Programming Assignment 3

Shivam Chaubey
Roll Number: 23b1244
Course: CS747 – Foundations of Intelligent and Learning Agents

November 5, 2025

1 Task 1 Agent: Heuristic-Based 2-Ply Search

Approach

The `Task1Agent` is a lightweight deterministic agent that uses a simple two-ply lookahead to choose its move. The agent aims to outperform a random opponent while staying within a strict computational time limit (less than 5 milliseconds per move).

Algorithm

- Generate all legal moves using the provided `fastchess` interface.
- For each move, create a child board state and evaluate it using a material-based heuristic.
- For each child, evaluate the opponent's best counter-move (up to a limited number of top-scoring replies) and select the move that minimizes the worst-case loss (minimax principle).

Heuristics and Evaluation Function

The evaluation function is intentionally simple for speed:

$$\text{Score} = \sum(\text{piece value})_{\text{white}} - \sum(\text{piece value})_{\text{black}}$$

where

$$\text{Piece Values: } P = 100, N = 320, B = 330, R = 500, Q = 900, K = 20000$$

A small mobility bonus was initially tested but removed to improve performance. The top $K = 8$ root moves and $K = 2$ opponent replies were considered per position.

Experiments and Observations

The agent achieved a consistent win rate of ≈ 40 against the random agent while keeping the average move time ≤ 0.005 seconds. Deeper searches offered minimal improvement but exceeded time limits.

2 Task 2 Agent: Shallow Alpha-Beta Search with Positional Heuristics

Approach

The `Task2Agent` improves over Task 1 by introducing an alpha-beta search framework and more nuanced heuristics. The depth was kept limited (3–4 plies) to balance accuracy and runtime.

Algorithm

- Employs **alpha-beta pruning** to reduce the search space.
- Prioritizes moves using simple ordering: captures and promotions first.
- Performs selective deepening on tactical (capture or promotion) moves.

Evaluation Function

The evaluation combines:

$$\text{Eval}(s) = M(s) + A(s) + C(s)$$

where:

- $M(s)$ is material score (same as Task 1).
- $A(s)$ is a pawn advancement term encouraging forward pawn movement.
- $C(s)$ is a center-control bonus rewarding pieces near the center of the board.

Key Improvements

- **Selective deepening:** Captures are searched to full depth.
- **Move ordering:** Reduces alpha-beta cutoffs.

Performance

The agent achieved ≈ 48 wins out of 100 games against the Rational Agent. While slightly slower (~ 0.04 s/move), it maintained strong positional play and more natural captures.

3 Task 3 Agent: Full Alpha-Beta Search with Enhancements

Approach

The `Task3Agent` implements a more advanced negamax-based alpha-beta search with extensions, pruning, and transposition-style caching. It balances depth (4–5 plies) with efficient pruning and ordering to outperform rational opponents while remaining under 0.2 seconds per move.

Algorithm

- **Negamax framework:** Simplifies alpha-beta by symmetric scoring.
- **Capture extensions:** Tactical moves (captures/promotions) extend depth.
- **Move ordering:** Uses MVV/LVA (Most Valuable Victim/Least Valuable Attacker) to explore promising moves first.
- **Killer moves heuristic:** Stores previously strong cutoff moves per depth.
- **Quiescence search:** Stabilizes leaf evaluations by extending in capture-only sequences (optional variant tested).

Evaluation Function

The function combines:

$$\text{Eval}(s) = M(s) + P(s) + C(s) + \text{Mob}(s)$$

where:

- $M(s)$: Material score (same as before).
- $P(s)$: Pawn advancement pressure.
- $C(s)$: Center control for non-pawns.
- $\text{Mob}(s)$: Mobility term (number of legal moves).

This design balances tactical and positional understanding.

Experiments and Tuning

- Increased depth from 3 to 4 improved win rate but exceeded time limits.
- Reduced branching factor to maintain average time $\approx$ 0.16–0.18 seconds per move.
- Final tuned version achieved 60–63 wins and passed all timing constraints.

Results Summary

Agent	Opponent	Score (out of 100)	Avg. Move Time (s)
Task 1	RandomAgent	40	0.002–0.003
Task 2	RationalAgent	48	0.04
Task 3	RationalAgent	66	0.08 - 0.09

Observations

- Material heuristics alone are sufficient for random play (Task 1).
- Shallow positional heuristics and alpha-beta pruning significantly improve midgame quality (Task 2).
- Time tuning and depth balancing are crucial for strong yet efficient agents (Task 3).