
作业 1：搜索

清华大学软件学院
人工智能导论, 2025 年春季学期

介绍

本次作业需要提交说明文档（PDF 形式）和 Python 的源代码。注意事项如下：

- 本次作业满分为 100 分，附加题 5 分，得分超过 100 分按 100 分记。
- 除简答题、编程题外的题目，请给出必要的解答过程，只有答案且过于简略的回答会酌情扣除一定分数。题目要求直接回答或只汇报结果的题目，则不需要给出过程或分析。
- **请不要使用他人的作业，也不要向他人公开自己的作业，复制网上内容须在报告中说明**，否则将受到严厉处罚，作业分数扣至-100（即倒扣本次作业的全部分值）。
- 完成作业过程中，如果使用了大模型辅助（如润色文笔、询问知识点等），请在作业末尾声明使用的方式和程度（不影响作业评分）。**禁止直接粘贴大模型输出的文本**，否则会扣除一定的作业分数。
- 统一文件的命名：{学号}_{姓名}_hw1.zip。所有解答和实验报告请写在一个 pdf 文件中，和代码一起压缩上传。

1 简答题（25 分）

1. 相比树搜索，图搜索作出了什么改进？
2. 相比宽度优先搜索（BFS），深度优先搜索（DFS）和一致代价搜索（UCS）分别有什么优劣？
3. 在约束满足问题（Constraint Satisfaction Problems）中，搜索算法为什么每次要选择约束尽量多的变量（most constrained variable）和约束尽量少的值（least constrained value）？
4. 随机集束搜索（Stochastic Beam Search）为什么要引入随机性？它是否能保证找到最优解？
5. Min-Max 搜索中，若被 Alpha-Beta 剪枝的节点状态值的符号反转，根节点决策有可能发生改变吗？为什么？

2 A* 算法的性质 (25 分)

在 A* 算法中, 记搜索起点为 S , 终点为 T 。对于节点 n , 记 $g(n)$ 表示当前搜索过程中从根节点到 n 的总代价, $h^*(n)$ 为 n 到目标节点 T 的最小代价, $h(n)$ 为 n 节点的启发 (heuristic) 函数。在以下问题中, 设 $h(T) = 0$, 且所有代价均非负。假设 A* 算法每次选择边缘集里 $f(n) = g(n) + h(n)$ 最小的节点。若采用树搜索, 试证明:

1. 若对任意节点 n 均满足 $h(n) \leq h^*(n) + C1$, 其中 $C1 \geq 0$ 为常数, 则 $g(T) \leq h^*(S) + C1$;
2. 若对任意节点 n 均满足 $h(n) \leq C2 \cdot h^*(n)$, 其中 $C2 \geq 1$ 为常数, 则 $g(T) \leq C2 \cdot h^*(S)$ 。

提示: 算法未必会找到 S 到 T 的最优路径。讨论当 T 即将取出边缘集时, 最优路径上位于边缘集中的点 n^* 。事实上, n^* 满足 $h^*(S) = g(n^*) + h^*(n^*)$ 。

3 蒙特卡洛树搜索 (45pt+5pt)

本题为编程题, 代码和相关文档在下发文件的 `./code` 目录下。

问题背景 五子棋 (GoBang) 是一种常见的棋类对弈游戏, 游戏中, 对局双方轮流在 15×15 的网格棋盘的空位落子, 首先将己方棋子在横向、纵向或斜向上连成 5 个者获胜。五子棋可以推广成 $n \times n$ 棋盘上的 k 子棋, 即将连成 5 子获胜改为连成 k 子获胜。我们常玩的井字棋 (Tic-Tac-Toe) 就是 3×3 棋盘上的 3 子棋。围棋 (Go) 的棋盘和棋子与五子棋类似, 但规则更为复杂。在本学期的作业中, 我们对围棋规则做了适当简化, 你可以参阅 `doc/AI2025-HW-GoEnvManual.pdf` 了解详细规则。你只需要大概了解围棋的规则即可完成作业, 不需要精通围棋知识。

任务目标 在本题中, 你将使用 MCTS 搜索得到井字棋的最优策略, 并且与随机策略和 AlphaBeta 搜索得到的策略相比较, 并尝试在 7×7 的围棋问题上运行 MCTS 算法。请完成以下内容, 根据要求使用和修改 `code` 路径下的代码, 提交你的代码和实验报告。**提交时请删除 `*.so`、`*/build/` 等临时文件, 仅提交代码, 本题的文字报告请和其他题目写在同一个文档中提交。**

1. 井字棋 (3×3 的三子棋) 中, 假设对局双方都绝对理性, 先手有必胜策略吗? 后手能保证不输吗? (直接回答)
2. 补充 `./mcts/uct_mcts.py` 中标注 `./code` 的内容, 实现 MCTS 算法。
3. 适当修改 `pit.py`, 在设定 $C = 1.0$, `n_rollout` = 7, `n_search` = 64 的情况下, 分别汇报井字棋游戏中, MCTS 算法对弈 Random 策略、MCTS 算法对弈 AlphaBeta 算法和 MCTS 算法对弈 MCTS, 在先手和后手情况下 (共 $3 \times 2 = 6$ 组) 的胜率和输率。
4. 取消 `./players/uct_player.py` 第 33 行处的注释, 对比在 $C = 0.1$, $C = 5.0$ 时 MCTS 输出的策略有何不同, 分析 C 的取值是如何影响 MCTS 输出的策略的。
5. 选取一组合适的 `n_rollout`、`n_search` 参数的取值, 在 7×7 的围棋游戏 中测试 MCTS 与 Random 策略对局先手和后手的胜率和输率, 并回答这两个参数对 MCTS 的搜索的速度和质量有何影响。
6. [附加题] (1pt) 选取上一问中的一局或几局对弈过程, 结合本作业中的围棋规则, 分析 MCTS 策略的每一步落子是否合理。
7. [附加题] (4pt) 当问题规模扩大时, 由于搜索树极大, AlphaBeta 搜索算法计算每一步落子都会花费极长的时间。为了解决这个问题, 我们做出一些“让步”: 当深度较浅时, 可以

通过搜索充分考虑对手的行为逻辑；但是当深度较大时，受时间限制，不能继续搜索。为了体现出不同后继的差异，我们可以对状态进行**评估**，用评估的分数代替真实搜索的结果。在 `other_algo/heuristic.py` 中我们提供了一个围棋的局面评估函数（分数越高，局面越好），请完成以下内容：

- (a) 补全 `other_algo/alpha_beta_search.py` 中 `AlphaBetaSearchWithHeuristic` 空缺的代码，实现利用启发式进行截断搜索的 AlphaBeta 剪枝搜索；
- (b) 类似的，我们也可以在 MCTS 中引入该评估函数替代或修正蒙特卡洛采样的评估结果，请修改 `mcts/uct_mcts.py` 实现一个带评估函数的 MCTS；
- (c) 最后，请利用 `pit.py` 测试 `RandomPlayer`、`AlphaBetaHeuristicPlayer` 和使用修改后 MCTS 的 `UCTPlayer` 三者棋力的强弱，并汇报结果。

提示：

- 1. **动手之前，请仔细阅读 `README.md`**，按照要求配置运行环境，并确保你已了解问题定义和代码框架。请尽量不要修改代码中未要求修改的部分，如确有必要请提交前与助教沟通。
- 2. 脚本 `pit.py` 中，`pit` 函数的 `log_output` 参数可以控制是否输出棋盘，对弈时使用 `HumanPlayer` 可以人工与算法对局。你可以利用这些进行调试。
- 3. 请确保你的代码拥有高可读性与可复用性，之后的作业会使用本次作业 MCTS 的部分代码。
- 4. 请仔细阅读 `README.md` 中对于游戏环境的描述，注意返回的 `reward` 代表的是**当前落子者**是否胜利，以及使用 `fork()` 方法复制棋盘。
- 5. 调试时，可以取消 `pit.py` 第 97 行处的注释固定随机数种子，方便复现 bug 或实验结果。
- 6. 如问题中没有进行指定，你可以自由选择合适的参数进行实验。
- 7. **推荐参考材料：**课件，以及 *A Survey of Monte Carlo Tree Search Methods*¹3.3 节。

4 理解 AlphaZero (5pt)

在标准的围棋棋盘大小 (19×19) 下，由于状态空间极大，本次作业中使用的基于 UCB (Upper Confidence Bound) 公式的 MCTS 无法在合理时间内得到高质量的策略。为了解决这个问题，AlphaZero 中，神经网络被嵌入 MCTS 以提高搜索的效率和质量。由于我们课程还未讲到神经网络，本次作业中的 MCTS 没有涉及到这部分内容。为了更好的了解大作业的整体结构，确保在后续作业中可以零成本复用本次作业的代码，请你简单阅读 AlphaGoZero 的论文²（其 MCTS 部分和 AlphaZero 的实现基本一致）和补充材料³，其中与本次作业内容较为相关的部分是 AlphaGoZero 论文的“Reinforcement learning in AlphaGo Zero”章节和补充材料第 14 至 15 页的内容。请在完成阅读后在实验报告中回答以下问题：

- 1. 为什么 AlphaZero 使用了 MCTS 而不是 AlphaBeta 作为搜索主干？
- 2. AlphaZero 中使用的基于 PUCT (Predictor + UCB Applied to Trees) 的 MCTS 与本次作业中基于 UCT (Upper Confidence Bounds Applied to Trees) 的 MCTS 有哪些区别？

[提示]：你也可以参考网上的其它资料。

¹<https://repository.essex.ac.uk/4117/1/MCTS-Survey.pdf>

²<https://www.nature.com/articles/nature24270>

³https://www.science.org/doi/suppl/10.1126/science.aar6404/suppl_file/aar6404-silver-sm.pdf

5 提交格式

- 请先删除 *.so *.png 文件和环境目录下的 build 文件夹，再将你的代码目录内**所有代码文件**和你的**文字报告**打包提交。统一文件的命名：{学号}_{姓名}_hw1.zip。
- 请将本次作业所有问题回答写在同一份报告中，报告请导出为 **pdf 格式**。