

Ch3

简介

1. 启发函数

$h(s)$ 是一个启发函数，用于粗略估算从初始状态到目标状态的代价。 $h^*(s)$ 是真实的代价。 $c(s, s')$ 是从状态 s 到状态 s' 的真实代价。

- 可计算性 (Computable)：启发函数必须是容易计算的，不能比直接求解问题本身更复杂
- **可接受性 (Admissible)**：启发函数的值必须是合法的，不能比直接求解问题本身更糟，即：

$$h(s) \leq h^*(s)$$

- 一致性 (Consistent)：保证已扩展节点不会被重新扩展，即不成环，类似于三角形定理：

$$h(s) \leq h(s') + c(s, s')$$

我们关注第二条，**可接受性**。

设计启发函数的技巧是：**松弛原问题的约束或条件**。例如曼哈顿距离或欧氏距离，用在寻路中，是忽略了不能穿过障碍物这个约束得到的启发函数。

常用启发函数就是各种距离：

- L1距离：**曼哈顿距离**，即四方向距离

$$h(s) = |x_1 - x_2| + |y_1 - y_2|$$

- L2距离：**欧氏距离**

$$h(s) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- L^∞ 距离：国王距离，也叫切比雪夫距离，八方向距离

$$h(s) = \max(|x_1 - x_2|, |y_1 - y_2|)$$

2. 评估函数

$f(s) = g(s) + h(s)$ 是评估函数，用于**评估从初始状态 s_0 经过中间状态 s ，到达目标状态的代价**， $g(s)$ 是从初始状态 s_0 到状态 s 的实际代价， $h(s)$ 是从状态 s 到目标状态的启发函数。

起点的评估就是 $h(s_0)$ ，终点的评估就是0。

3. N puzzle的不相交数据集

不相交数据集（Disjoint Set）是一种用于处理集合划分的数据结构，也叫**并查集**（Union Find）。这是一棵孩子记录父亲的多差树，支持Union和Find两个操作。

用一个数组 $father$ 表示，**下标 i 记录的 $father[i]$ 是它父节点的下标**，如果 $father[i] = i$ 则表示它没有父亲节点。根节点的父节点也是自己。

- Union操作：将 $node1$ 的父节点设置为 $node2$ 的父节点，逻辑上相当于把一棵树接在另一棵树的叶子上，**合成一颗大树**。
- Find操作：返回一个节点的**根节点**，逻辑上相当于返回某棵树的根节点。

GBFS

贪婪最优搜索算法**只考虑启发函数 $h(s)$ 的值而不考虑 $g(s)$** 。可以理解为 $f(s) = h(s)$ 。采用优先队列，具体实现类似于BFS。

既**不完备**（由于只考虑启发值，可能陷入死胡同）也**不最优**（是局部最优的而不是全局最优的）。

A Search

A算法是在GBFS的基础上，加入了 $g(s)$ 的信息，即 $f(s) = g(s) + h(s)$ 。采用优先队列。但这里A算法的启发函数 $h(s)$ 并不要求满足可接受性等性质，因此不保证其最优性。

由于考虑了 $g(s)$ ，因此A算法是完备的。但它**不一定是最优的**。

A* Search

A*算法在A的基础上要求启发函数 $h(s)$ **满足可计算性、可接受性、一致性等性质**，因此A*算法是最优的。

完备且最优。

IDA* Search

IDA*算法是A*算法的变种，与ID-DFS类似，采用**限制范围、逐级扩展**的思维。

IDA*算法的流程是**设置一个最大 f 值**（即评估函数 $f(s)$ 的最大值），运行A*算法，超出最大值的路径**停止搜索**（记录它的值 f_{curr} ，维护一个最小的 f_{min} ），所有路径都停止后若没有找到解则将 f 更新为刚刚记录的最小那个的 f_{min} ，再次运行A*算法，**重复流程**。

完备且最优，并且减小搜索范围。