Uninformed Search:

DFS: 效率低 DLS: 可能找不到 IDS:目标很深效率低 BFS

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **Time Complexity** | **Space Complexity** | **Derivative** |
| **DFS** | O(bm) | O(bm) |  |
| **DLS** | O(bl) | O(bl) | DFS |
| **IDS** | O(bd) | O(bd) | DLS |
| **BFS** | O(bd) | O(bd) |  |
| **UCS** | O(bd) | O(bd) | BFS |
| **BIDI（Bidirectional）** | O(bd/2) | O(bd/2) | BFS（两端  采用BFS） |

b, branching factor

d, tree depth of the solution

m, tree depth

l, search depth limit

State Space (aka Problem Space) = all possible valid configurations of the environment

Optimality / Admissibility – an admissible algorithm will find a solution with minimum cost

Completeness – a complete algorithm will find a solution (not all)

A goal test is applied to a node's state to determine if it is a goal node

Informed Search:

f(n) = g(n) + h(n)

其中h(n)就是heuristic函数，g(n)是cost估算函数。g(n)表示从初始节点到当前节点所需的开销（例如可以用路径的长度/权来表示），而h(n)表示从当前节点到目标节点还剩的差距（有多种表示方法，比如可以用当前状态与目标状态Conflicts的个数来表示），g(n)和h(n)两者共同决定了f(n)，即一个状态的质量/估价。有时为了区分g(n)和h(n)的权重，也可以在前面加上常系数，即表示为

f(n)=α\*g(n) +β\*h(n)

BFS: （1）将初始节点插入OPEN List中

当OPEN List非空时

（2）从OPEN List出堆，放入CLOSED List中，如果该节点即为目标节点则算法结束（backtracking路径）

（3）扩展该节点的所有后继节点

对每一个后继节点，若后继节点不在CLOSED中，则把它插入OPEN List中

A\*:（1）将初始节点插入OPEN List中

当OPEN List非空时

（2）从OPEN List出堆，放入CLOSED List中，如果该节点即为目标节点则算法结束（backtracking路径）

（3）扩展该节点的所有后继节点

对每一个后继节点

if后继节点在CLOSED List中，则抛弃它，continue

else if后继节点在OPEN List中

if后继节点的g(n)值比出现在OPEN List上的那个节点的g(n)值好

则用后继节点替换掉原来出现在OPEN List上的那个节点

else

把后继节点加到OPEN List中

A heuristic function that utilizes relative distance from the goal state.

A heuristic, h, is called consistent (aka monotonic) if, for every node n and every successor n’ of n, the estimated cost of reaching the goal from n is no greater than the step cost of getting to n’ plus the estimated cost of reaching the goal from n’:

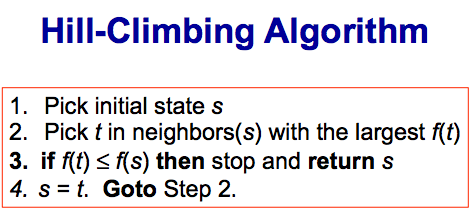
c(n, n’) ≥ h(n) − h(n’)

or, equivalently: h(n) ≤ c(n, n’) + h(n’)

•Values of f(n) along any path are nondecreasing

•Consistency is a stronger condition than admissibility

All consistent heuristics are admissible, but not all admissible heuristics are consistent

An admissible heuristic h(n) never overestimates the cost from n to the goal.



