DE GRUYTER
OPEN

Print ISSN: 1312-2622; Online ISSN: 2367-5357
DOI: 10.1515/itc-2018-0018

Population Based Algorithms for Artificial Neural Networks Training in Perfect Information Games

I. Blagoev, D. Keremedchiev, I. Zankinski

Key Words: *Perfect information games; artificial neural networks; population based heuristic optimization.*

Abstract. *This research focuses on population based heuristic optimization used for artificial neural networks training. The goal is artificial neural networks to be used as computer opponent in games with perfect information. Artificial neural networks have many applications and one of them is in the game theory. For perfect information games with relatively small board, artificial neural networks can perform well. Training of artificial neural networks with population based heuristics has an advantage that calculations can be done in a distributed computing system, for example a mobile game which communicates with remote server.*

1. Introduction

In game theory there is a particular branch devoted to games with perfect information. Perfect information games are interesting with the fact that each player has full information available for the game progress [1,2]. When each player has the full information about the game, each player is capable of estimating all possible game strategies and all different game scenarios. The most popular perfect information game is Chess. The game is still interesting even nowadays because it has huge amount of different gameplays. The game tree varies in its width and it has loops. Another even more complex game is the game of Go. It has high complexity because of the board size, which is 19x19 cells in its original variant. The simplest known perfect information game is the game Tic-Tac-Toe [3]. Tic-Tac-Toe is the best studied game in the class of perfect information games. Computer opponents in this game are programmed with the usage of exact numerical methods. In this research a game called Dice Overflow will be presented. It has higher complexity than Tic-Tac-Toe, but also it has much lower complexity than Go. The goals in this research are mainly practical and are related to the implementation of an effective computer opponent, which is capable of playing the game Dice Overflow. A three-layer artificial neural network is trained with population based optimization algorithms for this purpose.

The paper is organized as follows: Section 1 introduces the problem; Section 2 presents the game, artificial neural networks and population based optimization algorithms; Section 3 introduces practical implementation and some experimental results; and Section 4 concludes with some remarks for further research.

2. Computer Opponent Proposition

For the solution of the problem presented in the introductory part of this paper, a multilayer perceptron is proposed as a computer opponent in a perfect information game called Dice Overflow. The training of the multilayer perceptron is done with combination between differential evolution and back-propagation algorithms.

2.1. Dice Overflow

Dice Overflow (*figure 1*) is a perfect information board game developed in IICT-BAS as master thesis [4]. The game evolves the idea introduced in the game Overflow [5] (*figure 2*). Circles are used in the original game. At each turn, the player can select which circle to rise in its size. In Dice Overflow circles are replaced with dice. The game is for two players, who play in turns. First player plays with the red dice, second player plays with the blue dice. The board has size of 5x5 and each player starts his initial position with only one die on the board with value of five. On each turn players select only one die and rise its value by one. When the selected die has a value of six it overflows in the neighboring cells. This overflowing gives the name of the game. The four neighbors (left, up, right, down) start with dice, which has a value of one when neighbors are empty. The overflowed cell turns its die to value of three. If the neighboring cells are not empty each non-empty neighbor rises its value with one. If some neighbor has value of six it also overflows and a chain reaction of overflowing can be triggered. If the color of the neighboring dice is the opponent's color those dice are captured and their color is changed. The winner of the game is the player who converts all dice on the board in his/her color.

2.2. Artificial Neural Networks

Artificial Neural Networks started their development as ambitious goal to present mathematical model of the biological neural systems. As data structure commonly used Artificial Neural Networks are weighted directed graphs. In the case of Multilayer Perceptron the nodes of the graph are organized in layers. Usually each layer is fully-connected with the previous and the next layer. The input layer has connections only with the next layer and the output layer has connections only with the previous layer. The edges in the graph have weights, which represent the knowledge collected inside the Artificial Neural Network. During the training phase weights of the network are modified

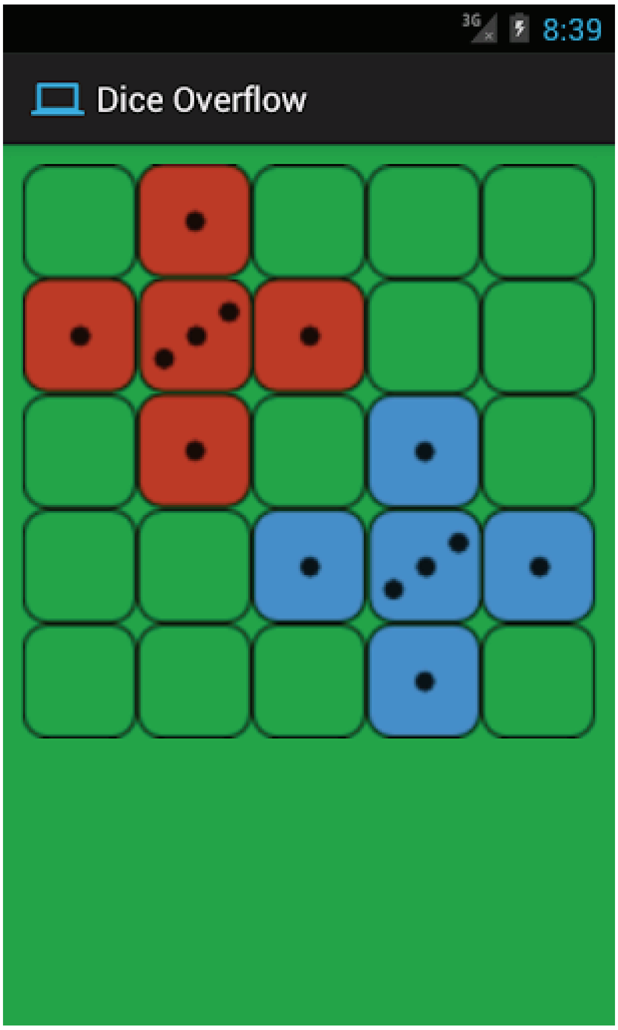

Figure 1. Dice Overflow

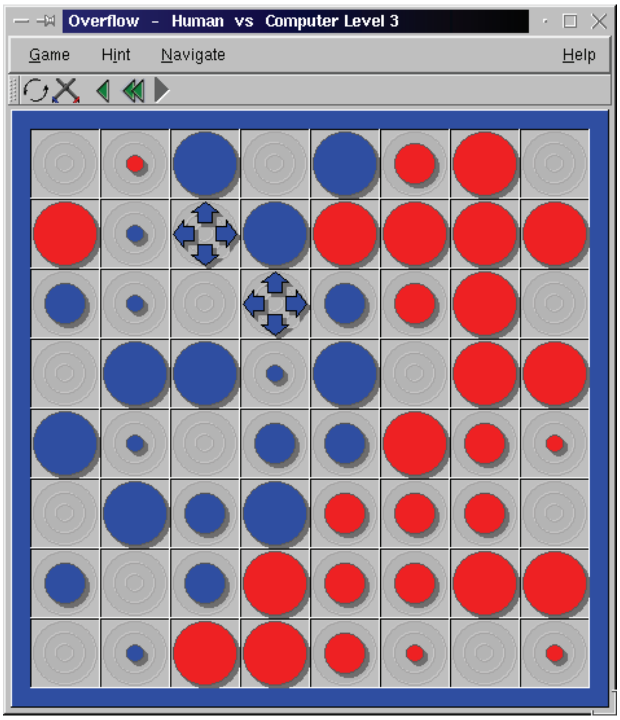

Figure 2. Overflow

according to specified training rule. When training is done by input-output examples it is called Supervised Training. Such Artificial Neural Networks are valuable tools for classification problems, pattern recognition, forecasting and other. The main advantage of Artificial Neural Networks is their ability for self-adaptation, by finding complex relations between input-output examples. Artificial Neural Networks are operating in two common phases - training and processing. During the training phase network topology should be established and proper values for the weights between neurons should be calculated. During the operation phase input information is supplied at the network input and the network gives its reaction at the output.

2.3. Population-Based Optimization

In the field of the global optimization methods there is a branch called Population-Based Heuristic Optimization. In this methods a set of points from the solution space are presented as population. Most of the population based optimization methods are also evolutionary, which means that some kind of evolving operators are applied over the individuals in the population. The ideas for such optimization methods are inspired by the theories of the natural evolution. Each individual in the population has its own fitness value (result of the target function calculation). According to fitness values some kind of selection is applied. Selected parents are recombined (in most cases by crossover) to create new generation. In some cases newly created individuals are slightly modified (some kind of mutation). Global heuristic optimization does not guarantee finding globally optimal solutions. In most of the cases found solutions are near the global optimums. In current research individuals in the population are the weights of the Multilayer Perceptron (matrix of real numbers). Uniform crossover is used with small change in weights by the usage of difference vector as mutation (Differential Evolution). Fitness value is calculated in a tournament between all networks in the population playing Dice Overflow game against each other.

2.4. Practical Implementation

Based on theoretical arguments Artificial Neural Network is organized to act as a computer opponent in a software implementation of the game Dice Overflow. The network has three layers and its training is done by Back Propagation and Differential Evolution. The network has no recurrent connections, because its only goal is it to recognize particular board patterns and to react with particular signal. The input information is organized with the following values of the input layer with 25 nodes: -0.90, -0.75, -0.60, -0.45, -0.30, -0.15, 0, +0.15, +0.30, +0.45, +0.60, +0.75, +0.90 The network uses hyperbolic tangent as activation function and that is why values of the dice are scaled between -1.0 and +1.0. Negative values are used for one of the players. Positive values are used for the other player. If a cell on the board is empty zero is used. The network has output of 25 nodes and each one gives evaluation of the position to be played. The most positive value in the output is the most promising move of the positive player. The most negative value in the output is the most promising move of the negative player. The size of the hidden layer is estimated with the usage of the pruning algorithm implemented in Encog software library. Artificial Neural Networks weights are stored as individuals in Differential Evolution population. For the tournament different weights are loaded inside the two opponent networks and the score of the game is used for fitness value. Each individual receives 3 points for win in the tournament, 1 point for tie game and 0 points for loss. Sum of the points is presented as final score and it is result of 2*N*(N-1) games, because each network plays against each another in a population of N individual. Each network plays against each other twice – first as blue player and second as red player. Elitism rule is applied in order for the best found solution to survive until the end of the training. Training termination has two constraints – predefined rate of success or predefined number of generations.

3. Experiments and Results

Four different experiments are done with Encog Artificial Neural Networks framework and Apache Commons Genetic Algorithms framework. The game Dice Overflow is implemented as Java program in Android application.

I the first experiment (*figure 3*) both computer players are doing random valid moves (Random Search Method). The performance of the both players is almost identical in this experiment.

For the second experiment player with random move strategy is playing against computer player with predefined discrete probability distribution patterns (*figure 4*). In this case predefined probabilities give some advantage to one of the players. The performance is not huge, but it is clearly visible.

When Artificial Neural Network is trained only with Back Propagation rule it behaves better than the discrete distribution pattern (*figure 5*).

The performance of the Artificial Neural Network is improved when Back Propagation training is combined with Differential Evolution.

4. Conclusion

The proposed theoretical solution is directly applicable in a real life software implementation of the game as Android mobile application. There are many other opportunities for experiments with evolution based population optimization algorithms. Because the mobile devices are widely available mobile distributed computing system can be created and parallel evolutionary algorithms can be researched.

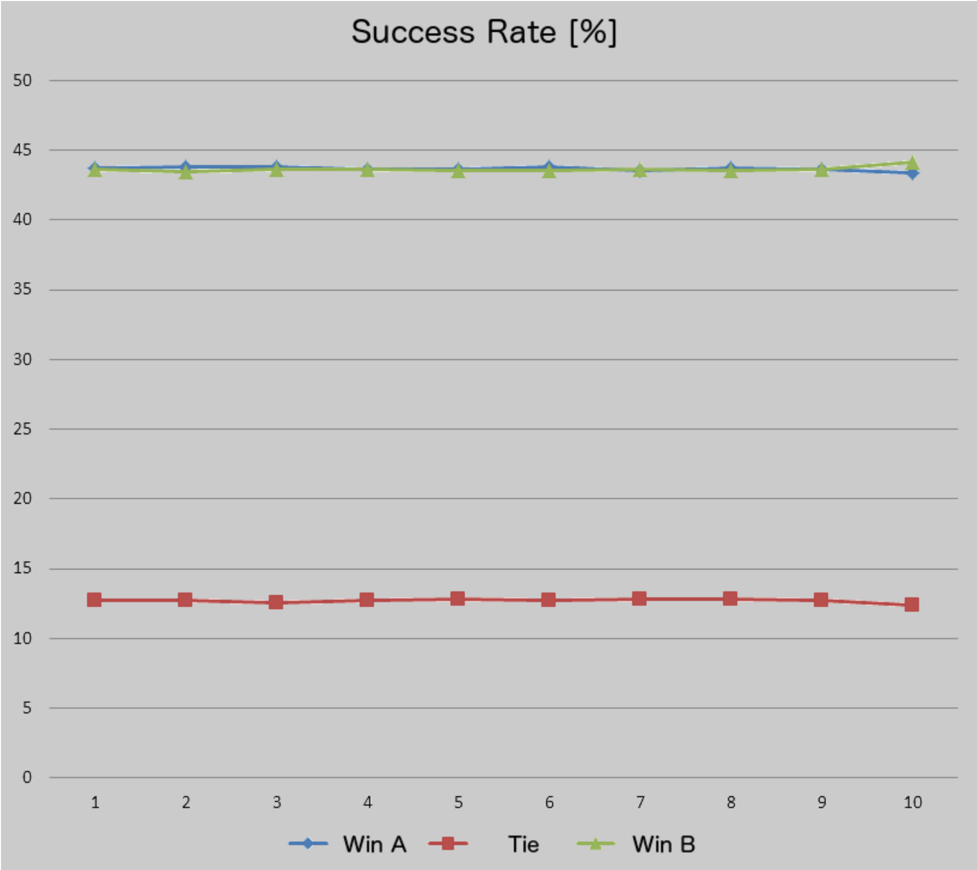

Figure 3. Random play of both players

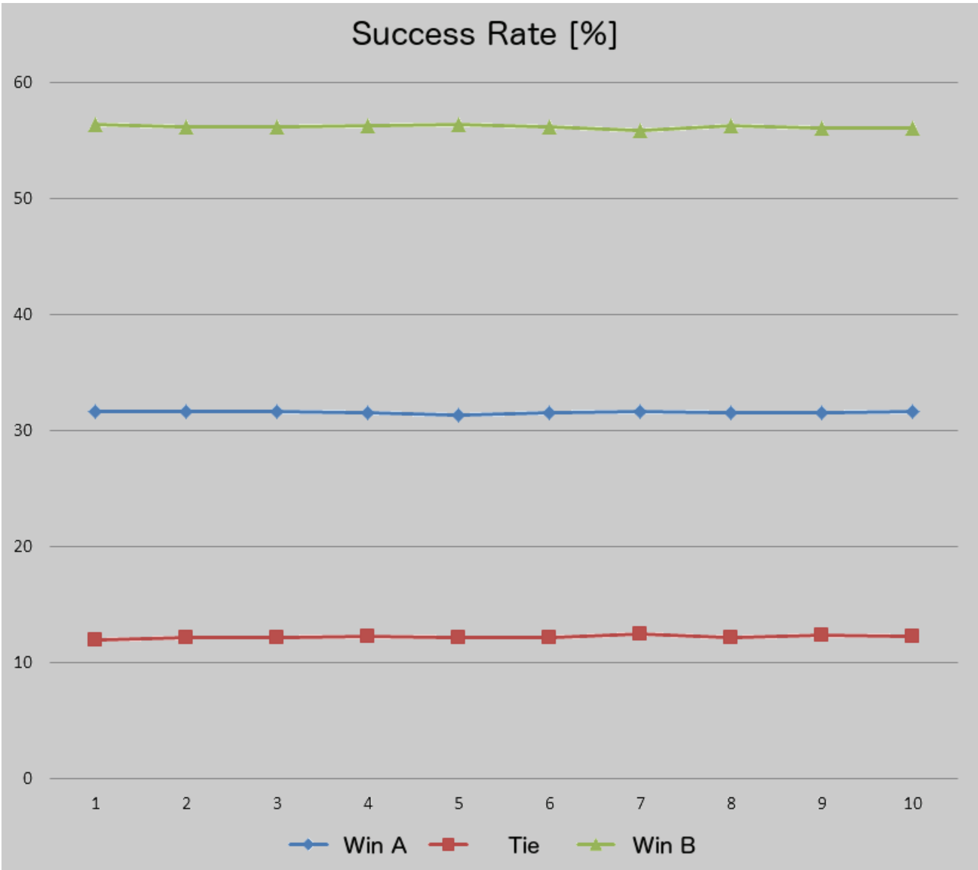

Figure 4. Random player and discrete pattern player

Figure 5. Random player against ANN player

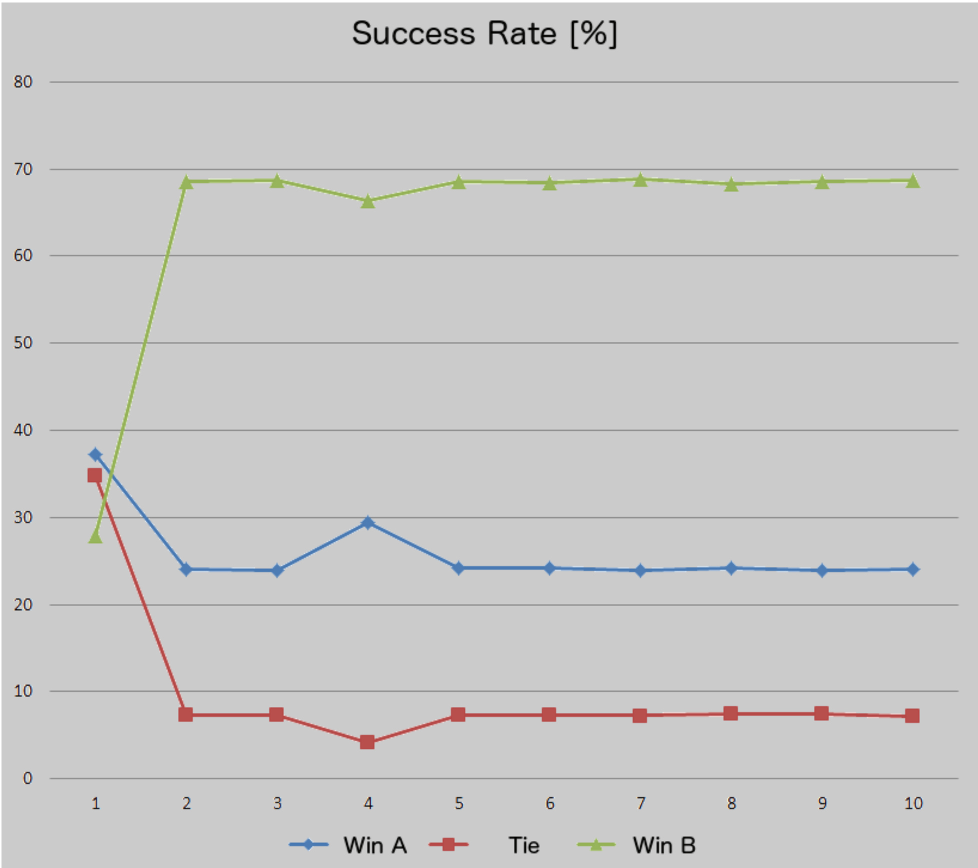

Figure 6. Random player against DE trained ANN

Acknowledgements

This work was done by ideas and private funding provided by Velbazhd Software LLC.

References

1. Fudenberg, D., J. Tirole. Game Theory. MIT Press. (Chapter 6, sect 1), 1993.

2. Gibbons, R. A Primer in Game Theory. Harvester-Wheatsheaf. (Chapter 3), 1992.

3. Beck, J. Combinatorial Games: Tic-Tac-Toe Theory. Cambridge University Press, 2008.

4. Balabanov, T., D. Ilieva. Dice Overflow Android Board Game, http://github.com/VelbazhdSoftwareLLC/DiceOverflow.

5. Laursen, T. OVERFLOW – The Game. http://overflow.sourceforge.net/.

Manuscript received on 20.05.2019

Ivan Blagoev *is a PhD student at Institute of Information and Communication Technologies – Bulgarian Academy of Sciences. His research is in the field of financial time series forecasting.*

Contacts:
e-mail: i.blagoev@iit.bas.bg

Denlyan Keremedchiev *is a programmer at Institute of Information and Communication Technologies – Bulgarian Academy of Sciences. He is working in the field of university education and IT training.*

Contacts:
e-mail: d_keremedchiev@bas.bg

Iliyan Zankinski *is a programmer at Institute of Information and Communication Technologies – Bulgarian Academy of Sciences. He is working in the field of artificial intelligence applied in real life problems.*

Contacts:
e-mail: iliyan@hsi.iccs.bas.bg

Contacts:
Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
Acad. Georgi Bonchev St., block 2, 1113 Sofia, Bulgaria
phone: +359 2 9793237
e-mail: iict@bas.bg