

A Heuristic Depth-Limited Search Agent based on Markov Bayesian Probability Inference for Recon Blind Multi-Chess

CS 4649: Robot Intelligence: Planning

Zhangqi Liu, Ruoyang Xu

Last Edited: April 21, 2019

Executive Summary

Proposed Method

Details of the Approach

Rationale and Result

Contributions

We divided the project into two parts: sensing and decision making. We agreed to encapsulate our code in a way that it would require minimum interfacing during code integration, which resulted in two individual files: `chesspiece.py` and `chessplay.py`, whereas `my_agent.py` calls the two files. This practice is to ensure code readability in `my_agent.py`.

Since we both think the other partner's part is more difficult to implement, we consider this project as equally contributed.

Liu ZHangqi FILL in HERE

Ruoyang Xu Responsible for the sensing part of the agent. Coded the entirety of `chesspiece.py`, the python instantiation of the sensing model for our agent. Notable contributions include: I wrote an available moves function for all the chess piece at any position used for propagation of probability that any piece is at any location, which is kept as 16 numpy matrices. The `chess.Board` class is not used due to the immense number of boards I would have instantiate to keep for each possibility ($16 \text{ pieces} \times 8 \times 8$ `chess.Board` objects that each keeps a 8×8 chess board vs a 16 by 8 by 8 matrix). I wrote the move result update for extrapolating if any particular chess piece is killed and/or at any particular position. I also wrote the sensing decision process in that aims to acquire maximize information by targeting the location with max entropy.

References

[1] Author "*place holder*". SIAM News, Volume 23, Number 5, September 1990, pp 1 & 18.