WrightEagle2008 Simulation 2D Team Description Paper

Ke Shi, Tengfei Liu, Aijun Bai, Wenkui Wang, Changjie Fan, Xiaoping Chen

Multi-Agent Systems Lab.,
Department of Science and Technology,
University of Science and Technology of China,
Hefei, Anhui Province, P.R. China
{shike15, liutf, baj, wangwk6, cjfan}@mail.ustc.edu.cn, xpchen@ustc.edu.cn
http://wrighteagle.org

Abstract. In WrightEagle2008, we continue to research based on the previous WrightEagle simulation 2D team. WrightEagle has won the runner-up of Robocup2007, the Champion of RoboCup2006, and the runner-up of RoboCup2005, as well as three consecutive championships of RoboCup China Open in recent years. In this paper, we present the innovations of our team since the last simulation league competitions, and related previous work that developed by other simulated RoboCup teams and ourself.

1 Introduction

The WrightEagle RoboCup Team was established in 1998, starting with only the simulation 2D team. In the following years, the 4-legged team, the simulation 3D team and the MSRS team have joined in WrightEagle. We have participated in annual competition of RoboCup from 1999. Last year, we have got two champions(Simulation 3D and MSRS), one runner-up(Simulation 2D) and a 4th place(4-legged) in RoboCup 2007, Atlanta.

We take simulation 2D as a typical problem of multi-agent systems, and our long-term goal is to do research in decision-theoretic planning and other challenging project in artificial intelligence.[1] We have explored how to solve MDPs using real-time dynamic programming.[2] Besides, we have studied decentralized POMDPs and apply the method we proposed in the defense of simulation 2D.[3]

In this paper, we present a brief description of the implementation principles of our new team. We focus not only in the high-level models for decision-making, but also implementation details in the low level. In high-level decision-making models, we try to adopt an anytime structure in our team in order to apply the POMDP techniques better. In low-level implementation details, we improved our visual model as well as our coach model. Besides, we develop a debugger to increase the efficiency of our development. Improvements of our new team are based on the implementation of the one we built last year. Thanks to the information of the CMUnited-98, FCPortugal2000 and Brainstormers2007.[4,5,6]

2 Anytime Structure

In our previous team, the program of an agent consists of two threads: parse thread and decision thread. In parse thread, the agent estimates the precise arrival time of sight message to decide whether to awake the decision part or continue to wait, so the remaining decision time of an agent in each cycle is uncertain. If the remaining time is short, the agent may be still in the process of decision-making at the end of the cycle, so he will do nothing in the current cycle. On the other hand, if the remaining time is quite long, the agent may waste much time after making the decision and sending the command. This two cases are very negative to the real-time dynamic environment of simulation 2D.

We applied an anytime structure in our new team version. First of all, we add a new thread for sending commands to the server, based on the old structure. This thread can keep synchronized with SoccerServer and make sure to send the optimal command of that time before the end of each cycle. This can avoid the first case mentioned above. Then, in the decision thread, we designed a new structure which was similar to a breadth-first search algorithm. The agent begins to make decisions at the beginning of each cycle. When the sight message arrives, he continues to make decisions after updating the world model until the end of the cycle. In this case, the agent can make full use of the time of each cycle. This method solved the problem as mentioned in the second case.

The new structure is shown in Figure 1 and Figure 2.

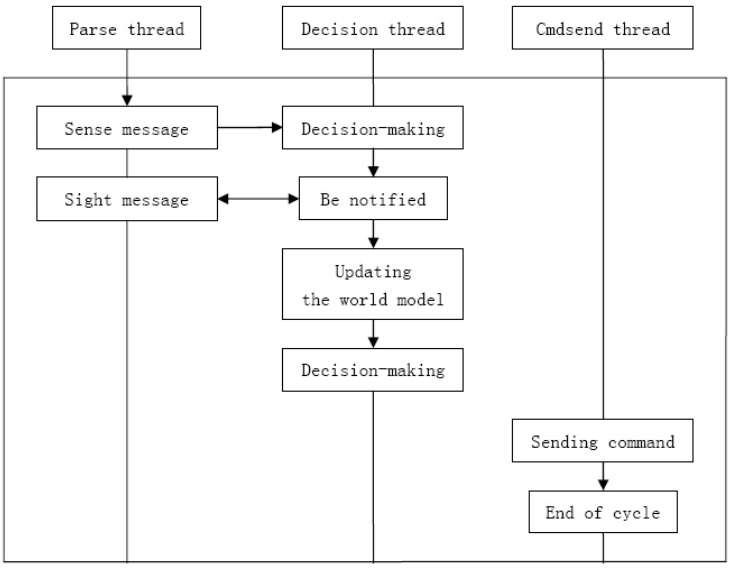

Fig. 1. The model of three threads

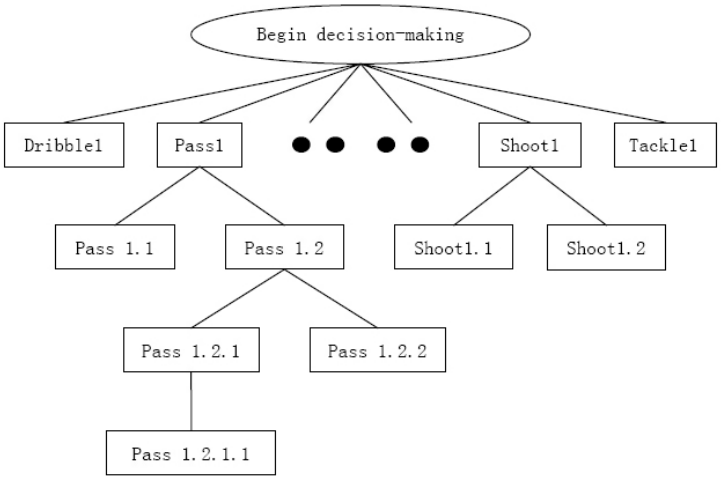

Fig. 2. Anytime structure in decision thread

3 Decision Evaluation Mechanism

Our long-term goal is regarding the whole decision process based on POMDPs, in which the probabilities of observation are transformed to the probabilities of state-transition of the next stage. In our new team version, the model is simplified to an MDP model with some additional restriction to take into account of the observation uncertainty.[2,3] In addition, the application of the anytime structure is to apply the MDP model better in our future team.

States S: In each state the ball is controlled by a player. When the ball is free, the agent directly predict the possible states of the ball under control, based on the interception model which is presently quite mature.
Action A: The atom actions prescribed by the server are not used. Instead, a behavior module is introduced with domain knowledge. Examples of actions that we defined are Pass, Dribble, Shoot and etc. Each action has its own formal model to compute the possible state and the probability of state transition.
Reward R: The basic idea is to discriminate several main situations. Different situations are appointing to different goals. The highest reward is achieved when the goal is reached; otherwise a small reward will be given according to state point sensitivity. A negative reward is given when fail to reach the goal.

A basic decision process is as follows. The agent analyzes the situation, confirm the current state and the goal state. The action generator can generate many

optional actions using the anytime structure mentioned above. When the cmd-send thread must send the command to SoccerServer, the decision thread will provide the most valuable action command at that time which is chosen from all the optional actions.

4 New Visual Decision Method

Since a new synchronous timer, which we prefer to use in our team, is used in rcssserver-12.0.0, the old visual decision method which is simply known as narrow-narrow-normal schema may not perform very well. Taking into consideration of improving the accuracy and timeliness of visual information, we developed a new visual decision method based on object-evaluation, here "object" may refer to ball player and so on. In our new method, every object will get a score according to its importance, historical information confidence and high-level decision-making feed back. Then visual decisions like turning neck or turning body are made to get the maximum score. We also considered the advanced visual decision. For example, the agent will intentionally take some advanced actions, like observing his particular teammate if he can pass the ball, which the high-level decision-making system may fail to reach. With this method we can improve the accuracy of some actions like passing, shooting etc.

5 The Coach

In the new server 12.0.0, the type of heterogeneous players has increased from 7 to 18, and some other related parameters have slightly changed. It's a good chance for the coach to show a better performance , that means choosing the best type for the specific player in the playfield.

We design a new heterogeneous player selecting model for the coach. In the new one, we take into account all the parameters about a player. To evaluate the impact of these parameters, we use a two-layer evaluating method. In the first layer, we set several scenes for specific role, then evaluate each player based on its performance. For example, a forward on the football field usually runs a long distance to launch a quick attack. So one of the scenes we set for forward player is letting him run as fast as he can with (or without) a ball for a long distance. If one heterogeneous player behaves very well in this scene, it will be considered as a potentially good forward and given a high evaluation value. In this scene testing layer, several parameters have been taken into account, but not all of them. In the second layer, we consider some individual parameters. A lot of experiments are done to determine the impact factor of specific parameter. Then we can slightly and cautiously amend the previous evaluation in order to make the evaluation more precisely. According to the evaluation we obtained, the coach can choose the best type for the player.

6 Offline Statistics Model

In the decision-making part, the possibility of the ball to arrive a point is a very important value. It has a great impact on predicting the real effects of some actions. In previous method, these values are specified based on human's experience, so it may not coincide with the real situation. In the competition, these values always vary online according to the style and aggressiveness of the opponent team. In this version of our team, we develop a new offline statistics model. We let a player kick the ball in a exactly same situation for many times, then we record the ball position of each cycle after the kick. Because of the noise, the ball position is uncertain. We have analysis our record, and we found the positions after a certain time is shown as a 2-demension normal distribution. We can calculate the five parameters of the distribution and get the possibility of each point. These will benefit the online decision-making in high-lever.

The result of offline statistics is shown in Figure 3.

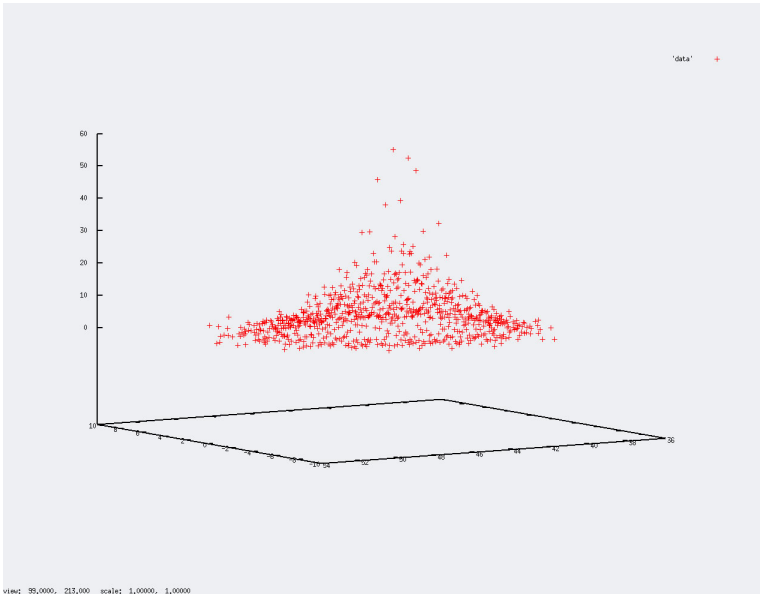

Fig. 3. The result of offline statistics

7 The Debugger

As is well known by all that a debugger plays an especially important part in developing a team, we therefore design a special one. The basic principle is

integrating the server and the agent, making them work as a whole without any network communication.

Details for the design: Change the communicating mode between agents and server. In the new debugger, a common buffer is adopted, which makes good use of the synchronous sight mode. In network mode, agents make decisions simultaneously, they send commands to the server, then server execute the command. And the sequence that sending the command by each agent is actually unimportant. Based on this fact, it's practical for all agents to make decisions one by one in a process and they all send the commands to the buffer, from which server will get all the commands and execute them finally. After completing the task, server will send the updated view message to the buffer for the agent to take. In this way, communication in network mode is well simulated.

Program which consists of only one single process or thread is easy to debug. As a result of the modification, the time consumed for network synchrony is greatly reduced. It's good for large quantity of training, because program can run more times in a time unit. Furthermore, we can debug several agents at a time in the debugger, and it's easier to call functions, to realize different kinds of special training, since server and agents are in the same process. Another thing we notice is that the same sequence of message in every circle will lead to the same decision made by agents. And the same random seed, together with the same command queue, will also lead to the same message and statement of the simulating world produced by server. Therefore it is possible to repeat one scene as many times as we want, which is great for us to test the effects of different methods in the same scene.

8 Conclusion

As is shown above, we spend a lot of time in improving the low-level implementation details and high-level decision making models. We evaluate our improvements by a lot of experiments to observe if it is efficient and useful and the result is good. All of features above enhance the strength our team. We hope more natural changes of Soccerserver take place, and we will catch up the the changes to a higher level.

References

1. Xiaoping Chen, et al, Challenges in Research on Autonomous Robots, Communications of CCF, Vol. 3, No. 12, Dec 2007.
2. Changjie Fan and Xiaoping Chen. Bounded Incremental Real-Time Dynamic Programming. IEEE Proceedings of FBIT 2007, Jeju Island, Korea, 2007.
3. Feng Wu and Xiaoping Chen. Solving Large-Scale and Sparse-Reward DEC-POMDPs with Correlation-MDPs. Proceedings of RoboCup Symposium 2007. Atlanta, America, July 2007.
4. P. Stone, P. Riley, M. Veloso: CMUnited-98 champion simulator team. AI Magazine, 2000.

5. Luis Paulo Reis, Nuno Lau: FC Portugal Team Description: RoboCup 2000 Simulation League Champion.
6. M. Riedmiller, T. Gabel: Brainstormers 2D Team Description 2007: RoboCup 2007 Simulation 2D League Champion.