Bilge Kaan Güneyli
2020400051

CMPE 480 HOMEWORK 1 REPORT

Heuristic Function:
pseudocode
ignore all obstacles
if knight can retrieve a pawn with one move**:**
    h(n) = (no. of pawns) * (cost of knight action)
elif rook can retrieve a pawn with one move**:**
    h(n) = (no. of pawns) * (cost of rook action)
elif bishop can retrieve a pawn with one move**:**
    h(n) = (no. of pawns) * (cost of bishop action)
elif knight can retrieve a pawn with two moves**:**
    h(n) = (no. of pawns + 1) * (cost of knight action)
else:
    h(n) = (no. of pawns + 1) * (cost of rook action)

- Ignoring the obstacles helps us in ensuring the admissibility and reducing the computation time.
- The checks are ordered in increasing cost to ensure admissibility.
- The checks end with two moves of rook because the rook can reach any spot this way. Therefore, adding more checks would be redundant.

Explanation of Admissibility:
The heuristic is admissible (never overestimates the result) due to the following reasons:

1. It ignores the obstacles which may cause additional path cost but can never create shorter options.
2. When a capturing is possible, it assumes the capturing is done with the cheapest way possible.
    1. After a capturing action, the heuristic assumes that the piece which has moved is in a position such that it can retrieve more pawns. Considering the other pieces are still in the same position (in which they can not retrieve any pawns or is not the cheapest option) the heuristic assumes other pawns are retrieved with the cheapest way possible as well.
3. When there is no capturing possible, the heuristic assumes we can reach a position open to capturing with the cheapest action (moving the knight). And then it applies the 2nd step
4. When the knight cannot capture any pawn within 2 moves, the heuristic applies the knowledge that rook can retrieve any pawn with 2 moves and then applies step 2.