Artificial Intelligence as a Research Tool: (Let's not go around in circles)

Jim Cater
Computing and Information Technology
Manukau Institute of Technology

An idea for an abstract strategy game was developed in 1987, but the game never quite 'clicked'. In 2003, using artificial intelligence techniques originally developed by the author in the 1970s, and applied in a chess-playing program, a computer test bed was written, to test subtle and not-so-subtle changes to the rules. Using the test bed, a major deficiency was identified. In a series of "eureka" moments, a new dimension was added to the game. Extensive testing with the test bed and with human-computer and human-human contests indicates that a game with good potential has evolved. The testing process has also revealed some aspects of minimax tree searching which are worthy of further investigation.

1. INTRODUCTION

This paper describes the use of artificial intelligence as a tool in the evaluation and refinement of an abstract strategy game.

The author developed an idea for a two-player board game in 1987. The idea was developed to the point where several prototype sets were made (by hand) and many games were played between volunteers at the time.

Despite the novel nature of the game, it never quite met the author's expectations in terms of playability.

In December 2002, the author resolved to test the worth of the game by writing a software test bed. The test bed delivered its verdict, and the reasons for the negative verdict were analysed.

In a series of "Eureka" moments, enhancements were invented, and tested both over-the-board and on the test bed. The resulting game has been developed as a commercial product and sold under the name "Trinity".

The use of artificial intelligence as a tool for evaluation and refinement of abstract strategy games has been shown to be extremely valuable.

2. THE ORIGINAL CONCEPT

The original concept was a grid of squares with a number of identical playing pieces. Each of the pieces was to occupy a single square on the grid, and to "attack" one other square on the grid. The attacking concept here is similar to that of the king being attacked (checked) in chess.

The chosen dimensions were a four-by-four grid and four pieces for each player. Each of the squares of the grid also contained a four-by four-grid. Each piece would be placed on one of the resulting 256 squares. If three pieces "attacked" a square containing a piece then the player whose piece was attacked would lose the game.

Various moving rules were tried, and the one chosen was that a piece could move one square in any direction, and attack any square.

It soon became obvious that keeping track of the attacks was quite difficult.

An enhancement was made to address this issue: Each of the four pieces would become a pair of pieces, the placement of the two representing firstly the placement of the original single piece and secondly the square that was attacked. The shapes of the pieces were designed to reflect the idea of a piece being attacked by up to three other pieces. The names ultimately adopted for the pair of pieces was "orb" and "arc". Each player had four orb-arc pairs, each pair being related by a splash of red, yellow, green or blue.

Games played by the author and others, while initially challenging, became a little boring when one developed sufficient expertise to avoid loss.

The game was shelved.

3. THE SOFTWARE TEST BED

In the late 1970s, the author had successfully created a chess playing program which ran on the ICL 1900 series mainframes. The program was developed to a reasonable standard of proficiency, tested against many colleagues who were chess players, including a New Zealand representative player. (The author regrets not furthering this development, but it could be a future project.)

In December 2002, the game once again came under scrutiny by the author, and the idea of writing a software test bed as an evaluation and refinement tool was envisaged and adopted.

Using the artificial intelligence techniques developed for the chess program, the test bed was written and testing commenced in January 2003.

4. THE TESTING PROCESS

The software was set up so that the computer could play against "itself" or a human.

Tree searching using pruning techniques and an adjustable evaluation heuristic resulted in a range of playing abilities from "easy to beat" to very competent. The program was set to have an adjustable depth of analysis, with adjustable search breadth. Sorting was used at every node to optimize the benefits of pruning.

The first stage of testing involved making changes to the rules to see what the effect would be. The outcomes of two such tests are worthy of mention.

1. An arc is required to move when its paired orb moves. When this requirement was removed, an important tactical subtlety was lost. Forcing an arc to move by attacking the related orb is an important dimension.

2. An arc can move anywhere on the board (except to a square containing a "friendly" orb). When this requirement was removed (and the arc required to stay within one square of its related orb), many tactical possibilities were lost.

When the initial testing was concluded, the main tests were commenced. The objective of the tests was to evaluate the extent to which games resulted in a "draw" because neither player could achieve a victory, when the players were of similar playing ability.

The paradigm adopted for testing was to set the computer to play against itself with various depths of analysis (simulating different playing strengths) and the number of moves to victory or draw by repeated moves was tabulated.

The results confirmed that two players of similar playing strength would usually draw a game, either by repeating sequences of moves, or by playing games which were judged too long.

5. OTHER GAMES

The author undertook an analysis of popular games in the strategy genre: chess, draughts (checkers), go, reversi (othello), and backgammon. The objective was determine if there was any fundamental difference to the game being analysed which may be causing the undesirable outcomes.

The analysis revealed a characteristic common in all the popular games: "irreversible moves".

The author set out to include such moves in the game.

6. EUREKA

In order to address the lack of irreversible moves, considerable thought was given to the game, and to situations in the game which could result in irreversible moves.

A third playing piece was added to the game, which would be introduced onto any square which came under three simultaneous attacks by arcs of one player. The piece endowed "ownership" of the square to the player for the rest of the game. The square became a sanctuary square for the player and a no-go square for the opponent. The new piece is a "halo".

Thus was introduced a concept of territory (squares), which could be won by either player. The restriction on movement made it easier to mount the required triple attack by arcs on one of the opponent's orbs.

The downside of the new piece was that it created a hiatus in the power struggle while the arcs were redeployed in a more attacking mode.

In a "eureka" moment, the problem was solved by allowing automatic follow-on moves by the three arcs to attack the opponent's orbs of the same colour (red, yellow, green or blue).

At the same time the concept of territorial acquisition was extended to give special value to territorial occupation, and the main aim of the game was altered. As well as winning by having three arcs attack one of the opponent's orbs, it is now also possible for a player to win by moving all orbs into halos. This latter winning configuration supersedes the other for more proficient players.

7. TESTING

Using the test bed with new rules and pieces, the previous experiments were repeated. The tests confirmed that a drawn game would occur only very rarely. Games played by humans against the computer and other humans have resulted in very positive feedback on the level of challenge and enjoyment generated.

Extensive analysis of some games has revealed that it is sometimes hard to see that one is losing, because of the deep analysis required to prove it, but a proficient player develops a "feel" for a position much as a strong chess player does.

8. COMMERCIALISATION

The author has decided to develop the game as a commercial product, marketed under the name "Trinity". As a novel feature, a program derived from the test bed is included with the board game so that players can test and hone their skills against a PC. A patent for the game concept and a trademark for the name have been applied for.

9. FURTHER ENHANCEMENT

The author considers that the playing strength of the software can be improved, and further work is planned.

One of the interesting conclusions to come out of this work is that if it is possible to have a static evaluation function which always represents the true worth of a position, without requiring extensive tree searching, the game is unlikely to be very challenging. Therefore it is necessary when writing such programs to give a lot of attention to refinement of tree searching algorithms. The author has developed techniques in this area which may be highly original, but which, for the moment, must remain as trade secrets.

The author may have to await development of other trinity-playing software to have his work truly tested.

10. CONCLUSION

The author has been unable to find another instance of artificial intelligence being used as a tool for evaluating and refining an abstract strategy game.

The process described in this paper has demonstrated the success of such a strategy. Particularly noteworthy was the speed at which possible rule changes could be tested, compared with using a panel of human players who would have had to be assembled and instructed. It is very probable that the results obtained were more reliable, as the software did not overlook a forced win (or loss) so long as it came within the horizon of analysis.