

搜索算法评价指标

完备性：若问题有解则必能找到（搜索过程记为A）

最优性：找到的第一个解是最优解（搜索过程记为A*）

时空复杂度

状态图

$\langle S, T, G \rangle$

S: 初始状态集合 T: 状态转换规则集合 G: 目标状态集合

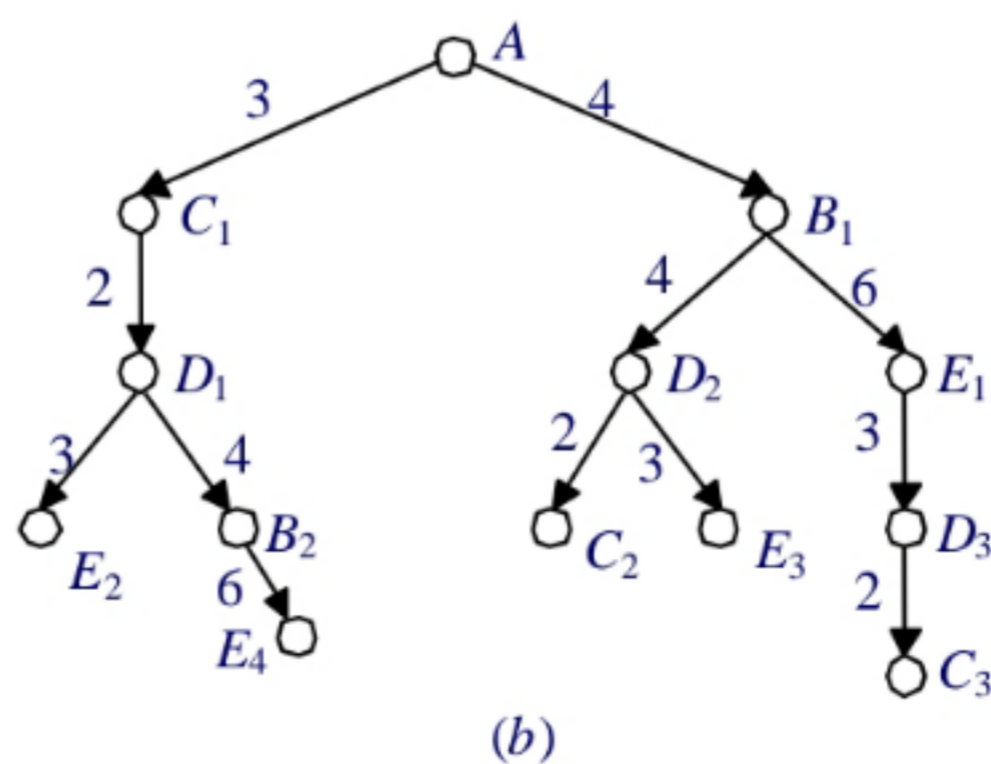
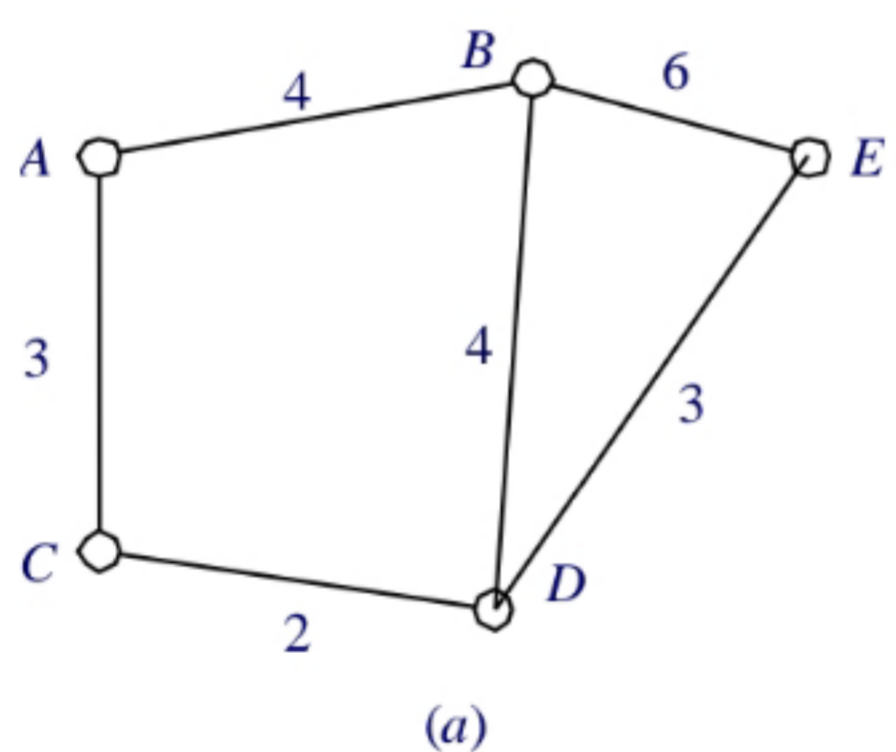
边上有数值的图称为加权状态图，其搜索树称为代价树

$g(x)$ 表示从初始节点 S_0 到 x 的代价 $C(x_i, x_j)$ 表示从父节点 x_i 到子节点 x_j 的代价

$$g(S_0) = 0$$

$$g(x_j) = g(x_i) + C(x_i, x_j)$$

可以把加权状态图当代价树来搜索：从初始节点起，将每个与 S_0 相邻的节点作为该节点子节点，其它节点依次类推，但对于其它节点 x ，不再将其父节点及祖先节点作为 x 子节点。



启发式搜索

评价函数 $f(n)$: 决定搜索优先级（下一步到哪个节点）

实际代价函数 $g(n)$: 从起点到当前节点的代价

启发函数 $h(n)$: 从当前节点到终点的预估代价

局部择优：扩展节点 n 后，仅对 n 的子节点按启发函数值大小以升序排序，再将它们放入 OPEN 表首部

称为贪心最佳优先搜索，约等于带启发函数的 DFS

$$f(n) = h(n)$$

时、空复杂度： $O(b^m)$ b 为搜索树分支因子
 m 为最大深度

全局择优：在 OPEN 表中保留所有已生成而未考察的节点，并用启发函数 $h(x)$ 对它们进行估价，从中选出最优节点进行扩展，而不管该节点出现在搜索树什么地方

称为最佳优先搜索，约等于带启发函数的 BFS

$$A^*: f(n) = g(n) + h(n)$$

可容性： $h(n) \leq h^*(n)$ ，即 $h(n)$ 不高估节点 n 到终点 T 的实际最小代价
保证 A^* 的最优性

一致性：对于任意节点 n 及其后继节点 n' ，有：

$$h(n) \leq C(n, n') + h(n')$$

其中 $C(n, n')$ 为 $n \rightarrow n'$ 的实际代价

保证 A^* 不走回头路

一致性必保证可容性

完备性：① 搜索树中分支数量有限

② 单步代价的下界是正数 (A^* 不能用在负权图中)

③ 启发函数有下界 ($h(n) \leq h^*(n)$ 对所有节点都成立)