

南 京 理 工 大 学

智能计算技术实验三

姓 名：蒋旭钊 学 号：918106840727

学院(系): 计算机科学与工程学院

专 业：计算机科学与技术

课 程：智能计算技术

2021年 11月

一. 问题重述

根据章节《5.3 状态空间的启发式搜索》中的 A*算法，实现小车寻路问题。

生成一个 $N \times N$ 的二维网格，随机指定一些格子为障碍，左下角有辆车（占一个格子）要去右上角，使用 A*算法先计算起点到终点的不撞到障碍的最短路径。计算完后，利用路径上每个点的局部曲率，以及到周围最近障碍的距离来控制车辆的速度。要求使用模糊逻辑建立曲率小，离障碍距离远，车速快三个模糊集合，构建曲率小——车速快的关系矩阵，距离远——车速快的关系矩阵，根据 A*路径上每个点实际的曲率和离障碍物的距离，使用模糊推理及过规则合成，得到目标的瞬时速度，车辆按瞬时速度从左下到右上沿着 A*路径行驶。

二. 算法介绍

A*算法：

- ① 把起点加入 Open List;
- ② 遍历 Open List 中的节点，寻找到当前 F 值最小的节点，把他作为当前要处理的节点 p;
- ③ 把这个节点从 Open List 中删除并放入 Close List 中;
- ④ 遍历节点 p 的每一个相邻节点 s，如果节点 s 在 Close List 中，则忽略对节点 s 的判定。如果 s 既不在 Open List 里又不在 Close List 里，则把 s 加入 Open List 中，并且把节点 p 设置为节点 s 的父方格，计算节点 s 的 F，G 和 H 值。如果节点 s 已经在 Open List 中，检查这条路径（即经由节点 p 到达节点 s）是否更好，用 G 值作参考。更小的 G 值表示这是更好的路径，如果经过 p 到达节点 s 的路径更优，则把 s 的父亲设置为 p，并重新计算它的 G 和 F 值，如果原来的路径更优，则保持节点 s 的 F、G、H 值以及父节点的值。
- ⑤ 若 Open List 为空，则没有找到目标最短路，查找失败。
- ⑥ Open List 不为空，遍历 Open List，若 F 值最小的节点为目标节点，则找到目标最短路，否则返回第 2 步。（原 blog 中错误的地方）
- ⑦ 从终点开始，每个节点沿着父节点移动直至起点，即得到路径信息。

模糊推理：

当前有条件论域和结论论域，得到条件论域到结论论域的模糊关系矩阵 R。通过输入的条件模糊向量于模糊关系 R 的合成进行模糊推理，得到结论的模糊向量，然后采用适当的决策规则，将模糊结论转换为精准量。

三. 实现思路

首先进行 A*算法的运行，然后基于 A*算法的路径，判断出小车通过模糊推理算出的行进速度。**主函数为：AiCar.java 中的 main 函数**

A*算法的实现:

Coord 类: 按照二维数组的特点, 坐标原点在左上角, 所以 y 是高, x 是宽, y 向下递增, x 向右递增, 我们将 x 和 y 封装成一个类, 好传参, 重写 equals 方法比较坐标 (x, y) 是不是同一个。

Node 类: 封装路径结点类, 字段包括: 坐标、G 值、F 值、父结点, 实现 Comparable 接口, 方便优先队列排序。

MapInfo 类: 是 A 星算法输入的所有数据, 封装在一起。

Astar 类: 定义了路径代价的估算策略 $F=G+H$ 。G 表示的是从起点到当前结点的实际路径代价, H 表示当前结点到达最终结点的估计代价, F 表示当前结点所在路径从起点到最终点预估的总路径代价。

其中 G 的计算, 由于横竖移动距离: 斜移动距离=1: 1.4, 我们在计算代价时取整数 10 和 14, H 计算采用曼哈顿距离进行估算。

通过运行 A*算法, 将运行中的路径以 `Stack<Node> path` 形式保存; 将其中一系列的 Node 信息传入模糊判断函数, 从而进行瞬时速度计算。

模糊判断的实现:

FuzzySet 类: 首先定义了“距离远——车速快的关系矩阵 DIS_SPEED”和“曲率小——车速快的关系矩阵 CUR_SPEED”, 将两者进行模糊关系的合成得到 MIX_SPEED。通过 `nearsetBar()` 函数计算最近障碍的距离以此构造输入条件向量 `disVector`, 通过 `getCurveVector` 计算曲率以此构造输入条件向量 `curVector`, 将两者进行模糊混合, 最后通过 `getSpeed()` 算出最终速度。

四. 实验结果

本实验没有进行输入, 定义了起点和终点, 展示了小车的运行轨迹, 打印出了小车在每个点的瞬时速度。

输出样例:

地图信息:

Map is:

```
|
|
|      0 0              0 0 0
|      0              0 0
|      0 0 0 0 0 0 0
|      0      0
|      0              0
|
```

The total cost is: 282

The routine of the car is:

```
|
|      * * * * * * * * * *
|    * 0 0              0 0 0 *
|    *  0              0 0  *
|    *  0 0 0 0 0 0 0      *
|    *  0 * * 0 * * * *
| *    0      *  0
|
```

速度信息:

The speed at (0,6) is 4 block/s
The speed at (1,5) is 1 block/s
The speed at (1,4) is 1 block/s
The speed at (1,3) is 2 block/s
The speed at (1,2) is 4 block/s
The speed at (2,1) is 2 block/s
The speed at (3,1) is 2 block/s
The speed at (4,1) is 2 block/s
The speed at (5,1) is 1 block/s
The speed at (6,1) is 2 block/s
The speed at (7,1) is 1 block/s
The speed at (8,1) is 1 block/s
The speed at (9,1) is 2 block/s
The speed at (10,1) is 2 block/s
The speed at (11,1) is 2 block/s
The speed at (12,1) is 4 block/s
The speed at (13,2) is 4 block/s
The speed at (12,3) is 4 block/s
The speed at (11,4) is 4 block/s
The speed at (10,5) is 2 block/s
The speed at (9,5) is 2 block/s
The speed at (8,5) is 2 block/s
The speed at (7,5) is 4 block/s
The speed at (6,6) is 4 block/s
The speed at (5,5) is 2 block/s