Report of COMP3211 Multi-Agent Path Finding (Individual)

1 Project description

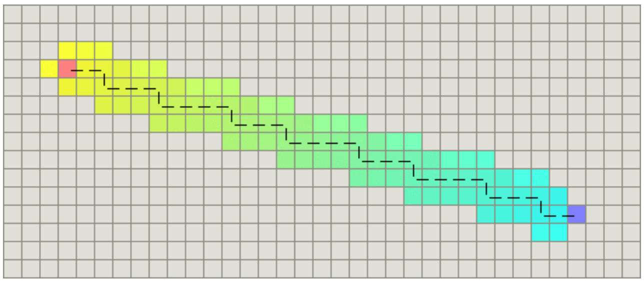
The task of this project is to design an environment specific controller for two-agent cases. To achieve this, a method of Conflict-based Search (CBS) combined with A* Search is used.

2 Methodology

The details of the CBS A* Search will be discussed here. The idea is to create a two-levels algorithm – high level and low level. The low level looks at each individual agent and finds the optimal route from given state to goal, while the high level makes sure that there will be no conflicts by looking into the constraints of each agent at particular state.

3 Low-Level A* Search

Astar(self, goal, game_state)



The low level of the algorithm utilizes A* Search to search for the shortest path for a given state to a goal. The procedures inside are tantamount to typical A*star algorithms, but the heuristic function shall be a vital point to note. In this implemented A* Search, Manhattan distance is used as the heuristic, where the cost for each move is set to be 1. This decision is made based on the 4-direction nature of the robots.

(heuristic_manhattan(goal, pos) -> float)

4 High-Level

Astar(self, goal, game_state, check = constraint_list)

The High-Level algorithm is the key for being a Conflict-Based Search. The only difference between the high-level and the low-level is the addition of a list which gives information about the constraints of an agent at a particular game state. The constraint list is generated by considering the state-specific possible actions of another agent. The high-level will utilize this information to generate a conflict-free solution. In particular, it aims to perform a best-first search on the available actions according to the costs.