WrightEagle 2D Soccer Simulation Team Description 2010

Aijun Bai, Jing Wang, Guanghui Lu, Yuhang Wang,
Haochong Zhang, Yuancong Zhu, Ke Shi, Xiaoping Chen

Multi-Agent Systems Lab.,
School of Computer Science and Technology,
University of Science and Technology of China,
Hefei, Anhui Province, China
baj@mail.ustc.edu.cn, xpchen@ustc.edu.cn

Abstract. WrightEagle 2D is a 2D soccer simulation team which has been participating in the RoboCup competition since 1999 and won 2 champion and 3 runner-ups in the past 5 years. In this paper, we briefly present our current research effort and some newly introduced techniques for improvment since the last competition.

1 Introduction

WrightEagle 2D soccer simulation team, which was established in 1998 as the first branch of WrightEagle Robocup Team developed by MAS Lab. of USTC, has been participating in annual competition of RoboCup since 1999. Recent years, we have won the Champion of RoboCup 2009, the runner-up of RoboCup 2008 and Robocup 2007, the Champion of RoboCup 2006, and the runner-up of RoboCup 2005.

We take RoboCup soccer simulation 2D as a typical problem of multi-agent systems, and our long-term goal is to do research in decision-making and other challenging projects in artificial intelligence.[1] This year, we have developed some new techniques for both the low level skills and the high level decision-making model in our new team WE2010, based on our research effort.[2][3][5] In this paper, we present a brief description of some of our progress mentioned above.

We have released our team's base source code WrightEagle_Base to the public as an open source software which can be freely accessed from our website[1]. We hope that our released software will be helpful to a new team who wants to participate in the RoboCup event and/or start a research of multi-agent systems.

2 ADMDP and its Application to the Team

In the 2D competition of soccer simulation league, one player need to dribble front with ball proximally as quickly as possible in some scene. The reason of dribbling proximally is that the player can pass the ball to the more dominate teammate at each

[1] `http://wrighteagle.org/2d/`

cycle and the reason of dribbling quickly is that it is more difficulty for the opponents to defend us. In the whole process, the player must take "kick" action at some cycles, or the ball will go out of his kickable area. In this proximal dribble problem, the player has two kinds of actions: "dash" and "kick". The player can get positive reward if taking "dash" action, while he can get zero reword if taking "kick" action. However, he can not continue his dribbling progress if not taking "kick" action. The objective of the player is to dribble as far as possible at a certain cycle.

Markov Decision Process is used for the multi-step planning problem under uncertain action results[4]. As we all know, the standard MDP model is a tuple: $\langle S, A, T, R \rangle$. S is a finite set of world state; A is a finite set of action; T is the state transition function; R is the immediate reward function. For describing the proximal dribble problem mentioned above, we propose the Action-Driven Markov Decision Process(ADMDP for short) based on the standard MDP model[5]. We define ADMDP as a tuple: $\langle M, b, p, f, h \rangle$:

- M is a standard MDP.
- b is a set of absorbing state, and the progress will not continue to other states once going into one of absorbing states.
- p is the absorbing state function, and $p(s, a)$ is the probability of taking action a at state s:
$$p(s, a) = Pr\{s_{t+1} \in b | s_t = s, a_t = a\}$$
- f is the action-driven function which is unrelated with the current state, and $f(a)$ means the immediate reward of taking action a. ($f(a) \leq 0$)
- h is a positive integer which means the finite decision step.

For an ADMDP problem, the agent will get positive reward or zero reward depending on the different actions after one step decision. Meanwhile, the agent will get negative reward if the successor states contain some absorbing states. Accordingly, the immediate reward function is defined as:

$$R(s, a) = f(a) - p(s, a)$$

If $f(a) > 0$, the action a is called "active action" which can bring a positive reward; if $f(a) = 0$, the action a is called "passive action" which can avoid the absorbing states though no positive reward.

In WE2010, we describe the proximal dribble problem with the model of ADMDP and solve it by using the value iteration algorithm[5]. The empirical result shows that the new algorithm is better than the old algorithm in the dribble successful probability, the dribble speed and the run time performence.

3 A-Star Based Motion-Planning

The dash problem, generally said the motion-planning problem, becomes more complicated than before since more dash directions were introdued in *rcssserver* 14.0. Under the consideration of effectiveness and efficiency, an a-star based heuristic search algorithm was adapted to solve thus motion-planing problem.

In our algorithm, each state in the search space represents the player's position, velocity, body direction and stamina, which can be changed according to the specific dash or turn action. A state with it's target distance less than a pre-specified buffer is a goal state. The heuristic function we used here (denoted as $f(x)$) is a sum of two fuctions:

- $g(x)$ the number of actual past cycles from the starting state,
- $h(x)$ the player's estimated cycles to a goal state with the effective maximum speed kept.

Clearly, $h(x)$ satisfies the admissible condition: $h(x) \leq \hat{h}(x)$. To reduce the time cost of searching procedure, some special pruning methods most of whith aims to reduce the need-to-search action space were used here. When testing our algorithm, we defined the $cycle\_dif$ as the total number of cycles needed to reach a goal state of the a-star based algothrim minus that of the orignal greedy algorithm. A detailed test result of 1024 randomized starting states with max number of nodes generated limitted to 1536 can be found in Table 1. It can be seen that, the a-star based algorithm can find a better solution than the original greedy althorithm in the case of more than 30 percent.

$cycle\_dif$	counts	percentage
0	666	65.039
-1	336	32.813
-2	21	2.050
-3	1	0.098

Table 1. testing result of a-star based algorithm v.s. greedy algorithm

4 Heterogeneous Goalie Selection

Heterogeneous goalie was introduced in *rcssserver* 14.0. We have not found some theoretical method for heterogeneous goalie selection currently, but we have built a simulation based selection system. In our system a virtual goalie of each heterogeneous type was placed at point $(0, 0)$ with randomized body direction and velocity , and was ordered to run to a nearby randomized intercepting point quickly. Assume that the number of cycles of running and the catchable probability of the last cycle was recorded as *cycle* and *prob*, the evaluation value of such heterogeneous type during this simulation process was calculated by the simple formula as flows.

$$eva = \frac{prob}{cycle + 1.0}$$

We then evaluate a heterogeneous type by calculating the sum of *eva* of many different simulation processes. Obviously, the larger the sum is, the better the real ability of catching ball of that heterogeneous type is. Using this selection method, we can find the best heterogeneous type for goalie under a very realistic match environment.

5 Conclusion and Future Work

In this team descreption paper, we present our current research effort and some newly introduced techniques from last competition. According to the test result, it can be seen that both the low level skills and the high level strategy of our team have been improved. We are still working hard to improve our team, with the research of MDP/POMDP based decision-making theory and the application of kinds of techniques from artificial intelligence.

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. Craig Boutilier, Thomas Dean and Steve Hanks: *Decision-theoretic planning: structural assumptions and computational leverage*, The Journal of Artificial Intelligence Research, 1999.11:1-94.
5. SHI Ke and CHEN Xiao-ping: *Action-Driven Markov Decision Process and the Application in RoboCup*, Journal of Chinese Computer Systems, in press.