**FIT3080 Assignment 1 Report**

**Andy Zhan (ID: 29659213)**

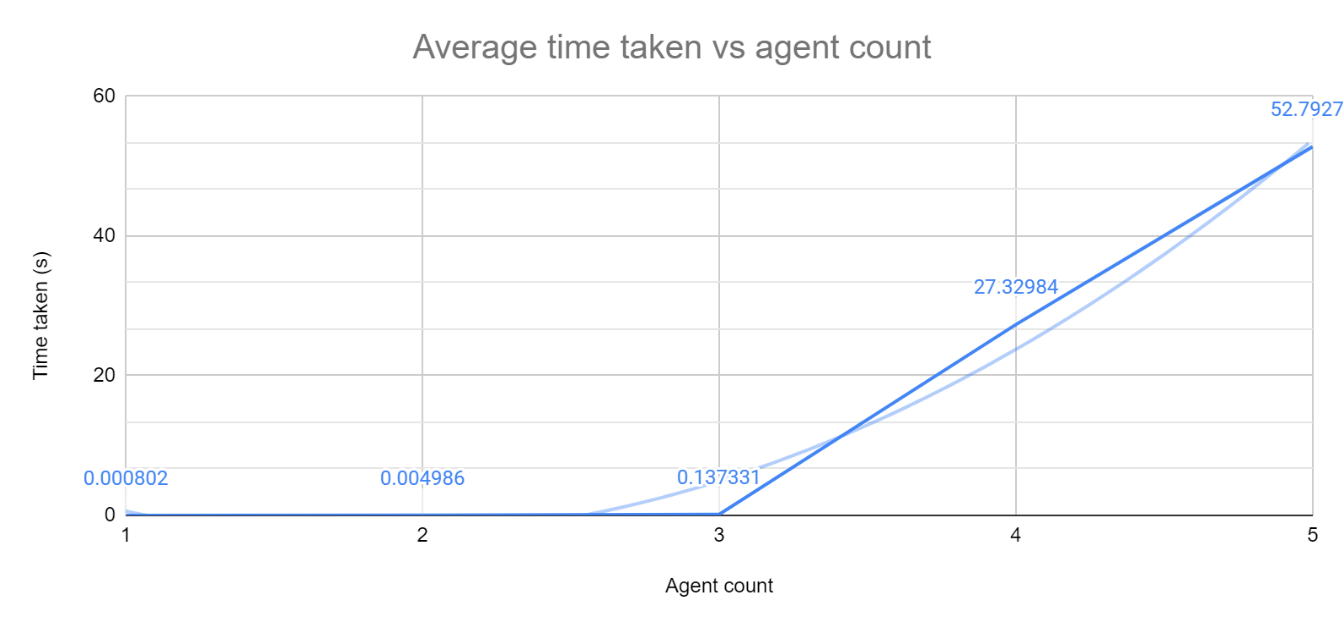
**Implementation**

For the Flatland scheduling problem, a multi-agent graph-based A\* search was used, with each node representing the current state of all the trains. For the heuristic, I used the maximum Manhattan distance from the agent’s position to the goal out of all the agents. I found that this was suitable, because it was a consistent heuristic, and it made sure that the agents moved together in order to minimise h. (As opposed to the sum of each agent’s Manhattan distance, which disregards that fact that the agents should move in synchronisation towards the goal and not one at a time.)

As a tie-breaker for nodes with the same f-value, I prioritised the ones with a smaller h-value, so nodes closer to the goal node were expanded first.

**Evaluation**

Below, you can find the average time taken (over 10 different map seeds) by A\* search, in seconds, plotted against the number of agents. For each agent count, the map size was agent count + 3.



A\* search is O(bn) where b is the number of possible actions for an agent on a particular turn (i.e. the branching factor), and n is the number of agents. This exponential complexity means that using algorithms based on A\* is unfeasible for large numbers of agents due to time and memory limitations. The exponential trend of the chart supports this conclusion. Running the algorithm on more than 5 agents took too long for data to be recorded.

**Analysis**

*This section refers to a paper on Conflict Based Search, cited below:*

Sharon, G., Stern, R., Felner, A. and Sturtevant, N., 2015. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence*, 219, pp.40-66.

Sharon, Stern, Felner and Sturtevant propose a solution called Conflict Based Search (CBS). While A\* finds a solution by modelling the problem as a single ‘joint agent’ system, CBS does not do so, and instead employs a dual-layered approach: first, on the higher level, it searches within a ‘Conflict Tree’ that is based on the conflicts between individual agents; then, searches are performed on a per-agent basis to meet the constraints imposed by the node from the higher level Conflict Tree. If there are still conflicts between agents after running the low-level search, nodes with constraints that solve the new conflict are expanded. In terms of the Flatland train scheduling problem, the main conflict would be multiple trains occupying the same location at the same time.

Both CBS and A\*(with an admissible heuristic) guarantee an optimal solution. The advantage that CBS has is that in a normal (i.e. not worst-case) scenario, it is able to quickly rule out the f-values that lead to conflicts, and continue looking for solutions that bypass them, while A\* must expand the Cartesian product of the single agent paths with that f-value. Additionally, the time expanding low-level CBS nodes is much less than the time taken expanding A\* nodes, because A\* is expanding multi-agent nodes while CBS expands single-agent ones. The open list of CBS is also smaller as its size is linearly proportional to that of the input graph, whereas A\* search’s handles the exponentially larger multi-agent state space.

Therefore, I believe that CBS would be more performant for the Flatland train scheduling problem, due to the exponential nature of A\* search (as seen in the previous section), as well as empirical results from Sharon, Stern, Felner and Sturtevant which demonstrate the efficiency of CBS compared to pure A\* search on a standard 8x8 grid multi-agent pathfinding problem.