COMP30024 Artificial Intelligence Project 1 Report

team member: XuLin Yang(904904), Liguo Chen()

search problem formulation

- ***State***: player pieces and obstacle pieces' location on board
- ***Action***: player can move, jump or exit one player piece per turn defined in specification
- ***Goal Test***: no player's piece on board
- ***Path Cost***: 1 cost per action

search algorithms

terminology

- ***b***: branching factor for search tree
- ***d***: length for the solution path in search tree
- ***δ***: relative error in heuristic = |h*(s) - h(s)|
- ***$dist_{SLD}$***: straight line distance in hexe

a* search

- time complexity
  - best case ∈ O(d) if we disregard the complexity of the heuristic calculation
  - average case ∈ $O(b^{\delta d})$ (from lecture)
  - worst case ∈ $O(b^{d})$ (because it is uniform cost search now)
- space complexity ∈ $O(b^{\delta d})$ (because "keep all nodes in memory")
- completeness
  - Yes, as we are guaranteed in the specification "at least one winning sequence of actions exists"
- optimality
  - Yes, as long as h(s) ≤ h*(s) ∀ s ∈ state space

heuristic function

- h(state) = $\sum_{piece \in player}(\lceil \text{dist}_{\text{SLD}}(\text{piece}) / 2 \rceil + 1)$
- admissibility:
  - Discussing Red player is similar for Green and Blue player as they are parallel cases.So we can only discuss Red player case at here.
    Fastest path for a single piece on board to reach in goal hexe is that the piece can jump to goal hexe as much as possible and then exit. i.e. h(piece to goal hexe) = # jump action = $\lceil \frac{number\ of\ move\ action}{2} \rceil$.
    Where #move action can be measured by SLD distance
    ∵# jump action is the ideal(lower bound of) length of path for the piece to reach the goal hexe as described above
    ∴h*(piece) ≥ # jump action + 1. Note: plus 1 for exit action
    ∴ h*(state) ≥ $\sum_{piece \in player}$h(piece) =$\sum_{piece \in player}(\lceil \text{dist}_{\text{SLD}}(\text{piece}) / 2 \rceil + 1)$

problem feature impact

search tree

- branching factor
- depth

other features of the input impact on search algorithm

- time complexity
- space complexity