人工智能实验一 说明文档

——N 皇后问题
王悦
PB13011058

实验题目：

N 皇后问题：N*N 的棋盘上摆放 N 个皇后，使其不能相互攻击，任意两个皇后都不能处于同一行、同一列或同一斜线上。

实验重点：

这个问题的难点在于，时间复杂度随着问题规模是指数型增长的，高效解决这个问题是本作业的重点。

实验要求：

棋盘中某个位置有一个障碍。分别处于障碍相反两侧，并和障碍在一条直线上的两个皇后不会相互攻击。障碍不可以放置皇。

实验记录：

本次实验实现了爬山算法和模拟退火算法，另外实现了效率更高的 QS4 算法。

1. 爬山算法
   算法思想：评价函数为冲突的皇后对，允许水平移动，但是限制移动步数上限 N/4，选择较陡的路径进行搜索，遇到局部最优解非全局最优解，重新随机状态开始爬山。初始状态行列均不冲突，后继函数为交换两行。障碍处理：鉴于只有一个障碍，则对求得的解作对称或者旋转(极小可能需要重新生成解)即可满足要求。
   空间复杂度：避免行列冲突，只考虑对角线冲突，使用两个长为 2*N-1 的整形数组纪录对角线上皇后数目 k，则该线上冲突对数为 k*(k-1)/2，去掉一个皇后，冲突对数减少 k-1 个；增加一个皇后，冲突对数增加 k 个。另外需要一个一维数组存储第 i 行的皇后位置 queen[i]，因此**需要空间为：N+2*(2*N-1)=O(N)**。
   时间复杂度：选择较陡的路径的含义为：首先遍历所有皇后，找到被攻击次数最多的皇后 q，即 q 所在的两条对角线上的皇后和最大；其次将 q 行与其他行进行尝试性交换，计算 value 的变化，减少最多的作为路径。计算 value 变化的方式在上一段空间复杂度提到，只需+k 或者-(k-1)即可，时间为 O(1)。**单次搜索的时间复杂度为 O(n)**，避免了全局搜寻最陡路径造成的 $O(n^2)$的开销，也避免了随机爬山的无用功，尽管随机爬山是 O(1)的。
   实验结果输出到 output_hill_climbing.txt 中，如要求一样。
2. 模拟退火算法
   算法思想：在爬山算法的基础上，设定退火状态参数，在搜索遇到局部最优解非全局最优解的时候，根据温度和 value 变化值设定概率接收 “走一条较坏的路” ,调整参数值(初始温度，退火速度，每温度运行步数)可以实现 1 概率求解，并且得到较好性能。障碍处理等同爬山算法。
   空间复杂度：同爬山算法。
   时间复杂度：同爬山算法。

实验结果输出到 output_simulated_annealing.txt 中，如要求一样。

3. QS4 算法

算法思想：基于 QS1 和 QS2 算法，对不同规模 N 设定参数值 C，C1，C2 不同。首先初始化前 N-C 行，要求不存在行冲突列冲突和对角冲突。对剩余行只要求不存在列冲突。初始 limit 为 C1*初始 value。(0<C1<1) 有一个数组用来存储冲突的皇后，在循环次数小于 C2*N 的时候，从冲突队列选取一行，再随机选取一行进行交互，value 变小则接受交换，并且 value 减少到小于 limit 时，limit 变为原来的 C1 倍，并更新冲突队列。处理完队列后，循环次数加一，重新处理队列（之前的交换可能造成新的冲突）。循环次数超过限制，重启算法，重新运算。

空间复杂度：存储方式与爬山相同，需要对角线存储冲突数 2*（2*N-1），需要存储皇后位置 N，额外需要存储冲突队列 N。**空间复杂度为 O(N)**。

时间复杂度：单次搜索时间复杂度为 O(1)。整体时间复杂度约为 O(NlogN)。

实验结果输出到 output_QS4.txt 中，如要求一样。

另附测试时间（单位：ms）：

实验平台：64 位 linux，编译器 g++，优化等级 O3，处理器 i5 三代，主频 1.8GHz，最大睿频 2.7GHz。

N	爬山	模拟退火	QS4
10	0.39	0.014	0.07
100	0.232	0.283	0.847
1000	3.37	3.334	1.193
10000	48.874	45.443	1.503
100000	276.025	261.8	19.006
1000000	8208.09	8541.16	376.785