2007

Evolving Players for an Ancient Game: Hnefatafl

Philip Hingston
Edith Cowan University

10.1109/CIG.2007.368094
This article was originally published as: Hingston, P. F. (2007). Evolving Players for an Ancient Game: Hnefetafl. Proceedings of IEEE Symposium on Computational Intelligence and Games. (pp. 168-174). Honolulu. IEEE. Original article available here

Proceedings of the 2007 IEEE Symposium on
Computational Intelligence and Games (CIG 2007)

Evolving Players for an Ancient Game: Hnefatafl

Philip Hingston
School of Computer and Information Science
Edith Cowan University
p.hingston@ecu.edu.au

Abstract— **Hnefatafl is an ancient Norse game – an ancestor of chess. In this paper, we report on the development of computer players for this game. In the spirit of Blondie24, we evolve neural networks as board evaluation functions for different versions of the game. An unusual aspect of this game is that there is no general agreement on the rules: it is no longer much played, and game historians attempt to infer the rules from scraps of historical texts, with ambiguities often resolved on gut feeling as to what the rules must have been in order to achieve a balanced game. We offer the evolutionary method as a means by which to judge the merits of alternative rule sets.**

Keywords: Evolution, board games, Hnefatafl

I. Introduction

Board games have long been a favourite source of challenge problems for artificial intelligence researchers. Perhaps this is because a good player of a complex board game encapsulates so much of what we aspire to in our artificially intelligent creations: choosing one's actions by reasoning about how they will affect the future. The well-defined and delimited nature of a game makes it possible to state the task precisely, while the need to account for an inventive and wily opponent provides depth and complexity. Decades of research have attempted to replicate human play as a process of symbolic reasoning, with the development of techniques such as alpha-beta search, based on using codified domain knowledge in the form of a board position evaluation function, and culminating in the success of computer chess systems such as Deep Blue ([1]).

However, human play is not entirely a rational process. In recent years, research on board games carried out by computational intelligence researchers has shed light on how instinctive abilities can be married with pure reasoning. A famous example is Fogel and Chellapilla's evolved checkers player, Blondie24 ([2][3][5]). In this work, the researchers combined instinct: an evolved or learned ability to judge the value of a board position, with reasoning: a game-tree search algorithm. Other researchers have explored Awari ([4]), Go (e.g. [6][10]), Othello (e.g. [8][12]), Ms PacMan ([7]), Backgammon (e.g.[9][11]) and many other games.

In this paper, we apply similar methods to an ancient Norse board game, Hnefatafl. Hnefatafl is not much played in the present day, and has not been studied by the computer game community, so little is known about good play for Hnefatafl – in fact, even the rules of the game are uncertain. We investigate the use of an evolutionary algorithm to evolve players for the game, and use these to judge the merits of different sets of rules.

II. Hnefatafl

Hnefatafl (translated approximately as "King's table") is an ancient board game, historically played in Iceland, Scandinavia, Ireland and Wales. The game is played between two players, designated White ("the Swedes") and Red ("the Muscovites"), on a board arranged as a square grid of square cells (see Fig 1). Each cell may be occupied by a white or red piece, or may be empty. White controls the white pieces, while Red controls the red pieces. White's aim is to get a designated white piece (the "King" – shown as a crown in the figure) to one of a number of "safe" cells, while Red's aim is to capture the king before it can do so.

Skill at Hnefatafl was once a requirement of Viking manhood. According to the instructions in a box set bought from the Icelandic National Museum, Earl Rögnvaldur Kali describes a noble Viking thus:

Tafl emk örr at efla,	*I play Tafl enthusiastically*
íþróttir kank níu,	*I can play nine sports*
tynik trauðla rúnum,	*I know all the runes*
tíð er bok ok smíðir,	*Often I read and craft*
skríða kank á skíðum,	*I can go down hills on skis*
skýtk ok raek, svát nýtir,	*I am fully able to row and bunt*
hvárt tveggja kank hyggja	*I am interested in both*
harpslátt ok bragpáttu	*Playing of instruments and poetry*

Here is the rule set for the version of Hnefatafl we used in this study (based on the rules given in the box set):

The Board: The board is 11 cells x 11 cells. The centre cell (the "castle") is where the King starts. The corners are the safe cells (the "burgs").

The Pieces: There are 13 white pieces, including the King, initially arranged as in Fig 1. There are 26 red pieces, distributed along the sides of the board as shown.

Moving: Red plays the first move and then players alternate. On his move, a player must move one of his pieces to a new cell, where pieces move in the manner of a rook in chess (i.e. any number of cells either horizontally or vertically, as long as the intervening cells are empty.) Once a piece has been moved, any captured pieces from the opposing side are

removed from the board. The one restriction is that only the King may move onto a burg or the castle.

Capturing: Any piece other than the King is captured and removed from the board if an opposing piece is moved so as to pincer it, occupying the cells immediately to the left and right, or immediately above and below the piece being attacked. For this purpose, a burg or the castle is regarded as occupied by an attacking piece. It is not possible to commit suicide: a piece that wedges itself between opposing pieces is not captured. A King can only be captured by surrounding it on all 4 sides, but the edge of the board and/or a burg or the castle can assist in the capture. Note that if the King is captured, Red immediately wins the game. Fig 2 shows some examples of pieces being captured.

Winning: White wins the game if he succeeds in moving the King to a burg. Red wins if he succeeeds in capturing the King.

As a practical matter and for completeness, we added the following rules:

Stalemate: If a player cannot play a legal move, his opponent becomes the winner.
Draw: In order to limit the length of games, we set an arbitrary limit of 100 moves, after which the game is declared a draw.

These last two have been added to cover two situations that the rules as stated do not mention. Since we intend to carry out experiments with automated play, we must cater for every eventuality.

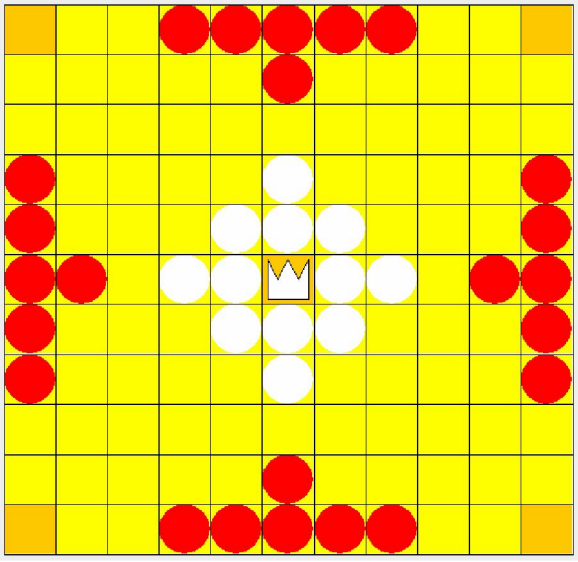

Fig 1: Initial board layout for our version of Hnefatafl

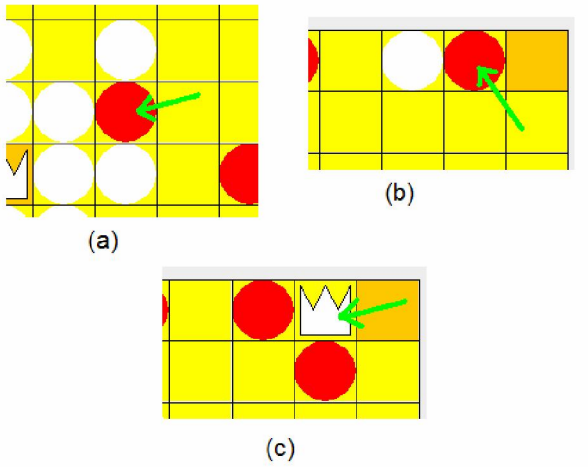

Fig 2: Examples of captures. In (a), the indicated red piece is captured by the white pieces above and below. In (b) the red piece is captured by the white piece to its left, with the assistance of a burg to its right. In (c), the king is captured by the red pieces to its left and below, with the assistance of the burg and the edge of the board.

Since the rules are asymmetric, two games are played to decide the winner of a match – one as White and one as Red. If one player wins more games, then he/she is declared the winner of the match. If each player wins an equal number of games, then the player with the greatest total number of captures wins the match. If the numbers of captures are also equal, then the match is a draw.

These rules are far from universally agreed. Points of difference between rule sets include:

- The board size can be 9x9 or 13x13;
- The number and initial arrangement of pieces can vary;
- The capture rule for the King may be made the same as for ordinary pieces;
- The King may be banned from assisting in captures;
- There may be a prohibition against repeating board positions;
- There may be a "shieldwall" rule, allowing the capture of a connected group of pieces by completely surrounding them up against one of the sides (reminiscent of capturing groups against the side of the board in Go);
- There may be a rule allowing the King and exactly one defender to be surrounded and captured;
- Red pieces may be restricted from moving through certain sections of the board.

Some of these are merely regional differences, but some result from the fact that the historical sources are incomplete or ambiguous about the rules, and game historians have had to infer (i.e. make educated guesses about) what the actual rules were.

III. EVOLVING PLAYERS

As there is little available knowledge on what makes for good Hnefatafl play, an evolutionary algorithm seems a sensible approach. Following earlier researchers, we coevolve populations of players.

The idea is to create players that select moves using a suitable game-tree search algorithm, based on an evolvable board position evaluation function. Typically, the fitness of a player (or rather its evaluation function) is measured by playing a sample of games against other players in the population.

The hope is that as the population evolves, players that "discover" general principles of good play will be favoured by selection. Unfortunately, all that the evolutionary algorithm really ensures is that players that perform well against the current generation of players will be favoured, which is not the same thing. Problems such as stagnation and cycling can occur.

Various methods have been proposed to prevent these problems. A common idea is to add some kind of diversity mechanism, such as fitness sharing, or imposing a spatial neighbourhood structure. Generally, larger populations are recommended.

Unfortunately, we had to use rather small populations. The reason is that Hnefatafl has a very high branching factor: over 100, compared with around 10-20 for chess or checkers. We implemented a simple-minded heuristic function (described later) and a standard alpha-beta search algorithm, and found that even with pruning, the effective branching factor is around 10-20. Pre-sorting based on heuristic function values did not improve the effectiveness of pruning. We also tried various other standard approaches to improve the speed of the search, including aspiration search, the use of transposition tables, and the memory-based search method MTD - none gave a significant improvement. Thus, we could not afford to use large populations, as the time needed to evaluate fitnesses of a large population would be prohibitive.

In order to offer some protection against stagnation, we included in the evaluation a sample of games against a "random" opponent – a player that evaluates all board positions equally, and selects between moves of equal value randomly. With a search depth of 1, this Random player plays random legal moves. The Random player has the nice property that it is not deterministic. Two evolved players playing against each other will usually repeat the same game over and over, so we cannot obtain a larger sample of games to evaluate players by having them play each other several times. We can use the Random player for this purpose.

Taking all this into consideration, we chose a (5+5)-Evolution Strategy (ES) as our evolutionary algorithm. ES's have proven to be effective at this type of parameter-tuning task, where the genome is a large vector of real values, and are reputed to be effective with small populations. Runarsson et al. ([9]) had success with this type of algorithm in evolving board evaluation functions for small-board Go. Blondie24 was evolved using a similar evolutionary algorithm.

Pseudo-code for the algorithm follows:

1 *Initialize:*

Set $\tau \leftarrow 1/\sqrt{2n}$. Set $\tau' \leftarrow 1/\sqrt{2\sqrt{n}}$.

Set $\sigma'_{k,j} \leftarrow 1/\sqrt{n}, k = 1\cdots 5, j = 1\ldots n$.

Set $\omega'_k \leftarrow (1/n)N(0,1), k = 1\cdots 5$.

2 *While more generations do*

3 *For k:= 1 to 5 do*

4 $\sigma_{k,j} \leftarrow \sigma'_j \exp(\tau' N(0,1) + \tau N_j(0,1)), j = 1\ldots n$

5 $\omega_k \leftarrow \omega'_k + \sigma_k N(0,1)$

6 *od*

7 *Evaluate fitness of members of* ω_k *and* ω_k' *by playing each against all the others, and against the Random player.*

8 $\omega' \leftarrow$ *select the best 5 from* $\omega + \omega'$

9 *od*

Fitness evaluation was done by playing matches between pairs of players or between a player and the Random player, and keeping track of the results of these matches. Due to the high branching factor of Hnefatafl, we used a search depth of 1 – i.e. players simply select the next legal move with the highest available evaluation. Every player played a match against each other player in the population, and a further 10 matches against the Random player. A player earned 2 points for winning a match, 1 point for a drawn match, 0 for a loss. From these results, we calculated the average number of points earned per match against other players, and also against the Random player. Finally, the fitness of a player was calculated as the sum of these two averages.

We ran the ES for 300 generations, and saved the highest fitness individual from each generation (note that this may not truly be the fittest individual, as the fitness evaluation is noisy). Further testing was then carried out on these fittest individuals to analyse their performance.

IV. NEURAL NETWORK FOR BOARD EVALUATION

In order to evolve an evaluation function for Hnefatafl, we must choose a suitable representation. Following previous researchers, we opted to use an artificial neural network. The board position is used to generate the input values to the network, and the network output value is taken as the estimated value of the board for Red. In order to facilitate games ending in a reasonable number of moves, we set the value of any board that is a win for Red to a value larger than any other board value, and for a White win, less than any other board value.

A good representation must strike a balance between expressive power and the size of the search space. Neural networks, with their universal approximation capabilities, fit the bill nicely in terms of generality, but we needed to be careful not to make the search space too large, especially as we have a limited budget of fitness evaluations with which

to tune the parameters of the networks. Again taking a lead from earlier researchers, we decided to reflect the spatial structure of the game in the network design. This both reduces the size of the search space and embeds some domain knowledge. Of course, in doing so, we acknowledge that we bias the search.

Fig 3 illustrates the spatial neural network structure we used. To improve the clarity of the diagram, we have show only a sample of neurons and a sample of connections. The input layer contains one neuron for each cell on the board. Their activation values are set at 0 for an empty cell, +1 for a cell occupied by a red piece, -1 for a white piece, -2 for the king. Next, there is a "feature" layer with three groups of spatially defined neurons and three additional neurons. The first spatially defined group contains one neuron for each vertical strip of 3 columns of cells on the board (thus, 9 neurons). Each of these has a connection from the corresponding input neurons. The next group contains one neuron for each horizontal strip of 3 rows. The last group contains one neuron for each 3x3 patch of cells on the board (thus, 81 neurons). Also in this layer is one neuron whose activation level is set to the number of red captures, one set to the number of white captures, and one set to 1 if Red is next to play, 0 otherwise.

The next layer (the "hidden" layer) contains 10 neurons each fully connected to the feature layer. Finally, the output neuron is fully connected to the hidden layer. All internal neurons also have a bias, and their activation function is the sigmoid, $1/(1+e^{-x})$. Thus, the total number of weights and biases in each network is 2862.

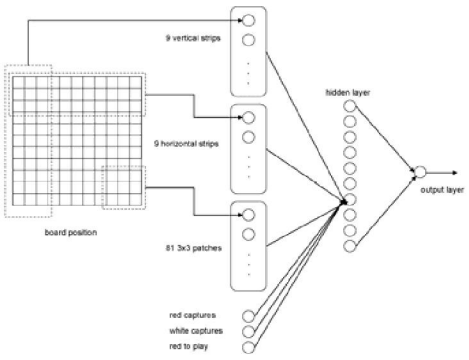

Fig 3: Structure of the neural networks used to represent evaluation functions.

We do not claim that this is the best possible representation. We did try a few other configurations, and this one performs at least as well. We chose the vertical and horizontal strips to match the pattern of movement of the pieces, and the dimension 3 for the 11x3, 3x11 and 3x3 sections to match the method of capture of pieces. We added the capture counts because these seem likely to be an important feature, and for our purposes, there seems little point in requiring the evolutionary process to discover them (which would be difficult given the way we have structured the rest of the network). The "red to play" neuron provides information that is obviously necessary for a good evaluation function, and cannot be deduced from the board position.

This network structure is roughly comparable with network structures used for other games. For example, in comparison to Blondie24 for checkers: the total number of weights and biases is comparable, Blondie had neurons for other sized patches (4x4, 5x5, 6x6, 7x7 and 8x8) but did not have neurons for vertical and horizontal strips, and Blondie had a "piece difference" input fed directly to the output neuron where we have "capture count" neurons as part of the feature layer.

V. Results and Discussion

To investigate the abilities of the evolved players, we saved the best player of each generation, and carried out detailed tests on these. Once again, these tests were time consuming, as they involve playing additional games. Therefore, we generally tested only every 50th generation.

As well as playing a large number (100) of games with a search depth of 1 (to get a more accurate estimate of performance), we played a moderate number (20 and 10 respectively) of games with search depth 2 and search depth 3 (searches deeper than 3-ply were too slow). Since we are interested in the actual playing strength of the players, not in their performance against the rest of the population, we tested the players against the Random player, and also against a simple, hand-crafted opponent.

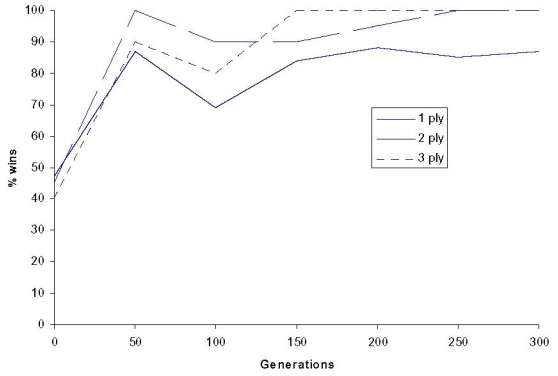

Fig 4: %wins for the best evolved player versus the Random player, with various numbers of plies, with the original rule set.

Fig 4 shows the performance of the fittest individual from each chosen generation against the Random player. The solid line shows the percentage of wins, with both players playing at 1-ply. This is a major component of the fitness function (which also includes performance against other players in the population, and a reward for draws over losses). The winning percentage rises from around 50% at the start to around 80% at the end of the run.

The other series show the percentage of wins with both players playing at 2-ply, and then at 3-ply. This gives an additional advantage to the evolved players, with winning percentage rising to better than 90% at 2-ply, and steady at 100% for 3-ply. Thus we can be confident that the evolved

player has learned some useful features for good play. However, learning appears to have stopped, and it may be that stagnation has set in.

A. A Simple heuristic

As an additional test, we designed a simple heuristic, based on a most rudimentary grasp of the game (because that is the only grasp we have!). It uses two features: piece capture difference (since matches can be won by capturing more pieces than the opponent, and presumably having more of ones own pieces on the board is an advantage), and the Manhattan distance of the King to the closest burg (on the expectation that it is good for White to advance towards a safe cell). Thus, the heuristic we used is:

$$\text{Value(board)} = (\text{redCaptures} - \text{whiteCaptures}) + 0.1 \times \text{kingDistance}$$

This heuristic gives many moves equal value, and we chose between the highest value moves randomly. Thus, like the Random player, each game against this Simple player is likely to be different, and we can play a sample of games against it.

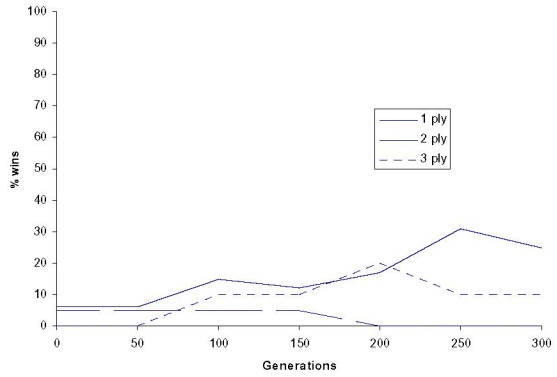

Fig 5: %wins for the best evolved player versus the Simple player, with various numbers of plies, with the original rule set.

Fig 5 shows the performance of the evolved players against the Simple player. From a starting point of around 5%, performance at 1-ply rises to around 30%. Thus, the evolved player is weaker than this naïve designed player. However, performance does improve, confirming that relevant features have been learned. Extending the search depth produces a curious result: the evolved player does much worse than might be expected at 2-ply, losing nearly all its games. But performance picks up at 3-ply, with the relative strength versus the Simple player being about the same as at 1-ply. We are unsure of the reason for the poor performance at 2-ply, but other authors have noted that evaluation functions evolved at one search depth are not always suited to use at other depths, as a weakness in the evaluation can be covered over by the search.

It is tempting to try to evolve players using the Simple player in place of the Random player, in the hope of evolving stronger players. In fact, we did some experiments along those lines, and found that the evolved players could compete about equally with the Simple player at 3-ply. However, this would not necessarily result in truly stronger play, just play that fares better against the Simple player. More satisfying would be to improve the co-evolutionary algorithm.

B. Red and white balance

Table 1 shows a the number of wins and losses for the best evolved player in the final generation, broken down to compare performance as Red to performance as White. The data shows that it is around 3 times easier to win as White, and 10 times easier to lose as Red.

TABLE 1 - NUMBERS OF WINS AND LOSSES AS RED OR WHITE FOR THE BEST EVOLVED PLAYER VERSUS RANDOM AND SIMPLE PLAYERS, WITH ORIGINAL RULE SET.

	won as red	won as white	lost as red	lost as white
1-random	21	40	21	1
2-random	0	6	1	0
3-random	1	10	4	0
1-simple	17	69	83	6
2-simple	0	0	18	4
3-simple	0	9	10	0
totals	39	134	137	11

This suggests that our chosen rule set is not well balanced – Red seems to have a much harder task than White. One alternative set of rules attempts redress the balance by making it easier to capture the King, and limiting the power of the King. The changed capture rule is:

Capturing: Any piece (*deleted: other than the King*) is captured and removed from the board if an opposing piece is moved so as to pincer it, occupying the cells immediately to the left and right, or immediately above and below the piece being attacked. For this purpose, a burg or the castle is regarded as occupied by an attacking piece. It is not possible to commit suicide: a piece that wedges itself between opposing pieces is not captured. *Added: The King cannot be used to capture another piece.* (*deleted: A King can only be captured by surrounding it on all 4 sides, but the edge of the board and/or a burg or the castle can assist in the capture.*) Note that if the King is captured, Red immediately wins the game.

We repeated our earlier experiments, using this alternative capture rule, with all other condition identical. Fig 6 shows the performance of the fittest individual from each chosen generation against the Random player when we use this alternative rule set. Once again, the evolved players learned useful features for good play, soundly defeating the Random player at each search depth, but with some performance drop this time for 2-ply.

In Fig 7, we see the same plot for games against the Simple player. With this alternative rule set, the evolved player actually beats the Simple player at 1-ply, and is nearly equal in strength at 3-ply. As with the original rule set, there is a significant drop in relative performance at 2-

ply. This is an encouraging result – remember that the evolved player has never seen the Simple player during learning.

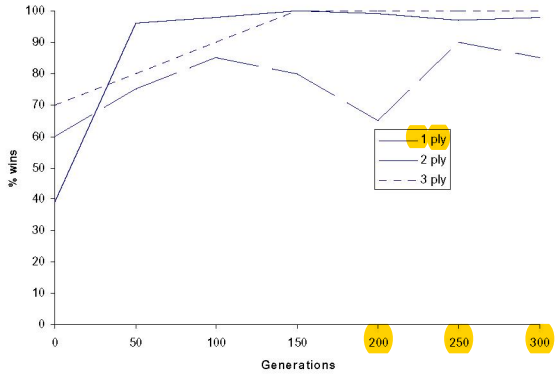

Fig 6: %wins for the best evolved player versus the Random player, with various numbers of plies, with the alternative rule set.

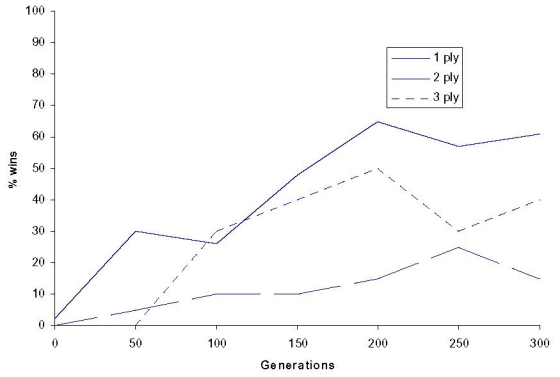

Fig 7: %wins for the best evolved player versus the Simple player, with various numbers of plies, with the alternative rule set.

Focusing now on the Red/White balance, it appears, from **Error! Not a valid bookmark self-reference.**, that we have gone too far in handicapping White, and it is now easier to win as Red, and easier to lose as White. However, looking more closely, we see that this is not the case at 3-ply, where the scales are nicely balanced. While this is the result of only a single run of the evolutionary algorithm, it suggests that the alternative rule set is the more balanced one. As a bonus, these more balanced rules appear to make the game easier to learn.

TABLE 2 - NUMBERS OF WINS AND LOSSES AS RED OR WHITE FOR THE BEST EVOLVED PLAYER VERSUS RANDOM AND SIMPLE PLAYERS, WITH ALTERNATIVE RULE SET.

	won as red	won as white	lost as red	lost as white
1-random	98	0	0	0
2-random	9	3	4	0
3-random	9	10	1	0
1-simple	93	14	7	84
2-simple	0	0	10	4
3-simple	5	5	4	4
totals	214	32	26	92

VI. CONCLUSION

In this paper, we have made a modest start towards developing computer players for Hnefatafl, an ancient ancestor of chess. We have demonstrated that it is possible to evolve a neural network to compute a meaningful board evaluation function using little or no domain knowledge. More usually in studies like this, there are skilled human players who can serve as a source of expertise, as well as a benchmark for computer players. In our case, there is little current day knowledge about how to play the game. In this situation, machine learning methods offer a way to develop some understanding of the principles of good play for the game.

Our other contribution is the idea of using evolutionary algorithms in evaluating candidate rule sets. Perhaps there are other historical games that could be studied in the same way.

We would like to challenge other researchers to develop better Hnefatafl players. Hnefatafl is an interesting game with a rich history, and there is certainly plenty of room here for improvement! A number of areas spring immediately to mind:

- The high branching factor severely limits the depth of searches. There is a need to solve this problem, so that larger populations, deeper searches and longer evolutionary runs can be used.
- The neural network structure used should be improved. The challenge is to provide enough representative power without the search space becoming infeasibly large.
- The board has a number of symmetries. These might be exploited.
- Other kinds of approaches might be needed. For example, in another game with a high branching factor, Go, a successful approach has been to generate promising moves, rather than evaluating all moves.

APPENDIX – SOME ONLINE HNEFATAFL RESOURCES

For the interested reader, here are some WWW starting points for information about Hnefatafl.

http://alumnus.caltech.edu/~leif/games/Hnefetafl/

This site has a good list of rules and variants, and provides a very weak player.

http://home20.inet.tele.dk/rnielsen/hnefatafl_online.html

This site allows for various configurations. The human plays Red.

http://www.irt.org/games/js/hnefat/

Has a fixed configuration and rules - different from ours. An applet allows two humans to play. There is no computer opponent.

http://www.scandinavica.com/games/tablut.htm

Includes a downloadable game.

http://www.northvegr.org/family/tafl/index.php

A 9x9 downloadable player.

REFERENCES

[1] Campbell, M., Hoane, A.J. and Hsu, F.-h. DeepBlue, in Schaeffer, J. and van den Herik, J. (ed.) "Chips Challenging Champions: games, computer and Artificial Intelligence", pp 3-9, Elsevier, Amsterdam, 2002.

[2] Chellapilla, K. and Fogel, D., Evolving neural networks to play checkers without expert knowledge, *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1382–1391, 1999.

[3] *Ibid.,* Evolving an expert checkers playing program without using human expertise, *IEEE Transactions on Evolutionary Computation*, vol. 5, pp. 422 – 428, 2001.

[4] Davis, J. E. and Kendall, G., An Investigation, using Co-Evolution, to Evolve an Awari Player. In proceedings of Congress on Evolutionary Computation (CEC2002), Hawaii, pp 1408-1413, 2002.

[5] Fogel, D., *Blondie24: playing at the edge of AI*. Morgan Kaufmann Publishers Inc., 2002.

[6] Kendall G., Yaakob R. and Hingston P., An Investigation of an Evolutionary Approach to the Opening of Go. *In proceedings of Congress on Evolutionary Computation 2004 (CEC'04)*, Portland, Oregon, 20-23, pp 2052-2059, 2004.

[7] Lucas, S.M., Evolving a Neural Network Location Evaluator to Play Ms. Pac-Man, *IEEE Symposium on Computational Intelligence and Games (2005)* , pp. 203- 210, 2005.

[8] Lucas, S.M. and Runarsson, T.P., Temporal Difference Learning Versus Co-Evolution for Acquiring Othello Position Evaluation. *IEEE Computational Intelligence and Games (2006)*, pp. 52-58, 2006

[9] Pollack, J.B. and Blair, A.D., Co-evolution in the successful learning of backgammon strategy. *Machine Learning*, 32:225–240, 1998.

[10] Runarsson, T.P.; Lucas, S.M., Coevolution versus self-play temporal difference learning for acquiring position evaluation in small-board Go, *IEEE Transactions on Evolutionary Computation*, vol.9, no.6pp. 628- 640, Dec. 2005

[11] Tesauro, G., Temporal difference learning and TD-gammon, *Journal of the ACM,* Vol 38, No. 3, pp 58-68, 1995.

[12] Yoshioka, T, Ishii, S. and Ito, M., Strategy acquisition for the game "Othello" based on reinforcement learning, *IEICE Transactions on Information and Systems* E82-D 12, pp. 1618–1626, 1999.