人工智能-第二次课程作业报告

授课教师：杨旭 作者：徐子航-61520711

1 问题描述

1.1 题目介绍

本作业基于 Eight Queen Puzzle,目标是基于回溯搜索(backtrackSearch ）和最小冲突搜索（ minConflict ）算法求该问题的解。该部分内容对应《Artificial Intelligence: A Modern Approach3rd》中的第六章内容：Constraint Satisfaction Problems。

该问题为一个 8*8 的棋盘，在该棋盘上摆放 8 个皇后，使其不能互相攻击，即任意两个皇后都不能处于同一行、同一列或同一斜线（主对角线、次对角线）上。

有多种摆放方案可以满足上述条件，例如：

0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0
1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 1
0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0

1.2 任务说明

使用回溯搜索算法和最小冲突搜索算法解决 Eight Queen Puzzle

1.3 实验环境

Visual studio 2022

1.4 评价标准

源代码部分（8 分）

1. 搜索算法的正确性：算法得出的结果可以通过 main.cpp 中 searchTest 函数的验证。
2. 搜索算法的搜索时间

回溯搜索可以在 5 秒内跑出 puzzle 的结果

最小冲突搜索大部分可以在 200 步之内得出结果。
3. 内存管理：算法的内存消耗需要符合该算法应有的空间复杂度。
4. 回溯搜索（第一个函数）和最小冲突（后四个函数）各占 4 分。
文档部分（2 分）
1. 算法、实验结果分析：根据实验结果分析、对比回溯搜索和最小冲突搜索，完成“人工智能-第二次课程作业报告”。（1 分）
2. 代码风格。(1 分)

2 实验方案

2.1 回溯搜索法

为了求得问题的解，先选择某一种可能的情况向前探索，在探索过程中，一旦发现原来的选择是错误的，就退回一步重新选择，继续向前探索，如此反复进行，直至得到解或 证明无解。可以预见的是，随着模型的增大，在这里是棋盘得到增大，尽管该算法总能找到解，但由于其 n^2 的时间复杂度，所需要的时间也会大大增加。

2.2 最小冲突搜索算法

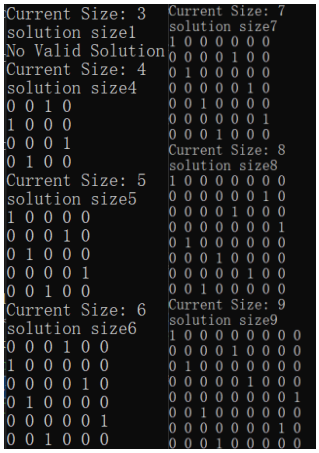
回溯法虽然能找到全部解，但这样势必是低效率的。为此，最小冲突算法选 z4 先进行初始化，然后每步移动一个皇后的位置，移动时要选择与其它变量冲突最小的方式。在实验中，有一个 minConflict 函数，里面有三个子函数。子函数分别用来获取冲突数量、随机选择一个有冲突的变量和计算使冲突最小的可能方式。如果有多种方式使冲突最小，则随机选择。

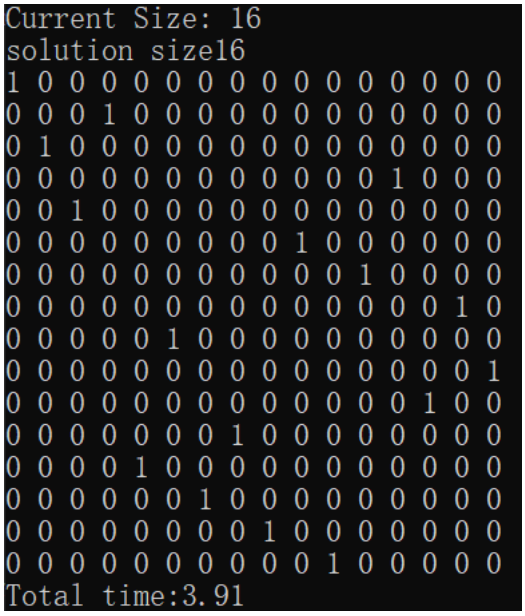
可以看到，比起回溯法，这里有多处要随机选择。这会导致算法最终有可能搜索失败，在 searchTest 中会输出 failed。

3 实验结果

3.1 回溯搜索法

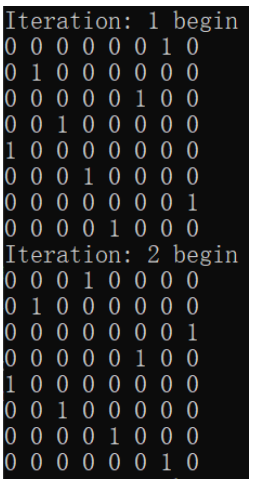
正确运行回溯搜索法的截图





3.2 最小冲突搜索法

正确运行最小冲突搜索法的运行截图



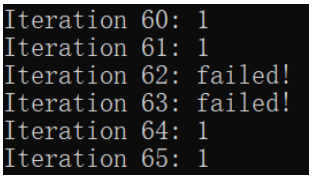
4 实验分析

在计算机算力足够的情况下,回溯搜索算法可以解决任意数量皇后问题，但通过个人编译结果分析，当皇后数量增大时会明显感觉多输出结果速度变慢，其原因也许是时间复杂度为 C^n。下表中的 size 为最大 size，时间为 size 从 1 到 Size n 的累计时间。可以看到，时间的增长几乎是指数级的。使用的计时函数是 clock()。

回溯法耗时记录

Size	8	16	20
Time/s	0.55	3.91	87

最小冲突算法采用某种特殊贪心规则，每次选取一个皇后移动到冲突最小的位置，这种算法能极大的缩短时间复杂度，同样是 size 为 20 的棋盘，用最小冲突算法总共迭代了 100 次，耗时也只有 4.141 秒。但是无论 maxstep 设置为多少，仍然有一定概率无法运行出解，可能是陷入某种死循环导致的。下图中就是 failed 的一个示例。



5 结论

这次实验其实主要是要看懂已给工程文件中定义的类和相关函数，就能顺利完成实验。实验二主要是关于 CSP 中两种算法解 N 皇后问题的实现。经过算法的比较，我深刻体会到了时间复杂度对算法实用性的影响。虽然回溯算法在理论上是能够解 N 皇后问题的，其中 N 为任意大，但由于其超高的时间复杂度，当 size=20 的时候，就已经要等比较长的时间了，我曾还想把 size 设为 32，64，128，但根本等不及。与回溯法相比，虽然最小冲突法不能穷尽所有解，有时还求解失败，但大部分情况下都能快速求解。这种局部搜索，牺牲一些准确与完备性，换取时间复杂度上的大幅优化的做法值得我学习。