WRITING ASSIGNMENT 2

Yahya Alhinai
alhin010@umn.edu
CSCI4511W

October 15, 2019

Paper Summary

The paper "Reusing Previously Found A* Paths for Fast Goal-Directed Navigation in Dynamic Terrain" has introduced a new search algorithm that called Multipath Generalized Adaptive A* (MPGAA*) and compared its functionally and efficiency with other competing algorithms that were well established at the time of this paper was published. Those search algorithms are aimed to be useful in an environment that changes its terrain whenever the agent moves or the goal state changes.

The presented algorithms MPGAA* is an improved version of GAA* which fundamentally updates heuristic values of states that were expanded by A* to facilitate subsequent searches which leads to speed up the process. A major improvement of MPGAA* over GAA* is the time-run speed up. Due to the fact that MPGAA* algorithms utilizing paths that were already computed and produced by A* before terrain changes and then stop once it finds such a path. To explain MPGAA* algorithm, it starts conducting A* search from the stare state to the goad state. Once it finds a path then the features of GAA* start updates the heuristic to make heuristic values more informed to speed up other A* reaches that will be conducted afterwards.

Once the terrain of the environment changes, the algorithm will perform a A* search to an already-existed path that was computed before terrain changes and connect the agent to it. MPGAA* stops once it finds a path which what make the already efficient. It's worth mentioning that D* Lite algorithm has been compared to the presented algorithms MPGAA* in several part of the paper because it is the direct competition to MPGAA* algorithm at goal-directed navigation.

What is a specific examples of a "dynamic terrain" where this algorithm would be useful?

Dynamic terrain is a description to an environment that might change as the agent is moving. The example of dynamic terrain (5X5 grid) that was given in the paper is a good basic description of the kind of environment that the algorithm MPGAA* would be useful operating in. In the example, the terrain is shown as black cells which indicates that visibility range is assumed as infinite. Which can be represented as block and the reset of cells remaining are unblocked. In the beginning, MPGAA* algorithm apply A* search to find an optimal path from current state to the goal. It will stay the same until terrain changes in this case the positions of the block cells. Once the terrain changes, another A* search will be exploring the terrain until it finds an already-exiting path from previous searches. Once it finds it the algorithm MPGAA* stops early. This is the reasons MPGAA* is generally faster that A*, GAA*, and D* Lite. This algorithm works in any environment that changes its terrain as the agent moves. For instance, an agent searching for an item in a room.

All the environments mentioned above has one thing in common: They all have bounded area and they are fully observable to the agent. Also, the number of searches that is done whenever there is a change in the environment is limited as well. Additionally, only small proportion of the environment changes at a time which allows the algorithm to adjust heuristic values incrementally as the terrain changes. If the environment is changing too rapidly, the algorithm MPGAA* will have the same efficiency as A* or even worse.

What is an example of a "dynamic terrain" where the proposed algorithm might not be useful?

An example of the algorithm MPGAA* might not be a good option is when the environment is vastly large, and the terrain are rapidly changing across the environment. This is due to one of algorithm's property that seeks to change the heuristic values for the better. Alongside changing heuristic values the algorithm has to establish consistency with every heuristic value that changed. The performance of the algorithm MPGAA* might suffers due to amount of heuristic value that seeks to change when the range of the environment is vastly big.

Does the proposed algorithm always outperform the "state-of-the-art"?

No, not always the presented algorithm MPGAA* outperforms "state-of-the-art" in this case (D* Lite) algorithm. This can be seen in table 4 in the paper. where at 30% rate change of the map in warcraft3 made D* Lite outperforms MPGAA* by more than 4%. This is due to the great deal of heuristic values that is being modify whenever there is a change in the environment which delays taking the next step. The reason D* Lite is faster in this situation is because whenever the is a change in the environment, it only updates the cost of edges and update the shortest path to the goal. D* Lite keeps doing this, without changing and recalculating heuristic values, until it reach the goal state.

What is the problem setting that they are solving for?

The authors experimented all four algorithms in path-planning in dynamic eight neighbour grids. There are 3 settings that they experimented with. In the setting 1, they used a random maps, which usually considered when evaluating goal-oriented navigation algorithms. In the setting 2, A room map from the Sturtevant repository can be considered an excellent simulation scenario for indoor navigation. In the setting 3, they used a game map can be regarded as an outdoor navigation goods simulation scenario.

Is the justification sufficient?

The justification is sufficient as they changed rates with 1, 5, 10, 15, 20, and 30 percent considering changes can occur in any part of the search space to see the percentage of runs in which MPGAA* is faster than D* Lite.