A Reward Function Generation Method Using Genetic Algorithms: A Robot Soccer Case Study

(Extended Abstract)

Çetin Meriçli
Computer Engineering Department
Boğaziçi University, Turkey
cetin.mericli@boun.edu.tr

Tekin Meriçli
Computer Engineering Department
Boğaziçi University, Turkey
tekin.mericli@boun.edu.tr

H. Levent Akın
Computer Engineering Department
Boğaziçi University, Turkey
akin@boun.edu.tr

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Robotics

General Terms

Algorithms

ABSTRACT

Immediate rewards play a key role in a reinforcement learning (RL) scenario as they help the system deal with the credit assignment problem. Therefore, reward function definition has a drastic effect on both how fast the system learns and to what policy it converges. It becomes even more important in case of multi-agent learning, where the state space usually gets even bigger. We propose a Genetic Algorithms (GA) based reward function shaping method for multi-robot learning problems and evaluate its performance in a robot soccer case study. A set of metrics calculated from the positions of the players and the ball on the field are used as the primitive building blocks of an immediate reward function, which is defined as a weighted combination of these metrics obtained using GA, yielding a significantly better soccer playing performance.

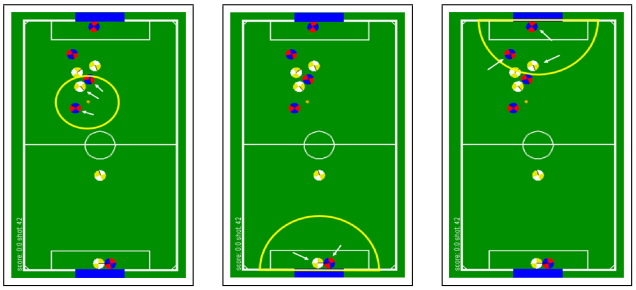
Keywords

Multi-robot systems, evolutionary algorithms, multi-agent learning, robot soccer, RoboCup

1. APPROACH

The environment in RL is usually considered as a Markov Decision Process (MDP) and the RL problem itself is formally defined as a tuple $< S, A, R, \pi >$ where, S is a set of states, A is a set of actions, R is a set of scalar rewards in $\mathbb{R}$, and π is a policy such that $\pi : S \rightarrow A$. The aim is to find a policy π which maximizes the cumulative reward [1].

We used the discrete state and action representations presented in [2]. The field is divided into regions for defense, mid-field, and forward partitions of right and left wings, yielding 6 regions on the field. We also used the vicinity occupancy metrics in state representation along with the relative distances of each player to the defined strategic points on the field.



(a) (b) (c)

Figure 1: Vicinity occupancy for (a) the ball, (b) the own goal, and (c) the opponent goal.

Most actions required to play soccer have two primitive behaviors in common: moving towards a point and moving away from a point. We modeled the actions using "aggression" and "fear" behaviors of Braitenberg vehicles [3], and defined five different actions; namely, attacking, supporting the attacker, defending, passing to the closest teammate, and passing to the teammate closest to the opponent goal.

We used a set of metrics calculated using instantaneous positions of the players and the ball [4, 5].

Vicinity occupancy is the ratio of the number of our players to the number of opponent players within a vicinity of a particular point of interest. Vicinity Occupancy is calculated for the ball, the own goal area, and the opponent goal area (Figure 1).

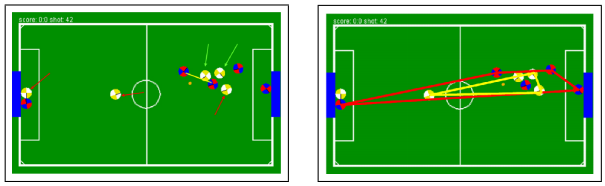
Pairwise separation is aimed to measure the degree of separation of an object of interest from the opponent team (Equation 1).

$$S_{obj} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m} separates(P^i_{own}, P^j_{own}, P^k_{opp}, obj)}{2} \quad (1)$$

The *separates* function is defined as

$$separates(P_1, P_2, P_3, obj) = \begin{cases} 1 & \text{if } \overleftrightarrow{P_1, P_2} \text{ intersects } \overleftrightarrow{P_3, obj}, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Cite as: A Reward Function Generation Method Using Genetic Algorithms: A Robot Soccer Case Study (Extended Abstract), Author(s), *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. XXX-XXX.



(a) (b)

Figure 2: a) Pairwise separation of ball from the opponent team: robots pointed with light arrows are separated from the ball, and (b) convex hulls of two teams.

where n is the number of own players, m is the number of opponent players, and P_{own} and P_{opp} are the sets of own and opponent players, respectively (Figure 2 (a)).

Convex Hull of a set of points is defined as the smallest convex polygon in which all of the points in the set lie. By substituting points with the players on the soccer field, we obtain the convex hull of a team. The area of convex hull for a team tends to measure the degree of spread of the team over the field. The value of this metric increases as the team members are scattered across the field (Figure 2 (b)).

Ball possession discretizes the ball possession information in the last two timesteps as *Own ball*, *Opp ball*, *Ball gained*, *Ball lost*, *Both have the ball*, and *None has the ball*.

For delayed rewards, a reward of 1 or -1 is received for own and opponent scores, respectively. A weighted combination of the instantaneous metrics measuring vicinity occupancy and separation for the own goal, the opponent goal, and the ball, areas of the convex hulls for our team and the opponent team, and the ball possession is used as the immediate reward signal.

Table 1: Score rates before and after training.

	Equal Weights	Learned Weights
Null Team	19.444	28.245
Brian Team	6.376	17.796
Kechze	8.75	12.666
SibHeteroG	-52.5	-4.900
Total:	-17.93	53.807

2. RESULTS AND CONCLUSIONS

Experiments were run on the TeamBots simulation environment [6]. A team of five players was considered. Only the field players were trained and a hand-coded goalie was used.

The weights for the metrics are encoded as a chromosome in a Genetic Algorithms (GA) setup. The calculation of the fitness value for a chromosome consists of two stages. A game of length 600 simulation steps is played against a moderate opponent using the current chromosome. Then, an evaluation game of length 200 is played and the resulting score rate (Equation 3) is used as the fitness value, where SR is the score rate, s_{own} is the own score, and s_{opp} is the opponent score. Real number chromosomes and roulette wheel selection policy were used in the GA system with population size 30 for 10 generations with crossover probability 0.9, mutation probability 0.2, and random selection probability 0.1.

$$SR = \begin{cases} s_{own}\frac{(s_{own}-s_{opp})}{(s_{own}+s_{opp})} & \text{if } s_{own} > s_{opp}, \\ s_{opp}\frac{(s_{own}-s_{opp})}{(s_{own}+s_{opp})} & \text{if } s_{own} < s_{opp}, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The immediate reward function using the best weight vector obtained at the end of training (Table 2) is compared against an immediate reward function where all weights are set as 1.0 (Table 1).

Table 2: Learned weights for metrics.

Metric Name	Learned weight
ownGoalVicinityOccupancy	3.745
ballSeparationOwn	2.116
oppGoalSeparationOpp	-1.604
oppGoalVicinityOccupancy	1.266
ballVicinityOccupancy	1.124
ownAreaOfConvexHull	0.910
ownGoalSeparationOpp	-0.748
ballPossession	0.634
oppAreaOfConvexHull	-0.425
ownGoalSeparationOwn	0.336
oppGoalSeparationOwn	-0.199
ballSeparationOpp	0.039

It can be deduced that the importance of a metric is proportional to the absolute value of its learned weight. Experiments show that the learned weighted combination of the defined metrics results in a significantly better soccer playing performance. Using Evolutionary Strategy (ES) as the optimization engine and trying to form completely new metrics out of only the position information of the objects on the field are among possible future extensions.

3. ACKNOWLEDGMENTS

This work is supported by The Scientific and Technological Research Council of Turkey project number 106E172 and Boğaziçi University Scientific Research Projects grant number 09M105.

4. REFERENCES

[1] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[2] Tekin Meriçli and H. Levent Akin. Soccer without intelligence. In *ROBIO09: Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics*, pages 2079–2084, Washington, DC, USA, 2009. IEEE Computer Society.

[3] Valentino Braitenberg. *Vehicles: Experiments in Synthetic Psychology*. MIT Press / Bradford Books, 1984.

[4] Çetin Meriçli. Developing a novel robust multi-agent task allocation algorithm for four-legged robot soccer domain. Master's thesis, Boğaziçi University, 2005.

[5] Çetin Meriçli and H. Levent Akın. A layered metric definition and validation framework for multirobot systems. In *RoboCup*, 2008. (to appear).

[6] Tucker Balch. *TeamBots Mobile Robot Simulator*. http://www.teambots.org, 2000.