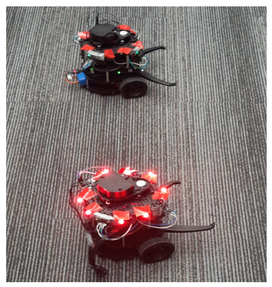
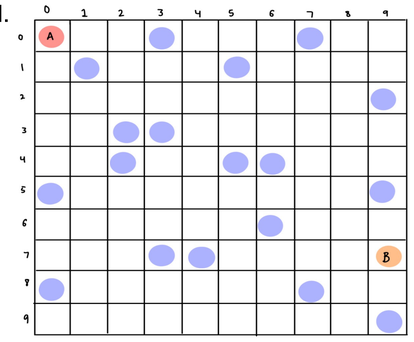
Efficient Warehouse Automation: Evaluating Deep Q-Learning, A * and SARSA Algorithms for Collision-Free Object Movement by Robots

Jhanak Gupta (College of Science and Engineering)
Professor : Maria Gini

Abstract

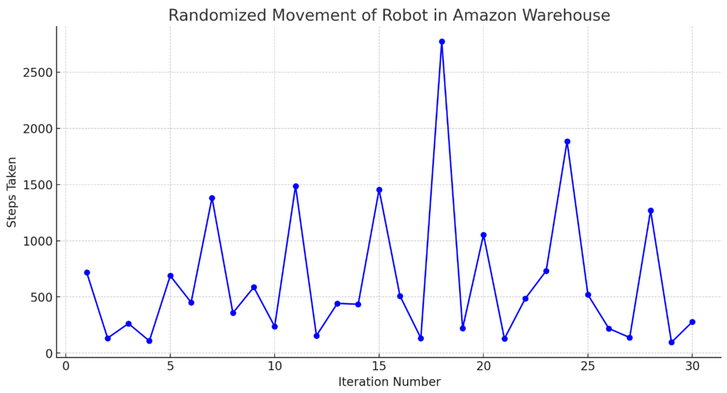
Warehouses are critical hubs for e-commerce businesses, which are experiencing rapid growth. In such environments, the movement of goods demands constant and precise coordination.
My research focused on revolutionizing warehouse operations by reducing the reliance on physical labor for routine tasks. I aimed at automating the process of moving boxes from one location to another. A crucial step in this concept is to first train the robots to move from initial location to the final location in the most effective manner. To make the research realistic, I added several barriers in an Amazon simulated warehouse to see and observe how can different reinforcement learning and reward shaping algorithms improve the process.
I specifically researched on Q-learning, A* algorithm and SARSA (State Action Reward State Action). Comparing different algorithms helped me examine the efficiency growth and training model development of my robot model.





Introduction

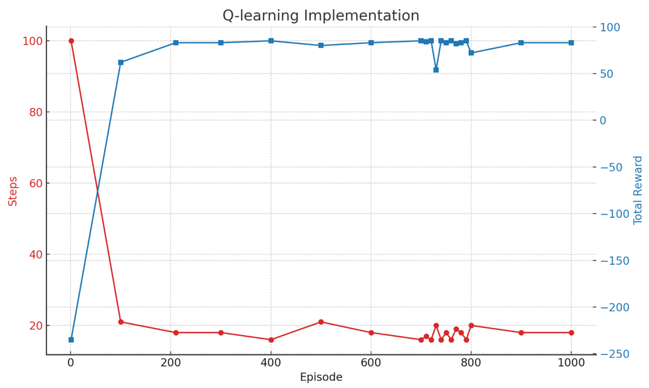
In my research, I methodically approached the challenge of optimizing robot navigation in a warehouse setting by following a structured sequence of steps. I began by establishing a baseline: I calculated the average number of steps a robot took to reach its destination using random movements. This initial data served as the benchmark I aimed to surpass.
With this benchmark in place, I first applied the Q-learning algorithm, specifically an epsilon-greedy strategy, to improve upon the initial random movements. This approach yielded a noticeable improvement in the efficiency of the robot's pathfinding.
To further refine the pathfinding, I then implemented the A* algorithm, renowned for its efficiency in finding the shortest possible path. This allowed me to draw a direct comparison between the optimal path provided by A* and the paths generated by other algorithms.
Finally, I explored the SARSA algorithm to extend the comparison and deepen my understanding of how these different learning methods might converge towards similarly effective solutions. This comprehensive comparison across multiple algorithms provided valuable insights into their relative strengths and capabilities in a practical, warehouse automation context.



Methodology

Q-Learning Applications and Observations :

1.Rapid Initial Learning: There is a steep decline in the number of steps taken by the robot at the beginning of the training. This indicates that the algorithm quickly learned more efficient paths from the start to the goal compared to random exploration.
2.Convergence: After the initial rapid learning, the line representing steps stabilizes, suggesting that the Q-learning algorithm has converged to a policy that consistently finds a path close to the optimal number of steps.
3.R**eward Optimization**: The total reward also shows a significant improvement initially, moving from a large negative value towards less negative (or potentially positive) values. This suggests that as the robot learns to take fewer steps, the penalties for extra steps decrease, leading to higher total rewards.
4.Consistency: In the latter part of the graph (beyond episode 200), the steps taken by the robot are consistently low with minimal variance, and the rewards are consistently less negative. This consistency is an indicator of the stability of the learning process and the effectiveness of the policy being learned.



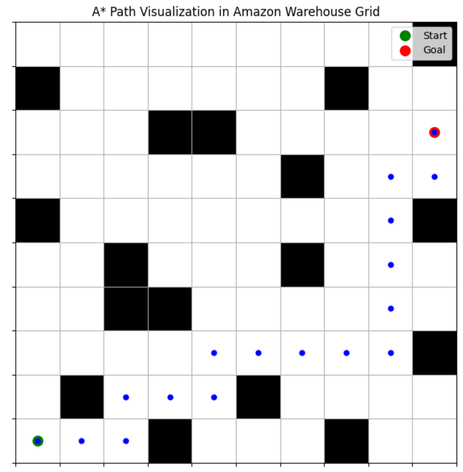
SARSA Applications and Observations:

1.Learning Stability: The median number of steps (18) being lower than the mean (23.508) indicates that the SARSA algorithm has episodes where it performs exceptionally well, but there are also episodes with higher step counts, which raise the average. The presence of these high-step episodes suggests some variability in performance, but the algorithm maintains stability for at least half of the time.
2.Efficiency Improvement: The minimum number of steps required to reach the goal (16) showcases the algorithm's ability to find efficient paths in the best cases. This number is close to the 25th percentile (16) and the median (18) suggests that the algorithm frequently finds near-optimal paths so **more suitable for complex maps**.
3.Performance Consistency: A majority of episodes (75%) requiring no more than 20 steps, as indicated by the 75th percentile, demonstrates that the algorithm consistently performs well, with a relatively small margin between typical (median) and slightly less common (75th percentile) outcomes.
4.Episodic Variability: The large increase in steps towards the 95th percentile (47.05) points to episodic instances where the SARSA algorithm may get stuck or take less efficient paths, potentially due to exploration or challenging scenarios posed by the environment. This is a key area where the algorithm could potentially be improved to reduce such outliers and increase overall robustness.
(Scope of Improvement)

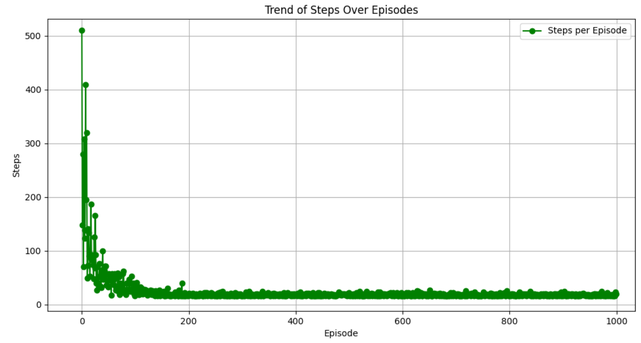
Conclusion

Compared to the deterministic A* algorithm that always finds the shortest path, the SARSA algorithm's performance, driven by exploration and learning, shows that it adapts and performs efficiently mapping the shortest part obtained by the A* algorithm.

Following is a demonstration of the shortest path that A* algorithm derived and the graphical representation of SARSA efficient trials.



A* Algorithm devised the shortest path in 16 steps



SARSA's median step score of 18 consistently close with A*'s 16

The consistency of achieving the 16-step path, especially given the presence of barriers and the need for dynamic decision-making, shows that SARSA is reliable in learning from the environment. The algorithm is not only finding the shortest path but is doing so consistently across a significant number of episodes.

References

1.Stefano V. Albrecht, Filippos Christianos, Lukas Schäfer. "Multi-Agent Reinforcement Learning: Foundations and Modern Approaches", non-final version. MIT Press, Fall 2024
2.Sven Gronauer and Klaus Diepold. "Multi-agent deep reinforcement learning: a survey", Artificial Intelligence Review (2022) 55:895–943 https://doi.org/10.1007/s10462-021-09996-w
3.John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov, "Proximal Policy Optimization Algorithms", arXiv:1707.06347v2 [cs.LG], 2017