



## RESEARCH ARTICLE

### A NEW SOLUTION FOR CHECKERS USING MINIMAX AND ALPHA BETA PRUNING

\*Gayathri. P., Santhi, H., Geraldine Bessie Amali, Gopichand, G.

School of Computer Science and Engineering, VIT University, Vellore, Tamil Nadu, India

#### ARTICLE INFO

##### Article History:

Received 09<sup>th</sup> August, 2017  
Received in revised form  
27<sup>th</sup> September, 2017  
Accepted 04<sup>th</sup> October, 2017  
Published online 30<sup>th</sup> November, 2017

##### Keywords:

Checkers,  
Alpha-Beta Pruning,  
Depth first search,  
Games.

#### ABSTRACT

In this paper, the sport of checkers has been selected to know the pattern of game development with comparison to alternative games like twitching Tac Toe, chess, four Queens and plenty of a lot of. Since there are numerous prospects to adapt the technique to play this game, we have a tendency to set to travel beside Alpha – Beta pruning. This approach is that the desired one since it saves time and provides house in doing therefore that area unit each nice benefits for any game. The objective of this project is to make one player v/s computer checkers game that's fun to play implementing the logic of checkers victimization Alpha – Beta pruning.

#### INTRODUCTION

In checkers game each the players is given with distinction items. They act moving their items. Players move their pieces diagonally from one square to a different square. Once a player jumps over their opponent's (the alternative player's) piece, you are taking that piece from the board. If you'll be able to take a bit, then you want to take a bit. Here we have a tendency to take the board to be 8x8 and variety of items is 8 on all sides. We specialize in the substitute Intelligence or AI for this game. We will predict moves up to two to three moves. We used Minimax algorithmic program beside Alpha-Beta pruning procedure. There are many ways that and approaches to unravel this downside however we have a tendency to selected this technique as a result of it saves time. Rest of the paper is organized as follows: section 2 contains literature review. Section 3 contains proposed model and Section 4 presents the conclusion.

#### Literature Review

The development of gambling has been vital to the refinement and testing of the techniques employed in computing. Several of those advances are going to be enumerated here. In [Barr and Feigenbaum, 1981] Russel and Wefald projected tree search algorithms that use metalevel computations to prune the search tree additional potency than the alpha-beta methodology.

\*Corresponding author: Gayathri. P.,

School of Computer Science and Engineering, VIT University, Vellore, Tamil Nadu, India.

Additionally in [Allen, 1989], Korf extended the alpha-beta pruning methodology to be used with multi-player games. A little any back, in [Chellipilla, 2000], Berliner developed an formula that uses interval bounds that additionally improve the alpha beta method. There has additionally been a move removed from numerical analysis functions. Several believe that analysis operate is more practical and time economical if they use evaluation techniques that are nearer to the human thought method. This involves the employment of pattern recognition techniques to decide whether or not one position is healthier or worse than another. this system has been used extensively in chess endgame programs. The CHINOOK program that Jonathan Schaeffer has developed at the University of Alberta has become usually recognized because the world champion of checkers. Also, others like Michie, Lee and Mahajan, and Schraudolph, Dayan and Sejnowski have applied reinforcement learning ways and temporal distinction techniques to gambling. All of those advances discusses would fairly useless if they applied to gambling. However, they'll apply to almost all fields during which programming is employed so as to search out solutions to a drag.

#### Proposed Model

We aim to implement the Minimax Algorithm and Alpha-Beta Pruning along with Transposition tables. We also explain opening and endgame databases. We are never sure what move the player choses to play and accordingly we make the game move forward.

#### Minimax Algorithm

This approach is employed to settle on the most effective doable move by evaluating a move pattern and predicting consecutive set of prospects.

The pc counterpart is selected as GHB whereas the human opponent or that player as we have a tendency to could say is MIN. Since we have a tendency to assume that each doable move will be judged and thought upon whereas conjointly assignment a selected numerical price. The thought behind this is often that whereas the sport is on, if the actual or current move favors MAX, is appointed as positive whereas move favors MIN move is appointed negative value. A tree is formed consisting of all doable moves or solutions for consecutive a part of the sport. The structure followed is:

- The root may be a GHB node.
- All the GHB nodes have MIN nodes as youngsters and every one the MIN nodes have GHB nodes as youngsters.

To create the answer tree every bit is examined and a replacement node is formed for every doable move. once all of the items are examined the main focus shifts to a different node that has not been occupied however. This method continues till the tree reaches a pre-specified depth. once the tree is complete, we want to see the most effective doable move by mistreatment the analysis perform. we have a tendency to assume that MIN can continually build the most effective doable move so we have a tendency to cannot merely realize the trail from the leaf with the very best price back to the basis. to search out the most effective move, the values at the leaves area unit accustomed "back-up" the tree. every MIN node receives the worth of its smallest kid and every GHB node receives the worth of its largest kid. If we have a tendency to continue till we have a tendency to reach the basis, the kid of the basis with the very best price is that the best move. Even though it's terribly clear-cut approach, whereas being true it still has its drawbacks. relying upon the sort of game being resolved, the branching issue could also be high leading to {a terribly|areally|a awfully} massive tree and thus time taken would be very massive that's  $O(n)$  and time taken to copy the tree is  $O(n)$ . This limitation will be corrected mistreatment Alpha-Beta Pruning.

### Alpha-Beta Pruning

In this method we have a tendency to might derive resolution tree by removing unwanted branches off the tree. during this procedure each generation of tree and derivation of resolution ar dealt seperately. By combining these 2 steps the search are often stopped at nodes that can't increase the general price of the position. so as to terminate the search associate degree ALPHA and a BETA price ar maintained. Alpha is associate degree boundary for the values of relevant MIN nodes and BETA is bound for relevant GHB nodes. Search typically begins from ROOT and income until leafs. The alpha and beta values enable US to discontinue the looking out within the following cases:

- A Min node with a BETA price but or equal than the Alpha price of any of the GHB ancestors. This node can pass its BETA price up as its insured price.
- A GHB node with associate degree Alpha price above or adequate to the beta price of any of its MIN node ancestors.

The Alpha and Beta values ar updated throughout the

search, with every GHB node having Alpha because the biggest current insured price of its MIN successors. The Beta price of every MIN node is that the minimum backed-up price of its GHB successors. The best attainable case for Alpha-Beta pruning would have the successors of every GHB node ordered from greatest to least, and also the successors of every MIN node ordered from least to greatest.

### Transposition Tables

Transposition tables are used for storing positions throughout search to avoid researching positions which will be reached by many completely different move sequences. The concept of the transposition table is same as that of the Hash table. In games like checkers or chess, this could speed up the search dramatically, particularly within the finish once the sport nears its conclusion. There is one disadvantage of mistreatment the Transposition tables. they cannot be reliable invariably. To be of any use, it should be doable to seem up positions quickly and with efficiency. This is often achieved through hash keys that function indices within the transposition tables. However the keys aren't distinctive then 2 completely different positions could hash into an equivalent table entry. In follow, however, the likelihood is extremely low and also the risk is usually outweighed by the gains in search time.

### Opening and Endgame Databases

Opening and Endgame databases area unit utilized in order to either provide the pc player a robust begin, or to urge to a win in response to any move that the human player might build. gap and endgame databases area unit accustomed strengthen one's algorithmic program and to assist the pc solve the tougher eventualities. the thought for moves information is to store all potential opening moves and responses to convey the pc the advantage when a such as variety of moves. Endgame information is employed below sure circumstances so as to make sure that the pc will get to a win once the items on the board get into a definite configuration. as an example, several endgame databases can take over once the sport gets all the way down to four items versus 3 items or four versus four items. Currently this configuration might ne'er be reached, in which case the endgame information can ne'er be used. The employment of the information in one's vice algorithmic program could be a good selection so as to construct a radical and sturdy program. For the gap moves database the program would merely use this information to calculate its moves up to five moves, as an example. The endgame information would check at each move the piece estimate the board, and once its specific variety of items was reached it might go.

### Evaluation Function

One of the foremost vital factors once one is attempting to return up with a difficult computer science for a checkers game is that the move analysis operates. This is often wherever the AI chooses that moves square measure higher than others, and ultimately, that is that the best move. The procedure chosen here was to gauge every move and provides it a numerical worth to be came and compared to the remainder of the moves doable from the given base position. After some analysis on move analysis operate, we have a tendency to found that the

final accord on manufacturing a numerical move evaluation operation is that the whole board is searched giving every position a worth supported what reasonably piece is there and wherever that location is on the board. Then all the worth's square measure extra up and therefore the value is came either absolutely or negatively reckoning on that player has a plus within the current scenario. There square measure many factors to be thought-about in bobbing up with a worth for a move. First, because the operate moves the board counters keeps track of variety the amount the quantity of black and red pawns (normal pieces) in addition because the number of black and red kings. Whichever facet has additional of every sort of piece can receive AN addition to its final total, with kings being weighted over pawns.

Pawns are weighted by however so much they're from being promoted. The operate adds to the ultimate score reckoning on however so much pawns that square measure already a part of the approach down the board square measure from being promoted. I say "items that square measure already a part of the approach down the board" as a result of we have a tendency to additionally wish to shield the rear row in order that the opponent can not be promoted. that the operate additionally takes under consideration the amount of items within the back row and adds that to the score. Another factor that the operate checks with relevancy kings is however so much a king is from being on a diagonal. Kings that square measure nearer to a diagonal square measure weighted additional. the ultimate factor checked within the analysis operate is whether or not the score is positive or negative to see that player presently has the advantage. According to AN formula, the player that's ahead can additional freely trade items than the one that's behind. Also, so as to present sure things additional priority, every now and then throughout the operate the score are going to be increased by one hundred.

### User Interface

The user interface that Maher, et al. created for their project in 2000 was the basis of our work. We removed all of the code that was related to the socket communications. Our code run implements data abstraction in terms on functions as per C++ language. The number has to be entered for the position of the pawn to be moved and then another input is asked regarding to the destination place and is it is a valid input then it moves the pawn to the next place.

### Conclusion

With current technology, we believe it is possible to weakly solve the game of checkers. A database containing a proof tree for the game may contain as few as 10 (Allis, 1991) nodes small number by today's standards. Strongly solving the game will be much harder, since it requires knowledge of all  $10^{18}$  legally reachable positions in the game. The combination of weakly solving the game with the deep search, end game databases and draw differentiation facilities, it will be possible to build a program that is close to ultra-strongly solving the game in practice. In future, we can make it GUI and make an app. So people can play checkers along with the artificial intelligence or 2 players.

## REFERENCES

- Chellipilla K. and Foegel D.B. 1999. Evolution, Neural Networks, Games and intelligence. Proc.IEEE, Vol 87, No.9, pp1471-1498.
- Chellipilla K. and Foegel D.B. 2000. Anaconda Defeats Hoyle 6-0 in proceedings of congress on evolutionary Computation, July 16-19, La Jolla Marriot Hotel, Callifornia, USA, pp857-863
- Schaeffer, Jonathan 1997. One Jump ahead: Challenging Human Supremacy in checkers. New York: Springer-Verlag.
- Schaeffer, J. 1996. One Jump ahead: Challenging Human Supremacy in checkers. New York: Springer-BERLIN.
- Schaeffer J., Lake, R., Lu P. 1996. Chinook The World Man Machine Checkers Champion. AI Magazine, 17(1), pp 21-30
- Allen, J. D. 1989. A note on the computer solution of connect-four", pp. 134{135 in *Heuristic Programming in Artificial Intelligence: The First Computer Olympiad* (edited by D. N. L. Levy and D.F. Beal), Ellis Horwood, Chichester, England.
- Allis, L. V. 1988. A knowledge-based approach to connect-four. The game is solved: White wins", M.Sc. thesis, Faculty of Mathematics and Computer Science, Vrije Universiteit, Amsterdam.
- Allis, L. V. 1994. Searching for solutions in games and artificial intelligence", Ph.D. thesis, Dept. of Computer Science, University of Limburg.
- Allis et al. 1991. L. V. Allis, H. J. van den Herik, and I. S. Herschberg, "Which games will survive", pp. 232{243 in *Heuristic Programming in Artificial Intelligence: The Second Computer Olympiad* (edited by D.N. L. Levy and D. F. Beal), Ellis Horwood, Chichester, England, 1991.
- Allis et al. L. V. Allis, H. J. van den Herik, and M. P. H. Huntjens, 1993. Go-Mokusolved by new search techniques", pp. 1{9 in *Proc. AAAI Fall Symposium on Games: Planning and Learning*, AAAI Press, Report FS9302, Menlo Park, California, 1993.
- Allis et al. 1994. L. V. Allis, M. van der Meulen, and H. J. van den Herik, "Proof-number search", *Artificial Intelligence*. 1994.
- Anonymous 1979. Anonymous, "Computer Checkers: The Tinsley Challenge", *Personal Computing* (July 1979), 88{89.
- Barr and Feigenbaum 1981. A. Barr and E. Feigenbaum, *The Handbook of Artificial Intelligence*, Kaufmann, Los Altos, CA, 1981.
- Berliner 1980. H. Berliner, "Backgammon computer program beats World Champion", *Artificial Intelligence* 14(1980), 205 - 220.
- Fortman, R. L. 1978. "Duke vs. Lowder", *Personal Computing*.

\*\*\*\*\*