搜索算法都可以看作在搜索树上的探索

一、uninformed

没有利用除“判断目标状态”外的其他信息

Tree-search versions

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes[1]	Yes[1,2]	No	No	Yes[1]	Yes[1,4]
Optimal cost?	Yes[3]	Yes	No	No	Yes[3]	Yes[3,4]
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$	$O(b^{d/2})$
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(b\ell)$	$O(bd)$	$O(b^{d/2})$

1、宽搜

加入时检测还是后续检测?

时间空间：O（b^d）

2、Uniform-cost search

- 先走cost小节点
- 记录从根节点的总cost
- 展开下一层时把探索过的中cost最大的×掉，即更小的要替换
- 在探索的时候对最优状态检测而不是产生的时候

> 有效防止遇到即停止

评价

- 反证法：

- Optimal

  when a node n is expanded, the optimal path to n.state has been found

- 时空复杂度都大

- Time complexity

  $O(b^{1+\lfloor C^*/\epsilon \rfloor})$, *where* C^* *is the cost of the optimal path*

- Space complexity

  $O(b^{1+\lfloor C^*/\epsilon \rfloor})$, *where* C^* *is the cost of the optimal path*

3、深搜

- 损失完备性和最优性
- 空间大幅降低

- Complete

  if $l \geq d$*, the depth of the shallowest goal node*

- Optimal

  if $l > d$*, no; if* $l = d$ *and all actions have the same cost, yes*

- Time complexity

  $O(b^l)$

- Space complexity

  $O(bl)$*, if using tree-search*

4、Iterative 深搜

- 最大搜索深度设定d，才能保证完备性和最优性，略微减小时间复杂度

increasing depth limit l Stop until $l = d$

- Complete

  if the depth d *of the shallowest goal node is finite*

- Optimal

  if all actions have the same cost

- Time complexity

  $db + (d-1)b^2 + (d-2)b^3 + \cdots + (1)b^d = O(b^d)$

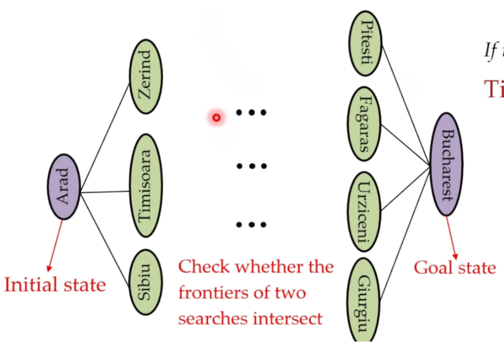
- Space complexity

  $O(bd)$*, if using tree-search*

5、Bidirecction 双向搜索

从初始和目标分别开始搜

- 很多情况不实用，不适合后向搜
- 牺牲空间，时间幂次减半



If using breadth-first search

Time complexity

$O(b^d)$

⇩

$O(b^{d/2})$

Not easy to use

二、Informed

1、Heuristic启发式函数：

> 比如用直线距离

- 估计离目标或最优还有多远
- 自变量依据的是状态而不是节点，因为不同节点可以状态相同
- 非负

admissible

> 指h(n)不会超过n到目标的实际花费h*(n)

A heuristic h is **admissible** if

$$0 \leq h(n) \leq h^*(n)$$

where $h^*(n)$ is the true cost of the optimal path from n to a goal

consistent

> n'是从n进行动作a'到达的后继，c为实际cost

A heuristic h is **consistent** if

$$h(n) \leq c(n, a', n') + h(n')$$

where n' is the successor of n generated by action a'

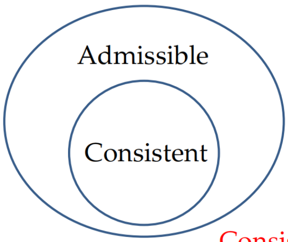
- 若consistent，则g+h在任意路径不递减

$$\begin{aligned} g(n') + h(n') &= g(n) + c(n, a', n') + h(n') \\ &\geq g(n) + h(n) \end{aligned}$$

Consistent

consistent<admissable

Consistent $\Rightarrow$ Admissible



- $h(n)$: estimated cost of the optimal path from the node n to a goal
- $h^*(n)$: true cost of the optimal path from the node n to a goal
- Assume that the nodes on the optimal path are $n, n', n'', \dots, n^*$

Consistent

$$h(n) \le c(n,a',n') + h(n') \le c(n,a',n') + c(n',a'',n'') + h(n'')$$
$$\le c(n,a',n') + c(n',a'',n'') + \cdots + h(n^*) = h^*(n) \quad \text{Admissible}$$

2、Greedy best-first贪心搜索

- 完备性：tree可能死循环（小范围走来走去，不肯走出舒适区），graph可以
- 最优性：无
- 最坏时空很大

- Time complexity
  $O(b^m)$, *where m is the maximum depth of any node, if using tree-search*
- Space complexity
  $O(b^m)$, *where m is the maximum depth of any node, if using tree-search*

3、A*

- 结合总cost和估计cost，即g+h

最优性

- tree有，graph无（探索时才判断，可以保证最优性）

> 因为graph不是探索时才判断

- tree要admissable才有最优性
- graph要consistent才有最优性（更严格）
- 探索时必最优，反证法

完备性

只要branching factor有限，且g+h不递减

复杂度

探索小于等于目标cost C*的所有节点，即得到时间最优

但空间还有技巧可以提升

4、Recursive best-first search

> 减小空间复杂度到和深度线性关系

- 每次探索，记录兄弟路径中次优cost值，等待可能的回退
- 当正选路径代价大于上一步候选路径时，回退到候选路径，正选变候选
- 可能需要回退到上两层以上的候选
- 规律：候选值非增

5、relax

TSP旅行商问题--->MST最小生成树

> 动作变成可以重复访问，只是重新访问cost为0
>
> 目标变成可以停留在任何城市，要求还是访问所有城市

- 不用管带不带环，因为带环一定可以优化cost

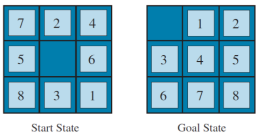
6、评判选择

- 比较generated nodes

Example: 8-puzzle

h_1 = the number of misplaced tiles

h_2 = the sum of the distances of the tiles from their goal positions



Start State Goal State

	Effective branching factor b^*		
d	Iterative deepening DFS	A*(h_1)	A*(h_2)
2	2.45	1.79	1.79
4	2.87	1.48	1.45
6	2.73	1.34	1.30
8	2.80	1.33	1.24
10	2.79	1.38	1.22
12	2.78	1.42	1.24
16	-	1.44	1.23

For each d, the average of 100 random instances

h_2 is better than h_1

Informed search is better