

人工智能课程算法可视化

- 开发平台: Qt 5.12.0
- 语言: c++

./code 存放的是相关源代码

./release 存放的是通过qt打包的可执行文件

运行./release/ai_practice.exe 后可以看到左上角的 treeWidget

- 旅行问题
 - DFS
 - BFS
 - 一致优先搜索
- 八数码
 - BFS
 - A*
- 八皇后
 - 爬山法
 - 模拟退火
- 一字棋
 - alpha-beta剪枝

可以点击上面具体的问题的实现算法进行查看

旅行问题

实现代码为 ./code/dfslog.cpp

- 演示前我们需要按 随机生成图 来随机生成一个无向图 旁边的 圆形排列 checkBox可以选择图的表现形式,多次取消勾选可以随机展示同一副图的不同形式.
- 对于每幅图,绿色边缘代表**起点**,红色边缘代表**终点**,黄色边缘代表**搜索过的点**,黄色填充代表**当前**.
- 对于BFS和DFS,点的标号代表名字.对于一致优先搜索,点的编号代表**点的代价**.
- DFS用递归实现;BFS用队列实现,出队列时判断是否为终点.一致优先搜索用STL的优先队列实现.
- 下一步 和 上一步 能够改变图形显示界面中的 步骤 ,范围为1-N, 完成 则直接显示第N步

八数码

实现代码为 ./code/a_star.cpp

- 同样,演示前我们需要按 随机生成 来随机生成一个排列方式 旁边的 仅显示正确步骤 checkBox可以选择搜索步骤的展示
- 在A*中,g(x)定义为**搜索深度**,h(x)定义为每个数到正确位置的**曼哈顿距离**
- 在BFS中利用了STL中的**map**进行状态去重和记录,A*利用了**迭代加深**的思想进行搜索
- 下一步 和 上一步 能够改变图形显示界面中的 步骤 ,范围为0-N, 完成 则直接显示第N步

八皇后

实现代码为 ./code/climb.cpp

- 同样,演示前我们需要按 随机生成 来随机生成一个排列方式,生成成功过后会弹出窗口显示是否搜索到正确解.
- 旁边的 最大搜索次数 可以改变最大搜索次数,按下重新搜索就可不换图重新搜索.
- h(x)定义为当前局面发生**冲突的皇后个数**
- 由于爬山法算法的局限性,很容易从某个状态开始就保持不变,陷入**局部最优**,因此随着步骤增多显示的图不会改变
- 由于模拟退火的随机性,同一个图,每次的搜索结果也会不同,即使不改变搜索次数也可能一会儿成功一会儿失败.
- 下一步 和 上一步 能够改变图形显示界面中的 步骤 ,范围为0-N, 完成 则直接显示第N步

一字棋

实现代码为 ./code/chess.cpp

- 开始前需要选择棋盘大小,胜利条件,搜索深度,先后手然后按 设置 开始
- 输入图中序号并按下确认下子,输入框内只能输入最多两位数字.
- 评价函数设置为用**自己棋子填满空位后能得到的'一'的个数**
- 由于评价函数太过简单,只有在井字棋时搜索能得到好一点的结果,其它情况下表现的都很蠢