

第三章

Exercise 17

1. Depth-first search always expands at least as many nodes as A* search with an admissible heuristic.
错。DFS 可能沿最短路径直接找到目标结点，仅需扩展 d （目标结点深度）个结点；而带有可采纳的启发式的 A* 算法可能因启发式引导选择探索周边结点以保证找到最优解。所以 DFS 扩展的结点数可能少于 A* 算法。
2. $h(n)=0$ is an admissible heuristic for the 8-puzzle.
对。因为代价总是非负的，所以结点 n 的实际代价一定大于等于 $h(n)=0$ ，所以 $h(n)$ 一定是可采纳的。是最弱的启发式，相当于无启发。
3. A is of no use in robotics because percepts, states, and actions are continuous.
错。虽然机器人感知和状态空间通常是连续的，但可通过离散化操作将问题转化为 A* 适用的离散状态空间，所以 A* 算法在机器人学中仍然有重要作用。
4. Breadth-first search is complete even if zero step costs are allowed.
对。BFS 的完备性与其遍历策略和状态空间的性质相关，与边成本无关。若存在解，即使允许零成本边，BFS 仍能保证在有限分支因子的状态空间中找到解。
(完备性：若有解，一定能找到解)
5. Assume that a rook can move on a chessboard any number of squares in a straight line, vertically or horizontally, but cannot jump over other pieces. Manhattan distance is an admissible heuristic for the problem of moving the rook from square A to square B in the smallest number of moves.
错。此问题中的代价是移动次数，移动一次最多可以消除一个方向上的任意差距，而曼哈顿距离刻画的是两跟方格的距离大小，二者无直接联系，甚至曼哈顿距离往往会高估计代价。例如，在以 A、B 为对角点的矩形边上无棋子阻挡时，曼哈顿距离一定大于等于 2，而实际代价始终为 2。综上，曼哈顿距离在此问题中是不可采纳的。

Exercise 25

Prove each of the following statements, or give a counterexample:

1. Breadth-first search is a special case of uniform-cost search.
 2. Depth-first search is a special case of best-first tree search.
 3. Uniform-cost search is a special case of A search.
1. **对。**当 UCS 中所有边的代价相等时，到每个结点的成本即等于该结点的深度，此时 UCS 的优先级队列会像 BFS 按层扩展的行为一致。
 2. **对。**Best-first tree 搜索按照评估函数 $f(n)$ 选择代价最小的结点进行扩展，若令 $f(n)=-\text{depth}(n)$ ，那么 Best-first tree 搜索会优先扩展深度最大的结点，此时即与 DFS 的行为一致。
 3. **对。**A* 搜索的评估函数为 $f(n)=g(n)+h(n)$ 。而当 $h(n)=0$ 时， $f(n)=g(n)$ ，此时 A 搜索完全按照路径成本扩展节点，与 UCS 的行为一致。

Exercise 38

The traveling salesperson problem (TSP) can be solved with the minimum-spanning-tree (MST)

heuristic, which estimates the cost of completing a tour, given that a partial tour has already been constructed. The MST cost of a set of cities is the smallest sum of the link costs of any tree that connects all the cities.

1. Show how this heuristic can be derived from a relaxed version of the TSP.
 2. Show that the MST heuristic dominates straight-line distance.
 3. Write a problem generator for instances of the TSP where cities are represented by random points in the unit square.
 4. Find an efficient algorithm in the literature for constructing the MST, and use it with A graph search to solve instances of the TSP.
1. TSP 问题是找到一条穿过城市的最小路径, 形成一个闭环。MST 在此问题中会生成一个连接所有城市的无环图 (即树), 它的总长度总是小于或等于闭环。故 MST 是可接受的。
 2. MST 给出的从当前结点到目标结点的距离必然是一条直线距离或多条直线距离的和, 在城市之间的距离满足三角不等式的情况下, MST 给出的距离必然大于两座城市的直线距离。

第四章

Exercise 4.1

Give the name of the algorithm that results from each of the following special cases:

1. Local beam search with $k=1$.
 2. Local beam search with one initial state and no limit on the number of states retained.
 3. Simulated annealing with $T=0$ at all times (and omitting the termination test).
 4. Simulated annealing with $T=\infty$ at all times.
 5. Genetic algorithm with population size $N=1$.
1. **爬山法**。因为只选取 1 个最优的后继节点。
 2. **BFS 广度优先搜索**。总是将每层的结点扩展后再扩展下一层结点。
 3. **爬山法**。温度 T 始终为 0, 选择更坏移动的概率始终为 0, 故总是选取更优移动。
 4. **随机游走**。 $T=\infty$ 时无条件接受所有移动, 相当于随机移动。
 5. **随机游走**。遗传算法个体数为 1, 则交叉无法产生新解, 仅能通过变异产生, 相当于随机游走。

第五章

5.2 d

这取决于初始状态的情况:

如果初始时两个 8-puzzle 都只差一步解决, 那么优先抛硬币的玩家获胜。

如果初始时只有一个 8-puzzle 差一步解决, 那么获胜者将是抛硬币最先抛中该 puzzle 的玩家。

+ 不完整
如果初始时两个 8-puzzle 都比较复杂 (需要大于一步解决) 时, 假设某一时刻玩家 1 抛中了一个只剩两步解决的 puzzle, 如果他选择靠近目标一步, 那么对手获胜的概率将是 $1/2 + 1/4 + 1/16 + \dots = 2/3$, 自己获胜概率为 $1/3$, 所以最佳行动应该为远离目标一步, 按这样考虑, 表现完美的玩家之间不可能分出胜负。

Exercise 5.7

Prove the following assertion:

For every game tree, the utility obtained by max using minimax decisions against a suboptimal min will never be lower than the utility obtained playing against an optimal min.

证明：考虑一个 MAX 节点，其有 n 个子 MIN 节点，这 n 个 MIN 节点的子节点均为叶节点。

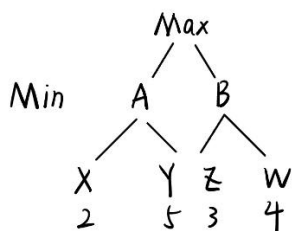
设第 i 个 MIN 节点的叶节点中最低效用为 C_i^* ，而第 i 个 MIN 节点选择了一个次低效用 C_i ，则有 $C_i > C_i^*$ 。

设 MAX 节点基于最优 MIN 节点最优策略的效用值为 $u_{\text{opt}} = \max(C_1^*, \dots, C_n^*)$ ，基于次优 MIN 节点次优策略的效用值为 $u_{\text{sub}} = \max(C_1, \dots, C_n)$ 。

由于 $C_i > C_i^*$ ，对 $i=1, \dots, n$ 均成立，所以必定有 $u_{\text{sub}} > u_{\text{opt}}$ 。得证。

Can you come up with a game tree in which max can do still better using a suboptimal strategy against a suboptimal min?

构造



如图，规定 MAX 和 MIN 需要同时做出行动，在双方都行动之前不知道对方会如何行动。

由于 MIN 层采取次优策略，A 节点会选择 $X=5$ ，B 节点会选择 $Z=4$ 。

若 MAX 采取最优策略，MAX 节点认为 MIN 采取最优策略，即 A 选择 $X=2$ ，B 选择 $Z=3$ ，那么 MAX 会选择 B。然而实际上 MAX 的选择得到的效用为 4。

若 MAX 采取次优策略，在同上猜测的情况下，MAX 会选择 A。实际上 MAX 的选择得到的效用为 5。可见 MIN 采取次优策略时，MAX 采取次优策略的收益可能高于采取最优策略的收益。

Exercise 5.13

Describe how the minimax and alpha-beta algorithms change for two-player, non-zero-sum games in which each player has a distinct utility function and both utility functions are known to both players.

If there are no constraints on the two terminal utilities, is it possible for any node to be pruned by alpha-beta?

What if the player's utility functions on any state sum to a number between constants $-k$ and k , making the game almost zero-sum?

1. 双人非零和博弈、独立效用函数的算法调整：

设两个玩家的效用函数分别为 $U_1(s)$ 和 $U_2(s)$

(1) 极小化极大算法的调整

节点数据：每个节点需要同时记录两个玩家的效用值 (U_1, U_2)

节点行为：MAX 节点 (玩家 1)：选择最大化 $U_1(s)$ 的行动

MIN 节点 (玩家 2)：选择最大化 $U_2(s)$ 的行动

策略调整：两玩家不再是单纯的对抗关系，可能还有合作关系，需考虑对方的效用函数对自

身决策的影响

(2) α - β 剪枝算法的调整

剪枝条件: **Max 节点** (玩家 1): 仅当某个子节点的 $U1$ 值高于当前已知的最高 $U1$ 值时, 保留该分支; 否则剪枝。

Min 节点 (玩家 2): 仅当某个子节点的 $U2$ 值低于当前已知的最高 $U2$ 值时, 保留该分支; 否则剪枝。

2. 无约束终端效用时的剪枝可能性:

如果两个玩家的终端效用函数无任何约束 (即 $U1$ 和 $U2$ 完全独立), 传统的 α - β 剪枝可能无法生效, 因为无法通过单一效用值的上下界直接推断其他玩家的选择。

3. 近似零和博弈约束下的剪枝可能性

改变约束条件: $U1 \in [-k - U2, k - U2]$

剪枝规则调整: **Max 节点** (玩家 1): 更新 α = 当前已知的 $U1$ 下限, 通过 $U1 \geq \alpha$ 剪枝。

Min 节点 (玩家 2): 更新 β = 当前已知的 $U1$ 上限, 通过 $U1 \leq \beta$ 剪枝。

约束受限问题的形式化描述

a. 变量集 $X = \{X_1, X_2, \dots, X_n\}$, 其中每个 X_i 是一个三元组 (t_i, r_i, p_i) , 分别表示该课程安排的时间、教室、教授

域集 $D = \{D_1, D_2, \dots, D_n\}$, 则每个 $D_i = T \times R \times P_i$, T 为可选时间集合、 R 为可选教室集合、

P_i 为能教授课程 i 的教授集合

约束集 C 包含的约束有:

(1) 教授时间冲突: 对于任意的 $i \neq j$, 若 $p_i = p_j$, 则 $t_i \neq t_j$

(2) 教室时间冲突: 对于任意的 $i \neq j$, 若 $r_i = r_j$, 则 $t_i \neq t_j$

(3) 教授能力匹配: 对所有 $i = 1, \dots, n$, 需要 $p_i \in P_i$

b. 变量集 $R = \{R_1, R_2, \dots, R_n\}$, 其中每个 R_i 都包含对应小矩形的左下角坐标 (x_i, y_i) 、宽度 w_i 和高度 h_i

域集 $D = \{D_1, D_2, \dots, D_n\}$, 设大矩形的宽度 W , 高度 H , 则每个 $D_i = [0, W - w_i] \times [0, H - h_i]$

约束集 C 包含的约束有:

所有小矩形必须在大矩形内: $x_i + w_i < W$ 且 $y_i + h_i < H$, 对任意 $i = 1, \dots, n$

任意两个小矩形不重叠: 对于任意的 R_i 和 R_j ($i \neq j$) 满足

$x_i \geq x_j + w_j$ (R_i 在 R_j 右侧) | $x_j \geq x_i + w_i$ (R_i 在 R_j 左侧)

| $y_i \geq y_j + h_j$ (R_i 在 R_j 上方) | $y_j \geq y_i + h_i$ (R_i 在 R_j 下方)

Exercise 6.3.#ACTA

Use the AC-3 algorithm to show that arc consistency can detect the inconsistency of the partial assignment $\{WA = green, V = red\}$ for the problem shown in Figure 6.1.

We'll trace through each iteration of the **while** loop in AC-3 (for one possible ordering of the arcs):

- Remove $SA - WA$, delete G from SA .
- Remove $SA - V$, delete R from SA , leaving only B .
- Remove $NT - WA$, delete G from NT .
- Remove $NT - SA$, delete B from NT , leaving only R .
- Remove $NSW - SA$, delete B from NSW .
- Remove $NSW - V$, delete R from NSW , leaving only G .
- Remove $Q - NT$, delete R from Q .
- Remove $Q - SA$, delete B from Q .
- remove $Q - NSW$, delete G from Q , leaving no domain for Q .

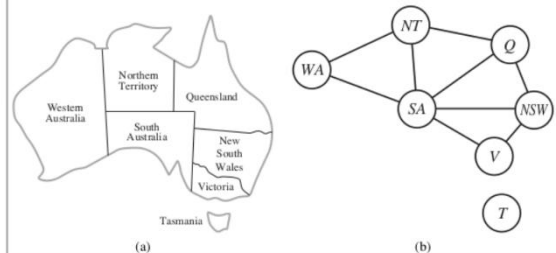


Figure 6.1 (a) The principal states and territories of Australia. Coloring this map can be viewed as a constraint satisfaction problem (CSP). The goal is to assign colors to each region so that no neighboring regions have the same color. (b) The map-coloring problem represented as a constraint graph.

