Lab A Report

Vassily Lombard, Charles Bowen-Rayer

February 29, 2024

Abstract

This report presents an investigation into the efficiency of various algorithms for solving maze traversal problems with the objective of finding prizes. Maze sizes of varying complexities were employed, and a range of algorithms including Depth-First Search, Breadth-First Search, Greedy Best-First Search, and A* Search were implemented and evaluated. The performance of each algorithm was assessed based on factors such as solution completeness, time efficiency, number of nodes expanded, and optimality of paths discovered. Results indicate that certain algorithms exhibit superior performance in specific maze configurations, highlighting the importance of algorithm selection based on maze characteristics. Insights gained from this project enhanced our understanding of search algorithms and their applications in artificial intelligence.

1 Introduction

Mazes have always been a subject of fascination, challenging people to navigate through intricate paths to reach a goal. In most cases, there is no obvious solution, making this problem-solving process intellectually stimulating for many, although this process is not efficient and time-consuming. In response, computer scientists developed new tools for optimizing this process. In this project, our main goal is to compare the efficiency of different search algorithms in finding prizes from a fixed starting point in mazes with different sizes. To give us a general idea of different algorithm's performance, we first compared Depth-First Search, Breadth-First Search, Greedy Best-First Search, and A* Search on mazes having one prize. Then, we implemented a* Search on mazes with multiple prizes, making the environment more complex. By subjecting these algorithms to mazes of varying sizes and configurations, we aim to discern patterns in their performance by comparing factors such as the number of positions explored, path cost, and running time. Finally, we were able to identify the most suitable algorithmic approaches for different maze types. In the subsequent sections of this report, we will detail our experimental methodology, present the results of our algorithmic analyses, and discuss the implications of our findings.

2 Methods and Models

In our project, we used several approaches to search the mazes. We implemented several algorithms, including Depth-First Search (DFS), Breadth-First Search (BFS), Greedy Best-First Search, and A* Search, each one representing a specific method for searching the maze.

DFS explores as far as possible along each branch before backtracking, utilizing a custom stack implementation. Conversely, BFS systematically explores all neighboring nodes to find the shortest path, employing a custom queue structure. For Greedy Best-First Search and A* Search, we utilized the Manhattan distance heuristic function to estimate the distance to the goal, with A* Search further considering the cost of reaching a node. In both cases, a priority queue data structure helped us keep track of the positions to visit. Then, we extended our A* implementation for multi-prize mazes, modifying the heuristic to consider multiple prizes efficiently. Through experiments on mazes of varying sizes and complexities, we evaluated the algorithms' performance in terms of solution completeness, computational efficiency, and path optimality. These experiments, alongside our heuristic-driven approach, form the basis for our analysis of maze-solving algorithms in different scenarios.

Algorithm	**Data Structure**	**Heuristic**
DFS	Stack	-
BFS	Queue	-
Greedy Best-First Search	Priority Queue	Manhattan distance
A* Search	Priority Queue	Manhattan distance

Table 1: Summary of Maze Traversal Algorithms

3 Results

The evaluation of maze traversal algorithms across varying maze sizes reveals intriguing insights into their performance. Examining the results presented in Table 3, we observe distinct trends across different algorithms and maze complexities. In the case of Depth-First Search (DFS), while it demonstrates relatively low path costs, ranging from 165 to 442 units in small, medium, and large mazes, respectively, its drawback lies in a high number of nodes expanded, indicating inefficiency in systematic exploration. On the other hand, Breadth-First Search (BFS) consistently reveals higher path costs but significantly lower node expansions compared to DFS, making it a favorable choice for scenarios prioritizing thorough exploration over path optimality. Greedy Best-First Search and A* Search, both leveraging heuristic functions, exhibit competitive performance across maze sizes, with relatively low path costs and moderate to low node expansions. Particularly noteworthy is A* Search's proficiency in large mazes, achieving comparatively low path costs with modest node expansions, highlighting its capability in efficiently navigating complex environments while considering the distance to the goal. These findings underscore the importance of considering both path cost and node expansion metrics when selecting a suitable algorithm for maze traversal tasks.

	Small Maze	**Medium Maze**	**Large Maze**
DFS	165, 274	116, 165	442, 831
BFS	45, 524	94, 609	148, 1256
Greedy Best-First Search	57, 155	118, 134	234, 291
A* Search	45, 238	94, 333	152, 1106

Table 2: Summary of Maze Traversal Algorithms Performances. (The first number represents the path cost and the second one is the number of node expanded)

The extension of the A* search algorithm for multiple prizes mazes has yielded intriguing results. Table 3 displays the algorithm's performance across various environments, each characterized by distinct complexities and numbers of prizes. As anticipated, the algorithm demonstrates superior performance in smaller mazes with fewer rewards. The adoption of the Manhattan distance heuristic function showcases competitive performance across maze sizes, with relatively modest path costs and node expansions. Nevertheless, while the path cost remains reasonable, it's crucial to acknowledge that it may not always produce the most optimal path. Future projects could delve into alternative approaches to ascertain the optimal path.

	Micro Maze	**Tiny Maze**	**Small Maze**	**Medium Maze**
A* Search	29, 31	58, 76	197, 335	303, 626

Table 3: Summary of A* Algorithm Performances in Multi-Prize Mazes. (The first number represents the path cost and the second one is the number of node expanded)

4 Conclusion

Our analysis of maze traversal algorithms has revealed clear patterns in their performance across varying maze sizes. While Depth-First Search (DFS) and Breadth-First Search (BFS) excel in small-scale mazes, Greedy Best-First Search and A* Search demonstrate superior efficiency in navigating larger and more complex environments. To further enhance our findings, future projects could explore additional heuristic functions and other search algorithms. The Manhattan distance heuristic function was efficient, but other admissible heuristic functions can improve the performance of our algorithms. Finally, our project offers insights into real-world applications such as enhancing robotic navigation, gaming experiences, and logistical route planning.