Ecole Publique d'ingénieures et d'ingénieurs en 3 ans

Rapport

Q-LEARNING IN WUMPUS WORLD GAME

le 6 janvier 2025,
version 1.1

Luciana Adrião dos Santos et
Vinicius Moreira Nascimento
Développeurs

Tuteur école : Régis Clouard
Tuteur école : Loïc SIMON

www.ensicaen.fr

TABLE DES MATIERES

TABLE DES FIGURES

TABLE DES TABLEAUX

Q-LEARNING IN WUMPUS WORLD

The Wumpus World is a turn-based logical game set in a grid where a player agent must find gold and escape without falling into pits or being caught by the Wumpus, a dangerous creature. The agent receives percepts like breeze (near pits) or stench (near the Wumpus) to make decisions, relying on reasoning to navigate the environment, avoid hazards, and achieve its goal.

1. Formulation of the Problem

The objective of the Rational Agent in the Wumpus World is to navigate a hostile environment consisting of pits, walls, and a Wumpus. The agent's goal is to locate the gold, retrieve it, and return to the starting position ([1,1]) with the maximum possible score. The agent must make decisions based on limited percepts ({Stench, Breeze, Glitter, Bump, Scream}) and navigate the grid safely while minimizing penalties. The Wumpus World Game can be classified as follows:

- **Perfect Information**: **No**

The agent has **partial observability**, as it can only perceive its immediate surroundings. The complete state of the environment (e.g., locations of the Wumpus, pits, and gold) is initially unknown.

- **Deterministic**: **No**

The game has elements of **uncertainty**, as outcomes depend on unknown factors (e.g., the positions of hazards). However, the agent's actions have deterministic effects once the environment is fully known.

- **Zero-Sum**: **No**

The Wumpus World is not inherently competitive or adversarial. It is a single-player game focused on **optimization** of the agent's actions to maximize score rather than directly opposing another player's goals.

- **Asynchronous**: **No**

The game is **synchronous**, as it operates in a turn-based manner where the agent executes discrete actions sequentially.

The Wumpus World is a **partially observable, stochastic, non-zero-sum, synchronous environment**, making it a challenging problem for developing reasoning and decision-making strategies in artificial intelligence

1.1. Challenges Identified

Implementing a reinforcement learning algorithm in an unknown environment with an undefined number of variables, such as the positions of obstacles, poses significant challenges. The algorithm must achieve optimal convergence while enabling the agent to navigate back to the starting point after successfully escaping the maze.

1.2. Objectives

Develop an agent that learns from the environment using the reinforcement learning technique (Q-Learning), capable of efficiently retrieving the gold and safely exiting the cave. The agent should minimize penalties, maximize the final score, and demonstrate robustness to varying grid sizes and random placements of hazards.

2. Solution with Its Justification

We decided to use an agent capable of learning from the environment through the Q-Learning algorithm, as introduced in class. This choice is based on the suitability of Q-Learning for the problem, given that the grid size provides a state space and possible actions that result in a Q-Table of manageable size. The main goal is for the agent to make its own decisions within the maze, applying the concepts studied in class. Additionally, we can compare it with other reinforcement learning solutions, as shown in the table below:

ALGORITHM	BRIEF EXPLANATION	CONCLUSION
Q-LEARNING	A model-free reinforcement learning algorithm that uses a Q-Table to estimate the utility of state-action pairs	**Chosen** for its simplicity and efficiency in handling discrete state and action spaces.

DEEP Q-LEARNING	An extension of Q-Learning that uses neural networks to approximate the Q-Table for large state spaces	**Not used** due to unnecessary complexity and computational demands for the relatively small grid size.
SARSA	Similar to Q-Learning but updates the Q-Table based on the agent's actual policy instead of the maximum reward.	**Not used** because it requires policy-specific updates, which are less exploratory compared to Q-Learning. Not studied to.
REFLEXIVE AGENT	A simple agent that selects actions based solely on current percepts, without maintaining a memory of past states.	**Not used** because it cannot handle partial observability or make informed decisions in complex environments like the Wumpus World.
RATIONAL AGENT	An agent that takes actions to maximize expected performance based on current knowledge and reasoning.	**Not chosen** as it requires complete or highly accurate knowledge of the environment.

Tableau 1 Comparison between possible solutions

2.1. Implementation Details

2.1.1. Learning Phase

The agent uses Q-Learning to learn the optimal policy for navigating the grid. Key parameters include:

- Learning rate (α): 0.9 – Determines the step size in the Q-Value updates, ensuring faster learning.
- Discount factor (γ): 0.975 – Ensures the agent values future rewards significantly, enabling long-term planning.
- Exploration rate (ε): 0.1 – Encourages exploration of the environment with a decay factor (*epsilon_decay* = 0.95) to shift towards exploitation as learning progresses.

The implementation employs a Q-Table of dimensions (gridSize, gridSize, 4, 6)), representing positions, directions, and six possible actions. This structure supports discrete state-action mapping, ensuring efficient training and decision-making.

2.1.2. Decision-Making

- Percept Analysis: The agent processes percepts such as stench, breeze, and glitter to infer environmental states and update its understanding dynamically.

- Action Selection: Actions are chosen using an $\varepsilon - \text{greedy}$ strategy. During exploration, actions are selected randomly, while during exploitation, the Q-Table guides the decision based on the highest reward.

- Reward System: Cells are evaluated for rewards based on their type (e.g., +1000 for gold, -1000 for pits or the Wumpus). Safety and navigation towards the starting position are also prioritized when gold is found.

2.1.3. Safety Mechanisms

For avoiding Hazards: The agent uses a list of dangerous cells (e.g., containing the Wumpus or pits) to prevent revisiting perilous areas. And for exploring Safe Cells one heuristic prioritizes safe and unexplored cells to optimize navigation, especially during the return to the starting position.

3. Quantitative Analysis of the Solution

3.1. Execution Statistics

- **Training Episodes**: 60
- **Average Score**: ~72.5 (calculated from observed scores with high variability)
- **Success Rate**: 66.7% (based on the results printed in screen).
- **Grid Sizes Tested**: 10x10
- **Hazard Configurations**: Randomized pits and Wumpus placement.

3.2. Time Complexity

The Q-Learning algorithm's time complexity is $O(episodes \times steps)$, where $episodes = 60$ and steps varies per episode. The complexity increases significantly with grid size due to the agent's exploration.

3.3. Space Complexity

The Q-Table's space complexity is $O(gridSize^2 \times actions)$, specifically $10 \times 10 \times 4 \times 6 = 2400$ entries for this configuration. This makes it memory efficient for small grids but infeasible for larger environments.

4. Criticism of the Solution

4.1. Strengths

The Q-Learning agent can learn and adapt to the environment with consistent improvement over episodes. The use of rewards and penalties encourages the agent to develop strategies for safe exploration and goal achievement. The implementation is relatively straightforward and computationally feasible for smaller grids.

4.2. Weaknesses

- **Inefficient Learning**: The agent demonstrates slow convergence, often failing to optimize its strategy fully within the training episodes.
- **Behavioral Bias**: The agent tends to overfit specific actions, such as consistently turning *right*, leading to "addictive" behavior. Additionally, after retrieving the gold, it sometimes irrationally attempts to climb in any situation, causing unnecessary failures.
- **Limited Exploration**: The exploration strategy $(\varepsilon - \text{greedy})$ can result in the agent failing to adequately balance exploration and exploitation, especially in complex environments.

4.3. Possibilities for Improvement

Transitioning to a **Rational Agent** may provide a more robust solution. According to JavaTPoint, logical equations can accurately determine the locations of hazards

(pits and Wumpus), resulting in more precise decision-making. Although this would increase the code's complexity, it would improve efficiency and reliability.

Also, enhanced training strategy, to consider using adaptive exploration strategies, such as Boltzmann exploration (not studied) or dynamic $\varepsilon - \text{greedy}$ (not studied), to reduce behavioral bias and enhance learning.

Another possibility is using one hybrid approach, that means combine Q-Learning with heuristic rules to improve performance in partially observable environments, ensuring the agent avoids irrational actions post-goal retrieval.

This analysis suggests that while Q-Learning is effective for basic learning and adaptation, a Rational Agent using logical reasoning would significantly improve overall performance and reliability in the Wumpus World environment.

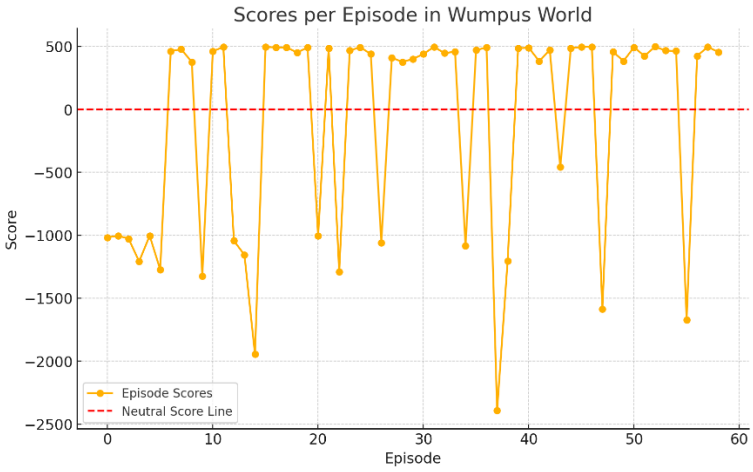

Figure 1 Scores per Episode in Wumpus Worl

ENSI CAEN

ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

Ecole Publique d'ingénieures et d'ingénieurs en 3 ans

6 boulevard Maréchal Juin, CS 45053
14050 CAEN cedex 04