

A-star算法

——一种启发式搜索算法



小组编号: G03

小组成员: 刘书宇 31801323

童峻涛 31801341

1.算法介绍

01

A*算法是一种求解最短路径最有效的直接搜索算法，也是目前最有影响的常用启发式算法

02

A*算法使用启发来引导搜索，同时确保计算出的路径代价最小

03

A*算法的启发函数为 $F(n) = G(n) + H(n)$ ，其中 $F(n)$ 是从初始状态经由状态 n 到目标状态总代价的估计值， $G(n)$ 是衡量某一状态在图中的深度（通俗的说就是当前已经走的步数）， $H(n)$ 是从状态 n 到目标状态的最佳路径的估计代价

04

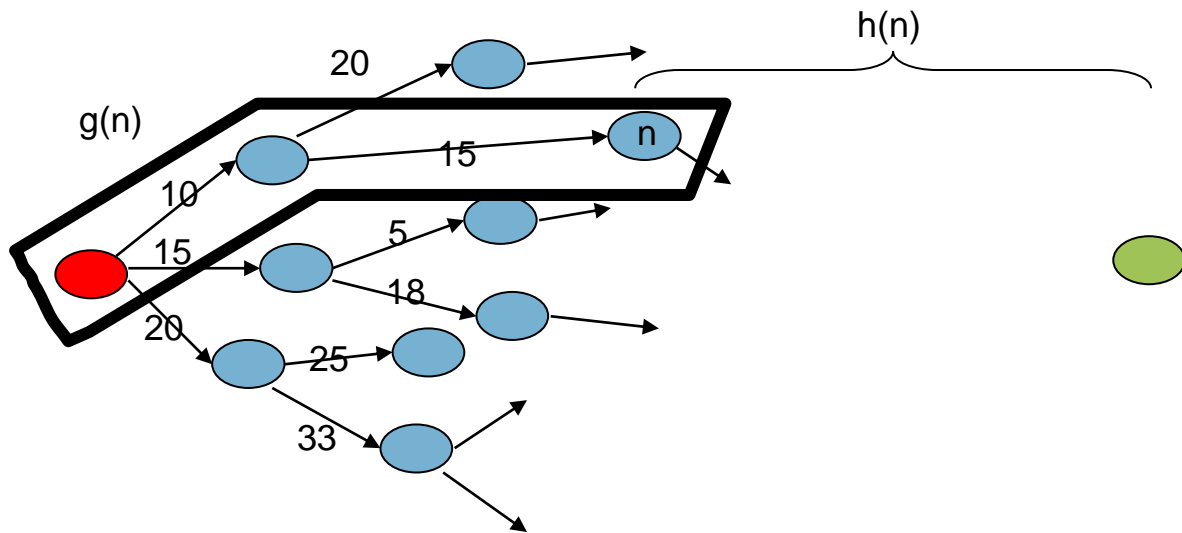
其中 $H(n)$ 估计代价在本次实验中采用 不同位数 和 曼哈顿距离

启发式函数为：

$$F(n) = G(n) + H(n)$$

$G(n)$ = “从起始点到达n的花费”

$H(n)$ = “估计从n到目标节点的最少的路径代价”



2. 八数码难题及拓展

2	8	3
1	6	4
7		5

初始状态



1	2	3
8		4
7	6	5

目标状态

问题描述:

3 × 3 九宫棋盘，放置数码为1-8的8个数字，剩下一个宫格，智能通过数字向空格的移动来改变棋盘的布局。

问题演变:

美国的科学魔术大师萨姆·洛伊德将三阶方阵扩大到了四阶方阵，使得运算规模急剧扩大。

解的存在性:

①如果一对数的前后位置与大小顺序相反，即前面的数大于后面的数，那么它们就称为一个逆序。

一个排列中所有逆序的总数叫做这个排列的逆序数。逆序数为奇数的排列叫做奇排列，逆序数为偶数的排列叫做偶排列。

②由初始状态和目标状态奇偶性相同的状态具有可达性。

2	8	3
1	6	4
7		5

初始状态



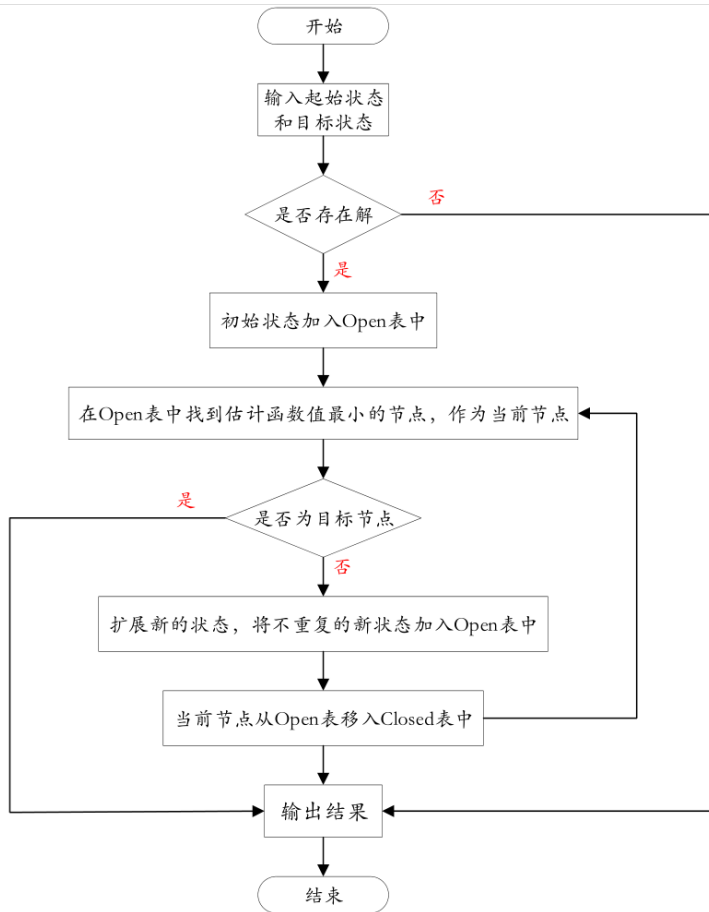
1	2	3
8		4
7	6	5

目标状态

2	8	3	1	6	4	7	5	和	
0	0	1	3	1	2	1	3	11	奇
1	2	3	8	4	7	6	5	和	
0	0	0	0	1	1	2	3	7	奇

逆序数计算：按顺序给出从第二个数开始的逆序数（0跳过）

3. 算法过程



4. 样例分析

2	8	3
1	6	4
7		5

初始状态

1	2	3
8		4
7	6	5

目标状态

Open表	Closed表
初始化: (S(4))	()
一次循环后: (A(5), B(7), C(7), D(7))	(S(4))
二次循环后: (E(6), B(7), C(7), D(7), F(8))	(S(4), A(5))
三次循环后: (G(5), B(7), C(7), D(7), F(8))	(S(4), A(5), E(6))
四次循环后: (H(4), B(7), C(7), D(7), F(8), I(8))	(S(4), A(5), E(6), G(5))
五次循环后: H为目的状态, 搜索成功	(S(4), A(5), E(6), G(5), H(4))

```
(n^2-1)puzzle ×
-----以下为运行过程-----
已经找到最优解!
搜索的次数: 11 移动总步数: 5

移动第 0 步时的状态如下:
[2, 8, 3]
[1, 6, 4]
[7, 0, 5]
总代价F为 6 当前深度G为 0 估计代价H为 6

移动第 1 步时的状态如下:
[2, 8, 3]
[1, 0, 4]
[7, 6, 5]
总代价F为 5 当前深度G为 1 估计代价H为 4

移动第 2 步时的状态如下:
[2, 0, 3]
[1, 8, 4]
[7, 6, 5]
总代价F为 6 当前深度G为 2 估计代价H为 4

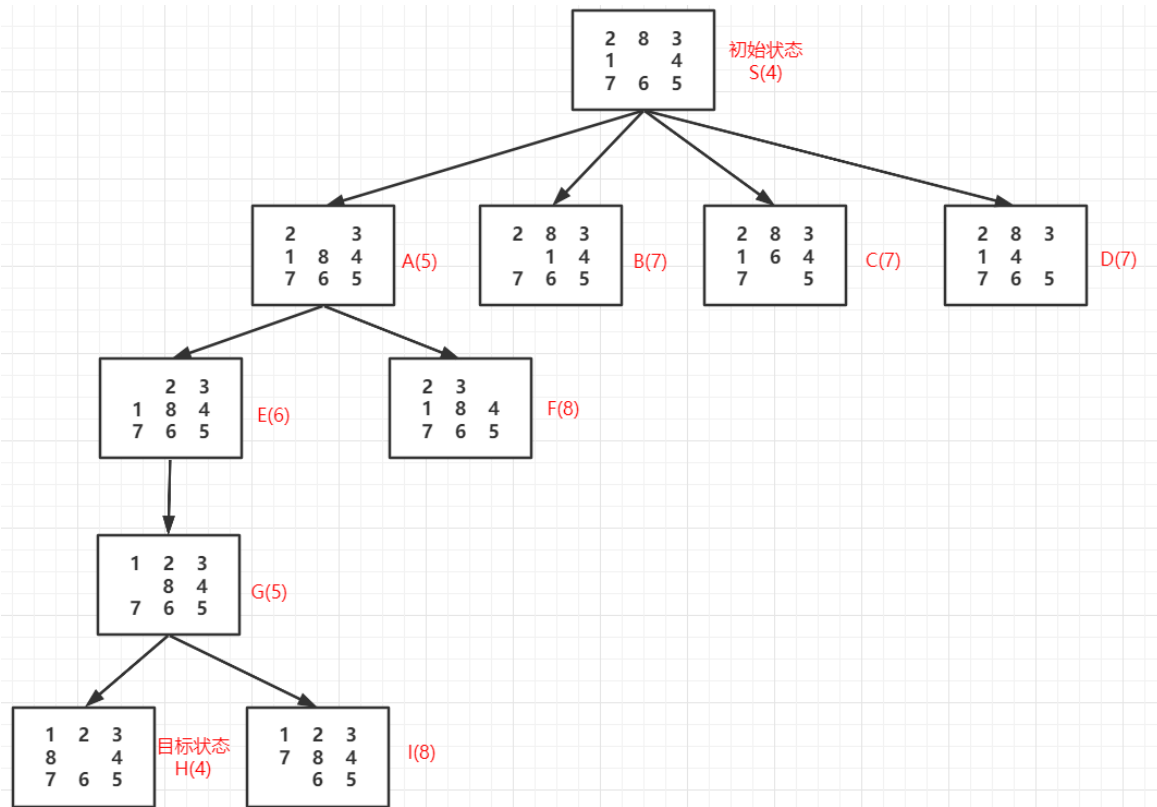
移动第 3 步时的状态如下:
[0, 2, 3]
[1, 8, 4]
[7, 6, 5]
总代价F为 7 当前深度G为 3 估计代价H为 4

移动第 4 步时的状态如下:
[1, 2, 3]
[0, 8, 4]
[7, 6, 5]
总代价F为 6 当前深度G为 4 估计代价H为 2

移动第 5 步时的状态如下:
[1, 2, 3]
[8, 0, 4]
[7, 6, 5]
总代价F为 5 当前深度G为 5 估计代价H为 0

运行花费的时间: 1164  毫秒(μs)
```

5. 搜索树状态



6.伪代码实现

```
open=[Stat]
closed=[]
while open不为空{
    从open中取出估价值f最小的节点n
    if n == Target
        return 从Stat到n的路径 //找到了!!!
    else{
        for n的每个子节点x{
            if x in open{
                计算新的f(x)
                比较open表中的旧f(x)和新f(x)
                if 新f(x) < 旧f(x){
                    删掉open表里的旧x, 加入新x
                }
            }
            else if x in closed{
                计算新的f(x)
                比较closed表中的旧f(x)和新f(x)
                if 新f(x) < 旧f(x){
                    remove x from closed
                    add x to open
                }
            }
            else {
                计算f(x) add x to open
            }
        }
        add n to closed
    }
}
```

7.实验结果

	启发函数h (n)					
	不在位数			曼哈顿距离		
初始状态	2	8	3	2	8	3
	1		4	1		4
	7	6	5	7	6	5
目标状态	1	2	3	1	2	3
	8		4	8		4
	7	6	5	7	6	5
生成节点数	11			9		
运行时间（微秒）	1236us			954us		

表1 不同启发函数h (n) 求解8数码问题的结果比较

	启发函数h (n)							
	不在位数				曼哈顿距离			
初始状态	5	1	2	4	5	1	2	4
	9	6	3	8	9	6	3	8
	13	15	10	11	13	15	10	11
	14		7	12	14		7	12
目标状态	1	2	3	4	1	2	3	4
	5	6	7	8	5	6	7	8
	9	10	11	12	9	10	11	12
	13	14	15		13	14	15	
生成节点数	132				136			
运行时间（微秒）	15658us				16011us			

表2 不同启发函数h (n) 求解15数码问题的结果比较

8. 实验结论

- 不同的**估价函数**对求解问题的解对搜索算法性能具有影响
- 随着N的增大，算法的时间空间**复杂度**都会随之逐渐增加



A*启发式算法的**特点**:

1. **完备性**: 肯定能找到最优解（除非不存在解）
2. **最优性**: 找到的解花费最小
3. **速度快**: 扩展更少的节点（取决于估价函数的选择）

9.参考资料

文献:

- [1]付宏杰,王雪莹,周健,周孙静,朱珠,张俊余.八数码问题解法效率比较及改进研究[J].软件导刊,2016,15(09):41-45.
- [2]温安国,李松年.N数码问题直接解及优化研究[J].计算机应用与软件,2010,27(05):266-268+277.
- [3] 靳海亮,王赢乐,袁鸣,陈梦龙.改进A*的高层建筑逃生路径规划算法研究[J].测绘通报,2019(11):17-21+25.
- [4]卜奎昊,宋杰,李国斌.基于A*算法的八数码问题的优化实现[J].计算机与现代化,2008(1):29-31.

博客:

- [1] Ta_Ex_ (2018-12-02) : 十五数码A*算法[https://blog.csdn.net/Ta_Ex_/article/details/84726264].
- [2] small_bright_ (2020-04-10) : 八数码问题的A*搜索算法[https://blog.csdn.net/small_bright_/article/details/105437321].
- [3] Ajinkya Sonawane (2018-09-16) : Solving 8-Puzzle using A* Algorithm.[<http://blog.goodaudience.com/solving-8-puzzle-using-a-algorithm-7b509c331288>].



兄弟 牛逼

Thank you

For your listening and watching.