INFO-F-311: Artificial Intelligence - Project 1: Search

Your Name

1 Preamble

This report outlines the implementation of artificial intelligence techniques based on graph search techniques like Breadth-First Search (BFS), Depth-First Search (DFS), and the A* algorithm.

The primary languages and tools used are Python 3.10 and Poetry.

1.1 The Problems

1. **SimpleSearchProblem**: The goal is to reach the exit of the environment with multiple agents.
2. **CornerSearchProblem**: The agents must pass through all four corners of the environment before reaching an exit.
3. **GemSearchProblem**: The agents need to collect all gems in the environment before reaching an exit.

2 SimpleSearchProblem

2.1 Problem Modeling

This section describes the is_goal_state and get_successors methods.

2.2 Breadth-First Search

The Breadth-First Search algorithm is implemented in `search.py` via the `bfs` function.

2.3 Depth-First Search

The Depth-First Search algorithm is implemented in `search.py` via the `dfs` function.

2.4 A* Search

The A* algorithm is implemented with Manhattan distance as the heuristic.

3 CornerSearchProblem

3.1 Problem Modeling

The problem aims to pass through all four corners of the grid.

3.2 Heuristic

A consistent heuristic more efficient than Manhattan distance is developed.

4 GemSearchProblem

4.1 Problem Modeling

The problem aims to collect all gems in the environment.

4.2 Heuristic

A consistent heuristic more efficient than Manhattan distance is developed.

5 Understanding the Code

The codebase includes utility functions for calculating distances, an abstract SearchProblem class, and concrete problem classes like SimpleSearchProblem, CornerSearchProblem, and GemSearchProblem.

5.1 Key Methods

- `is_goal_state()`: Determines if a state is the goal state.
- `get_successors()`: Generates possible successor states from the current state.
- `heuristic()`: Calculates the heuristic value for the A* algorithm.

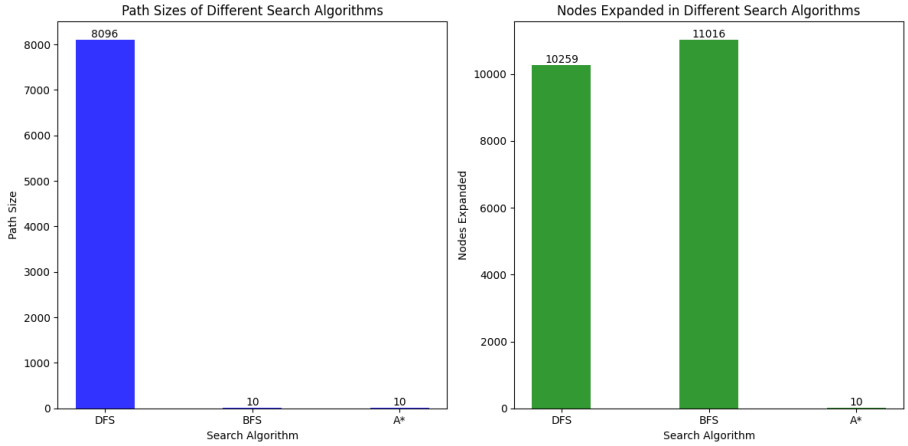
6 Results

6.1 Path Size Comparison

Comparison of the path sizes found by BFS, DFS, and A* on level 3.

6.2 Node Expansion Comparison

Comparison of the number of nodes expanded in BFS, DSF, and A* during the solution of level 3.



Algorithm	Path Size	Nodes Expanded
DFS	8096	10259
BFS	10	11016
A*	10	10

6.3 Discussion

- **Efficiency**: A* is the most efficient in terms of both path size and nodes expanded.
- **Optimality**: BFS and A* find the optimal path, while DFS does not.
- **Resource Usage**: DFS and BFS are resource-intensive, whereas A* is memory-efficient.
- **Applicability**: A* is often the best choice when an admissible heuristic is available.
- **Path Quality**: DFS finds a path but not necessarily the shortest one.
- **Deterministic Outcome**: A* and BFS provide a more predictable and optimal outcome compared to DFS.

7 Heuristics Development

heuristic for CornerSearchProblem and GemSearchProblem.

8 Optimizations

- Use a priority queue for the A* algorithm.
- Cache heuristic values to avoid redundant calculations.

9 Tool Usage

Explanation of the use of tools like ChatGPT in the project.

9.1 Initial Phase: Data Dump

In the initial stage, I fed ChatGPT with as much contextual information as possible.

- INFO-F-311$_Pproject11 - recherche - v3.zip$

- Project challenges
- Project resources
- Project timeline

  This process is analogous to brainstorming with another person but in a more structured and data-driven manner.

9.1.1 Memory

Each time new

9.2 Fine-Tuning Perception

After the initial data input, I fine-tuned how ChatGPT interprets the provided information. This involved adjusting the focus towards specific problem statements and goals. The objective was to align the AI's understanding with the project requirements.

9.3 Filtering Output

ChatGPT's output also underwent a filtration process to distill the most useful and relevant suggestions or solutions. This step ensured that the AI's contributions were directly applicable to the project.

9.4 Comparative Analysis

While ChatGPT was processing data and generating insights, I also developed my own ideas and solutions. This dual approach allowed for a comprehensive comparative analysis, which often led to more refined and effective strategies for tackling the project's challenges.

9.5 Synergistic Collaboration

The combination of human intuition and AI-generated insights created a synergy that enriched the project's development process. It was like having an extra team member who could offer a different perspective, thereby broadening the scope and enhancing the quality of the project.

9.6 Advantages & Limitations

The primary advantage of using ChatGPT was the acceleration of the ideation and problem-solving phases. However, it's essential to note that while the tool provided rapid insights, the quality and applicability of these insights still required human judgment for effective implementation.