Evolving RoboCup2D Agents Based on PSO

Liu Zhao
College of Computer Science and Technology
Wuhan University of Science and Technology
Wuhan, China
liuzhao@wust.edu.cn

YE Shaoxiong
College of Computer Science and Technology
Wuhan University of Science and Technology
Wuhan, China
625517303@qq.com

Dong Chunchen
College of Computer Science and Technology
Wuhan University of Science and Technology
Wuhan, China
2256397010@qq.com

Yang Yang
College of Computer Science and Technology
Wuhan University of Science and Technology
Wuhan, China
youngyang_li@foxmail.com

Abstract—In order to build intelligent robots to accomplish soccer game tasks, this paper introduces evolutionary computing in agent architecture for perception, planning, and action: (1) an architecture based on PSO is proposed, which made up of 4 levels: atomic action, combo action, behavior and policy. (2) by offline training, agents format perception rules and relevant parameters, to optimize perception method for the position, orientation and other information; (3) according to the granularity, functions, and parameters manually specified, PSO builds a set of combo actions, which described by atomic actions, parameters and execution results; (4) according to game environment and a few task rules, PSO searches for task, behavior, and combo actions, as a whole, to accomplish the game tasks. The simulation experiments on RoboCup2D platform show that, agent based on PSO is a robust and flexible robot control method: given evaluation methods and implementation frames, it is able to learn rapidly in real environment, and displays planning behavior without the use of classical planning techniques.

Keywords—RoboCup2D; agent; architecture; planning; particle swarm optimization

I. INTRODUCTION

RoboCup and other robot soccer games provide challenging platforms for robotics and AI research[1]. Among them, the RoboCup2D, a platform enables for two teams of 11 simulated autonomous robotic players to play soccer, is very suitable to research: a) agent architecture, b) machine learning, especially agent cooperation and confrontation, and c) real-time agent planning[2].

According to perception, planning, action and their relationships, agent architectures are divided into: a) hierarchical structure, b) reactive structure, and c) hybrid hierarchical/reactive structure. For soccer robot or most other agent applications, the hybrid architecture is more successfully than hierarchical or reactive structures. As a typical instance of hybrid architecture, a triple-tower architecture and teleo-reactive programs[3] have been proposed in 2001 by Nils Nilsson as a robust agent control structure. After that, they have been successfully applied to a variety of domains. As shown in Fig.1, triple-towers means: a) perception tower, b) model tower and c) action tower. Main features of the architecture shown below:

1) Teleo-reactive programs were applied in the action tower. It is a mixture of top-down hierarchical control structure and agent-based bottom-up approach, using a tree data structure called the TR. The agent is directed towards a goal based on continuous evaluation of perceptual inputs. During each iteration cycle, searching start from the root task, value the condition of each task in turn until it finds the first established the tasks node, and then immediately executed action or action sequence associated with it.

2) Perception rules applied to perception tower. They create increasingly abstract descriptions of the current environmental situation starting with the primitive predicates produced by the robot’s sensory apparatus.

3) Truth maintenance system, or TMS, applied to model tower, which continuously kept faithful to the current environmental situation.

Figure 1. Nilsson’s triple-tower architecture

The designs of rules and actions in the architecture are the key and the hardest work in such applications. Besides Nilsson, many researchers [4,5] have researched the automation of them. In 2003, genetic programming was applied in the architecture for block stacking robots by Kochenderfer[6]. As an evolutionary computing method, genetic programming is proposed to generate teleo-reactive rules and actions of robots. Evolving teleo-
reactive programs not only improve the adaptability of the system, but also proved to be a new approach to gain knowledge of block stacking actions.

Similar to Nilsson’s triple-tower architecture, the most popular approach to building RoboCup2D agents is hybrid hierarchical/reactive architecture, shown as Fig. 2.

Policy layer is the highest layer among decision modules. For global team interests, it determines the behavior of the agent and the general tactics. Task layer is to develop specific tasks, including agent moving, ball catching, ball intercept, ball dribbling, ball passing, ball shooting, ball bashing, and so on. Action planning is arrangement basic actions which is acceptable for platform to execute.

There are some disadvantages in the traditional soccer agent architecture. Firstly, knowledge and preconditions about actions are very difficult to acquire. Although some methods, such as grammatical inference [], knowledge mapping, Q-learning, and reinforcement learning methods have been used, flexible and effective knowledge and veracious preconditions often fail to meet the requirements. Secondly, the architecture could not timely response to dynamic environment, and it is rigid when the top policy changed from one to another. Most of scenarios in the games are arranged by the programmer. Thirdly, and the most important is that, it didn’t take account of the interactive relation between the layers. Infact, it is not reasonable make the top policy without calculating of advance action layer and basic action layer, especially when the related information is deficient.

II. ARCHITECTURE

To get enough action knowledge which is essential for the agent decision making, agent actions are broken down into 4 levels: atomic action, combo action, behavior and policy. Soccer robot architecture based on PSO made up of two parts: offline learning architecture, and online strategy architecture. PSO is used to perception training, action training, and decision making.

A. Offline Learning Architecture

Perception training and action training are conducted under offline condition. By sense and behavior in a specified environment, PSO can get perception rules, perception filter, and sense parameter. Offline learning architecture is shown as Fig.3.

B. Online Strategy Architecture

Decision-making can be seen as a multivariable optimization process. Multivariable means a combination of task, behavior, and action, which determine the correctness and efficiency of action results. Although the component of the solution, include task, behavior and combo action, is hierarchical and related to each other, but PSO can process them according to the final action results.

III. COMBO ACTION LEARNING BASED ON PSO

Combo action is designed to get general action skills through training. Combo action learning is to get action sequence, action precondition, action duration, and action results.

A. Atomic Action Set

In RoboCup2D, agent has some basic actions, such as dash, turn, kick, tackle, catch, move, turn neck, change view, say, point to, attention to. When assigned parameters, those actions could be output and executed. They are simplified and formulized, form an atomic action set, as Tab.1 show.
Frame design for PSO is very important. Frames should not only be easy to apply existing knowledge, but also must be prepared to form new knowledge. Based on atomic action set, seven combo action frames are designed according to soccer game tasks and agent environment:

1. Hard_kick.
2. Dribble_ball.
3. Dribble_pass.
4. Location_arrive.
5. Location_rushing.
7. Block_tackle.

### TABLE I. ATOMIC ACTION SET

<table>
<thead>
<tr>
<th>Action</th>
<th>1st</th>
<th>2nd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dash</td>
<td>Power range (-100,100)</td>
<td>null</td>
</tr>
<tr>
<td>Kick</td>
<td>Power range (0,100)</td>
<td>Direction range (-100,100)</td>
</tr>
<tr>
<td>Tackle</td>
<td>Power range (-100,100)</td>
<td>null</td>
</tr>
<tr>
<td>Turn</td>
<td>Angle range (-180,180)</td>
<td>null</td>
</tr>
</tbody>
</table>

### TABLE II. PARTICLE STRUCTURE DESIGN

<table>
<thead>
<tr>
<th>Particle Structure</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>applied layer no. list</td>
</tr>
<tr>
<td>...</td>
<td>particle fitness</td>
</tr>
<tr>
<td>...</td>
<td>top policy layer</td>
</tr>
<tr>
<td>...</td>
<td>behaviors layer structure_1</td>
</tr>
<tr>
<td>...</td>
<td>behaviors layer structure_2</td>
</tr>
<tr>
<td>...</td>
<td>basic actions layer structure_1</td>
</tr>
<tr>
<td>...</td>
<td>basic actions layer structure_2</td>
</tr>
</tbody>
</table>

After the 7 action frameworks were designed, for each combo action, initialize n frameworks as a population of evolutionary computation. Execute every combo action, and evaluate performance and efficiency. Then, some bad frameworks are phased out, and nice frameworks can be crossover and mutated. New frameworks form a new population, a new loop repeated if the combo action result is not satisfactory.

![Figure 5. Combo Action Frame](image_url)

![Figure 6. Evolving Combo Actions Based on PSO](image_url)

### IV. ONLINE STRATEGY BASED ON PSO

We integrate planning layers with each other in the particles, which means, different planning layers be valued as a whole, but be initialized and evolved respectively. During valuating, some knowledge about the relationship between layers is taken into account. Fig.7 shows soccer robot strategy architecture based on PSO.

![Figure 7. Soccer robot strategy based on PSO](image_url)

In the soccer robot strategy, we found that, without existing knowledge, it is not feasible to search solutions in real time. So we decompose the main goal into some tasks and apply related knowledge in the strategy. To make the...
swarm particles chase after different tasks at different situation, the structure of it is designed as Tab.2 shows.

With every planning layer structure designed as Tab.3 shows:

TABLE III. PLANNING LAYER STRUCTURE DESIGN

| Planning Layer Structure | …… parameters of the structure velocity vector of the structure position vector of the structure |

In the PSO, particles presented different solutions to different tasks. The structures be used, value-function, initialized and evolved methods are varied according to the situation of the game.

V. SIMULATION STUDY AND CONCLUSION

We implemented the soccer robot strategy based on PSO on RoboCup2D platform. The simulation experiments show that, it is a robust and flexible soccer robot control method.

Statistics of instantaneous working particles kinds show as Tab.4. Results show that, there are always more than one tasks being prepared simultaneously, but only the best one be selected and executed. The turning from one task to another is smooth and efficient.

TABLE IV. WORKING PARTICLES FREQUENCY

<table>
<thead>
<tr>
<th>Particles</th>
<th>Tasks</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hard_kick</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>Dribble_ball</td>
<td>35</td>
</tr>
<tr>
<td>3</td>
<td>Dribble_pass</td>
<td>46</td>
</tr>
<tr>
<td>4</td>
<td>Location_arrive</td>
<td>44</td>
</tr>
<tr>
<td>5</td>
<td>Location_rushing</td>
<td>36</td>
</tr>
<tr>
<td>6</td>
<td>Block_robot</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>Block_tackle</td>
<td>15</td>
</tr>
</tbody>
</table>

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant No. 51174151).

REFERENCES