## 数独问题

### 1 实验任务

·9\*9的矩阵，要求每一行，每一列，每个九宫格都是1-9这九个数字且不能重复。给定一9\*9矩阵，里面有部分数空缺，要求找出满足上述要求的一个矩阵

·可选算法：搜索+剪枝（递归+回溯）

### 2 实验原理

#### 2.1 状态空间法

**2.1.1 问题的构成**

状态空间表示法就是以“状态空间”的形式来表示问题及其搜索过程的一种方法，它是人工智能中最基本的形式化方法，是讨论问题求解技术的基础，其由以下几个基本要素构成。

**状态：**表示问题求解过程中不同时刻问题状况的数据结构，一般用一组变量的有序集合表示，可形式地表示为： = {, , … ,}(i=0, 1, 2,…, n)，

如在本实验中的状态可表示为，每一步9\*9矩阵的值。

**算符：**引起状态中某些变量发生变化，从而使问题由一个状态变为另一个状态的操作，可分为走步、过程、规则、数学算子、运算符号或逻辑符号等，如在本实验中算符即是在9\*9矩阵中空位置填上1-9这九个数字。

**状态空间：**一个问题的全部状态及一切可用算符构成的集合，常用一个三元组表示为(S, F, G)，其中，S为问题的所有初始状态集合，F为算符的集合，G为目标状态的集合。

**问题的解：**从问题的初始状态集S出发，经过一系列的算符运算，到达目标状态，

由初始状态到目标状态所用算符的序列就构成了问题的一个解。

**2.1.2 求解的过程**

问题的求解过程就是不断把算符作用于状态的过程，具体可以描述为：

1）将适用的算符作用于初始状态，以产生新的状态

2）把一些适用的算符作用于新的状态，如此反复，直至产生的状态为目标状态

3）由初始状态到目标状态所用算符构成的序列就是该问题的一个解

#### 2.2 回溯算法

回溯法思路是把问题的解空间转化成了图或者树的结构表示，然后使用深度优先搜索策略进行遍历，遍历的过程中记录和寻找所有可行解或者最优解。基本思想类同于：图的深度优先搜索以及二叉树的后序遍历。

回溯法按深度优先策略搜索问题的解空间树。首先从根节点出发搜索解空间树，当算法搜索至解空间树的某一节点时，先利用剪枝函数判断该节点是否可行（即能得到问题的解）。如果不可行，则跳过对该节点为根的子树的搜索，逐层向其祖先节点回溯；否则，进入该子树，继续按深度优先策略搜索。

回溯法的基本行为是搜索，搜索过程使用剪枝函数来为了避免无效的搜索。剪枝函数包括两类：1使用约束函数，剪去不满足约束条件的路径；2使用限界函数，剪去不能得到最优解的路径。如在本实验中，当某一状态中某一行，或某一列，或某个九宫格存在数字重合，则无需继续搜索其后继状态。

### 3 程序设计

#### 3.1 文件描述

**main.py**：主程序

·draw(ax, grid, stack=None)：绘制九宫格

·prune(ax, old\_grid, stack)：剪枝

·dfs(ax)：深度优先搜索求问题的解

#### 3.2 关键代码

**深度优先搜索**

def dfs(ax):

stack = []

old\_grid = my\_grid.copy()

while True:

status, empty, new\_grid = prune(ax, old\_grid, stack)

if status == True:

return new\_grid

elif status == False:

while True:

node = stack[-1]

if node['value'] == []:

stack.pop()

else:

i, j = node['i'], node['j']

v = node['value'].pop()

node['v'] = v

old\_grid = node['grid'].copy()

old\_grid[i][j] = v

break

else:

i, j = empty[0]['i'], empty[0]['j']

v = empty[0]['value'].pop()

node = {

'i': i,

'j': j,

'v': v,

'value': empty[0]['value'].copy(),

'grid': new\_grid.copy()

}

stack.append(node)

old\_grid = new\_grid

old\_grid[i][j] = v

### 4 实验结果与分析

#### 4.1 运行示例

如图12所示，白色数字为题目所给出的，绿色数字是深度优先搜索栈中保存的结点，红色数字是由栈中结点搜索的结果。

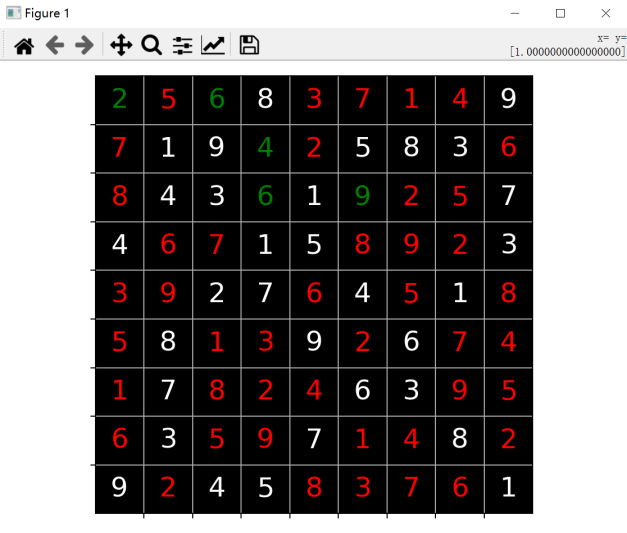
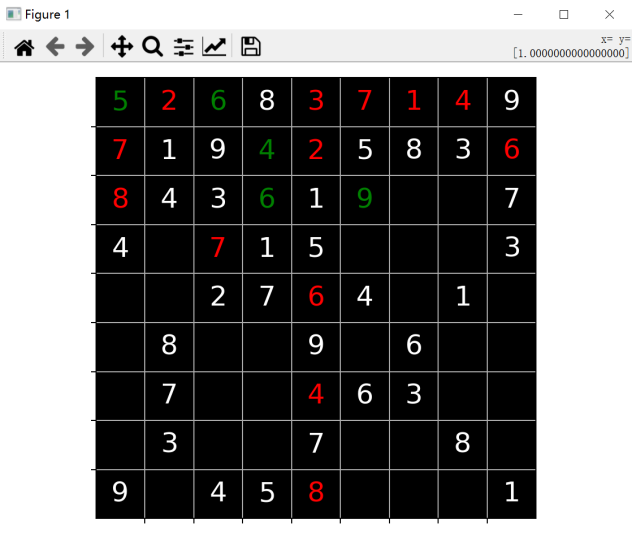
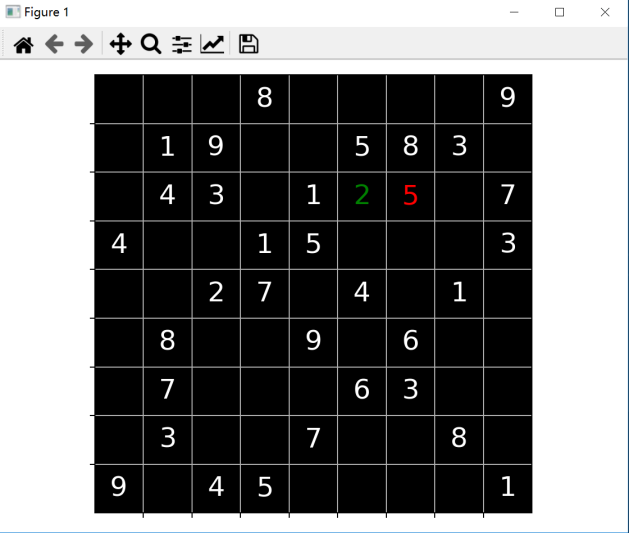


图12 数独

#### 4.2 结果与分析

本次实验中，我采用回溯法实现了对数独问题的求解，并利用剪枝策略大大减小了问题空间的搜索时间，效果令人满意。通过这次实验，我对人工智能课上的状态空间表示法有了更深的理解，对搜索算法有了更清晰的认识。