A “Never-loose” Strategy to Play the Game of Tic-tac-toe

[1]Amarjeet Singh, [2]Kusum Deep
[1,2]Department of Mathematics
Indian Institute of Technology Roorkee
Roorkee-247667, Uttrakhand, India
amarjeetiitr@gmail.com, kusumdeep@gmail.com

[3]Atulya Nagar
[3]Department of Mathematics and Computer Science1
Liverpool Hope University
Liverpool, L16 9JD, UK.
[3]nagara@hope.ac.uk

Abstract — **In much of the literature available to solve the tic-tac-toe board game, the common approaches, such as, co-evolution, neural networks, evolutionary programming and genetic algorithm are used. In the present work we present a deterministic approach for playing tic-tac-toe game, in which 9 objective functions are defined to decide player’s best move. For choosing the best from the solutions generated, by combination of mutating zero as 1, certain axioms are defined. The beauty of this method lies in the fact that if the player decides a move with this method, he never loses the match whenever the player begins the game. We suspect that functions can be defined on the similar grounds for other existing board games and some of this work is in progress and will be reported elsewhere. Some implications on these lines have been made as recommendations in this paper.**

Keywords-Tic-tac-toe; multi-objective; game

I. INTRODUCTION

Playing a game is one kind of test-bed to test the intelligence and physical skills in humans. Board games test intelligence and sports games test physical athleticism. Tic-tac-toe is: a zero sum, two player, perfect information, board game. In general, game tree is used to decide the next move while playing board games. The complete tree of tic-tac-toe game has 986,409 nodes with 9! leaf nodes. Without symmetry, total number of possible tic-tac-toe games are 255,168 [1] in which, when ‘X’ makes the first move, X wins 131,184 games, looses 77,904 games and draws for 46,080 games. However, in symmetry cases, i.e. after rotating or reflecting one game as a copy of the other, total possible games are 26,830 [2,3]. Based on game tree a program can be designed. But for the games like Chess, Checker, Go, complete tree cannot be constructed because of their high branch factors. In these games, partial game trees are constructed in which the value of the leaf nodes are evaluated by an evaluation function, which consists of board game’s features and search algorithms, min-max or alpha-beta, which are used to decide best moves.

The complete game tree consists of an initial position to all possible moves made from each position till the game ends. The game tree for tic-tac-toe is easily searchable, but the complete game tree for larger games, like chess are too large to search. Instead, a chess playing program searches a partial game tree. Typically, as many plies from the current position as it can search i.e. increasing the search depth, improves the chances of picking the best move. This was shown by Al-Khateeb and Kendall [4,5] through experiments but increasing the plies of game tree also increases the complexity and computational time.

In this paper we propose an alternate scheme that gets rid of ply concept and the solutions are generated based on available choices of moves and the best move is then selected through the fitness of solution i.e. the move that is evaluated by a defined objective function.

The paper is organized as follows. Previous works to generate a fast and efficient tic-tac-toe playing algorithm are discussed in section II. Analysis, mathematical model and procedure of generating solutions are described in section III. In section IV, a game is played between player and opponent. Conclusions are drawn in section V.

II. PREVIOUS EFFORTS IN TIC-TAC-TOE

Researchers are working in the arena of tic-tac-toe board game in order to generate a fast and efficient tic-tac-toe playing algorithm.

Fogel [6] used neural networks, a multi-layer feed forward perceptron, to achieve a high level of play in the game of Tic-tac-toe and used evolutionary programming for adapting the structure and weights of neural network. Huchmuth [7] demonstrated how to evolve a perfect tic-tac-toe strategy which never losses a game it plays through genetic algorithm. In [8], genetic algorithm is used to evolve a number of no-loss strategies and compare these strategies with existing methodologies. Ling and Lam [9] proposed an algorithm to play tic-tac-toe and this algorithm is learned by a neural network with double transfer functions, which is trained by genetic algorithm. Mohammadi [10] evolve tic-tac-toe playing algorithms using co-evolution, interactive fitness and genetic programming.

978-1-4673-6751-6/14 $31.00 © 2014 IEEE
DOI 10.1109/ISCMI.2014.13

Most of the work in the realm of the tic-tac-toe games use co-evolution and neural networks. In this paper, a mathematical formulation is presented which models how a player decides to place 'X' in tic-tac-toe board. In this, tic-tac-toe problem is considered as a multidimensional, multi-objective problem. The mathematical model is defined for taking decisions and 4 axioms are defined which help decide on best solutions. This procedure gives a guarantee that the player never loses the game, whichever this player begins the game.

III. TIC-TAC-TOE

Tic-tac-toe, also known as Noughts and Crosses, is a two player game. Players take their turn by marking X and O alternatively in any of the one from nine boxes, whichever be the empty. Tic-tac-toe's board consists of a 3x3 grid. The first player has 9 choices and second player has 8 choices to mark in the first move; thereafter the choices reduce one after each turn. The objective of the game is to place three of the player's symbol in a row, either vertically, horizontally or diagonally before the opponent manages to align their three symbols in this fashion. A winner is one who marks their symbol in a row first and the game draws, if player or opponent is not able to put three symbols in a row. Hence, result of the game playing is a win/lose or a draw.

A. Game Analysis:

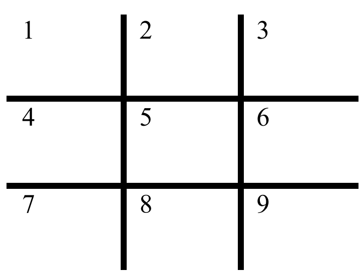

Figure 1. Tic-tac-toe board with box number

In Fig. 1, the box of the board are numbered 1 to 9 and these total 9 positions are divided in 8 possible winning sequences i.e.: 123, 456,789, 147, 258, 369, 159, 357. These sequences can be categorized either a row 123, 456, 789 or a column 147, 258, 369 or a diagonal sequence 159, 357. So based on these sequences, total 9 positions can be divided in three categories

1) 5th position, centre of the board, used 4 times in the sequence.

2) 1,3,7,9 positions, corner of the board, used 3 times each in the sequence.

3) 2,4,6,8 positions, side position, used 2 times each in the sequence.

This implies that the center is most important incomparison to others. So it should be filled first; and then corner positions are important in comparison to the side positions, so these positions should be filled next before filling the side positions. A player will win if they fill in any of these sequences by competing row, column or diagonal with the same symbol. When the game is being played, the aim of the player is to:

a) Try to stop the opponent from winning the game and

b) Try to win.

Hence the player gives the preferences first to a then to b and then to c's position. The player takes their turn by marking 'X' at any empty position of the board and the opponent by marking 'O'. Let two function $f(i)$ and $g(i)$ be defined as $f(i)$ = number of 'O' in the i^{th} sequence and $g(i)$ = number of 'O' and 'X' in the i^{th} sequence.

The Player plays the game following a process. This playing process can be described as below:

(i) Check 5^{th} position, if it is empty then mark it by 'X'.

(ii) Evaluate $g(i)$ for all sequence, $i = 1, 2, \ldots, 8$

If $g(i) = 2$, for some i^{th} sequence

① If $f(i) = 2$ or $f(i) = 0$, for this i^{th} sequence, then fill empty position by 'X' in this i^{th} sequence.

② If $f(i) = 1$, player does not give the preference to fill this sequence.

(iii) If $g(i) = 3$, for some i^{th} sequence

① If $f(i) = 0$ for this i^{th} sequence, then

In this sequence, all three positions have been filled by 'X' and player wins.

② If $f(i) = 3$, for this i^{th} sequence, then

In this sequence, all three positions have been filled by 'O' and opponent wins.

③ If $f(i) = 1$, then i^{th} sequence have one 'O' and two 'X' or $f(i) = 2$, then i^{th} sequence have two 'O' and one 'X' and player leaves to evaluate this i^{th} sequence next times.

If $g(i) = 3$, for all sequences and no player wins, then game draws.

(iv) Otherwise, choose an empty position from 1, 3, 7, 9

(v) If no position empty in 1, 3, 7, 9 then choose the empty position from 2, 4, 6, 8 and fill that position by 'X'.

In first step, the player searches the 5th position and if it is empty then they marks X. In second step, the player searches the sequence in which the player has already mark two X and one is empty or the opponent has mark two O and one is empty, if any of these are found then X is marked in an empty position in this sequence. In third step, a check is made to see whether this has resulted into a win or loss or game draw. Otherwise in fourth and fifth step the Player chooses the positions from amongst 1, 3, 7, 9 or 2, 4, 6, 8.

Based on above process, we now formulate a mathematical model below:

B. Mathematical Modal

Consider a tic-tac-toe game which consists of 9 boxes, numbered 1 to 9, as shown in Fig. 1, in which the player places 'X' and an opponent places 'O' alternatively in their turn. It is assumed that a tic-tac-toe is a 9 dimensional problem and solutions of the problem can be represented by $(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9)$, where $x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9$ are the variables corresponding to the 1, 2, 3, 4, 5, 6, 7, 8, 9 boxes respectively. Each of these variables can take -1, 0, 1 value. In the beginning of game, all these variables are zero. When the player takes their turn by placing 'X' in i^{th} box, the corresponding variable x_i will become 1 and when the opponent take their turn by placing 'O' in the j^{th} box, the corresponding variable x_j will become -1. Hence, when the player begins the game, in their first move, the player has 9 beginning solutions *i.e.* 9 choices of moves. Whereas when the opponent begins, the player has 8 beginning solutions. To decide which move should be taken following objective functions can be used:

$$f_1 = x_1 + x_2 + x_3, \quad f_2 = x_4 + x_5 + x_6,$$
$$f_3 = x_7 + x_8 + x_9, \quad f_4 = x_1 + x_4 + x_7,$$
$$f_5 = x_2 + x_5 + x_8, \quad f_6 = x_3 + x_6 + x_9,$$
$$f_7 = x_1 + x_5 + x_9, \quad f_8 = x_3 + x_5 + x_7,$$
$$f_9 = f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8$$
$$= 3x_1 + 2x_2 + 3x_3 + 2x_4 + 4x_5 + 2x_6 + 3x_7 + 2x_8 + 3x_9$$

All of these objective functions are to be maximized and a solution will be considered as a best solution, based on following axioms:

1) If $f_i = 3,\ i = 1,2,\ldots,8$ for any solution then the corresponding solution will be the best solution whatever the value of f_9 and the player will win in this scenario and this will be called the winning solution.

2) If at some solution $f_i = -2,$ for any $i, i = 1,\ \ldots, 8$ and at some solution $f_i > -2, i = 1,\ldots,8$ then the best solution is always selected from the solution in which all $f_i > -2, i = 1,2,\ldots,8$ first.

3) If $f_i = 2,$ for any $i = 1,2,\ldots,8$ and rest $f_i > -2$ at some solution then the corresponding solution is the best solution.

4) If 1, 2 and 3 do not meet in the solutions then decision will be given based on f_9 objective function, best value of f_9 will give the good move. If two solutions have equal f_9 value then decision is based on $f_1,\ldots,f_8$. In this case, best solution will be considered one which has a smaller number of negative values for the objective functions $f_i, i = 1,2,\ldots,8.$

C. Procedure of generating solution:

In the beginning of the game, each variable is assigned zero value *i.e.* $(0,0,0,0,0,0,0,0,0)$ is an initial solution. There are two possibilities: either the player begins the game or the opponent begins the game. We discuss both of these scenarios separately.

1) When player begins the game: solutions are generated

$(1,0,0,0,0,0,0,0,0),(0,1,0,0,0,0,0,0,0),(0,0,1,0,0,0,0,0,0)$

$(0,0,0,1,0,0,0,0,0),(0,0,0,0,1,0,0,0,0),(0,0,0,0,0,1,0,0,0),$

$(0,0,0,0,0,0,1,0,0),(0,0,0,0,0,0,0,1,0),(0,0,0,0,0,0,0,0,1).$

Fitness of these solutions is evaluated by $f_1,\ldots,f_9$ objective functions and the best solution, amongst them, is selected through above axioms. Then the opponent takes their turn based on their decision, i.e. in which box to place 'O' such that the corresponding variable value becomes -1. In the next move the player has only 7 solutions, based on the combination of mutating zero as 1 in the solution that are generated. Once a variable alters the value from zero to 1 or -1, variable value cannot be altered again in the entire game. Then the opponent plays the game by placing O in the empty box and in the 3rd move of the player only 5 solutions are generated; and so on. In the last move of the player, only one solution is generated, if the game is not won by player or opponent in the earlier moves.

2) When the opponent begins the game: in this case, the opponent place 'O' in any of the 9 box of the tic-tac-toe board based on their decision, the corresponding variable becomes -1. After the opponent's turn, the player generates 8 solutions by mutating zero as 1, one by one. At each of the solutions, value of the objective functions $f_i, i = 1,\ldots,9$ are evaluated and the best solution is selected, amongst them, based on best solution selection axioms and the player puts 'X' in appropriate box selected through this best solution. If two solutions are equally good then one of them can be selected randomly. In the next move, the opponent moves, by placing 'O' in the box, based on their decision. After the opponent's move, the player generates 6 solutions, evaluates fitness of the solutions, then select the best solution and places 'X' accordingly. This procedure goes on till the player wins/loses or the game results into a draw.

IV. Game Playing

Here, a game is played between player and opponent. In the game, the player begins the game and decides their move by objective function's results and places 'X' accordingly whereas the opponent places 'O' randomly. The results are shown in table 1.

For player's X_1 move, 9 solutions are generated and objective function's values at these solutions are evaluated. By axiom 4, Solution 5 is the best solution one as compared to other solutions, since $f_9 = 4$. suppose the opponent now selects 3rd position on the board to place O_2. The player generates 7 solutions for X_3 move and evaluates objective functions value at these solutions. Solution 1, 5 and 7 all have $f_9 = 4$ but solution 5 has two $f_1 = -1, f_6 = -1$ but in solution 1, $f_6 = -1$, and solution 7, $f_1 = -1$, has only one function value -1. By axiom 4, solution 1 and 7 are equally good, so the player can choose any of them (i.e. from 1 or 5). Here, we assume that the player chooses 1. Suppose the opponent selects 9th position to places O_4. Now, the player generates 5 solutions for X_5 move and evaluates the objective functions' value at these solutions. Since all the solutions have $f_6 = -2$ except for the solution 3, by axiom 2, solution 3 is the best solution amongst them. Suppose the opponent selects the 4th position on the board to places O_6. The player generates 3 solutions for X_7 move and evaluates the objective functions' value. In the solution 1 and 3, $f_5 = 2$ and other function $f_i > -2$ and solution 3 has less negative functions $f_i, i = 1, 2, \ldots, 8$ number. According to axiom 3, solution 3 is best solution. Now, suppose opponent selects 2nd position on the board to places O_8. The player places X_9 in 7th box, which is the only empty box.

TABLE I. SOLUTIONS AND CORRESPONDING OBJECTIVE FUNCTION VALUES

Player's X_1 Move										
	Solution	***Evaluation***								
	($x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9$)	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
1	(1,0,0,0,0,0,0,0,0)	1	0	0	1	0	0	1	0	3
2	(0,1,0,0,0,0,0,0,0)	1	0	0	0	1	0	0	0	2
3	(0,0,1,0,0,0,0,0,0)	1	0	0	0	0	1	0	1	3
4	(0,0,0,1,0,0,0,0,0)	0	1	0	1	0	0	0	0	2
5	**(0,0,0,0,1,0,0,0,0)**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**1**	**4**
6	(0,0,0,0,0,1,0,0,0)	0	1	0	0	0	1	0	0	2
7	(0,0,0,0,0,0,1,0,0)	0	0	1	1	0	0	0	1	3
8	(0,0,0,0,0,0,0,1,0)	0	0	1	0	1	0	0	0	2
9	(0,0,0,0,0,0,0,0,1)	0	0	1	0	0	1	1	0	3
Player's X_3 Move										
1	**(1,0,-1,0,1,0,0,0,0)**	**0**	**1**	**0**	**1**	**1**	**-1**	**2**	**0**	**4**
2	(0,1,-1,0,1,0,0,0,0)	0	1	0	0	2	-1	1	0	3
3	(0,0,-1,1,1,0,0,0,0)	-1	2	0	1	1	-1	1	0	3
4	(0,0,-1,0,1,1,0,0,0)	-1	2	0	0	1	0	1	0	3
5	(0,0,-1,0,1,0,1,0,0)	-1	1	1	1	1	-1	1	1	4
6	(0,0,-1,0,1,0,0,1,0)	-1	1	1	0	2	-1	1	0	3
7	(0,0,-1,0,1,0,0,0,1)	-1	1	1	0	1	0	2	0	4
Player's X_5 Move										
1	(1,1,-1,0,1,0,0,0,-1)	1	1	-1	1	2	-2	1	0	3
2	(1,0,-1,1,1,0,0,0,-1)	0	2	-1	2	1	-2	1	0	3
3	**(1,0,-1,0,1,1,0,0,-1)**	**0**	**2**	**-1**	**1**	**1**	**-1**	**1**	**0**	**3**
4	(1,0,-1,0,1,0,1,0,-1)	0	1	0	2	1	-2	1	1	4
5	(1,0,-1,0,1,0,0,1,-1)	0	1	0	1	2	-2	1	0	3
Player's X_7 Move										
1	(1,1,-1,-1,1,1,0,0,-1)	1	1	-1	0	2	-1	1	0	3
2	(1,0,-1,-1,1,1,1,0,-1)	0	1	0	1	1	-1	1	1	4
3	**(1,0,-1,-1,1,1,0,1,-1)**	**0**	**1**	**0**	**0**	**2**	**-1**	**1**	**0**	**3**
Player's X_9 Move										
1	(1,-1,-1,-1,1,1,1,1,-1)	-1	1	1	1	1	-1	1	1	4

X_3	O_8	O_2
O_6	X_1	X_5
X_9	X_7	O_4

Figure 1. Resultant board

Result: result of the game is draw.

V. CONCLUSIONS

An attempt has been made to solve the tic-tac-toe board game. In this procedure, the solutions are generated based on available possible moves of the state of the board i.e. branch factor of game tree at that node and which move should be chosen is decided by objective functions which are defined by the authors. The next question arises whether these types of objective functions can be defined for other board games (e.g. Chess), which helps the player chose the move following a similar procedure as reported in this paper. The benefit of the procedure is that there is no need to construct the ply of the game. So, effort of the player is thus reduced.

REFERENCES

[1] H. Bottomley, How many Tic-Tac-Toe (noughts and crosses) games are possible?, http://www.se16.info/hgb/tictactoe.htm

[2] S. Schaefer, How many games of Tic Tac-Toe are here? http://www.mathrec.org/old/2002jan/solutions.html.
[3] M. Gardner, “Ticktacktoe Games”, Wheels, Life and Other Mathematical Amusements. W.H. Freeman and company, pp. 94-105, 1983.
[4] B. Al-Khateeb and G. Kendall, “Effect of Look-Ahead Depth in Evolutionary Checkers.” Journal of Computer Science and Technology, 27(5), 996-1006, 2012.
[5] Singh, A., & Deep, K., “Use of Evolutionary Algorithms to Play the Game of Checkers: Historical Developments, Challenges and Future Prospects.” In Proceedings of the Third International Conference on Soft Computing for Problem Solving, Springer India, 55-62, 2014.
[6] D. B. Fogel, "Using evolutionary programming to create neural networks that are capable of playing tic-tac-toe." In Neural Networks, 1993, IEEE International Conference on, pp. 875-880. IEEE, 1993.
[7] G. Hochmuth, "On the genetic evolution of a perfect Tic-tac-toe strategy." Genetic Algorithms and Genetic Programming at Stanford, 75-82, 2003.
[8] A. Bhatt, P. Varshney, and K. Deb, "In search of no-loss strategies for the game of tic-tac-toe using a customized genetic algorithm." In Proceedings of Genetic and Evolutionary Computation conference (GECCO-2008),(Atlanta, USA), pp. 889-896, 2008.
[9] S.H. Ling and H.K. Lam. "Playing Tic-Tac-Toe Using Genetic Neural Network with Double Transfer Functions." JILSA 3, no. 1 (2011): 37-44, 2011.
[10] H. Mohammadi, N.P.A. Browne, A.N. Venetsanopoulos, and M.V. Santos, "Evolving Tic-Tac-Toe Playing Algorithms Using Co-Evolution, Interactive Fitness and Genetic Programming." International Journal of Computer Theory and Engineering, Vol 5:5, pp. 797-801, 2013.