

局部搜索算法的实验分析

实验零：补全代码，并将填空内容写在实验报告中

```
hill_climb.hpp
// 状态转移（需要参考 state, selection 的接口）——1 行
state=state.neighbor(permutation[selection.selected_index()]);

conflicts_minimize.hpp
// 将变元新值产生的冲突数的评估价值提交给选择器（需要参考 selection, problem
的接口）——1 行
selection.submit(value_of(-problem.conflicts_of(selected_index)));

simulated_anneal.hpp
if (value_diff > 0 || RandomVariables::uniform_real() < std::exp(value_diff
/ temperature)) {
    state = new_state;
```

实验一：爬山算法求解 n 皇后问题

实验目的：掌握爬山算法的思想与实现，探究影响爬山算法效果的因素

实验步骤：认真阅读作业要求说明文档中的说明，调整爬山算法的参数，尝试在 5 秒钟之内求解皇后数 n 尽可能大的 n 皇后问题（仅需输出 1 个解）

实验结果：

1. 参数设置情况

参数	参数取值
问题模型	QueensSwapState
随机重启尝试次数	5
单次最大爬山步数	4*n (840)
选择算法	f_selection (首个更优选择)
状态估值函数（若选择算法不是轮盘赌则不填此项）	/

2. 5 秒内解出的最多皇后数 n 为：( 210 )
3. 简述除参数外对算法做出的修改（无修改则填无）：无

实验结果分析：

默认值可达标。

爬山算法思想是在每一步选择当前邻域最优解以逼近全局最优，实现时从初始解出发迭代评估相邻解，保留更优解直至无改进时终止。

使用 QueensSwapState 建模，f\_selection 是贪心策略，每次选择第一个比当前状态更优的邻居跳转，不需要遍历所有邻居，运行效率较高，运行时间显著优于最大选择和轮盘赌，更适合实验一要求的限制时间内求出 1 个解的方法。

在 200 左右的问题规模下，猜测状态空间比较好（局部极值和平台不多），3 种算法在找到解方面都可以达成，在时间成本上差异比较大。

实验二：模拟退火算法求解 n 皇后问题

实验目的：掌握模拟退火算法的思想与实现，探究影响模拟退火算法效果的因素

实验步骤：认真阅读作业要求说明文档中的说明，调整模拟退火算法的参数，尝试在 5 秒钟之内求解皇后数 n 尽可能大的 n 皇后问题（仅需输出 1 个解）

实验结果：

1. 参数设置情况

参数	参数取值
问题模型	QueensMoveState
随机重启尝试次数	36
温度随时间的变化函数	$\text{Temperature\_schedule\_move}/\exp(n/\log(n)-\text{time}/16n)$
时间结束时的温度	$1e-18$
状态估值函数	最大冲突数-当前冲突数

2. 5 秒内解出的最多皇后数 n 为：( 250 )
3. 简述除参数外对算法做出的修改（无修改则填无）：无

实验结果分析：

默认值可达标。

模拟退火算法主要是为了解决爬山法容易陷入局部最小值的问题而设计的，允许在早期依概率跳到更差状态，因此尝试次数可能增加。

与其他实验对比：最多皇后数 n 又小幅提升，可能说明模拟退火算法可以更快跳出局部极值/平台，找到全局最优解，可以求解状态空间更复杂的问题。

实验三：遗传算法求解 n 皇后问题

实验目的：掌握遗传算法的思想与实现，探究影响遗传算法效果的因素

实验步骤：认真阅读作业要求说明文档中的说明，调整遗传算法的参数，尝试在 5 秒钟之内求解皇后数 n 尽可能大的 n 皇后问题（仅需输出 1 个解）

实验结果：

1. 参数设置情况

参数	参数取值
进化代数	$8*n(640)$
种群大小	$8*n(640)$
突变概率	0.9

2. 5 秒内解出的最多皇后数 n 为：( 80 )
3. 简述除参数外对算法做出的修改（如基因编码、交叉、变异方式等，无修改则填无）
- 找到一个解之后即终止运行，不再继续迭代。
- 在 class QueensPopulation 中添加 bool Solution\_found, 用 if ( \_adaptability[i] == 0) { 检查是否找到解
- ```
        solution_found = true;
```

实验结果分析：

默认值即可达标。

本实验用遗传算法求解n皇后问题，使用皇后所在列作为基因序列，选择适应度最高的个体作为亲本单点交叉，再以 0.9 的概率对个体进行编译。适应度为-冲突数，算法目标为适应度=0。

该算法对突变率敏感。n=60 时，突变率设为 0.9 时可以在 8\*n 个 Epoch 内达标。突变率稍有减小则有可能陷入局部最优，n>50 时则无法在 8\*n 代内找到最优解。n=50 时，取突变率 0.8 比 0.9 所用 epoch 数量少，可能说明适当降低突变率可以提升解的稳定性。

与本次作业的其他算法对比，发现遗传算法可解决的问题规模较小，这说明虽然遗传算法在某些问题上很有优势，但遗传算法的不稳定性对其解决问题的能力造成了比较大的限制。

实验四：冲突最小化算法求解 n 皇后问题

实验目的：掌握约束满足问题模型，掌握冲突最小化算法的思想与实现，探究影响冲突最小化算法效果的因素

实验步骤：认真阅读作业要求说明文档中的说明，调整冲突最小化算法的参数，尝试在 5 秒钟之内求解皇后数 n 尽可能大的 n 皇后问题（仅需输出 1 个解）

实验结果：

1. 参数设置情况

| 参数                       | 参数取值        |
|--------------------------|-------------|
| 随机重启尝试轮数                 | 10          |
| 单轮变元最大修改次数               | 4*n (40000) |
| 选择算法                     | 最大选择        |
| 冲突数估值函数（若选择算法不是轮盘赌则不填此项） | /           |

2. 5 秒内解出的最多皇后数 n 为：( 10000 )

3. 简述除参数外对算法做出的修改（如选择待修改变元、待修改变元选择新值方式等，无修改则填无） 无

实验结果分析：

默认值即达标。

冲突最小化算法的思想是通过调整变量或策略来降低系统中各元素间的冲突程度，实现时通常基于启发式方法迭代优化配置，直至冲突值达到可接受范围或稳定状态。

使用最大选择策略的 n 值 (10000) 显著大于首次最优选择和轮盘赌选择的 n 值 (均<2000)，说明最大选择策略在此处是最优的。

冲突最小化算法可以解决大规模的问题，与其他实验对比说明其在大规模问题上的表现明显优于其他几个算法。

四百万皇后：

```
Input N: 4000000
Initialized: pre-positioned queens = 3999993, initial conflicts = 8
Final conflicts: 0, Time used: 2.521 sec.

-----
Process exited after 4.735 seconds with return value 0 (2062 ms cpu time, 282648 KB mem used).
Press ANY key to exit...|
```

注：由于电脑配置问题此次部分代码在学校机房电脑完成编译和运行，故可能有少量代码未更新为最新版本，以报告内容为准。