An evolutionary tic-tac-toe player

Belal Al-Khateeb

Department of Computer Science,
College of Computer,
University of Anbar,
Ramadi, Al-Anbar, Iraq
Email: belal@computer-college.org

Abstract: In this paper, artificial neural networks are used as function evaluators in order to evolve game playing strategies for the game of tic-tac-toe. The best evolved player is tested against an online perfect tic-tac-toe player, and also against a nearly perfect player which allows 10% random moves and finally against five selected human players. Those players are with different playing abilities. The results are promising, suggesting many other research directions.

Keywords: artificial neural networks; tic-tac-toe; evolutionary algorithms.

Reference to this paper should be made as follows: Al-Khateeb, B. (2012) 'An evolutionary tic-tac-toe player', *Int. J. Reasoning-based Intelligent Systems,* Vol. 4, No. 4, pp.182–185.

Biographical notes: Belal Al-Khateeb received the BSc (Hons.) (first class) degree in Computer Science from Al-Nahrain University, Baghdad, Iraq, in 2000, and the MSc degree in Computer Science from Al-Nahrain University, Baghdad, Iraq, in 2003, and the PhD degree from the School of Computer Science, University of Nottingham, Nottingham, UK, in 2011. He is currently a Lecturer at the College of Computer, University of Anbar. He has published over 14 refereed journal and conference papers. His current research interests include evolutionary and adaptive learning particularly in computer games, expert systems, and heuristics and meta/hyper-heuristics. He has a particular interest in computer games programming.

This paper is a revised and expanded version of a paper entitled 'An evolutionary tic-tac-toe player' presented at the '2nd Conference on Computer and Information Technology (CCIT' 2012)', Ramadi, Iraq, 4–5 April 2012.

1 Introduction

The game of tic-tac-toe is a commonly played game. Many who play the game develop some strategies on their own which usually do not let the player lose the game. However, in a significant proportion of the games played, the game ends with a draw. In the past, researchers have studied computing methods to generate efficient strategies for not having to lose the game even once. Hochmuth (2003) demonstrated how a genetic algorithm (GA) can be used to evolve a perfect tic-tac-toe strategy, which never loses a game it plays. He concluded that there are 827 unique game states that are encountered during game play and concentrated on finding a single no-loss strategy. The study reported a single no-loss strategy, but did not provide the description of that strategy to know its properties. Soedarmadji (2005) suggested a decentralised decision-making procedure to find a competent strategy which forces a draw or a win depending on the proficiency of the opponent player. Although such a goal should result in a no-loss game-playing strategy, Bhatt et al. (2008) observed that the resulting strategy reported in the study loses in at least three different scenarios. In the work of Bhatt et al. (2008), the goal was to revisit the use of GAs in finding not one but as many no-loss strategies as possible. For this purpose, a representation scheme similar to that in Hochmuth (2003) was used and designed as new ways of evaluating a solution through matrix processing in MATLAB, a new initialisation scheme, customised GA operators with a controlled elite preservation scheme and two-tier GA procedure. Interestingly, the study is able to find more than 72,000 no-loss strategies for playing the game of tic-tac-toe, which were not reported earlier. Furthermore, the paper analysed the set of 72,657 no-loss solutions to arrive at a number of efficient strategies which produce excellent win-to-draw ratio, a matter which has not also been paid much attention in the past. The results of this study are interesting and may motivate similar such studies for other board games as well.

Our experiments demonstrate that the evolutionary algorithms can be used to evolve a reasonably good tic-tac-toe players in a short time without injecting human experts knowledge.

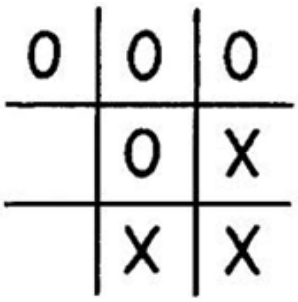
The rest of the paper is organised as follows: in Section 2 the nature of the game of tic-tac-toe together with related work is presented. Section 3 gives background about the evolutionary algorithms. The statistical tests are discussed in Section 4. Section 5 shows the experimental set-up. Section 6 presents the results and a conclusion is presented in Section 7.

2 Tic-tac-toe

This section gives a brief description of tic-tac-toe game, and then summarises previous approaches to learning to play it. Tic-tac-toe, also known as noughts and crosses, is

played on a 3×3 board between two players, black and white (black moves first). Each player has five pieces, which are placed on empty squares. Pieces are fixed in their positions. The player who succeeds in placing three respective marks in a horizontal, vertical, or diagonal row wins the game. Figure 1 shows a tic-tac-toe board.

Figure 1 Tic-tac-toe board



In this paper, our goal is to evolve a reasonably good tic-tac-toe player in a short time without injecting any human expert knowledge rather than the basic rules of the game. The results of this study are interesting and may motivate similar such studies for other board games as well.

3 Background

Evolutionary algorithms draw their inspiration from the real life behaviour of how humans learn and how they continuously evolve throughout their life time. These algorithms form part of the machine learning research domain. For more than a decade, researchers have attempted to get machines to learn some given task by embedding knowledge into the machine, integrating the machine with a database(s) that contains human knowledge or providing a learning algorithm to enable it to learn by itself.

Based on Michalski et al. (1983), there are two basic forms of learning, these being knowledge acquisition and skill refinement. In knowledge acquisition, the person learns some new information and has the ability to apply that information in an effective manner. In skill refinement, with some basic information at the start of the learning process, the person repeats a certain activity in order to continually improve their skill level. Humans learn by a combination of these two activities.

In Fogel's work (Chellapilla and Fogel, 1999a; Chellapilla and Fogel, 1999b; Chellapilla and Fogel, 2000; Chellapilla and Fogel, 2001; Fogel, 2002; Fogel and Chellapilla, 2002; Fogel et al., 2004; Fogel et al., 2005; Fogel et al., 2006), no information was given to the machine. The machine learnt to play a game by itself through its experience, using co-evolutionary learning. Fogel combined evolution strategies with neural networks and used a minimax search tree as a look ahead mechanism to find potentially good moves for the game of checkers (Chellapilla and Fogel, 1999a; Chellapilla and Fogel, 1999b; Chellapilla and Fogel, 2000; Chellapilla and Fogel, 2001; Fogel, 2002; Fogel and Chellapilla, 2002) and chess (Fogel et al., 2004; Fogel et al., 2005; Fogel et al., 2006). Blondie24 (an automated checkers player) can play at the human expert level. The work on chess (Blondie25) is still ongoing, but initial results are promising.

4 Statistical test

A student *t*-test with the following settings: unequal variances, alpha (α) = 0.05, and one-tail test, will be used in order to determine whether two players are statistically the same. It is worth mentioning that any two players are statistically the same if the value (called *p* value) obtained from the *t*-test is less than alpha.

5 Experimental set-up

For the purpose of investigating our hypothesis (producing a reasonably good tic-tac-toe player in a short time without injecting any human expert knowledge), the following evolutionary algorithm is used:

1 Initialise a random population of 20 neural networks (players) sampled uniformly [–0.5,0.5] for the weights.

2 Each neural network is fully connected and consists of nine input nodes, each one represents a square from the tic-tac-toe board, two hidden layers consist of five and three nodes respectively, and finally one output node. The output (in the range [–1,1]), is calculated using the hyperbolic tangent, of the neural network represents how strong is the tic-tac-toe board.

3 Each player has its associated self-adaptive parameter, initialised to 0.05.

4 Play each player against all other players using round robin tournament.

5 For each game, the player receives a score of +1 for a win, 0 for draw and –1 for a loss.

6 Games are played until either side wins, or until nine moves are made by both sides, in which case a draw is declared.

7 After completing all games, the 10 strategies that have the highest scores are selected as parents and retained for the next generation. Those parents are then mutated to create another 10 offspring using the following equations:

$$s_i(j) = s_i(j)exp(tN_j(0,1)), j = 1, ..., N_w$$

$$w_i(j) = w_i(j) + s_i(j)N_j(0,1), j = 1, ..., N_w$$

where N_w is the number of weights and biases in the neural network $t = \frac{1}{\sqrt{2 \times \sqrt{N_w}}}$, and $N_j(0,1)$ is a standard Gaussian random variable resembled for every *j*.

8 Repeat steps 4 to 7 for 2000 generations

The best player from the above algorithm, which is obtained over 2000 generations, is called TTT. In order to make sure that the TTT is not a 'fluke' of optimisation, we decided to construct 20 players, comparing them by using round robin tournament and test if they are statistically the same by using student *t*-test (assuming unequal variances, $\alpha = 0.05$, and one-tail test) for their total score (each win got three points, a draw got one point and a loss got zero points). The null hypothesis is that two players are the same if the *p* value obtained from the *t*-test is greater than alpha. Bearing in mind that all the players are end products (always playing with the same strategy). Table 1 shows the results.

Table 1 Number of scored points for all the players

Player	*Points*
TTT(10)	51
TTT(15)	51
TTT(6)	50
TTT(14)	50
TTT(17)	50
TTT(4)	49
TTT(12)	49
TTT(1)	48
TTT(7)	48
TTT(16)	48
TTT(20)	48
TTT(8)	47
TTT(11)	47
TTT(18)	47
TTT(2)	46
TTT(5)	46
TTT(19)	46
TTT(3)	45
TTT(9)	45
TTT(13)	45

Based on Table 1, there is no statistical difference between the players as the *p* value (*p*-value = 1) for the one tail *t*-test is greater than alpha. So as all the players are statistically the same, we decided to choose the player with the most number of points to be the baseline player, TTT.

6 Results

In order to test the outcome of the evolutionary tic-tac-toe player, TTT is set to play 50 games against an online program, which can be found at http://www.agame.com/game/tic-tac-toe.html. As the online program is designed to avoid lose (perfect player), a high number of draws will be considered as a success for the TTT player. Also TTT is set to play 50 games against a nearly perfect tic-tac-toe player (it is a perfect player but allowed to make 10% random moves). The results are shown in Tables 2 and 3. Finally the performance of TTT is tested against five selected human players. Those human players are with different abilities (they are sorted in ascending order according to their strength) and each player played 50 games against TTT. The results are shown in Tables 4 through 8.

All the experiments were run using the same computer (1.33 GHz Intel Atom processor and 2GB Ram). All the experiments to evolve the players were run 2000 generations took about two days.

The results in Table 2 show that TTT drew 43 games, out of 50, against a perfect tic-tac-toe player, which reflects a success for the evolved TTT player, bearing in mind that TTT is trained for two days only. Table 3 shows that TTT is better than a nearly perfect player, as the results show that TTT won 28 games and lose only two games, out of the 50 played games, which reflect a success for the evolved TTT player.

Table 2 Results when playing TTT against online program

	Opponent: Online Program		
	Win	Draw	Lose
TTT	0	43	7

Table 3 Results when playing TTT against a nearly perfect player

	Opponent: Nearly Perfect Player		
	Win	Draw	Lose
TTT	28	20	2

The results in Table 4 show that TTT is better than the first selected human player, as the results show that TTT won 35 games and lose only two, out of 50 played games, which clearly indicates a success for our hypothesis. Table 5 shows that TTT won 27 games and lose only 5 games, out of the 50 played games, against the second selected human player, which reflect another success for the evolved TTT player. According to the results in Table 6, TTT is better than the third selected human player as TTT won 20 games and lose 10, out of the 50 played games, while the results in Table 7 show another success for TTT, as TTT won 14 games and drew 14 games, out of the 50 played games, against a strong human player. Finally Table 8 shows that TTT drew 39 games, out of the 50 played games, against the fifth selected human player. The fifth player is played 50 games against the perfect online tic-tac-toe player in which all the games end in draw, which clearly indicates that the fifth selected human player is a perfect player. Those results reflect the success for the evolved TTT player as getting 39 draws against a perfect player is clearly considered as a success.

Table 4 Results when playing TTT against Human_Player1

	Opponent:Human_Player1		
	Win	Draw	Lose
TTT	35	13	2

Table 5 Results when playing TTT against Human_Player2

	Opponent:Human_Player2		
	Win	Draw	Lose
TTT	27	18	5

Table 6 Results when playing TTT against Human_Player3

	Opponent:Human_Player3		
	Win	Draw	Lose
TTT	20	20	10

Table 7 Results when playing TTT against Human_Player4

	Opponent:Human_Player4		
	Win	Draw	Lose
TTT	14	14	22

Table 8 Results when playing TTT against Human_Player5

	Opponent:Human_Player5		
	Win	Draw	Lose
TTT	0	39	11

As TTT is better than a nearly perfect tic-tac-toe player and also better three of the selected human players and most importantly is of comparable performance with a perfect players (online and human) and a strong human player then it seems quite appropriate to use the evolutionary approach to construct a strong tic-tac-toe players.

7 Conclusions

This paper has used an evolutionary algorithm to construct a strong tic-tac-toe player, called TTT. The hypothesis of the paper was to evolve reasonably good tic-tac-toe players in a short time without injecting any human expert knowledge rather than the basic rule for the game. TTT was evolved in two days, and the proposed algorithm shows promising results when tested against various tic-tac-toe players, which clearly achieve the hypothesis.

Based on the results in Tables 3–6, TTT found to be superior to some of the selected players. Also the result in Tables 2, 7 and 8 showed that TTT is of comparable performance to strong players.

Based on the results it would seem appropriate to use evolutionary algorithms to evolve reasonably good tic-tac-toe players in a short time.

References

Bhatt, A., Varshney, P. and Deb, K. (2008) 'In search of no-loss strategies for the game of tic-tac-toe using a customized genetic algorithm', *GECCO 2008*, pp.889–896.

Chellapilla, K. and Fogel, D.B. (1999a) 'Evolution, neural networks, games, and intelligence', *Proceedings of the IEEE*, Vol. 87, pp.1471–1496.

Chellapilla, K. and Fogel, D.B. (1999b) 'Evolving neural networks to play checkers without relying on expert knowledge', *IEEE Transactions on Neural Networks*, Vol. 10, pp.1382–1391.

Chellapilla, K. and Fogel, D.B. (2000) 'Anaconda defeats hoyle 6-0: a case study competing an evolved checkers program against commercially available software', *Congress on Evolutionary Computation*, La Jolla Marriot Hotel, La Jolla, California, USA, pp.857–863.

Chellapilla, K. and Fogel, D.B. (2001) 'Evolving an expert checkers playing program without using human expertise', *IEEE Transactions on Evolutionary Computation*, Vol. 5, pp.422–428.

Fogel, D.B. (2002) *Blondie24 Playing at the Edge of AI*, Academic Press, USA.

Fogel, D.B. and Chellapilla, K. (2002) 'Verifying anaconda's expert rating by competing against Chinook: experiments in co-evolving a neural checkers player', Neurocomputing, Vol. 42, pp.69–86.

Fogel, D.B., Hays, T.J., Hahn, S.L. and Quon, J. (2004) 'A self-learning evolutionary chess program', *Proceeding of IEEE*, Vol. 92, pp.1947–1954.

Fogel, D.B., Hays, T.J., Hahn, S.L. and Quon, J. (2005) 'Further evolution of a self-learning chess program', *Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games (CIG05)*, Essex, UK, pp.73–77.

Fogel, D.B., Hays, T.J., Hahn, S.L. and Quon, J. (2006) 'The Blondie25 Chess Program Competes Against Fritz 8.0 and a Human Chess Master', *Proceedings of the IEEE 2006 Symposium on Computational Intelligence and Games (CIG06)*, Reno, USA, pp.230–235.

Hochmuth, G. (2003) *On the Genetic Evolution of a perfect Tic-Tac-Toe Strategy*, Stanford University Book Store, pp.75–82.

Michalski, R.S., Carbonell, J.G. and Mitchell T.M. (1983) *Machine Learning: An Artificial Intelligence Approach*, Tioga Publishing Company, Palo Alto, California.

Soedarmadji, E. (2005) 'Decentralized decision making in the game of tic-tac-toe', *Proceedings of the 2005 IEEE Symposium on Computational Intelligence and Games*, pp.34–38.