

人工智能实验报告

实验名称： 局部搜索—八皇后问题

|  |  |  |  |
| --- | --- | --- | --- |
| 学生学号 | 202110120401 | 202110120420 | 202110120408 |
| 学生姓名 | 刘广润 | 赵威 | 喻慧超 |
| 分工 | K-Means实现,测试算法性能，撰写报告 | 算法速度测试，绘制表并进行对比，撰写报告 | 优化算法、撰写报告 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **实验评价结果** | | | | |
| **占比** | 10% | 40% | 50% | 总分 |
| **评分标准** | 报告的规范性。报告中的术语、格式、图表、数据、公式、标注及参考文献是否符合规范要求。 | 报告的严谨性。结构是否严谨，论述的层次是否清晰，逻辑是否合理，语言是否准确。 | 实验的充分性。实验是否完备，数据是否合理，是否有创新性成果或独立见解。 |
| 评分 |  |  |  |  |

报告提交时间： 2023 年 12 月 23 日

# 试验任务

要求在国际象棋棋盘中放置8个皇后，使得任何一个皇后都不能攻击其他任意一个皇后。国际象棋的规则为一个皇后可以攻击同一行、同一列、同一对角线上的棋子。图1.1中的布置并不满足要求，因为第4列和第7列的皇后在同一对角线上。

# 实验平台

Windows操作系统，Python编译环境，math，random，matplotlib等程序库。

# 实验原理

## 贪婪爬山法

贪婪爬山法是一种局部搜索算法，它从一个随机的初始状态开始，不断选择当前状态的最佳邻居状态作为下一个状态，即选择能够使目标函数值最优（例如，在最大化问题中是最大、在最小化问题中是最小）的邻居。算法在每一步都局部地做出最优选择，希望这样能找到全局最优解。然而，贪婪爬山法可能会遇到局部最优解而提前停止搜索，因此并不总是能保证找到全局最优解。在八皇后问题中，贪婪爬山法通过选择减少皇后相互攻击的冲突数来移动每个皇后。

## 侧向移动爬山法

侧向移动爬山法是贪婪爬山法的一个变体，它允许在搜索过程中，当无法找到更好的邻居状态时，做出水平移动（即选择一个与当前状态有相同评价值的邻居）。这种方法旨在防止算法过早地停滞在局部最优解，通过允许在一定次数内的侧向移动来探索平坦区域，增加了找到更好解的可能性。但侧向移动爬山法仍可能会在没有改进的情况下陷入循环，因此通常会设置侧向移动的次数上限。

## 随机爬山法

随机爬山法是爬山算法的一种，它在每一步并不总是选择最好的邻居状态，而是从当前状态的邻居中随机选择一个状态作为下一个状态。这种方法引入了随机性，使得算法有一定的概率跳出局部最优解，增加了探索解空间的随机性。随机爬山法的一个缺点是它可能不如贪婪爬山法那样直接朝着解的方向前进，因此可能需要更多的时间来找到解决方案。但在某些情况下，引入的随机性可以帮助算法避免局部最优的陷阱。

## 模拟退火算法

模拟退火算法是一种基于概率的优化技术，受到物理学中金属退火过程的启发。它结合了随机搜索和局部搜索的特点，旨在找到全局最优解而不仅仅是局部最优解。

模拟退火的核心思想是在搜索过程中允许临时接受比当前解更差的解，从而有机会跳出局部最优解。算法开始时设定一个较高的“温度”，这时接受较差解的概率较高。随着迭代的进行，温度逐渐降低，接受较差解的概率也逐渐减小。这个过程类似于金属退火，先是加热然后缓慢冷却，以达到更稳定的结构。

在每一步，模拟退火算法随机选择一个邻居状态，如果这个邻居状态比当前状态更好，则移动到这个邻居状态；如果邻居状态更差，算法也可能根据一定的概率接受这个更差的状态，这个概率随“温度”降低而减小。

模拟退火算法的性能在很大程度上取决于温度降低的速率（冷却计划）和初始温度的选择。正确调整这些参数对于算法能否有效地找到全局最优解至关重要。

# 实验步骤

## 算法实现

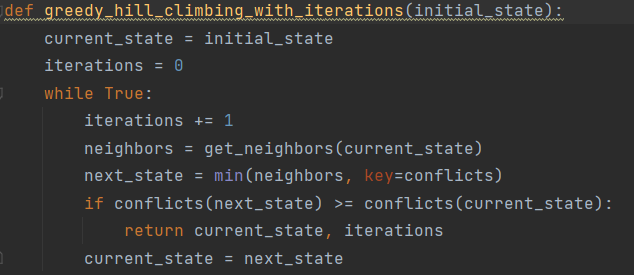


图 1 贪婪爬山算法实现

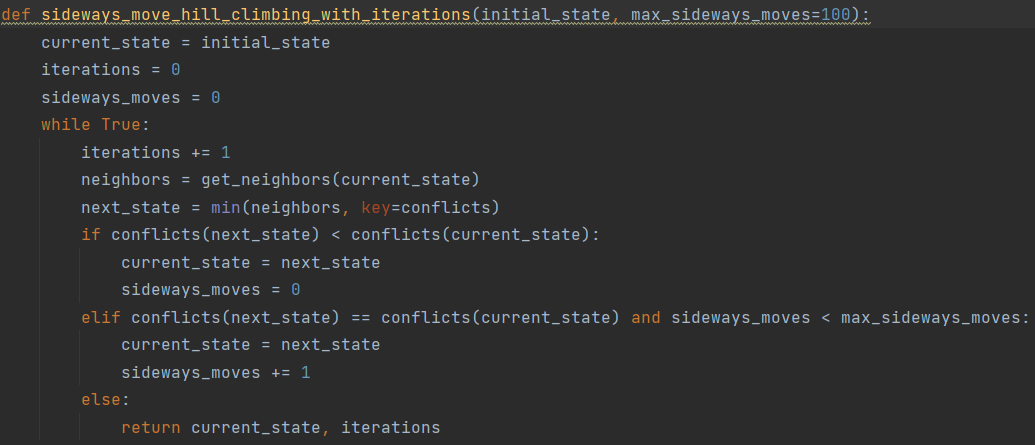


图 2 侧向移动爬山算法实现

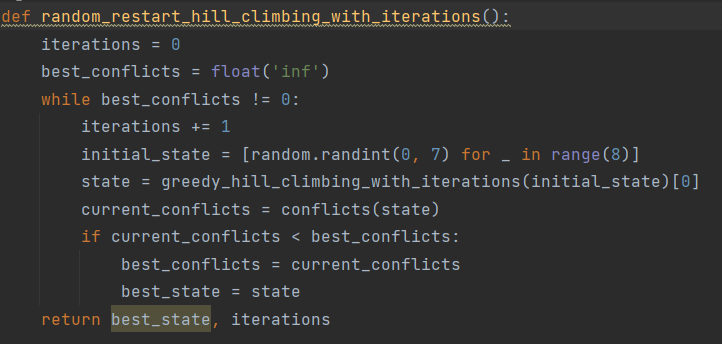


图 3 随机爬山算法实现

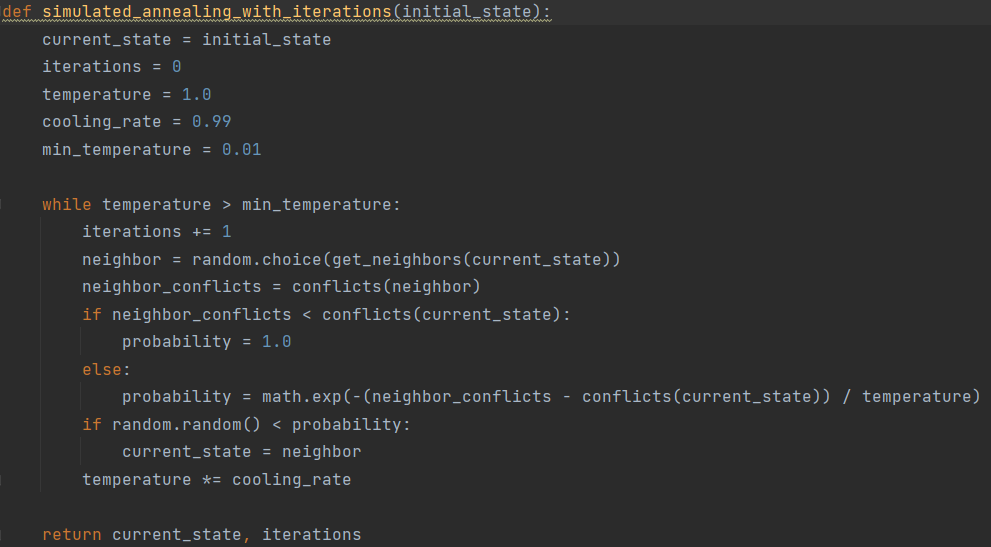


图 4 模拟退火算法实现

## 实验结果

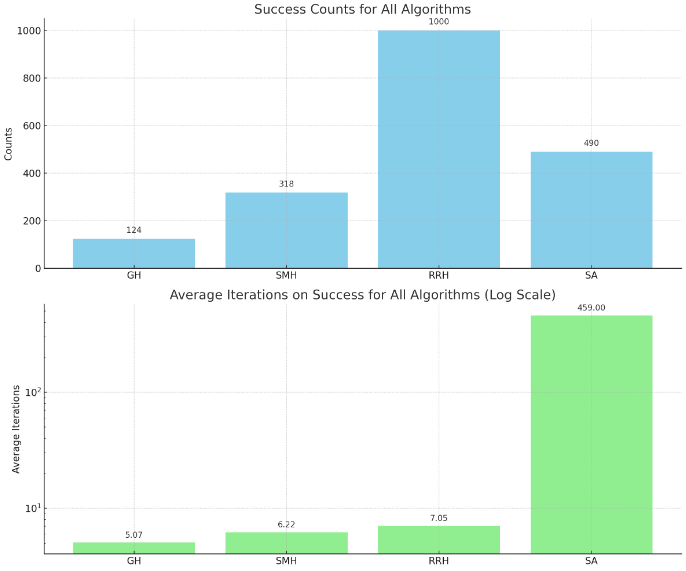


图 5 随机产生1000个起始点成功求解次数(上)求解成功迭代次数(下)

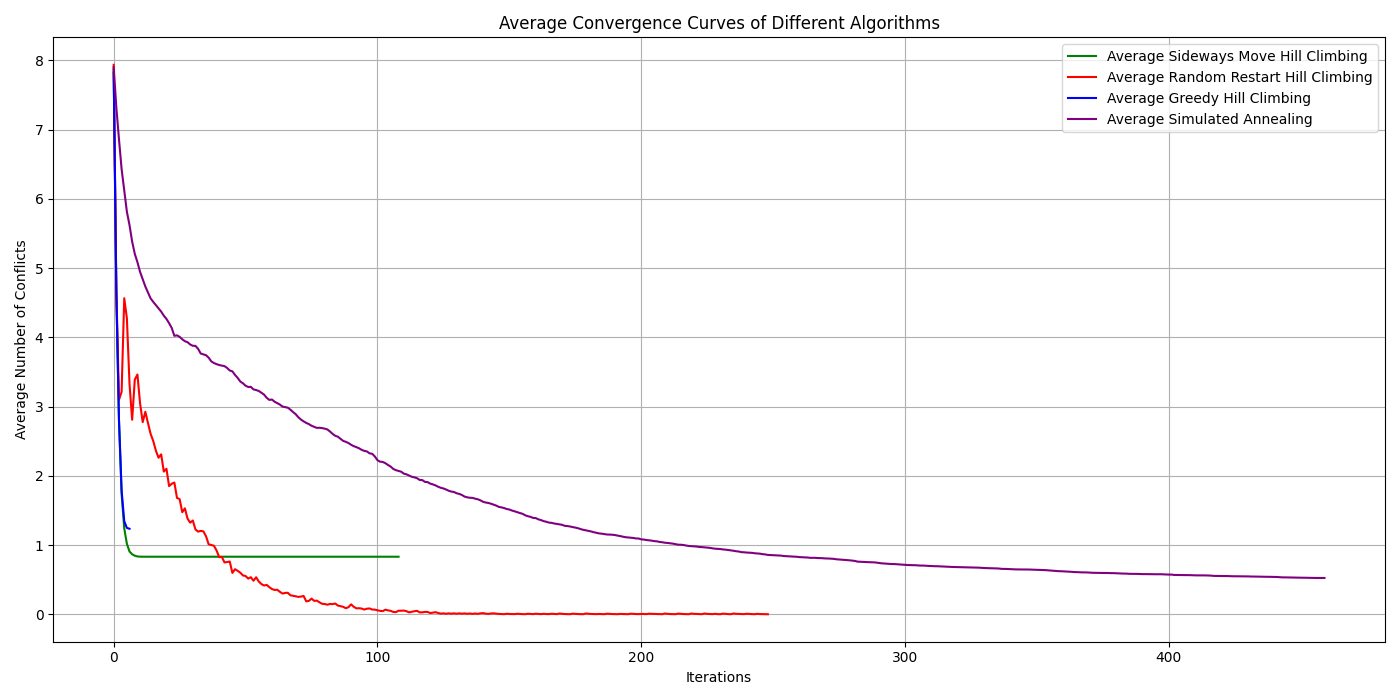


图 6 平均收敛曲线

# 实验结果分析

## 成功求解次数\迭代次数

### GH

* 成功次数较低，因为在复杂的问题中贪婪爬山法容易陷入局部最优解。
* 平均迭代步数最少，表明当算法找到解时，它能够快速收敛。

### SMH

* 成功次数高于贪婪爬山法，因为侧向移动允许算法在局部最优解中探索，从而有更多机会找到全局最优解。
* 平均迭代步数略高于贪婪爬山法，反映了侧向移动可能会使算法在局部最优解处停留更长时间。

### RRH

* 成功次数非常高，表明通过从多个随机初始状态开始几乎总是能找到问题的解。
* 平均迭代步数较低，表明尽管算法可能需要多次重新启动，但每次成功找到解时通常不需要太多迭代。

### SA

* 成功次数较高，这表明模拟退火算法能够有效地逃离局部最优解并探索解空间。
* 平均迭代步数显著高于其他算法，这是因为模拟退火算法在每一步都有可能接受一个更差的解，这样的随机探索过程通常需要更多迭代步骤。

## 平均收敛曲线

### GH

贪婪爬山法的曲线迅速下降并很快达到了低冲突数，这表明该算法通常能够快速找到冲突数较少的解，但由于其贪婪的性质，它可能会陷入局部最优解，因此曲线在达到低冲突数后趋于平稳。

### SMH

侧向移动爬山法的曲线类似于贪婪爬山法，但在到达低冲突数的过程中有轻微的波动。这些波动可能代表了算法在局部最优处进行侧向移动的尝试，这有助于算法跳出局部最优并继续寻找更好的解。

### RRH

随机重新启动爬山法的曲线在初始阶段迅速下降，这是因为每次重新启动都给了算法一个新的随机初始状态，并且每次重新启动都可能直接或间接带来更少的冲突数。随着足够多的重启，这个算法最终能找到无冲突的解决方案。

### SA

模拟退火算法的曲线呈现出一个平缓的下降趋势，这反映了模拟退火算法在搜索过程中的概率性特征。由于算法在降温过程中逐步减少接受更差解的概率，因此收敛过程比其他算法更为缓慢，但这种渐进的搜索有助于它避免陷入局部最优解。

# 代码

https://github.com/SonicAge1/LocalSearch