Homework Assignment 2

Kavya Kotian

March 10, 2019

1 In the problem of building railway, pieces don't fit exactly but allow for up to 10 degrees of rotation to either side of the "proper" alignment. Explain how to formulate the problem so it could be solved by simulated annealing.

1.1

States can be in any configurations of 32 pieces. The 2 main operations are:

- Changing the position of a piece
- Rotation of a piece

Simulated annealing consists of random walk and hill Climbing. It allows a lot of random moves in the beginning and gets stricter with Time. It considers the next steps with respect to probability of e^{kT}
We can take our heuristics as a sum of :

- Angle of rotation more than 10 degrees
- Tracks which don't fit correctly

The probability for simmulated annealing can be given by $e^{Delta(R)}$
where Delta(R) is the difference between the next state and the current state.
R = (sum of distortion in angle of rotation + total number of misfit tracks)

2.1

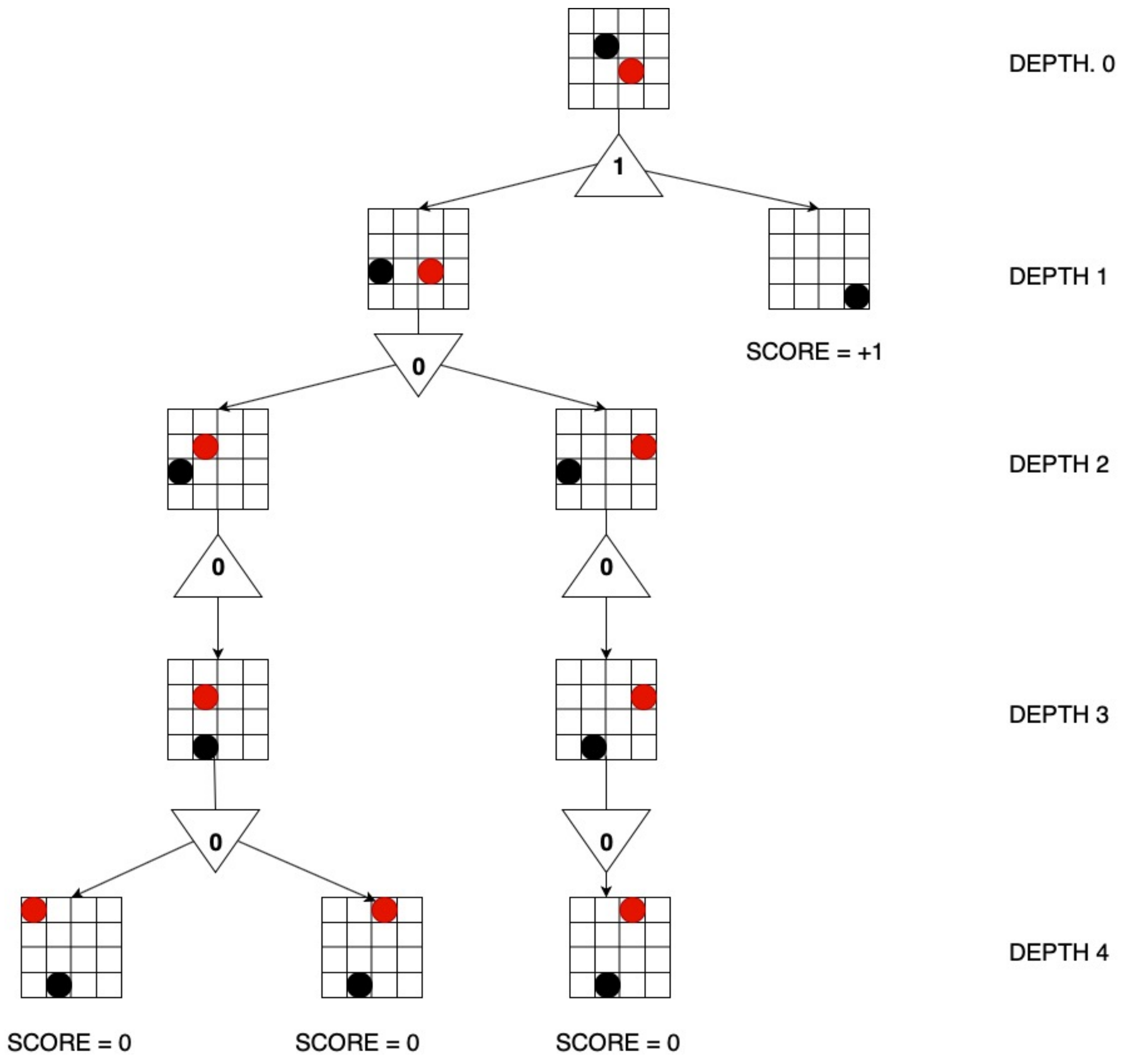

Figure 2.1: Checkers Min-Max Tree

3 Problem of placing K knights on an N × N chessboard such that no two knights are attacking each other, where K is given and K ≤ N^2

3.1 Constraint Satisfaction Problem (CSP) variables

X is a set of variables which consists of 'K' knights $\{X_1, ..., X_k\}$.
Y is a set of variables which correspond to the location on the N x N board$\{Y_1, ..., Y_k\}$.

3.2 Possible values of each variable

Variable X can store the location of the knight $\rightarrow$\{X co-ordinate, Y co-ordinate\}
Variable Y can store if the location $\rightarrow$\{ occupied, empty\}

3.3 Variable Constraint

To check if 2 Knights X_1, X_2 attack each other: If Location Y_2 belongs to the List of locations found on addition of the set $\{(1,2),(-1,2),(1,-2),(-1,-2),(2,1),(-2,1),(2,-1),(-2,-1)\}$ with Location Y_1

3.4 Problem of putting as many knights as possible on the board with- out any attacks. Explain how to solve this with local search by defining appropriate ACTIONS and RESULT functions and a sensible objective function.

- Assign n$sizeofboard$ knights randomly on the board
- Use minimum-conflict heuristic to update the position of the knight such that no knight attacks the other
- Check using minimum-conflict heuristic if you can add more knights

4

$X_n \rightarrow$ #rows, #columns, or diagonals with exactly n X's and no O's.
$O_n \rightarrow$ #rows, #columns, or diagonals with exactly n O's and no X's.

Position	**Utility Function Value**
any position with X3 = 1	+1
any position with O3 = 1	1
All other terminal positions	0

For nonterminal positions,
$Eval(s) = 3X_2(s) + X_1(s) - (3O_2(s) + O_1(s))$.

4.1 How many possible games of tic-tac-toe are there?

Taking into account our player is X, let us consider all the possibilities in which X wins -
X wins if it occupies all the spots in a row, column or a diagonal. Possibilities = 8
Total 8*3! Possibilities
A player can win in minimum 5 moves. 5 moves consist of 3 winning 'X' moves and 2 'O's in one of the remaining spots

- Player wins in 5 moves = 8*3!*(6*5) = 1440
- Player wins in 6 moves = 8*3!*(6*5) = 5760 possibilities and
  number of cases eliminated =6*3!*2*3!= 432 moves
  5760 - 432 = 5328
- Player wins in 7 moves = 47952
- Player wins in 8 moves = 72576
- Player wins in 9 moves = 81792
- Player draws in 9 moves = 46080

Total number of possibilities = 255168

4.2 Show the whole game tree starting from an empty board down to depth 2 (i.e., one X and one O on the board), taking symmetry into account.

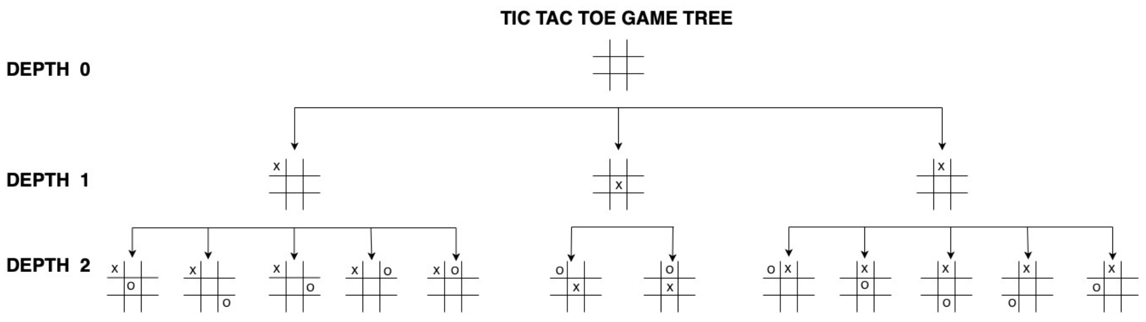

Figure 4.1: Game Tree

4.3 Mark on your tree the evaluations of all the positions at depth 2.

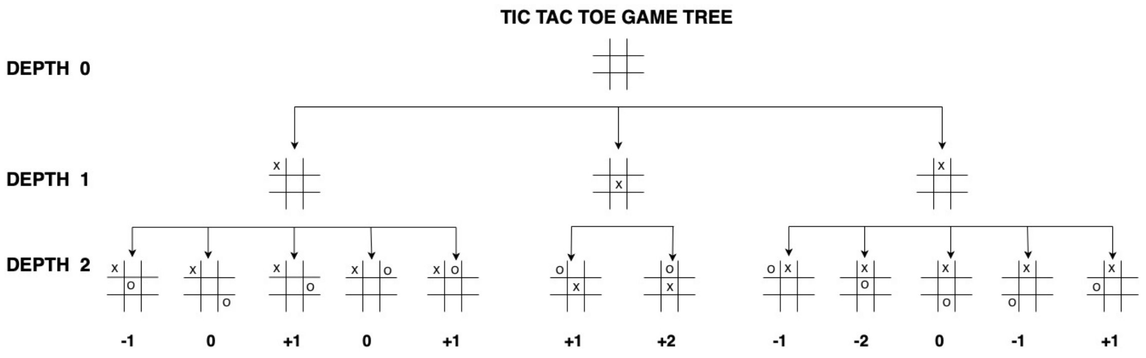

Figure 4.2: Evaluations at Depth 2

4.4 Using the minimax algorithm, mark on your tree the backed-up values for the positions at depths 1 and 0, and use those values to choose the best starting move.

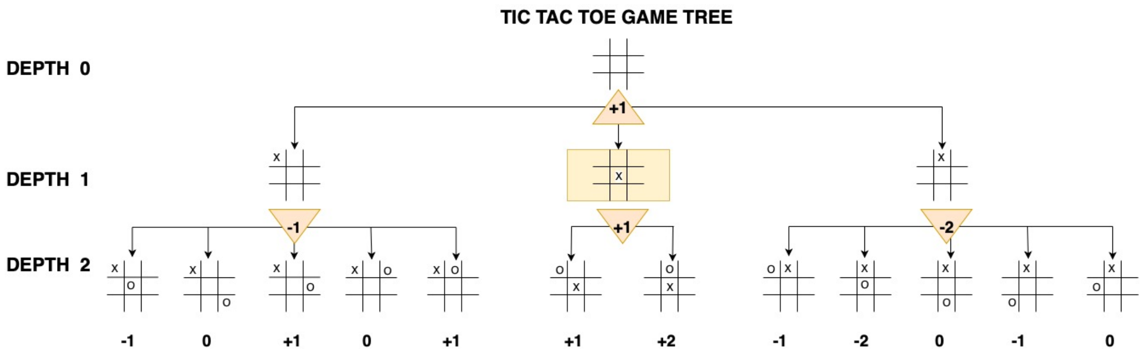

Figure 4.3: Min Max

The Best Starting Move is with X at the center.

4.5 Circle the nodes at depth 2 that would not be evaluated if alpha–beta pruning were applied, assuming the nodes are generated in the optimal order for alpha–beta pruning.

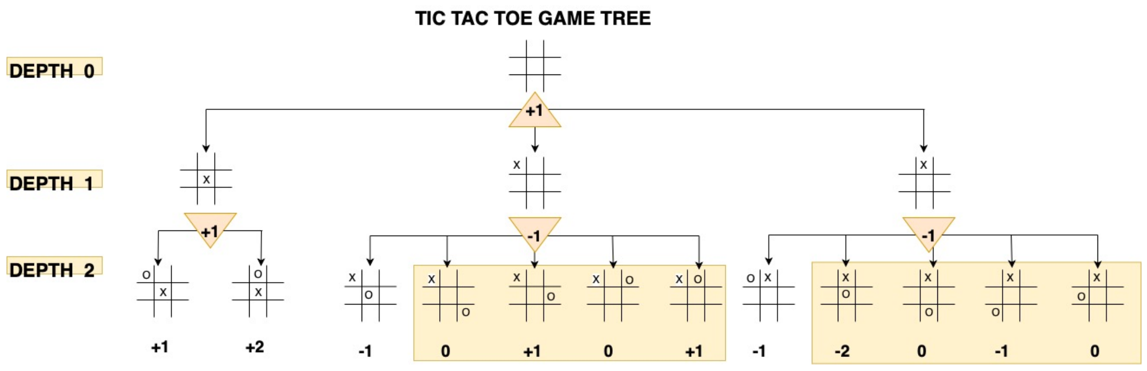

Figure 4.4: Alpha-Beta Pruning