

# 人工智能作业2

---

1.

o a)

变量是光源放置的空间, 即  $X = \{1, 2, 3, 4, 5\}$

值域是光源类型, 即  $D = \{A, B, C, D\}$

o b)

一元约束:

$C = \{C_1, C_2, C_3, C_4, C_5\}$ , 其中

$C_1 = ((t_1), t_1 = A, B, C, D)$

$C_2 = ((t_2), t_2 = B, C, D)$

$C_3 = ((t_3), t_3 = C, D)$

$C_4 = ((t_4), t_4 = C, D)$

$C_5 = ((t_5), t_5 = A, B, C, D)$

二元约束:

$C = \{C_1, C_2, C_3, C_4\}$ , 其中

$C_1 = ((t_1, t_2), t_1, t_2 \text{中至少有一个是} A)$

$C_2 = ((t_2, t_3), t_2, t_3 \text{中至少有一个是} B)$

$C_3 = ((t_3, t_4), t_3, t_4 \text{中至少有一个是} C)$

$C_4 = ((t_4, t_5), t_4, t_5 \text{中至少有一个是} A)$

o c)

$D_1 = \{A\}$

$D_2 = \{B\}$

$D_3 = \{C, D\}$

$D_4 = \{C, D\}$

$D_5 = \{A\}$

o d)

■  $(t_1, t_2, t_3, t_4, t_5) = (A, B, C, C, A)$

■  $(t_1, t_2, t_3, t_4, t_5) = (A, B, C, D, A)$

■  $(t_1, t_2, t_3, t_4, t_5) = (A, B, D, C, A)$

2.

o a)

若不考虑一个格子至多一架飞行器的约束, 那么每一架飞机有 $n^2$ 种选择, 那么对于 $n$ 架飞机, 一共会产生 $(n^2)^n$ 种状态, 所以状态空间大小是 $n^{2n}$

o b)

计算分支因子时应考虑分支最多的情况，最多的情况即是所有的飞机都有5种选择（上下左右移动或者不动），一共有 $n$ 架飞机，所以分支因子大小是 $5^n$

o c)

若空域中没有其他飞行器，启发函数可以选择用曼哈顿距离，因为曼哈顿距离一定比到目的地的真实代价要少。

$$h_i = |n - i + 1 - x_i| + |n - y_i|$$

o d)

选择3号- $\min\{h_1, \dots, h_n\}$ 作为启发函数。首先我们可以设所有飞机飞到指定地点的总代价为 $S$ （每一个飞机动作一次代价都会加1），那么一定有

$$S \geq \sum_{i=1}^n h_i \geq n * \min\{h_1, \dots, h_n\}$$

又因为每一个回合对于 $n$ 架飞机行动的步数是小于等于 $n$ 的（若一架飞行器因为跳跃走了2步，但前提是有一架飞行器不动，所以与每一架飞机各移动一次代价一样）。

因此要让所有飞行器都到目的地，需要的最少次数是

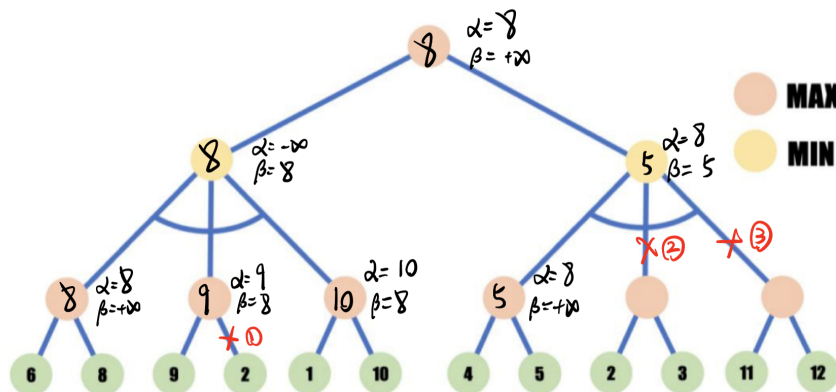
$$n * \min\{h_1, h_2, \dots, h_n\} / n = \min\{h_1, h_2, \dots, h_n\}$$

这样才能使得启发函数小于真实路径代价，才是可采纳的。

如果采用 $\max\{h_1, \dots, h_n\}$ 作为启发函数，因为本题中可能会有越过一架飞机的情况，相当于走了曼哈顿距离的两步，导致启发函数估计的代价值大于了真实路径代价，显然是不可采纳的。

综上所述，应选择 $\min\{h_1, \dots, h_n\}$ 作为启发函数

3.

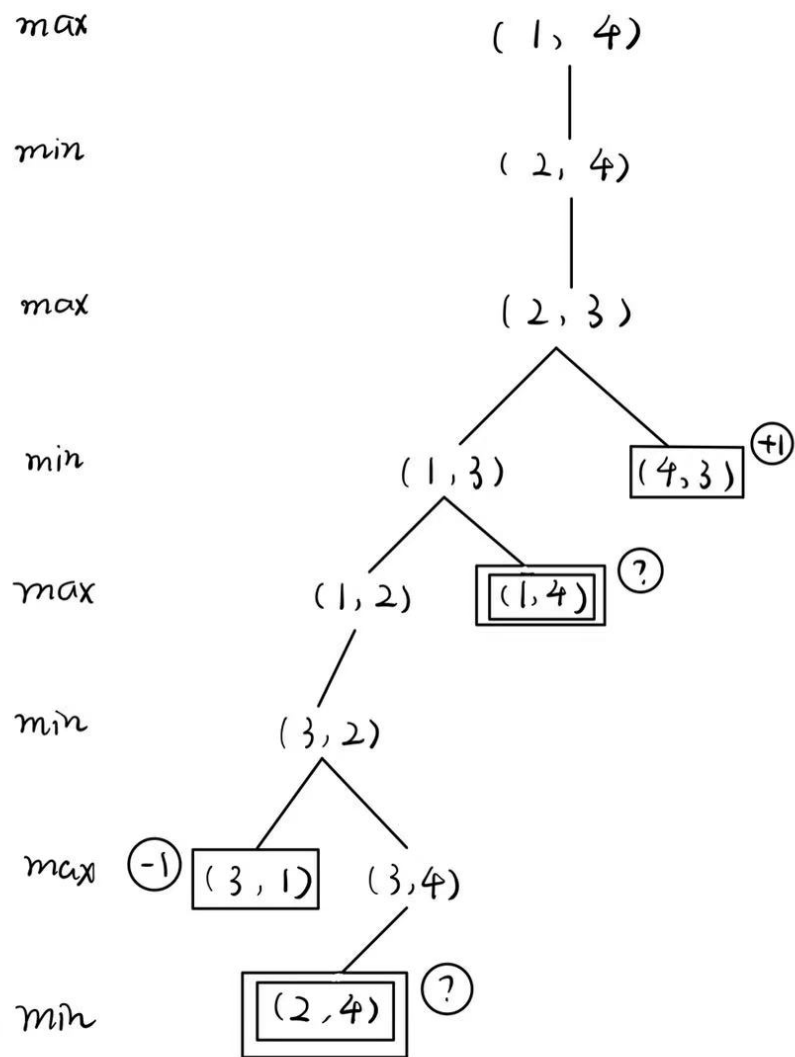


如图所示，标号为1, 2, 3的边分别是先后被剪的边

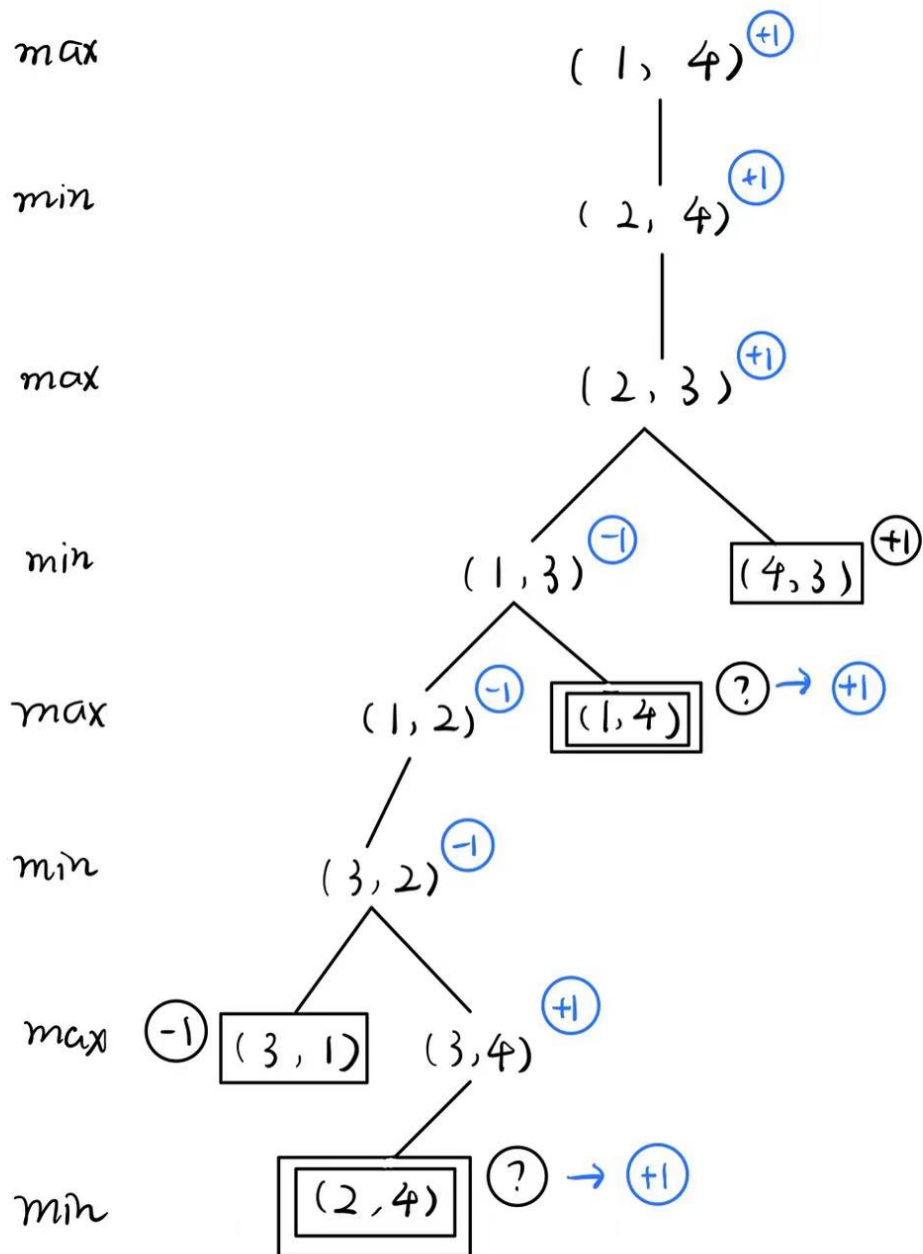
- o  $\alpha - \beta$ 剪枝实际上是对极大极小值算法的优化，旨在减少其搜索树中被极大极小算法评估的节点数，它的基本思想是根据上一层已经得到的最优结果，也即是阈值 $\alpha$ 和 $\beta$ 的作用，来判断我还需不需要考虑某节点的后继节点，例如本例中被剪掉的2、3号枝，是在发现最左边的节点效用值为5之后，更新其父节点 $\beta$ 值为5(因为是MIN节点)，这时候 $\alpha = 8 > \beta = 5$ ，也即说明之后的节点我不需要关心了，因为不会影响我上上一层节点的选择，这样就可以剪掉之后的枝。
- o 可以看到，在极大节点的 $\alpha$ 剪枝和极小节点的 $\beta$ 剪枝的配合之下，能够减少很多不必要的搜索，从而减少分支因子，在空间上和时间上效率得到提升。

4.

o a)



o b)



- 对问号的处理：问号的出现肯定是出现了之前出现过的节点，而问号其实可以通过之前的状态得到，例如(2, 4)在之前就出现过，其效用值为+1，那么就可将再一次出现的(2, 4)的效用值也改为+1，从而可以更新其上层节点。
- 原因：首先这样做遵循了极大极小值搜索的规则，保证其完备性。
- c) 因为标准的极大极小值算法是去寻找一条效用值最大的路径，很显然对于一个包含循环的游戏，按照标准的极大极小值算法寻找的路径效用值会不断攀升。