

CSCI 4511W Writing Assignment 2

Academic Paper Analysis

Brian Cooper
coope824@umn.edu
University of Minnesota

October 14, 2019

1 Paper Analysis

Reusing Previously Found A Paths for Fast Goal-Directed Navigation in Dynamic Terrain*¹ is a paper by Hernández, et al. that considers a modified version of the classic A* pathfinding algorithm titled *Multipath Generalized Adaptive A**, or MPGAA*. The MPGAA* algorithm itself is an extension of *Generalized Adaptive A** (GAA*), which itself is an extension of a variant called *Repeated A**. Repeated A* extends A* by simply following a discovered path through running A* until the global cost function has changed or the goal has been reached, while GAA* takes this a step further by introducing dynamic heuristic updating after each A* state expansion. Specifically, the h -values (as in the original A* algorithm) are updated after each search, and also if a cost of an arc (edge) has decreased, the h -values of the states in question are updated accordingly. The authors clarify that the second h -value update protocol is required to retain heuristic consistency, while the first h -value update protocol automatically retains consistency.

Throughout the paper, there is extensive mention of *dynamic terrain*, which entails a fluid environment (i.e., a non-static terrain that may change at any time). For the author's experiments, they used "Random, room and Warcraft 3 maps" (Hernández et al.). The terrains are outlined in greater detail below:

- **Random maps:** 1000×1000 grids with 10% obstacle ratio
- **Room maps:** 512×512 grids, 40 rooms per grid
- **Warcraft 3 (36 maps):** 512×512 grids with 5% obstacle ratio

¹Hernández, C., Asín, R., Baier, J. A. (2015). Reusing Previously Found A* Paths for Fast Goal-Directed Navigation in Dynamic Terrain. *Association for the Advancement of Artificial Intelligence*. Retrieved from <http://www.cs.toronto.edu/~jabaier/publications/HernandezAB15.pdf>

Each setting (random, room, and Warcraft 3) involved 100 randomly-generated instances. In the case of the Warcraft 3 maps, 100 instances were generated for each map.

The paper touched on the idea that game maps often involve dynamic terrain (and therefore are useful for testing MPGAA* on when it comes to outdoor navigation scenarios). Similarly, room maps are good representations for indoor navigation scenarios. Other "dynamic terrains" that MPGAA* might perform well on are map navigation environments (such as Google Maps or OpenStreetMap paths and similarly autonomous vehicle sessions), air traffic environments (due to constantly-changing flight traffic), and robotic warehouse navigation environments (such as those used in Amazon fulfillment centers). On the contrary, MPGAA* may not perform as well as other algorithms in environments with low agent movement (relatively low movement steps) and a high change rate quotient, such as navigating an agent through a crowd of people.

Although MPGAA* often outperforms the state-of-the-art D*Lite algorithm, it does not in every situation. The situations that D*Lite is likely to perform better on are situations where k (the number of moves) is small and where cr (the change rate, or phase state possibility of switching blocked cells to unblocked and vice versa) are relatively high. It is important to note that the change rate was used to determine a $\frac{cr}{2}$ chance of changing (un)blocked cells to their counterpart, and the authors used each of {1%, 5%, 10%, 15%, 20%, 30%} change rates for the pathfinding experiments.

2 Paper Evaluation

Overall, the paper introduces MPGAA* in a concise manner. The authors concretely described the segments of the code that, by inclusion or exclusion, would convert the algorithm into a different flavor of A*, even tapping into a modification that marks the relationship between A* and Dijkstra's algorithm. They also mathematically verified the properties of their algorithm (although I would expect this from any academic paper). I think it is good to extend and modify algorithms (both popular and unpopular), as this can drive innovation and has the possibility of filling a gap in industrial and academic applications where algorithms are used, perhaps better than an existing solution in use.

As a final (pedantic) note, I thought it was strange that the authors introduced MPAA* in the "Memory Analysis" subsection before defining what it is in the "Summary and Perspectives" section. This doesn't detract from the paper's delivery in my opinion, I just originally thought it was a typo (wherein the authors meant to write "MPGAA*") before reading further.