Project 1: Study HTN Heuristics

Your Name
Student ID: XXX

June 28, 2024

1 Introduction

Hierarchical Task Network (HTN) [8] planning represents a sophisticated approach within artificial intelligence for automating complex tasks by breaking them down into simpler, manageable subtasks [4]. Unlike traditional automated task planning methods that focus on state-space search, HTN planners utilize a hierarchical structure, allowing for a more structured and human-like way of organizing and executing tasks. This hierarchical approach enables HTN planners to handle a wide range of applications, from robot mission planning and AI decision-making in games to autonomous driving path planning [9].

The development of HTN planning stems from the need to manage increasingly complex tasks in both industrial and everyday applications. Traditional planning methods often struggled with scalability and flexibility, leading to the advent of HTN planning which integrates domain knowledge and decomposition rules to efficiently tackle complex problems. By leveraging high-level actions and refining them into primitive actions, HTN planning offers a robust framework for automated task execution.In recent years, research in HTN planning has seen significant advancements, particularly with the integration of heuristic algorithms to enhance planning efficiency [7]. Various tools and frameworks, such as PANDA [6], PDDL [5], and Python-based planners [2], have been developed to facilitate the implementation of HTN planning. These tools are supported by extensive research published in top conferences like IJCAI and AAAI, where innovative approaches and improvements are regularly introduced.

Despite its advancements, HTN planning faces several challenges. One of the primary issues is the scalability of planners when dealing with large and complex task networks [3]. The computational overhead associated with decomposition and refinement processes can become substantial, necessitating more efficient heuristic methods [1]. This project will explore how to integrate HTN planning into real-world applications and better apply it to reality, such as robot task planning, artificial intelligence decision-making in games, and autonomous driving path planning.

2 Goals of the project

I currently plan to apply hierarchical task networks to path planning in autonomous driving. Firstly, I will delve into the existing Python based HTN planners, such as those on PANDA and PDDL, to understand their underlying logic and heuristic methods. At the same time, I will think about how to integrate these planners into the auto drive system to ensure efficient communication between the planning module and the vehicle recognition algorithm. In this project, I will first identify and analyze common autonomous driving scenarios using deep learning, such as urban driving, highway navigation, and obstacle avoidance, and then combine HTN planning methods to solve specific path planning problems in these scenarios. But due to my limited personal abilities, I may not be able to do path planning in all scenarios. This project will consider utilizing heuristic features, including:

- A Generic Method to Guide HTN Progression Search with Classical Heuristics
- Delete-and Ordering-Relaxation Heuristics for HTN Planning
- An Admissible HTN Planning Heuristic
- Loop Detection in the PANDA Panning System

Encoding and implementing the HTN planner using these heuristic functions will be a key focus, and then designing experiments to evaluate its effectiveness and performance. Performance will be evaluated using metrics such as path optimization, computational efficiency, security, and adaptability to dynamic environmental changes. In addition, a functional prototype will be developed to demonstrate the ability of the HTN planner in autonomous driving, and thorough testing and validation will be conducted to ensure reliability and robustness.

3 Risk assessment

Possible risks associated with the project are described below.

- Route Planning and Navigation: incorrect or suboptimal route selection leading to delays or unsafe paths. Evaluate traffic conditions, road types, weather forecasts, and historical accident data to choose the safest and most efficient route.

- Scenario Coverage: Due to limited personal abilities and resources, the project may not cover all possible autonomous driving scenarios. This could result in the planner being less effective or even failing in untested or rare scenarios.
- Integration Challenges: Integrating HTN planners with the existing autonomous driving system may pose significant challenges. Ensuring smooth communication between the planning module and the vehicle's recognition algorithms requires a deep understanding of both systems and could lead to compatibility issues.

4 Required resources

The hardware and software resources required for this project are listed below:

- A workstation or server with a CPU (i5 or i7)
- Stable internet connection
- Python ≥ 3.6
- Pytorch, Sklearn, Pandas, Numpy

5 Platform

Tkinter: A built-in GUI library in Python, suitable for creating simple user interfaces.

- System control and monitoring:

  Create a control panel to start and stop HTN Planner. Real time monitoring of the status and path planning results of the auto drive system.

- Parameter settings:

  Provide an interface for users to adjust the parameters and algorithm options of HTN planning. Real time updating and displaying the impact of parameters on path planning effectiveness.

- Result display and analysis:

  Visualize the path planning results on the map. Display computational performance indicators, such as path optimization degree, computation time, etc

6 Outline timetable

An outline timetable for the project is presented below. All dates refer to the year 2024.

From	To	Activity
June 1	June 25	1. Gather and set up hardware and software resources; initial literature review on HTN planning. 2. Deep dive into existing Python-based HTN planners to understand underlying logic and heuristic methods.
June 25	June 30	Integration planning: devise strategies for integrating HTN planners with autonomous driving systems; initial scenario analysis.
July 1	July 10	Continue coding HTN planner; implement and test heuristic functions; refine scenario analysis.
July 11	June 20	1. Integrate HTN planner with vehicle recognition algorithms; ensure efficient communication between modules. 2. Conduct initial tests of HTN planner by coding.
July 21	July 28	Analyze test results; optimize HTN planner based on feedback; focus on path optimization and computational efficiency.
July 29	August 3	Implement additional heuristic functions; further testing and validation in diverse scenarios.
August 4	August 6	Design experiments to evaluate effectiveness and performance; establish metrics for path optimization, security, and adaptability.
August 7	August 10	Iterate on planner design; refine code and heuristics based on experimental results; prepare for final testing.
August 11	August 12	Conduct thorough testing and validation of the prototype in various scenarios; ensure re1liability and robustness.
August 13	August 15	Finalize documentation and project report; submit project deliverables; project review and wrap-up.

Table 1: Outline timetable

References

[1] Ebaa Alnazer, Ilche Georgievski, and Marco Aiello. "On bringing HTN domains closer to reality-the case of satellite and rover domains". In: *International Conference on Automated Planning Systems (ICAPS) Workshop on Scheduling and Planning Applications (SPARK)*. 2022, p. 13.

[2] Alexandre Angleraud et al. "Virtual teaching for assembly tasks planning". In: *2020 IEEE International Conference on Human-Machine Systems (ICHMS)*. IEEE. 2020, pp. 1–6.

[3] Yash Bansod et al. "Integrating Planning and Acting With a Re-Entrant HTN Planner". In: *ICAPS Workshop on Hierarchical Planning (HPlan)*. 2021, pp. 22–54.

[4] Kutluhan Erol, Dana Nau, and James Hendler. "Toward a general framework for hierarchical task-network planning". In: *methods* 1.t1 (1993), N2.

[5] Daniel Höller et al. "HDDL: An extension to PDDL for expressing hierarchical planning problems". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 06. 2020, pp. 9883–9891.

[6] Daniel Höller et al. "The PANDA framework for hierarchical planning". In: *KI-Künstliche Intelligenz* 35.3 (2021), pp. 391–396.

[7] Tommaso Mannucci, Robert Zimmermann, and Christian Frese. "Extending Reward-based Hierarchical Task Network Planning to Partially Observable Environments". In: *2024 10th International Conference on Automation, Robotics and Applications (ICARA)*. IEEE. 2024, pp. 178–184.

[8] Negin Nejati, Pat Langley, and Tolga Konik. "Learning hierarchical task networks by observation". In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 665–672.

[9] Jongwon Seo et al. "Task planner design for an automated excavation system". In: *Automation in Construction* 20.7 (2011), pp. 954–966.