$	$H_6=6$	$H_7=7$	$H_8=8$	$H_9=9$	$H_{10}=10$
A_1	T_1	T_2								
A_2	T_7	T_8	T_9	T_{10}	T_{11}	T_{19}	T_{12}	T_2	T_6	T_{21}
A_3	T_{13}									
A_4	T_8	T_9	T_{11}	T_2	T_5	T_4	T_{21}			
A_5	T_9	T_{11}	T_2	T_6	T_5	T_3	T_4	T_{21}	T_{16}	
A_6	T_2	T_6	T_3	T_4	T_{16}					
A_7	T_9	T_2	T_3	T_4	T_{16}					
A_8	T_2	T_6	T_3	T_4	T_{21}	T_{16}				
A_9	T_2	T_6	T_3	T_4	T_{16}					
A_{10}	T_{13}	T_{17}								
A_{11}	T_7	T_2								
A_{12}	T_2									
A_{13}	T_{10}	T_{11}	T_2	T_4						
A_{14}	T_{11}									
A_{15}	T_{11}									
A_{16}	T_{12}									
A_{17}	T_7	T_9	T_{10}	T_{11}	T_{19}	T_{12}	T_2	T_{16}		
A_{18}	T_5									
A_{19}	T_5									
A_{20}	T_7									
A_{21}	T_7									
A_{22}	T_7	T_8								
A_{23}	T_5									
A_{24}	T_5									
A_{25}	T_6									
A_{26}	T_2	T_6	T_4							
A_{27}	T_{12}									
A_{28}	T_8									
A_{29}	T_5									
A_{30}	T_7	T_6								
A_{31}	T_2	T_3	T_4							
A_{32}	T_4	T_{21}								
A_{33}	T_3									
A_{34}	T_7	T_8								
A_{35}	T_7	T_8								
A_{36}	T_8	T_{12}								
A_{37}	T_9	T_{11}								
A_{38}	T_9	T_{10}	T_{11}							
A_{39}	T_{16}									
A_{40}	T_{14}									
A_{41}	T_3									
A_{42}	T_8	T_{21}								
A_{45}	T_{17}									
A_{46}	T_{17}									
A_{47}	T_{20}	T_{18}								
A_{48}	T_{20}	T_{18}								
A_{49}	T_{19}	T_6								
A_{50}	T_2									
A_{51}	T_8									
A_{52}	T_7	T_8	T_9	T_{10}	T_{11}	T_{20}	T_{18}			

```
Procedure Select_Task ( )
Begin
   1. Given a scenario the attributes are assigned boolean values
      namely True and False.
   2. The boolean values of the attributes are propagated up from the
      leaf nodes to the task nodes in the decision trees of defender,
      forwarder or goalkeeper based on the role of the agent.
   3. Based on the values propagated, one or more task nodes become
      solvable.
   4. If only one task is solvable then the agent is assigned that
      task as the final decision.
   5. If more than one task is solvable then heuristic values (as
      given in the heuristic table 4) are assigned to attributes and
      propagated to the associated activated task using procedure
      Ranking (Task).
End.
```

```
Procedure Ranking(Task min_task)
Begin
      //Imp is the total contribution of the AND nodes to the task
      //nodes
      Constant Imp //initialize Imp to a value >=10
      Int min = α //minimum task value is initially assumed to be α
      For all tasks T solvable do
      Begin
            //calculating each AND nodes contribution to the task
            //node
            Int Every_node_imp = Imp / No_of_and_nodes in T
            Calculate Task_value as follows
                  *For an AND node the attributes' values are added
                  before propagating it to the parent node.
                  *For an OR node the lowest value is propagated up
                  to the parent node.
                  * finally Task_value+= Every_node_imp
            if min< Task_value then min_task = T
      End
      Return min_task
End.
```

3. Illustration

Figure 4 depicts a scenario in which the ball is with team B. The defenders of team A are trying to get the ball to pass it to the forwarders who are waiting for the ball. The players of team A are marked as F_i, D_i and G and the players of team B are marked as OF_i, OD_i and OG where $1 \leq i \leq 5$.

The attributes listed in the previous sections are first checked for their truth-values. The truth-values of the attributes for each player are set . Using these values and the heuristic values of the attributes, the decision trees are used to choose the most appropriate task for each player.

TEAM A (shaded black). As the goalkeeper is already positioned at the center of the goalpost he needs to wait for the ball to come closer to the goalpost. Hence the goalkeeper watches out for the ball to come closer for him to defend the goalpost (T_{15}).

As for the defenders, the defender D3 nearest to the ball moves towards the ball (T_2) while the rest of the defenders try to mark their close opponent (T_{16}), in order to intercept the ball if it is passed to that opponent. For defender D3 both T_2 and T_{16} are enabled. After the application of heuristic values to the leaf nodes and the usage of the procedure *Ranking* () the values of tasks T_2 and T_{16} are 14.5 and 17.3 respectively (assuming Imp to be 10). Hence T_2 is assigned to D3.

As for the forwarders, all of them wait for the ball to be passed to their side (T_{15}).

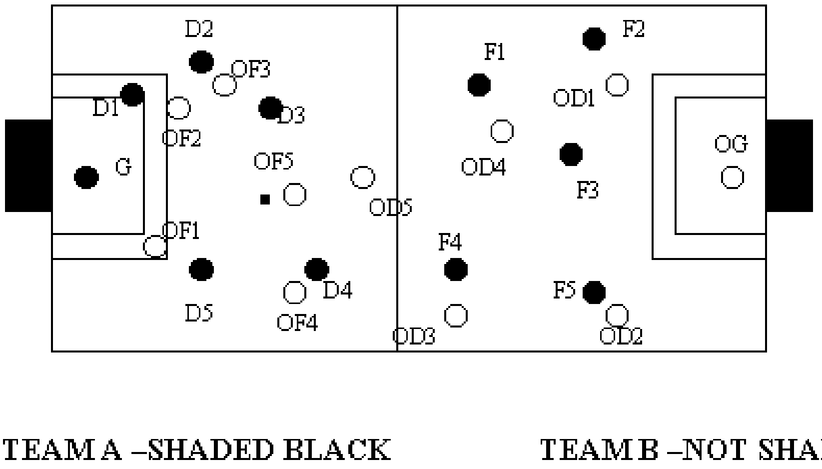

Fig.4. A scenario with team B in *Attack* behavior and Team A in *Defense* behavior

TEAM B (not shaded). The goalkeeper of TEAM B stands idle (T_{15}), as the ball is not in his area of defense.

The defenders are keeping track of the opponent forwarder closest to each one of them (T_{16}) except OD5, which is trying to bring the ball within its view range (T_1).

The forwarders OF2, OF3, OF4 keep track of the opponent near them (T_{16}) so that the opponent does not intercept the ball. The forwarder OF5 dribbles the ball as all his teammates are being marked and is waiting for an opportunity to either kick the ball to the goalpost or pass the ball to one of its teammates. The forwarder OF1 can either move towards his teammate T_3 (OF5) or towards the ball T_2. As per the heuristic priority obtained for the two tasks moving towards the ball with value 8 is more advantageous than moving towards the teammate with a value 9.3. So the player opts to move towards the ball.

4. Conclusion and Future Enhancements

In this paper two behaviors of soccer agents have been considered namely attack and defense. The agent's perception of the world has been represented as attributes of the dynamic world. The tasks that an agent has to perform in a given scenario depend on the boolean values taken by these attributes. The tasks have been formulated as decision trees based on the roles that an agent could assume namely defender, forwarder and goalkeeper. Coordinated behaviors are also encoded in the decision trees of individual agent. Based upon the attributes' boolean values and heuristic values applied on to the leaf nodes of the decision trees, an ultimate task is invoked for an agent as the final decision.

Though effort has been made to identify all necessary tasks and attributes, more tasks and predicates that are specific and those that represent the entirety of the world can be added as need be and on experience. Taking decisions based on reinforcement learning [5] can enhance the proposed decision mechanism. Usage of parallel algorithms for decision-making will further improve the performance of the proposed method

References

1. Peter Atone, *RoboCup2000 - The fourth robotic soccer world championships*. Available at http://citeseer.nj.nec.com/stone00robocup.html
2. Silvia Coradeschi, Lars Karlsson, *A Decision-Mechanism for Reactive and Cooperating Soccer-Playing Agents*, Department of Computer and Information Science, Link.oping University, Link.oping, Sweden. Available at http://www.ep.liu.se/ea/cis/1997/001/

3. Elaine Rich, Kevin Knight, *Artificial Intelligence*, TATA McGraw – Hill Publications, Edition 1991.
4. Walter Brenner, Rudiger Zarnekow, Hardmut Wittig, I*ntelligent Software Agents*, Springer – Berlag Publications, Edition 1998.
5. Reinforcement learning repository. Available at www.cse.msu.edu
6. Coradeschi, A., Karlsson, L. and T.orne, A. 1996 Intelligent Agents for Aircraft Combat Simulation, in Proceedings of the Sixth Conference Coradeschi, A., Karlsson, L. and T.orne, A. 1996 *Intelligent Agents for Aircraft Combat Simulation*, in Proceedings of the Sixth Conference on Computer Generated Forces and Behavioral Representation, Orlando, FL. Available at http://www.ida.liu.se/ silco.
7. Fenster, M., Kraus, A. and Rosenschein, J. A. 1995 *Coordination without Communication: Experimental Validation of Focal Point and Techniques*, in Proceedings of the First International Conference on Multi-Agent Systems (ICMAA-95), San Francisco, CA.
8. T. Scmitt, M. Klupsch, R. Hanek, and T. Bandlow. Agilo RoboCuppers: RoboCup team description.In M. Veloso, E. Pagello, and H. Kitano, editors, RoboCup-99: Robot Soccer World Cup III. Springer
9. Verlag, Berlin, 2000.Ahehory, O. and Kraus, A. 1996 *Cooperative goal-satisfaction without communication in large-scale agent-systems*, in Proceedings of ECAI96, Budapest, Hungary.
10. Tambe, M. 1996 *Tracking Dynamic Team Activity*, in Proceedings of AAAI96, Portland, Oregon.
11. Information about RoboCup available at http://www.robocup.org/
12. Jan Murray, *Soccer agents think in UML*, Master’s thesis, Universitat Koblenz-Landau, Facbereieh Informatik, 2001. Available at www.uni-koblenz.de