CSAI 301- Fall 24

Project Description

Phase I: Application of Search Algorithms to an AI Problem (45%)

Due: Week 8

Objective:

In this phase, your goal is to apply various search algorithms to an AI problem of your choice. You will model the problem, implement appropriate search techniques, and compare their performance.

Tasks:

1. **Problem Modeling**:
    - **State Space Representation**: Define the state space based on your chosen problem.
    - **Initial and Goal States**: Specify the initial state and the goal state.
    - **Actions**: Define the possible actions within the problem environment.
    - **Transition function**: Define how new states are generated when actions are applied.
    - **Estimate the problem size**: Your selected problem should be of medium size (state space size is from 1K to 10 K) to make sure your selected problem is **sufficiently complex to apply different algorithms** (e.g., search algorithms, reinforcement learning) while ensuring you can still fully analyze and compare the results within reasonable resources.
2. **Modeling Assumptions**:
    - Clearly state any assumptions made regarding the environment, constraints, or dynamics.
3. **Search Algorithms (Uninformed Search)**:
    - Implement and compare **BFS**, **DFS**, **UCS** and **IDS** in solving the problem.
    - Analyze time complexity, memory usage, completeness, and optimality.
4. **Heuristic Search**:
    - Define two heuristic functions suitable for the problem.
    - Implement and compare **Greedy Best-First Search** and **A*** using these heuristics.
5. **Local Search**:
    - Implement **Hill Climbing**, **Simulated Annealing,** and **genetic algorithms** to find the solution.
6. **Comparison of Methods**:

   - Compare the performance of all algorithms based on criteria such as time and space taken, optimality of solution, and completeness.

7. **Visualization**:
   - Present graphical representations or search trees for each algorithm to illustrate explored states and the final path/solution.

Phase I Deliverables:

1. **Video Demo**: A 3-minute video demonstrating your problem modeling, approach, and progress.
2. **Report**: A concise report summarizing the problem, the algorithms used, their performance, and analysis and reflections on the results.
3. **Code Submission**: Commented and well-structured code for all implemented algorithms.

Phase II: Application of Reinforcement Learning (50%)

Due: Week 13

Objective:

In this phase, you will model the same AI problem as a reinforcement learning (RL) task and apply **Q-learning** to optimize the solution.

Tasks:

1. **State Space Representation**:
   - Define the state space for the problem and identify initial and goal states.
2. **Action Set**:
   - Specify the possible actions within the problem domain.
3. **Reward Function**:
   - Design a reward function that incentivizes the agent to reach the goal efficiently.
4. **Q-Learning Setup**:
   - Implement **Q-learning** with an appropriate learning rate and discount factor.
   - Initialize all Q-values and define exploration-exploitation strategies.
5. **Training the Agent**:
   - Train the agent over a specified number of steps, allowing it to learn through interactions with the environment.
   - Allow the agent to explore the environment and present how the results have been changed.

6. **Performance Monitoring**:
    - Present the Q-values midway through the training and at the end.
    - Analyze the agent’s behavior, movement patterns, and the policy it learned.

Phase II Deliverables:

1. **Video Demo**: A 3-minute video showcasing the agent’s learning process and final results.
2. **Report**: A detailed report describing your design, Q-learning setup, results, and analysis.
3. **Code Submission**: Well-commented and organized code implementing Q-learning.

Grading Criteria:

Phase I (45%):

- **Problem Modeling (10 points)**: Clearly defines the state space, actions, and assumptions.
- **Search Algorithm Implementation (15 points)**: Correct implementation of BFS, DFS, UCS, IDS, heuristic-based algorithms, and other search algorithms.
- **Heuristic Design (5 points)**: Well-constructed heuristics that improve search performance.
- **Comparison & Analysis (3 points)**: Comprehensive analysis of time complexity, optimality, and other factors.
- **Visualization (3 points)**: Clear and intuitive visual representation of the search process and solution.
- **Report Quality (3 points)**: Concise, well-organized explanation of problem, algorithms, and results.
- **Video (3 points):** Clear 3-min presentation with clear takeaways messages.
- **Code quality (3 points):** Well-commented, organized, and running code.

Phase II (50%):

- **State Space Representation (6 points)**: Comprehensive definition of states and actions.
- **Q-Learning Setup (15 points)**: Correct and efficient implementation of Q-learning.
- **Q-Table & Result Analysis (10 points)**: Presentation of Q-values and clear analysis of results.
- **Performance Evaluation (10 points)**: Evaluation of the agent’s policy and learning performance.
- **Report Quality (3 points)**: Organized and insightful report on design choices, results, and conclusions.
- **Video (3 points):** Clear 3-min presentation with clear takeaways messages.

- **Code quality (3 points):** Well-commented, organized, and running code.

*****Oral Discussions & Code Understanding (applicable for each phase):**

- **Code Explanation**: Ability to explain every part of the code **(what if questions)** and the theoretical underpinnings of the algorithms.

- **Theory Discussion**: Depth of knowledge on search algorithms and reinforcement learning concepts.

- **Each item above will be evaluated during the discussion as follows**: Unable to answer questions (-10%), Able to answer some questions (-5%), Able to answer all questions correctly (-0%).

- Failing to answer questions during the discussion in both parts will result in losing 20% of the corresponding phase grade.

Suggested Topics:

- Autonomous Vehicle Navigation
- Robot Path Planning
- Game Strategy Optimization
- Network Traffic Management
- Task Scheduling
- Personalized Recommendation System
- Dynamic Resource Allocation
- AI-Based Medical Diagnostics
- Autonomous Drone Navigation
- Warehouse Robot Coordination

Late Submission Policy:

- Late submissions will be penalized with 25% of the grade deducted for each day late. No submissions will be accepted after two days.

Plagiarism:

- Any form of plagiarism will result in a failing grade for the project and may lead to further disciplinary action.