Simulation League, Team *Polytech100*, pages 102–108
http://www.ep.liu.se/ea/cis/1999/007/23/

Polytech100

Polytech100

Leo Stankevich

LS: Saint Petersburg State Technical University

Abstract.

THE COGNITIVE APPROACH TO DESIGN OF SOCCER AGENT

L. Stankevich

Saint-Petersburg State Technical University, Politekhnicheskay 29, (812)247-42-14

e-mail: stankevich@phtf.stu.neva.ru

Abstract

Soccer (association football of robot) makes a good example of the problem of the real world which is moderately abstracted. This play has being chosen as one of standard problems for study on multi-agent systems. We are developing the soccer agent basing on the cognitive approach. Our soccer agent can learn whether he shoots a ball or passes it because he has the neurological modules inside. .

1 Introduction

The distributed artificial intelligence (DAI) and multi-agent systems (MAS) research directions became active at present. We see wide using of application MAS-technology for design DAI real-time control systems of robots grouped with the common work goal. The complexity of the design DAI real-time systems with MAS-technology has caused the machine learning (ML) using. The computation complexity is being replaced with system learning. The max reaction to situation can be gotten only taking dynamic changes into account.

The MAS problems are widely investigated in computer graphics (CG) [3-6]. We have CG approaches to design of agents. In this paper we present a hierarchy of multiple levels of control in MAS that breaks the general problem of character behavior and animation into a number of sub-tasks. Subdividing the problem allowed us to tackle all parts thereof and create the effective solutions. We propose a powerful methodology allowing us to reformulate the most intricate problems of intelligent agent control that can be easily described in informal (verbal) way, as well as mathematical problems and find the efficient algorithms to solve them. The detailed descriptions of optimal foot placement and local trajectory planning algorithms are presented.

Soccer (association football of robots) is team game in which a players have a cooperation [1,7]. This is a real-time game where situation changes dynamically. Soccer was chosen as one of problems for studying on multi-agent systems [9,10]. We are designing the soccer agent using the cognitive approach. We present the decision making algorithm for cooperative action among soccer players. We developed the learning modules for partial implementation of decision making at high and low behavior levels of soccer agent. This soccer agent can be used as a client of Noda's Soccer server for participation in simulation league of RoboCup [8,11]. The soccer team

for participation in RoboCup is organized in Saint-Petersburg State Technical University [2].

The specific features of our agent considered in this paper follow.

1. The agent is designed as a cognitive system that is the learning intelligence system with nervous- system behavior, function, and structure. The knowledge acquisition is produced with learning in the work process. The knowledge is being kept and used in the associative neurological form.

2. The specific evaluation function used for a decision making. This function is used at high level control with a decision tree. It can be tuned during the learning process for adaptation to environment.

3. The middle level behavior function set is used for agent's individual tactics with operative change for adaptation to game situations.

4. The specific learning module for automatic interaction among others agents during game. It improves coordination at attack or defense.

2 Methodology MAS problem decisions

After a thorough investigation of the MAS problems we found that it is virtually impossible to employ sophisticated algorithmic and conventional if-else approach only as it requires an immense number of cases to take care of making the whole system unmanageable. Moreover, some tasks appeared to be NP-hard so it is useless to find the exact solution.

Therefore we came up with the idea of conjunction of algorithmic methods with numeric function minimization. The main idea of the method follows. We define a space of states P and assume that at any given time the agent is in a certain state (describing its position, orientation, speed, health, weapon, etc.). When the agent is going to make a decision the characters state changes (e.g., it moves to another point). Hence we determine a set of permissible states P and associate some weight (stress) with the move from one state to another. Then our goal is to locally minimize the constructed stress function S(p) over the set of all permissible states thus finding the best new state as it is expressed by equation ($\star$). To make a multistep minimization (e.g., in local trajectory planning) we employ algorithmic approach to subdivide the problem into several local sub-tasks and then combine the results. Also this allows us to select optimal solutions while interacting (and counteracting) with other characters (and human players). Similar methods of conjunction of numerical minimization and heuristic algorithms on discrete structures (weighted graphs) are used on many levels of abstraction.

The use of the minimization approach poses a new set of problems. It is difficult to determine how to construct the stress function S(p) and tune up its parameters. Indeed, each function is a sum of multiple appropriately weighted primitive functions $S_i(p)$ that measures the stress of the transfer to another state. In certain cases, the number of primitive functions rapidly grows up to ten and more thus making it difficult to accord them. Additionally, each primitive function $S_i(p)$ may have many adjustable parameters - see equation ($\star\star$).

$$Min_p S(p), S(p) = \Sigma_{i=1}^{n} S_i(p), (p = 1, ..., P) \qquad (*)$$

where p - a point from the set of permissible states P

n - the number of primitive stresses (the number of factors affecting the walk)

$S_i(p)$ - primitive stress function

$$S_i(p) = C_i(G_i)\Phi_i(p, G_i, Q_i) \qquad (**)$$

where G_i - set (vector) of global parameters (supplied by upper levels of control)

Q_i - set (vector) of function-dependent constants (to be adjusted at the stress function construction)

$C_i(G_i)$ - relative weight of stress function $S_i(p)$ depending on global parameters.

Function Φ_i is usually taken from the class of functions Φ and has the following properties:

- it has a single minimum;
- it grows to infinity when its arguments run to infinity.

3 Soccer agent's behaviors

The soccer agent has a multi-level behaviors like football man does. The agent is based on simulating several behavior functions that formally can be mapped as:

SBF = (CF, MF);

CF = insf(EF, DM, IA, BL);

EF = insf(TM, CD); DM = insf(AT, DF);

IA = insf(MS, RL, FM); BL = insf(CH, GM);

MF = insf(Pass, Dribble, Avoiding, Pressing, Intersect);

MF = compf(Kick, Turn, Dash, Inhibit),

where: SBF - Soccer behavior functions; CF and MF - Cognitive and Motion functions; EF - Evaluation function; DM - Decision making function; IA - Interaction function of players; BL - Behavior learning function; TM - Teammate positions; CD - Contradictor positions; AT - Attack decision; DF - Defense decision; MS - Message change; RL - Role of agents; FM - Formation from roles; CH - Coaching learning; GM - Game learning. They functions determine the team strategy and tactics in battle with contradictors. Low level includes the individual motion skills of players (MF). These are Pass, Dribble, ball Intersect, Avoiding of obstacles, Pressing on contradictors. The executive final actions of players include Kick, Turn, Dash, and Inhibit. The operations 'insf' and 'compf' mean 'insert functions' and 'component functions'.

4 Evaluation situation and position of the players

The correct evaluation of situation and position of the players is fulfilled with special evaluation function (stress function). This function evaluates the position where the player has ball with relation to teammates and contradictors. The function for the i-th player is:

$$S_{ij} = F\{C_{ij}, W_{ij}, \Sigma_{q=0}^{n}[\alpha_q, P_i(q, M(t))]\},$$
$$i = 1, ..., 11$$

where C_{ij} - interaction force coefficient of i-th and j-th players from coach; W_{ij} - adaptive coefficient for mapping of the game experience with similar contradictors; α_q - prediction coefficient on period t; $P_i(q, M(t))$ - position evaluation with current position $M(t)$ teammates and contradictors; F - constrained function. This function P is implemented on neurological module with learning.

5 Decision making

The choice of pass, dribble and shoot is executed on decision tree. There are branches of the tree that help to decision making for player with ball or without it; The attack and defense in the current game situation. The decision making algorithm includes:

- calculation $K_{ij}, (j = 1, ..., 11)$ for i-th player with ball;
- choice of of Q perspective candidates on max K_{ij};
- calculation $K_{qj}(q = 1, ..., Q)$ with respect to q-th perspective;
- choice of the action on max K_{qj} (pass or dribble); the player without ball makes decision in branches of decision tree basing on hard algorithm.

6 Neurological module

The partial evaluation function $P_I(q, M(t))$ is implemented with neurological module that can learning on examples (patterns). This is associative fuzzy-logical neural network with layered structure. Formally it can be represented as:

$$NM = \{X, H_x, St, W, H_y, Y\}$$

where X and Y - input and output vector variables; H_x and H_y - hidden variables for activator of module; St and W - structure and connection weights for the activator of module. This module executes any vector-vector mapping $X \longrightarrow Y$ on many examples (patterns) with learning. It can be used for implementation of individual skills too.

The neurological modules based on fuzzy logic are used in the agent. There are follow procedures are used in the modules:

Fuz - fuzzification calculating the membership degree μ^q_{xi} of variable x_i to q-th fuzzy granule this variable;

Wagr - weight aggregation calculating the membership degree μ^q_y of variable y_j to fuzzy granule this variable;

Dfuz - defuzzication calculating the value of output variable y_j;

Wupd - weight update at learning of the neurological module by examples of map X to Y.

Let us consider on the example of two input variables x_1 and x_2 and one output variable y. Let every variable is decomposed on the three fuzzy granules and membership function have the triangular form. Then the procedures look as:

Fuz:

$$\mu^q_x = ((x - l_q)c^{-1}_{ql} \leftarrow l_q < x < m_q) \vee ((r_q\text{-}x)c^{-1}_{qr} \leftarrow m_q < x < r_q) \vee (0 \leftarrow else) \; ;$$
$$c_{ql} = m_q - l_q \; ;$$
$$c_{qr} = r_q - m_q \; ;$$

Wagr:

$$\mu^q_y = \vee^{i=9}_{i=1}(w^q_i \wedge^{q,r=3}_{q,r=1} (\mu^q_{x1}, \mu^r_{x2})) \; ;$$

Dfuz:

$$y^q = 0.5(y^q_n + y^q_r) = 0.5(\mu^q_y/cqn + \mu^q_y/cqr) \; ;$$
$$y = \Sigma^K_{q=1}\mu^q_y y^q / \Sigma^K_{q=1}\mu^q_y$$

Wupd:

$$w_{p+1} = w_p(p/p{+}1){+}w_p(\mu^*_y/\mu_y)/(p{+}1) \; ; \; p = 1, ..., n_p$$

Where: l_q, μ_q, r_q - coordinates of left, middle, and right points of triangle base in membership function, $\wedge$ - fuzzy intersection (min), $\vee$ - fuzzy unify (max), w_i - weight of structure connection of the module's activator.

7 Individual skills

The individual component ball actions of agent are implemented on hard algorithm or neurological module. We used combine method. The actions such as shoot, run, turn dash, inhibit are executed on hard algorithm. The action as pass, dribble, intercept, avoiding, pressing , corner-kick are more complex agent skills. They was implemented on neurological module with ML methods for tuning of dribble with avoiding obstacles.

The goalkeeper-agent has some specific features. It could predict ball motion direction and intersect its. The ML methods are used for the tuning of skills of ball intersection too.

8 Interaction of agents

The multi-agent soccer system is client-server program environment. The agents are divided on two teams that have cooperation and collaboration. A team agents have their own roles and formation. Let

$$CA = \{ca_1, ..., ca_m\} semcR = \{r_1, ..., r_m\}$$

, where ca_i - i-th agent; r_i - i-th role. Formation F is components U_j from roles R as

$$F = \{R, U_1, ..., U_k\} \ ;$$
$$U_j \subset R \ ;$$
$$U = \{r_{i1}, ..., r_{ij}\}(r_{ij} \neq r_{iq})$$

A map $CA \rightarrow R$ is flexible as it depends on time. The roles can be assigned to different homogeneous agents. We have used the special neurological modules for agent interaction. Formation can affect the agent's external behaviors by specifying inter-role interaction. Since roles can be re-used among formations, their formation-specific interactions cannot be included in the role definitions. Instead of it the interactions are a part of the formation specification.

9 Conclusion

In result, it is designed the cognitive soccer agent that has flexible behavior structure with learning. This agent is being tested for participation in RoboCup. The testing games have showed some advantage of this agent in team tuning that can be produced in game time. After learning of the agents while playing next game can be more successful.

Bibliography

1. Gorodetsky V.I. RoboCup'98 - World Cup on football among robots PCWeek / RE no. 30-31, 1998 (Rus. lang.).

2. Stankevich L.A. There are competed the Saint-Petersburg soccer-robots Sport Marketing, no. 1, 1999 (Rus. lang.).

3. S. Zhukov, A. Iones, “Real-Time Behavioral Control of Intelligent 3D Agents in Complex Synthetic Environment”, in Proc. of SCAI ´97 (FIN, Helsinki, 1997).

4. B. M. Blumberg, “Old Tricks, New Dogs: Ethology and Interactive Creatures”, Ph.D Thesis, M.I.T., Media Laboratory, Learning and Common Sense Section, 1996.

5. B. M. Blumberg, T. A. Galyean, “Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments”, SIGGRAPH’95 Conference Proceedings, pp.47- 54.

6. Interactive Computer Animation, Ed. by N.M. Thalmann, D. Thalmann, Prentice Hall, 1996.

7. Michael K. Sanota. Real-time intelligent behavior in dynamic environments: Soccer-playing robots. Master’s thesis, Univ. Of British Columbia, 1993.

8. I. Noda. Soccer server: a stimulator of RoboCup AI Symposium’95, Japanese Society for AI, 1995.

9. M. Veloso, P. Stone, K. Hun, and S. Achin. The CMUunited-97 Small Robot Team RoboCup-97: The First Robot World Cup Soccer Games and Conferences, Ed. H. Kitano, Springer Verlag, Berlin, 1998.

10. P. Stone, M. Veloso. A layered Approach to Learning Client Behaviors in RoboCup Soccer Server Applied Artificial Intelligence, 12, 1998.

11. Result of RoboCup-97. Accessible from http://www.RoboCup.org.