人工智能导论第一次作业

2018080111 软件82 覃逸成

1. 第一题(10分) 证明以下论点或者举出反例:

1. 宽度优先搜索(BFS) 是一致代价搜索(UCS) 的一种特殊情况。

答：正确。宽度优先搜索为每层节点的代价函数（到root的cost）相同的UCS。

1. 宽度优先搜索(BFS) 是完备的。

答： 正确。因为BFS是一种完备策略,即只要问题有解,它就一定可以找到解。并且,广度优先搜索找到的解,还一定是路径最短的解。

1. 树搜索的深度优先搜索(DFS) 不是完备的。

答：不正确。树搜索的深度优先搜索必须规定了一个深度界限，使搜索在规定范围进行就是完备的。

1. 一致代价搜索(UCS) 是A∗搜索的一种特殊情况。

答：正确。UCS是使A\*搜索的贪心算法启发式函数h(n)≡0的算法。

1. 第二题(15分) 考虑将k匹马放置在n × n的国际象棋棋盘上, 要求两匹马不能相互攻击, 其中k ≤ n^2。
2. 将这个问题形式化为CSP, 在你的形式化中, 变量是什么, 变量的可能值是什么?

答：变量为棋子的坐标X ={X1,X2,X3...Xk},变量的可能值为D={D1，D2,D3..Dk),其中Di=(xi,yi)满足1 <= xi,yi <= n

1. 变量受到那些约束, 约束怎么定义?

答： 显式约束为（X1,X2）∈{(1,1),(1,2)... },...

约束定义：要求X1和X2不会出现(|x1-x2|==2 && |y1-y2|==1)或(|x1-x2|==1 && |y1-y2|==2)的情况

(3) 现在考虑在互相不攻击的约束下如何放置更多的马, 请用一种局部搜索方法(比如模拟退火算法、爬山法) 求解该问题，并写出解决方法的伪代码。

答：如果k>=n^2/2，取k=n^2/2，因为最优情况下n\*n国际象棋只能放置n^2/2个马。

function SIMULATED-ANNEALING(problem, schedule) returns a solution state current ← problem.INITIAL //INITIAL最初随机放置k个棋子的状态

while( J(current)!=0 and T > T\_min) //小于T\_min时结束搜索

{

new <- schedule(t)

　　dE = J(new) - J(current);

　　If(dE >= 0) //表达移动后得到更优解，则总是接受移动

current = new;

else

　　{

if (exp( dE/T ) > random(0,1))

current = new ;

}

　　T = r \* T //0<r<1 ，r越小搜索时间越短，r越大得到全局最优解可能较高

}

3 第三题(15分)

AC-3在Xi的值域中删除任何值时，都把每条边弧(Xk, Xi)放回到队列里, 即使Xk中的每个值都和Xi的一些剩余值相容。假设对每条弧(Xk, Xi), 记录Xi中与Xk的每个值都相容的剩余值的个数。如何有效的更新这些数字使得弧相容算 法的复杂度为O(n^2d^2 )。

答：改进AC3，初始化时维护counter记录每条弧中每个值相容的剩余值。在移除某个值后，不用进行constraint check，只需遍历表和更新相容剩余值的个数（都-1），如果没有0则不用移除。在不牵扯约束具体内容的情况下，已经具有最优的最差时间复杂度。

作业一.对比并分析在不同规模迷宫下，各算法的搜索算法用时，展开节点数，路径代价等指标

|  |  |  |  |
| --- | --- | --- | --- |
| tinyMaze | 算法用时 | 展开节点数 | 路径代价 |
| DFS | 0.0 | 15 | 10 |
| BFS | 0.0 | 15 | 8 |
| A\* | 0.0 | 14 | 8 |

|  |  |  |  |
| --- | --- | --- | --- |
| mediumMaze | 算法用时 | 展开节点数 | 路径代价 |
| DFS | 0.0 | 146 | 130 |
| BFS | 0.0 | 269 | 68 |
| A\* | 0.0 | 221 | 68 |

|  |  |  |  |
| --- | --- | --- | --- |
| bigMaze | 算法用时 | 展开节点数 | 路径代价 |
| DFS | 0.0 | 390 | 210 |
| BFS | 0.0 | 620 | 210 |
| A\* | 0.1 | 549 | 210 |

数据量小时，三种搜索算法表现基本无差异。数据量很大时，深搜虽然展开节点数较少，但不一定能找到最优解，宽搜在寻找最短路径上效率很高，但相对占用的内存较大，A\*算法是比较平衡的算法。

作业二.说明你的启发函数是良定义的：非负，一致性，并且在目标状态下的值为0

答：h(n)设定为没访问过的两个food中最大的曼哈顿距离（如果只有一个food直接计算）+ 当前位置到这两个food中更短的曼哈顿距离为代价。

因为曼哈顿距离满足非负性，在目标位置时值为0，且当前状态到目标状态的估计值<=实际到目标状态的代价，且每一个结点n和通过任一行动a生成的n的每个后继结点n’，从结点n到达目标的代价估计满足h(n) <= c(n, a, n1) + h(n1)的一致性要求。