Improved technique in Tic-Tac-Toe game to minimize the condition of draw using min-max over optimal strategy

Siby Samuel*
sibysam@gmail.com

Kajal Mahawar*
kajalmahawar01@gmail.com

Isac João França*
isacjoaofran.a@gmail.com

*St. Aloysius College (Autonoous), Jabalpur, M.P. India

Abstract

In Tic-Tac-Toe game when it is played between two players, one player being the user and another player as the computer, it has been observed by us that applying optimal strategy it usually ends up in a win or draw condition for the first player. In this paper, we have developed a simulation model using min-max algorithm over optimal strategy by giving the players five more moves to change the previous input to minimize the draw scenario and to increase the complexity level of the game.

Keywords: game theory, min-max, optimal strategy, Tic-Tac-Toe.

1. Introduction

Tic-Tac-Toe is a popular game. It is a game of simple rule, and easy to learn. The origin is unknown with indications stated by the ancient Egyptians that they found the Tic-Tac-Toe pattern scribbled on the rocks over more than 3500 years. Later they found fun in using this pattern for playing a game. Then the game became popular being played on wooden board or table or even in a piece of paper.

The Tic-Tac-Toe game involves filling up a 3x3 grid with either crosses (**'**X**'**) or noughts (**'**O**'**). The first player will get maximum of five turns and second player will get maximum of four turns after which the victory/draw of the game is declared. The player who first encounters three crosses ('X') or three noughts ('O') in a particular row, column or diagonal is declared as winner. There are 3^9 = 19,683 possible states in the game. The purpose of filling the nine spaces can be considered as filling the sequence of nine boxes that is maximum three in a row, column or diagonal. Therefore, there are 9! = 362,880 ways to fill the 9^{th} position [2].

Interestingly, Tic-Tac-Toe game can be reviewed using game theory, in which rational decisions of winning strategy can be considered between the two players. Tic-Tac-Toe uses the strategy that puts the player in the most preferred position irrespective of the strategy of his opponents. This strategy is called an optimal strategy.

Tic-Tac-Toe uses min-max strategy to choose an optimal move for a player assuming that the opponent is also playing optimally. The goal of min-max strategy in the game is to minimize the maximum loss (minimize the worst case condition). In min-max strategy, the game is said to be fair if maximum value = minimum value = 0, and it is said to be strictly determinable if maximum value = minimum value! = 0.

In 2012, Al-Khateeb [1] proposed artificial neural networks are used as function evaluators in order to evolve game playing strategies for the game of tic-tac-toe. In 2017, Garg [2] presented a simulation algorithm to predict the win, or draw of a game by knowing the first move of a player. In 2003, Hochmuth [3] demonstrated how a genetic algorithm can be used to evolve a perfect Tic-Tac-Toe strategy, which never loses a game who plays the game. In 2011, Ling [4] proposed an algorithm which is learned by a neural network with double transfer function (NNDTF), which is trained by genetic algorithm (GA). In 2013, Mohammadi [5] presented a novel use of Genetic Programming, Co-Evolution and Interactive Fitness to evolve algorithms for the game of Tic-Tac-Toe. In 1995, Pilgrim [6] offered a rule-based expert systems development program called Tic-Tac-Toe as part of week-long summer computer science workshop for Middle school students. In 2017, Sharma [7] described two heuristic based algorithms; they are Min-Max and Max-Min for efficient task scheduling mechanism that should be able to minimize completion time, maximize resource utilization and minimize makespan.

We observed that using optimal strategy of the game theory, the first player has the maximum chance to win the game or it will end in draw condition. In this paper, we suggest the min-max strategy over the optimal strategy to overcome the previous drawback where both the players will get the wining condition or else end in a draw.

The paper is organized as: Section 2 discusses the preliminaries and notations. Section 3 demonstrates strategies of Tic-Tac-Toe. Section 4 presents tree representation of various strategies of game. Section 5 presents the proposed model of Tic-Tac-Toe using min-max over optimal strategy. Section 6 presents the flowchart and Section 7 concludes the study.

2. Preliminaries and Notations

Tic-Tac-Toe game uses crosses (‘X’) to specify the first player’s move and noughts (‘O’) to the second player’s move. There are only three types of moves initially, namely corner, edge and center. The positions 0,2,6,8 are called ‘corners’, 1,3,5,7 are called ‘edges’ and position 4 is called ‘center’ position, please refer (cf. Figure 2.1) .

0	1	2
3	4	5
6	7	8

Figure 2.1 Board position

Min-max: it is a decision making condition to calculate the optimal move. The condition evaluates the minimum loss and maximum profit. Using it in tic-tac-toe game, a player tries to ensure two cases:

- Maximize a player’s own chances to win.
- Minimize opponent’s chances to win.

- *Maximize profit*: the profit can be maximized by forking or winning. Wining if there are two X or O in a row then play third chance to get three in a row.
- *Minimize loss*: The loss can be minimized by a block. Block if two X or O of the opponent are in a row then block it, or else block opponent's fork.

Notations

Crosses ('X'): first player's move.

Noughts ('O'): second player's move.

Maximize player: positive value (+1).

Minimize player: negative value (-1).

3. Strategies of Tic-Tac-Toe

3.1 Basic moves

There are various basic moves that a player uses to find a winning strategy:

a. If a player takes three corners, the player has three options to win.

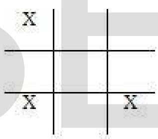


Figure 3.1 (a)

b. By taking the positions 0, 2 or 2, 8 or 6, 8 or 0, 6 that is corners and the position 4 that is center again it will give three options to win. But the opponent will try to block the next two positions that will be next to each other.

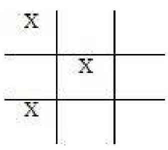


Figure 3.1 (b)

3.2 Various winning cases of Tic-Tac-Toe

The Tic-Tac-Toe game has eight possible ways to win:

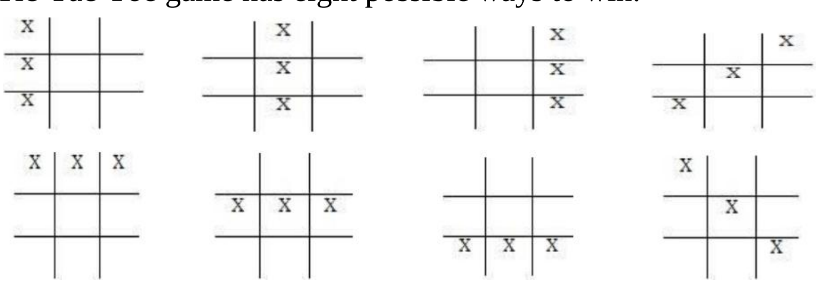


Figure 3.2

From the above Figure 3.2, it shows the winning strategies, in which we can see that out of 8 cases, 4 cases use the fourth block (center). Most players will choose that block, they figured out that it gives them the best chance of winning, because it has more possibilities than, say taking first block that is block zero (0), where the players have only three possible paths.

3.3 Draw cases

When there are no three crosses (‘X’) or noughts (‘O’) in a row or column or diagonal, the draw condition is declared. The following are the draw cases:

X	X	O
O	O	X
X	O	X

Figure 3.3 (a)

X	O	X
X	X	O
O	X	O

Figure 3.3 (b)

4. Tree representation of various strategies of game

The players play alternatively between MAX placing crosses (‘X’) and MIN placing noughts (‘O’) until we reach leaf nodes of terminal states.

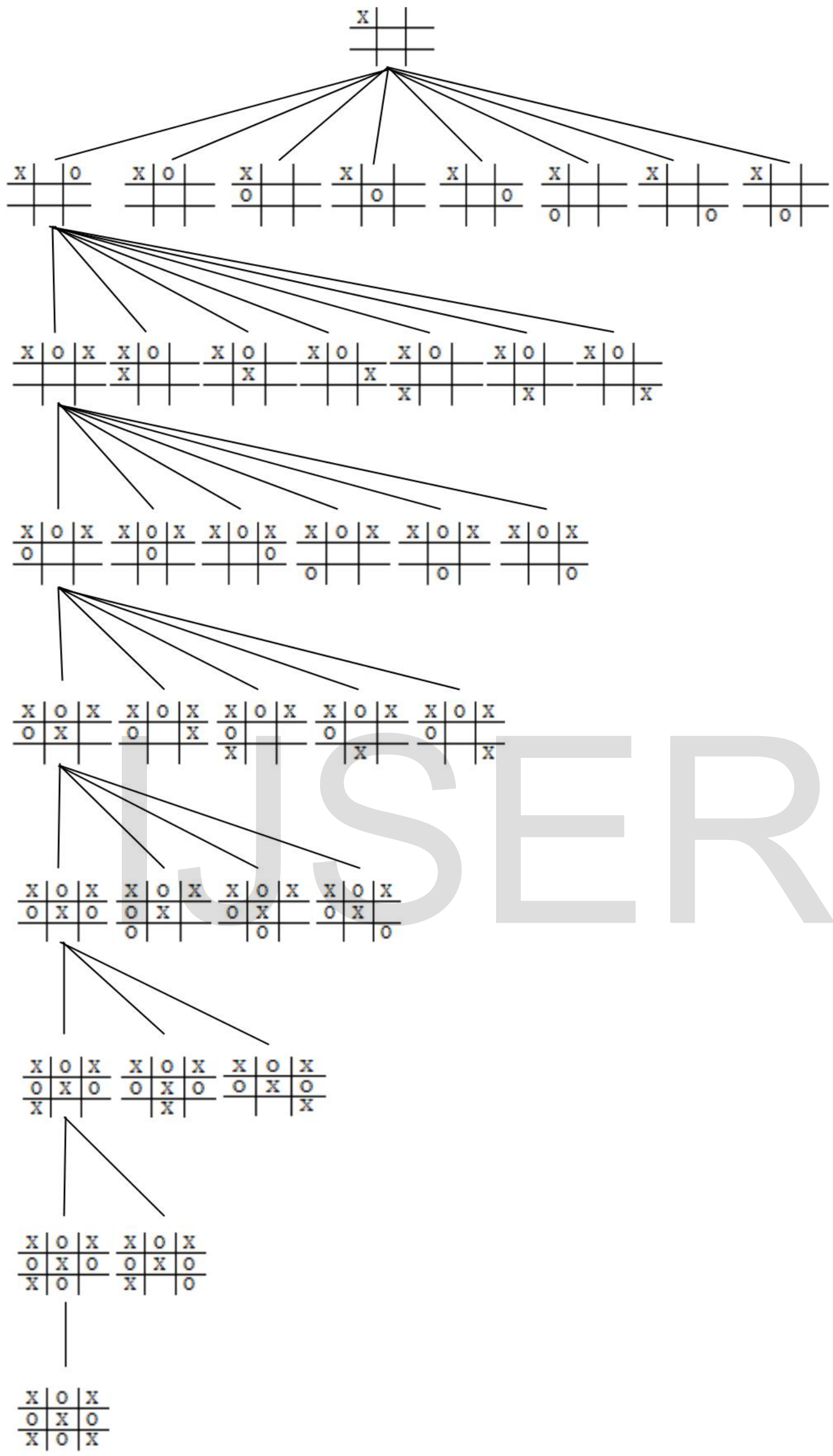


Figure 4.1

ISSN 2229-5518

5. Proposed model of Tic-Tac-Toe using min-max over optimal strategy

In min-max strategy, the first player is offered the first move and the player will have maximum number of move as compared to the second player.

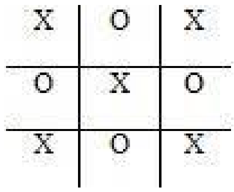


Figure 5.1 Condition when the players placed all the nine positions.

The first player can choose any position (cf. Figure 5.1 (a)), if the player chooses top right corner and the second player places ‘O’ in the center, (cf. Figure 5.1 (b)). The first player again places in another corner that is in bottom right corner, the min-max strategy evaluates the minimum profit of ‘X’ and second player places in right edge to block the first player’s chance, (cf. Figure 5.1 (c)). Then the first player places ‘X’ in left edge so that the second player does not win in the second move, (cf. Figure 5.1 (d)). The min-max evaluates the maximum profit of the first player and for blocking the second player chance of winning, places ‘X’ at top edge, and the second player minimizes the opponent’s win by placing ‘O’ at top left corner, (cf. Figure 5.1 (e)). Therefore, this will lead to a draw as (cf. Figure 5.1 (f)).

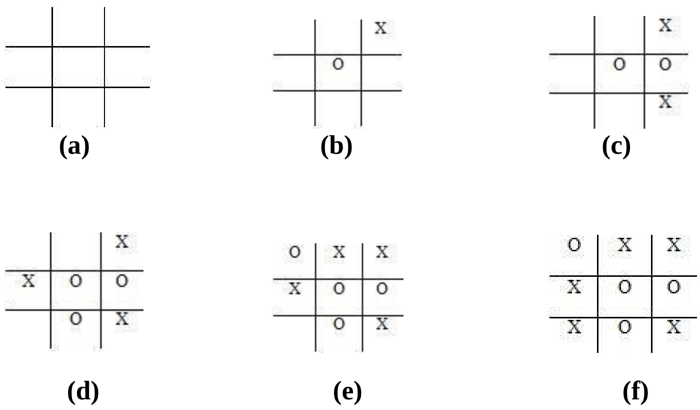


Figure 5.1 (a) Board position, (b) first player chooses top right corner, (c) player chooses another corner, (d) player puts right edge to block the opponent, (e) player again blocks the opponent by placing in top edge and (f) draw.

As we can see from the above figure, by using the crosses (‘X’), the first player places five moves and the noughts (‘O’), the second player places four moves.

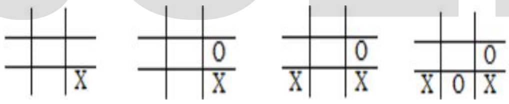
ISSN 2229-5518

In optimal strategy, the first player will get higher chances of winning or draw as compared to second player (in which it will always end up in draw or loss but never win). In this case, the first player will choose the corner or center position as its priority, and wait for the second player to place in the edge position by mistake if the second player does not know the optimal strategy.

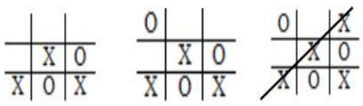
The optimal strategy of choosing corner or center position has the following algorithm:

5.1.1 Algorithm for choosing corner

1) Start from the corner (0, 2, 6, and 8).
2) If the opponent plays on an edge (1, 3, 5 and 7), play in the corner that forces the opponent to play on another side.
3) When the opponent blocks playing on the edge, conquer the center or play in another corner.
4) After conquering the center or another corner, the three **'**X**'** or **'**O**'** are formed.



(‘a) (‘b) (c) (d)



(e) (f) (g)

Figure 5.1.1

In algorithm for choosing corner, the first player will start in a corner, suppose position 8, if the second player plays on an edge (position 5), first player plays in the corner (position 6) that forces the opponent to play in another side that is position 7. When the second player blocks playing on the edge, the first player will conquer the center or play in another corner. In this case, the player will play in the center position, and the opponent will block in one of the corner (position 0 or 2). After conquering the center, the three crosses ('X') are formed.

5.1.2 Algorithm for choosing center

1) Start from the center (4).
2) If the opponent plays on the edge, play in the corner of this edge.
3) The opponent will be forced to block by placing in the far corner.
4) Make three in a row, column or diagonal by placing in the corner or edge aligned to the conquered corner or edge.

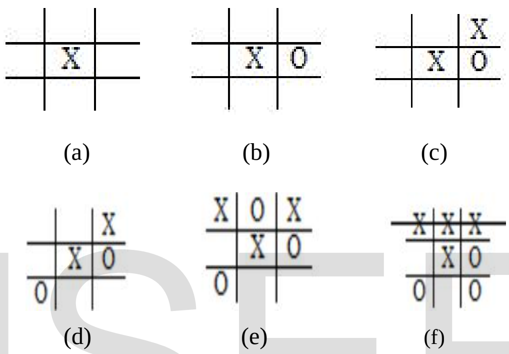
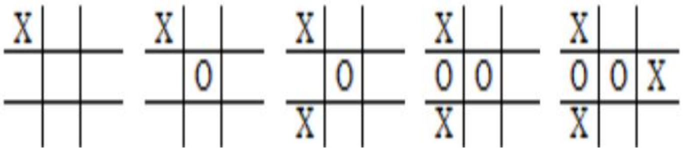


Figure 5.1.2

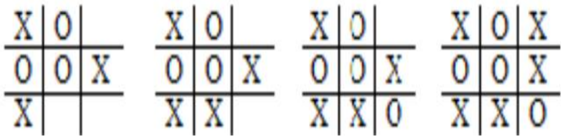
In algorithm for choosing center, the first player starts from center position. If the opponent plays on an edge (position 5), then first player plays in the corner of this edge (that is position 2). The opponent will be forced to block by placing in the far corner in the position 6. Then make three in a row or column by placing in the corner or edge aligned to the conquered corner or edge.

5.1.3 When both players are playing optimally, a draw condition is presented

In this case, the first player plays optimally by choosing the corner (position 0). Then second player also plays optimally and chooses the center position. Then first player plays in another corner in position 6. After that, second player chooses the edge (position 3) for blocking the opponent. This makes two in a row for the second player, then for blocking the opponent player, the first player chooses the edge in position 5. After that, second player chooses another edge (position 1) to make two in a column. Then again first player chooses another edge (position 7) to block the opponent. Then second player chooses the corner in position 8 to block to minimize opponent’s chance to win. Then in the last move, the first player plays in the corner (position 2) and this ends up in a draw.



(a) (b) (c) (d) (e)



(f) (g) (h) (i)

Figure 5.1.3

In our proposed model, we provide initially three moves for each player. If in the three moves any player wins then game ends, but if no winner is found then the game ends up in a draw, and we provide the players five more moves to change their previous input. After the five moves and still no winner is found, a draw is declared, otherwise a winner is declared.

In the first move, the first player chooses the corner (position 0) and second player chooses the edge (position 1). Then in second move, the first player chooses another corner (position 8) and second player chooses the center to minimize the opponent’s win. Then at third move first player choose another edge (position 7) and second player choose corner (position 6). In the given three moves, both the players end up in a draw.

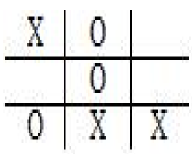


Figure 5.2

ISSN 2229-5518

When a draw is declared, five more moves are provided to change the previous inputs.

Case 1*:* In the first case, the first player changes the previous input from position 7 to position 2, and the second player changes from position 6 to position 7 and second player wins the game.

Move 1:

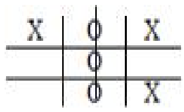


Figure 5.2.1

Case 2: In the second case, the same draw scenario is presented. In the first move, the first player changes from position 8 to position 2 and the second player from position 6 to position 5, still no winner. In the second move, the first player changes the position 0 to position 3 and the second player changes from 5 to position 0. In the third move, the first player changes the position 3 to position 5 and the second player changes from position 1 to position 8 and a win is declared in the third move out of the given five for the second player.

Move 1:

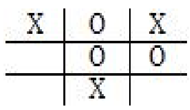


Figure 5.2.2(a)

Move 2:

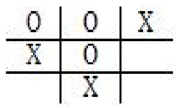


Figure 5.2.2(b)

Move 3:

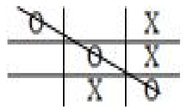


Figure 5.2.2(c)

Case 3*:* In the third case, the same draw scenario is presented. In the first move, the first player changes from position 0 to position 2 and the second player from position 1 to position 5. In the second move, the first player changes the position 8 to position 3 and the second player changes from 5 to position 8. In the third move, the first player changes the position 3 to position 0 and the second player changes from position 4 to position 1. In the fourth move, the first player changes the position 2 to position 4 and the second player changes from 6 to position 5. In the last move, the first player changes the position 0 to position 2 and the second player changes from position 8 to position 6. After each player plays in five given moves, a draw is declared.

Move 1:

	X	
	O	O
O	X	X

Move 2:

		X
X	O	
O	X	O

Move 3:

X	O	X
O	X	O

Move 4:

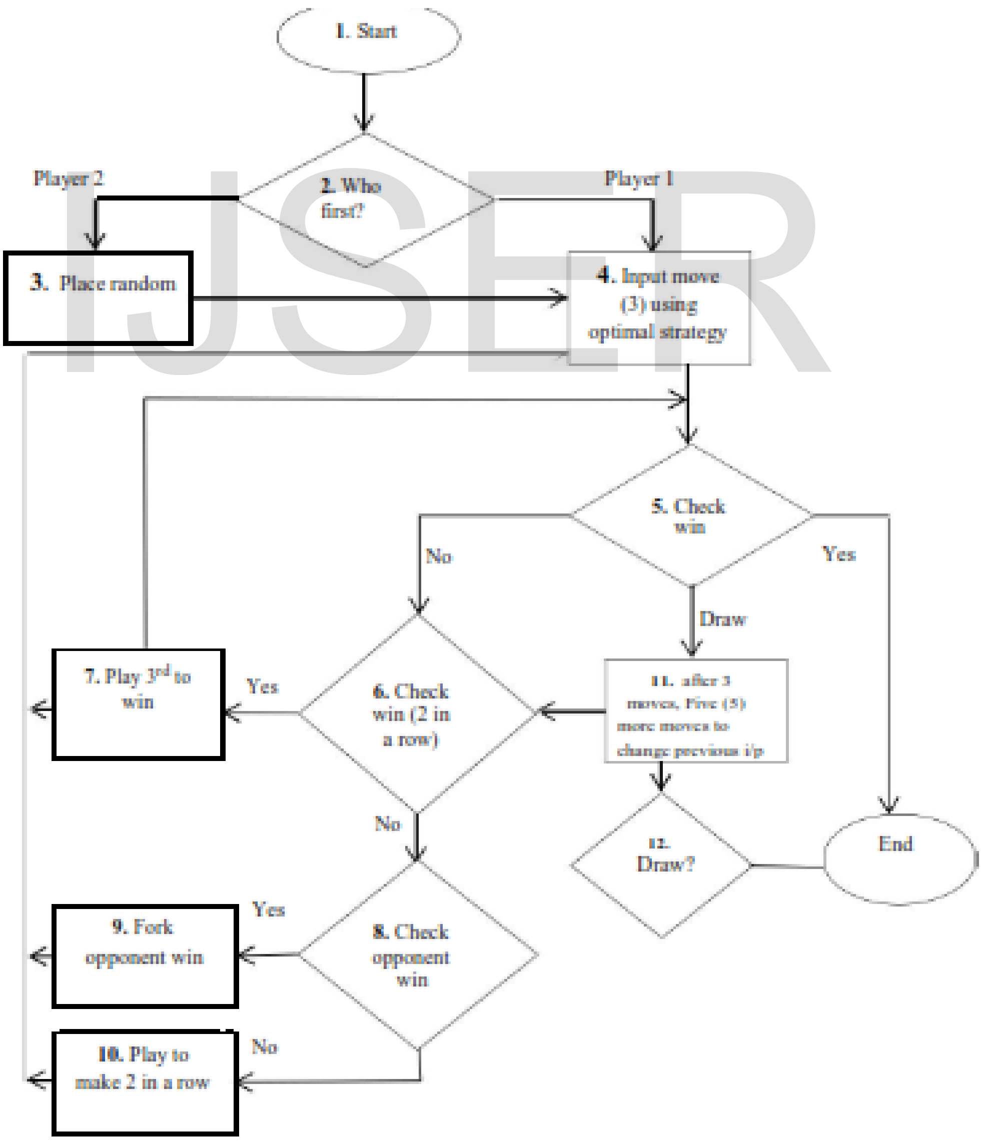
X	O	
	X	O
	X	O

Move 5:

	O	X
	X	O
O	X	

Figure 5.2.3

6. Flowchart



7. Conclusion

In this paper, we present a model to give both player the same chance of winning condition irrespectively of the strategy. The model is implemented using game theory, min-max and optimal strategy. In an ideal scenario, a player must calculate all the possibilities to ensure the success not only by blocking the other player's success but also ensuring that blocking the opponent will not give more vulnerabilities.

References

1. Al-Khateeb, B., An evolutionary tic-tac-toe player, International Journal of Reasoning-based Intelligent System, Vol. 4, No. 4, pp.182-185, 2012.
2. Garg, R., Nayak, D., Game of Tic-Tac-Toe: Simulation using Min-Max Algorithm, International Journal of Advanced Research in Computer Science, Vol. 8, No. 7, pp.1074-1077, 2017.
3. Hochmuth, G., On the Genetic Evolution of a Perfect Tic-Tac-Toe Strategy, Stanford University Book Store, pp.75-82, 2003.
4. Ling, S., H., Lam, H., K., Playing Tic-Tac-Toe using Genetic Neural Network with Double Transfer Functions, Journal of Intelligent Learning Systems and Applications, Vol. 3, pp.37-44, 2011.
5. Mohammadi, H., Nigel, P., A., Venetsanopoulos, A., Santos, M., Evolving Tic-Tac-Toe Playing Algorithms Using Co-Evolution, Interactive Fitness and Genetic Programming, International Journal of Computer Theory and Engineering, Vol. 5, No. 5, pp.797-801, 2013.
6. Pilgrim, A., R., Tic-Tac-Toe: Introduction Expert Systems to Middle School Students, Summer Computer Science Workshop, pp.340-344, 1995.

7. Sharma, N., Tyagi, S., Atri, S., A Comparative Analysis of Min-Max and Max-Min Algorithms based on the Makespan Parameter, International Journal of Advanced Research in Computer Science, Vol. 8, No. 3, pp.1038-1041, 2017.