DOI: http://doi.org/10.32792/utq.jceps.10.01.05

RSPSO Algorithm for Finding the Best Point for Robot to Score a Goal

Asseel Jabbar Almahdi

University of Thi-Qar, Computer Science and Mathematics Collage, Computer Science Department

Received 3/7/ 2018 Accepted 17/7/2019 Published 20/1/2020

Abstract:
The issue of robotic football game is one of the most complex multi-agent systems. In order to make the game more enjoyable and exciting by scoring goals, therefore it's important for a robot football agent to have a technique on how to score a goal. This paper proposes RSPSO algorithm, where the particle swarm optimization algorithm (PSO) based on Q-learning is used to find the best point from the goal gate to shoot the ball toward it. As we show in simulation, our method enables the robot to learn soccer skills to score a goal.

Key words: Reinforcement learning, Particle swarm optimization (PSO), robot soccer, Q-learning.

1.Introduction:

RoboCup matches has hosted a location for discuss in the space of artificial intelligence (AI) and robotic [1]. These matches have been the basis with the standard issue design of testing various technologies in that field. For ease of research in various AI and robotic fields, in addition to the real robot leagues, the simulation league was also introduced, among which the football simulation league is the most popular and oldest league in the RoboCup matches. The issue of football is one of the most complex multi agent systems, in which the agents play the role of football players. Since the main goal of a football game is to score goals, it's important for a robot football agent to have a clear policy or technique on how to score a goal in a particular situation [2]. An AI method, Reinforcement learning (RL) is one of the foundations of learning in smart systems that act on the basis of the causal relationship [3]. In this method, the learning of the intelligent agent, depending on the situation in the environment, performs an action on the environment and waits for the result of its operation, this result can be in the form of a reward or punishment. If the result is in the form of a reward, the performed action is desirable and the agent approaches the target which is located in that environment, but if the result is in the form of punishment, the action is undesirable and the agent has gone away from his goal. The agent must learn what actions to take in order to earn a better reward and ultimately achieve his goal. The Q-learning algorithm is one of the reinforcement learning method (RL) which acts on the Markov property and do not require knowledge of the environment and is used to detect a best politics for model-free problems [4].

In robot soccer space, reinforcement learning is implemented to learn surveillance of hardware such as walking types for humanoid-robots [5, 6] and ball trapping for 4-legged robots [7]. It is also implemented to learn soccer skills individually and team level strategy. Many of examples of this learning such as kicking pattern [8], shooting skills methods [9], dribbling method [10], aggressive defense behavior [10] and scoring goals penalty [11]. Also, Stone's Keep Away game [12] is an RL application to control the attack of a 2D robot soccer [13].

Reinforcement learning has been applied previously on the robotic soccer. The author in [14] used it to learn the real robot to hit a ball into the goal while avoiding a contender. Also using goal scoring as the goal-state, RL was successfully applied to let agents' team to learn cooperative passing and hitting ball rules using a Monte Carlo learning method, as dissenting to the TD learning [15]. But in [16], observational RL was used to update players' locations on the yard Depending on the location of the ball. In [17] reinforcement learning was suggested to improve soccer skills for robots, where Multi-Layer Perceptron was utilized in the algorithm as a function approximator to treat the continuous state variables. While Batch Reinforcement Learning was used by apply Q- batch instead of Q-learning to obtain effective robotic soccer controllers on physical podiums [18]. The paper in [19] proposed approach that uses reinforcement learning with decision-making system for Robotic Soccer Games, where the ISVM is used to assess situations and the rewarding.

In this paper, the particle swarm optimization algorithm based on Q-learning is used to find the best point from the target to shoot the ball toward it. The rest of the paper is arranged as follows: Section 2 presents the main concepts of Particle Swarm Optimization algorithm. Section 3 discusses the RSPSO Algorithm. Section 4 and Section 5 offers Simulation results and Conclusion of the paper respectively.

2. Particle Swarm Optimization:

PSO is a bio-inspired approach that optimize a problem by seeking iteratively to improve a candidate solution and moving the particles with consider to a given measurement of fineness. Each particle's motion is affected by its local best-known position (pbest) but, is also planned toward the best-known positions (gbest) in the search space, then updated as better positions. This is prospective to move the particles toward the best solutions [20]. Each particle (i) tries to change its position by using the following equations to get the best solution:

$$v_i(t+1) = \omega\, v_i(t) + c_1 r_1\big(pbest_i(t) - x_i(t)\big) + c_2 r_2\big(gbest_i(t) - x_i(t)\big) \qquad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \qquad (2)$$

Where $v_i(t)$ is the particle velocity at time t, $x_i(t)$ is the particle position at time t, $pbest_i(t)$ is the best particle individual solution at time t, $gbest_i(t)$ is the best particle swarm solution at time t, c_1 and c_2 are acceleration coefficients, r1 and r2 are random values and $\boldsymbol{\omega}$ is inertia weight.

3. RSPSO Algorithm:

The purpose of using a reinforcement learning algorithm with particle swarm optimization algorithm is to increase the ability of search. In the reinforcement learning algorithm, each agent must learn which operations are better in a defined state and select them with a high eventuality. This is the ability for all agents, since each agent can experience different cases on the search space, and it's best to experience a subjective policy based on its location. For each particle in Q-Learning Table (QT), the state and the action must be defined, and the method of updating (QT) and choosing the convenient action is predefined. In the learning process of each particle when an agent wants to go the next step, an action is selected (PSO parameters) based on his position and (QT) and move based on the motion policy (PSO rules), and after gets a new point and a new fitness updates the (QT) based on the new state.

In this paper, the following equation shows the objective function in finding the best point from the goal gate for goal scoring.

$$O = sec^2(a) \quad if \;\; a \leq 90 \tag{3}$$

or

$$O = 1 + tan^2(a) \quad if \;\; a > 90 \tag{4}$$

where:

$$a = \text{Proposed ball angle of the PSO} - \text{Target position angle} \tag{5}$$

Then, the agents must choose their own parameters to move to the next step. The update of the velocity and position of each particle are given by the following equations:

$$x_i^n = x_i^n + v_i^n \tag{6}$$

$$v_i^n = v_i^n + c_1 r(D_1) + c_2 r(D_2) \tag{7}$$

$$D_1 = pbest_i^n - x_i^n \tag{8}$$

$$D_2 = gbest_i^n - x_i^n \tag{9}$$

where n is the next number and i is the particle 's number.

A state in the learning algorithm Q must be a situation in which the algorithm chooses the appropriate action based on it, in the proposed method, the state is known as the two distance of a particle in the repetitions. These two distances D_1 , D_2 are chosen as two dimensional states, because they are very useful parameters that effect on the new speed and position of the particle.

To choose an action, we must initialize a Q-table. At first, the values of the cells in the Q-table are equal. Consequently, after a move, the value of the cells should be filled with a proper state and action based on the equation below, and all the cells in table Q will be changed to that state by normalization. The following function is for filling the table cells and to select the parameter for a move.

$$Q_{t+1}(state: D_1.D_2.action: c_1.c_2.r.\omega.v) = \alpha\, Q_t(state: D_1.D_2.action: c_1.c_2.r.\omega.v) + \beta(f_t - f_{t+1}) + R \qquad (10)$$

Where: α, β are constant values that effect on the learning rate.
$(f_t - f_{t+1})$:is the fitness for a minimization problem.
R :is a reward.
Next step is selecting a parameter. It is a procedure that pick out PSO parameters for an agent based on its Q table. This procedure should select a cell from the Q table, so it randomly selects a set of parameters, and a cell with higher Q values has linearly a higher chance of choosing it. Finally calculate the fitness and check stop condition. If the algorithm does not stop, it will continue and update the Q-tables of all agents and will repeat the main loop for selecting a parameter. See the flow chart figure (1).

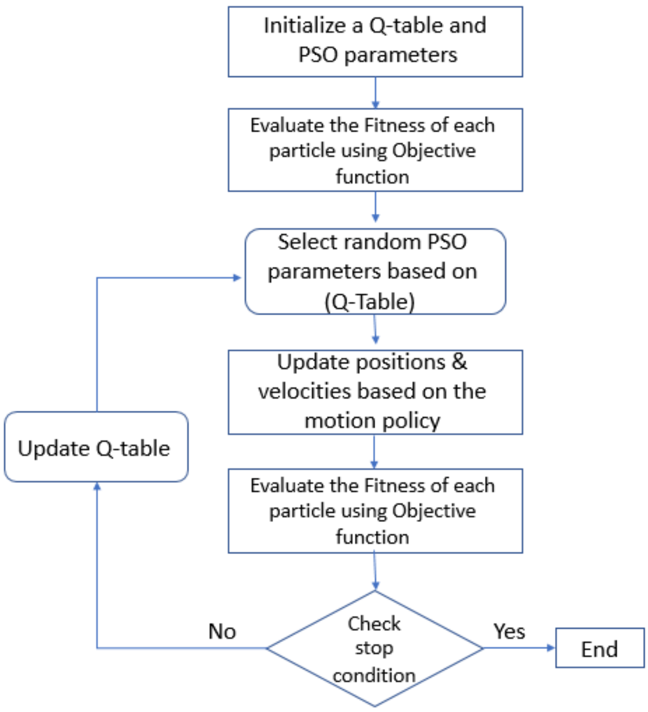

Figure 1: flow chart of RSPSO Algorithm

4. Simulating Results:

The simulation of the robot and the environment was carried out using MATLAB. To show the results, the robot shoots the ball towards the gate of the goal at different distances and we consider the success rate at these distances. The success rate is the ratio of the number of goals scored to the total number of shots hit towards the gate. For example, if the player is 1.5 meters away from the goal, 75% of the number of shots that will be pushed to the goal gate will be turned into goals, but at lower distances, such as 15 cm, almost all the strikes toward the gate they turn into goals.

According to the chart below Figure (2), it is possible to observe a higher success rate to score goals when the distance is closer to the goal gate.

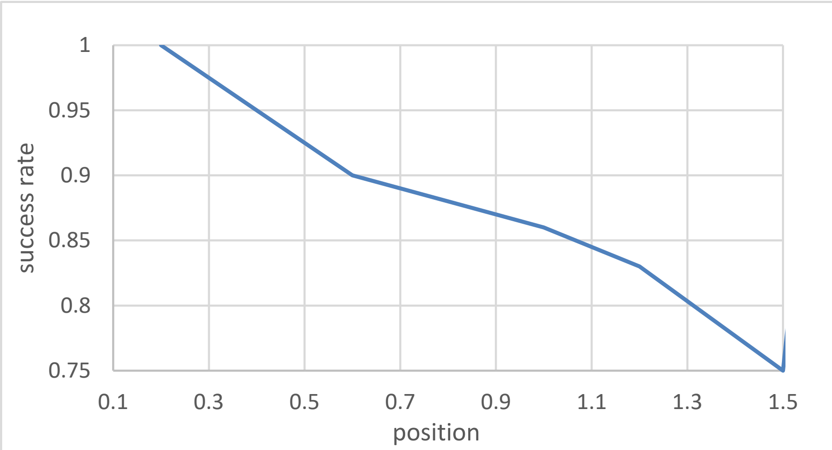

Figure 2: A chart to measure the success rate in scoring goals according to the position

In the Figure (3), an overview of the simulation environment is shown for the condition that the ball has turned into a goal.

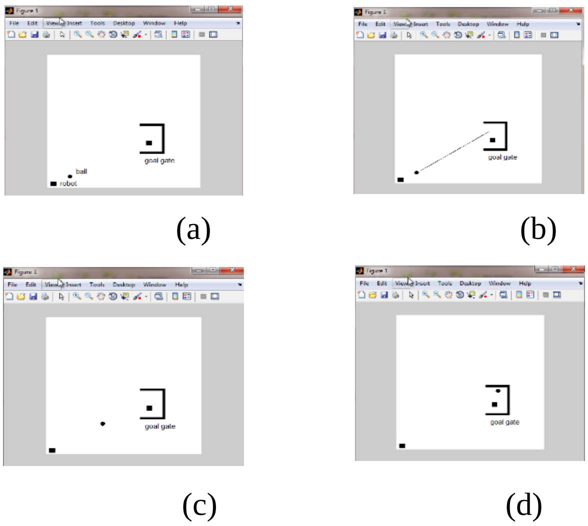

Figure 3: simple environment shows goal scoring screenshots

5. Conclusion:

This paper presents, RSPSO algorithm which is combining of a Q-learning and optimum particle swarm optimization algorithms to find the best point or location of the robot soccer to score goals. The optimum particle swarm optimization algorithm was used to determine the direction of the ball according to the target direction by checking the distance between the goal position and the optimal point of the ball based on selecting the parameters from the Q-table. To demonstrate the performance of the proposed approach, experiments using simulation are performed as shown in figure (3).

References

[1] Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., Robocup, H.M.: "A challenge problem for ai. AI mag.", **18**(1), 73, 1997.

[2] Visser U & Burkhard HD, "RoboCup: 10 years of achievements and future challenges", AI Magazine, 28(2), p. 115, 2007.

[3] A. Likas, "Multivalued Parallel Recombinative Reinforcement Learning: A Multivalued Genetic Algorithm", In Proc, HERCMA Conference, 1998.

[4] Watkins CJCH," Learning from Delayed Rewards", Doctoral Dissertation, University of Cam-bridge (MA), 1989.

[5] Budden DM," Applications of machine learning techniques to humanoid robot platforms", Doc-toral Dissertation, University of Newcastle, Australia, 2012.

[6] Ogino M, Katoh Y, Aono M, Asada M & Hosoda K, "Reinforcement learning of humanoid rhythmic walking parameters based on visual information", Advanced Robotics, 18(7), pp. 677-697, 2004.

[7] Kobayashi H, Osaki T, Williams E, Ishino A & Shinohara," A, RoboCup 2006: Robot Soccer World Cup X", Springer, Berlin, Germany, Available from http://dx.doi.org/10.1007/978-3-540-74024-7_8, 2007.

[8] Riedmiller M & Gabel T, "On experiences in a complex and competitive gaming domain: Reinforcement learning meets RoboCup", Proceedings of the Computational Intelligence and Games, CIG 2007, IEEE Symposium, pp. 17-23, 2007.

[9] Duan Y, Liu Q & Xu X, "Application of reinforcement learning in robot soccer", Engineering Applications of Arti_cial Intelligence, 20(7), pp. 936-950, 2007.

[10] Riedmiller M, Gabel T, Hafner R & Lange S," Reinforcement learning for robot soccer", Autonomous Robots, 27(1), pp. 55-73, 2009.

[11] Hester T, Quinlan M & Stone P, "Generalized model learning for reinforcement learning on a humanoid robot", Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 2369-2374, 2010.

[12] Stone P, Sutton RS & Kuhlmann G, "Reinforcement learning for RoboCup soccer keepaway, Adaptive Behavior", 13(3), pp. 165-188, 2005.

[13] Neri JRF, Zatelli MR, Farias dos Santos C & Fabro JA, "A proposal of Q-learning to control the attack of a 2D robot soccer simulation team", Proceedings of the Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), pp. 174-178, 2012.

[14] E. Uchibe. "Cooperative Behavior Acquisition by Learning and Evolution in a Multi-Agent Environment for Mobile Robots". PhD thesis, Osaka University, January 1999.

[15] P. Stone and M. Veloso. "Team-partitioned, opaque-transition reinforcement learning". In M. Asada and H. Kitano, editors, RoboCup-98: Robot Soccer World Cup

II. Springer Verlag, Berlin, 1999. Also, in Proceedings of the Third International Conference on Autonomous Agents,1999.
[16] T. Andou. "Refinement of soccer agents' positions using reinforcement learning". In H. Kitano, editor, RoboCup-97: Robot Soccer World Cup I, pages 373-388. Springer Verlag, Berlin, 1998.
[17] M Yoon, J Bekker, S Kroon, "New reinforcement learning algorithm for robot soccer", ORiON journals, Volume 33 (1), pp. 1-20, 2017.
[18] Jo˜ao Cunha · Rui Serra · Nuno Lau., " Batch Reinforcement Learning for Robotic Soccer Using the Q-Batch Update-Rule", Springer Science+Business Media Dordrecht, J Intell Robot Syst, DOI 10.1007/s10846-014-0171-1, 2015.

[19] HAOBIN SHI, ZHIQIANG LIN, KAO-SHING HWANG," An Adaptive Strategy Selection Method with Reinforcement Learning for Robotic Soccer Games", IEEEAccess, VOLUME 6, 2018.
[20] Kennedy, J.; Eberhart, R. "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks IV. pp. 1942– 1948, (1995).