CS 3642 — Artificial Intelligence

Reading and Outline Notes Assignment 2

Name: James Widner
ID: 001121770
Date: September 17, 2025

Summary of Chapter 3.4: Uninformed Search Strategies

Uninformed search strategies, also called blind search, are algorithms that systematically explore a problem space without domain-specific heuristics. They rely only on the initial state, the set of possible actions, and the goal test. Although these strategies can be inefficient, they form the foundation of search in Artificial Intelligence.

Key Strategies

- **Breadth-First Search (BFS):** Explores nodes level by level, guaranteeing the shallowest solution. Complete and optimal when costs are uniform, but memory use grows exponentially.
- **Uniform-Cost Search (UCS):** Expands the node with the lowest path cost. Guarantees an optimal solution for variable costs, making it practical for route-finding.
- **Depth-First Search (DFS):** Explores as deep as possible before backtracking. Requires little memory but risks infinite descent and does not guarantee optimality.
- **Depth-Limited Search (DLS):** DFS with a fixed cutoff depth. Prevents infinite loops but can cut off solutions deeper than the limit.
- **Iterative Deepening Search (IDS):** Applies DLS repeatedly with increasing limits. Balances the memory efficiency of DFS with the completeness of BFS.
- **Bidirectional Search:** Runs two searches simultaneously (from start and goal) and meets in the middle. Can drastically reduce time complexity when the goal is well-defined.

Practical Example

Google Maps provides an intuitive analogy. Uniform-cost search reflects how Maps finds the shortest or fastest route, while bidirectional search can improve efficiency by searching from both the start and destination. However, these systems usually employ informed search with heuristics (like traffic or speed limits) layered on top of the uninformed foundations.

Reflections and Questions

While uninformed search provides completeness and guarantees in certain cases, it does not capture subjective or context-sensitive goals. This raises interesting questions:

1. Does Google Maps actually use bidirectional search in practice to optimize route computation?
2. Could we design a special navigation app that prioritizes scenic or enjoyable routes (e.g., backroads, fewer highways) instead of just minimizing time or distance?

3. How might qualities like “scenic value” be modeled as part of a cost function so that uniform-cost search could be adapted to optimize for enjoyment rather than efficiency?