五子棋 AI 项目报告

John Class 2023
卢黎鸣 523031910726

1 效果展示

1.1 可交互前端

基于 html 实现了可视化图形界面，用户可以通过鼠标点击的方式与 AI 进行对弈。

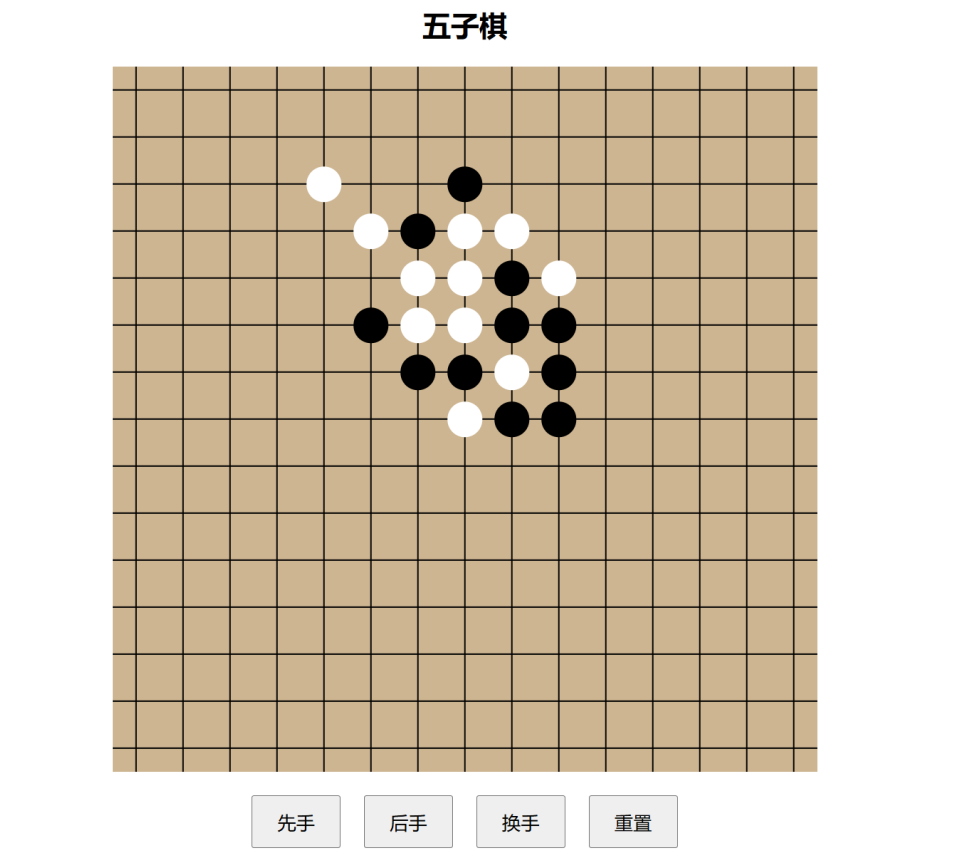

图 1: 可交互前端示例

1.2 棋力测试

与助教下发的基于 MCTS 算法实现的 baseline 程序进行对弈，得到了不低的胜率。

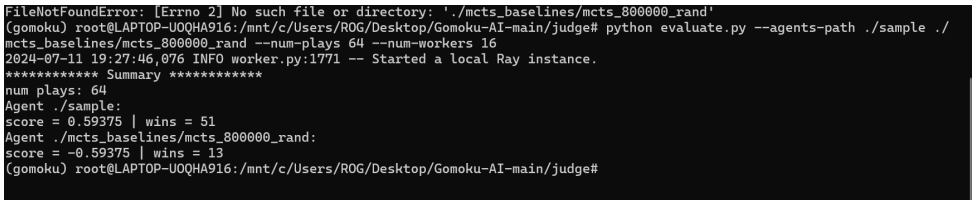

图 2: 与 baseline_800000_rand 对弈结果

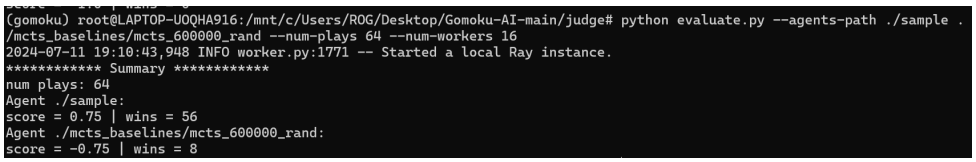

图 3: 与 baseline_600000_rand 对弈结果

2 运行方法

2.1 环境配置

请在 Unix 环境下运行（如 WSL，MacOS）。除了 C++ 外还需要配置 Python 环境。必须使用 Python3.7 版本及以上。完成本项目至少需要以下 Python 包：

- subprocess
- timeout_decorator
- numpy
- sys
- time
- Flask

可以使用以下命令来安装包依赖：

```
pip install -r requirements.txt
```

（以上摘自项目 github 文档 Rick0827/Gomoku-AI: Program Design and Data Structure III (John Class 2024) Project Repo (github.com)）

2.2 运行程序

配置好环境后，在/Gomoku-AI-main/demo 目录下输入：

```
python run.py
```

再用浏览器打开 http://127.0.0.1:5000。

看到棋盘页面后请先选择点击“先手”或是“后手”，即可与 AI 开始对弈。（若直接点击棋盘会造成 AI 未响应的情况）

2.3 AI 版本

前端默认连接的五子棋 AI 为/Gomoku-AI-main/judge/sample，此为基于极大极小搜索算法实现的五子棋 AI，棋力较强。此外还提供了基于蒙特卡洛树搜索算法实现的五子棋 AI /Gomoku-AI-main/judge/mcts。该 AI 棋力较弱，若想与该 AI 对弈，可以在 /Gomoku-AI-main/demo/run.py 中修改 ai_path 从 .../sample 为 .../mcts。

3 实现原理

3.1 sample.cpp:

极大极小值搜索：算法的基础使用了 Min-Max 搜索算法，通过搜索当前局面上各对双方最有利的点位进行模拟对弈。程序中给出了 evaluate 函数对棋盘进行估价，每次 AI 会选择使棋盘价值对本方提升最大的点位落子。搜索深度为六层。

Alpha-Beta 剪枝：六层搜索深度过大，无法在 5 秒内计算完毕，此处使用 Alpha-Beta 剪枝进行优化。即如果在 max 层搜到了一个大于上一层最小值的节点，可以跳过整个子树的搜索，从而减小搜索范围进行提速。

启发式搜索：Alpha-Beta 剪枝的提速效果一定程度上依赖对于点的搜索次序，我们希望先对“好”的点进行搜索，从而后面的“坏”点可以直接被减掉。对于一个格点“好”“坏”的评判基于启发式函数 evaluate_point，若该点落子后形成的棋形越强，则得分越高。最后程序只搜索排名前三十的点。

算杀：目前对于各个点的搜索深度均为六层，但我们希望在一些特殊点，如冲三、冲四上能够进一步加深深度甚至算到必杀棋。程序实现了十层算杀，每层只考虑己方的冲三、冲四等进攻棋形、与对方的冲四或防守，若能算得杀棋，则按照此策略落子。

3.2 mcts.cpp:

蒙特卡洛树搜索算法：极大极小值搜索算法中，局面的评估函数依赖人为定义，并不能“客观”反映局势的好坏。蒙特卡洛树搜索算法，通过每次选择最“优”（UCB 值最大）的格点，进行随机落子直至一方胜利，来进一步评判该格点的好坏。程序在每一次落子前会对全局进行 50000 次模拟对弈，选择得分最高的格点点进行落子。

4 瓶颈与缺陷

- 静态的局面评估函数。每一层搜索中，局势分数都需要全局扫描重新计算，效率较低，事实上可以优化为对于最新落子的周围进行扫描更新即可。
- 实现代码不够精简。在评估函数中，已经做了一定的封装，但总体代码风格仍然较为杂乱不便阅读。
- 缺少对于“换手”机制的运用。该 AI 可以满足玩家的“换手”需要，但不会主动运用“换手”这一机制来提高胜率。
- MCTS 算法实现中过大的内存占用。该算法在实现时，每一个搜索树的节点都保存了当前整个棋局状态，造成了较大的内存开销，导致搜索次数无法进一步提升。可以用状态压缩的方法，用 15 个 16 位整数来表示整个棋盘。
- 缺少悔棋功能。暂不支持悔棋功能，使得玩家体验有所下降。

5 致谢

极大极小搜索算法基础参考了五子棋 AI 教程第二版 五子棋 AI 教程第二版。
部分前端语言实现与学习由 ChatGPT-4o 辅助完成。