

Directed Study – Report

Jéssica Pauli de C. Bonson (B00617515)

1. Methodology

The goal of this research project is to develop an AI-player for backgammon using the NEAT algorithm. To accomplish this the ANJI (Another NEAT Java Implementation) implementation was used. A new package *backgammon* was created inside ANJI that contained the code specific for the backgammon game.

The backgammon Java code was initially developed in an independent way. The program was able to play a game between two artificial players that played randomly. The game generates all valid moves and then feed it to a random board evaluator. The Java class *SecureRandom* is used to generate all the necessary random numbers. The reward function produces a 1 for a win and a -1 for a loss, it doesn't considers gammons and backgammons. The doubling cube also isn't implemented. An example of how the board game is printed in a turn is shown in Figure 1.

```
INFO -----
Backgammon, turn: 33
dice [0]: 1| dice [1]: 6|

white bar: 1 | white beared: 0
black bar: 0 | black beared: 0

IS RACING? false

[11] : [1]BLACK           | [12] : [0]NEUTRAL
[10] : [0]NEUTRAL        | [13] : [0]NEUTRAL
[9 ] : [0]NEUTRAL        | [14] : [1]WHITE
[8 ] : [0]NEUTRAL        | [15] : [0]NEUTRAL
[7 ] : [0]NEUTRAL        | [16] : [0]NEUTRAL
[6 ] : [3]BLACK          | [17] : [0]NEUTRAL

[5 ] : [3]BLACK          | [18] : [2]WHITE
[4 ] : [3]BLACK          | [19] : [0]NEUTRAL
[3 ] : [2]BLACK          | [20] : [3]WHITE
[2 ] : [3]BLACK          | [21] : [0]NEUTRAL
[1 ] : [4]WHITE          | [22] : [0]NEUTRAL
[0 ] : [4]WHITE          | [23] : [0]NEUTRAL
```

Figure 1 A board state printed by the program

In the second phase of the implementation the independent code was integrated with ANJI, so the board evaluator would be a NEAT network. Here the board state is mapped as an input to NEAT. There are a total of 29 inputs, including checkers in the bar for both player, checkers beared for both players, the total number of checkers in each position and a bias. The values for the opponent checkers are negative while the values for the current player are positive. Co-evolution is used to evolve the networks, where a network plays for a pre-defined number of matches against another network in a single-elimination tournament. In this sort of tournament if a network loses the set of matches against another network then it can not be the champion anymore. At the third development phase the benchmark player Pubeval was added to the project, so the performance of the developed networks could be tested against it.

2. Results, Conclusion and Future Work

The current results are that the code for the backgammon game was successfully integrated with the NEAT algorithm. The benchmark Pubeval was also successfully integrated. The current goal is to finish an environment that enables the NEAT algorithm to run for a long time with ways to evaluate it while running. The high-performance environment will be accomplished by running the code on a remote server. To be able to better evaluate the state of the run the current project will be improved to generate the training and testing curves. The training curve will show the fitness of the champions of each generation. The testing curve will be created by making the champions of the generations play against Pubeval after some amount of generations.

After accomplishing the tasks stated previously, this is the intended future work: 1) Run NEAT for a long amount of time, in different runs, to achieve a network that wins at least around +0.596 ppg against Pubeval (then being similar or better than Tesauros's TD-Gammon 2.1 with 1-ply search [1]); 2) Do experiments and analyze the species to see if a better performance is obtained when the matches are between individuals from the same or of different species; 3) After achieving a network with a good performance, analyze what strategies are being used and why the network is getting a good performance; 4) Write a paper about the research project and publish it in the Fall Term of 2013.

References:

[1] http://www.scholarpedia.org/article/User:Gerald_Tesauro/Proposed/Td-gammon