COMS 572
Lab 2
Eranda Sooriyarahchi

English Draughts with Alpha Beta Agent

A general alpha-beta game playing agent which takes the current state as input and returns a move/action to be made by the agent was developed for a game of American checkers. For the implementation of the Alpha Beta Search, AlphaBetaSearch.java file was used from the AIMA code provided by the book on the following url:https://github.com/aimacode/aima-java/blob/AIMA3e/aima-core/src/main/java/aima/core/search/adversarial/AlphaBetaSearch.java
Additionally for Checkers GUI the following repository from GitHub was used:
https://gist.github.com/89515522549/e2effc34a8b3a43504cd

6. Compare the effect of increasing search depth (come up with a method to demonstrate your point).
7. Implement at least two evaluation functions with vary quality. Compare the effect of improving the evaluation function.

Performance measure of the AI agent
When the agent is taking a longer duration of time, the agent has taken a more thorough and thought-out decision, i.e. it is more difficult to beat the AI agent. Therefore, I measured the time taken by the agent to make moves as it positively correlates to difficulty of beating the AI agent, hence as a positive performance measure.

Evaluation Functions

Condition A:
Used only the score ratio for the evaluation function. At depth 5, 10, and 15.
Condition B:
Used the combination of 3 evaluation functions. Score ratio, score proportion and King distance ratio at single depth of 10 and 5.

The evaluation function that I used initially was the score ratio. The Kings were valued 10 and the men were valued at 5.
Score of the black side/ Score of the red side = Score ratio , and this was further divided by total score to normalize it.
Score Proportion = Score Difference between two side/ Total score
and
King Distance ratio = King Average Difference in Distance/ 7.0 , were the final two evaluation functions that were used.

Under Condition A:

Condition A was used to compare the effect of increasing the depth.
At Depth 5, Average time taken by AI agent: 18ms on average.
At Depth 10, Average time taken by AI agent: 157ms on average.
At Depth 14, Average time taken by AI agent: 6187ms on average.
As the depth increased, the time taken by the AI agent to complete a move increase significantly, which corresponds to the difficulty to beat the AI agent. Additionally, these results were similar to the subjective observation, the difficulty playing against the AI agent became significantly harder as the depth increased.

Under Condition B:

With 3 evaluation functions:

At Depth 5: Average time taken by AI agent: 29ms
At Depth 10: Average time taken by AI agent: 686ms.

As you can observe, if you compare the average time taken for Depth 5 with only one evaluation function was 18ms compared to 29ms with 3 evaluation functions.

At depth 10, you can see the increase in time as the number of evaluation functions increased, more noticeably. At Depth 10 with only one evaluation function was 157ms compared to 686ms with 3 evaluation functions.
That increase in time is significant. And as the longer time taken by the AI agent implies better performance of the agent, the difficulty of beating AI agent was greater with more evaluation functions.

In conclusion, when the depth was increased, the performance of the AI agent was improved. When the number of evaluation functions were increased, the performance of the AI agent again improved.