

X 数独问题

问题描述：

原始的数独问题：在 9*9 格的方格中，玩家需要根据已知数字，推理出所有剩余空格的数字，并满足每一行、每一列、每一个粗线宫（3*3）内的数字均含 1-9，不重复。

总的约束即为：

- 1) 每一行、每一列、每一个粗线宫（3*3）内的数字均含 1-9，不重复。
- 2) 九宫格的两条对角线内的数字也均含 1-9，不重复。

算法思想：

利用递归，每填一个格子之后，走向更深一层。设置一个判断函数，在递归的每一层，利用这个判断函数判断现状态是否出现不满足约束的情况。如果出现，那么就回溯。

优化方法：

对于每一个位置，设置一个长度为 10 的数组，其中 0 的位置代表这个位置还能填入的数字的数目。1-9 代表能否填入对应的数字，0 代表不可以，1 代表可以。

每一次去选择下一个填入的位置时，先进行比较，选择出各位置的数列[0]位中数值最小的那一个。同时对其同一行、同一列、同一格或者是可能存在对角线上的位置对应的数列中位置进行修改。

搜索遍历的结点数：

	1	2	3
未优化	240	16668	8232801
优化	167	22196	2119738

搜索花费的时间：

	1	2	3
未优化	0.08823s	0.09919s	4.014s
优化	0.08657s	0.1414s	4.753s

分析：

可以看到在优化了的情况下，访问的节点数一般都是减少的（例 2 不知道怎么回事）。

但在访问较少节点的情况下，在例 3 中却花费的较多的时间。其原因应归结于在寻找最小的可能节点的时候，要遍历整个数独，也就导致了搜索时间的增长。同时在更新其他点状态的时候，也会花费不少的时间，这也就导致了时间的增加。

思考题:

- 1、可以。首先设置一个集合 A，将题目给定的数字放入 B。再设置一个集合 T，放入所有空的位置。对所有 B 中的位置进行随机填入 1-9 的数字。然后记录行、列、格、对角线上冲突的对数（任意两个为一对），求得总和作为 f。每一次改变其中 T 中一个数字，使其向 f 更小的方向发展，或者利用模拟退火进行降温，亦或是利用遗传算法进行运算。
- 2、由于这些算法都是局部最优方法，可能会一直停在一个局部最优解的附近，导致没法得到全局最优解，也就没法得到输出。