Applications of Partially Observable Monte Carlo Tree Search Algorithms to Reconnaissance Blind Chess

Brady Garrison | **Dr. Raj Dasgupta**

U.S. NAVAL RESEARCH LABORATORY

SEAP
SCIENCE AND ENGINEERING APPRENTICE PROGRAM

Project Objective and Intern Contribution:

Our aim was to implement and test the Single-Observer-Information-State Monte Carlo Tree Search (SO-ISMCTS) and Belief-State Monte Carlo Tree Search (BS-MCTS) algorithms for limited information games, particularly Reconnaissance Blind Chess

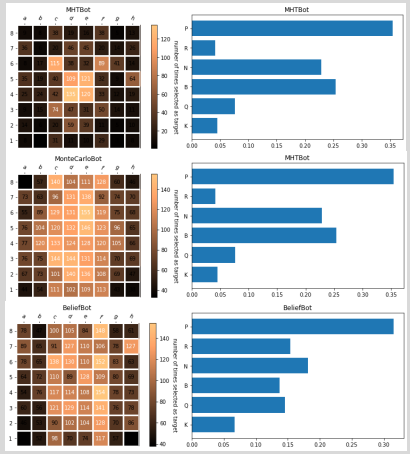
The method we used to accomplish this aim was to program the algorithms in Python and use libraries like the Reconnaissance Blind Chess library, AnyTree, and matplotlib to visualize their performance

I was assigned to implement BS-MCTS and SO-ISMCTS for both full information tic tac toe and for limited information Reconnaissance Blind Chess. I also was assigned to improve performance against baseline bots included in the RBC library and on online servers. In addition, I made metrics to gauge bot performance and tree structure.

My contributions were implementing code and algorithms based on assignments from my mentor. I also spent time researching different algorithms, RBC approaches, and programming tools.

Data from Local RBC Testing

White Bot	Black Bot	White Wins	Black Wins	Draws/Errors
AttackerBot	BeliefBot	11	7	0
AttackerBot	MHTBot	9	2	6
AttackerBot	MonteCarloBot	13	5	0
BeliefBot	AttackerBot	0	16	1
BeliefBot	PasserBot	12	13	0
BeliefBot	RandomBot	5	14	0
BeliefBot	TroutBot	1	21	0
MHTBot	AttackerBot	10	6	3
MHTBot	PasserBot	21	0	5
MHTBot	RandomBot	20	0	1
MHTBot	TroutBot	6	14	2
MonteCarloBot	AttackerBot	3	13	1
MonteCarloBot	PasserBot	18	4	1
MonteCarloBot	RandomBot	14	5	0
MonteCarloBot	TroutBot	0	22	0
PasserBot	BeliefBot	9	14	0
PasserBot	MHTBot	0	5	16
PasserBot	MonteCarloBot	9	12	0
RandomBot	BeliefBot	12	6	1
RandomBot	MHTBot	0	14	5
RandomBot	MonteCarloBot	12	6	1
TroutBot	BeliefBot	20	1	0
TroutBot	MHTBot	9	8	3
TroutBot	MonteCarloBot	20	0	0



Top left: Example board for RBC
Bottom Left: Results of a tournament run between baselines (Attacker, Passer, Trout), a heuristic-based bot I programmed last summer (MHTBot), and the new algorithms, BSMCTS (BeliefBot) and SO-ISMCTS (MonteCarloBot)
Right: Distributions of target squares on a chess board and pieces chosen by MHTBot, MonteCarloBot, and BeliefBot

1. Answer: What are you most proud of this summer [with respect to your experience/project]?
I am very proud that I was able to implement algorithms that made complex decisions via tree structures from pseudocode. I am happy that my bots demonstrated complex behavior.

2. Answer: Why was the internship valuable?
This internship gave me further experience with programming in python, understanding and writing pseudocode, documenting and testing implementations, and making presentations. I learned a lot from discussions with my mentor and presentations from other experienced researches.

3. Answer: Advice for future cohorts?
Even as things become difficult, continue pushing forward and striving for improvement as the best results are achieved with perseverance.

Results / Accomplishments / Next Steps:

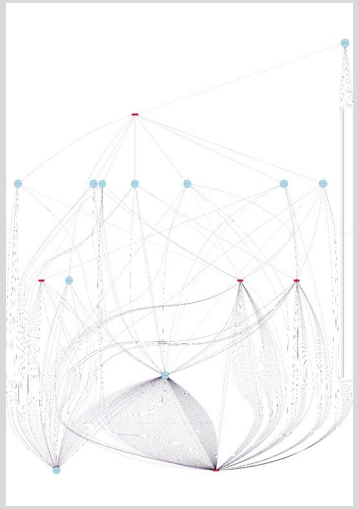
We demonstrated the potential applications of partially observable Monte Carlo Tree Search algorithms to complex situations and the relative performances of Belief State and Information State based approaches

Reconnaissance Blind Chess is characterized by high uncertainty and long-term strategy, making it comparable to real world scenarios. The impact for the Navy is that approaches that solve such a challenge can be used for tactics and other real-world problems.

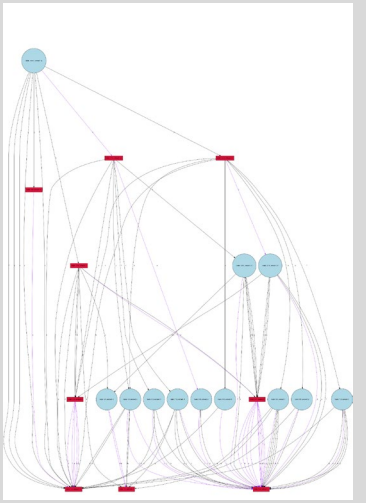
What's most important is understanding the intricacies of complicated algorithms and how they can be made to perform efficiently in challenging scenarios.

In the future this work will be able to compete in RBC tournaments and hopefully perform well against other high-level algorithms. The tree structures implemented can also be used as baselines for Reinforcement Learning implementations.

Tree Graphs for Tic-Tac-Toe



Tree graph for BS-MCTS, circular nodes are player nodes, square are opponent nodes, purple lines are selected actions



Tree graph for SO-ISMCTS , circular nodes are player nodes, square are opponent nodes, purple lines are selected actions