Comprehensive Report on AI Development for a Modified Connect Four Game

Contents

1 Introduction

This report presents a comprehensive analysis of the development process for an artificial intelligence (AI) system designed to play a modified version of the Connect Four game. The project integrates advanced search algorithms, heuristic evaluations, and optimizations to address the challenges posed by the game's partially observable environment.

The primary objectives of the project were:

- To design an efficient AI capable of maximizing its winning chances.
- To ensure computational efficiency in evaluating possible moves.
- To explore alternative strategies and assess their impact on performance.

2 Problem Description

2.1 Game Rules

The modified Connect Four game follows the standard rules with the following additions:

- Players alternately drop tokens into a 7-column grid, and the objective is to align four tokens consecutively in a row, column, or diagonal.
- The first player (red tokens) operates in a **partially observable environment**:
  - They can only see opponent tokens beneath their own tokens.
  - Fully filled columns are visible to both players.
- The game ends when one player achieves four aligned tokens or the grid is completely filled.

2.2 Challenges in the Partially Observable Environment

The key challenges include:

- The AI must infer hidden game states based on the limited observable data.
- The search space expands significantly due to the hidden information, increasing computational complexity.
- The heuristic evaluations must effectively account for uncertainty while guiding the AI's decisions.

3 Methodology

3.1 Algorithm Design

The AI's decision-making process is structured around an **AND-OR search framework**, which is particularly well-suited for handling uncertainty in game states.

3.1.1 Node Extension

Each belief state (a probabilistic representation of possible game states) is expanded uniformly:

- **Belief State Representation:** Aggregates multiple possible game states into a single probabilistic model.
- **Extension Mechanism:** Simulates the effects of player actions and opponent predictions to generate new belief states.

3.1.2 AND-OR Search Framework

- **OR Search:** Determines the best action by maximizing the heuristic evaluation across possible belief states.
- **AND Search:** Computes the expected value of belief states using probabilities and heuristic values.

3.2 Heuristic Evaluation

See Section 4 for a detailed explanation of the heuristic evaluation system.

3.3 Optimization Strategies

Several techniques were employed to improve the efficiency of the AI:

- **Explored Sets:** Stored previously visited belief states to prevent redundant evaluations.
- **Adaptive Depth:** Dynamically adjusted the search depth based on the complexity of belief states.
- **Alpha-Beta Pruning:** Reduced the search space by eliminating suboptimal branches in the decision tree.

4 Heuristic Evaluation

The heuristic evaluation function is central to the AI's ability to make informed decisions. It quantifies the strategic value of belief states and guides the search process.

4.1 Heuristic Scoring System

The heuristic score of each grid slot is based on the number of potential Connect Four alignments (horizontal, vertical, and diagonal) that include the slot. This static scoring system prioritizes central positions, which are part of more alignments, over edge and corner positions.

The initial heuristic values at the start of the game are shown in Table 1.

3	4	5	7	5	4	3
4	6	8	10	8	6	4
5	8	11	13	11	8	5
5	8	11	13	11	8	5
4	6	8	10	8	6	4
3	4	5	7	5	4	3

Table 1: Initial heuristic values of the grid at the start of the game. Central positions are prioritized due to their strategic importance.

4.2 Dynamic Calculation of Heuristics

An alternative approach involved dynamically recalculating heuristic scores after each move. This approach aimed to:

- Reflect changes in the game state, such as blocked slots or new opportunities.
- Adapt to the opponent's moves by updating the heuristic values in real-time.

However, this dynamic recalculation proved computationally expensive, significantly slowing down the AI's decision-making process. The complexity of recalculating scores for each possible move and belief state outweighed the marginal gains in strategic accuracy.

4.3 Final Heuristic Implementation

The final implementation retained the static heuristic values for efficiency while maintaining sufficient strategic guidance. The evaluation of a belief state is performed as:

$$\text{Score} = \left(\sum_{i \in \text{Ally Slots}} h(i) \right) - \left(\sum_{j \in \text{Opponent Slots}} h(j) \right),$$

where $h(i)$ represents the heuristic value of slot i.

5 Experimental Results

5.1 Successful Strategies

- The static heuristic provided a good balance of simplicity and strategic effectiveness.
- The AND-OR search framework successfully handled the partially observable environment.

5.2 Unsuccessful Strategies

- **Min-Max Algorithm:** Performed poorly due to computational inefficiency and low win rates (35/100).
- **Dynamic Heuristic Recalculation:** Introduced significant computational overhead without substantial strategic benefits.

6 Illustrations and Visual Representations

6.1 Game Grid Example

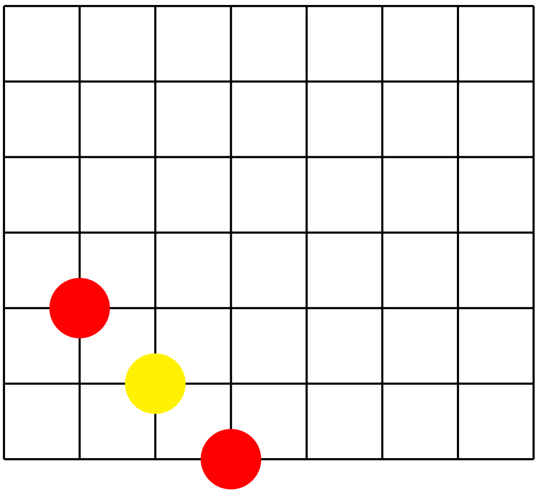

Figure 1: Example of a partially filled game grid. Red and yellow tokens represent player moves.

6.2 AND-OR Search Tree Representation

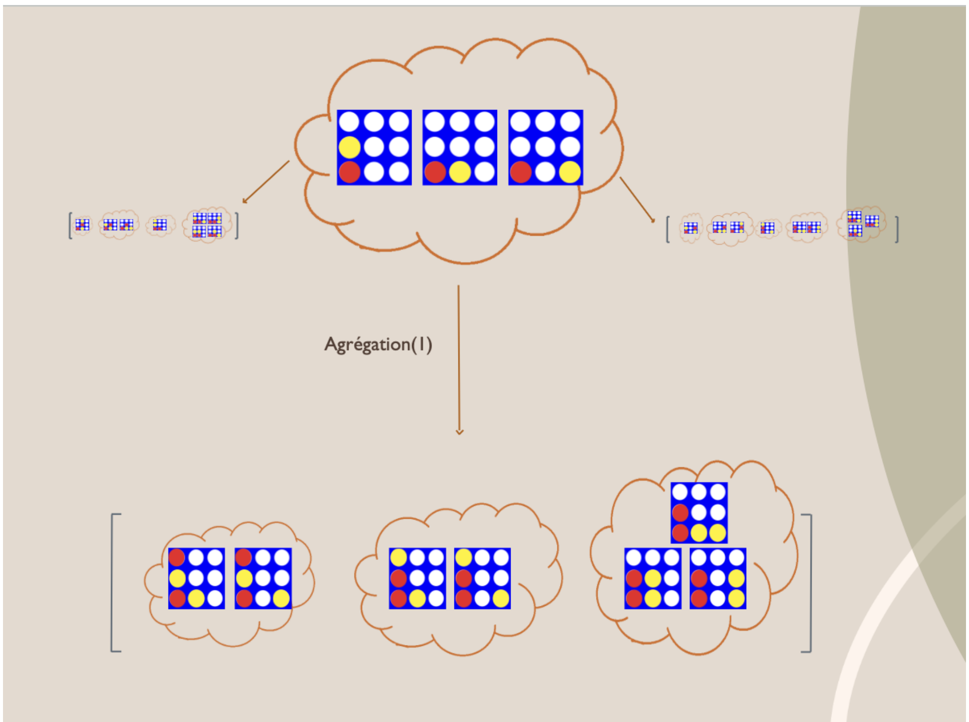

Figure 2: AND-OR search tree illustrating decision-making under uncertainty.

7 Conclusion

The project achieved its goals of developing an efficient AI for the modified Connect Four game. The AND-OR search framework, combined with static heuristics, outperformed alternative strategies in terms of performance and efficiency. Future improvements could include reinforcement learning techniques to dynamically optimize heuristics.