§CS 520: Assignment 1
Fast Trajectory Replanning

Xiaoyang Xie
167008240

Yikun Xian
168000142

Department of Computer Science
Rutgers University, New Brunswick, NJ

09 Octorber 2015

Abstract

Heuristic search algorithms like A* can be adapted to solving path planning problems of directed goal and unknown environment In this report, we mainly discuss and evaluate three variants of A* algorithms, namely Repeated Forward A*, Repeated Backward A* and Adaptive A*. In the experiment, we first generate two sets of 1000 random grid-based maps, where one follows the assignment requirement containing approximate 30% obstacles and 70% roads, and the other is the set of corridor-like mazes generated by DFS with randomly expanded nodes. Then we compare three algorithms by such evaluation indices as number of expanded nodes, number of explored nodes, total cost of steps, optimal cost of steps, etc. The result shows that 1) the average number of expanded nodes per map and explored nodes per map of Adaptive A* are respectively 28.06% and 24.00% less than those of Repeated Forward A*; 2) cost of steps for all three algorithms is pretty much the same. This indicates that Adaptive A* is greatly optimized in path replanning phase, while there is no significant improvement in actual moving phase. Finally, we discuss and calculate how to optimize data structures to store states as many as possible within only 4M memory. This is the practical problem in the situation where computational resources are rare and precious.

Part 0 Setup Environment

We simulate all path finding processes based on the framework of GridWorld[1], an AP case study project from collegeboard [1]. It provides graphical user interface based on Java AWT where visual objects can interact and perform customized actions in a two-dimensional grid map. In the next part, we will first illustrate original GridWorld framework and our enhancement of displaying colored path. This mainly involves the engineering work, so if you want to directly delve into algorithm analysis, please skip it.

0.1 GridWorld Architecture and Modification

The framework structure of original GridWorld project are divided into four parts: Actor, Grid, GUI and World, as shown in Figure 1. The Actor package contains objects whose behavior on the map can be arbitrarily defined by rewriting following method in each inherited class:

[1] https://www.collegeboard.org/

```
@Override
public void act()
```

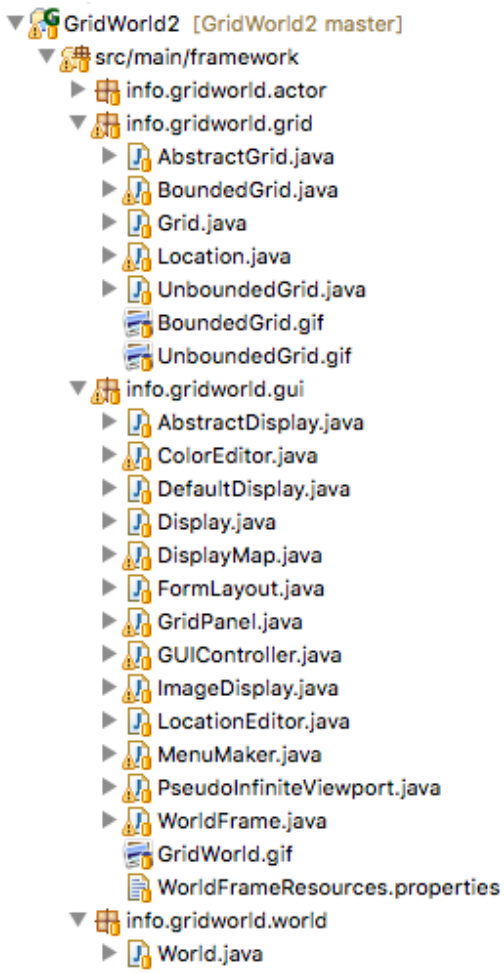

Figure 1: GridWorld package structure

The Grid package defines bounded grid map features and connection to actors. The GUI package encapsulates low-level Java AWT to provide APIs for visualization and interactions of map and actors. The World package provides high-level integration of actors and world.

In order to visualize the presumed unblocked path and the set of nodes expanded and explored after each planning, we enhance the *GridPanel* class in the GUI package by implementing following method:

```
private void drawColoredLocations(Graphics2D g2)
```

Meanwhile, following five abstract methods are added to *Grid* class so that inherited classes like *BoundedGrid* and *UnboundedGrid* should provide implementation of how to configure colors on each grid.

```
ArrayList<Location> getColoredLocations();
Color getColor(Location loc);
void putColor(Location loc, Color color);
void removeColor(Location loc);
void resetColors();
```

0.2 Maze Generation Algorithm

0.3 How to Run

Our project is

Part 1 Understanding the Methods

Part 2 The Effects of Ties

Part 3 Forward vs. Backward

Part 4 Heuristics in the Adaptive A*

Part 5 Heuristics in the Adaptive A*

Part 6 Memory Issues

References

[1] CollegeBoard. AP central gridworld case study. http://apcentral.collegeboard.com/apc/public/courses/teachers_corner/151155.html. [Online; accessed 2-October-2015].

[2] Sven Koenig and Maxim Likhachev. Real-time adaptive a*. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 281–288. ACM, 2006.

[3] Wikipedia. Maze generation algorithm — wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Maze_generation_algorithm&oldid=679876968. [Online; accessed 4-October-2015].