Optimizing Robot Navigation: A Comparative Analysis of Shortest Path Algorithms through Graph Deep Reinforcement Learning

ECE5984 Deep Reinforcement Learning – Project Proposal

Timo Thans {timot}, **Sophia Rubsamen** {sophiar@vt.edu}
Department of Electrical and Computer Engineering, Virginia Tech

I. Problem Statement

In this project, we aim to address the critical challenge of developing an algorithm that leverage graph deep reinforcement learning (RL) to find the shortest path to target nodes within a network. Graph cominatorial problems have been studied by computer scientists for decades as these problems impact many fields such as robotic routing in an autonomous warehouse. The traditional shortest path algorithms, while effective in static graphs, often fall short in environments where the graph's topology and edge weights can change in real-time due to obstacles, varying conditions, or updated priorities. Because graph problems are NP-hard, the dynamic nature of the environment makes it much challenging for existing methods to find reasonable solutions.

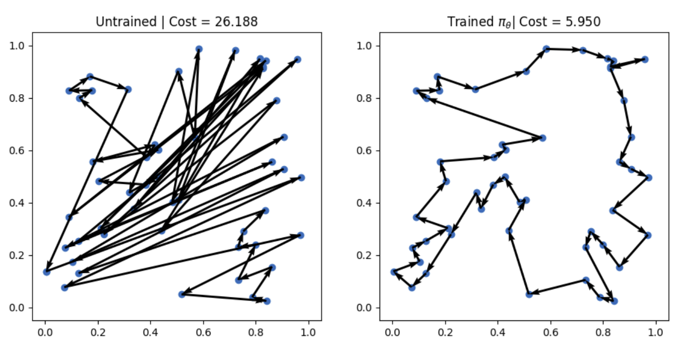

Fig. 1. Trained agent on a Travelers Sales Person problem

By incorporating graph deep RL, our proposed solution aims not only to adapt to such change more efficiently but also to enhance the overall flexibility of the graph. This adaptability is expected to result in more efficient, real-time decision-making processes that can significantly optimize operational efficiency and reduce time-to-target in complex, ever-changing environments.

II. Approaches

We have established three approaches to be worth investigation. Two are deep learning methods, and one is the A-start shortest path algorithm used as baseline.

A-star: is an algorithm to find the shortest path to a target node. It is currently the most efficient algorithm to do this, because it uses heuristics to guide its search. A-star estimates the cost to reach the target form each node and prioritizes the exploration of nodes based on this estimate. This typically leads to faster discovery of the shortest path to the target node, especially in graphs with a large number of nodes or when the target is far from the start node.

AM: the attention model as defined in [1], it can be considered a Graph Attention Network. These apply the attention mechanism (focusing on specific parts of the input data most relevant to the task). The paper describes an attention based encoder-decoder model with a stochastic policy $p(\pi, s)$ for selecting a solution π given a problem instance s. The encoder produces embeddings of all input nodes. The decoder produces the sequence π of input nodes, one node at a time.

T-GCN: The model described in [2] is a temporal graph convolutional network (T-GCN), which combines elements of both a graph convolutional network (GCN) and a gated recurrent unit (GRU). The GCN component captures spatial dependencies within the provided graph data, leveraging graph convolution operations to learn representations of nodes based on their neighboring nodes' information. The GRU component learns dynamic changes in the environment, enabling the model to effectively capture temporal dependencies over time.

III. Expected Experimental Results

Our selected algorithms will initially undergo testing and performance evaluation within established benchmark environments, such as RL4CO [3], as depicted in Figure 1. By utilizing a straightforward grid-world environment for our preliminary assessments, we aim to discern the performance disparities across various methodologies, including a comparison with a conventional non-deep learning baseline. To align our benchmarks with our objectives, we will modify the test conditions not only to address the Traveling Salesperson Problem (TSP), but also effectively tackle the shortest path problem. Within the benchmark setting, we plan to introduce variations to the layout to evaluate the algorithm's adaptability and scalability. These modification may include the introduction of strategic obstacles and stochastic elements within the graphs. Furthermore, we intend to experiment with denser graphs to enhance the precision of the algorithm.

We anticipate that in both the Traveling Salesperson and Shortest Path Problems, Graph Neural Network (GNN) agents should perform comparably to A* agents for small instances. As the complexity of the problems increases, we expect GNN agents to outperform A* agents due to their ability to handle more complex features. Moreover, we anticipate that both graph neural network models will surpass the performance of A* agents in dynamic environments. GNN agents can adapt their predictions based on real-time data, whereas A* agents rely on precomputed heuristics.

Fig. 2. PrimeVision Warehouse Robot

Upon successful completion of the initial trials, we anticipate progressing to deploy our algorithm within the PrimeVision robotic simulation platform, as illustrated in Figure 3. Within this advanced environment, the robots are designed to adhere to 'feasible trajectories' – these are pathways deemed executable by the robots, derived from the optimized graph results.

To integrate seamlessly with robots' control systems, it will be necessary to tailor the graph outputs. This entails estimating the robots' internal physics models and devising a meethod to translate the optimal graph pathways into a series of feasible waypoints for the robots. However, the inclusion of this phase in our project is contingent upon its alignment with the projected timeline and scope. Additionally, we must consider the extend to which this phase contributes to our primary research goal: the application of deep learning in solving graph-based problems.

IV. Potential Contributions

The aim is to compare and develop the most efficient version of a shortest path finding algorithm using graph deep reinforcement learning (RL) within an autonomous system, like the PrimeVision robots navigating a warehouse, could present multiple contributions beyond improving the efficiency of the warehouse robots.

- **Adaptability**: dynamically adjusting routes in response to obstacles, changing targets, or evolving conditions within the warehouse.

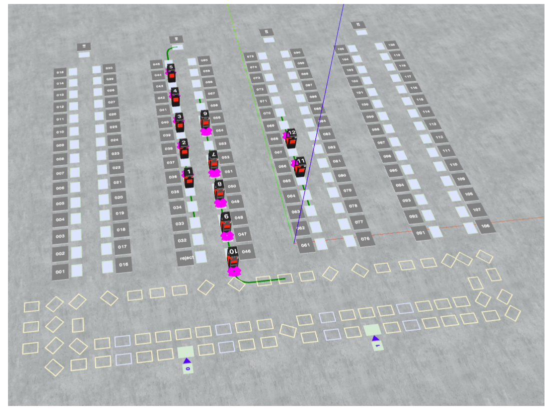

Fig. 3. PrimeVision Robotic Simulation Environment

- **Scalability**: exploring how the algorithms scale with the complexity of the network would be an interesting open problem to research. Perhaps, even handling many robots without a significant loss in performance or efficiency.
- **Real-Time Learning and Improvement**: demonstrating how these algorithms learn and impove over time, adapting to operational patterns to optimize future pathfinding tasks. This could involve learning from past obstructions or identifying frequently traveled paths that may benefit from optimization.
- **Human-Robot Interaction (HRI) Efficiency**: analyzing and enhancing the efficiency and safety of human-robot interactions could be a noteworthy contribution. Optimizing the robots' paths to reduce human interference and improve collaboration.

References

[1] Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! 2019.

[2] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[3] Federico Berto, Chuanbo Hua, Junyoung Park, Minsu Kim, Hyeonah Kim, Jiwoo Son, Haeyeon Kim, Joungho Kim, and Jinkyoo Park. RL4CO: a unified reinforcement learning for combinatorial optimization library. 2023. https://github.com/ai4co/rl4co.