Optimizing Food Delivery by Robots in Restaurants

Presented by: Emaan Bashir

Agenda

Introduction

- Running a restaurant smoothly is more than just serving tasty food
- Efficient restaurant operations are paramount for ensuring customer satisfaction and profitability
- Critical aspect of restaurant management is the optimization of the flow of orders from the kitchen to the customers' tables
- Diners expect prompt service and seamless experiences
- Restaurants might consider replacing the waiters with autonomous agents to improve the customer experience.

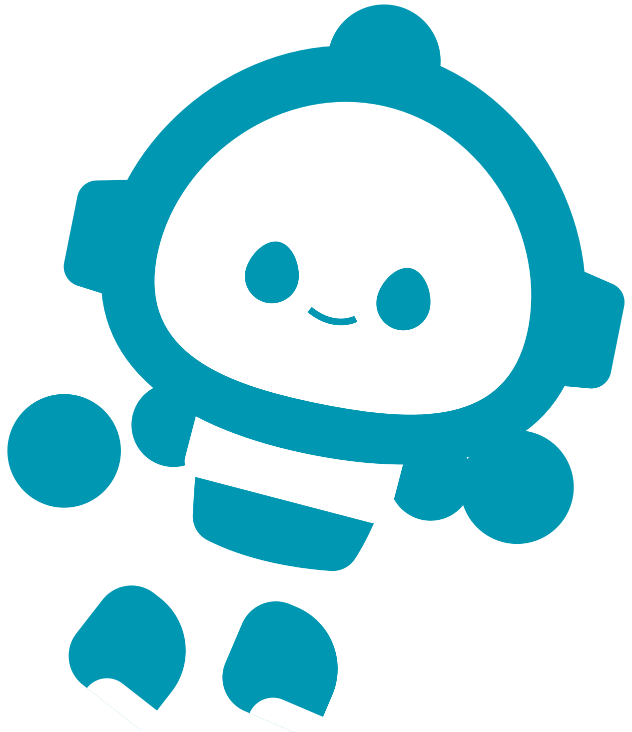
Core Idea and Research Question

- Simulate a restaurant environment where the bots are responsible for collecting food from the kitchen and delivering it to the customers, while avoiding collision with other bots and customers.

- Analyze the impact of the number of bots on the customer waiting time.

- Analyze the impact of different path finding algorithms on customer waiting time.
  - A* Algorithm
  - Dijkstra's Algorithm
  - Depth First Search (DFS)
  - Breadth First Search (BFS)

How do the number of bots and the choice of path finding algorithm impact the average customer waiting time in a restaurant environment ?



Related Work

Food Delivery Automation in Restaurants Using Collaborative Robotics	**Items-mapping and route optimization in a grocery store using Dijkstra's, Bellman-Ford, and Floyd-Warshall Algorithms**	**Optimal and Efficient Path Planning for Partially-Known Environments**
Authors: Albin Antony & P.Siviraj	**Authors:** Dela Cruz, V. Magwili, E.Mundo, B.Gregorio, L.Lamoca	**Author:** A.Stentz
• Proposes a centralized system that coordinates a team of robots in a restaurant that optimally assign tasks to different robots based on factors like distance, time and energy. • Emphasis on task optimization and use of A* for navigation aligns with our study's focus on efficient resource allocation and algorithm's impact on customer waiting times.	• Explores the application of different algorithms to find the optimal shopping routes in grocery stores to assist customers efficiently locate items using the computed shortest paths displayed on an android application. • The core objective of optimizing customer experiences by minimizing traversal times aligns with our investigation	• Addresses the challenge of trajectory planning for mobile robots in partially known environments, where the robot lacks complete information about its surroundings • Introduces an algorithm D* which is capable of planning paths in unknown or changing environments efficiently and optimally. • The principles of adaptive path planning resonate with our exploration of pathfinding algorithm's suitability in dynamic environments.

Technologies Used

- Python Tkinter
- Matplotlib

TKINTER
GUI FOR PYTHON

matplotlib

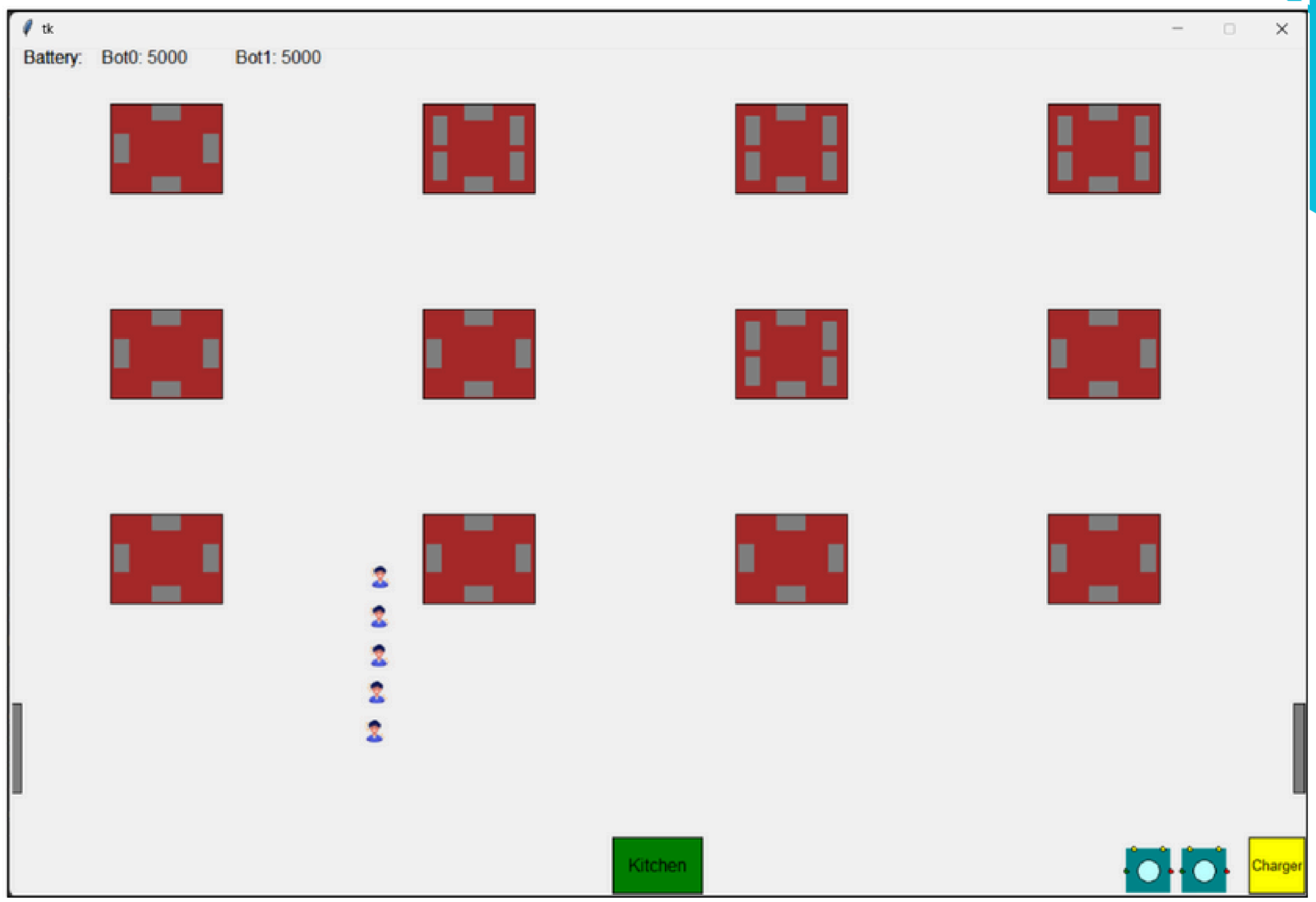
Challenges

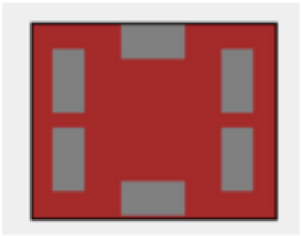
- Optimizing the robot movement in the restaurant environment.
- Handling the interaction of bots with customers and other bots.
- Handling the slow movements in the Tkinter window due to numerous updates and interactions overwhelming the event loop, hindering smooth GUI responsiveness.

Simulation Environment and Agent

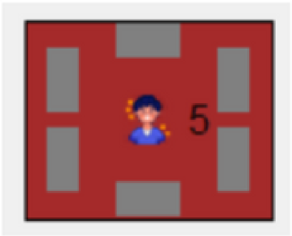
Simulation Environment

It consists of passive objects; tables, kitchen, doors and charging station.





(a) Table without customers



(b) Table with customers

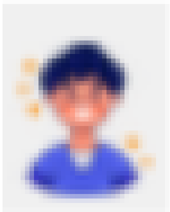
Agents

- Manager

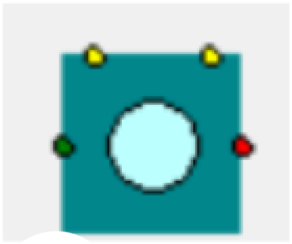
  Assigns responsibilities to the robots

- Customers

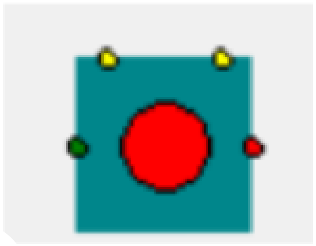
- Robots



(c) Customer



(d) Original



(e) Low Battery Robots

(f) Carrying Food

System Design

Obstacle Avoidance

- Obstacle Avoidance in Customers
  - On collision, customer turns between 25-45 degrees angle towards the target
  - Keeps on turning until it avoids collision.
- Obstacle Avoidance in Robots
  - Robots use different path finding algorithms to find the path towards the target.
  - If it collides with another bot, the path is recalculated to avoid that collision
  - For collision with customers, the bot stops until the customer himself moves out of the way.

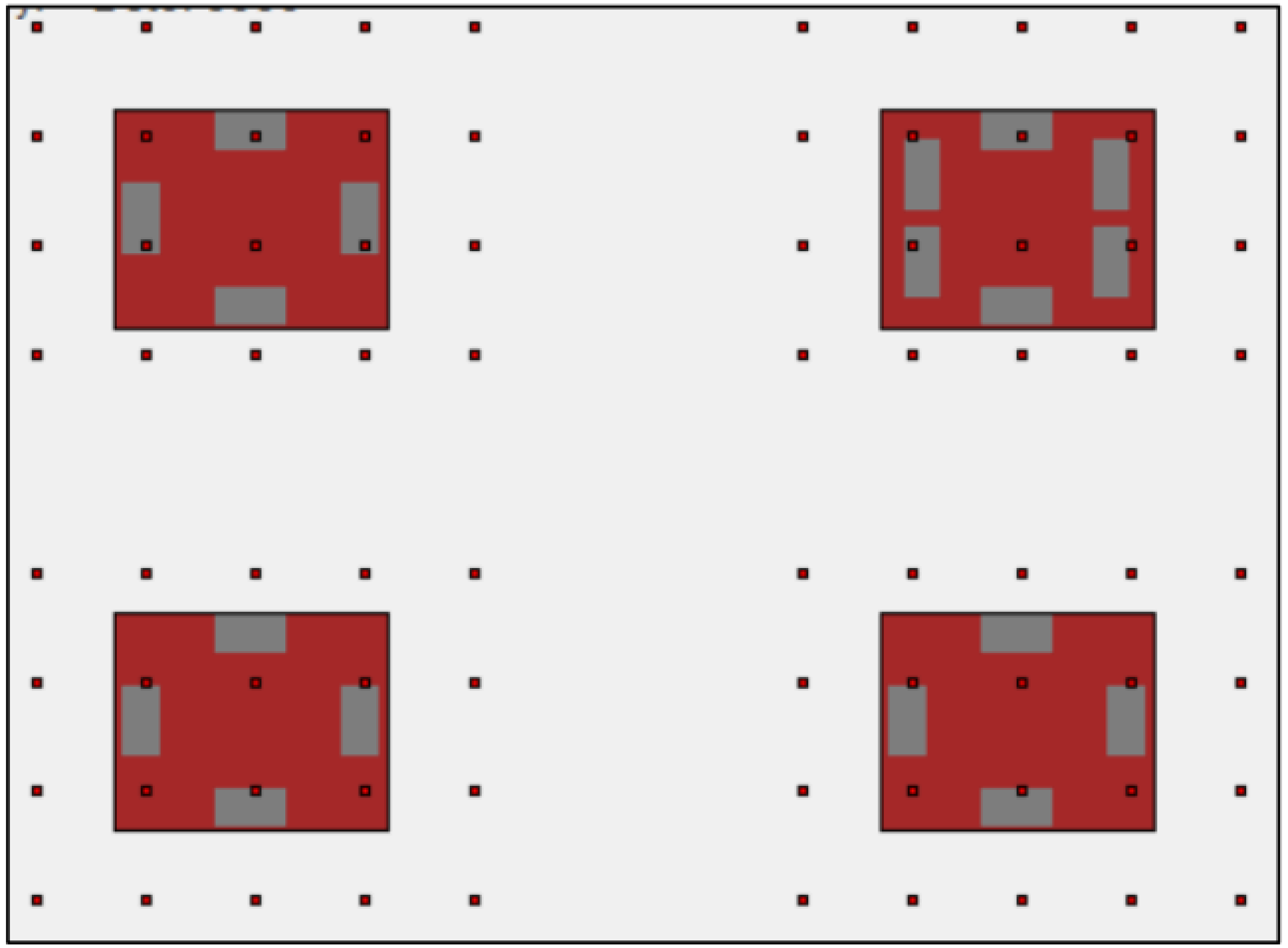


Figure: Grid Cells to Avoid in Path Navigation

Path Navigation

Four different path finding algorithms are used to find the optimal path:

- A* Algorithm
- Dijkstra's Algorithm
- Depth First Search (DFS)
- Breadth First Search (BFS)

Whenever a robot is assigned a task, it calculates the path to the target object while avoiding obstacles.

- The window is divided into a 29 by 19 grid, with each cell of size 40*40.
- The grid cells to avoid are calculated using position and dimensions of obstacles.
- The start cell, target cell and the cells to avoid are provided as parameters to the path finding algorithms.

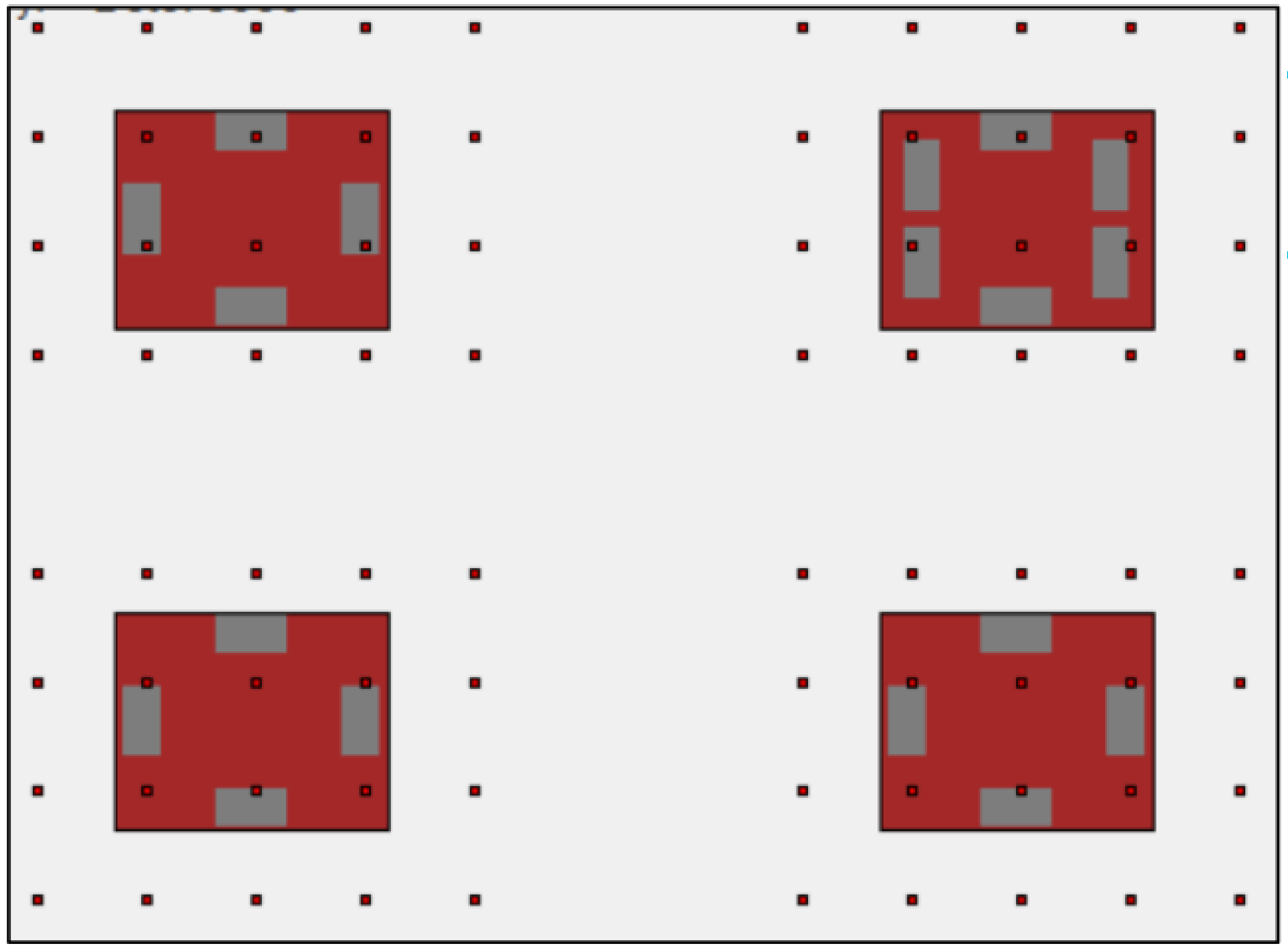


Figure: Grid Cells to Avoid in Path Navigation

Experiment Setup

In order to carry out the experiments, a simulation function was used to run the restaurant environment without displaying the GUI.

- The restaurant simulation was run for 300 seconds each time and the average customer waiting time was returned.
- The simulation was run 10 times for each parameter combination and the average waiting times for each iteration were recorded to get a more accurate result.
- Each of the mentioned path finding algorithms was used one by one.
- The number of bots were varied from 1 to 5 for each path finding algorithm.
- The results were recorded in an excel sheet and box plots and line graphs were created to compare the results.

Experiment Results and Discussion

Effects of Number of Robots on Waiting Time of Customers for each algorithm

Depth First Search

	robots: 1	robots: 2	robots: 3	robots: 4	robots: 5
0	6260.6	6955.526	5205.731	5014.789	4909.808
1	7147.6	6576.143	5205.731	4961.292	4909.808
2	6708.8	6419.455	5205.731	4961.292	4869.338
3	6568.667	6121.49	5205.731	4961.292	4817.6
4	6408.263	5996.608	5205.731	4961.292	4817.6
5	6674.696	5831.944	5151.118	4961.292	4817.6
6	6832.179	5567.517	5061.386	4961.292	4770.224
7	7078.75	5373.081	5061.386	4909.808	4770.224
8	7182	5276.063	5014.789	4909.808	4770.224
9	7102.649	5205.731	5014.789	4909.808	4723.987

Experiment Results and Discussion

Effects of Number of Robots on Waiting Time of Customers for each algorithm

Breadth First Search

	robots: 1	robots: 2	robots: 3	robots: 4	robots: 5
0	3143.043	2629.466	2220.342	2091.511	2043.976
1	2682.947	2585.843	2211.132	2087.886	2031.684
2	2576.167	2545.088	2186	2083.671	2023.092
3	2810.518	2502.816	2170.662	2078.962	2018.825
4	2841.738	2439.221	2170.662	2074.403	2014.651
5	2738.591	2411.084	2165.28	2074.403	2003.225
6	2599.496	2341.293	2146.319	2065.14	1995.005
7	2532.177	2318.248	2141.718	2056.816	1987.639
8	2907.349	2311.66	2100.239	2048.431	1983.418
9	2821.821	2226.442	2095.871	2048.431	1983.418

Experiment Results and Discussion

Effects of Number of Robots on Waiting Time of Customers for each algorithm

Dijkstra's Algorithm

	robots: 1	robots: 2	robots: 3	robots: 4	robots: 5
0	3401.583	3097.311	2135.204	1837.348	1682.468
1	3685.948	3041.942	2085.589	1811.787	1673.008
2	3433.747	2971.832	2038.612	1784.536	1660.32
3	3395.7	2915.02	2003.521	1774.319	1651.052
4	3494.92	2756.065	1983.505	1763.559	1636.357
5	3443.444	2597.566	1968.468	1753.385	1630.036
6	3313	2516.475	1938.738	1748.515	1620.362
7	3150.355	2315.981	1896.954	1721.707	1613.864
8	3094.925	2271.705	1882.96	1695.642	1607.418
9	3135.004	2244.913	1851.398	1689.485	1596.126

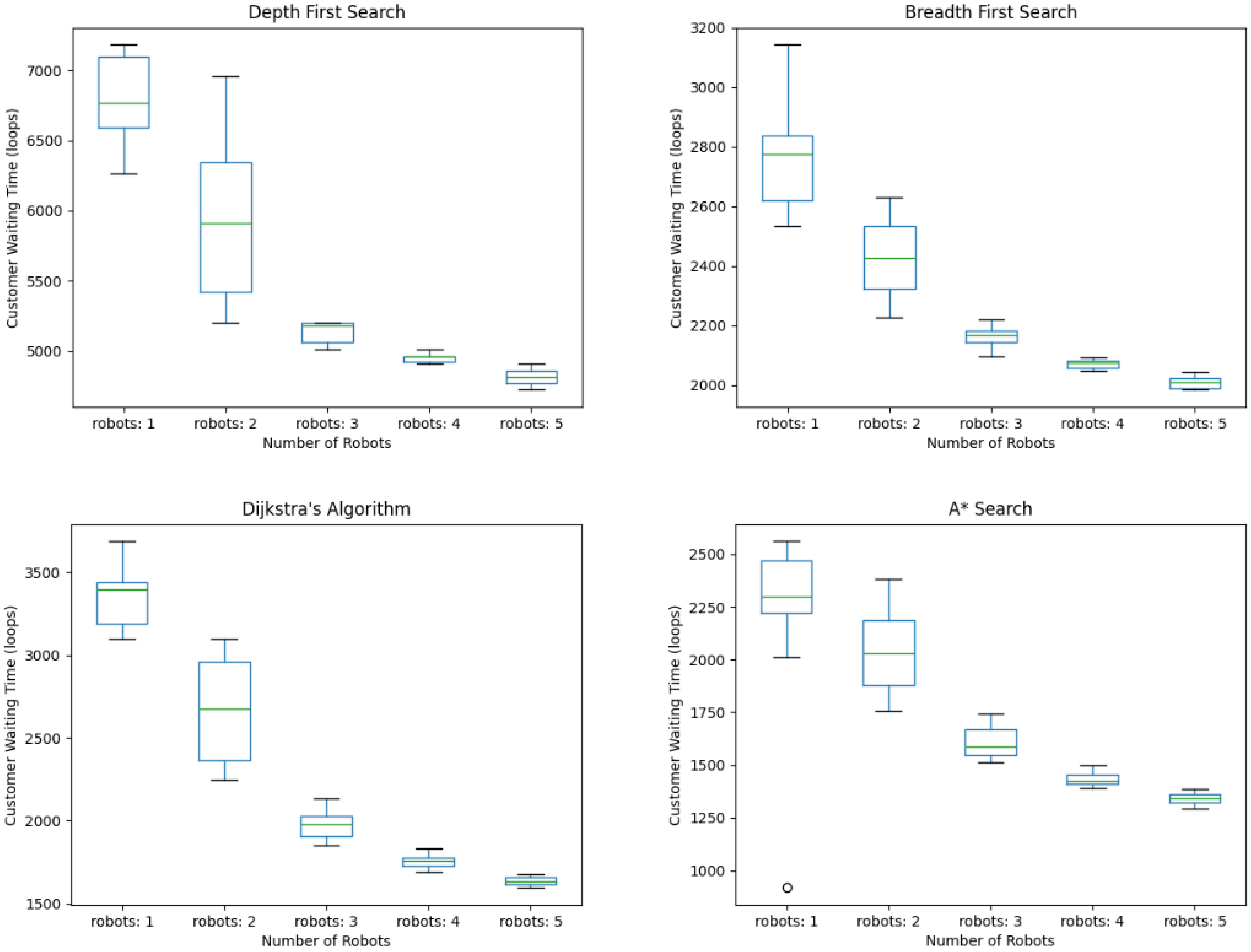
Experiment Results and Discussion

Effects of Number of Robots on Waiting Time of Customers for each algorithm

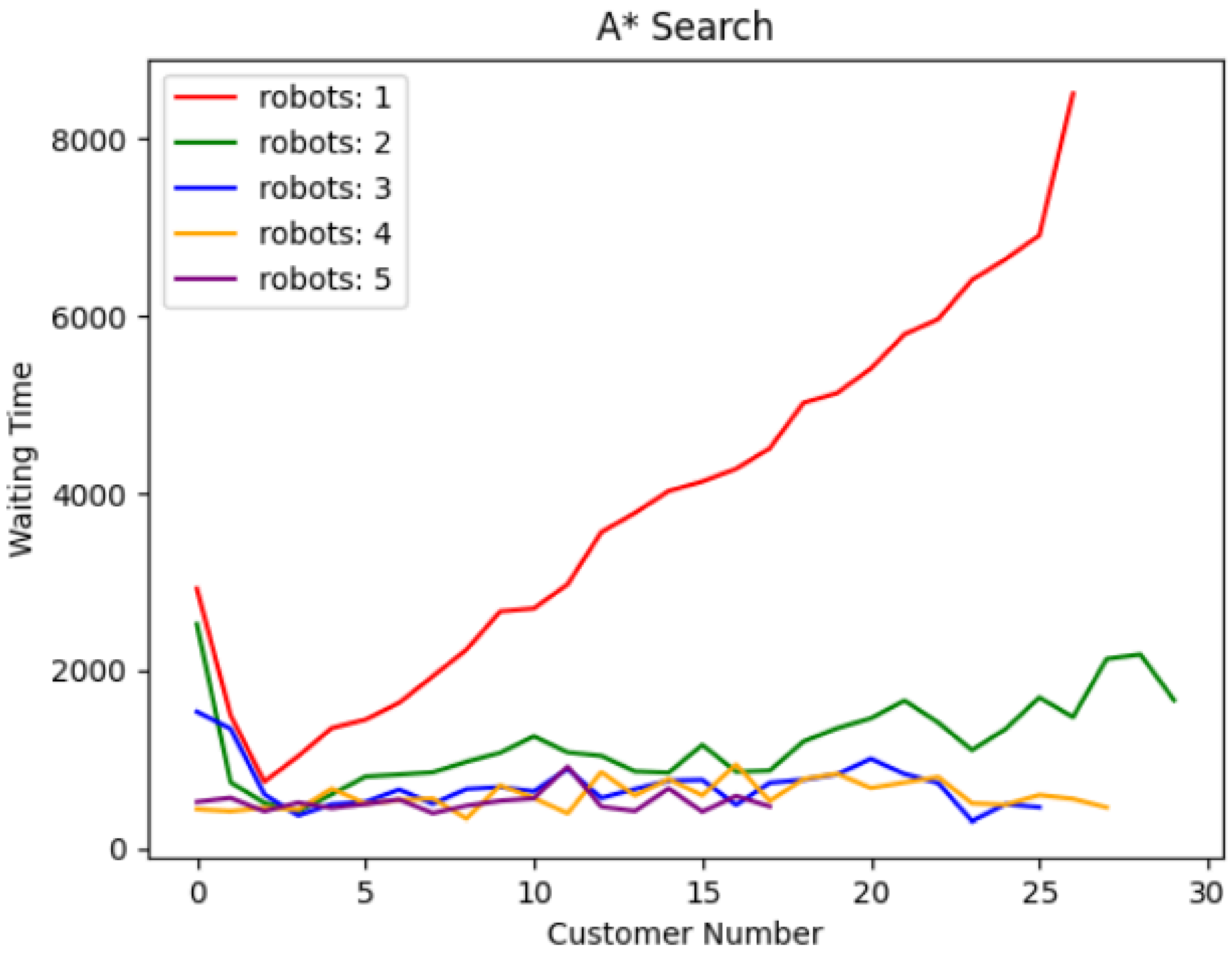
A* Algorithm

	robots: 1	robots: 2	robots: 3	robots: 4	robots: 5
0	919.75	2382.386	1742.434	1499.821	1385.795
1	2313.343	2329.105	1705.27	1469.372	1378.686
2	2013.082	2207.677	1679.367	1459.283	1363.209
3	2411.395	2126.209	1644.022	1438.269	1355.717
4	2283.905	2067.509	1591.059	1430.854	1347.912
5	2235.722	1993.472	1581.434	1420.142	1340.075
6	2215.448	1957.419	1561.168	1412.029	1327.947
7	2516.848	1850.673	1540.27	1407.36	1322.674
8	2561.157	1810.023	1533.446	1400.047	1311.971
9	2489.419	1756.345	1514.555	1390.771	1291.287

Boxplots for Customer Waiting Time vs Number of Robots



Plot for Customer Waiting Time vs Customer Number



Experiment Results and Discussion

Effects of Number of Robots on Waiting Time of Customers for each algorithm

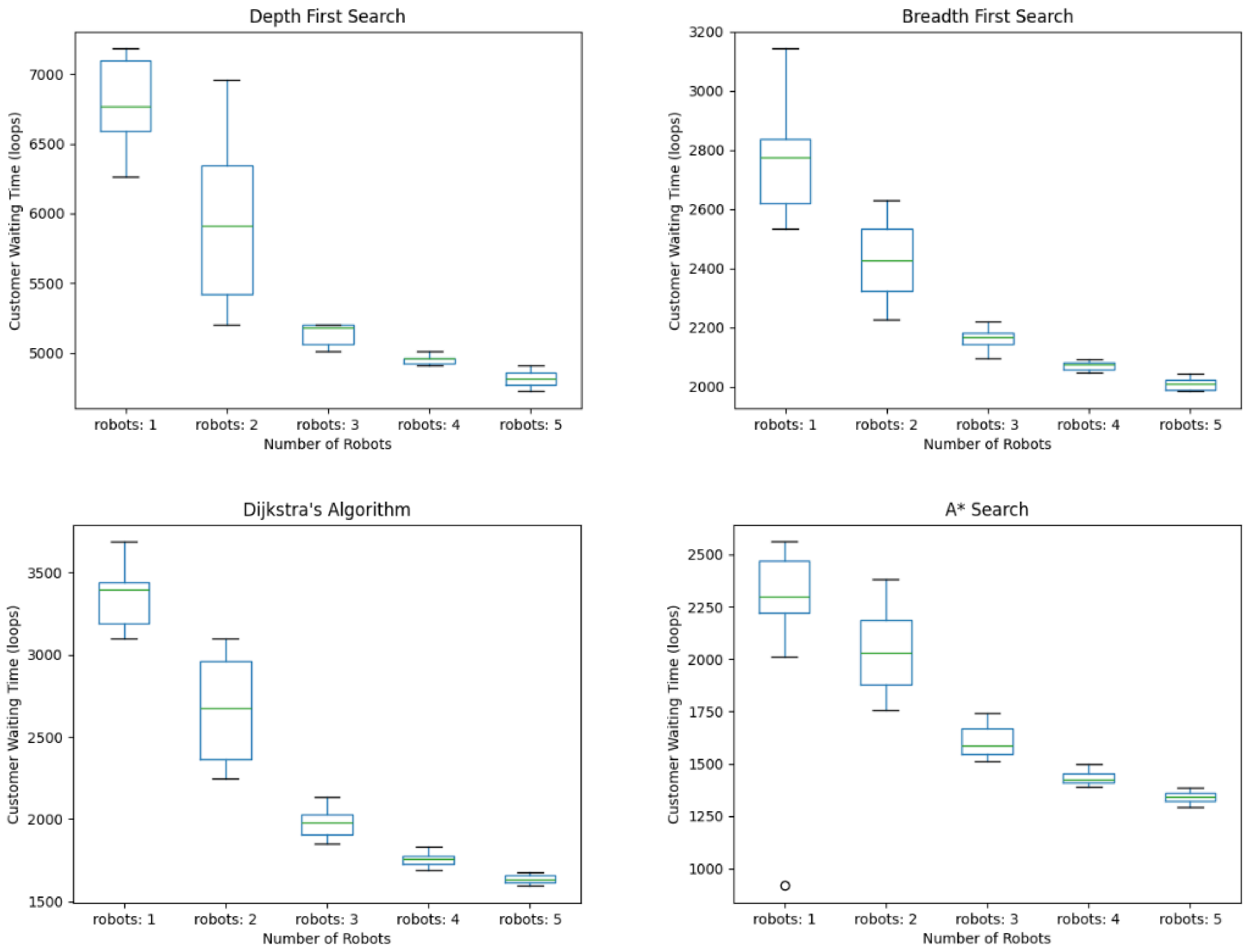
The following observations were made from the discussed plots.

- Waiting time decreases as the number of robots increases.
- The effect of number of robots decreases as we continue to increase the number of bots.
- The variance in the waiting time decreases with the increase in the number of robots.

Experiment Results and Discussion

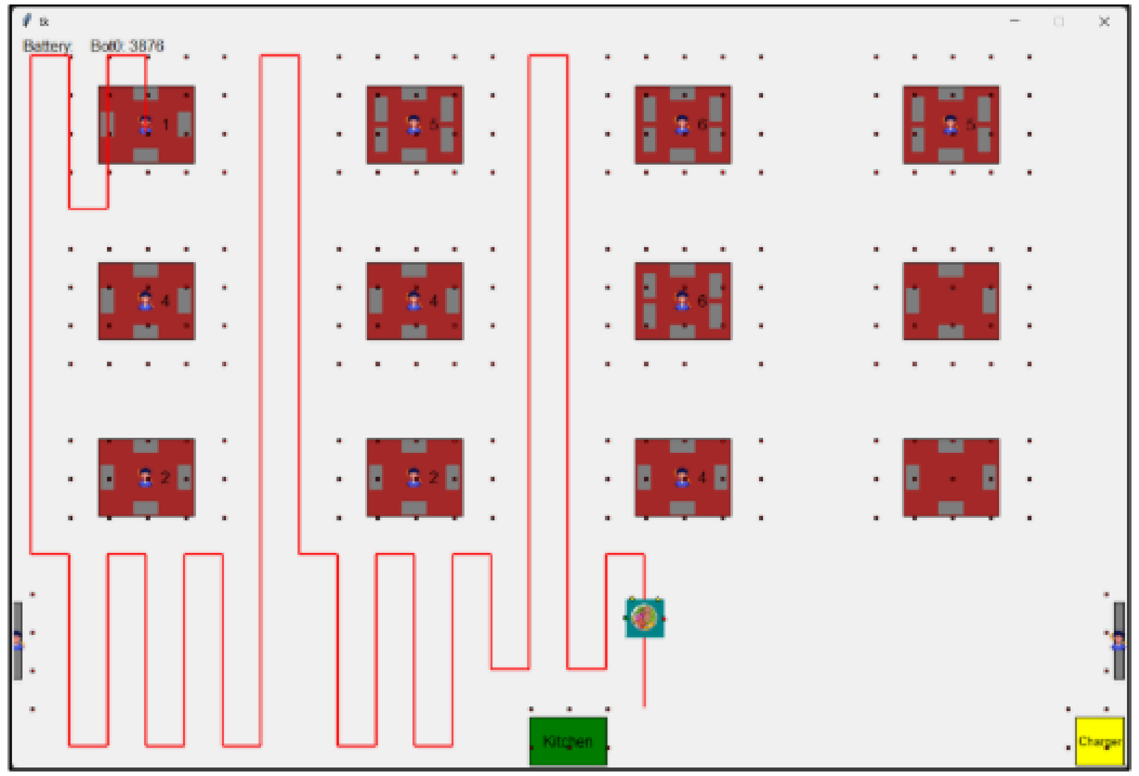
Effects of Path Finding Algorithm on Waiting Time of Customers

- Depth First Algorithm has the highest waiting time.
- BFS and Dijkstra’s algorithm have a similar range of waiting time.
- A* has the lowest waiting time.

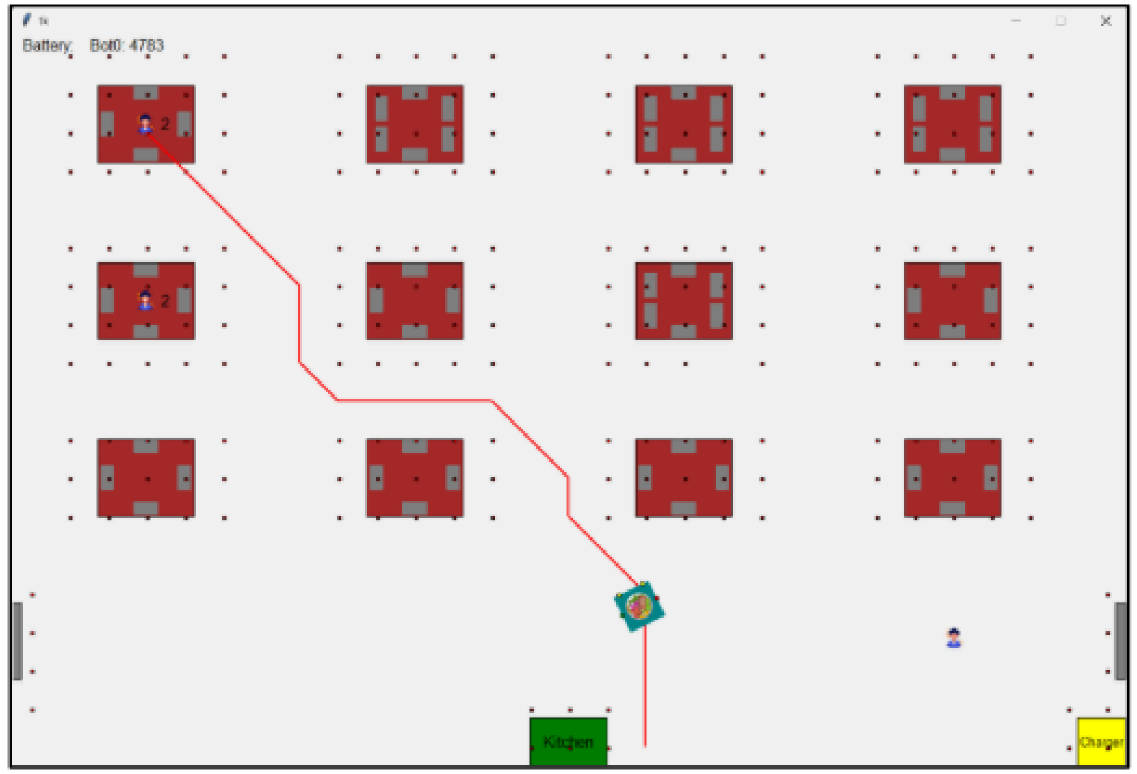


Navigation paths using different algorithms

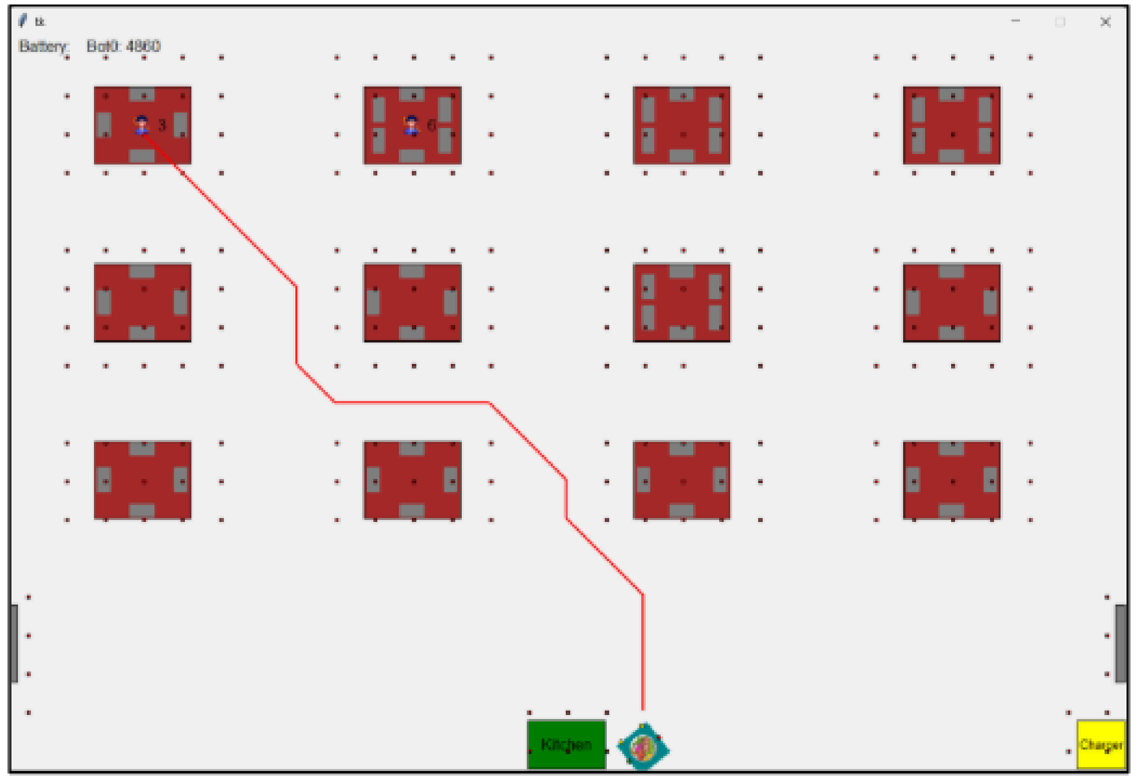
- DFS produces the longest path.
- BFS and Dijkstra's algorithm produce similar paths.
- A* produces the shortest path as it has less turns than DFS and dijkstra's.



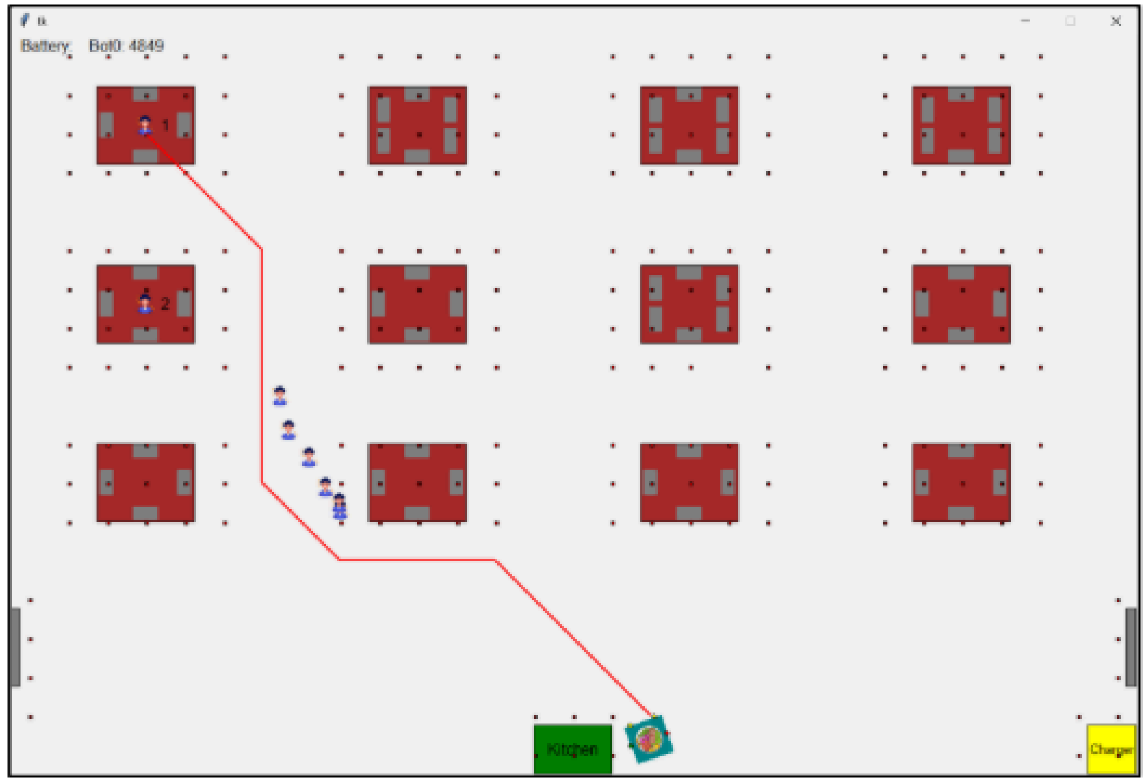
(a) Depth First Search



(b) Breadth First Search



(c) Dijkstra's Algorithm



(d) A* Algorithm

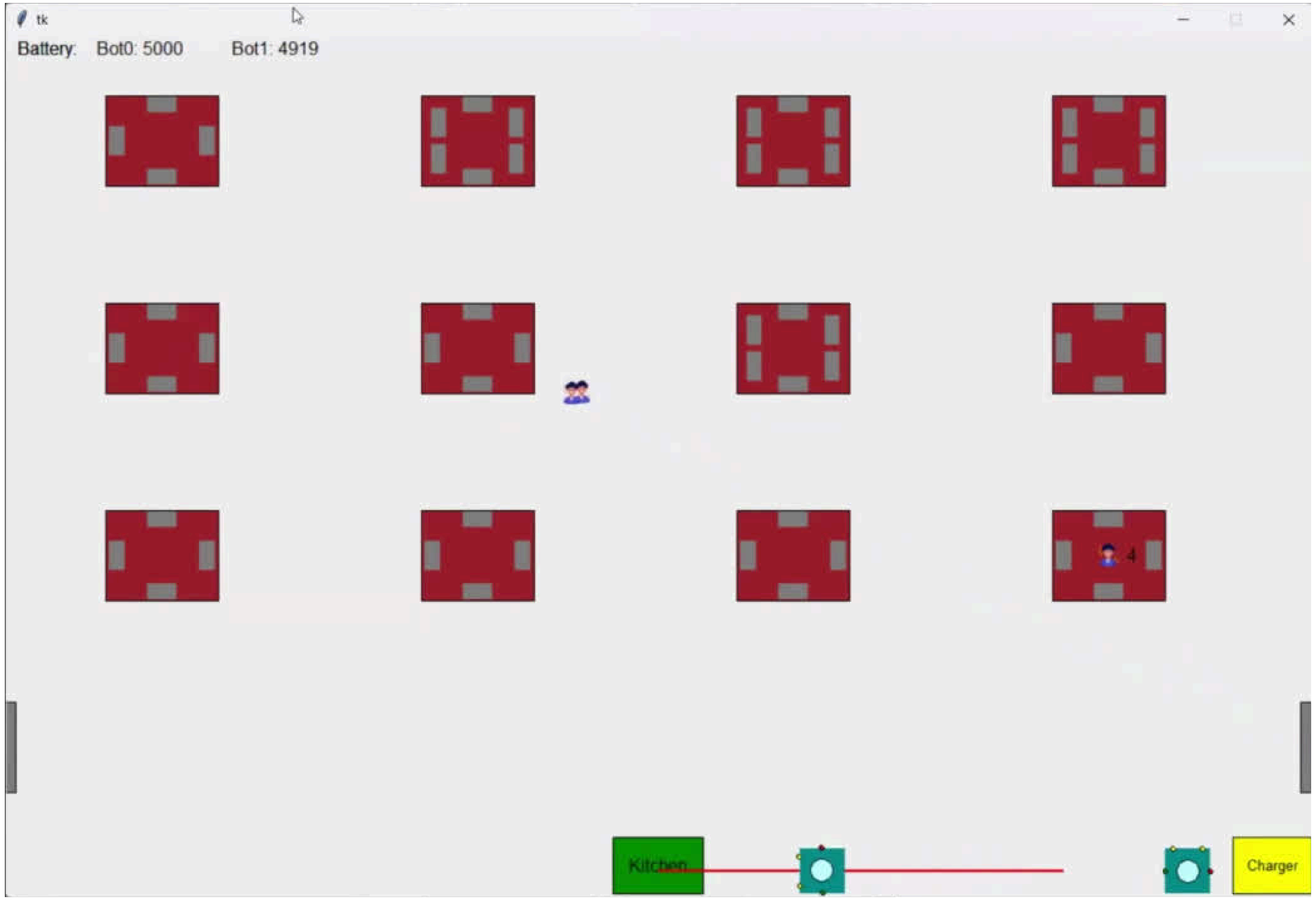
Conclusion

- Increasing the number of robots reduces waiting times, but there's a threshold where more robots offer diminishing returns, emphasizing the need for balance.
- For fewer number of robots, the waiting time increases more sharply for each successive customer.
- Pathfinding algorithms greatly affect waiting times, with A* standing out as the most efficient choice compared to Depth First, Breadth First, and Dijkstra's algorithms.
- A combination of an optimal number of robots and an efficient pathfinding algorithm is essential for enhancing restaurant operations and overall efficiency.

Future Work

- The further work on this project will focus on managing more tasks for the robots like cleaning, guiding the customers etc.
- Further research could explore restaurant layout optimization and dynamic adaption of the algorithms to real-time conditions.
- The future work will also include further improvement in the path finding algorithms
- Currently this report covers four path finding algorithms. Some other algorithms can also be explored in the future.

Demonstration



References

[1] Antony, A., & Sivraj, P. (2018, July 11-12). Food Delivery Automation in Restaurants Using Collaborative Robotics. 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India. https://ieeexplore.ieee.org/document/8597280

[2] Dela Cruz, J. C., Magwili, G. V., Mundo, J. P. E., Gregorio, G. P. B., Lamoca, M. L. L., & Villasenor, J. A. (2016, November 22-25). Items-mapping and route optimization in a grocery store using Dijkstra's, Bellman-Ford and Floyd-Warshall Algorithms. 2016 IEEE Region 10 Conference (TENCON). Singapore. https://ieeexplore.ieee.org/document/7847998

[3] Stentz, A. (1994). Optimal and efficient path planning for partially-known environments. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation (pp. 3310-3317). Volume 4. doi: 10.1109/ROBOT.1994.351061. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=351061

Thank You