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NxN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。

# 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：**

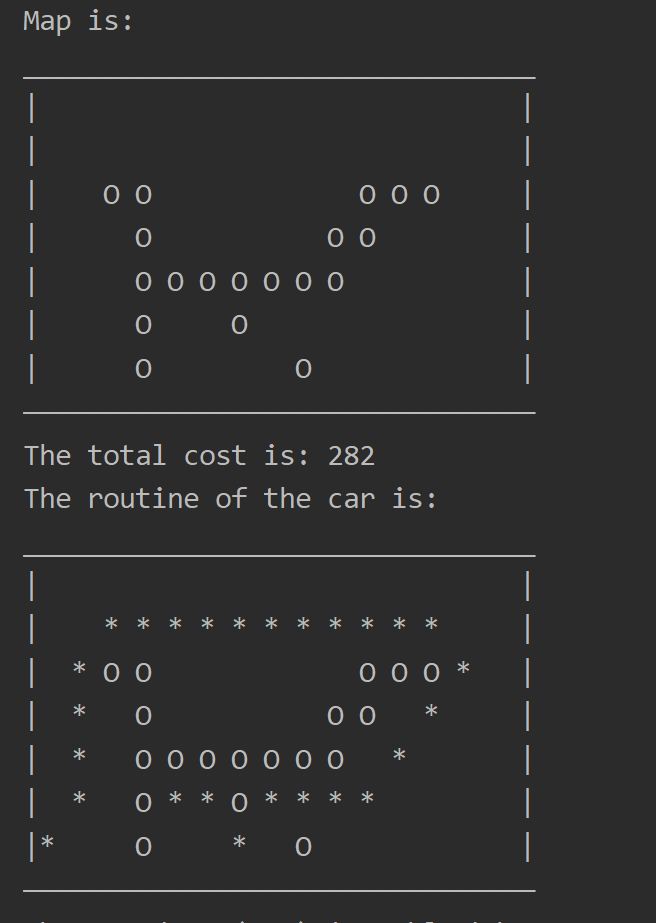
FuzzySet类：首先定义了“距离远——车速快的关系矩阵DIS\_SPEED”和“曲率小——车速快的关系矩阵CUR\_SPEED”,将两者进行模糊关系的合成得到MIX\_SPEED。通过nearsetBar（）函数计算最近障碍的距离以此构造输入条件向量disVector，通过getCurveVector计算曲率以此构造输入条件向量curVector，将两者进行模糊混合，最后通过getSpeed（）算出最终速度。

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：



速度信息：

