Slide 1: title page

Slide 2: Introduction

Objective:

- Explore a range of search algorithms for maze navigation.

- Understand their methodologies, strengths, and weaknesses.

Significance:

- Insight into diverse search strategies' relevance.

- Crucial role in problem-solving across domains.

- Adaptability to solve complex real-world problems.

Slide 3: List of Algorithms

1. \*\*Algorithms Used:\*\* Seven distinct algorithms utilized for maze navigation.

2. \*\*Differentiation:\*\* Categorized into informed (heuristic-based) and uninformed (blind search).

3. \*\*Approaches to Maze Navigation:\*\*

- \*\*BFS:\*\* Systematic exploration, layer by layer.

- \*\*DFS:\*\* Delves deeply into branches before backtracking.

- \*\*UCS:\*\* Considers path cost for each node.

- \*\*Greedy:\*\* Prioritizes nodes based on heuristic estimates.

- \*\*A\*:\*\* Balances actual and estimated costs for optimal paths.

- \*\*Weighted A\*:\*\* Introduces weighted heuristic for suboptimal paths.

- \*\*Beam:\*\* Explores multiple paths with limited breadth, sacrifices optimality.

Slide 4: Breadth-First Search (BFS)

- \*\*Overview:\*\* BFS is a systematic exploration algorithm for maze navigation.

- \*\*Systematic Exploration:\*\* Iteratively explores all adjacent nodes layer by layer from the starting point.

- \*\*Advantages:\*\* Optimal solution for identical cost actions due to its nature of searching layer by layer.

- \*\*Limitations:\*\* Exhibits high time and space complexity (O(b^d)) which might be impractical for large search spaces.

- \*\*Maze Examples:\*\* Visual representations of paths discovered by BFS in various maze scenarios.

Slide 5: Depth-First Search (DFS)

- \*\*Overview:\*\* DFS explores deeply into a single branch before backtracking.

- \*\*Strategy:\*\* Emphasizes deep exploration, prioritizing the deepest unexplored node.

- \*\*Completeness:\*\* Completeness relies on checking for cycles in finite state spaces; not complete in infinite state spaces.

- \*\*Limitations:\*\* Non-optimal and exhibits high space complexity (O(bd)).

- \*\*Maze Examples:\*\* Visual depiction of the extended paths taken by DFS in different maze scenarios.

Slide 6: Uniform Cost Search (UCS)

- \*\*Overview:\*\* UCS prioritizes nodes based on path cost.

- \*\*Completeness:\*\* Complete when no action costs less than a set limit to prevent infinite paths.

- \*\*Optimality:\*\* Ensures optimal solutions due to considering minimum-cost paths.

- \*\*Complexity:\*\* Discusses time and space complexity implications (O(b^((C\*)/ϵ))).

- \*\*Maze Examples:\*\* Displays optimal paths found by UCS in various maze configurations.

Slide 7-8: Greedy First Search (Manhattan & Euclidean Heuristics)

- \*\*Overview:\*\* Discuss heuristic-driven exploration, favoring immediate gains.

- \*\*Completeness:\*\* Completeness requires redundant path checking; otherwise, not complete.

- \*\*Optimality:\*\* Speed-driven, sacrificing optimality for quicker solutions.

- \*\*Complexity:\*\* Time and space complexities range from O(bd) to O(b^d) based on heuristic quality.

- \*\*Maze Examples:\*\* Visual representation of paths discovered by Greedy Search using both Manhattan and Euclidean heuristics.

Slide 9-10: A\* Search (Manhattan & Euclidean Heuristics)

- \*\*Overview:\*\* Explores A\* algorithm employing heuristics to guide pathfinding.

- \*\*Completeness:\*\* Always complete but optimal only with admissible heuristics.

- \*\*Optimality:\*\* Achieves optimality when heuristics do not overestimate.

- \*\*Complexity:\*\* Time and space complexity varies, worst-case scenario at O(b^d).

- \*\*Maze Examples:\*\* Presentation of optimal paths discovered by A\* using both Manhattan and Euclidean heuristics.

Slide 11: Algorithm Comparison

- \*\*BFS:\*\* Systematic exploration, optimal for identical cost actions, high time and space complexity (O(b^d)).

- \*\*DFS:\*\* Deep exploration, backtracking, non-optimal, space complexity (O(bd)).

- \*\*UCS:\*\* Prioritizes path cost, complete under certain cost conditions, guarantees optimal solutions, time and space complexity (O(b^((C\*)/ϵ))).

- \*\*Greedy Search:\*\* Heuristic-based, not complete or optimal, time and space complexity varies (O(bd) to O(b^d)).

- \*\*A\* Search:\*\* Utilizes heuristics, complete with admissible heuristics, optimal, time and space complexity varies (typically O(b^d)).

Slide 12: Discussion and Conclusion

\*Objectives & Outcomes\*

- \*\*Objective:\*\* Evaluate maze navigation using diverse search algorithms.

- \*\*Outcome:\*\* Successful comparison of seven algorithms in varied maze scenarios.

\*Significance\*

- \*\*Adaptability:\*\* Highlighted algorithms' versatility across maze complexities.

- \*\*Real-World Use:\*\* Demonstrated applicability beyond mazes in problem-solving and planning.

\*Key Insights\*

- \*\*Algorithmic Trade-offs:\*\* Explored trade-offs in completeness, optimality, and complexity.

- \*\*Heuristic Impact:\*\* Noted heuristic influence on pathfinding efficiency.

\*Relevance of Algorithms\*

- \*\*Problem-Solving Tools:\*\* Emphasized their role in diverse problem-solving domains.

- \*\*Continuous Evolution:\*\* Stressed ongoing exploration for solving complex challenges.