

人工智能实验报告

A*算法

小组 lgd-cs

赖启东* 14347058

黎丁嘉 15336077

李振宇 15336092

刘键涵 15336110

洪培衍 15336059

2015级计算机科学与技术

School of Data and Computer Science, Sun Yat-Sen University, China

laiqd@mail2.sysu.edu.cn

December 26, 2017

Abstract

这次实验我们采用A*算法解决八数码问题，使用康托展开方法来哈希数码放置状态，使用红黑树存储Open表和Close表。我们高效地找到八数码问题的最优解。

1 Introduction

1.1 Problem Description

在一个 3×3 的方棋盘上放置着1, 2, 3, 4, 5, 6, 7, 8八个数码, 每个数码占一格, 且有一个空格。这些数码可以在棋盘上移动, 其移动规则是: 与空格相邻的数码方格可以移入空格。对于指定的初始棋局和目标棋局, 给出最短的数码移动序列。

1.2 Algorithm Review

A*算法是一种启发式算法[1]。A*算法是A算法中的一类(满足某个约束)。启发式函数的形式定义为

$$f(S) = g(S) + h(S) \quad (1)$$

其中 $g(S)$ 表示从初始状态到状态 S 搜索费用的估计, $h(S)$ 表示从状态 S 到终止状态的估计。在或图搜索中的Open表中节点按照其启发式函数 f 大小排序, 并且每次取出节点取出具有最小/最大 f 的节点搜索算法叫A算法。

进一步定义

$$f^*(S) = g^*(S) + h^*(S) \quad (2)$$

$f^*(S)$ 表示为初始状态经过状态 S 到最终状态的最优路径搜索费用, $g^*(S)$ 为初始状态到 n 的实际最小费用, $h^*(S)$ 为 S 到最终状态的实际最小费用估计。令满足

$$h(S) \leq h^*(S) \quad (3)$$

的A算法为A*算法。

2 Algorithm Design

2.1 状态的表示

若把 3×3 那个空格看做放了0，把三行连成一行，数码的放置状态等同于0到8的一个全排列。所以有效状态数为 $9! = 362880$ 。于是可以用一个 $[0, 362880)$ 的整数对应一个状态(哈希)。

康拓展开是一种特殊的哈希，可以表示一个排列在 n 个元素的所有全排列中的字典序名次。对于 n 个数的排列排名计算公式为：

$$rk_a = a_n * (n-1)! + a_{n-1} * (n-2)! + \dots + a_2 * 1! + a_1 * 0! (0 \leq a_i < i, 1 \leq i \leq n) \quad (4)$$

其中 a_i 表示第 i 个元素在当前没出现的元素中(序列中第 i 项及其后面的项)排在第几，从0开始计数。

使用康托展开就可以把状态与一个数字对应起来了。这样存放在 $Open$ 表里就可以只是一个数字，大大减少存储量，也加快程序运行的速度。

可以看出哈希一个状态和一个哈希值转为一个状态时间复杂度为 $O(n^2)$ ，其中 n 为数码个数。

2.2 启发式函数

$g(S)$ = 初始状态到状态 S 的最小花费。 $h_1(S)$ = S 中错位数码个数，这个是书本上的启发式函数。 $h_2(S)$ = S 中每个数码当前位置与目标位置曼哈顿距离和。显然移动一次最多改变一个数码的位置，所以函数 h_2 是符合要求的。实验中分别使用了这两个函数，并进行对比。

2.3 数据结构

我们采用的是红黑树存储 $Open$ 表和 $Close$ 表。因为红黑树插入，删除，查询最小值时间复杂度都为 $O(\log_2 m)$ (其中 m 为红黑树节点个数， m 是 $O(n!)$ 级别的)，十分优秀。而且C++的set库中把红黑树封装好了，直接用它提供的接口即可，非常方便。最后老师要求提供中间搜索时两个表的内容，C++中的红黑树也可以使用迭代器读取红黑树里的节点信息。所以我们采用了红黑树这种数据结构的存储。

2.4 时空复杂度

令 n 为数码个数，空间复杂度为状态个数 $O(n!)$ ，时间复杂度为 $O(k(n! \log_2 n! + n^2 n!))$ ，由于 $O(\log_2 n!) = O(n \log_2 n)$ ，所以可以化简为 $O(kn^2 n!)$ 。其中 k 为每个状态平均扩展个数。

有相当一部分状态是无效状态，加上 n 只有9，所以这个时间复杂度是优秀的。给定任何初始状态都可以在1秒内得出结果。

2.5 算法框架

令 $father_i$ 为排名为 i 的状态的前驱状态排名(即从哪个状态扩展而来的)。可得如下算法框架。

Algorithm 1 A*

```
Close  $\leftarrow$  empty
Open  $\leftarrow$  初始状态
repeat
   $u \leftarrow$  Open里估价最小的节点(状态)
  Open  $\leftarrow$  Open  $- u$ 
  Close  $\leftarrow$  Close  $\cup u$ 
  now  $\leftarrow$   $u$ 的康拓展开
  repeat
     $nxt \leftarrow$  now的一个交换
     $v \leftarrow$   $nxt$ 的哈希值
     $vf \leftarrow$   $nxt$ 的估价
    if  $v$ 在Open里且估价比之前低 then
       $f_v \leftarrow vf$ 
       $father[v] \leftarrow u$ 
    else if  $v$ 不在Close也不在Open里 then
       $f_v \leftarrow vf$ 
       $father[v] \leftarrow u$ 
      Open  $\leftarrow$  Open  $\cup v$ 
    end if
  until 对now进行完搜索
until Open为空
```

3 Computational Experiment



Figure 1: 运行截图

运行结果上图3。左边的框是随机产生的八数码初始状态，中间的两个框显示的是使用 $h_1(S)$ 作为估价函数的结果，右边的两个框显示的是 $h_2(S)$ 作为估价函数的结果。上面的框有搜索时中间信息(*Open*表和*Close*表等)，下面的框是最优解的路径信息。

可以看出使用 $h_1(S)$ 生成了3555个节点，使用 $h_2(S)$ 生成了600个节点。此需要生成的节点远少于使用 $h_1(S)$ 的。可以看出 $h_2(S)$ 更具有信息性。

另外两个估价函数下A*算法都很快跑出了最优解(都不到0.03秒)。我们随机生成过很多数据，包括最终无解的情况。两个程序都在1秒内跑出来了。这体现了A*算法的优越性，启发式算法比盲目搜索求解速度快。

4 Conclusion

这次实验我们学会了A*算法的思想和基本框架，了解了A*算法的三个性质：可采纳性，单调性，信息性。同时我们学习了康拓展开的知识，通过小组讨论后使用较高效率的红黑树数据结构对A*搜索进行优化。最终我们得出了渐进复杂度很优的算法，比较好解决了八数码问题。对比传统的广度优先搜索，A*算法比其智能。但是在八数码问题上还没有体现出很明显的优势(随机生成的数据都在1秒内跑完了)。希望在以后的应用中能看到。

5 实验分工

李振宇完成A*算法的核心部分，并在MFC完成初版程序。

黎丁嘉修改了界面，完善了程序输出内容。

赖启东优化程序结构，提高程序运行效率。

刘键涵构想了两种不同的 $h(S)$ 函数。

洪培銮调试程序并debug以完成最终程序。

实验报告由赖启东和李振宇撰写，平时实验过程由李振宇和黎丁嘉完成。

References

- [1] 朱福喜, “人工智能基础教程(第二版),” 2011.