

中山大学
数据科学与计算机学院
软件工程
人工智能

A* 算法

Submitted By:

徐伟元 16340261

熊永琦 16340258

李天译 16340122

Submitted To:

王甲海

教授

大数据与计算智能研究所

2019-1-12



Abstract

本实验利用 A* 算法分别使用两种启发函数实现了八数码问题的求解，并提供了动态界面展示整个搜索过程，还进行了与九数码问题的有关讨论。

1 实验题目

以八数码问题为对象，编程解决：

1. 利用 A* 算法求解八数码问题，在输出界面上动态显示 OPEN 表的结点数和评估函数最小的结点
2. 比较两种启发函数（上课和书上讲到的 $h1(n)$ 和 $h2(n)$ ）的搜索效率，在输出界面上动态显示 OPEN 表的结点数、总扩展的结点数和评估函数最小的结点
3. 输出 OPEN 表中在最佳路径上的结点及其评估函数值
4. 验证凡 A* 算法挑选出来求后继的点 n 必定满足: $f(n) \leq f^*(S0)$
5. 验证 $h1(n)$ 的单调性，显示凡 A* 算法挑选出来求后继的点 n_i 扩展的一个子结点 n_j ，检查是否满足: $h(n_i) \leq 1 + h(n_j)$
6. 如果将空格看作 0，即九数码问题，利用相似的启发函数 $h1(n)$ 和 $h2(n)$ ，求解相同的问题的搜索图是否相同？
7. 写出能否达到目标状态的判断方法

2 A* 算法

2.1 A* 算法原理

A* 算法是从 A 算法约束而来，而 A 算法又是由通用图搜索算法约束得到的。我们知道，A 搜索是定义启发式函数 $f(n) = g(n) + h(n)$ 并在或图

通用搜索的取节点步骤中按函数值的大小取出一个节点的特殊通用图搜索算法。

在启发式函数中, $g(n)$ 表示从初始状态到 n 点的搜索费用估计, 可以根据搜索路径算出准确值, 而 $h(n)$ 表示从 n 到目标状态的接近程度的估计, 由于此时未找到解的路径, 只能粗略求出估计值。

若规定 $h(n) > 0$, 并且定义 $f^*(n) = g^*(n) + h^*(n)$, 其中 $f^*(n)$, $g^*(n)$ 分别为前面的 $f(n)$, $g(n)$ 的实际最小值。A* 算法就是要求其中 $h(n) \leq h^*(n)$ 的 A 算法。

2.2 两种启发函数

按照实验要求, 本实验分别采用两种启发函数 $h1(n)$ 和 $h2(n)$

$h1(n)$: 放错位置的数字个数

$h2(n)$: 所有数字当前位置到正确位置的曼哈顿距离和

2.3 八数码问题

算法能够给出正确的最佳路径解, 首先运行会产生随机打乱的初始状态, 并使用 $h1$ 进行启发式搜索, 显示结果后继续运行, 会使用与之前相同的初始状态并用 $h2$ 进行启发式搜索。某次运行的结果如 Figure 1 所示。

在代码中, 每一轮生成后继节点后加入对于 $f(n) \leq f^*(S_0)$ 是否成立的判断, 可以验证八数码问题满足该不等式, 因此八数码问题使用的是 A* 算法; 在生成子结点后加入是否满足 $h(n_i) \leq 1 + h(n_j)$ 的判断, 则可以验证 $h(n)$ 函数满足单调性。

2.4 九数码问题

与八数码问题的代码基本相同, 区别在于对两个启发式函数均需要加入对九数码问题中的 0 错放的计算。

Step i: 7	Step i: 6
Open: 7 nodes	Open: 6 nodes
Best path till now:	Best path till now:
1 2 3	1 2 3
8 9 4	8 9 4
7 6 5 $h(n)=0$	7 6 5 $h(n)=0$
1 2 3	1 2 3
9 8 4	9 8 4
7 6 5 $h(n)=1$	7 6 5 $h(n)=1$
9 2 3	9 2 3
1 8 4	1 8 4
7 6 5 $h(n)=2$	7 6 5 $h(n)=2$
2 9 3	2 9 3
1 8 4	1 8 4
7 6 5 $h(n)=3$	7 6 5 $h(n)=3$
2 8 3	2 8 3
1 9 4	1 9 4
7 6 5 $h(n)=3$	7 6 5 $h(n)=4$
2 8 3	2 8 3
1 6 4	1 6 4
7 9 5 $h(n)=4$	7 9 5 $h(n)=5$
2 8 3	2 8 3
1 6 4	1 6 4
7 5 9 $h(n)=5$	7 5 9 $h(n)=6$

Figure 1: h_1 和 h_2 的搜索路径结果.

但是九数码问题使用相似的启发函数时，虽然得到正确解的路径相同，求解相同问题的搜索图却不相同。如 Figure 2 所示，给定相同初始状态均使用 $h1$ 的搜索路径一致，但由于评估值不同，导致九数码问题需要扩展更多的结点。

2.5 其他讨论

2.5.1 随机产生初始状态的方法

由于 $h1(n)$ 函数的启发性较差，对于一些随机生成的八数码情况，可能需要搜索上千甚至上万步才能得到最终解，这不利于我们对于实验结果的观察，因此在打乱时我们采用空格随机向四个方向之一移动有限步的方法生成初始状态。这样做的好处一方面由于相当于是从目标状态逆推，生成的初始状态一定是可达目标状态的，另一方面避免了过多步收敛的问题（测试中通常十几步，最多几百步就能够得到最优解）。

2.5.2 n 数码问题可解的条件

经查阅相关资料得知，此类问题能够到达目标状态的条件是：初始状态与目标状态的逆序对数同奇偶。[1] 由于 A^* 算法具有可采纳性，如果问题有解，则 A^* 算法一定能找到最优解。

3 测试及结论

八数码问题采用 $h1(n)$ 时的 OPEN 表中结点数和搜索到最优解所需步数一般都较采用 $h2(n)$ 时的多，在问题变得更加复杂（例如那些使用 $h1(n)$ 需要搜索上万次的情况）时，使用 $h2(n)$ 的优势尤其明显。可以说明 $h2(n)$ 是相比 $h1(n)$ 效率更好的启发函数。这可能是由于 $h2(n)$ 采用了曼哈顿距离计算，可以更好衡量当前状态到目标状态的实际最小费用估计。

Step i: 8	Step i: 9
Open: 8 nodes	Open: 10 nodes
Best path till now:	Best path till now:
1 2 3	1 2 3
8 9 4	8 9 4
7 6 5 $h(n)=0$	7 6 5 $h(n)=0$
1 2 3	1 2 3
9 8 4	9 8 4
7 6 5 $h(n)=1$	7 6 5 $h(n)=2$
9 2 3	9 2 3
1 8 4	1 8 4
7 6 5 $h(n)=2$	7 6 5 $h(n)=3$
2 9 3	2 9 3
1 8 4	1 8 4
7 6 5 $h(n)=3$	7 6 5 $h(n)=4$
2 8 3	2 8 3
1 9 4	1 9 4
7 6 5 $h(n)=3$	7 6 5 $h(n)=3$
2 8 3	2 8 3
1 6 4	1 6 4
7 9 5 $h(n)=4$	7 9 5 $h(n)=5$
2 8 3	2 8 3
1 6 4	1 6 4
9 7 5 $h(n)=5$	9 7 5 $h(n)=6$

Figure 2: 均使用 h_1 时八数码和九数码问题的搜索路径结果.

References

- [1] <https://blog.csdn.net/u012160954/article/details/50557174>