Any-Angle Path Planing on Grids. Comparison of Approaches.

Ivan Oleinik, Vlad Milshin

June 1, 2021

Abstract

We study different modifications of regular A* algorithm for any-angle path planing on grids. We present a comparative analysis of several algorithms. Regular A* 2^k path planner finds an optimal solution to any path planning problem over 2^k-neighborhood, but suboptimal in terms of any-angle neighbourhood.[1] Theta* is a variation of A* that propagates information along grid edges without constraining paths to grid edges. It is simple to understand and implement, fast and finds short paths. However, it is not guaranteed to find true shortest paths.[2] Finally, one of the most competitive any-angle pathfinding algorithm ANYA finds optimal paths by searching over sets of states represented as intervals.[3]

1 Introduction

Given a two-dimensional continuous terrain is discretized into a grid with blocked and unblocked cells. Our objective is to find a short unblocked path from a given start vertex to a given goal vertex (both at the corners of cells). A* finds grid paths (that is, paths constrained to grid edges) quickly, but grid paths are often not true shortest paths (that is, shortest paths in the terrain). In order to find better solutions we are studying and analyzing the other approaches. In the following sections we consider each of the studied algorithms.

2 2^k-neighborhood-based Approach

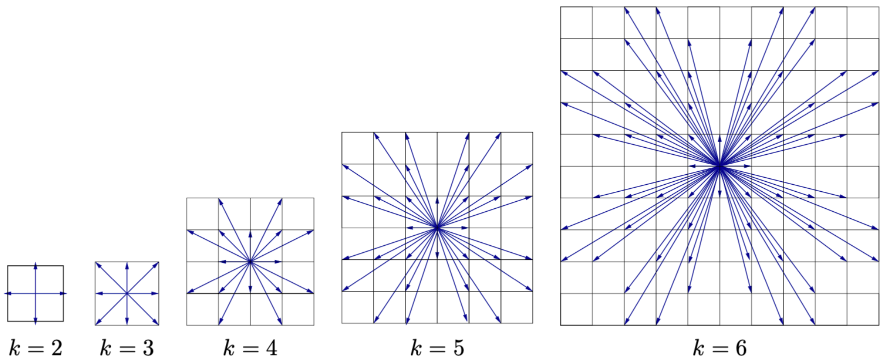


Figure 1: The 4-, 8-, 16-, 32-, and 64- neighborhoods.

We define the 2^k-neighborhood in terms of the 2^{k-1}-neighborhood: 2^k-neighborhood is a union of 2^{k-1}-neighborhood and 2^{k-1} pairwise sums of adjacent neighbors from 2^{k-1}-neighborhood. 4-neigboorhood can be seen in the picture above. The consept of 2^k-neighborhood allows us to propose admissible heuristics that are perfect for obstacle-free grids, which generalize the well-known Manhattan and Octile distances.[1]

3 Theta*

In this section, we introduce Theta* (Nash et al., 2007), version of A* for any-angle path planning that propagates information along grid edges without constraining the paths to grid edges. It combines the ideas behind A* on visibility graphs and A* on grids. Its paths are only slightly longer than true shortest paths (as found by A* on visibility graphs), yet is only slightly slower than A* on grids. The key difference between Theta* and A* on grids is that the parent of a vertex can be any vertex when using Theta*, while the parent of a vertex has to be a neighbor of the vertex when using A*. We first introduce Basic Theta*, a simple version of Theta*. Theta* is identical to A* except that, when it updates the g-value and parent of an unexpanded visible neighbor s' of vertex s, it considers two paths instead of only the one path considered by A*. Theta* also considers the path from the start vertex to the parent of vertex s and from the parent of vertex s to vertex s' in a straight line. This path is not considered by A* and allows Theta* to construct any-angle paths.

4 ANYA

Where other works find approximate any-angle paths by searching over individual points from the grid, ANYA finds optimal paths by searching over sets of states represented as intervals. Each interval is identified on-the-fly. From each interval ANYA selects a single representative point that it uses to compute an admissible cost estimate for the entire set. ANYA always returns an optimal path if one exists. Moreover it does so without any offline pre-processing or the introduction of additional memory overheads. In a range of empirical comparisons we show that ANYA is competitive with several recent (suboptimal) online and pre-processing based techniques and is up to an order of magnitude faster than the most common benchmark algorithm, a grid-based implementation of A*. In rough overview:

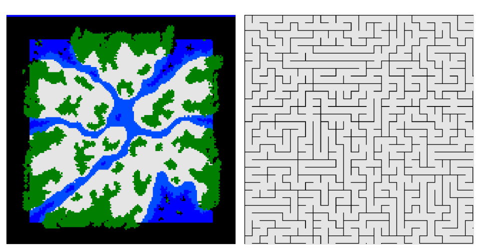
- Where other methods search over the individual nodes of the grid, ANYA searches over contiguous sets of states that form intervals.
- Each ANYA interval has a single representative point that is used to derive an admissible cost estimate (i.e f-value) for all points in the set.
- To progress the search process ANYA projects each interval, from one row of the grid onto another, until the target is reached.

5 Experiments

In this section, we compare ANYA to regular A* 2^k path planner and Theta* with respect to their path length, number of vertex expansions and runtime (measured in seconds).

5.1 Benchmarks

In our comparison, we use two sets of maps from the MovingAI repository (Sturtevant, 2012). The first set, game maps, are from Warcraft III — which contains 36 maps of size 512 × 512, 100 paths per map. The second set, are 60 maze maps of size 512 × 512, 100 paths per map.



(a) Warcraft III (b) Mazes

Figure 2: Examples of the maps.

5.2 Empirical Findings

The objective of our evaluation was twofold. First, we wanted to investigate the impact on solution quality of using $\mathcal{N}_{2^k}$ with A*, the most standard heuristic search algorithm, and to compare the obtained solution with those generated by the any-angle path planners ANYA and Theta*. Second, we wanted to investigate the impact that increasing k had on the runtime performance of A* using h_{2^k} heuristic and the Euclidean distance. In addition, we compare the runtime performance with ANYA and Theta*. Figure 3 shows the average

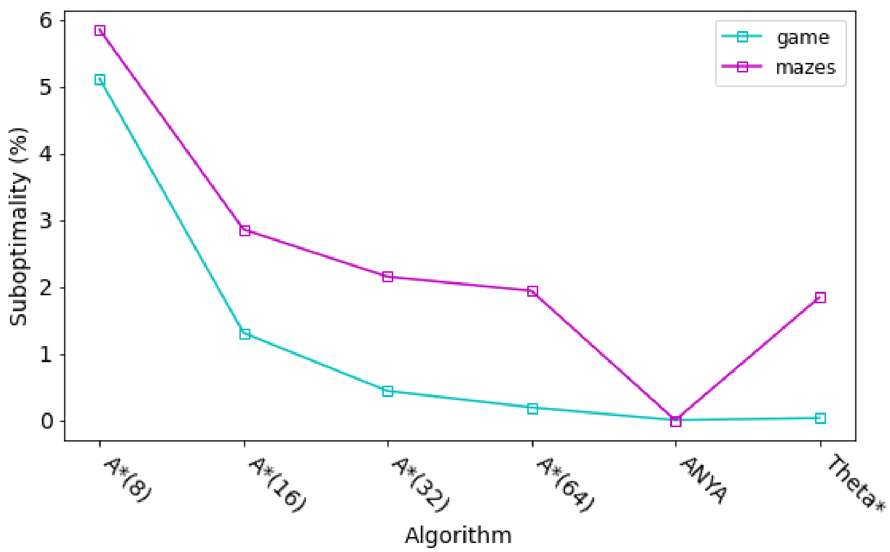


Figure 3: Average suboptimality per algorithm.

percentage suboptimality of Regular A* for different neighborhoods, ANYA, and Theta*. We evaluated four values for 2^k: 8, 16, 32, and 64. We make the following observations.

- Compared with ANYA, an optimal any-angle planner, we observe that A* obtains almost optimal any-angle paths on the Games maps when k over 6 (64-neighborhood). Specifically, over the 64-neighborhood, the suboptimality is only 0.19%. On the Mazes maps the suboptimality is 1.94%.

- Compared to Theta*, we observe that Theta* finds better solutions in Game Maps and Maze Maps. However, the difference in suboptimality between A* for 64-neighborhood and Theta* is only 0.1%.

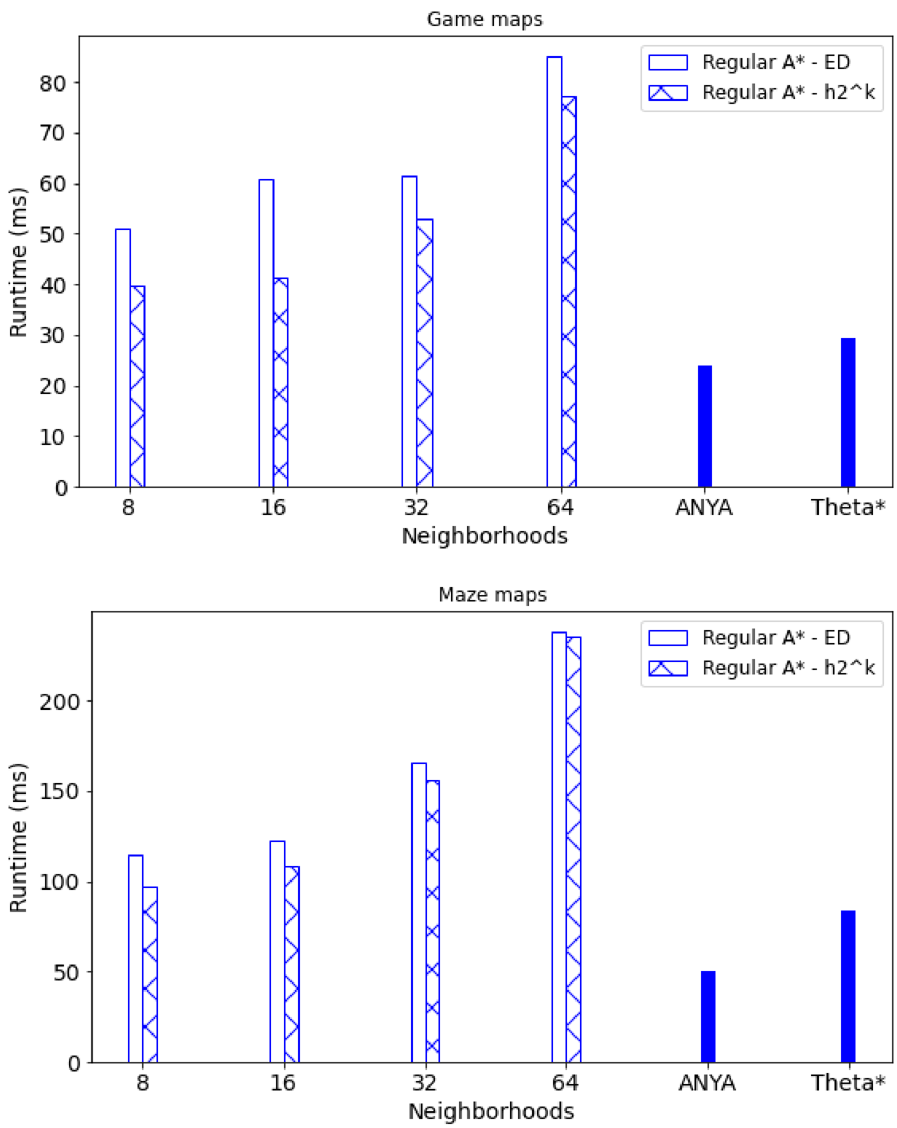


Figure 4: Average runtime per neighborhoods.

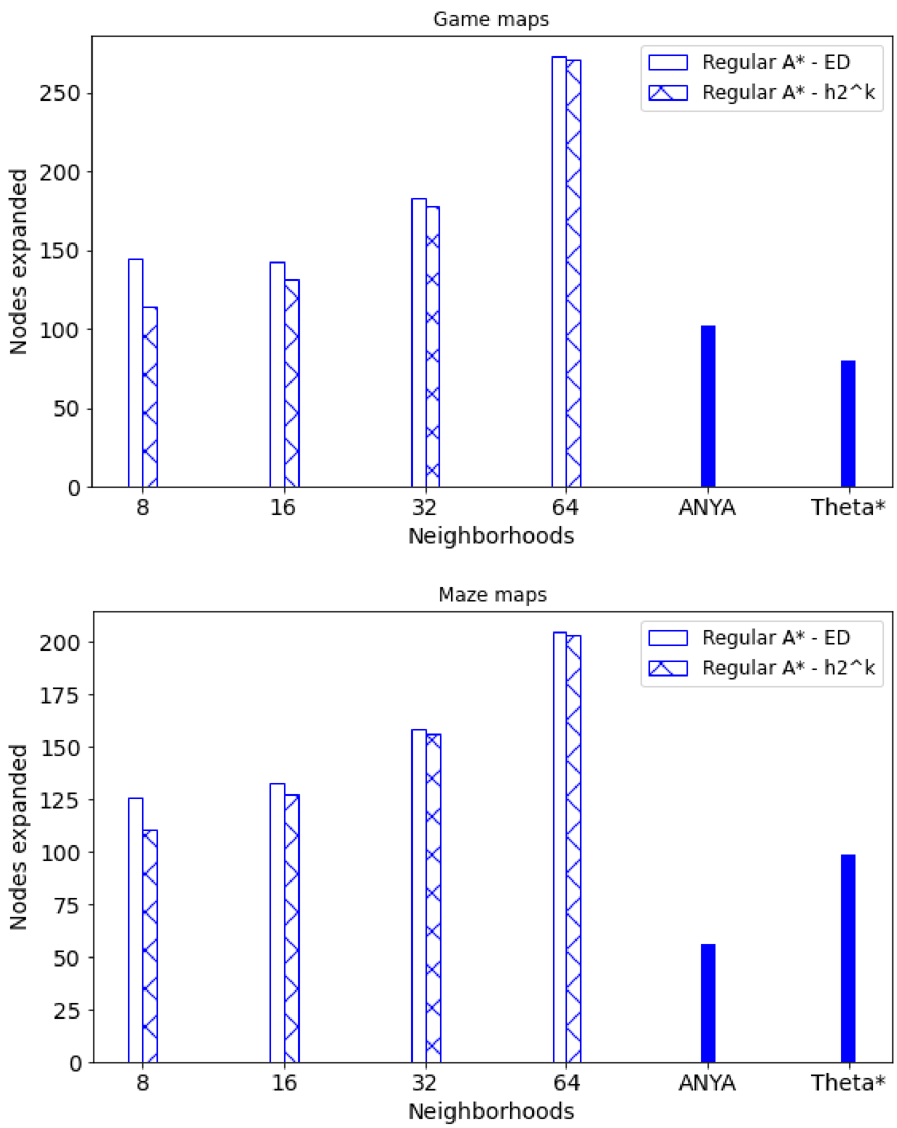


Figure 5: Average number of nodes expanded per neighborhoods.

In order to understand the effects over runtime and the number of nodes expanded of the different algorithms, we present Figures 4-5, which show the average runtime and the average number of nodes expanded of Regular A* for different neighborhoods, ANYA and Theta*. We evaluated h_{2^k} heuristic for the Regular A* and the Euclidean distance (ED) for each algorithm. We evaluated the four values for k. We make the following observations.

- **On Regular A*.** Runtime increases with k on the Game and Maze maps. This can be explained by the larger branching factor, which increases exponentially with k. A* runs faster using the h_{2^k} heuristic than using the Euclidean distance for small values for k. For intermediate and higher values for k, A*, used with ED, is slightly slower due to the overhead in computing h_{2^k} heuristic and because h_{2^k}, when k is large, tends to be more similar to the Euclidean distance. In all maps, A* with h_{2^k} is faster than A* with the Euclidean distance for every k. We observe that the search time per expansion increases when the neighborhood size increases.
- **On ANYA.** ANYA planner outperforms all the algorithms, except Theta* in Game Maps in the average number of nodes expanded.
- **On Theta*.** Theta* outperforms Regular A* for all values of k.

6 References

1. Rivera, N., Hernández, C., Hormazábal, N. and Baier, J.A., 2020. The 2^k Neighborhoods for Grid Path Planning. Journal of Artificial Intelligence Research, 67, pp.81-113.
2. Daniel, K., Nash, A., Koenig, S. and Felner, A., 2010. Theta*: Any-angle path planning on grids. Journal of Artificial Intelligence Research, 39, pp.533-579.
3. Harabor, D.D., Grastien, A., Öz, D. and Aksakalli, V., 2016. Optimal any-angle pathfinding in practice. Journal of Artificial Intelligence Research, 56, pp.89-118.