# Genetic Based Modeling of a Multi-Agent Environment Using OWA Operator

A. Nayyeri, M. Yabandeh, H. Mousavi, and C. Lucas

***Abstract*—In this paper, we try to explain our approach to the problem of decision making in a multi-agent, non-deterministic environment, like soccer field. Our strategies have been tested in Robocup competitions, and could register important accomplishments. After reducing the problem in terms of the well-known problem of finding the best coefficients for computing the correctness of an action, we've used three different methods for calculation of the overall fitness, with respect environmental parameters. These methods are: getting simple weighted average with fixed weights, using GA to find the weights of the parameters, and finally using OWA operator instead of weighted averaging with GA. Due to the huge size of the problem we had to use techniques like bounded rationality and memorized estimation to reduce the amount of computation.**

***Index Terms*—Bounded rationality, genetic algorithm, memorized estimation, OWA operator, robocup.**

## I. Introduction

DYNAMIC, noisy, continuous, and undecidable environment with partial information of soccer playing make it a good sample of a very complex multi agent system for testing and developing AI algorithms. Soccer simulation is a category of Robocup competitions, which tries to simulate the soccer playing environment; keeping its properties as much as possible.

A soccer server is developed to provide the environment and handle different events of the game. Also there is a program for each player that simulates the existence of him by sending and receiving messages to and from the server. Each player receives some local and noisy information about the objects in the field from the server periodically. This information is organized due to the player position and view direction. The farther distance between the object and player, the more noise affected on the objects information sent to the player.

Each game consists of 6000 time cycles. Each player has the chance to choose a simple low level action, such as kick, turn or dash, and send it to the server during the cycle period. So the time of making decision is restricted. The server then, simulates the situation of the field. Simulation of the next position and velocity of an object influenced by some noise; causes the nature of the game not decidable. More information about the soccer simulation could be found in Robocup Soccer Server Manual, 2002.

To make the problem well defined in order to have a good measurement out of quality of the applied methods different goals have been defined so far. The amount of time that a team can possess the control of the ball is one of such assessments introduced by [8]. In this approach which is very similar to one of well-known exercises of real soccer, the players of one team tries to keep the control of the ball in a predetermined area, while opponents struggle to intersects their passes. Other works tried to apply Reinforcement Learning to improve shooting in robotic soccer [9]. Also [10] has used Monte Carlo learning approach to learn a full team of agent to shoot and pass properly. Learning approaches has been applied to update player positions based on the movement of the ball in [11]. Moreover, [12] used learning methods to make players with high quality low level skills such as passing, ball intersection, and dribbling.

Overall, the duties of a player could be divided to keeping a world model of the objects, which were seen, choosing a strategic position for standing on it, choosing a high level action for doing with ball such as dribble or pass and how to do a high level action, using low level ones. One of these high level actions is passing the ball. It seems it is essential for a good soccer team to have players who can pass the ball well between themselves. So, in this article we speak about how to choose the best point to pass the ball. One problem doing such a thing is to define what a good pass is, and how should a good pass be recognized from a bad one.

The problem will be explained more exactly in section two, Problem formulation. Section three is devoted to explaining our strategies and analyzing and comparing them. Section four is the statistical result of our experiences. Conclusion and future work will be the last section of this paper.

## II. Problem Formulation

As mentioned in the previous section, the problem is finding the best point to pass the ball for the ball holder. It is worth noting that we consider best point to pass instead of best player. The kind of looking at the problem helps to include all kinds of pass such as forward pass to a besieged teammate and etc.

One primitive difficulty is the formulation of the problem, or simply finding a suitable definition for the quality of a pass to a point. As it seems obvious there are different parameters which play important rolls in determining the value of a pass. For example a pass could be considered good if it provides the receiver a good chance for goal getting. Obtaining a single value of these parameters is the main problem.

At first glance the effectiveness of the pass to every points of the field should be obtained and these values are compared and finally the best will be chosen as the target

Manuscript received August 2, 2006; revised March 10, 2007.

The authors are with the ECE Department, University of Tehran, Tehran, Iran (e-mail: a.nayyeri@ece.ut.ac.ir, m.yabandeh@ece.ut.ac.ir, h.mousavi@ece.ut.ac.ir, lucas@ipm.ir).

Publisher Item Identifier S 1682-0053(07)1544

1682-0053/07$10 © 2007 JD

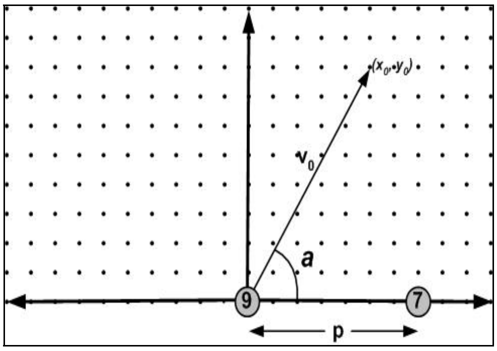


Fig. 1. Player number 7 wants to intercept the ball that is thrown from the point 0, 0 with velocity $v_0$, to $x_0$, $y_0$.

for the pass. The other matter is the huge size of the problem. The player who has the ball control can kick it to every points of the field; the domain of the problem is not finite. Also after digitizing the fields the size of necessary computation can be very large. On the other hand it is real time decision making, it means only on time actions could be effective. Our approach to these problems is discussed in detail in the following subsections

### *A. Parameters*

As stated previously, different parameters are considered to define the value of a pass. Here is the list of the parameters. Note that all of them will be normalized:

- Catch chance: It is a compare between the time that our teammate needs to get the ball, and the nearest opponent needs to do so. In fact it determines that in a race for obtaining the ball how much we can be sure that our teammate will be the winner.

$$CatchChance_i = \frac{t^{teammate_i}_{INTERSCTION}}{\min\limits_{j=1..11} t^{opponent_j}_{INTERSECTION}}$$

- Free space: It is related to the number of opponents around the pass point. It seems better to send the ball to a point where the density of opponent's players is low. Then the pass receiver will have more time to decide about his next action.

$$FreeSpace_i = Numbert^{opponents}_{Circle(passPo\text{int},R)}$$

- Goal chance: It relates to the chance of goal getting for the receiver of the ball; the chance of entering the ball to the goal with a direct kick toward the opponent goal.

$$GoalChance_i = \frac{\sum SafeShootingAngles}{GoalViewAngle}$$

- The distance from the Y vector and its complement: It is to consider if moving the game to the center of field is more useful or sending it to the sides. Coefficients of these two parameters will be tuned to determine the important one based on the position and roll of the receiver.
- The distance to opponent goal and its complement. Similar to two above parameters, these ones different between the pass to forward and backward.

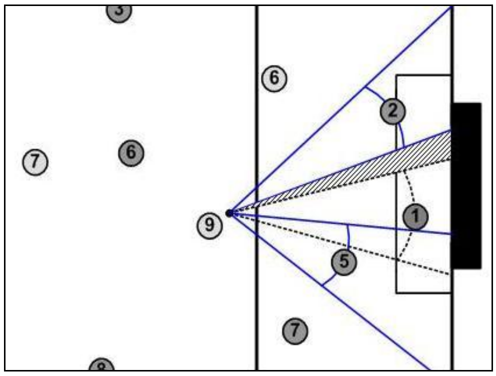


Fig. 2. The shadowing area is secure for shooting. Nor number 1 neither number 2 is capable of intersecting ball before it enters the goal if number 9 shoots the ball through secure angles.

### *B. Difficulties of Computing Parameters*

There were different kinds of complexities in computing of the parameters. The way of computing two of them are explained here and their difficulties are debated.

The first one is the catch chance. This parameter should represent the probability of successful receiving of the ball by the teammate. The difference between the time that our teammate needs to get the ball and the shortest needed time for opponents to catch it can represent this parameter after normalization.

As the environment is noisy, it is not easy to measure the required time to intercept the ball. The noise of the system affects the ball speed, receiver movement and also the kick action of the sender. One recommendation is to train a system to estimate the needed time. For example a neural network can do so.

***Memorized Estimation*** - We have solved this problem differently. We've tried to compute the solution for all possible inputs offline and store them in the secondary storage. They are loaded to the memory at the beginning of the game. This helps to overcome to the problem of decision making in real time.

Consider 300 discrete candidate points on a coordinate plane. The ball and a player have been placed on the center. The player kicks the ball to each candidates point k times. At each time a player placed in the <P, 0> coordinates try to intercept the ball. The average of the k times that takes for receiver to achieve this goal will be recorded as the required value. This task is done for each integer value of P. At the beginning of each game these values are loaded to memory and could easily be used during the game without any additional computation.

The second parameter that its computing is described here is the goal chance. For measuring this parameter we simply compute the maximum safe area from the final point of standing of the receiver for a straight kicking to goal.

Each opponent in front of the kicker makes some area about itself dangerous depends on his distance to the kicker, meaning that kicking to that area will be intersected by the opponent with a high probability. Existence of multiple opponents provides multiple discrete angles safe for the player to kick to the goal. It is obvious that the bigger one is desired.

The problem is time-consuming nature of computing these dangerous angles. We used a similar solution using

Memorized Estimation technique. Putting a player in front of a kicker with distance $d$ we let the kicker kick with different angles respect to the opponent. The maximum angle that opponent is succeed to intersect the ball will be recorded as dangerous angle of distance $d$ in secondary storage and will be used during game without any time-consuming computation.

### *C. Bounded Rationality, Considering Only Some Points of the Field*

"Bounded Rationality is the property of an agent that behaves in a manner that is nearly optimal with respects to its goals as its resources will allow." - H. A. Simon. We have used bounded rationality to decrease the amount of computation.

Because it is not possible to compute the combination functions for all points of the field in a time slice of thinking and acting, we had to decrease the number of candidate points for computation. One possible solution for such a problem is to consider the field as a discrete plane, and measure the function only for points with integer coordinates. Also as a matter of fact some points are far and irrational enough to be omitted from search list.

## III. How to Solve the Problem

Having pass parameters ready, the question is how to obtain a single value that represents the quality of the pass. Our first solution was using weighted average of the parameters. In this manner the coefficient of each parameter is showing the importance of it for us. Determining these coefficients by hand was the first approach that will be mentioned by simple approach but the result of this solution was not satisfactory.

Consider some snapshots of the game are presented; each shows the start of a pass. Each frame has adequate information for calculating parameters. And also suppose that we have the value of these passes. So our main problem changes to the problem of finding coefficients that produce most similar result to our sample ones. In other words we want to minimize the square error between the gaining values and our appropriate answers. Therefore the problem is reduced to the problem of finding minimum. Different search methods are today exist to solve this problem. Now we can use the known Genetic Algorithm to minimize a value.

The best reference to judge about the score of a pass is the receiver of it after it has been completed. In this time the receiver could be sure about the successful receiving of the pass, the time that it has to choose his high level next action before nearing of the opponent and the chance of getting goal with a direct kick. The receiver produces a score with respects to these parameters using some constant coefficients that have been determined experimentally. Actually to collect these data we let numerous games to be performed. For each successful pass in the game the pass frame plus the score given to it have been saved. We call them together a test frame.

The used GA method is represented in more detail here:

***Representation*** – Each chromosome is simply a list of all the coefficients.

***Genetic Operators*** – We have used ordinary mutation and crossover for producing next generations.

***Fitness Function*** – As mentioned before for computing the fitness of a chromosome it is tested with different collected test frames and then its square error will be computed. At the last part a minus is multiplied to it make the chromosome with less error higher in fitness. So if we show the expected value of the pass in the test frame $i$ with $E_i$ and the value computed by the chromosome with $V_i$ then the total fitness of the chromosome is:

$$\frac{((V_1 - F_1)^2 + (V_2 - F_2)^2 + \dots + (V_f - F_f)^2)}{f}$$

### *A. Applying OWA Fitness Function to the GA*

Some times it seems that the importance of a parameter is related to its size. May be a low parameter should be considered more in final decision making than a normal parameter, in some cases. One reason behind such an approach is that a parameter is more critical in its boundary values, and we should pay much more attention to it. For example if a pass produces medium chance for goal getting, medium move to forward, also places the receiver in a sparse condition, but the probability of interrupting the ball with the opponents' player is high the pass can not be considered as a good one. Also a pass with seldom-good parameters except that the parameter for the sparseness of opponents in the receiving point is too small may not be a good one. Thus we decided to apply OWA operator instead of weighted average.

Yager introduced a new technique, Ordered Weighted Averaging [1]. This operator has been discussed in different papers later. We've also used this operator instead of our weighted average here.

***Definition*** – A mapping F from $R^n \rightarrow R$ is called an OWA operator of dimension n, with an associated n vector

$$w = (w_1, w_2, \dots, w_n)^T$$

where all $w_i$s are normal, and

$$w_1 + w_2 + \dots + w_n = 1$$

Then

$$F(a_1, a_2, \dots, a_n) = w_1 b_1 + w_2 b_2 + \dots + w_n b_n$$

where $b_j$ is the $j$-th greatest element in $(a_1, a_2, \dots, a_n)$.

### *B. Fitness Function Refinement*

After computing the fitness function with the above methodologies, we face a new difficulty. It is that the fitness values are very close to each other. On the other hand we've used proportionate selection for producing new generations. In this case the GA will converge very slowly, since higher valued chromosomes are not given much more opportunity for production. Therefore the population will be slowly replaced with higher fitted people, and if the values are very similar, may be it won't converge in real time.

There exist diverse solutions to overcome this trouble. One of them is to change the methodology of selection. For example Baker ranking selection, E. J. Baker (1985) can be used. The other is to change the fitness function to increase the distance among people; scaling fitness. The latter can be performed by multiplying a number to all the fitness

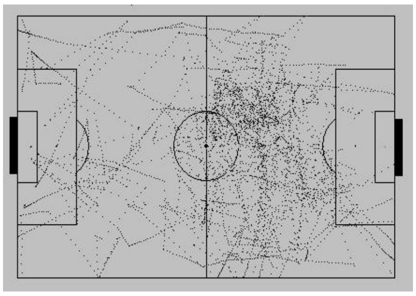

Fig. 3. The ball distribution for the game in which the parameters tuned experimentally.

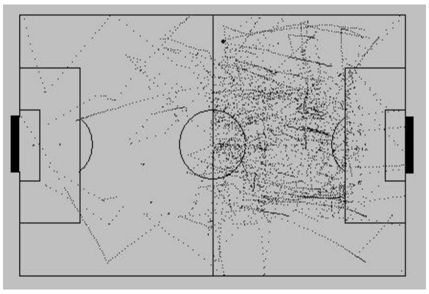

Fig. 4. The ball distribution for the game in which the parameters tuned using GA, with weighted average.

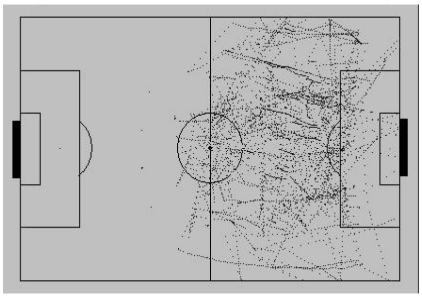

Fig. 5. The ball distribution for the game in which the parameters tuned using GA, with OWA operator.

values or squaring all of them. We've multiplied all of them with two. Overall, these solutions help the individuals with higher fitness values to converge faster.

## IV. STATISTICAL RESULTS

For making a comparison among the approaches, and to recognize the effect of the algorithms, statistics can be used. The collected statistics are divided into two categories, statistics of the ball distribution during the game, and actual results of the game. These statistics are followed:

### A. Game Results

The most primitive way for comparison of the algorithms is to compare the results of them. For gathering the necessary information, many games should be held. The team had to play in front of a fixed opponent with each of the coefficients. For the sake of accuracy each game was performed ten times. Overall results are shown in Table I.

### B. Ball Distribution

We also compute the ball position distribution during three games and compare them with each other. The figures are illustrating the case. In all of the pictures our team was playing at the left side of the field.

TABLE I

STATISTICS OF THE GAMES' RESULTS. THE NUMBERS REPRESENT OUR TEAM GOAL NUMBERS VERSUS THE OPPONENT'S.

| | Best Result | Worst Result | Average Result |
|---|---|---|---|
| Simple Approach | 1-0 | 0-1 | 0.37-0.5 |
| GA Mean Weight | 12-0 | 3-0 | 6.0-0.1 |
| GA OWA | 13-0 | 2-0 | 6.7-0.1 |

## V. CONCLUSION AND FUTURE WORK

It is evident in the statistics that using genetic algorithm improves the overall results. Using of the OWA operator also has a great impact on the outcome. The reason can be the adaptability of the OWA operator nature and the soccer environment for deciding. Bounded rationality and memorized estimation also helped us to reduce the amount of computation. Also some verifications of fitness function can improve the convergence time of the GA.

We think the most obvious shortcoming of our team is the confidence of knowledge base. We think the GA performs perfectly in a known environment, so it will work much better if we can build a more confident environment model. We use GA offline because of its high computation load. Tuning coefficient online to adapt in front of current opponent could improve the power of the team considerably.

## ACKNOWLEDGMENT

This research was performed for participation in Robocup Competitions Fokuoka 2002 and Padova 2003 as the Thunder team. We would like to thank other team members, Dr. Oroumchian (Team's Coach), Mr. Ehsan Khosrowshahi Asl, and Mr. Saeed Amizadeh. A preliminary version of this paper was presented in WAC 2004 conference.

## REFERENCES

[1] R. R. Yager, "On ordered weighted averaging aggregation operators in multi-criteria decision making," *IEEE Trans. on Systems, Man and Cybernetics,* vol. 18, no. 1, pp. 183-190, Jan./Feb. 1988.

[2] R. R. Yager, "Families of OWA operators," *Fuzzy Sets and Systems*, vol. 59, no. 3, pp. 125-143, Aug. 1993.

[3] J. C. Fodor and M. Roubens, *Fuzzy Preference Modelling and Multicriteria Decision Support*, Theory and Decision Library, series D: System Theory, Knowledge Engineering and Problem Solving, vol. 14, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[4] F. Herrera, E. Herrera-Viedma, and J. L. Verdegay, "Direct approach processes in group decision making using linguistic OWA operators," *Fuzzy Sets and Systems*, vol. 79, no. 2, pp. 175-190, Apr. 1996.

[5] M. Chen, *et al.*, User Manual, Robocup Soccer Server, 2002.

[6] J. E. Baker, "Adaptive Selection Methods For Genetic Algorithm," *in Proc. of First Int. Conf. on Genetic Algorithms*, pp. 101-111, Hillsdale, New Jersey, US: Lawrence Erlbaum Associates, 1985.

[7] H. A. Simon, *Models of Bounded Rationality*, Cambridge Massachusetts: MIT Press, 1982.

[8] P. Stone, R. S. Sutton, and S. Singh, "Reinforcement learning for 3 vs. 2 keepaway," in *RoboCup2000: Robot Soccer World Cup IV*, P. Stone, T. Balch, and G.Kreatzschmarr, editors, pp. 249-258, Berlin: Springer-Verlag, 2001.

[9] E. Uchibe, *Cooperative Behavior Acquisition by Learning and Evolution in a Multi-Agent Environment for Mobile Robots*, PhD. Thesis, Osaka University, Jan. 1999.

[10] P. Stone and M. Veloso, "Team-partitioned, opaque-transition reinforcement learning," in *RoboCup-98: Robot Soccer World Cup II,* M. Asada and H. Kitano, editors, pp. 206-212, Berlin: Springer-Verlag, 1999.
[11] J. S. Albus, *Brains, Behaviour, and Robotics*, BYTE Books, Peterborough, New Hampshire, 1981.
[12] M. Riedmiller and A. Merke, "Karlsruhe brainstormers - a reinforcement learning approach to robotic soccer," in *RoboCup 2000: Robot Soccer World Cup IV*, Stone P., Balch T., and Kraetszchmar G., editors, pp. 435-440, Berlin: Springer-Verlag, 2001.
[13] H. Kitano, *et al.*, "The Robocup synthetic agent challenge 97," *RoboCup-97: Robot Soccer World Cup I*, pp. 62-73, London: Springer-Verlag, 1998.
[14] R. H. Crites and A. G. Barto, "Improving elevator performance using reinforcement learning," in *Advances in Neural Processing Systems 8*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, Cambridge: MIT Press, 1996..
[15] K. Kostiadis and H. Hu, "KaBaGeRL, generalisation and reinforcement learning for possession football," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Hawaii, US, 2001.

**A. Nayyeri** received his B.Sc. and M.Sc. degrees in Computer Engineering from University of Tehran, Tehran, Iran, in 2004. He has the experience of working as a Research Assistant in the Router Laboratory, University of Tehran. His main research interests include network algorithms, and distributed systems. He is also interested in distributed intelligent decision making**.**

**M. Yabandeh** first received the B.S. degree in Computer Science and Engineering from University of Tehran, Tehran, Iran, in 2005. Then, he received his M.S degree in the same university and working as a research assistant in the Router Laboratory, University of Tehran. His main research interests include distributed computing, computer networks, operating systems, and theoretical computer science.

**H. Mousavi** was born in Tehran, Iran, in 1982. He graduated in Computer Engineering from Tehran University, Iran, in 2004. He received the M.S. degree in Software Engineering from Sharif University of Technology, Iran, in 2006.

He achieved silver medal in Iranian National Olympiad of arithmetic in 1999. He served as a Research Assistant in Router Laboratory of University of Tehran from 2002 to 2005. He also served as a TA in both Sharif University of Technology and Tehran University since 2002 in courses such as, operating systems, advanced algorithms, software engineering, and automata theory & machine languages. His research interests include web search engines, sensor networks, operating systems, semantic web, software engineering concepts, and Algorithms.

**C. Lucas** first received his M.S. degree in Electrical Engineering from the Faculty of Engineering (Fanni) University of Tehran, Tehran, Iran in 1973 and the Ph.D. degree from the University of California, Berkeley, in 1976. He is a Professor, and a member (as well as the founder- Director) of Center of Excellence for Control and Intelligent Processing, Department of Electrical and Computer Engineering, University of Tehran, as well as a Researcher at the School of Cognitive Sciences (SCS), Institute for Studies in Theoretical Physics and Mathematics (IPM), Tehran, Iran. He has served as the Director of Intelligent Systems Research Faculty, IPM (1993-1997) and Chairman of the ECE Department at the University of Tehran (1986-1988).