PROPOSAL – RECONNAISSANCE BLIND CHESS AGENT

26^{th} August 2021

1 Overview of Reconnaissance Blind Chess (Recon Chess):

Reconnaissance Blind Chess (RBC) is a chess variant designed for research in artificial intelligence (AI). RBC includes imperfect information, long-term strategy, explicit observations, and almost no common knowledge. These features appear in real-world scenarios and are challenging to even the current state-of-the-art algorithms. Each player of RBC controls traditional chess pieces but cannot directly see the locations of opponent's pieces. Rather, the player learns partial information each turn by privately sensing a chosen 3x3 area of the board. The approximate total number of game states that can be considered in a game of RBC is approximately 10^{93} times that of a game of chess, because each state requires keeping track of not only what the game board actually looks like, but the information that each player has acquired.

2 Motivation:

Leveraging knowledge gained through learning from a fully observable deterministic environment in the Partially Observable variant of the same environment is something we humans do, for instance, it can be exemplified by how humans learn driving in an ideal setting where there isn't any uncertainty and can be constructed as an MDP and apply that knowledge to drive in traffic which resembles a POMDP. An algorithm inspired by the above idea, which can utilize a learned model from Fully Observable deterministic setting of a use-case for planning in Partially Observable setting of the same would be very close to how humans learn and plan in the real world and would have great research value and applicability.

Recon Chess acts as an ideal test bed for hypothesizing and testing the applicability of such an algorithm as a learned model could be built for planning and playing in traditional chess and utilize it for planning in Recon Chess which is a partially observable variant of chess. Since Recon Chess has states whose count is approximately 10^{93} times that of traditional chess, it also provides a way to test the scalability and applicability of the aforementioned idea for large scale environments. Working on building one such algorithm not only addresses the final outcome of planning in POMDPs but also the steps involved, and the questions tackled during the process, if carefully investigated with mathematical reasoning would act as valuable contributions to the research community in itself.

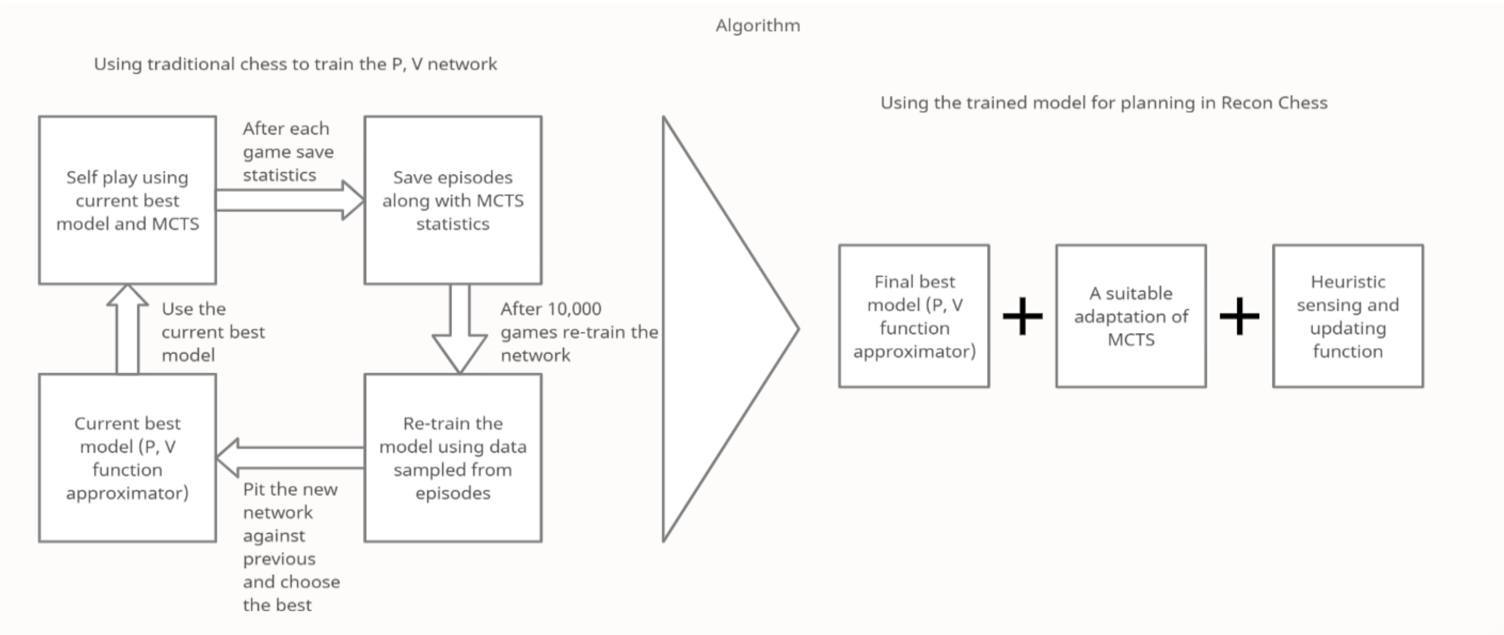
3 Goal:

- To build a fully functional agent which can compete and surpass other existing agents capable of playing Recon chess. Advance and develop from state-of-the-art algorithms in the areas of perfect information games (AlphaZero, MuZero) and imperfect information games (ReBeL) to come up with approaches which could be generalized to various other imperfect information games and different domains as well.
- Come up with an elegant and generalized way of initializing and maintaining distributions across the necessary possible states which can be applied to other games as well, with slight or no modifications.

4 Contributions of this work:

If the end goals of this project are achieved, this work would provide a framework which can be used for agents acting in imperfect information environments, which is close to how the real-world is. Recon chess being a game with substantial strategic depth, an algorithm which can handle the intricacies and complexities of it would be of great value for further research towards multi-faceted agents which can learn and act in different environments.

5 Schematic representation of the algorithm:



6 Brief overview of progress so far:

Repository link: https://github.com/rvteja24/reconChessAgent

- Discussed and decided on the techniques to explore in order to perform in the above use-case.
- Set up Deep Neural Networks for approximating Policy and Value functions.
- Incorporated these DNNs into the Monte Carlo Tree Search.
- Came up with the techniques to discretize and maintain all the possible valid board states at each transition and the distribution across them obtained from the policy network.
- Came up with a way to efficiently use the sensing action to gather maximum information by focusing on the 3x3 grid with maximum entropy/uncertainty.

$$H(S) = \arg\max_{(i,j)} \sum_{x=i-1}^{x=i+1} \sum_{y=j-1}^{y=j+1} C(x,y)$$

$$C(i,j) = \begin{cases} 0 & if\ P(i,j) = 1\ or\ P(i,j) = 0\ or\ i > 7\ or\ i < 0\ or\ j > 7\ or\ j < 0, \\ 1 & if\ 0 < P(i,j) < 1\ and\ 0 \leq i \leq 7\ and\ 0 \leq j \leq 7. \end{cases}$$

H(S) is the heuristic function whose input S is a 8x8 matrix with probabilities of a piece being there in a specific cell represented by P(i, j). C(i, j) translates this into a 0-1 representation of uncertainty for a specific cell. H(S) scans the entire 8x8 matrix and chooses the i,j whose encompassed 3x3 grid has highest level of uncertainty.

- Implemented method to prune invalid board states and update the probability distribution based on the sensory observations.
- Setup for self-play using MCTS and DNN for traditional chess and methods for acquiring MCTS statistics related to policy distribution and values of the states.

7 Bottlenecks:

Infrastructure for running self-play to generate episodes and training the DNNs. An average game of chess of about 250 moves each takes 3 minutes to finish with 40 MCTS simulations per move. So, generating episodes from self-play might be a bottleneck.

8 Current plan for further work:

- Train the DNNs to efficiently perform in traditional chess.
- Integrate these DNNs for the use with MCTS like planning algorithm in recon chess.
- Come up with an efficient way to sample across the distribution of board states during MCTS to find the best possible move across the possible states in recon chess. (Since recon chess is a POMDP environment at scale this in itself would be a very valuable part of the work.)
- Fine tune DNNs for recon chess.
- Evaluate against openly available recon chess baseline agents.

9 Avenues to evaluate the agent:

• NeurIPS 2021 Competition: https://rbc.jhuapl.edu/ (Registration deadline: 2021-10-01, Tournament date: 2021-10-20)
• Hidden Information Games Competition (2021): http://higcompetition.info/about/ (Deadline for submission of agent: 2021-11-30)