

Chess with more than 2 dimensions

TEAM -AE Algorithm and Analysis Project

Abstract- This document covers some of the traditional methods of playing 3-d chess , mainly emphasizes on structure of 3-d and 4-d chess and to find answer to question like whether is it possible to develop a reasonable AI for these variants.

Index Terms- Kubikschach , Raumschach.

I. INTRODUCTION

This article will help us to learn about how can we visualize multidimensional chess with more than 2 dimensions. We will analyze two traditional variants of 3-d chess and will try to find that which one is strategically more balanced . It's already known that how much labor humanity has put in to finally come up with fantastic chess AI. The most important thing while writing ai for chess is position evaluation function. We need to work on some of the strategies of 3-d chess which we can actually use in our evaluation function. Next thing to remember is that once we increase dimensions , complexity of game increases (total possible moves exponentially increases).

II. KUBIKSCHACH

It's most trivial design of 3-d chess game with structure $8*8*8$. So basically it's like a building of 8 chess board floors where pieces can now even move from one floor to another. Rules are as follows:

- 1) Pieces are exactly same.
- 2) One player has all the pieces on top layer and the other has on bottom layer.
- 3) In one move you can decide whether to move your piece within same layer or you can change layer and move to corresponding square of the other layer. Note- King cannot change his layer.
- 4) If player do not have some pieces on a common layer for 3 consecutive moves then the game is aborted in a draw.

Now let's try to figure out major missing things in this variant. First of all it is very easy for a king to draw a game in loosing position by doing nothing and never attacking the enemy. So most of the game will have no result actually. Now how can one even expect to make AI for a game which actually most of the times does not end up in favor of one player. Let's assume we even tried to develop ai then what will be the complexity of algorithm.

Total possible moves in one turn are approximately of order $512*511/2$ i.e. 130816.

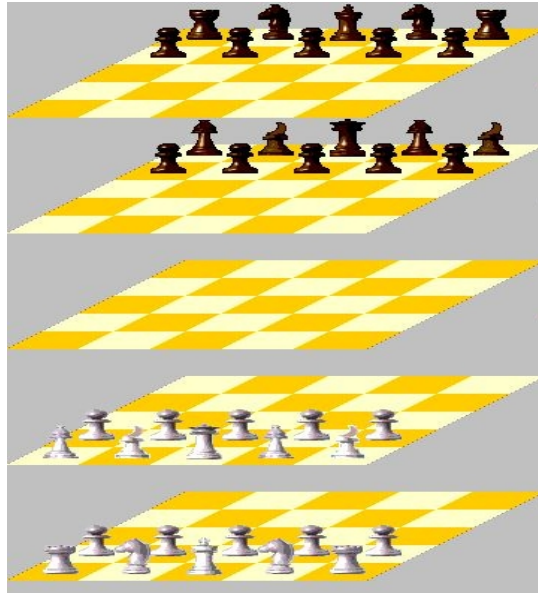
Now if we try to use minimax algorithm for depth 2 then the complexity will turn out to be square of 130816. We are facing a disaster here as we know we cannot allow this much operations for one move only. But any way we know that we have cheap tricks like using game strategies to evaluate a position or using some existing position from our database which actually allows us to not evaluate all the possibilities in mini-max algorithm , but the problem Kubikschach is that there are no actual strategies which you may use in evaluation function . There is no theory about this variant that what positions may lead to your advantage. All of this simply is not possible because of the complexity and uncertainty of the game. So to be honest I believe that the most that we can do here is just choosing a random move or any move which defends your most needed pieces.

III. RAUMSCHACH

Ferdinand Maack developed Raumschach in 1907. He tried all the possible structure of 3-d chess boards and finally realized that the most balanced game can be made by using $5*5*5$ chessboard. Unlike $8*8*8$ This is not as easy and uncertain like Kubikschach. Let's first see the rules of the game which are given as follows:-

A. Pieces And Initial Board Setup

It has all the pieces that normal chess has except it has two extra pawns for each color and a new piece called Unicorn, 2 for both colors. The initial setup is as shown in the figure below



B. How the pieces move:-

- A rook moves through the six *faces* of a cube in any rank, file, or column.
- A bishop moves through the twelve *edges* of a cube.
- A knight makes a (0,1,2) leap (the same effect as one step as a rook followed by one step as a bishop in the same outward direction) enabling it to control 24 different cells from the board's center.
- A unicorn moves in a manner special to a 3D space (i.e. triagonal movement) through the *corners* of a cube, any number of steps in a straight line.
- The queen combines the moves of a rook, bishop, and unicorn.
- The king moves the same as the queen but one step at a time.
- A pawn, as in chess, moves and captures always forward toward the promotion rank. This includes moving one step directly upward (for White) or downward (for Black), and capturing one step diagonally upward (White) or diagonally downward (Black), through a front or side cube *edge*. In Raumschach there is no pawn initial two-step move (and consequently no capturing *en passant*), and no castling.

I was playing this game online with other random people for past few days and I have found some of the crucial strategies that we have to keep in mind when making evaluation function for the same. So the first thing that I realized is the relative importance value of all pieces which according to me is:-

- King = infinity
- Queen = 10
- Unicorn = 7
- bishop = 5
- knight = 4.5
- rook = 3.5
- pawn = 1

The next thing is heat map of all pieces. This heat map cannot be fixed anymore and have to dynamic depending on which layer it is.

Also checkmating in this game is not so easy and according to me we need atleast 24-27 valued material (according to my assigned value to each piece) excluding value of kings.

We cannot store whole of the positions in database but yeah atleast we can save some opening upto 3-4 moves in depth-tree data structure (and yes ! the fun fact: it has some cool openings I got checkmated by someone in less than 20 moves).

Okay so our evaluation function is not that good at all. There are other things that matter like the layer in which your king is should have atleast 3-4 pieces defending it. Well it does not matter using this evaluation only we can actually use mini-max algorithm upto 6-7

depth . I know it will not be that strong move but will still be a reasonable one. So we may not have introduced idea for a strong chess engine which could beat kasparov or magnus , but still atleast it could beat majority of people who don't professionally play chess .

IV. 4-D CHESS

Now let's dive into 4-d chess now. First of all , the most important thing is that how do we actually visualize 4-d chess. Some of the people who play 4-player chess on a octagonal board and claim it to be a 4-d chess , which in my opinion is definitely wrong. So how can we visualize a dimension which don't really exist? Answer is simple just make a stack of few 3-d chess-boards. So now the pieces can move in 4 dimensions actually. It's similar to like you can move a piece inside a cube and can now even move it between such similar multiple cubes also. So $8*8*8$ chessboard will be like a stack of eight $8*8*8$ cubes. Now on we want to make ai for this game also. First of all we want to make our game balanced so that there are some strategies which matters in winning the game in 4-d chess. So I if you try many different sizes of the board then you will realize that the best one is the $5*5*5*4$ chessboard, which means that it is a stack of four $5*5*5$ chess pieces . The rules and pieces are very much similar to Raumschach. The only difference is that now the pieces have one more dimension to move. Pawns can move forward directly to the next cube also. Well on observing the flow of game it can be easily be realized that there is no need of applying methods like mini-max algorithm. We just have to check all next possible moves. The reason behind this is that now the pieces can even change their cube also and there are only 4 cubes. So you only need to be worried about defending or attacking the pieces in just next move only as there are lot of ways in which you can escape from checkmate or save your piece. So in this game I believe that the checkmate is only possible if one of the player seriously blunders . Initially both players will loose some of their pieces really fast. But once when there are less pieces then the game is all about not blundering. If you move your piece to a safe square in next move there 70 percent of chances that it is not attacked in next opponent's move. But to deal with rest 30 percent possibility we will do a 2 or 3 level depth mini-max or alpha-beta pruning to decide a move for ai. Yeah there are also some changes in the evaluation function which are that specially in this variant knight really becomes very useful piece and is actually more important strategically than even queen.

V. CONCLUSION

I think that we can easily develop a reasonable ai for 3-d and 4-d chess but the how strong our ai is still in question. There in nothing wrong with our proposed algorithm of alpha-beta pruning but the problem is in writing a good evaluation function . It is not clear and there is not much theory about any positions and strategies of these variants of chess. We just considered few strategies in 3-d chess but there can be actually more which no human has found till now as the complexity of game is increased by many folds in these variants.

REFERENCES

- [1] Wikipedia
- [2] chessvariants.com