Q-Learning based Market-Driven Multi-Agent Collaboration in Robot Soccer

Article · April 2012

CITATIONS
25

READS
75

3 authors:

Hatice Kose
Istanbul Technical University
72 PUBLICATIONS **689** CITATIONS

SEE PROFILE

Kemal Kaplan
Bogazici University
13 PUBLICATIONS **105** CITATIONS

SEE PROFILE

H. Levent Akin
Bogazici University
115 PUBLICATIONS **701** CITATIONS

SEE PROFILE

Q-Learning based Market-Driven Multi-Agent Collaboration in Robot Soccer

Hatice Köse, Utku Tatlıdede, Çetin Meriçli, Kemal Kaplan and H. Levent Akın

Boğaziçi University
Department of Computer Engineering
34342 Bebek, Istanbul, TURKEY

Abstract. This work proposes a novel approach for introducing market-driven multi-agent collaboration strategy with Q-Learning based behavior assignment mechanism to the robot soccer domain in order to solve issues related to multi-agent coordination. Robot soccer differs from many other multi-agent problems with its highly dynamic and complex nature. Market-driven approach applies the basic properties of free market economy to a team of robots, to increase the profit of the team as much as possible. For the benefit of the team, robots should work collaboratively, whenever possible. Through Q-learning, a more successful behavior assignment policy have been achieved after a set of training games and the team with learned strategy is shown to be better than the original purely market-driven team.

1 Introduction

Although the utilization of multi-robot teams has become popular recently, as their performance are shown to be better, more reliable and more flexible than single robots, in a variety of tasks, the applications are still limited. Problems in the coordination of the robots, efficient usage of limited resources and communication burden discourages researchers to work on real-time problems with dynamic environments. Robot Soccer is a good testbed for multiagent collaboration in real-time, complex and dynamic environments.

Recently market-driven approach was introduced as an alternative method for robot coordination in Dias and Stenz [1]. It is highly robust and avoids single point failure, while increasing the team performance considerably. There are several applications of market-driven approach. The work in Zlot *et al.* [2] introduces the approach to multi-robot exploration. In Gerkey and Mataric [3] a work on auction based multi-robot coordination is presented. These implementations seem to work well but limited due to the static nature of the environment. Domains like agricultural areas are simple, static and do not require fast task allocation, planning and coordination as in robot soccer.

In Köse *et al.* [4] and [5], a market based algorithm is used for multi-robot coordination for robot soccer. Although the algorithm works well, a more flexible approach could be achieved by embedding a learning policy for providing an adaptive approach

to the behavior assignment mechanism. In this paper, in order to consider this possibility, a market-driven collaborative task allocation algorithm with Q-Learning based behavior assignment mechanism for robot soccer domain is proposed.

In robot soccer, teams of robots, that are capable of sight and movement, play matches against each other, and at the end of the game, the team with the highest score wins the match like in real soccer. In order to play soccer, the player robots must detect their location, the goals, the ball, the members of their team and the opponent team members(optional for high level planning), and place the ball in the opponent team's goal to score.

The use of reinforcement learning can provide policy learning capability. Consequently, the team can have a flexible behavior assignment policy when facing changing situations and as the team plays more games it can extend its experience, in other words, learn how to assign behaviors in certain conditions more successfully for the next time.

The rest of the paper is organized as follows: In the second section, reinforcement learning is introduced briefly. Market-driven method is defined in the third section. In the fourt section detailed information about the proposed approach can be found. In the fifth section, the results of the application of proposed approach are presented. In the last section, conclusions and suggestions for future work are given.

2 Reinforcement Learning

Reinforcement Learning (RL) is a learning method that can be used when the agent is only informed about the degree of correctness (or wrongness) of a sequence of actions that moves the agent from one state to another while searching a combination of actions to reach the goal state (Figure 1). Depending on this information, sometimes called *reward* (or *penalty*, if it denotes a wrong sequence of actions) or *reinforcement*, the agent learns to behave in an optimal sequence of actions in order to reach the goal state. Each action in each state is associated with a value called Q-value denoting the benefit for taking that action in that state. Q-values are refined through the learning process.

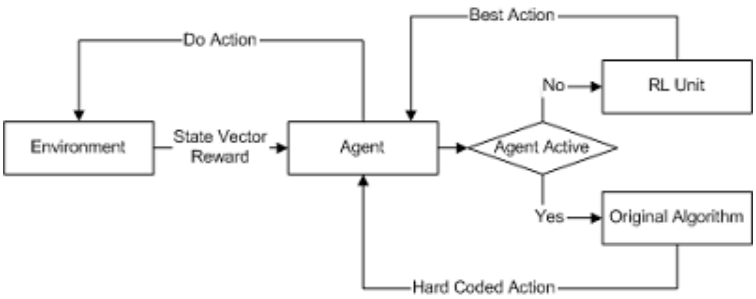

Figure 1. RL working schema

Peng [6] defines Q(λ)-Learning as a variant of RL and an extension to simple Q-learning. Q-learning algorithm uses only one step data while updating Q-values. Sutton [7] states that, eligibility traces can be used to keep track of all the actions taken by the agent to reach a terminal state. Q(λ) is widely used and it is generally believed to outperform simple one-step Q-learning, since it uses single experiences to update multiple state/action pairs (SAPs) that have occurred in the past. Generally, the Q-values learned by the agents are represented in tabular form with one output value for each input tuple. But it is not possible to represent more realistic world models with this approach, where the number of states can be prohibitively large or continuous. One way of handling such problems is using a function approximator.

Cerebellar Model Articulation Controller (CMAC) was introduced by Albus in 1975 [8] as a simple model of the cortex of the cerebellum. It is a biologically inspired learning method similar to neural networks. The main reason for using the CMAC is its efficiency in learning and operation, which makes it suitable for function approximation.

3 Market-Driven Method

The main goal in the free-markets is maximizing the overall profit of the system. If each participant in the market tries to maximize its profit, as a result of this, the overall profit for the system is expected to increase. The idea of the market-driven method for multi-robot teams is based on the interaction of the robots among themselves in a distributed fashion for trading work, power and information and hence "Collaboration by competition / cooperation". In general, there is an overall goal of the team (i.e., building the map of an unknown planet, harvesting an agricultural area, sweeping buried landmines in a particular area, etc..). Some entity outside of the team is assumed to offer a payoff for that goal. The overall goal of the system is decomposed into smaller tasks and an auction is performed for each of these tasks. In each auction, the participant robots (which are able to communicate among themselves) calculate their estimated cost for accomplishing that task and offer a price to the auctioneer. At the end of the auction, the bidder with the lowest offered price will be given the right of execution of the task and receives its revenue on behalf of the auctioneer. There are many possible actions that can be taken. A robot may open another auction for selling a task that it won from another auction, two or more robots may cooperatively work and get a task which is hard to accomplish by a single robot, or for a heterogeneous system, robots with different sensors/actuators may cooperate by resource sharing (for example, a small robot with a camera may guide a large robot without a vision system for carrying a heavy load).

In order to implement the strategy, a cost function is defined for mapping a set of resources (required energy, required time, etc...) to a real number and the net profit is calculated by subtracting the estimated cost for accomplishing the task from the revenue of the task.

3.1 Market-driven method in the Robot Soccer Domain

In Köse *et al.* [4] a market based algorithm was proposed for multi-robot coordination for robot soccer. Here, each team member calculates costs for its assigned tasks, including the cost of moving, aligning itself suitably for the task, and cost of object avoidance, then looks for another team member who can do this task for less cost by opening an auction on that task. If one or more of the robots can do this task with a lower cost, they are assigned to that task, so both the robots and the team increase their profit. Other robots take actions according to their cost functions (each takes the action which is most profitable for itself). Since all robots share their costs, they know which task is appropriate for each one so they do not need to tell others about their decisions and they do not need a leader to assign tasks. If one fails, another would take the task and go on working.

Consequently, the most important task is scoring the goal (making the ball enter into the opponent's goal). On the other hand, opponent goals should be avoided. The one who is the closest to the ball tries to get the ball to avoid opponent's ball possession, and to shoot it to the opponent goal. But if there are any other robots between the robot who has the ball and opponent goal, a shoot would not be successful. In such a case, it might be more appropriate for the robot not to take the risk and pass the ball to another team member who has a clearer sight.

Meanwhile, the robots calculate their defense cost for role assignment. After the primary attacker is assigned as explained above, the one with the lowest defense cost is assigned as the defender, and it tries to take the best position between the ball and its own goal area to avoid a possible opponent goal. The rest of the team is assigned to their roles such that; in the order of cost values, the robot with the second best cost would serve as the secondary attacker which would help the primary attacker in case of failure of a shoot, or completing an attack by having the ball after the ball is deflected from some obstacle on its way to the goal area. So it would place itself symmetrically to the position of the ball to catch the ball easily if it bounces back, the remaining robot is assigned as the third attacker, it tries to go to the ball, to help the other attackers in case of any failure.

The approach is shown in the flow chart given in Figure 2. The robot with the smallest score cost C_{ES} will be the primary attacker. Similarly the robot, except the primary attacker, with the smallest $C_{defender}$ cost will be the defender. If $C_{auctioneer}$ is higher than all passing costs ($C_{bidder(i)}$) then the attacker will shoot, else, it will pass the ball to the robot with the lowest $C_{bidder(i)}$ value. The cost functions used in the implementations are as follows:

$$C_{ES} = \mu_1.t_{dist} + \mu_2.t_{align} + \mu_2.clear_{goal} \tag{1}$$

$$C_{bidder(i)} = \mu_1.t_{dist} + \mu_2.t_{align} + \mu_2.clear_{teammate(i)} + C_{ES(i)}, i{\neq}robotid \tag{2}$$

$$C_{auctioneer} = C_{ES(robotid)} \tag{3}$$

$$C_{defender} = \mu_5.t_{dist} + \mu_6.t_{align} + \mu_7.clear_{defense} \tag{4}$$

where robotid is the id of the robot, t_{dist} is the time required to move for specified distance, t_{align} is the time required to align for specified amount, μ_i are the weights of

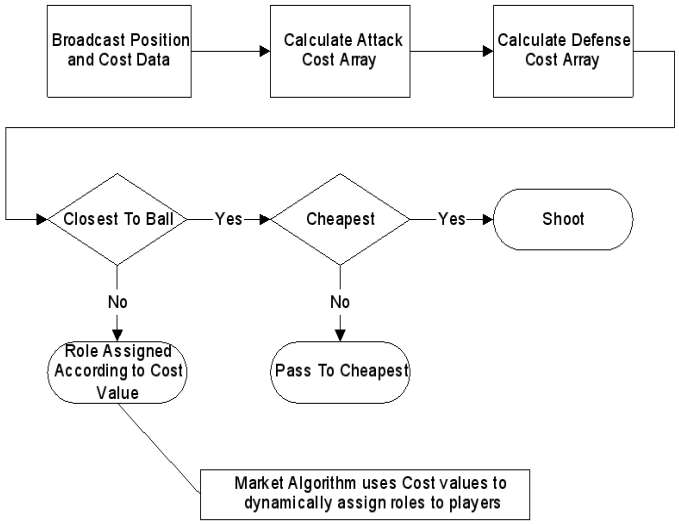

Figure 2. Flow chart for task assignment

several parameters to emphasize their relative importance in the total cost function, $clear_{goal}$ is the clearance from the robot to goal area-for object avoidance, $clear_{ball}$ is the clearance from the robot to ball-for object avoidance, $clear_{defense}$ is the clearance from the robot to the middle point on the line between the middle point of own goal and the ball-for object avoidance, and similarly $clear_{teammate(i)}$ is the clearence from the robot to the position of a teammate. Each robot should know its teammates score and defence costs. In our study each agent broadcasts its score and defence costs. Since the auctioneer knows the positions of its teammates, it can calculate the $C_{bidder(i \neq robotid)}$ value for its teammates.

The game strategy can easily be changed by changing the cost functions in order to define the relative importance of defensive behavior over offensive behavior, and this yields greater flexibility in planning, which is not possible with the current algorithms.

4 Proposed Approach

In the proposed approach in this study, RL implementation replaces the role assignment in the original market algorithm mentioned above, with a Q(λ)–Learner. CMAC is used for function approximation and state generalization. Although the market algorithm dynamically assigns roles to the players, the roles are assigned only to one player. RL implementation queries the action set and assigns the best action to the agent, thus enables multiple agents acting in the same role at the same time.

4.1 Market QL Team Setup

The Market algorithm defines the separation between active and passive players by checking the closeness of the player to the ball. The first player always fills the goalie role so four of the players are actively learning. But for each simulator step, since the

closest player still acts the same as in the original algorithm, there are at most three players using RL to decide which role to fill. The roles are as follows:

- Attacker,
- Secondary attacker,
- Defender,
- Secondary defender.

State representation consists of perceptual and logical parameters. The perceptual parameters are relative distance to the ball, two goals and other players (4 teammates and 5 opponents in this case). Each relative distance variable is composed of two parameters which are distance r and angle θ between the normal line and the agent. The logical parameters are the cost values (4 players' offensive and defensive cost values) and the closest player to the ball. There are 24 perceptual parameters and 9 logical parameters so totally 33 parameters are used to construct state vector.

Since the nature of the reinforcement learning is adaptive, the converged policy is very related to the opponent team chosen for training. Moderate level team selection gives best performance results among different alternatives. The details of the market team setup and training procedure can be found in Tatlıdede [9].

5 Tests and Results

The proposed approach in Figure 2 is implemented on Teambots, a realistic multi-robot simulator designed for multi-agent applications by Balch [10]. In the tests the robots are controlled in a distributed manner. Notice that while keeping the general market-driven idea, various kinds of game strategies, tasks and scenarios can be developed. In this work, one of these is implemented to show the power of the method.

For locomotion, a potential field based method which is trained with genetic algorithms is used to get a smooth movement [11]. In order to form a basis for comparison first the 'MarketTeam' [4](team of robots controlled by market-driven strategy) played against 30 games against each team. The results are tabulated in Table 1.

Table 1. Opponent team descriptions and match results with original MarketTeam

Opponent Team Name	W	T	L	Own scores	Opponent scores
AIKHomoG	20	5	5	51	22
RIYTeam	20	7	3	51	17

Among these teams, RIYTeam [12] has a simple bidding mechanism for task allocation based on the distance to ball, and uses a potential field based method which is trained by genetic algorithms as MarketTeam to move after tasks are allocated.

AIKHomoG is one of the best teams in Teambots that uses a complex but static game strategy.

In the test matches, MarketTeam was faster relative to the similar teams with complex behaviors due to its simpler structure, and less costly communication scheme. The game plan would change simply by changing the cost functions in order to define relative importance of defensive behavior over offensive behavior, and this gives a great flexibility in planning, which is not achievable with the current algorithms. If the winner decides that it can not shoot the ball to goal since its front view is blocked by opponent defenders, then it passes the ball to the team member at the best position. In figures 3 and 4, the team member takes the ball from winner and shoots it to the goal which will possibly ended with a goal.

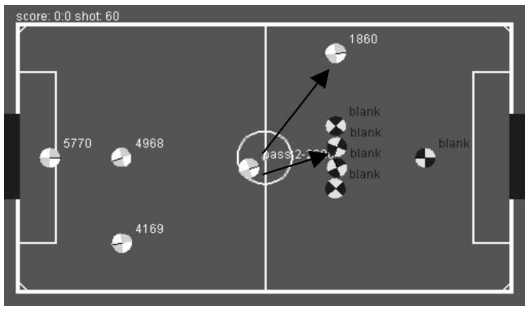

Figure 3. Auctioneer passes the ball

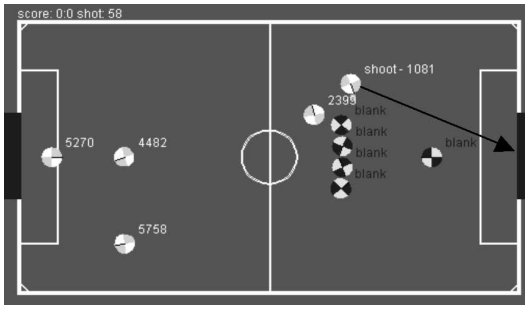

Figure 4. The new winner shoots the ball to goal

5.1 Learning Dynamic Role Assignment

The learning task in this work is learning dynamic role assignment for a market based team. The Market team developed for the previous tests is used as the core of the new team. The goalie and attacker are assigned to their roles by the same algorithm but the passive players' roles, which are DEFENDER, SECOND and THIRD, are now determined by Q(λ)-Learner. Besides the default three actions used by MarketTeam,

Table 2. Opponent team descriptions and match results with new MarketTeam

Opponent Team Name	W	T	L	Own scores	Opponent scores
AIKHomoG	23	4	3	81	30
RIYTeam	24	4	3	82	19

ATTACKER role is added to the action set. Relative distances to other players, ball and two goals are perceptual state variables for the agent. Distance parameter is parametrically quantized to generate CLOSE, NEAR, FAR and TOO FAR values. The angle between normal and the distance vector is quantized as its quadrant value. Defensive and offensive cost values are logical state variables, which are supplied by MarketTeam cost calculation engine.

Although MarketTeam's role assignment is dynamic, the roles are static. By using RL, MarketQL's policy enables multiple players being assigned to the same role, this fact increases the performance (Table 2).

In the multi-agent domains, especially soccer domain, team heterogeneity due to different roles increases the overall performance [13]. MarketTeam uses a hybrid solution that assigns a goalie by player number and assigns the rest of the roles according to cost functions. Although the role assignment is dynamic, this approach still lacks of flexibility where flexibility is having more than one player assigned to the same role. Reinforcement Learning implementation queries the action set in perceived state and assigns resultant action (in our case, the appropriate role) to the player, hence allows multiple players assigned to the same role.

In the market algorithm if the active player does not have the cheapest cost to shoot, it passes the ball to the player with the cheapest cost, which is generally the secondary attacker in our case. While our team is attacking, RL decides to assign two players as secondary attackers, which increases the possibility of controlling the ball when the active player passes the ball.

6 Conclusion

The domain of multi-agent applications is a very challenging research area, which allows more robust, flexible, and faster solutions with less cost, more useful in areas such as industrial (e.g agriculture), military applications (e.g mine sweeping), planetary explorations, etc., where single robot or human solutions are either inefficient or environment is hazardous. Robot soccer is one of the new test beds for multi-agent domain, with its highly complex, dynamic and noisy nature bringing more limitations and complexity. So, contrary to the domains with static environment no real-time requirement, solutions for such a domain are not trivial, in general. The proposed approach is introduced to this domain for the first time in this work.

In our test domain, there are only five players and one of them has a statically assigned role (the goal keeper), the remaining four robots are assigned to appropriate behavioral roles according to the sitiuation of the game. The distributed nature of the approach avoids single point failures of centralized approaches and bring robustness. Even if some of the robots are injured or lost, the team continues to work, and the assigned goals of lost robots are handled and accomplished by the remaining robots, eventually.

Besides the simplicity of cost calculation, communication and task allocation allowed fast playing which is a must for real-time games. With better alignment, and shooting (with even a perfect plan, it is not trivial to make the robots do what they should, they can not align and shoot to the place they should perfectly, this is even worse in real robots so locomotion is a bottle-neck to test the algorithms) and new tasks to use for defenders, the success would be improved. Dynamic task allocation makes the approach adaptive, robust and suitable for real-time events, where team always has a plan B according to the situation.

Embedding of Q(λ)-Learning as the policy learning algorithm, brought a more successful behavior assignment policy after a set of training games have been played. As it can be seen from the experimental results, the team with learned strategy performs better than the original purely market-driven team since it has learned to assign behaviors adaptively. For example, instead of assigning fixed number of roles to the players such as one defender, one primary attacker, one secondary attacker and one tertiary attacker, the team learned to assign a second defender to the tertiary attacker when the team needs a more defensive behavior.

The main possible disadvantage of market-driven task allocation for robotic soccer domain is the time requirement for the auctioning and utility calculation processes. Since some form of utility calculation is necessary in nearly all multi-agent task allocation problems, optimization of utility calculation is independent from market-driven approach. Time requirement constraints for auctioning can be satisfied by setting appropriate auction duration and time out for auctioneer values.

One of the advantages of the market-driven approach is that, communication burden is reduced by making only neighbor robots to communicate among them. The disadvantage that this limited communication might be that, system would converge on a suboptimal solution since the entire team is not communicating with each other. But this is not the case for soccer domain since it has currently a limited number of players (e.g. not more than five players) which do not cause communication burden. Thus it does not suffer from suboptimal solutions.

As a conclusion, there are still many open issues in multi-agent approach to robotic soccer domain and market-driven approach provides a robust and efficient planning and resource sharing. As a future work, usage of the market-driven idea in multi-agent localization, and perception would be done.

7 Acknowledgement

This project is supported by the Boğaziçi University Foundation and by Boğaziçi University Research Fund project 03A101D.

References

1. M. B. Dias, and A. Stenz, *A Market Approach to Multi-robot Coordination,* CMU-RI-TR-01-26, August. (2001)
2. R. Zlot, A. Stenz, M. B. Dias, and S. Thayer, Multi-Robot Exploration Controlled by a Market Economy, Proceedings of the *IEEE International Conference on Robotics and Automation*, May. (2002)
3. P.B. Gerkey, and M.J. Mataric, Sold!: Auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation, special issue on Advances in Multi-Robot Systems,* **18(5)**, 758–786.(2002)
4. H. Köse, K. Kaplan, Ç. Meriçli and H.L. Akın, Genetic algorithms based market-driven multi-agent collaboration in the robot-soccer domain. *FIRA Robot World Congress 2003.*(2003)
5. H. Köse, Ç. Meriçli, K. Kaplan, and H.L. Akın, All bids for one and one does for all: Market-driven multi-agent collaboration in robot soccer domain. *Computer and Information Sciences-ISCIS 2003, 18th International Symposium Antalya, Turkey, November 2003 Proceedings* **LNCS 2869**, 529–536.(2003)
6. J.Peng, and Williams R. Incremental multistep qlearning. *Machine Learning* pp. 283–290.(1996)
7. R. S. Sutton, Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in Neural Information Processing Systems MIT Press.*(1996)
8. J.S. Albus, Data storage in the cerebellar model articulation controller (CMAC). *Transactions of the ASME: Journal of Dynamic Systems, Measurement, and Control* pp.228–233.(1975)
9. U. Tatlıdede, *Reinforcement Learning of Multiagent Team Behaviour,* Master Thesis, Boğaziçi University, July. (2003)
10. T. Balch, Teambots, *http://www.cs.cmu.edu/ trb/TeamBots/Domains/SoccerBots,* (2000)
11. K. Kaplan, *Design and Implementation of fast controllers for Mobile Robots,* Master Thesis, Boğaziçi University, January. (2003)
12. RIY *http://robot.cmpe.boun.edu.tr/riy/ riy.php.* (2003)
13. T. Balch, and L. Parker *Robot Teams: From Diversity to Polymorphism.* A. K. Peters.(2002)