

《人工智能概论》大作业 1 指导书

基于搜索策略的八数码问题求解

大作业题目：

基于搜索策略的八数码问题求解

大作业目的：

加深对搜索策略的理解，尤其是对启发式搜索的基本原理的理解，使学生能够通过编程实现图搜索的基本方法和启发式搜索算法，并能够解决一些应用问题。

大作业要求：

使用盲目搜索中的宽度优先搜索算法或者使用启发式搜索中的全局择优搜索或 **A*** 算法。

每人提交一份大作业报告，该报告包括设计、实现、测试、实验对比结果分析、结论、个人体会与总结，具体见格式要求。**大作业程序需验收通过**。

大作业提交内容：课程大作业纸质报告 1 份；将课程大作业电子报告和课程大作业程序及代码压缩包提交至网络教学综合平台。

大作业提交截止时间：以网络教学综合平台中的截止时间为准。

对任意的八数码问题，给出求解结果。例如：对于如下具体八数码问题：



通过设计启发函数，编程实现求解过程，如果问题有解，给出数码移动过程，否则，报告问题无解。

250	123
873	804
641	765

求解步骤：

步骤一.设计八数码格局的隐式存储的节点结构：

将表示棋局的状态用如下向量表示:

$$A=(X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$

约束条件: $X_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$

$X_i \neq X_j$, 当 $i \neq j$ 时。

初始状态: $S_0 = (1, 3, 2, 4, 0, 5, 6, 7, 8)$

目标状态: $S_g = (1, 2, 3, 8, 0, 4, 7, 6, 5)$

步骤二.计算两个节点之间的可达性。

(1) 可以通过限定时间阈值或步骤阈值, 判定两个节点之间的可达性。

(2) 通过计算八数码节点的逆序数判断。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**。一个排列中逆序的总数就称为这个排列的**逆序数**。逆序数为偶数的排列称为**偶排列**; 逆序数为奇数的排列称为**奇排列**。如 2431 中, 21, 43, 41, 31 是逆序, 逆序数是 4, 为偶排列。计算八数码节点的逆序数时将代表空格的 0 去除, 如初始状态排列为

(1, 3, 2, 4, 5, 6, 7, 8) 逆序数为: $0+1+0+0+0+0+0=1$ 即为奇排列

目标状态排列为 (1, 2, 3, 8, 4, 7, 6, 5) 逆序数为: $0+0+0+4+0+2+1+0=7$ 即为奇排列, 具有**同奇或同偶**排列的八数码才能移动可达, 否则不可达。

步骤三.设计估计函数与启发函数, 估计函数 $f(n)$ 定义为: $f(n)=d(n)+h(n)$ 其中, $d(n)$ 表示节点深度。

启发函数 $h(n)$ 可参考如下定义方法:

(1) 启发函数 $h(n)$ 定义为当前节点与目标节点**差异**的度量: 即当前节点与目标节点格局相比, 位置不符的**数字个数**。

(2) 启发函数 $h(n)$ 定义为当前节点与目标节点**距离**的度量: 当前节点与目标节点格局相比, 位置不符的数字移动到目标节点中对应位置的**最短距离之和**。

启发函数可参考如下定义方法:

(3) 启发函数 $h(n)$ 定义为每一对逆序数字乘以一个倍数。

(4) 为克服了仅计算数字逆序数字数目策略的局限, 启发函数 $h(n)$ 定义为位置不符数字个数的总和与 3 倍数字逆序数目相加。

步骤四 .选择并设计搜索算法(至少选择 1 个)

(1) 使用盲目搜索中的宽度优先搜索算法。

(2) 使用启发式搜索中的全局择优搜索算法。

(3) 使用 A* 算法。

步骤五 设计输入输出

输入: 初始节点, 目标节点 格式: 132405678

123804765

可采取以下三种方式输入：命令行、文件 `start.txt`、GUI 输入。

输出：如果无解在屏幕输出“目标状态不可达”

如果有解请在屏幕输出“最少移动 n 步到达目标状态”， n 为最少移动的步数，并记录从初始状态到目标状态的每一步中间状态，并将这些状态保存至 `result.txt` 文件中。

以上过程可采取以下三种方式输出：命令行、文件 `result.txt`、GUI 输出。

步骤六 编写代码，调试程序。

至少给出 5 组初始节点和目标节点对，并记录程序运算结果。节点对要包括以下三种情况：不可达；最少移动步数 ≥ 10 ；最少移动步数 ≤ 6

拓展实验：（可任选 1 个完成）

1. 记录步骤四中 3 个搜索算法的执行时间，并比较 3 者的效率。
2. 在启发式搜索中，分别采用步骤三中启发式函数(1)(2)(3)(4)，并比较四者的效率，思考如何进一步改进启发式函数。
3. 试分析在所有可能的初始状态和目标状态之间最长的最小移动步数是多少？