

中山大学计算机学院

人工智能

本科生实验报告

(2022 学年春季学期)

课程名称: Artificial Intelligence

教学班级	教学一班	专业 (方向)	计算机科学与技术
学号	21307018	姓名	李浩辉

一、 实验题目

完成盲目搜索（我这里选择了迭代 DFS）与 A*搜索解决迷宫问题。

二、 实验内容

1. 算法原理

(1) 迭代加深算法

可以分为三个算法递推形成：

DFS（深度优先搜索）----->深度受限搜索----->迭代加深搜索

DFS 递归定义

a、访问节点

b、依次 DFS 该节点的子节点

深度受限搜索

提前设置好一个 limit, 在 DFS 前提下若某一个节点访问时深度(代价)

已经超过 limit, 则不再继续 DFS 其子节点。

迭代加深搜索

依次由 0 到无穷增长 Limit, 作深度首先搜索。

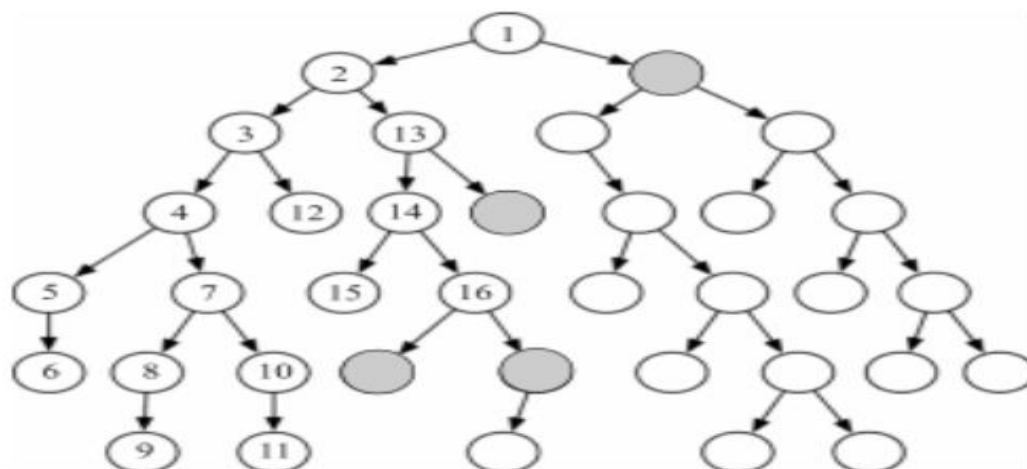


图 1: DFS 搜索次序

(2) A*算法

算法定义

- 准备两个 List, open 表和 closed 表, open 初始只装载起点, 两个表都需要记录父指针 (起点父指针可以设置为空)
- 准备一个启发式函数 $h(n)$, 用以估计与终点的代价。同时需要 $g(n)$ 记录起点到 n 顶点的代价。得到:

$$f(n) = g(n) + h(n)$$

- open 列表里面删除其中 $f(n)$ 最小的节点, 加入到 closed 列表。
- 将上述节点的邻节点作以下操作:

如果它已经出现在 open, 更新 f 值和父指针 (如果 f 更小);

如果它已经出现在 closed, f 更小则移动至 open 并更新;

如果均未出现过, 加入到 open 记录即可。

- 循环到 c 继续完成算法, 直到得到终点在 closed 追踪得到路径。

启发式函数 $h(n)$

$h(n)$	理解	性能
$h(n)=0$	$h(n)$ 不起作用，退化为 BFS 算法	可得到最优路径，盲目搜索
$h(n) \leq h^*(n)$		可得到最优路径，并且性能取决于 $h(n)$ 差异值
$h(n)=h^*(n)$	最佳	最优路径且不会扩展旁路
$h(n) > h^*(n)$		不保证最优路径

性质	定义	作用
可采纳性	$h(n) \leq h^*(n)$	不进行环检测前提下可得到最优路径
单调性	$h(n) \leq \text{cost}(n, n^*) + h(n^*)$	任何情况都可以得到最优解

2. 伪代码&关键代码

(1) 迭代加深搜索

//深度受限, limit 记录限制, pre 记录上一步操作便于回溯

```
def DFS_limit ( limit, pre ):
```

```
    if 已达终点:
```

```
        返回 true
```

```
    if 路径长度已经超过 limit:
```

```
        返回 false
```

```
    if 可左移:
```

```
        当前状态左移
```

```
        if DFS (limit,1) 返回 true
```

```
            返回 true
```

```
        回溯状态即右移
```

```
    if 可下移:
```

```
        ..... //同理
```

```
    if 可右移:
```

```
        ..... //同理
```

```
    if 可上移:
```

```
        ..... //同理
```

```
    返回 false
```

//main 函数

limit 从 0 到 无穷 (行数*列数) :

```
    DFS_limit (limit, 0) //参数 limit, 以及 0 代表上一步无操作
```

得到结果就停止循环并输出结果



(2) A*算法

Def Astar ():

*//A*算法*

// h (n)设置为与终点的几何距离

//这个 h (n) 同时满足可采纳性以及单调性

新建两个列表 open, closed

open 列表放入起点坐标以及对应 f 值, 其父指针设为空

While open 列表不为空:

 取 open 列表中 f 值最小的项

 If 是终点:

 根据父指针依次递推得到路径

 返回路径

 退出当前函数

//不是终点则进行以下操作

 Open 删除该节点

 在 closed 列表中加入这一项

 循环对这一项的邻节点以下操作:

 if 该节点在 open 列表中出现:

 If 当今 f 值更小:

 更新 f 值以及父指针

 elif 该节点在 closed 列表出现:

 If 当今 f 值更小:

 Closed 中删除该节点

 open 列表新增以及更新如今 f 和父指针

 Else 即该节点未出现在 open 以及 closed 中

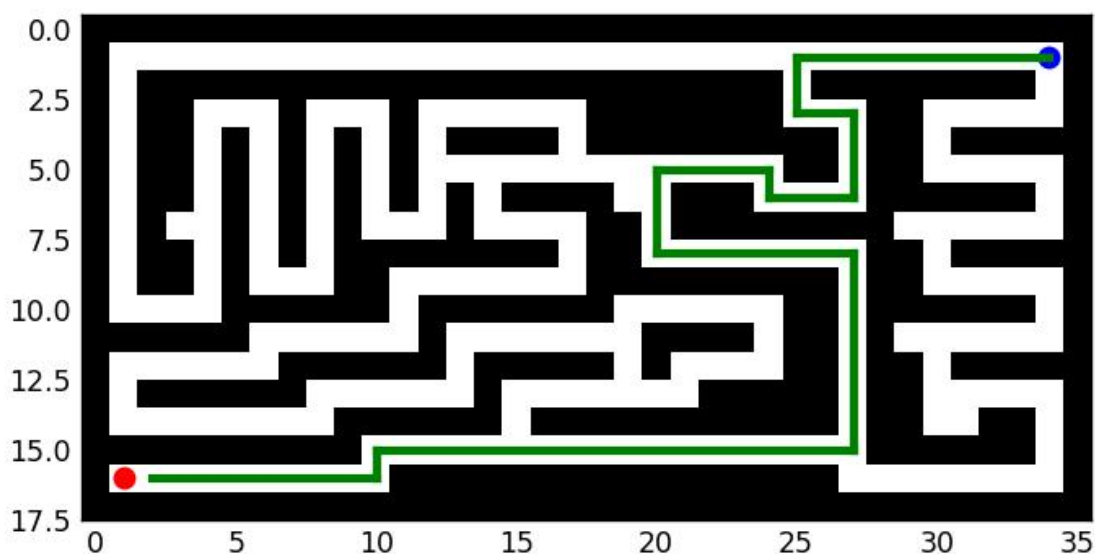
 在 open 中新增该节点

3. 创新点&优化

(1) $h(n)$ 的选用为几何距离，这个 $h(n)$ 同时满足可采纳性以及单调性（三角形两边之和大于第三边可证明），因此任何时候 A* 算法都可以得到最优解。相比于曼哈顿距离运行效率可能更低一点，但是在数据量不大的情况下相差不多。

(2) 本次实验最大的亮点应该是对迷宫的可视化，为了更好展示迷宫内容调用了 numpy 等库（信号与系统课程中学习过）的内容更好将整个迷宫可视化，如下图。

蓝色为起点，红色为终点

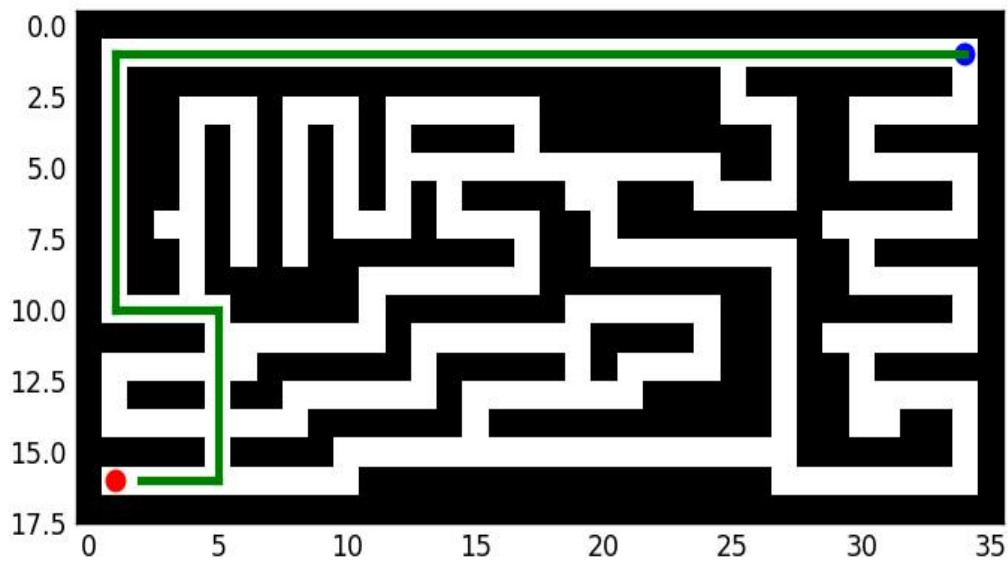
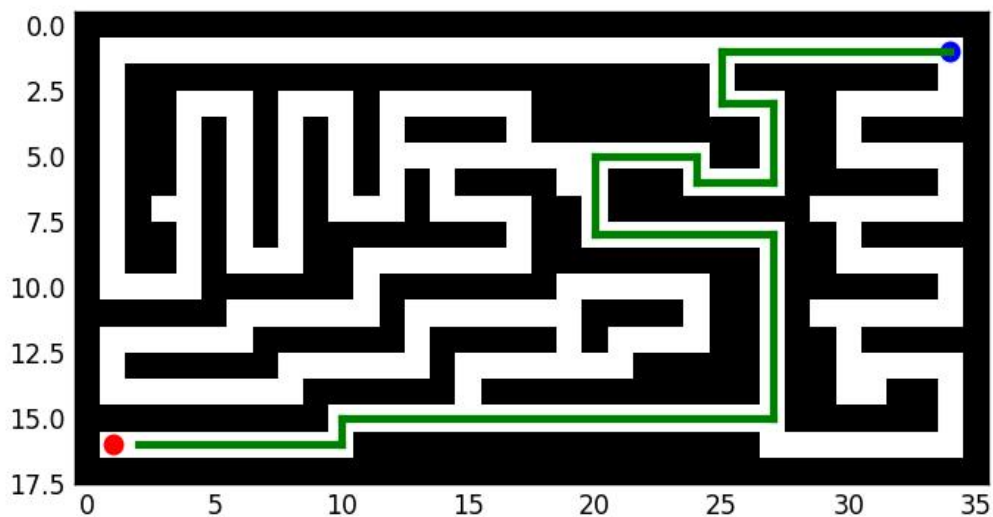




三、 实验结果及分析

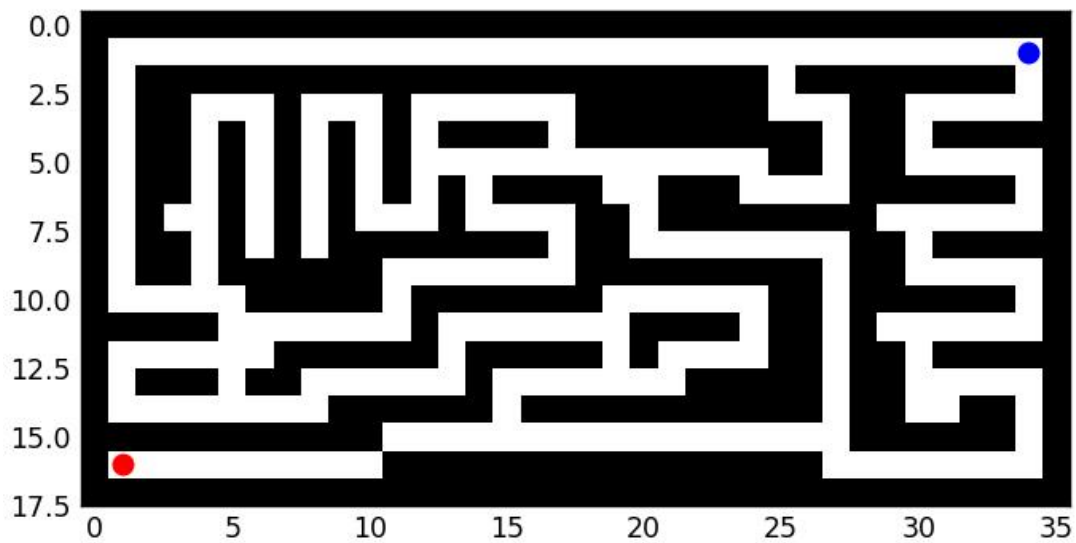
1. 实验结果展示示例

在原地图基础上随机更改一些地图设置：





如果没有路径则不会显示路径



2. 评测指标展示及分析

算法	时间 1	时间 2	时间 3	平均时间
迭代加深	0.00388	0.00594	0.00468	0.00483
A*算法	0.14916	0.12496	0.13525	0.13645

（以上时间为生成路径的时间，不包括生成迷宫可视化图的时间）

由此可以看出：

A*算法相较于迭代加深 DFS 时间优越性十分明显，甚至在小的数据面前已经有近乎两个数量级的领先。



四、 参考资料

《人工智能》——贲可荣、张彦铎