



南京理工大学
NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

智能计算技术 课程实验报告

学生姓名 胡楚麒

学 号 920106840416

指导教师 王欢

2023 年 12 月

实验一 基于 A*算法的无人车路径规划

一、实验目的

实验前要求掌握状态空间图的知识表示方法，基于状态空间图的知识搜索一般过程。掌握基于状态空间图的知识表示的要素，掌握状态空间图的知识搜索流程图，通过最短路径求解问题培养应用图搜索解决实际问题的能力。

通过实验掌握A*智能搜索算法及其在无人车路径规划中的应用。提升使用智能搜索算法解决实际问题的能力。

二、实验要求

C/C++、Python、matlab、Java等软件编写程序。

三、实验内容

生成一个NxN的二维网格，随机指定一些格子为障碍，并设定左上角有辆无人车（占一个格子）要去右下角，使用A*算法为该无人车计算起点到终点的不到障碍的最优路径。

四、开发环境

语言：python

系统：win10

集成开发环境：PyCharm

五、实验思路

1.地图设置

首先，我们把地图栅格化，把每一个方格的中心称为节点；这样我们便把我们的搜索区域简化为了2维数组。数组的每一项代表一个格子，它的状态只有两种要么可走和要么不可走。通过计算出从初始节点到目标节点需要走过哪些方格，就找到了路径。一旦路径找到了，便从一个方格的中心移动到另一个方格的中心，直至到达目的地。

2.A*算法的搜索

一旦我们把搜寻区域简化为一组可以量化的节点后，我们下一步要做的便是查找最短路径。在A*中，我们从起点开始，检查其相邻的方格，然后向四周扩展，直至找到目标。

从起点A开始，定义A为父节点，并把它加入到OPEN表中。现在OPEN表里只有起点A，后面会慢慢加入更多的项。例如一个节点A周围共有8个节点，定义为子节点。将子节点中可走的节点放入OPEN表中，成为待考察对象。若某个节点既未在OPEN表中，也没在CLOSE表中，则表明还未搜索到该节点。

此外，A*算法的代价函数的设置也至关重要。移动代价评价函数为： $f(n)=g(n)+h(n)$ 。 $f(n)$ 是从初始状态经由状态n到目标状态的代价估计， $g(n)$ 是在状态空

间中从初始状态到状态 n 的实际代价， $h^*(n)$ 是从状态 n 到目标状态的最佳路径的估计代价，通常无法直接得出。

六、实验结果

1.搜索过程及结果



如上图所示，绿色为起点，蓝色为障碍物，红色代表最终目标。其中我们可以看到，随着程序的运行，红色方块逐渐漫开，大体上朝终点前进。红、黄、蓝即是程序在搜索过程中访问到的点，我们可以看到搜索过程基本朝着终点方向铺开，很好地避开了无用路径。说明对于 $h^*(n)$ 的选择是合理的。

七、实验心得

通过实验，我深入理解了A*算法在无人车路径规划中的应用。首先，将地图栅格化，形成状态空间图，简化搜索区域。其次，A*算法通过启发式搜索，从起点到目标点寻找最短路径。代码中使用曼哈顿距离作为启发式函数，但也可根据实际情况选择其他函数。在实验中，我成功生成N×N的二维网格，运用A*算法找到无人车避开障碍的最优路径。这锻炼了我图搜索解决实际问题的能力，加深了对路径规划和算法优化的理解。通过调整地图和算法参数，我更好地掌握了A*算法的原理和应用，提升了在路径规划领域的实际应用水平。

总之本次实验还算较为简单，程序能够很好地给出最短路径。

实验二 博弈树搜索

一、实验目的

通过实验掌握与或树的知识表示方法，基于与或树的知识搜索一般过程。掌握基于与或树的知识表示的要素，掌握与或树的知识搜索流程图，通过博弈树求解问题培养应用与或树搜索解决实际问题的能力。

二、实验要求

C/C++、Python、matlab、Java等软件编写程序。严格按照与或树的搜索流程图编写程序，必须使用alpha和beta减枝，并比较它使用和不使用alpha和beta减枝的优缺点。

三、实验内容

实践博弈树搜索——“5x5格子的一字棋问题”，即五子棋，参照课件PPT上的例子来实现。要求是Max方和Min方都用博弈树来决策，或者一方使用博弈树决策，一方随机或手工走棋，并使用alpha和beta减枝。

四、开发环境

语言: python

系统: win10

集成开发环境: PyCharm

五、实验思路

1.估价函数

算法的总体思路与课堂上介绍的 3x3 的棋盘类似。只不过这里胜利的条件发生了变化，而且为了便于更改棋盘的大小，即对于一个五子棋问题，可以将棋盘的大小设置为5x5。

2.博弈树搜索及 alpha-beta 剪枝

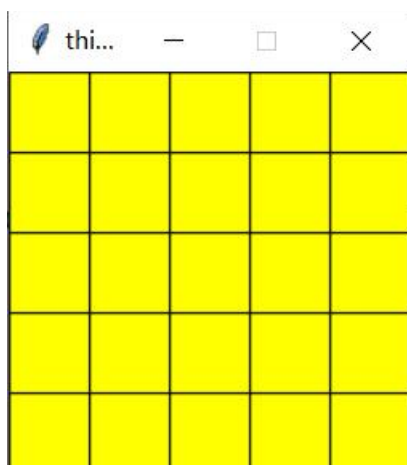
默认设置深度为3，即MAX 走一步，MIN 走一步。

此外在 AI 生成博弈树并搜索以确定下一步时，程序还会给出搜索过程及该次搜索使用 alpha-beta 剪枝的次数，以及总的搜索次数。

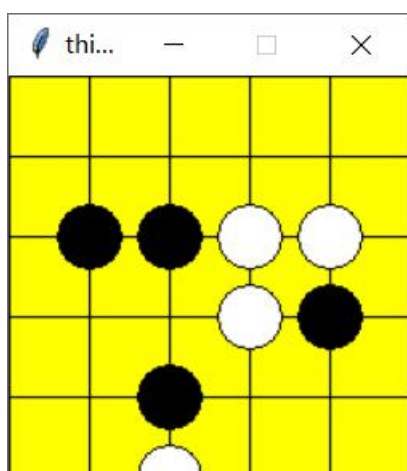
3.棋盘绘制

使用开源库 graphics

六、实验结果



初始棋盘如上。



决策过程中如上，附带部分剪枝日志。

```
[ (2, 2), (3, 3), (4, 3), (4, 2), (2, 4), (2, 5), (1, 2), (4, 4), (5, 5) ]
-5.0alpha:-99999999beta:0.0
[ (2, 2), (3, 3), (4, 3), (4, 2), (2, 4), (2, 5), (1, 2), (4, 4) ]
5.0alpha:-0.0beta:5.0
[ (2, 2), (3, 3), (4, 3), (4, 2), (2, 4), (2, 5), (1, 2), (4, 4), (5, 4) ]
5.0alpha:-0.0beta:5.0
[ (2, 2), (3, 3), (4, 3), (4, 2), (2, 4), (2, 5), (1, 2), (4, 4), (5, 3) ]
5.0alpha:-0.0beta:5.0
[ (2, 2), (3, 3), (4, 3), (4, 2), (2, 4), (2, 5), (1, 2), (4, 4), (4, 5) ]
0.0alpha:-5.0beta:0.0
[ (2, 2), (3, 3), (4, 3), (4, 2), (2, 4), (2, 5), (1, 2), (4, 4) ]
0.0alpha:-99999999beta:0.0
[ (2, 2), (3, 3), (4, 3), (4, 2), (2, 4), (2, 5), (1, 2), (1, 4) ]
本次共剪枝次数: 138
本次共搜索次数: 2057
```

七、实验心得

通过实验二，我深入学习了与或树和Alpha-Beta剪枝在博弈树搜索中的应用。在解决五子棋问题时，我灵活运用与或树建模游戏状态，设计评估函数和决策过程。Alpha-Beta剪枝的引入有效减少了搜索空间，提高了算法效率。与或树的灵活运用使得Max和Min方能够轮流决策，通过评估函数选择最优走法。实验中的代码框架为解决复杂问题提供了基础，通过比较带有和不带有Alpha-Beta剪枝的版本，我更深刻地理解了剪枝对搜索效率的影响。这次实验提升了我在博弈树搜索和优化算法方面的实际应用能力。

实验三 使用概率推理或可信度推理实现图像区域的材质识别

一、实验目的

通过实验掌握以贝叶斯概率或可信度为基础的不确定推理方法，并根据实际问题灵活应用这些算法。

二、实验要求

C/C++、Python、matlab或Java等编写程序。

三、实验内容

给定100幅图像及其对应的材质分割与识别结果（每个独立的区域表示一种材质，区域颜色表示材质类别：混凝土/石头/砖瓦、钢材/钢筋、木材、塑料四种材质类别）。

手工标注50幅测试图像（每人选择对应序号的文件夹中的50幅测试图像，序号见附图，第一幅已用红笔标注示意），获得其材质区域分割结果，然后使用算法对每个手工标识的区域提取颜色、纹理等特征，进行材质识别（推理）。

比如，可利用训练集获得四个材质类别的 $p(C_i)$ ，以及 $p(X|C_i)$ 的参数，对每幅测试图像，利用标注的区域计算 $p(X|C_i)$ ，而后根据贝叶斯公式推理出它的类别（即使得 $p(C_i|X)$ 最大的类别）；或者也可以利用可信度方法中的可信度计算公式 $CF(H)=CF(H,E)*\max(0,CF(E))$ 求解。

四、开发环境

语言: python

系统: win10

集成开发环境: PyCharm

五、实验思路

通过对自己分配的63号文件使用anlabeling进行标注，学习了如何使用软件对图片不同物体进行标注区分。利用辅助的100训练集获得四个材质类别的 $p(C_i)$ ，以及 $p(X|C_i)$ 的参数，对每幅测试图像，利用标注的区域计算 $p(X|C_i)$ ，而后根据贝叶斯公式推理出它的类别。其中，标注类别及序号如下图所示。

名称	序号	颜色
Metal	1	红色
Plastic	2	绿色
Stone	3	黄色
wood	4	蓝色

六、实验结果



七、实验心得

通过图像材质识别实验，我深刻领悟了贝叶斯概率在不确定性问题中的实际应用。在使用Python编写程序的过程中，我成功运用了先验概率和条件概率，实现了对手工标注测试图像中材质区域的准确识别。特别重要的是对图像特征的巧妙提取，我选择了颜色和纹理等关键特征，以确保对图像材质的有效表征。通过实际代码的编写，我不仅提高了编程能力，还深入理解了贝叶斯推理方法在图像处理和模式识别领域的实际应用，为我在人工智能领域的学习提供了有益经验。