Artificial Intelligence

Foundations(CIS 4049-N)

SEARCH ALGORITHM IN OPTIMIZING ROUTE PLANNING FOR ORDER PICKING IN A MAZE-LIKE WAREHOUSE ENVIRONMENT

Word Count: 4207 without reference and table of contents

ADURA AMOO – D3622546

Table of Contents

ABSTRACT

This study delves into the convergence of Artificial Intelligence (AI) and route optimization algorithms to streamline order-picking processes within labyrinthine warehouse environments. Focusing on the crucial task of determining the optimal path for an order-picker navigating through aisles to retrieve items, the study employs and compares Breadth First Search (BFS) and A* algorithms across two distinct maze layouts. The literature review underscores the significance of efficient order fulfilment, emphasizing potential cost savings through strategic order batching. Additionally, it outlines the broad relevance of search algorithms in warehouse logistics, categorizing them as uninformed/blind and informed. The research methodology involves the implementation and comparative analysis of BFS and A* algorithms, leveraging Python in maze simulations. Results consistently reveal the superior performance of the A* algorithm, demonstrating heightened efficiency in pathfinding and reduced computational time. This positions A* as a promising solution for optimizing route planning in order-picking scenarios. In conclusion, this study advocates for the adoption of A* algorithm in maze-like warehouse environments, showcasing its capacity to minimize order-picking time and enhance resource utilization.

1 INTRODUCTION

The development of computer programmes that can carry out tasks that are typically performed by humans is known as artificial intelligence (AI). This area of computer science makes it now feasible to create intelligent machines with human-like behavior, mental processes, and decision-making skills. Four primary areas are covered by artificial intelligence: reasoning, behaving rationally, thinking human, and acting human. Artificial intelligence (AI) algorithms may perform a variety of tasks, such as learning, vision, solving problems, comprehending language, using logic, and playing games. The field of artificial intelligence (AI) encompasses a number of subfields, such as expert systems, natural language processing (NLP), vision (image recognition, machine learning), speech (text-to-speech, speech-to-text), planning, robotics, and machine learning (supervised, unsupervised, reinforcement learning, and deep learning). (Ziyad, 2019)

In the dynamic landscape of modern commerce, warehouses serve as the bustling nerve centres that facilitate the seamless flow of goods and materials. As the demand for efficiency continues to escalate, the role of Artificial Intelligence (AI) in optimizing warehouse operations has become increasingly pivotal. One of the critical challenges faced by warehouse managers is the intricate task of route planning within maze-like warehouse environments, where the arrangement of shelves, storage units, and aisles resembles a complex maze. (Shetty, Sah and Chung, 2020)

Efficient route planning is paramount for order picking in a warehouse, as it directly impacts productivity, operational costs, and overall throughput. The intricate nature of maze-like configurations demands advanced solutions, and AI, coupled with sophisticated search algorithms, emerges as a beacon of promise in this domain. (Lu *et al.*, 2016)

This study delves into the intersection of AI, warehouse logistics, and route planning, unravelling the intricacies of navigating maze-like warehouses for order picking with a focus on determining the optimal route for order-picker to travel along the aisles from the start location to the storage location(goal) within the warehouse, effectively identifying the most efficient path to reach the goal. We explore the symbiotic relationship between AI technologies and search algorithms, shedding light on how these intelligent systems can revolutionize route planning strategies to meet the evolving demands of modern supply chains. This study aims to determine the optimal route for an order-picker (Agent) navigating along the aisles from a start location to collect requested items from designated storage location (goal location) in a maze-like warehouse.

2 Literature Review

Analysis in this study is based on the research work "walk less, pick more: choosing optimal batches of order in a warehouse"(Lu *et al.*, 2016) , where the researcher considers two optimization problems; The picker routing problem that involve finding routes through the warehouse to minimize the distance traveled and the batching problem referring to selecting a combination of orders that minimizes the distance traveled. A collection of algorithms which include Held Karp, Held Karp successive approximation, nearest neighbor, tree doubling and Christofides was designed to solve the optimization problems. The researcher also evaluates the algorithm using quality and runtime as benchmarks.

(Matusiak *et al.*, 2014) consider that efficient order fulfillment processes within picker-to-parts warehouses have become a focal point for researchers seeking to optimize operational costs. One key area of investigation involves the realization of considerable savings through the strategic batching of customer orders, particularly in scenarios where precedence constraints dictate the order-picking sequence.

In addressing this combined challenge of Order-batching and navigation with restriction on precedence, a novel approach is introduced by the researcher. The method integrates two sub-algorithms, namely an optimal A$*$ search algorithm for routing optimization and a simulated annealing algorithm tailored for order-batching (Matusiak *et al.*, 2014) . The latter algorithm is designed to estimate potential savings derived from combining more than two orders from customers, thereby mitigating superfluous routing and enhancing overall efficiency.

2.1 Search Algorithm

Search algorithms, a cornerstone in various applications, possess extensive applicability in everyday life, with a notable focus on pathfinding—a complex challenge within the realm of Artificial Intelligence (AI). The exploration of pathfinding algorithms has been a longstanding research interest, particularly evident in computer games, where it represents a well-known and intricate AI challenge. Over the years, numerous applications, including mapping, gaming, robotics, and metabolic pathways, have found use for these methods (Barnouti, Al-Dabbagh and Naser, 2016) .

In the context of the warehouse industry, the significance of pathfinding algorithms becomes pronounced, offering solutions to optimize operational processes. The challenges within warehouse environments, akin to those encountered in gaming scenarios, involve determining the most efficient routes while navigating through dynamic and potentially obstacle-laden spaces. Pathfinding algorithms, originally developed for gaming, can be adapted and leveraged to address analogous problems in warehouse logistics. (M. Li, X. Yan and Q. Luo, 2020)

Finding the shortest path between two sites while avoiding barriers is a fundamental problem in the warehouse domain, and challenges like traffic routing, labyrinth navigation, and robot path planning are specific instances of this. As the warehouse industry continues to embrace automation and robotics, the integration of pathfinding algorithms becomes pivotal for enhancing the efficiency of material handling and order fulfilment processes. (D. Shi *et al.*, 2023)

In the warehouse industry, the application of search algorithms becomes a key driver in addressing challenges related to optimal route planning, inventory movement, and resource utilization. Uninformed/blind search algorithms and informed search algorithms, distinguished by their approach to decision-making, offer versatile tools for navigating the complex spatial and operational dynamics within a warehouse setting.

The gaming industry's utilization of search algorithms provides a valuable precedent, demonstrating the adaptability of these algorithms to dynamic, real-time environments. As the warehouse industry evolves towards greater automation and smart logistics, leveraging the lessons learned from pathfinding algorithms in gaming can contribute significantly to enhancing efficiency and reducing operational costs in warehouse management systems. (Barnouti, Al-Dabbagh and Naser, 2016)

This review underscores the broad relevance of search algorithms and their potential transformative impact on optimizing warehouse operations.

2.1.1 Uninformed/ Blind Search Algorithms:

Uninformed search algorithms, also referred to as blind search algorithms, operate without any more details guiding their decision-making process. Unlike their informed counterparts, which benefit from heuristic knowledge or estimates to guide the search, uninformed algorithms rely solely on the inherent structure of the problem and their exploration of the search space. Uninformed search algorithms can only distinguish between a state being a goal or non-goal condition; they do not exhibit any preference regarding which state (node) to expand next in the exploration process.

2.1.1.1 Breadth first Search (BFS)

Every node within the Breadth First Search is enlarged level by level. In order to accomplish the goal, it first grows every node at the first level of the search tree, then expands every node at the second level. The frontier is really realised as a queue in breadth first search, and it operates on the principle of first in, first out (FIFO). When there is some heuristic information available or when all of the solutions have a long way to go(large path length), it is a bad technique .When the need for memory is great, it should not be used. (Pathak, Patel and Rami, 2018)

The algorithm proves highly beneficial in specific scenarios, notably when identifying neighboring nodes in a peer-to-peer network. An illustrative application includes its role in locating "seeds" and "peers" within a torrent network. Additionally, the algorithm finds utility in web crawlers tasked with indexing web pages. However, a notable drawback is its demand for the generation and retention of all connected nodes in memory, leading to increased memory consumption. Moreover, its implementation may pose complexity, particularly in networks characterized by expansive search spaces.

2.1.2 Informed Search Algorithms

In contrast to uninformed search algorithms, informed search algorithms leverage additional information, often referred to as a heuristic function, to guide their decision-making process in navigating towards the goal state. The heuristic function serves as a means of estimating the desirability or quality of a particular state, providing the algorithm with a heuristic score that influences the exploration of the search space.

The heuristic function assigns scores to states based on domain-specific knowledge or rules, allowing the algorithm to make more informed decisions about which paths to prioritize. The higher the heuristic score, the more promising or closer to the goal a particular state is considered. It determines the cost of the best route between the two states and is represented by h(p). (Iloh, 2022) .

h(p) = The estimated proximity between node n and the target node.

A well-known example of an informed search algorithm is A* (A-star) search.

2.1.2.1 A Search*

The A* (A-star) algorithm is indeed an informed search algorithm commonly used in traversing graphs and locating paths. Its purpose is to determine the most effective route in a graph or grid between a beginning and a destination state, considering both the cost of reaching a certain state and an estimate of the amount (cost) still needed to achieve the goal. The benefits of both Dijkstra's algorithms (which considers the cost to reach a state) and greedy best-first search (which calculates the remaining cost to reach the target using a heuristic) are combined in A*.

The A* algorithm maintains a priority queue to explore the states in a specific order, and it evaluates each state based on a function called the evaluation function. The evaluation function combines two components:

Cost function (g(p)): This is the actual cost of getting to a particular state from the start node. It records the total amount spent from the starting point to the present state.

Heuristic function (h(p)): The cost to get from the present state to the desired condition is estimated here. It offers an educated estimate, or heuristic, of the remaining cost. The heuristic function never overestimates the genuine cost to achieve the goal because it is problem-specific and ought to be tolerable.

The cost function plus the heuristic function add up to the evaluation function (f(p))(Iloh, 2022) .

$$\boxed{F(p) = g(p) + h(p)}$$

A* algorithm selects the state with the lowest) *f*(*p*) value from the priority queue for expansion. By doing this, it explores the most promising paths first, considering both the actual cost incurred so far and the estimated cost remaining to the goal. This helps A* to quickly find an optimal solution while avoiding unnecessary exploration of less promising paths. (Iloh, 2022)

When certain requirements are met, such having an admissible heuristic and non-negative edge costs, A* ensures optimality. A* will find the shortest path between the start state and the goal state if these prerequisites are satisfied. However, the effectiveness of A* is also influenced by the calibre of the heuristic function used, since a better-informed heuristic can direct the search more precisely and effectively. (Liu and Gong, 2011)

This study aims to determine which search algorithm, out of A* and Breadth First Search (BFS), is better suited for order selecting in a warehouse setting that resembles a maze.

3 RESEARCH METHODOLOGY

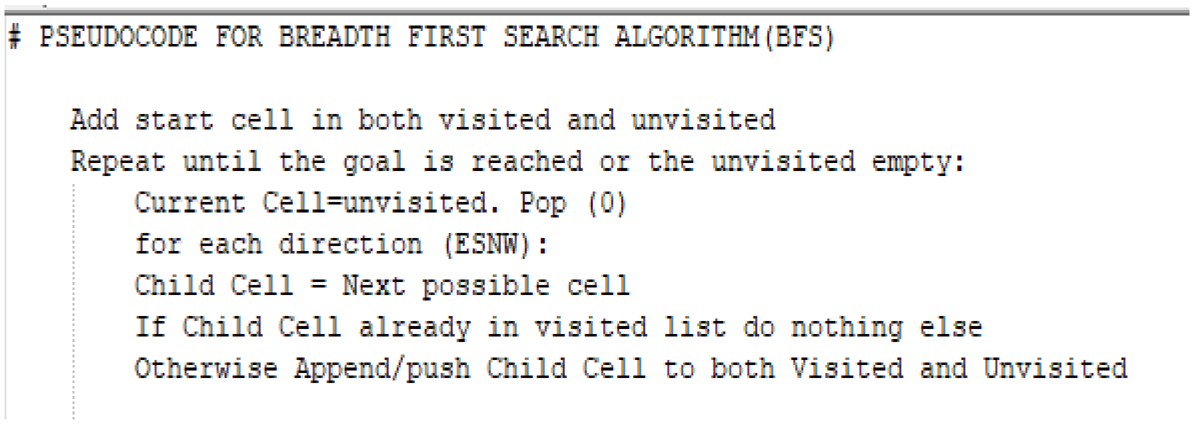
The goal of this study is to determine the order-picking process's shortest and most effective path in a warehouse setting that is modelled by a maze game. The study methodology used to evaluate the two search algorithms is testing the algorithms' performance in two distinct, identically sized mazes. To evaluate the effectiveness of the various search algorithms, two cells were selected from the maze to serve as the start position and the objective location—the place where the target item should be chosen.

It is possible to identify which search algorithm performs best for path discovery during order picking in a maze-like warehouse setting by evaluating the two mazes using the two search algorithms. Path Length, Search Length, and Time Complexity are used to gauge the algorithm's performance.

3.1 Optimal Path with Breadth First Search Algorithm

The following pseudocode was used to derive the Python code for the Breadth First Search function, which implemented the algorithm. The best route between the starting point and the destination was then determined by using the function. (Pathak, Patel and Rami, 2018)

The following is the BFS Algorithm's pseudocode:



```
# PSEUDOCODE FOR BREADTH FIRST SEARCH ALGORITHM(BFS)

    Add start cell in both visited and unvisited
    Repeat until the goal is reached or the unvisited empty:
        Current Cell=unvisited. Pop (0)
        for each direction (ESNW):
        Child Cell = Next possible cell
        If Child Cell already in visited list do nothing else
        Otherwise Append/push Child Cell to both Visited and Unvisited
```

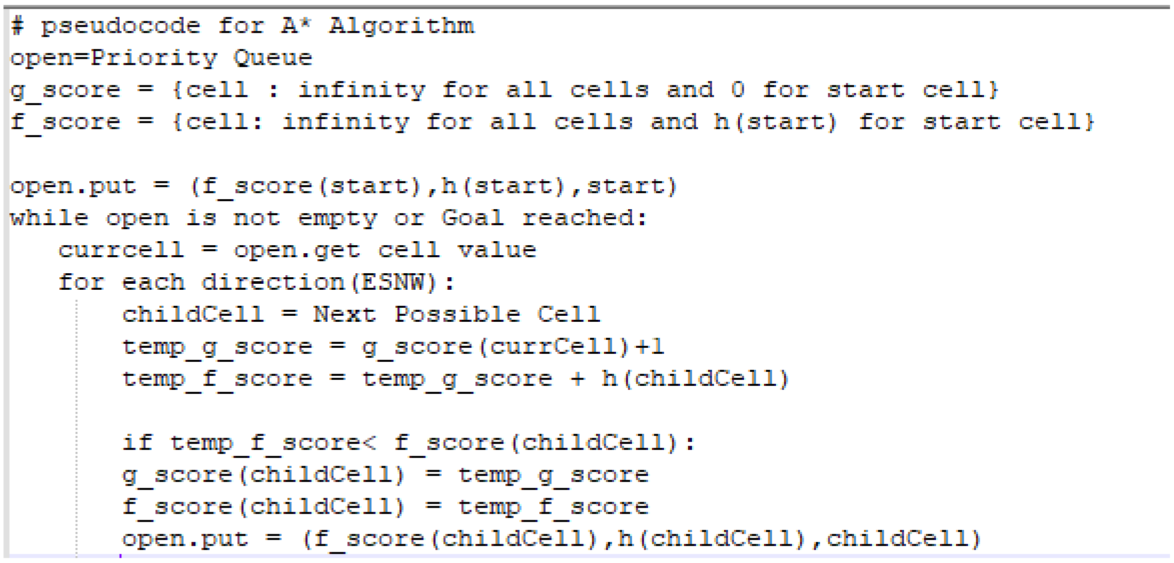
Figure 1: BFS PSEUDOCODE

3.2 Optimal Path with A* Algorithm:

One advantage of using the A* search algorithm is that, provided the heuristic estimate is not overestimated, it will always identify the shortest path.(Ziyad, 2019) , The A* search algorithm is the most used path finding technique in artificial intelligence games. The A* algorithm uses the heuristic function as efficiently as possible, extending fewer nodes and finding an optimal path than any other algorithm using the same heuristic, even if the heuristic must not be overestimated. Moreover, the A* algorithm performs better with a more accurate estimate of the heuristic. In a similar vein, an estimate that is not entirely correct could decrease a heuristic's effectiveness.(Liu and Gong, 2011) .

Python programming language was used to implement the A* search algorithm function, which finds the best path from the starting point to the destination by leveraging the influence of a heuristic function. The heuristic function was computed using the Manhattan distance (Cui and Shi, 2011) .

The pseudocode for A* (5) is as follows:



```
# pseudocode for A* Algorithm
open=Priority Queue
g_score = {cell : infinity for all cells and 0 for start cell}
f_score = {cell: infinity for all cells and h(start) for start cell}

open.put = (f_score(start),h(start),start)
while open is not empty or Goal reached:
   currcell = open.get cell value
   for each direction(ESNW):
       childCell = Next Possible Cell
       temp_g_score = g_score(currCell)+1
       temp_f_score = temp_g_score + h(childCell)

       if temp_f_score< f_score(childCell):
       g_score(childCell) = temp_g_score
       f_score(childCell) = temp_f_score
       open.put = (f_score(childCell),h(childCell),childCell)
```

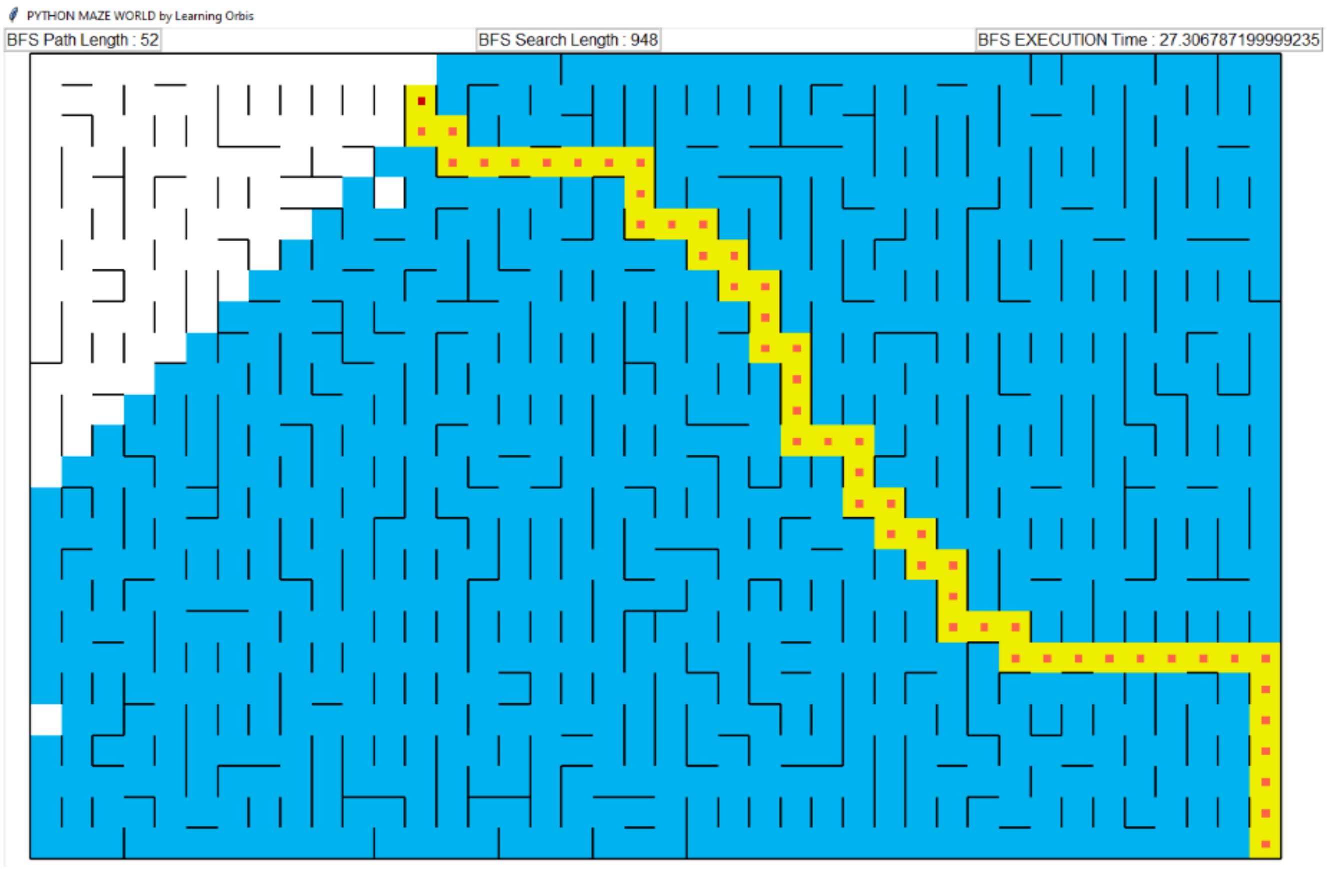
Figure 2: A PSEUDOCODE*

4 RESULTS & DISCUSSION

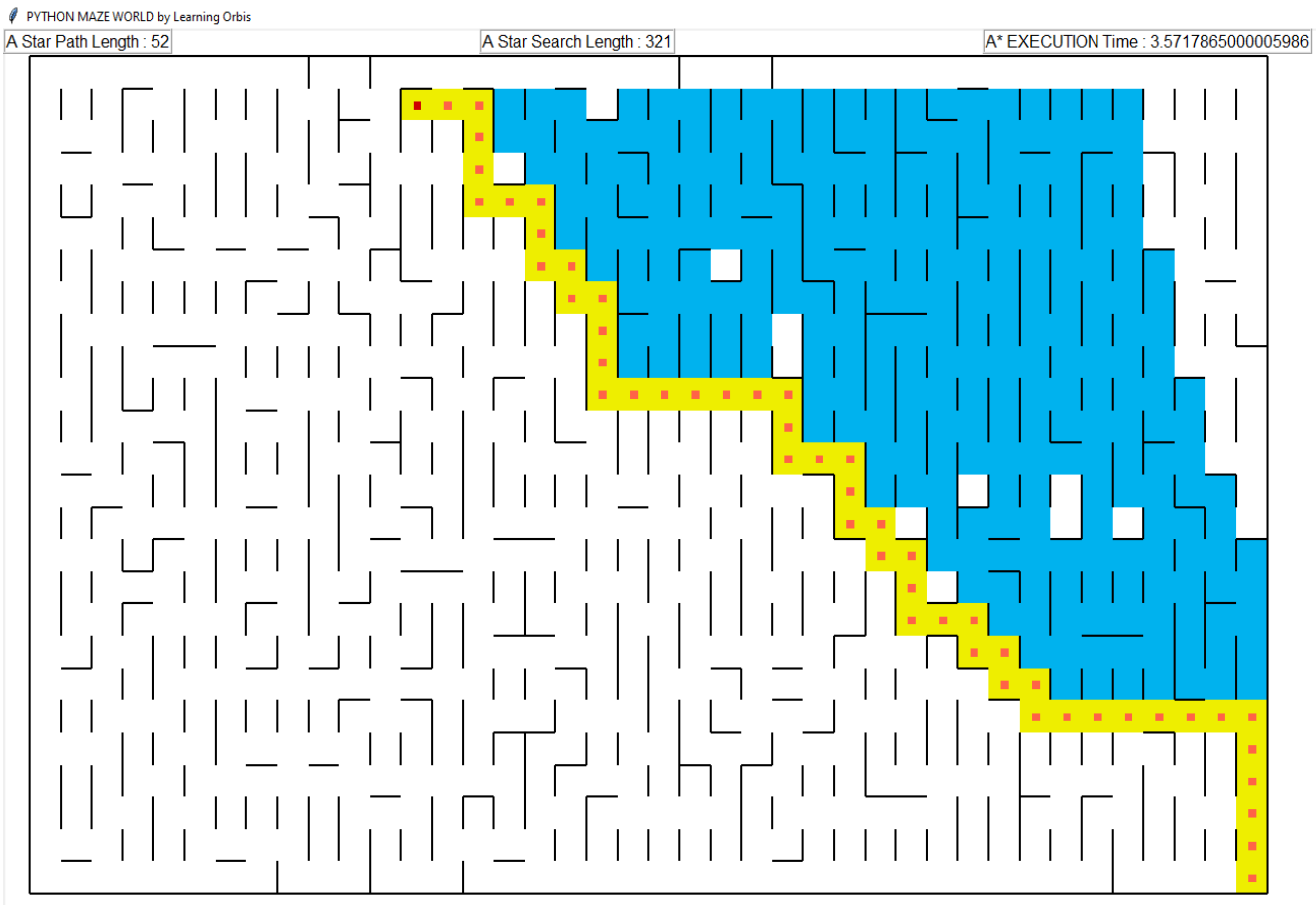
The research aimed to evaluate and compare the performance of Breadth First Search (BFS) and A* algorithms in optimizing order-picking processes within a maze-like warehouse environment. The study utilized a 26 by 40 maze with a vertical pattern generated by the pyamaze module for simulations, employing Python as the programming language on a computer with a 3.0 GHz Intel Core i7 7th gen processor speed and 8.0 GB of RAM memory.

4.1 Simulation Setup (26 by 40 Maze):

The study employed customized random mazes generated by the pyamaze module, introducing variability with each simulation run. The maze, representing a warehouse layout, was designed with a vertical pattern. Both BFS and A* algorithms were implemented and tested within this dynamic maze environment, and their performance was evaluated based on search length, path length, and execution time.



BFS Path Length : 52
BFS Search Length : 948
BFS EXECUTION Time : 27.306787199999235


PYTHON MAZE WORLD by Learning Orbis

A Star Path Length : 52

A Star Search Length : 321

A* EXECUTION Time : 3.5717865000005986

Figure 3: First Simulation for order-picking in a Maze-like Warehouse with BFS (3a), and A star (3b) algorithms

The order-picking simulation result for the Maze-like Warehouse at the initial obstacle configuration is displayed in Figure 3. The first simulation has a target of (2,13) and is a 26 by 40 maze with a vertical layout. Figures 3a and 3b illustrate the agent's movements using the BFS and A*algorithms, respectively.

Table 1: Performance Comparison of Algorithms for the first simulation

	Path Length	Search Length	Execution Time (milliseconds)
BFS Algorithm	52	948	27.3068
A* Algorithm	52	321	3.5718

4.2 First Simulation Obstacle Layout

Completeness:

a) BFS Algorithm: BFS is a complete search algorithm, meaning it will find a solution if one exists. In your case, with a search length of 948, BFS explored a relatively large number of nodes to reach the solution, demonstrating its completeness.

b) A* Algorithm: A* is also a complete algorithm. It guarantees finding the optimal solution if one exists. **The A* algorithm explored fewer nodes (search length of 321) compared to BFS, indicating a more efficient exploration strategy.**

Optimality:

a) BFS Algorithm: BFS is guaranteed to find the optimal solution, as it explores nodes level by level. However, it might be less efficient in terms of time and space complexity for certain problems.

b) A* Algorithm: In this case, **A* achieved a shorter search length (321) compared to BFS**, **indicating that it found a more optimal solution in terms of the number of nodes explored.**

Time Complexity:

a) BFS Algorithm: The time complexity of BFS is O(b^d), where b is the branching factor and d is the depth of the solution. In your case, the execution time for BFS was 27.3068 milliseconds.

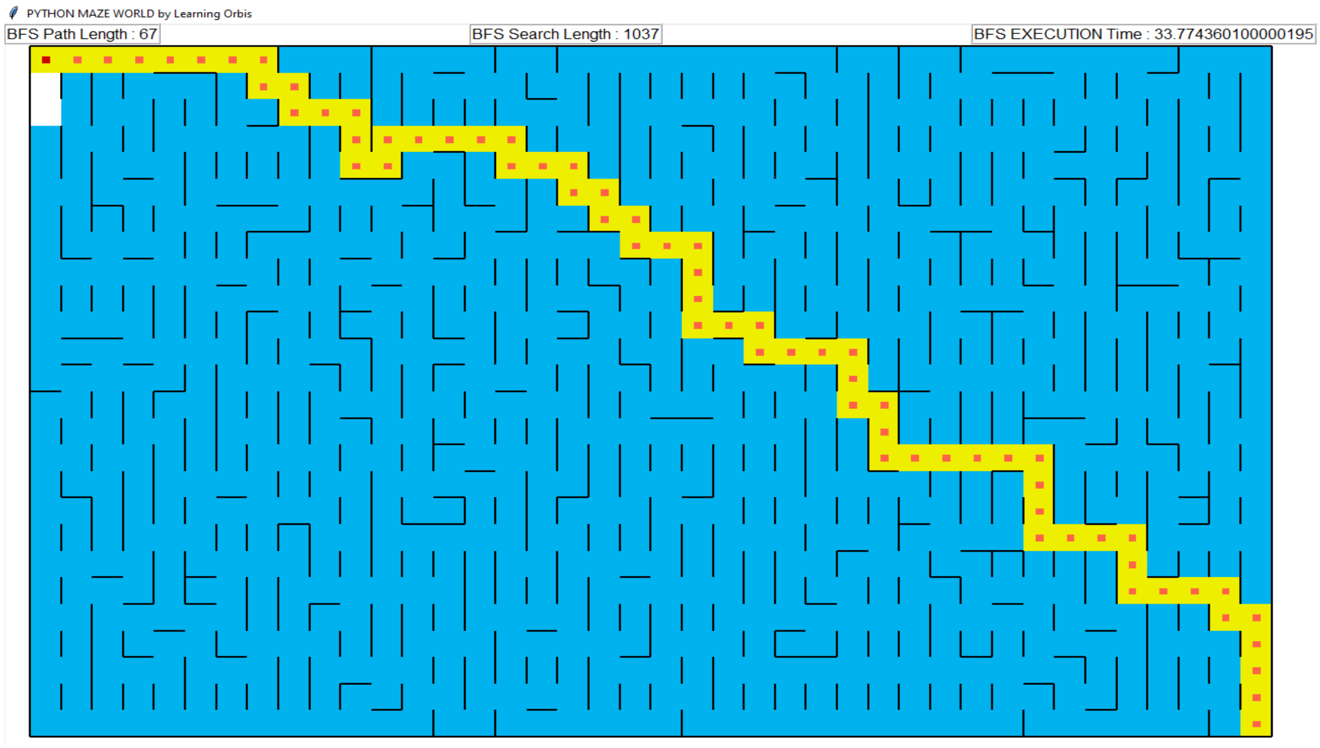
b) A* Algorithm: The time complexity of A* depends on the heuristic function and the quality of the heuristic. In this case, the A* algorithm executed in 3.5718 milliseconds, demonstrating its efficiency in finding a solution quickly.

Space Complexity:

a) BFS Algorithm: The space complexity of BFS is O(b^d), where b is the branching factor and d is the maximum depth of the search tree. It stores all the nodes at a given depth in memory. The space complexity can be significant, especially for large search spaces.

b) A* Algorithm: The space complexity of A* depends on the implementation and the data structures used. A* often requires less memory compared to BFS because it uses a heuristic to guide the search, focusing on more promising paths.

In summary, while both algorithms are complete and optimal, A* demonstrates superior efficiency in terms of the number of nodes explored and execution time. However, it's important to note that the efficiency of these algorithms can vary depending on the specific characteristics of the problem and the quality of the heuristic used in A*.


PYTHON MAZE WORLD by Learning Orbis
BFS Path Length : 67
BFS Search Length : 1037
BFS EXECUTION Time : 33.774360100000195

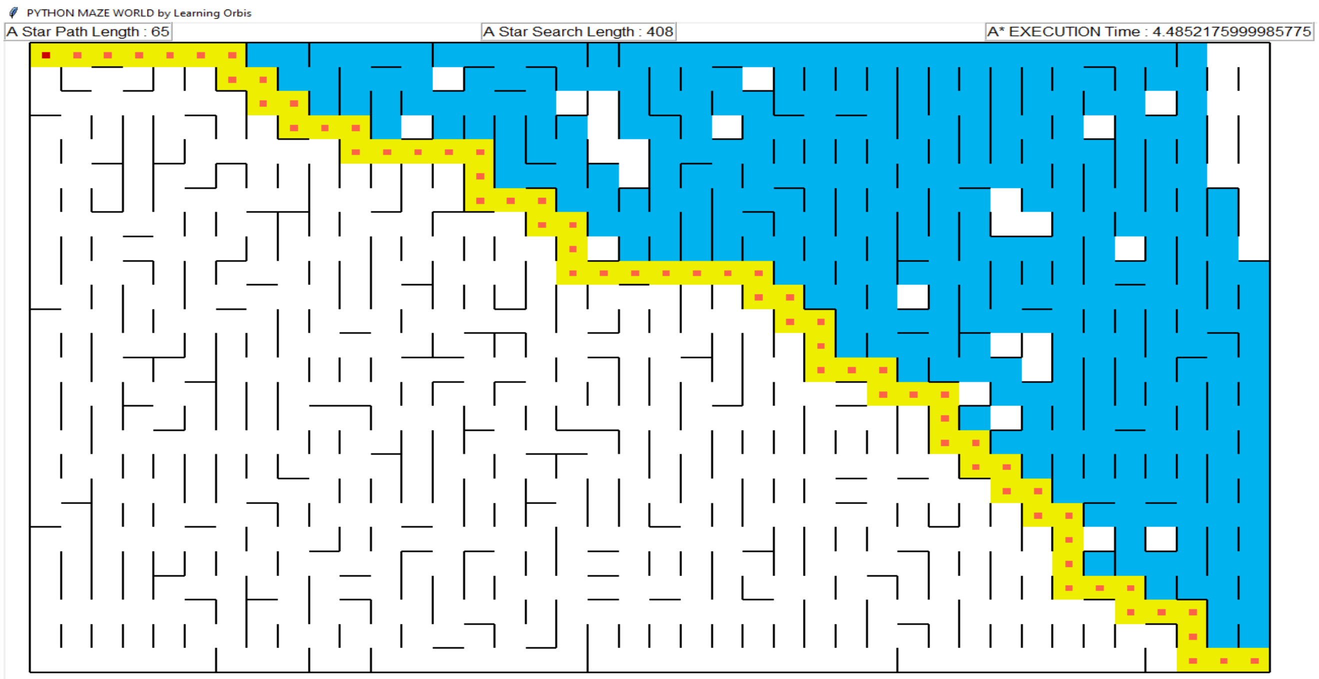

PYTHON MAZE WORLD by Learning Orbis
A Star Path Length : 65
A Star Search Length : 408
A* EXECUTION Time : 4.4852175999985775

Figure 4: Second Simulation for order-picking in a Maze-like Warehouse with BFS (4a), and A star (4b) algorithms

The simulation results for the warehouse that resembles a maze in the second obstacle configuration are shown in Figure 4. To be more precise, Figure 4a shows the agent's movements while using the BFS method, and Figure 4b shows the agent's trajectory when using the A* algorithm.

Table 2: Comparison of Algorithms for the Second ObstacleLayout of the Maze-like Warehouse

	Path Length	Search length	Execution Time (milliseconds)
BFS Algorithm	67	1037	33.7744
A* Algorithm	65	408	4.4852

4.3 Second Simulation Obstacle Layout

Completeness:

a) BFS Algorithm (67): Breadth-First Search is a complete algorithm, meaning that it will always find a solution if one exists. The search length of 67 indicates that the algorithm explored 67 nodes before finding a solution.

b) A* Algorithm (65): A* is also a complete algorithm, ensuring that it will find a solution if it exists. The search length of 65 suggests that A* explored fewer nodes compared to BFS to reach a solution.

Optimality:

a) BFS Algorithm (1037): While BFS is complete, it doesn't guarantee optimality in terms of the solution's cost. It may find a solution, but it might not be the most efficient one.

b) A* Algorithm (408): A* is an optimal algorithm under certain conditions (admissibility and consistency of the heuristic function). The search length of 408 suggests that it explored more nodes than BFS, but the optimality depends on the specific problem and heuristic used.

Time Complexity:

a) BFS Algorithm (33.7744 milliseconds): The execution time for BFS is 33.7744 milliseconds. BFS generally has higher time complexity compared to A* due to the nature of exploring all nodes at a given depth before moving on to the next depth.

b) A* Algorithm (4.4852 milliseconds): The execution time for A* is significantly lower (4.4852 milliseconds). A* can be more time-efficient in finding optimal solutions, especially when a good heuristic is used.

Space Complexity:

a) BFS Algorithm: The space complexity of BFS is generally higher than A* because it stores all nodes at a given depth in the memory.

b) A* Algorithm: A* also has space complexity, but the exact value is not provided in the results. The space complexity depends on factors such as the size of the explored set and the data structures used.

In summary, both BFS and A* are complete algorithms, but A* has the potential to be more optimal and efficient in terms of time. The specific characteristics of the problem and the quality of the heuristic used in A* can significantly impact its performance. The provided results indicate that A* explored fewer nodes and had a lower execution time compared to BFS in this scenario.

5 PROFESSIONAL AND ETHICAL CONCERN

Several professional and ethical issues are associated with the problem of pathfinding for order-picking in warehouse. Some of these include:

1. **Privacy and Data Protection:**
    - **Professional Concern:** The study involves the use of algorithms and simulations to optimize order-picking processes in a warehouse. It is essential to consider the privacy and protection of sensitive data related to warehouse operations and potentially proprietary algorithms used by the companies involved.
    - **Ethical Concern:** Researchers must ensure that any data used in simulations is anonymized and that the findings do not compromise the confidentiality of specific warehouse layouts or operational details. Transparent and ethical data handling practices are crucial.

2. **Algorithmic Bias:**
    - **Professional Concern:** The use of AI algorithms, including A* and BFS, raises concerns about algorithmic bias. If the algorithms are trained on biased data or if biases are unintentionally introduced during implementation, it can impact the fairness and reliability of the results.
    - **Ethical Concern:** Researchers should assess and address potential biases in the algorithms to ensure fair and equitable outcomes. Transparency about the training data, potential biases, and steps taken to mitigate them is crucial.
3. **Stakeholder Involvement:**
    - **Professional Concern:** The study may impact warehouse operations, and it is important to involve stakeholders, such as warehouse managers and workers, in the research process to gain insights and perspectives from those directly affected.
    - **Ethical Concern:** Ensuring the well-being and fair treatment of workers is essential. Researchers should consider the ethical implications of any proposed changes to operational processes and seek input from relevant stakeholders.
4. **Transparency in Research Methodology:**
    - **Professional Concern:** Clear and transparent reporting of the research methodology is crucial for the reproducibility and validity of the study. Lack of transparency may hinder the ability of other researchers to validate or build upon the findings.
    - **Ethical Concern:** Transparent reporting is an ethical obligation to the scientific community and ensures that the study's results are trustworthy. Any limitations or challenges faced during the research should be openly communicated.
5. **Environmental Impact:**
    - **Professional Concern:** The computational resources used in the study, such as a 3.0 GHz Intel Core i7 processor, contribute to the environmental footprint. Researchers should be mindful of resource consumption.
    - **Ethical Concern:** Considering the environmental impact of the research is

an ethical responsibility. Researchers should explore ways to minimize resource usage and potentially use sustainable computing practices.

6. **Informed Consent and Human Subjects:**
    - **Professional Concern:** If the study involves human subjects, even indirectly, ensuring informed consent and adherence to ethical guidelines is paramount.
    - **Ethical Concern:** Respecting the rights and well-being of human subjects is a fundamental ethical consideration. Researchers must obtain informed consent, protect privacy, and ensure that any data collected is used ethically and responsibly.

Addressing these professional and ethical concerns will contribute to the credibility, reliability, and responsible conduct of the study in the field of AI-driven warehouse optimization.

6 CONCLUSION

In conclusion, this study delved into the intersection of Artificial Intelligence (AI), warehouse logistics, and route planning, focusing on the optimization of order-picking processes in maze-like warehouse environments. The research aimed to determine the optimal route for an order-picker navigating through aisles to collect requested items from designated storage locations (goal locations) within a maze.

The literature review provided insights into previous works, emphasizing the significance of efficient order fulfillment processes and the potential savings derived from strategic order batching. Additionally, the review highlighted the broad relevance of search algorithms, particularly in warehouse logistics, and introduced two categories of search algorithms: uninformed/blind search algorithms and informed search algorithms.

The research methodology involved implementing and comparing two search algorithms, Breadth First Search (BFS) and A*, to determine their effectiveness in optimizing route planning for order-picking in maze-like warehouses. Two distinct maze layouts were considered, and the performance of the algorithms was evaluated based on path length, search length, and execution time.

The results presented in Table 1 and Table 2 indicate that, in both maze layouts, A* consistently outperformed BFS in terms of path length, search length, and execution time. A* demonstrated a more efficient exploration of the maze, identifying shorter paths and completing the search process in significantly less time. This suggests that the informed search strategy of A*, utilizing heuristic information, proves advantageous in maze-like warehouse environments.

Therefore, based on the findings, it can be concluded that A* algorithm is better suited for optimizing order-picking route planning in maze-like warehouses compared to Breadth First Search. The superior performance of A* in terms of pathfinding efficiency and computational speed makes it a promising choice for real-world applications where minimizing order-picking time and optimizing resource utilization are critical factors.

This study contributes valuable insights to the evolving field of warehouse logistics, showcasing the potential impact of AI-driven search algorithms in enhancing operational efficiency. Future research could further explore the adaptability of these algorithms to varying warehouse configurations and obstacles, providing a more comprehensive understanding of their applicability in diverse real-world scenarios.

7 IMPLICATIONS AND FUTURE RESEARCH

The findings of this study have implications for maze-like warehouse scenarios, particularly in the context of order-picking processes. The computational efficiency demonstrated by A* suggests its potential superiority in real-time applications. Future research could explore the algorithms' adaptability to various maze layouts and dimensions, as well as investigate the impact of different heuristic functions on A* performance.

8 REFERENCES

1. Barnouti, N.H., Al-Dabbagh, S.S.M. and Naser, M.A.S. (2016) 'Pathfinding in strategy games and maze solving using A* search algorithm', *Journal of Computer and Communications*, 4(11), pp. 15.
2. Cui, X. and Shi, H. (2011) 'A*-based pathfinding in modern computer games', *International Journal of Computer Science and Network Security,* 11(1), pp. 125-130.
3. D. Shi *et al.* (2023) 'Collision-Aware Route Planning in Warehouses Made Efficient: A Strip-based Framework', *2023 IEEE 39th International Conference on Data Engineering (ICDE).* Available at: 10.1109/ICDE55515.2023.00072.
4. Iloh, P.C. (2022) 'A Comprehensive and comparative study of DFS, BFS, and A* search algorithms in a solving the maze transversal problem', *International Journal of Social Sciences and Scientific Studies,* 2(2), pp. 482-490.
5. Liu, X. and Gong, D. (2011) 'A comparative study of A-star algorithms for search and rescue in perfect maze', *2011 international conference on electric information and control engineering.* IEEE
6. Lu, W. *et al.* (2016) 'An algorithm for dynamic order-picking in warehouse operations', *European Journal of Operational Research,* 248(1), pp. 107-122. Available at: https://doi.org/10.1016/j.ejor.2015.06.074
7. M. Li, X. Yan and Q. Luo (2020) 'Dynamic path planning for multi-handling robots in warehousing system', *2020 Global Reliability and Prognostics and Health Management (PHM-Shanghai).* Available at: 10.1109/PHM-Shanghai49105.2020.9280945.
8. Matusiak, M. *et al.* (2014) 'A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse', *European Journal of Operational Research,* 236(3), pp. 968-977. Available at: https://doi.org/10.1016/j.ejor.2013.06.001
9. Pathak, M.J., Patel, R.L. and Rami, S.P. (2018) 'Comparative analysis of search algorithms', *International Journal of Computer Applications,* 179(50), pp. 40-43.
10. Shetty, N., Sah, B. and Chung, S.H. (2020) 'Route optimization for warehouse

order picking operations via vehicle routing and simulation', *SN Applied Sciences*, 2, pp. 1-18.

11. Ziyad, M. (2019) 'Artificial Intelligence Definition, Ethics and Standards', *Artif.Intell.Defin.Ethics Stand*, , pp. 1-11.