

AUT Parsian Team Description Paper 2014

Mohammadhossein Malmir, Shahin Boluki,
Majid Kamali, Mohammad Bababeig, and Mojtaba Yeganejou

Amirkabir University of Technology (Tehran Polytechnic), Hafez Ave., Tehran, Iran
{mhmalmir, sh.boluki, majidkamali1370,
mohammad.bababeik, yeganejou.tux}@gmail.com

Abstract. AUT Parsian team is started as a 2D soccer simulation team with the aim of preparing a platform for applying machine learning techniques and advanced artificial intelligence tools. In this paper we introduce some of our efforts and researches and some of the algorithms that are newly implemented in our team. We introduce our new strategy based offensive decision making system and modifications to our previous task evaluator that is application of Fuzzy in the scoring method of the decisions. We specify the changes made in our passing skill in order to reduce the computational complexity. We also consider the determination of positions, neck angles and view settings of our agents in the field as a coverage problem and mention our current efforts on solving it.

1 Introduction

AUT Parsian team is started as a 2D soccer simulation team with the aim of preparing a platform for applying machine learning techniques and advanced artificial intelligence tools. The AUT team was established in 2011. Our goal is to use 2D soccer simulation as a multi-agent environment to develop our team in such a way that advanced Artificial Intelligence (AI) and machine learning tools have the main role in developing our team performance. Our team is based on agent 2D base (release 3.1). We have participated in RoboCup2012 in Mexico City and RoboCup2013 in Eindhoven under the name of AUT-2D and AUT-MasterMinds respectively. We have achieved the 6th place in 2012 and 7th place in 2013. Our team's name is changed due to the fact that the AUT Soccer Simulation Team and Parsian Small Size Soccer Team have started a joint collaboration, and we have implemented ideas from 2D Soccer Simulation in the Small Size Soccer League.

In this paper we introduce some of our team's new algorithms. In section 2 we describe our with-ball decision making consisting of our new strategy based offensive decision making and the visual software to define these strategies namely *sedit* which is our modified version of *fedit2* released by Helios Soccer Simulation Team. We also describe our new Fuzzy [1, 2] scoring method in our task evaluator. Section 3 mentions the changes in our passing skill which are mainly on the generation and calculation of potential target points for passing that reduces the computational complexity and run time of our passing algorithm. In section 4 we

consider the determination of the positions, neck angles and view settings of our agents as a coverage problem to extract the best and most update information of the field as a team and deliver it to the agent which has the most important role in current situation, e.g. the ball owner agent in offensive situations. This method should be somehow compatible with the other algorithms implemented for the positioning and neck control actions. For this, we have strict constraints on some agents' positions and more freedom for some other agents. Section 5 gives a summary of our discussion and future work.

2 With-ball Decision Making

With-ball decision making is the decision which the agent that owns the ball should make to whether shoot the ball to opponent's goal, pass the ball to other agent (including deciding to pass to which agent), or dribble with ball or simply hold the ball. In with-ball decision making we apply both our modified previous version of task evaluator [3], and the new strategy based decision making. The strategies are determined beforehand and off-line. We define our strategies as sequences of actions by a human intuition operation as an expert, with the graphical environment of our *sedit* software. These strategies can be learned from log files as the last sequence of actions that led to the goal by a team, too [4]. To be more precise, the task evaluator makes decision of current action to be done based only on current situation and do not consider the next steps. But strategy based decision making makes the decision of current action to be done as a member of a sequence of decisions and actions leading to the full implementation of the strategy sequence. So, selecting and performing each strategy requires some initial conditions in the field and game situation to hold true, and also a specific prediction of field to hold true with some approximation. In the situations that no strategy is defined or the conditions of no strategy holds true, we only use the output of our task evaluator as our decision making output. But when we have output decision from both algorithms we choose one of these decisions based on some features like how much the predicted game and field condition is likely to the predefined situation and condition of the selected strategy. In subsection 2.1 we discuss the Fuzzy modification of our task evaluator and in section 2.2 we elaborate on the new strategy based decision making.

2.1 Fuzzy Scoring Method

As mentioned in [3, 5] in task evaluator, first the scoring chance by a kick is checked. Then our pass and dribble is virtually executed and the best output of each of them is collected in an array. Then these outputs are scored and the one which gets the bigger score is chosen. In fact the score is calculated considering the role and position of the ball owner agent, the condition of the field and whether the situation is now defensive or offensive. The new modification is as follows. In previous version the field was divided into regions with specific scoring methods and the score calculation was only based on the region which

the position of the agent which owns the ball laid on. But our current approach is that the field is divided into fuzzy regions such that each position on the field has a membership function of being in the defined regions, so we have a more reliable score calculation method with more smooth changes.

2.2 Strategy Based Decision Making

In [6–8] Helios team has introduced an on-line cooperative behavior planning using a tree search method. In our strategy based decision making we have a different approach in which we generate a strategy off-line with human intuition with a graphical interface or with learning sequences from log files. In fact each strategy is a sequence of decisions and actions which leads to a specific objective which is usually scoring a goal. Each strategy is based on specific conditions on field situation such that when the conditions of a strategy do not exist, implementation of that strategy is not considered. But when the condition holds true, each agent considers its decision and action as a member of a sequence of decision and actions. So when making with-ball decisions we consider these predefined action sequences. In other words when the current and the predicted situation and condition of field is similar to the predefined j^{th} strategy with some approximations, the agent considers its decision as a member of that strategy sequence. Another interesting aspect is that the performing of the predefined sequence should not be considered only from the beginning. In other words, if an agent finds the current and predicted future situation proper for performing a sequence of actions from the i^{th} action step and decision of the j^{th} strategy (sequence) till the last step of that strategy, but the previous decisions were not made based on the sequence, it will start the sequence from the i^{th} step.

We have developed *sedit* based on the released *fedit2* by Helios Soccer Team. In *sedit* the action sequence and the areas which the predefined conditions should be checked can be defined in a graphical manner. In the current version one can select from the following actions; pass, shoot, dribble and move. The direction and conditions can be defined, too. The area which the conditions of implementation of a strategy should hold true can be defined in circular areas. After saving the sequence it is written in a file as *tactics-data*. All the strategy files are saved and can be accessed in the code like a formation file. A picture from the *sedit* interface is shown in Fig. 1. Note that the picture does not show a sequence and strategy and only shows discrete actions and decisions. It aims to depict a general view of the program.

3 Tree Search Based Pass Target Generation

As mentioned in [3] our team’s passing skill consists of two steps. First, the generation of potential pass target points, and second, giving a score to each target point based on different factors mentioned in [3] such as the safety of that pass. Since the calculations of scores and safety of the potential pass targets can be too much, the target point generation is a crucial task due to the fact

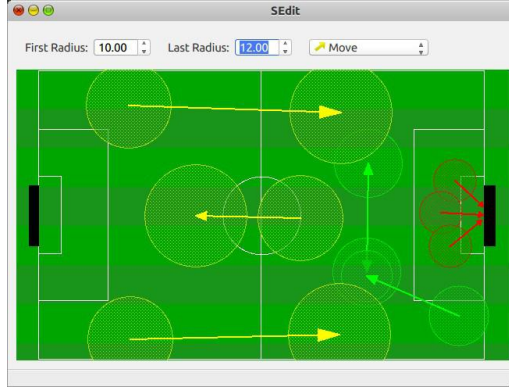


Fig. 1. General View of *sedit*.

that we do not want to miss a potential good target point; nevertheless we need low computational complexity that does not lead to a system breakdown. For this we use a tree search based pass target generation to lower the amount of computations with the least chance of missing a potential good target position by employing recursive algorithms. The algorithm is as follows. At the initialization step, the ball owner agent chooses the potential suitable pass receivers by some initial simple condition checking. Then for each potential receiver agent the following algorithm is employed.

First a sector encompassing the receiver with the passer agent as its center is generated with angles and lengths specified with respect to the relative distance and relative angle of the passer and receiver agent. Then, the recursive procedure is as described in the following. Each sector is divided into smaller parts and each part is scored. Then, the smaller parts are sorted regarding their scores and a specified number of the parts with the highest calculated scores are selected as the input for the next iteration. The sample result of three iterations of the algorithm is depicted in Fig. 2.

With proper tuning of the algorithm's parameters including the angle and size of the initial sector, the number of smaller parts that the sector is divided to, the way the sector is divided into smaller parts in each iteration, the number of smaller parts gathered as the input to the next iteration and finally the number of iterations, one can achieve the mentioned goal that is lowering the chance of missing a potential good target point with the least computations. In this algorithm in addition to lowering the total number of potential target points generated, the computations and processing in the first iterations is lowered, too.

4 The Coverage Problem

In the last years, we had well-developed decision algorithms in each part of our team like shoot, pass, dribble or without-ball positioning, however, we didn't

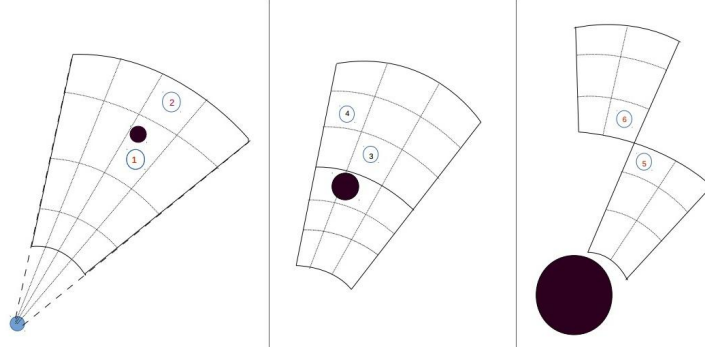


Fig. 2. Three iterations of the algorithm. In each iteration the numbered regions are selected as the input to the next iteration.

get the best result from them because of the lack of confidence in the agent's perception of the environment due to their unsuccessful view area achieved by a partly random turn necks. So, we come to this idea that we should employ an algorithm that gives agents a better perception of the soccer field environment i.e. teammates, opponents and the ball.

In this approach, we considered that the agent who must have the best cover of the field is our nearest agent to the ball, which in offensive situations it is the one that decide what our team should do and in defensive situations it is the most effective agent of us to destroy the opponents play. So, all the other agents must help this agent by somehow covering all the important areas in a cooperated way and send this information to the agent by the say ability. In [9] the author has divided the sensor coverage problem to three categories, based on the coverage type, namely, point, area and barrier coverage problems. To solve this problem in a much easier way we assumed to cover some dynamic points, instead of an area. These points of course are the points that are important for the effective agent to take an action. In other words, these important points are the positions of the ball, opponents and teammates. For optimality of the algorithm, each of these maximum 22 points must be chosen by one of our agents, and the lower number of agents of us used to do this coverage in each cycle the better result we will get because of the lower possibility of say conflicts and increased speed of transferring information between agents to the final agent (the nearest to ball). In order to fulfill this need, we find the budgeted maximum coverage problem a helpful and similar problem.

4.1 The Budgeted Maximum Coverage Problem

The budgeted maximum coverage problem is: given a collection S of sets with associated costs defined over a domain of weighted elements, and a budget L , find a subset of $S' \subseteq S$ such that the total cost of sets in S' does not exceed L , and the total weight of elements covered by S' is maximized [10]. This problem is NP-

hard. In [10] the authors have proposed a $(1-1/e)$ approximation algorithm. We use this problem idea and their proposed algorithm to solve our sensor coverage problem.

Due to the fact that each agent must pay attention to one other to hear its said strings, we developed a tree-like communication networks between the agents in the way that each upper agent in the tree hear the said words of its below nodes, combine them and send it to the upper node, till the goal agent that is the top node of the tree. We define the length of the say chain (number of branches) from the agent that covers some points to the goal agent as a cost of the set that this agent covers and the importance of the covered position by that agent for the goal agent as the weight of the point. Besides choosing which agents are considered for this goal we have some more inputs like some agents' positions, the neck angles and view mode of the agents that we can change them to reach our need. By applying the above algorithm we reach to an optimal and good global vision with better confidence for the goal agent (nearest to ball).

5 Summary

In this paper we introduced some of our novel algorithms. We introduced our new with ball decision making system in section 2. We also described our new pass target point generation and calculation in section 3. Finally we proposed our coverage problem in section 4. Our efforts are currently focused on improving the strategy based decision making and *sedit* interface described in section 2.2, and solving the proposed coverage problem in section 4.

References

1. Ross, T. J.: Fuzzy Logic with Engineering Applications (2nd ed.). John Wiley & Sons (2004)
2. Wang, L.X.: A Course in Fuzzy Systems and Control (1st ed.). Prentice-Hall (1997)
3. Malmir, M., Boluki, S., Simchi, M., Hediehloo, H.: AUT-2D Team Description Paper 2012, RoboCup 2012.
4. Carbonell, J.G., Michalski, R.S., Mitchell, T.M.: Machine Learning: An Artificial Intelligence Approach, volume II. Morgan Kaufmann Publishers, Los Altos, CA, Vol. 2, pp.167-192. (1986)
5. Malmir, M., Boluki, S., Simchi, M.: AUT-MasterMinds Team Description Paper 2013, RoboCup 2013.
6. Akiyama, H., Nakashima, T., Aramaki, S.: Online cooperative behavior planning using a tree search method in the robocup soccer simulation. In: Proceedings of 4th IEEE International Conference on Intelligent Networking and Collaborative Systems (INCoS-2012). (2012)
7. Akiyama, H., Nakashima, T., Yamashita, K.: Helios 2013 Team Description Paper, RoboCup 2013.
8. Akiyama, H., Nakashima, T.: Helios 2012 Team Description Paper, RoboCup 2012.
9. Wang, B.: Coverage control in sensor networks. Springer (2010)
10. Khuller, S., Moss, A., Naor, J.: The budgeted maximum coverage problem. Information Processing Letters. 70, 39-45 (1999)