Problem Solving and Search in Artificial Intelligence (Week 3)

1. Problem-Solving Agents

- **Definition**: A restricted form of a general agent that can solve problems by formulating goals and searching for solutions.
- **Components**:
  - **State**: Description of the current world state.
  - **Goal**: Desired outcome the agent aims to achieve.
  - **Problem**: Formulation of the problem based on the current state and goal.

Key Concepts

- **Initial State**: The starting point of the agent.
- **Actions**: Possible operations the agent can perform.
- **Goal Test**: Determines if the current state meets the goal.
- **Path Cost**: The cost associated with the path taken to reach the goal.

Example Problem

- **Scenario**: Traveling from Arad to Bucharest.
  - **States**: Various cities.
  - **Actions**: Driving between cities.
  - **Solution**: Sequence of cities (e.g., Arad → Sibiu → Fagaras → Bucharest).

2. Problem Types

- **Deterministic, Fully Observable**: Single-state problem where the agent knows its exact state.
- **Non-Observable**: Sensorless problem where the agent has no idea of its current state.
- **Nondeterministic/Partially Observable**: Contingency problem where new information is provided during execution.
- **Unknown State Space**: Exploration problem where the agent must discover the state space.

Examples

- **Vacuum World**:
  - **Single-State**: Start in a known location.
  - **Sensorless**: Start in an unknown location.
  - **Contingency**: Actions may have unpredictable outcomes.

3. Search Algorithms

Uninformed Search Strategies

- **Breadth-First Search (BFS)**:
  - **Strategy**: Expand the shallowest unexpanded node.
  - **Properties**:
    - Complete: Yes (if branching factor is finite).
    - Time Complexity: $O(b^{(d+1)})$.
    - Space Complexity: $O(b^{(d+1)})$.
    - Optimal: Yes (if all step costs are equal).
- **Depth-First Search (DFS)**:
  - **Strategy**: Expand the deepest unexpanded node.
  - **Properties**:
    - Complete: No (can fail in infinite-depth spaces).
    - Time Complexity: $O(b^m)$.
    - Space Complexity: $O(bm)$.
    - Optimal: No.
- **Uniform-Cost Search**:
  - **Strategy**: Expand the least-cost unexpanded node.
  - **Properties**:
    - Complete: Yes (if step cost is greater than zero).
    - Time Complexity: $O(b^{(C^*/\varepsilon)})$.
    - Space Complexity: $O(b^{(C^*/\varepsilon)})$.
    - Optimal: Yes.
- **Iterative Deepening Search**:
  - **Strategy**: Combines depth-first and breadth-first search.
  - **Properties**:
    - Complete: Yes.
    - Time Complexity: $O(b^d)$.
    - Space Complexity: $O(bd)$.
    - Optimal: Yes (if step cost = 1).

Informed Search Strategies

- **Best-First Search**:
  - **Evaluation Function**: Uses a heuristic to estimate the desirability of nodes.
- _A Search*_:
  - **Evaluation Function**: $f(n) = g(n) + h(n)$, where:
    - $g(n)$: Cost to reach node n.
    - $h(n)$: Estimated cost to reach the goal from n.
  - **Properties**:
    - Complete: Yes (unless there are infinitely many nodes).
    - Time Complexity: Exponential in the relative error of h.

- Space Complexity: Keeps all nodes in memory.
- Optimal: Yes.

4. Heuristics

- **Definition**: Functions that estimate the cost of the cheapest path from a node to the goal.
- **Types**:
  - **Admissible Heuristics**: Never overestimate the true cost.
  - **Consistent Heuristics**: Satisfy the triangle inequality.

Examples of Heuristics

- **8-Puzzle**:
  - h1(n): Number of misplaced tiles.
  - h2(n): Total Manhattan distance.

5. Summary of Algorithms

Algorithm	Complete	Time Complexity	Space Complexity	Optimal
Breadth-First Search	Yes	O(b^(d+1))	O(b^(d+1))	Yes
Uniform-Cost Search	Yes	O(b^(C*/ε))	O(b^(C*/ε))	Yes
Depth-First Search	No	O(b^m)	O(bm)	No
Iterative Deepening	Yes	O(b^d)	O(bd)	Yes
A* Search	Yes	Exponential in [relative error in h × length of solution]	Keeps all nodes in memory	Yes

6. Questions and Answers for Exam Preparation

Q1: What is a problem-solving agent?

- **A1**: A problem-solving agent is a type of agent that can formulate goals and search for solutions based on its current state and the actions available to it.

Q2: What are the main types of problems in AI?

- **A2**: The main types of problems include:
  - Deterministic, fully observable (single-state problems).
  - Non-observable (sensorless problems).
  - Nondeterministic/partially observable (contingency problems).
  - Unknown state space (exploration problems).

Q3: Describe the breadth-first search algorithm.

- **A3**: Breadth-first search (BFS) expands the shallowest unexpanded node first. It is complete if the branching factor is finite, has a time complexity of O(b^(d+1)), and is optimal if all step costs are equal.

*Q4: What is the A search algorithm?**

- **A4**: A* search is an informed search algorithm that uses an evaluation function f(n) = g(n) + h(n), where g(n) is the cost to reach node n and h(n) is the estimated cost to reach the goal. It is complete, optimal, and uses memory to keep all nodes.

Q5: What are heuristics in the context of search algorithms?

- **A5**: Heuristics are functions that estimate the cost of the cheapest path from a node to the goal. They help guide the search process and can significantly reduce the search space.

Q6: What is the difference between admissible and consistent heuristics?

- **A6**: Admissible heuristics never overestimate the true cost to reach the goal, while consistent heuristics satisfy the triangle inequality, ensuring that the estimated cost is always less than or equal to the cost of reaching a neighboring node plus the estimated cost from that neighbor to the goal.

Q7: Why is iterative deepening search preferred in some scenarios?

- **A7**: Iterative deepening search combines the benefits of depth-first and breadth-first search, providing completeness and optimality while using only linear space, making it efficient for large search spaces.

These notes and questions should provide a comprehensive overview for exam preparation on the topics of problem-solving and search algorithms in artificial intelligence.

Problem Solving in Artificial Intelligence (Week 4)

1. Problem Solving: Knowledge Representation and Inference

- **Key Concepts:**
  - Problem-solving in AI involves representing knowledge and using inference to derive solutions.
  - Search strategies are crucial for navigating through problem spaces.

2. Search Strategies

2.1 Best-First Search

- **Definition:** A search strategy that uses an evaluation function (f(n)) to determine the desirability of nodes.
- **Implementation:** Nodes are ordered in the fringe based on their desirability.
- **Special Cases:**
  - **Greedy Best-First Search:** Uses only the heuristic (h(n)).
  - *A Search:** Combines cost to reach the node (g(n)) and heuristic (h(n)).

2.2 Greedy Best-First Search

- **Evaluation Function:** (f(n) = h(n))
- **Properties:**
  - **Completeness:** No, can get stuck in loops.
  - **Time Complexity:** (O(b^m))
  - **Space Complexity:** (O(b^m))
  - **Optimality:** No.

*2.3 A Search**

- **Evaluation Function:** (f(n) = g(n) + h(n))
  - (g(n)): Cost to reach node (n)
  - (h(n)): Estimated cost from (n) to goal
- **Properties:**
  - **Completeness:** Yes (unless infinite nodes exist).
  - **Time Complexity:** Exponential.
  - **Space Complexity:** Keeps all nodes in memory.
  - **Optimality:** Yes, if (h(n)) is admissible.

2.4 Heuristics

- **Admissible Heuristic:** $h(n) \leq h^*(n)$ (never overestimates).
- **Consistent Heuristic:** $h(n) \leq c(n, a, n') + h(n')$ for every successor n'.

3. Local Search Algorithms

3.1 Hill-Climbing

- **Description:** Iteratively moves to the neighbor with the highest value.
- **Challenges:** Can get stuck in local maxima.
- **Random-Restart Hill Climbing:** Overcomes local maxima by restarting the search.

3.2 Simulated Annealing

- **Idea:** Allows "bad" moves to escape local maxima, gradually decreasing their frequency.
- **Properties:** If temperature decreases slowly, it can find a global optimum.

3.3 Genetic Algorithms

- **Description:** Combines pairs of states to generate successors.
- **Process:**
  - Selection, crossover, and mutation.
  - Fitness function evaluates the quality of states.

4. Examples and Applications

4.1 Example: Romania Problem

- **Goal:** Travel from Arad to Bucharest.
- **States:** Various cities.
- **Actions:** Driving between cities.

4.2 Example: n-Queens Problem

- **Goal:** Place n queens on an $n \times n$ board without conflicts.
- **Local Search:** Move queens to reduce conflicts.

5. Continuous State Spaces

- **Description:** Optimization problems in continuous spaces, e.g., airport placement.
- **Methods:** Gradient descent and Newton-Raphson for optimization.

Questions and Answers for Exam Preparation

*Q1: What is the difference between Greedy Best-First Search and A Search?**

- **A1:** Greedy Best-First Search uses only the heuristic (h(n)) for node evaluation, while A* Search uses both the cost to reach the node (g(n)) and the heuristic (h(n)) to evaluate nodes.

*Q2: What are the properties of A Search?**

- **A2:**
  - Complete: Yes (unless infinite nodes exist).
  - Time Complexity: Exponential.
  - Space Complexity: Keeps all nodes in memory.
  - Optimal: Yes, if the heuristic is admissible.

Q3: Explain the concept of admissible heuristics.

- **A3:** An admissible heuristic is one that never overestimates the cost to reach the goal from any node (n). It ensures that the search algorithm can find the optimal solution.

Q4: What is the main challenge of Hill-Climbing search?

- **A4:** The main challenge is that it can get stuck in local maxima, where no neighboring state has a higher value, preventing it from finding the global maximum.

Q5: How does Simulated Annealing differ from Hill-Climbing?

- **A5:** Simulated Annealing allows for "bad" moves to escape local maxima, while