模拟退火算法实现报告

王资 18214668 计算机技术

1.基本思想

解决 n 后问题，给定一个初始温度，在每个温度下做足够次数的状态转换，保证到达平衡状态，每个温度达到平衡状态后下降温度，那么最终会找到一个解。

2.代码中的数据结构

int n，代表 n 后问题的规模；int* x，代表棋盘；bool* conf 标识当前棋盘中存在冲突的皇后。

3.程序流程

(1) 随机选择一个解 i，k=0，t0=Tmax(与 n 成正相关)，计算 f(i);

(2) 如果满足结束条件，则转(13)

(3) Begin

(4)　如果在该温度内达到了平衡条件，则转(11)

(5)　Begin

(6)　　从 i 的邻域 N(i)中随机选择一个解 j，计算指标函数 f(j)

(7)　　如果 f(j)<f(i)，则 i=j，f(i)=f(j)，转(4)

(8)　　计算　$Pt(i \to j) = e^{-\frac{f(j)-f(i)}{t}}$

(9)　　如果 Pt(i->j)>Random(0,1),则 i=j，f(i)=f(j)，转(4)

(10)　End

(11)　tk+1=Drop(tk),k=k+1

(12) End

(13) 输出结果

4.调试过程出现的问题和解决方法

根据模拟退火的数学原理，在温度较高的情况下，达到平衡需要的次数不需要太高，因为各个状态的概率是相等的，而在温度较低的情况下，需要尝试多次状态的转移，因此算法几乎只会往优化的方向移动。按照这个思想，本来的设计是平衡条件根据温度的改变而改变，然而对解决 n 后问题的效率没有任何收

益，反而是固定达到平衡条件的次数在一个较小的值时，得到解的速度是最快的。这可能是因为 n 后问题比较特殊，该问题的最优解有极多，所以在每个温度达到平衡的这个条件并不必要，也能随着温度的逐渐降低找到其中一个最优解。

5.运行结果

对于 n=1000 的结果

```
/home/wz/CLionProjects/n-queens/cmake-build-debug/n_queens
temperature dropped 39724 times
641 523 847 466 162 501 316 547 762 560 34 383 283 423 551 938 652 663 660 524 579 315 649 536 764 771 380 460 574 691 312 138 565 229 208 610 581 326 975 62
1000-queens in simulate anneal takes 237s

Process finished with exit code 0
```

跟一般的随机局部搜索算法相比，模拟退火只需要<5min 的时间，而前者需要近 1h.

6.总结

本实验实现了解决 n 后问题的模拟退火算法，本次实验有较多的超参数，包括初始温度、每个温度达到平衡需要的状态转移次数、温度下降的方法等。初始温度需要与问题的规模有关，而不是一个固定的数值，因此设定为 n 的 3/2 次方；通过实验发现对于 n 后问题每个温度达到平衡需要的状态转移次数对找到最优解并不敏感，反而只要能够保证温度逐渐下降就能找到最优解，因此每个温度达到平衡需要的状态转移次数设置为 2；温度下降的方法设置为每次达到平衡后温度下降.01

7.存在问题和改进设想

由于算法需要大量计算棋盘冲突，因此优化计算冲突的算法将对整个算法的效率大有提高，计算冲突数需要的时间消耗较大，是 O(n^2)量级的，而每次皇后位置的交换只交换了 2 个，因此可能可以只关注这 2 个变化了的值，使得冲突数计算可以下降到 O(n)量级，若是能实现这样高效的冲突计算，就能对算法的效率做一大提升。