What makes MCTS different from past decision-making algorithms like Minimax?

For tic tac toe game, if the game tree is too large and complex minimax become useless if not using depth-limited search. Another issue is that if minimax it’s too complex that cannot write good evaluation function to tell AI what good or bad position is.

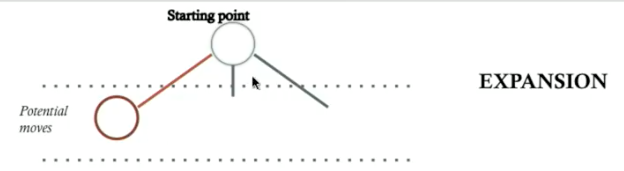
Monte Carlo Tree Search

How could it possibly “plan” ahead when there are so many potential moves and counter moves in Go?

MCTS builds a statistic tree (detailing value of nodes) that partially maps onto the entire game tree. The statistic tree guides the AI to look only at the most interesting nodes in the game tree. The way to update its nodes it’s by running simulations which starts up at random.

How AI knows which node is going to make?

MCTS begins making it starts tree and then it will call one of these functions including (selection, expansion, simulation, update deepening on the situations)



Above picture shows here are child notes haven’t explored yet, so it will add a new node to the stats tree, representing a position in the game tree that the AI will “investigate” (how good the move is) next.

Starting from the position represented by the left child node, make random moves repeatedly until the game is won, or lost

And then depending on win or lose, update left child node in starts tree with relevant starts. Like 1/1 means 1 win out of 1 simulation. And then it will do the same thing with the next child nodes.



After three Childs explored. The number say that the left node is better move, but chances are, these numbers (produced by a single random simulation) are probably not a good indicator of how good any of the moves are!

Tips: More simulations will make them more accurate.

After all child nodes has been visited at least once. Next step is to select which child node to be investigated further.

Tips: the higher the value, the “better” the move is (for producing a win)

Selection based on:

1. How good are the stats?
2. How much has child node been “ignored”

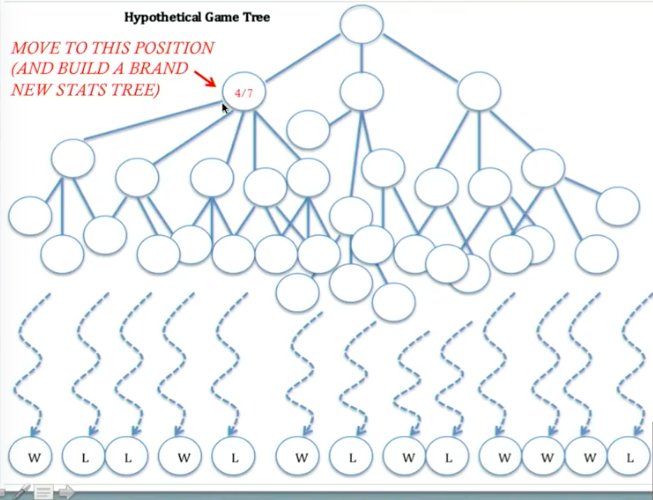


This function allows AI to determine balance between exploration and exportation.

Based on how good the value is and then it will call expansion again. Start the simulation from child node of the child node, and then update the note continues up the starts tree!

If we tell the AI to step with the “planning” at this moment and just make a decision, what move will it make?

The AI will choose the highest simulations, so the node with the most number of highest simulations will be selected.



MCTS allows AI’s to be more adaptable in terms of solving a variety of different problems.

MCTS works by building and updating statistic tree using simulation which determines favourable moves as well as guides its own self-improvement.

For Love letter card game, Perfect Information Monte Carlo(PIMC) and determinized Minimax and found that none of the algorithm could show the stable results due to game randomness.

Single Observer-Information Set Monte Carlo Tree Search(SO-ISMCTS) as the main algorithm for comparison for its ability to search a single tree for games of imperfect information. It has shown good results for more complex card games.

Also, SO-ISMCTS searches more deeply than PIMC with the same computational budget.

Knowledge based agent

1. Knowledge based agent enforces game moves in accordance to the actual game rules.

* For example, if you have a guard, the knowledge based agent handles the selection of the guess card as: the card which is encouraged the minimum number of times in the played deck is chosen. Therefor the algorithm can deduce the number of non-played cards for each character in the game.
* The rule-based approach prefers to make a move with the smallest valued card in the hand in order to save the highest one (each character card has a unique value).

1. Secondly the algorithm checks for the game move that can lead to an immediate win in single turn.

Perfect Information Monte Carlo (PIMC) algorithm

PIMC finding moves that can lead to an immediate win (mentioned from knowledge-based agent rule 2). Then determinization is done and algorithm removes played deck from the full deck.

At the expansion state a child node which is not a part of the game tree yet is added to the tree. After that, at simulation step, game is played till the end and no new node is appended to the game tree.

**Single Observer-Information Set Monte Carlo Tree Search (SO-ISMCTS) algorithm**

Comparing PIMC, SO-ISMCTS algorithm preforms more iterations and greater playing strength is expected from it.

SO-ISMCTS compare to Determinized minimax, minimax builds a whole game tree and then traverse it, so its performance is slower compared to the previous algorithms. The result provides interesting evidence that perfect information game does not show good results for the love letter game.

SO-ISMCTS compare to knowledge-based agent, SO-ISMCTS shows better performance than knowledge-based agent.

After using four algorithms among SO-ISMCTS with knowledge-based, PIMC, and determinized minimax play, we have found that the playing strength is not statistically affected by the number of iterations played in Love Letter game.