Reinforcement Learning-Based Decentralized Multi-Agent Pathfinding Project Report

Avaniko Asokkumar

Department of Mechanical Engineering, College of Design and Engineering

National University of Singapore

Special Term Part 1, 2024

ABSTRACT

This research addresses the challenges of Multi Agent Pathfinding (MAPF) in autonomous navigation, focusing on improving the Attention-based Long-horizon Pathfinding in Highly-structured Areas (ALPHA) reinforcement learning model. Traditional MAPF algorithms often struggle with scalability and redundancy in path generation, particularly as the number of agents increases. This project introduces a novel approach utilizing homotopy detection and classification to identify topologically distinct paths in a structured two-dimensional grid. By implementing an algorithm that classifies paths into Uniform Visibility Deformation (UVD) classes, the methodology effectively reduces redundant paths and distinguishes critical nodes from noncritical ones, enhancing computational efficiency. This system also identifies potential congestion areas, aiding in collision avoidance. Initial findings indicate that this approach can significantly optimize runtime and improve scalability, making it applicable to complex environments such as autonomous vehicles and warehouse robotics. While further integration with reinforcement learning training is ongoing, the results suggest a promising framework for more efficient and scalable MAPF solutions. This research contributes to the advancement of autonomous navigation technologies, offering a robust method for managing multi-agent interactions and pathfinding in both structured and potentially unstructured environments.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my lab professor, Dr. Guillaume Sartoretti, for his invaluable guidance, support, and encouragement throughout this research project. I am also profoundly grateful to my PhD candidate mentor, Chengyang He, for his mentorship, insightful feedback, and continuous assistance. My heartfelt thanks go to all the members of the MARMoT lab for their collaborative spirit and contributions, which have greatly enriched this research experience. Additionally, I would like to thank the National University of Singapore for providing the resources and environment necessary for this research.

TABLE OF CONTENTS

I. INTRODUCTION 3

II. METHODOLOGY 6

III. FINDINGS 9
IV. DISCUSSION 11
V. CONCLUSION 12
VI. REFERENCES 14
VII. APPENDIX 15

I. INTRODUCTION

Autonomous navigation can be broken down into three main components: perception, planning, and controls. Perception involves how autonomous agents perceive the environment around them

through sensors and external inputs. Planning deals with the decision-making process that determines the high-level or low-level trajectory the agent will take. Controls focus on controlling actuators and motors to stably follow the generated trajectory. This report will focus on a subset of planning: Multi Agent Pathfinding (MAPF).

While there is abundant prior work related to single-agent path planning, MAPF is a problem increasing in importance, with applications in autonomous automobiles, search-and-rescue missions, and warehouse robots. For autonomous vehicles, low-level tasks such as lane switching and highway ramp merging, as well as high-level tasks like route planning around traffic congestion, all rely on some form of path planning and could be improved if integrated into a multi-agent system. Search and rescue robots often have to explore unknown or partially known environments and coordinate complex operations within them. Warehouse robots deal with highly-structured, known environments, but have their own set of challenges.

The main issue that differentiates multi-agent systems from single-agent systems is the possibility of agent-agent interactions. In multi-agent path finding, the primary problem that arises from these interactions is having to solve collisions, situations in which two or more agents attempt to occupy the same space at the same time [1].

There are certain deterministic approaches to the multi-agent path planning problem, most notably the M* algorithm [2]. While this does yield a relatively efficient, but complete and fully optimal solution, it lacks scalability in systems with large numbers of agents. Other traditional algorithms for multi-agent path planning often scale poorly as the number of agents increases [3].

This project focuses on a reinforcement learning (RL) solution to MAPF. Reinforcement learning is a form of machine learning that works similarly to the concept of operant conditioning in

psychology, where agents take certain actions and are given a reward in return. The reward can either be positive or negative, with positive rewards encouraging certain actions and negative rewards deterring them. Reinforcement learning also incorporates the state of the agent, as a certain action in one state might be more beneficial than if done in another state. Upon training an agent in an environment with rewards, the agent learns a set of actions to take based on each possible state, known as a policy. This policy can be applied to environments and situations different than what it was trained on, which allows for scalability in both environment size and agent number.

ALPHA

The focus of this paper will be on an improvement to the Attention-based Long-horizon Pathfinding in Highly-structured Areas (ALPHA) RL model [3].

ALPHA is an extension of a previous work, Pathfinding via Reinforcement and Imitation Multi-Agent Learning (PRIMAL). PRIMAL was a similarly structured RL approach to MAPF, but the main constraint it was bound by was inability to train on global structure as a feature [4]. Agents training in PRIMAL only had access to local information within a certain field of view (FOV), but ALPHA solves this problem by introducing the global environment as a sparse graph, which can be converted into a feature that the model can train on [3].

During the project, I focused on a method to improve the performance of ALPHA by removing redundancy in path and graph generation and introducing a new characterization of the global state representation. Specifically, I implemented an algorithm for finding topologically distinct paths from source to target given a graph representation of a structured environment and an obstacle map.

II. METHODOLOGY

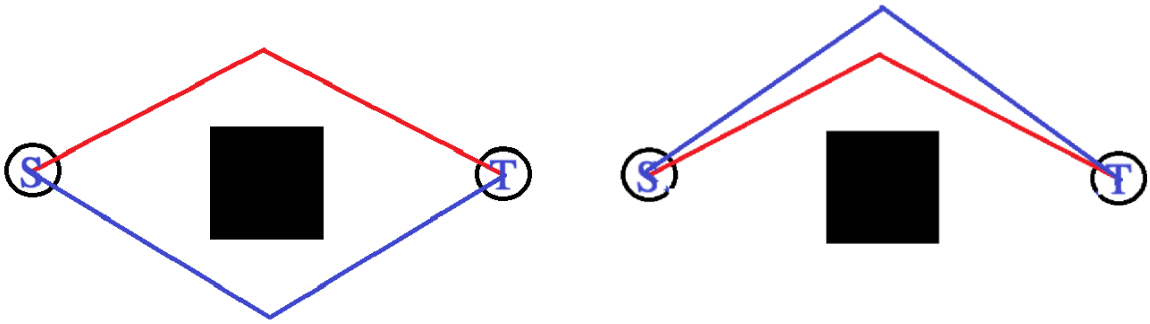
EXISTING METHODS

There are several ways to perform homotopy classification in both 2D and 3D space. Bhattacharya et al. have demonstrated using Cauchy's Residue Theorem to classify trajectories into homotopy classes in 2D [5]. A more straightforward, but computationally intensive approach is the use of Uniform Visibility Deformation (UVD) classes, which has been explored in various studies, including RAPTOR and CTopPRM [6][7]. A simpler version of a Visibility Deformation class, UVD classes are classes of trajectories that are topologically distinct from each other and generated through discretization of paths and checking of interference with obstacles[6].

PROJECT METHODOLOGY

The goal of this project was to generate topologically distinct paths from a source to a target.

The specific iteration of the MAPF problem addressed by this model is of a structured two-dimensional grid, with free space and obstacles, such that agents can only travel through free space, and are unable to pass through obstacles. In a two-dimensional space, topologically distinct paths are paths that cannot be deformed into each other without passing through or interfering with obstacles. If one were to imagine a source node on the left side of a map, a target node on the right side, and a finite area obstacle between them, it can be said that there are two topologically distinct paths from source to target. The first path would be to go clockwise about the obstacle, and the second path would be to go counterclockwise about the obstacle. These two are distinct because one cannot smoothly deform the clockwise path into the counterclockwise path without passing through or interfering with the obstacle in the middle.



1a 1b

In (a) the two paths are topologically distinct, whereas in (b) the two paths are topologically identical

Existing algorithms requiring homotopy identification that utilize Uniform Visibility Deformation (UVD) classes often employ some form of Probabilistic Roadmap (PRM) or custom graph generation to form an initial graph [6]. However, due to the randomized nature of these generation techniques, further pruning is required in order to construct a sparse graph.

The motivation behind using this algorithm for detection relates to removing sources of redundancy in path planning. If two paths are considered topologically identical despite traversing through different nodes, then considering these paths as distinct during training is inefficient in certain cases. Classification of nodes as critical and noncritical can also be done to assist in identifying possible collisions just based on the map rather than being time and state dependent.

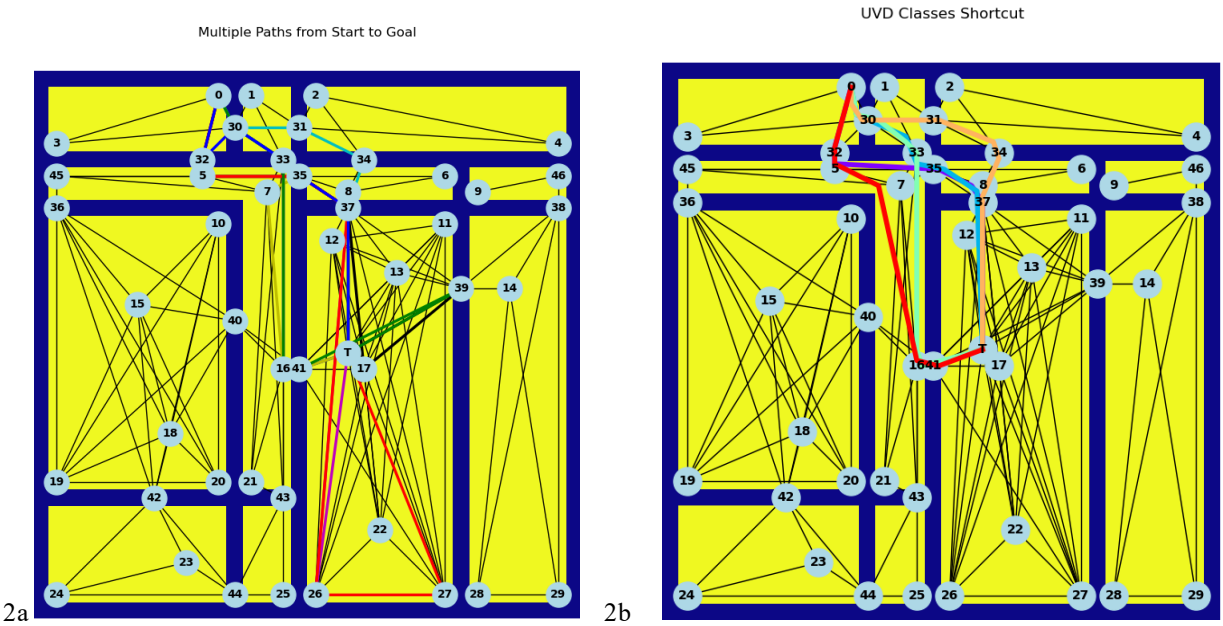
The implementation of used the following structure: sample a finite, non-exhaustive number of paths from the original graph, classify these sampled paths into UVD classes using the algorithm from RAPTOR, and return the set of paths and nodes associated with these paths to be packaged into features usable for RL training.

In existing methods, the use of PRM or other graph generation methods allowed for collision-free paths. Since the graph generated by ALPHA is an abstract representation, distinct from a motion-planning graph used in RAPTOR, preprocessing is done to the graph to eliminate any intersections

between original BFS-generated paths and the obstacles in the map, while still preserving the shape of the map.

The two canonical graph search algorithms, Breadth-First Search (BFS) and Depth-First Search (DFS) were studied for the initial search. Due to the nature of the graph representation not containing explicit information about the obstacles present, there is no simple heuristic that would allow for the generation of more topologically distinct paths. Hence, a sampling was used from BFS. The use of BFS in the original sampling allows for greater divergence in paths due to the graph structure, as opposed to what could be produced through DFS. Sampling was used rather than a fully exhaustive set of paths to optimize runtime.

These paths are passed to a C++ program that runs the computationally heavy UVD classification algorithm, first generating topologically shortened paths, then classifying them based on similarity of points. These classes are then returned to the Python program, where the importance of each node is evaluated by assigning a "density value," where the number of occurrences in each UVD class is divided by the number of total paths. This way, nodes extraneous to a certain topologically distinct class have low density values, and nodes necessary to the class have a value of 1.



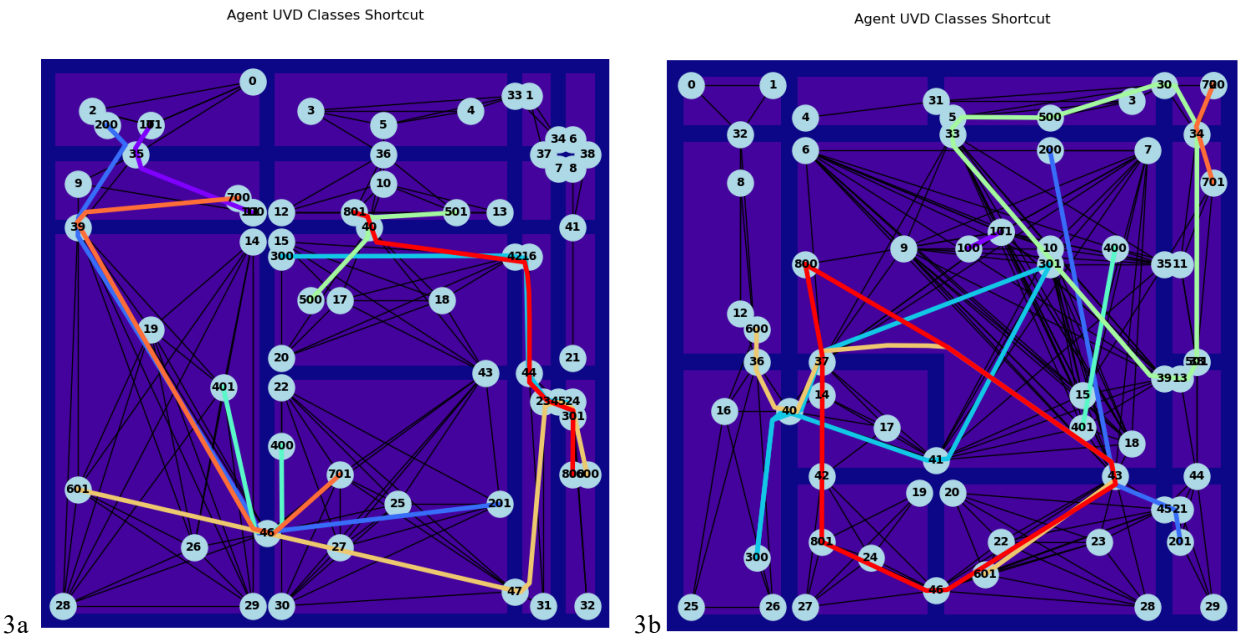
2a 2b

Figure 2: The above figure is for a single agent calculation. (a) shows a sampling of 10 distinct paths from source to target (node 0 to node 'T'), and (b) shows the 5 topologically distinct paths that the initial sampling was classified into.

III. FINDINGS

TRAINING OPTIMIZATION

While the integration of this program with the RL training in ALPHA is still in progress, it is expected that this will have an improvement on runtime, due to the low time cost of the homotopy detection algorithm.



3a 3b

Figure 3: The following maps show the multiagent iteration of the algorithm, where agent starts are marked by nodes in the 100s, and agent goals are marked by nodes in the (100s) + 1. (a) shows a scenario where most agents only have one distinct homotopy class, whereas (b) shows a scenario where most agents have multiple distinct homotopy classes. Each color represents paths belonging to a specific agent.

A main concern with graph generation in both structured and unstructured environments is the complexity of the graph. A dense graph with more nodes and edges is computationally more expensive. However, due to the ability of this homotopy detection to identify critical and noncritical nodes, the graph can be further pruned to ignore noncritical nodes, and observe areas of possible congestion within the graph in a multi-agent scenario.

CONGESTION

A point of interest with decentralized MAPF is identifying places where collisions are likely to occur, and the concept of congestion is pertinent to this point. Congestion describes how crowded with agents a certain area or individual cell might be, whether it is across all timesteps, or within a short finite horizon.

Identifying topologically distinct paths from source to target for each agent allows the idea of congestion to be transformed into a feature based on the overlap of paths.

IV. DISCUSSION

The results of this project suggest significant potential for improving the efficiency and scalability of Multi Agent Pathfinding (MAPF) through the use of homotopy detection and classification. By integrating the concept of Uniform Visibility Deformation (UVD) classes with the ALPHA RL model, the methodology outlined in this report addresses key issues in traditional MAPF algorithms, such as redundancy and computational complexity.

One of the primary advantages of this approach is its ability to reduce redundant path generation. Traditional graph generation methods, like Probabilistic Roadmaps (PRMs), often require extensive pruning to eliminate superfluous paths. However, by identifying and classifying paths into topologically distinct classes early in the process, the algorithm significantly minimizes the number of paths that need to be considered during training. This not only reduces the computational burden but also enhances the efficiency of the learning process for the RL model.

The implementation of a density value system further refines this approach by distinguishing critical nodes from noncritical ones. This differentiation allows for a more focused and efficient use of computational resources, ensuring that the RL model trains on the most relevant parts of the graph. Moreover, the ability to identify areas of potential congestion through path overlap provides a valuable feature for managing multi-agent interactions. This can be particularly useful in scenarios where collisions are likely to occur, enabling better coordination and collision avoidance strategies.

Another significant outcome of this project is the potential for scalability. The methodology's reliance on sampling rather than exhaustive path generation ensures that it can handle larger and more complex environments without a proportional increase in computational requirements. This scalability is crucial for real-world applications, such as autonomous vehicles and warehouse robotics, where the number of agents and the complexity of the environment can vary significantly.

However, there are limitations and areas for further research. While the integration with RL training shows promise, the full impact on runtime and efficiency is still under investigation. Future work should focus on optimizing this integration and evaluating its performance in diverse environments. Additionally, the current implementation assumes a structured two-dimensional grid, which may not fully capture the complexities of unstructured or three-dimensional spaces. Extending the methodology to handle these cases would be a valuable next step.

V. CONCLUSION

In conclusion, this research demonstrates a novel approach to improving Multi Agent Pathfinding (MAPF) by integrating homotopy detection and classification with reinforcement learning. The methodology effectively reduces redundancy and computational complexity by identifying topologically distinct paths and distinguishing critical nodes. This approach not only enhances the efficiency of the ALPHA RL model but also provides valuable features for managing multi-agent interactions and congestion.

The findings suggest significant potential for scalability, making the methodology suitable for a wide range of real-world applications. While further research is needed to fully optimize the integration with RL training and extend the approach to more complex environments, the results of this project represent a promising step towards more efficient and scalable MAPF solutions.

Overall, this research contributes to the advancement of autonomous navigation and pathfinding technologies, offering a robust framework for addressing the challenges associated with multi-agent systems. By continuing to refine and expand this methodology, future work can further enhance the capabilities and applications of autonomous agents in various fields.

VI. REFERENCES

[1] Stern, R., Sturtevant, N. R., Felner, A., Koenig, S., Ma, H., Walker, T. T., ... & Barták, R. (2019). Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks. *Symposium on Combinatorial Search (SoCS)*. Retrieved from https://arxiv.org/pdf/1809.03531

[2] Wagner, G., & Choset, H. (2011). M*: A complete multirobot path planning algorithm with performance bounds. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3260-3267. https://doi.org/10.1109/IROS.2011.6095022

[3] He, C., Yang, T., Duhan, T., Wang, Y., & Sartoretti, G. (2023). ALPHA: Attention-based Long-horizon Pathfinding in Highly-structured Areas. *arXiv preprint arXiv:2310.08350*. Retrieved from https://arxiv.org/pdf/2310.08350

[4] Sartoretti, G., Kerr, J., Shi, Y., Wagner, G., Kumar, T. K. S., Koenig, S., & Choset, H. (2019). PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 3652-3659. Retrieved from https://arxiv.org/pdf/1809.03531

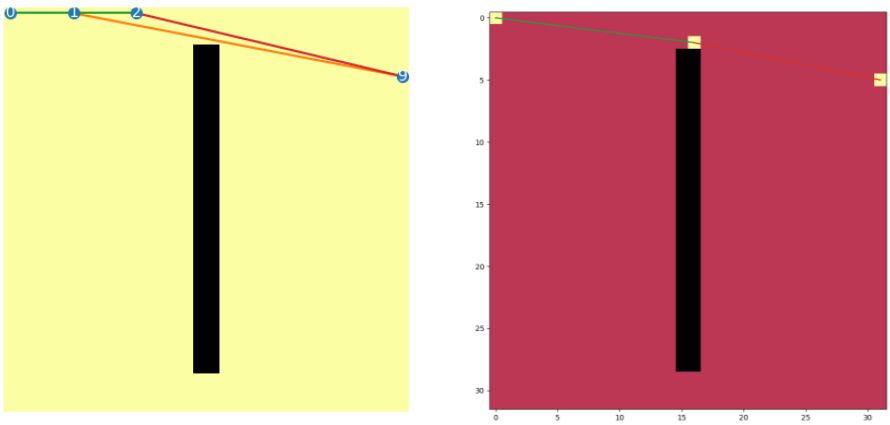
[5] Bhattacharya, S., Likhachev, M., & Kumar, V. (2011). Identification and Representation of Homotopy Classes of Trajectories for Search-based Path Planning in 3D. *Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania*. Philadelphia, PA 19104.

[6] Zhou, B., Pan, J., Gao, F., & Shen, S. (2020). RAPTOR: Robust and Perception-aware Trajectory Replanning for Quadrotor Fast Flight. *arXiv preprint arXiv:2007.03465v1*. https://doi.org/10.48550/arXiv.2007.03465

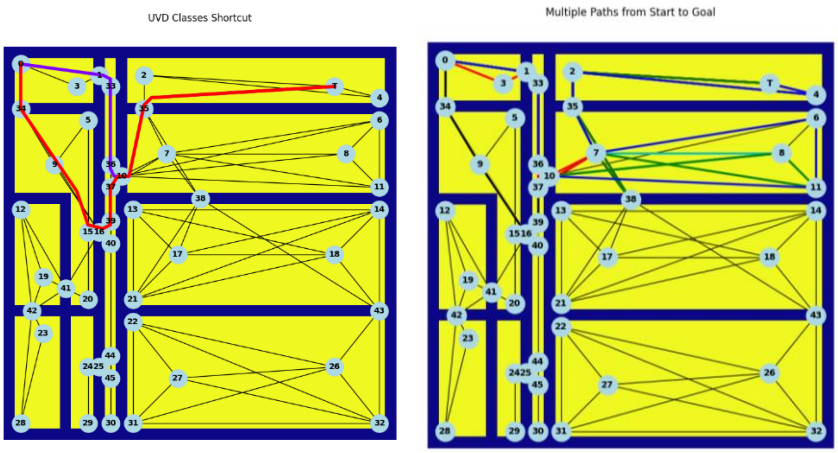
[7] Novosad, M., Penicka, R., & Vonasek, V. (2023). CTopPRM: Clustering Topological PRM for Planning Multiple Distinct Paths in 3D Environments. *IEEE Robotics and Automation Letters*. Advance online publication. https://doi.org/10.48550/arXiv.2305.13969

VII. APPENDIX

A: Example of path convergence between two topologically distinct paths (0, 2, 9) and (0, 1, 9) by UVD classification algorithm.



B: Multiple paths converging into 2 distinct homotopy classes



C: Multiple paths converging into 3 distinct homotopy classes