**人工智能原理实验**

**实验二 算法求解八数码问题实验**

**组长：**周俊峰

**组员：**周俊峰

1. **实验目的**

熟悉和掌握启发式搜索的定义，估价函数和算法过程，并利用A\*算法求解N数码难题，理解求解流程和搜索顺序

1. **实验内容**

以8数码问题和15数码问题1为例实现A\*算法的求解程序，要求设计两种不同的估价函数。

1. **实验要求**

1.分析不同的估价函数对A\*算法性能的影响。

2.根据宽度优先搜索算法和A\*算法求解8、15数码问题题的结果,分析启发式搜索的特点

3.画出A"算法求解N数码问题的流程图。

5.总结实验心得体会。

1. **实验内容**

**基本思想：**

定义一个评估函数f，对当前的搜索状态进行评估，找到一个最有希望的节点进行扩展。评估函数的形式如下：

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

其中：

n是被评估的节点

f(n) 是节点n从初始点到目标点的估价函数。

g(n) 是在状态空间中从初始节点到n节点的实际代价。

h(n)是从n到目标节点最佳路径的估计代价。

f(n)=g(n)+h(n) 表示从初始节点s经过节点n到目标节点g的评估代价

保证找到最短路径（最优解）的条件，关键在于估价函数h(n)的选取。估价值h(n)<= n到目标节点的距离实际值，这种情况下，搜索的点数多，搜索范围大，效率低。但能得到最优解。如果估价值>实际值, 搜索的点数少，搜索范围小，效率高，但不能保证得到最优解。

搜索中利用启发式信息，对当前未扩展结点根据设定的估价函数值选取离目标最近的结点进行扩展，从而缩小搜索空间，更快的得到最优解，提高效率。

根据需要open表和close表记录结点信息。OPEN表保存所有已生成而未考察的节点，CLOSED表中记录已访问过的节点。同时open表和close表里面所包含的结点状态不重复。

**整体思路就是根据代价最小进行扩展结点直到找到目标。**

其中五个估价函数(calH)代码分别为：

function cost=calH0(node, dis)

function cost=calH3(node, dis)

a= reshape (node.con,1, nosDim^2);

cost = nixudui(a)\*2;

end

function cost=calH4(node, dis)

cost=nixudui(node.con)\*3+

length(find(node.con~=dis));

end

cost = 0;

end

function cost=calH1(node, dis)

cost = length(find(node.con~=dis));

end

function cost=calH2(node, dis)

nosDim = 3;%%维度

cost = 0;%%初始化

for i=0:(nosDim^2-1)%%寻找位置不符的数

[m,n]=find(node.con==i); %当前

[x,y]=find(dis == i); %目标

cost=abs (x-m) + abs(n-y) +cost;

end

end

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **表格一** | **启发函数 h(n)** | | | | |
|  | **宽度优先算法（H0）** | **不在位数**  **（H1）** | **（H2）** | **（H3）（1倍/2倍）** | **（H4）** |
| **初始状态** | **初始状态 目标状态**  initMatrix (:)=**[4 1 5 3 8 7 6 0 2] [0 1 2 3 4 5 6 7 8]**  **02** | | | | |
| **目标状态** |
| **最优解**  **(移动步数)** | **18** | **18** | **18** | **18/18** | **24** |
| **扩展节点数**  **(close表）** | **15379** | **1093** | **460** | **211/186** | **100** |
| **生成节点数**  **(open表）** | **8187** | **642** | **272** | **129/126** | **145** |
| **运行时间（秒）** | **515.709** | **2.166** | **0.472** | **0.112/0.108** | **0.091** |

**8数码总结**

f(n)=g(n)+h(n),由CH0-CH4的方法不难看出，保持了搜索的宽度有限成分，g（n）的比重越大，越倾向于宽度优先搜索方式，便于找出最优解，但从CH0的时间和节点可以看出最优解18得出会耗费几乎近9分钟。（CH3有两个数据对应相应的逆序对倍数）

h(n)的比重越大，表示启发性能越强，在特殊情况下，如果只希望找到到达目的结点的路径而不关心会付出什么代价，则g(n)的作用可以忽略。当h(n)>>g(n)时，有利于提高搜索的效率，但影响搜索的完备性。如CH4虽用24步但只需0.091秒就跑完。

1. **实验总结**

在实验中，估计函数由宽度和启发式构成，通过8数码的matlab实验中，我们体会到算法好坏的重要性，也知道了结点越多，耗时越多，但却能最接近最优解；相反，若想追求搜索效率，则可以牺牲节点，用步数换取，在学和行的实践中对估计函数和A搜索算法有了更深层的理解。