ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/276412172

Use of Evolutionary Algorithms to Play the Game of Checkers: Historical Developments, Challenges and Future Prospects

Conference Paper · January 2014

DOI: 10.13140/RG.2.1.1039.8880

CITATIONS
0

READS
247

1 author:

Amarjeet Singh
Indian Institute of Technology Roorkee

15 PUBLICATIONS 92 CITATIONS

SEE PROFILE

All content following this page was uploaded by Amarjeet Singh on 21 June 2015.

The user has requested enhancement of the downloaded file.

Use of Evolutionary Algorithms to Play the Game of Checkers: Historical Developments, Challenges and Future Prospects

Amarjeet Singh and Kusum Deep

Abstract The objective of this paper is to study the historical development of computer programmers for playing the game of checkers. Since the game-playing is a NP-hard problem, it would be interesting to use evolutionary algorithms to solve them. The question is can a programme be developed which can beat humans with complete success, it may appears that some challenges may also be formed which may substantiate the argument of the paper. Further, these challenges also form a part of this study.

Keywords Checker game · Evolutionary algorithms · Chinook

1 Introduction

Checkers is one of the world's oldest, two players board game, played on square checker board which consists 64 light and dark squares. Checkers is widely played in the United States and the British Commonwealth. Each player has 12 playing pieces which are disk-shaped. Normally, the colours of pieces are red and white.

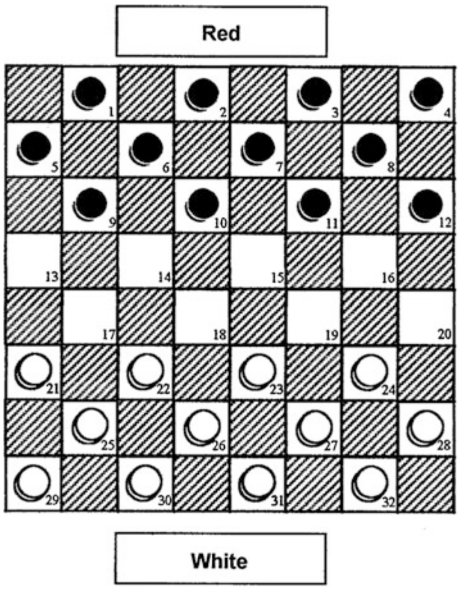
In the beginning of the game, each contestant has 12 pieces arranged on the board. The red pieces occupy square 1–12 and the white pieces occupy 21–32. Red moves first, players take their turn by advancing a piece diagonally forward to an adjoining vacant square. If an opponent's piece is in an adjoining diagonally vacant square, with a vacant space beyond, it must be captured and removed by

A. Singh (✉) · K. Deep
Department of Mathematics, Indian Institute of Technology Roorkee,
Roorkee 247667 Uttarakhand, India
e-mail: amarjeetiitr@gmail.com

K. Deep
e-mail: kusumdeep@gmail.com

M. Pant et al. (eds.), *Proceedings of the Third International Conference on Soft Computing for Problem Solving*, Advances in Intelligent Systems and Computing 259,
DOI: 10.1007/978-81-322-1768-8_5, © Springer India 2014

Fig. 1 Checker board



diagonally jumping over it to the empty square. If this square presents the same situation, successive jumps forward in a straight or zigzag direction must be completed. If there is more than one way to jump, then player has a choice. If a piece reaches the king row (last row) then it can move backwards also. A player will win when an opponent's all pieces are captured or blocked. A game is declared draw, when neither side can force a victory nor the trend of play becomes repetitive (Fig. 1).

2 Historical Developments

The development of board game programs started during and after Second World War, the original work began on chess. The first work on chess was done Shannon [1]. In 1950, a paper on computer chess entitled "Programming a computer for playing chess" by Shannon was published, which describes how a machine or computer could be made to play a game of chess. Minimax procedure, based on evaluation function of a given chess position, was used. In evaluation function, the value of the black position was subtracted from that of the white position. Material was counted as 1, 3, 5 and 9 points for pawn, knight or bishop, rook and queen respectively. Some positional factor, subtracting 0.5 point for each doubled pawn, backward pawn and isolated pawn and mobility was added with 0.1 point for each legal move available. He gave the king artificial value of 200 points and used only checkmate to capture of the king.

Dietrich G. Prinz, colleague of Alan Turing, developed the first limited chess program in 1951. The computer, Ferranti Mark 1, could not play full game powerfully but was able to find best moves if it was only two moves away from checkmate.

Alan Turing, the father of modern computing, experimented machine routing for playing chess. Turing had no computer to run his heuristic chess program. Turing simulated the operation of the program by hand, using paper and pencil. The output of this program was not so good but his work was appreciated as this was the first time somebody defined rules to play a game like chess.

Samuel [2] wrote the first checkers program for IBM 701 and learning program in 1955 and discussed two procedures: (1) a rote learning procedure in which a record was kept of the board situation encountered in actual play together with information as to the results of the machine analysis of the situation; this record could be referenced at terminating board situations of each newly initiated tree search and, in effect, allow the machine to look ahead further than time would otherwise permit, (2) a generalization learning procedure in which the programme continuously re-evaluated the coefficients for the linear polynomial used to evaluate the board positions at the terminating board situations of a look-ahead tree search. Minimax algorithm is used to backup scores assigned to the terminating situations and to select the best move and assumed that opponent would also apply the same selection rules on his turn. The rote learning procedure, learn continuously but was very slow and effective in the opening and end game phases of the play. Two copies of his program played against each other and the weaker program was eliminated, and produced a new second copy. After repetition of this process, a stronger program was produced. In 1962, Samuel's program beat Robert Nealey, a blind checkers master, but lost 8 games in a row in 1966 against Walter Hellman and Derek Oldbury.

The shortcomings of this program may be highlighted as follows:

1. The incorrectness of the assumption of linearity which underlie the use of a linear polynomial.
2. The inadequacies of the heuristic procedures used to prune and to terminate the tree search.
3. Slowness of the learning procedure.
4. The absence of any strategy considerations for altering the machine mode of play in the light of the tactical situations as they develop during play and
5. The absence of an effective machine procedure for generating new parameters for the evaluation procedure.

The signature table technique was also introduced by Samuel [3] to express the observed relationship between parameters in subsets; values read from the tables for a number of subsets are then combined for the final evaluation.

In 1970, at Duke University [4], a checkers program was written by Eric Jensen, Tom Truscott and Alan Bierman, which bet Samuel's program in a 2-game match, and went on to lose against grandmaster Elbert Lowder in a 5-game match with 2 losses, 2 draws and a win.

In 1989 and 1990, three strong programs emerged during the first and second computer Olympiads that was held in London (1) Colossus (Martin Bryant) (2) Checkers (Gil Dodgen) (3) Chinook (Jonathan Schaeffer et al.). The programs Colossus and Checkers, commercially available, were aimed at PC market and both were ranked among the top twenty players in the world that time, while Chinook was a research project. Jonathan Schaeffer made a team of people to work with him on Chinook. Norman Treolar was, checkers team expert, responsible for the evaluation function and the opening book; Robert Lake was in charge of the endgame databases. Later on, Martin Bryant supplied a very big and very good opening book for Chinook. Endgame databases, distance to win database, introduced in Chinook, which gave the computer perfect knowledge for all positions with 8 pieces or less on the board of the form win/loss/draw. Chinook uses some fixed predefined strategies for game start and game end section. In the middle of the game, Chinook uses evaluation function in which expert knowledge was captured. Chinook was based on game tree and alpha–beta search (traditional game theory mechanics) and expert knowledge. The win/loss/draw scheme is interesting, because it has to store less information and therefore you can compress these databases much more. The 8-piece database turned out to be about 6 GB in compressed form.

Initially, the American Checker Federation and English Draughts Association, the governing bodies for checkers, did not allow the Chinook-Tinsley (world champion) match and said that the world championship was for humans, not for machines but at the end, they agreed to allow the match and gave the title "Man–Machine World Championship" to that match.

In 1992, Chinook-Tinsley match was played but Chinook lost with 2 wins and 4 losses [5–7]. In 1994, after introducing the 8-piece database, a rematch was started. Due to bad health, Tinsley confiscated the match after 6 drawn games, he died shortly afterwards. Chinook went on to beat Don Lafferty, the second best player in the world after Tinsley in 1995 in a very close match (1 win and 31 draws), and finished clear first far ahead of the field in the 1996 national tournament in the US.

Researchers, working in the field of artificial intelligence, want to create intelligent game program which is capable of defeating human experts. Several approaches have been proposed for different games including neural networks for Backgammon, special purpose hardware called Deep Blue for chess and the application of expert knowledge with relatively small computational power for checkers. Recently introduced methods to play the checker game, mostly based on a learning theory to generate an intelligent agent, and learn how to play without prior knowledge. Mostly, these techniques depend on expert knowledge and training evaluation function, relevance factor for the evaluation, the weight of the evaluation factors, opening knowledge, and an end game database.

Chisholm and Bradbeer [8] used genetic algorithm to optimize the board evaluation function of a game playing program and a pool of checkers programs was played against each other in a round-robin tournament to evaluate the fitness of each player and genetic algorithm was used to preserve and improve the best performance. One of the greatest achievements in the realm of checkers game is Blondie24.

Chellapilla and Fogel [9], in which after injecting little expert knowledge into the algorithms, Blondie24, was able to play a game of checkers at the human expert level. To find the potentially good moves, Fogel used a minimax search tree as a look ahead and evolution strategies with neural networks. In Blondie24, the strategies do not all play the same number of game because some would be selected as opponents more often than others.

Franken and Engelbrecht [10] investigated the effectiveness of various particle swarm optimizer structure to learn and how to play the game of checkers and used co-evolutionary techniques to train the game playing agents, performance was compared against the player making moves at random. In co-evolutionary process, population of game playing agents consist a neural network and genetic makeup, translate to the neural network weights and updates to produce stronger game playing agent based on the evolutionary strategy. During each play's move, a single ply depth game tree was constructed and NN was used as an evaluation function for the leaf node of the game trees. After taking different swarm sizes, varying hidden layers sizes, comparison in performance was made and the highest scores of the agent were 82.4.

Hughes [11] used an on-line evolutionary algorithm to co-evolve move sets for both players in the game, playing the entire length of the game tree for each evaluation and chromosome was defined such that each chromosome has 100 genes, each per move. Each gene, a real value in the range [0,1), was decoded into a move by multiplying the real value by the total number of currently available moves, rounding the nearest integer and using this as an index for a single move from the list of available moves.

In "Evolutionary approach to the game of checkers" by Kusiak et al. [12] presented a new method of genetic evolution of linear and non linear evaluation functions in the game of checkers, in which evolutionary heuristic generators in two player games domain was evaluated using the game of US checkers. Twenty five components, based on basic board features, were used in the evaluation functions. Linear and non linear heuristic evaluations were considered. Each linear heuristic consisting of linear combination of the parameter set and non linear heuristics were composed of a small number of IF-conditions which divided the entire game into disjoint stages. At each stage, linear combination of parameters was considered and the coefficients of linear combination were optimized by evolutionary process.

Dura and de Oliveira [13] used board features such as material strength, piece mobility, pawn structure, king safety and control of the centre in parameterized board evaluation function whose weight are optimized using PSO and through simulation result shows that PSO can provide faster learning results than simulated annealing under similar experimental conditions, especially in the presence of bounded computing time.

Al-khateeb [14], in his Ph.D. thesis, introduced a round robin tournament into the evolutionary phase of evolutionary checkers program to eliminate the randomness and to enhance its learning ability. Individual and social learning were also introduced and n-tuple systems into evolutionary checkers produced a good player which learn faster and take less computational time in comparison to other approaches. Piece difference has a significant effect on learning abilities, shown through experiments. Thirty feed forward neural network players was played against each others for 140 generations and obtained best player was tested against an evolutionary program based on Blondie24.

"Immune based fuzzy agent plays checkers game" by Cheheltani and Ebadzadeh [15] discussed immune based approach, in which permanent memory cells cause to omit the process of learning for any played strategy and consequently increase the speed of decision making process. In this method, memory cells represent actions that have the best local payoff, for that current state of the game and are generated simultaneously by learning process. Because of these cells decision making system decide better, considering the previous and future state of the game.

Based on Fogel's work to evolve Blondie24, Al-Khateeb and Kendall [16] evolved different checkers players. Through experiments, many checkers players were produced of various depths of ply during learning. Through three different experiments, it was shown that players with higher ply perform better than those with a lower ply, when playing at fixed ply of six, players trained with higher plies also performed better than the players trained with a lower ply. When playing at the ply that they were trained at and players trained on higher ply perform significantly worse when playing at a lower ply. Also results show that increasing the depth level by one will give a different performance depending upon the level.

Generally, the games draw; when the world's best players play the game of checkers. To avoid this and make the game more competitive, two-move ballot is used. In which first move of both players are discussed. There are 49 possibilities to play first two moves. After a lot of research on first two moves, it is found that 6 out of 49 are unbalanced, as it gives an advantage to one side over the other. So, only 43 moves are considered. A card is randomly chosen, showing which of 43 moves is to be played. The original game, with no forced opening moves, is called Go-As-You-Please (GAYP).

Rating of checkers players is calculated by standard system formula, in which initial rating for a player R_0 is 1,600.

Rating of opponent:

$$R_{new} = R_{old} + C(outcome - W).$$

Where outcome: outcome of match 1 for a win, 0.5 for a draw and 0 for a loose.

R_{new}, R_{old} are new rating for the player and current player's rating and is calculated by the formula:

$$W = 1/(1 + 10^{((R_{opp} - R_{old})/400)}),$$

R_{opp}: Opponent's rating and C is 32 for ratings less than 2,100, 24 for rating between 2,100 and 2,399, and 16 for ratings at or above 2,400.

A player is called expert, master and senior master if its rating are in 2,000–2,199, 2,200–2,399 and 2,400+ respectively; and class A, B, C,…, J if the player has rating in 1,800–1,999, 1,600–1,799, 1,400–1,599,…, below 200.

3 Conclusions and Future Challenges

The discussion in the context reveals the fact that people often enjoy playing board games because it does not only place some intelligent challenge before the player, it also evinces a sense of satisfaction after playing well. People use knowledge and search to make their decisions during this process. Without perfect knowledge, mistakes are made and even world champions occasionally loose the match.

Many matches were played between checkers program and checker masters and in which checker master have been defeated however the checker game is not perfectly solved i.e. there is no checker program that may always win whenever be the match being played. It is a challenge to develop a tireless, robust, unbeatable Checker player program, which is available to the user at any time to play.

With the advent of some new nature inspired optimization techniques being invented, there is a wide scope of research in this area where the newly developed nature inspired optimization techniques can be used to design program for playing board game like checkers.

References

1. Shannon, C.E.: Programming a computer for playing chess. Phil. Mag. **41**(314), 256–275 (1950)
2. Samuel, A.L.: Some studies in machine learning using the game of checkers. IBM J. Res. Dev. 210–229 (1959)
3. Samuel, A.L.: Some studies in machine learning using the game of checkers. II—recent progress. IBM J. Res. Dev. **11**(6), 601–617 (1967)
4. Fierz, M.: A brief history of computer checkers. http://www.fierz.ch/history.htm
5. Schaeffer, J.: Man versus machine: The silicon graphics world checkers championship, technical report 92-120, Department of Computing Science, University of Alberta (1992)
6. Schaeffer, J., Treloar, N., Lu, P., Lake, R.: Man versus machine for the world checkers championship. AI Mag. **14**(2), 28–35 (1993)
7. Schaeffer, J., Lake, R., Lu, P.: Chinook the world man-machine checkers champion. AI Mag. **17**, 21–30 (1996)
8. Chisholm, K.J., Bradbeer, P.V.: Machine learning using a genetic algorithm to optimise a draughts program board evaluation function. In: IEEE International Conference on Evolutionary Computation, pp. 715–720 (1997)
9. Chellapilla, K., Fogel, D.B.: Evolving an expert checkers playing program without using human expertise. IEEE Trans. Evol. Comput. **5**(4), 422–428 (2001)
10. Franken, N., Engelbrecht, A.P.: Comparing PSO structures to learn the game of checkers from zero knowledge. In: The 2003 IEEE Congress on Evolutionary Computation, CEC'03, vol. 1, pp. 234–241 (2003)

11. Hughes, E.J.: Checkers using a co-evolutionary on-line evolutionary algorithm. In: The 2005 IEEE Congress on Evolutionary Computations, vol. 2, pp. 1899–1905 (2005)
12. Kusiak, M., Walędzik, K., Mańdziuk, J.: Evolutionary approach to the game of checkers. In: Adaptive and Natural Computing Algorithms, pp. 432–440. Springer Berlin Heidelberg (2007)
13. Duro, J.A., De Oliveira, J.V.: Particle swarm optimization applied to the chess game. In: IEEE World Congress on Computational Intelligence, IEEE Congress on Evolutionary Computations, pp. 3702–3709 (2008)
14. Al-Khateeb, B.: Investigating evolutionary checkers by incorporating individual and social learning, N-tuple systems and a round robin tournament. PhD Dissertation, University of Nottingham, UK (2011)
15. Cheheltani, S.H., Ebadzadeh, M.M.: Immune based fuzzy agent plays checkers game. Appl. Soft Comput. **12**(8), 2227–2236 (2012)
16. Al-Khateeb, B., Kendall, G.: Effect of look-ahead depth in evolutionary checkers. J. Comput. Sci. Technol. **27**(5), 996–1006 (2012)
17. Schaeffer, J.: One Jump Ahead: Computer Perfection at Checkers. Springer, New York (2009)

View publication stats