ABSTRACT

Title of Thesis:	Learning Metareasoning Policies for Motion Planning
	Siddharth Gopal Master of Science
Thesis Directed by:	Professor Jeffrey Herrmann Institute for Systems Research

Metareasoning is the process of reasoning about reasoning. This thesis applies metareasoning to motion planning and evaluates three different metareasoning policies. Two policies are rule-based policies and are human specified. The third policy is a smart metareasoning policy that learns from the robot's past experiences, particularly the front camera images. The data is obtained by running the robot without a metareasoner in modular test scenarios which can be combined to form multiple real-world situations. The policy is stored in the form of the weights of a neural network. The neural network-based model used for this research is a multi-input classifier that chooses an optimal planner combination from amongst eight different planner combinations. The metareasoners are tested on a Unity simulator with a Clearpath Warthog ground robot. This thesis tests the performance of the robot under eight different test scenarios for eight different planner combinations and shows an improvement in the robot's success rate when using a metareasoner. Lastly, this thesis also provides a comparative study between a rule-based metareasoner and a smart metareasoner by introducing two new test scenarios which are

not part of the robot's past experiences.

Learning Metareasoning Policies for Motion Planning

by

Siddharth Gopal

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2023

Advisory Committee:
Professor Jeffrey Herrmann, Chair/Advisor
Assistant Professor Michael Otte
Associate Professor Mumu Xu

Acknowledgments

I would like to express my deepest gratitude to Professor Jeffrey Herrmann. This research would have been impossible without his valuable advice, practical suggestions, unwavering support and belief in my ideas.

I would also like to thank Professor Michael Otte and Professor Mumu Xu for their insights, feedback and guidance that has shaped the direction of this research.

I am grateful for the helpful advice that I have received from Kevin Carey and Sidney Molnar throughout this challenging process.

Lastly, I would like to thank my family and friends. I would be remiss in not mentioning the encouragement and stimulating conversations that Smriti Gangwani and Vishal Gattani have provided me that has enabled this research.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motion Planning and Robotics

Motion planning is the computational process of determining a sequence of actions that will allow a robot to move to a desired location by avoiding obstacles and adhering to constraints such as energy and safety. Motion planning is a critical part of robotics, without motion planning robots would be limited in their ability to perform complex tasks and would be restricted to repetitive tasks with programmed motions.

Over the past three decades motion planning algorithms have been a crucial research area in robotics [1]. One of the earliest path planning methods called the visibility graph method, uses the the A* search algorithm [3]. This method is attributed to NJ Nilsson and dates back to 1969. Since then there have been significant advances in the field of motion planning with the configuration space being formalized [2], and with the introduction of algorithms like the Potential Field Method [5], Probabilistic Roadmap [4] and Rapidly-Exploring Random Trees [6].

1.2 Metareasoning

Metareasoning refers to the process of reasoning about reasoning. It is a metacognitive processes that monitors and controls ongoing thinking of a system and thus has the ability to direct the course of its computations according to the situation the system finds itself in. Metareasoning can thus take many forms, from monitoring one's own thinking and evaluating the quality of one's reasoning, to selecting appropriate strategies for a given task, to adjusting one's thinking based on feedback or new information.

Similar to the A* algorithm that decides which node to expand by evaluating each frontier node for the cost of a complete solution path constrained to pass through that node and expanding the frontier node that appears cheapest; the metalevel selects one out of multiple possible computation steps by estimating their utilities [7].

While metareasoning can be used in various different ways to enhance the process of motion planning, this dissertation proposes the use of metareasoning to select from amongst different combinations of global and local planners to reduce computation load on the robot and improve its performance.

1.3 Motivation

Over the past decade multiple papers have been published that show ways in which metareasoning can be used to improve planning in robotics. For instance, a safety metareasoning system has been proposed by Svegliato et al. [8] that runs task processes and safety processes in parallel with a conflict resolver for arbitration. This paper avoids monolithic decision-making and satisfactorily demonstrates the effectiveness of using metareasoning in simulations of planetary rover exploration. Another paper by Ortiz-Haro et al. [9] showcases the use of a metareasoning strategy in task and motion planning to extract plans which are not geometrically feasible and reduce calls to the motion planner, while using the Logic Geometric Programming approach for task and motion planning. This metareasoning approach delivers a speed up in problem solving. Moreover, a metareasoning approach has also been used to tune the hyper parameters of anytime algorithms [10]. It can thus be observed that research in metareasoning is focused on reducing the computational complexity of a problem through adjusting existing algorithms, however, the use of metareasoning to reduce computational complexity through switching between algorithms is an area of research that has not actively been explored and is the focus of this thesis.

There are a wide variety of path planning algorithms from deterministic graph search algorithms like A* and D* to probabilistic sampling based methods like RRT to potential field methods. Each of these algorithms have their own strengths and weaknesses, for instance, unlike the probabilistic sampling-based methods, the graph based algorithms are deterministic and will find a solution if it exists, however, they often tend to be much more computationally expensive. It is thus evident that there is benefit to be gained, in using metareasoning for switching between motion planning algorithms based on the situation the robot finds itself in. Moreover, choosing an appropriate algorithm for the increasingly dynamic environments that robots find themselves in today can be challenging and a metareasoning approach that switches between motion planning algorithms might be able to help alleviate this problem.

1.4 Contributions

The contributions made by this thesis are divided into the following three subsections:

1.4.1 Study of global and local planner combinations

This thesis provides an in depth study of the performance of various planner combinations by simulating tests on a Clearpath warthog ground robot. Complex environments are broken down into smaller environments to create conditions under which a single planner combination might prevail. These environments are referred to as test scenarios for the remainder of this thesis. Information gained from running the robot on the test scenarios, informs the data-driven approach for metareasoning. This information may also be used to predict the robots performance under various real-world scenarios that can be constructed by combining the test scenarios. It has been observed that certain planner combinations use less resources without compromising on the performance. For instance, the planner combination of SLGP and MPPI works very well under test scenarios 1,2,3 and 4. Moreover, the planner combination of SBPL and NLOPT use less resources and work well under test scenarios 5,6,7 and 8. Hence, this data can be used to make an informed decision about the choice of planners to use on the robot.

1.4.2 Testing rule-based metareasoning policies

This thesis uses a number of different metareasoning policies to determine the impact of metareasoning on the process of motion planning in robots. To begin with, an uninformed rule-based metareasoner is used to reboot the planning process in a robot. These rules are human specified and constructed through understanding the limitations of each planner. Another rule-based metareasoning policy switches between the SLGP MPPI planner combination and the SBPL NLOPT planner combination to best utilize their individual strengths. Both these policies show improved success rates on the test scenarios. Consequently, this work demonstrates the feasibility of using metareasoning for switching between motion planners to improve the robot's success rate while completing a mission.

1.4.3 Learning metareasoning policies

This thesis proposes a data-driven approach that can be used to learn metareasoning policies for motion planning. This approach helps generalize the conditions under which a planner combination would perform optimally. This thesis proposes a novel method that uses the robot's past experiences i.e. information obtained through running the test scenarios, to learn a metareasoning policy (smart metareasoner). This method involves supervised learning that trains a multi-input classifier to learn and store a metareasoning policy in the form of a neural networks weights. It has been observed that the process of switching between the planners can be computationally expensive, however, using a metareasoner increases the success rate of the robot. The rule-based metareasoner and the smart metareasoner are tested on test scenarios which the robot has not seen before (test scenario 9 and 10). It is observed that the smart metareasoner generalizes well to these test scenarios and performs better than a rule-based metareasoner.

1.5 Overview

Chapter 2 of this thesis details background information on metareasoning, motion planning and the different tools used to build the infrastructure that is used to test the motion planning and metareasoning algorithms. Chapter 3 describes the experimental methodology used for creating test scenarios, collecting data, running tests and constructing metareasoning policies. This chapter also discusses the training of a classifier that learns a metareasoning policy. Chapter 4 provides the results obtained for all the test scenarios and also discusses insights. Chapter 5 summarizes the results and draws relevant conclusions. It also suggests future work for this research.

Chapter 2: Background

2.1 Overview

2.2 Army Research Lab Ground Autonomy stack

The Army Research Lab (ARL) Ground Autonomy stack is a software repository that uses ROS [18] and focuses on cross-functional development and collaboration. ROS is a flexible framework for writing robotics software. In addition, ROS allows the use of Unity as a simulation environment. Unity, in turn, allows visual SLAM and real-time LiDAR simulations. Both of these are essential features that allow the autonomy stack to develop a grid in the form of a costmap which is used for motion planning. The architecture of the autonomy stack can be seen in Figure 2.1. This figure is borrowed from [11]. The ground robot simulated for this research is a Clearpath Warthog UGV [19]. The two sensors that create the grid for motion planning are an Ouster OS1 LiDAR and an Intel Realsense camera.

2.3 Motion Planning Algorithms

Looking for an appropriate motion planning algorithm that suits the restrictions of a use case can be challenging because of the overwhelming number of planning algorithms. A motion planning algorithm often minimizes an objective optimization function that helps recognize the

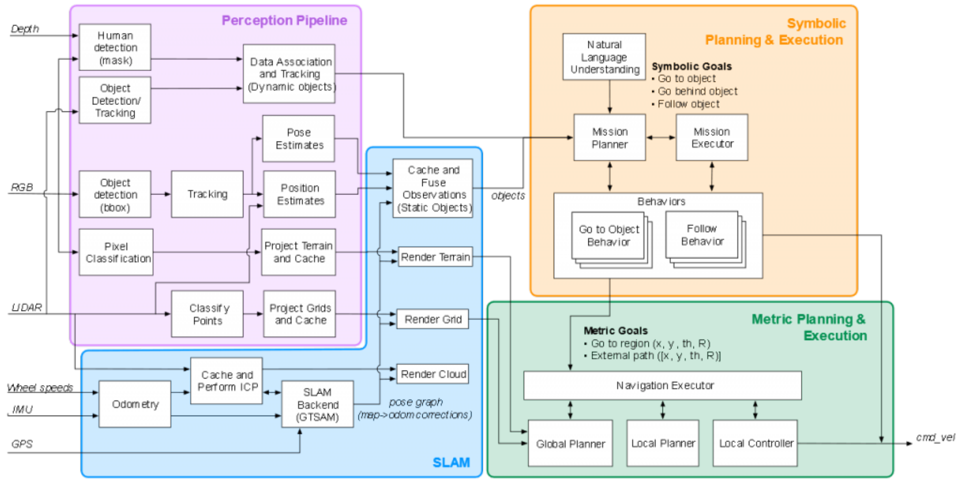


Figure 2.1: ARL Ground Autonomy Reference architecture. Source: [11].

path that takes the least amount of time or reduces resource utilization while complying with imposed restrictions. Previous comparative studies for path planning algorithms, like the work by Khanmirze et al. [20] focus on classic algorithms like Djikstra, A*, PRM and RRT. These algorithms are seldom used in real-world applications since they do not scale well, can not handle the complexities of real-world applications and are too slow for real-time use cases. It is thus essential to compare improved and current versions of these algorithms that find direct application in robots today.

A primary distinction between global planners and local planners is based on the size of the grid on which the planning occurs. The size of the grid is dependent on the amount of the world that the robot can perceive. The global planner generates a globally optimal path. This path may pass through obstacles which are not in the robots purview. On reaching closer and recognizing these obstacles, the local planner ensures that there exists a path that avoids these obstacles. It may choose to temporarily deviate from the global plan, alternatively it may request a new global plan. The global planner maintains its own planning search space, often in the form of a costmap through which it recognizes viable paths for the robot. The planning search space is a representation of the environment and is directly modeled from the environment. The planning search space may be created by tessellating the environment's surface into cells and arranging them in the form of a grid. The planning search space may also be created in the form of roadmaps. A roadmap is a graph whose nodes represent possible states of the robot, and the edges indicate the process required to go from one state to another. A state lattice graph is a form of a roadmap that is used in planners like EASL.

A total of 6 planning algorithms are included in this research; details about these planners and related planners are provided in Sections 2.3.1, 2.3.2, 2.3.3, 2.3.4, 2.3.5 and 2.3.6.

2.3.1 Search Based Planning Library (SBPL)

The search-based planning library is a library for planning with heuristic search. Maxim Likhachev developed this library in collaboration with Willow Garage [12]. Search-based planners require the construction of a graph, a cost function, and a heuristic function before the graph search algorithm can be applied. The graph is constructed using a SLAM algorithm and perception algorithms that are part of the ARL autonomy stack. The cost function is represented in the form of a cost map, and the heuristic function is dependent on the choice of the graph search algorithm being used. Of the multiple graph search algorithms present in SBPL, Anytime Repairing A*, commonly referred to as ARA*, is used in this research. ARA* uses the A* search algorithm with inflated heuristics. This algorithm quickly provides a feasible solution that is often sub-optimal and then continually works on improving it. The ARA* uses the A* algorithm in succession, decreasing the inflation factor with each run. Furthermore, it reuses the search effort from the previous A* run to make it more effective. This algorithm maintains its speed by not re-expanding locally inconsistent states. Locally inconsistent states are states whose g-value (cost to reach that state from the start node) has decreased before the next time the state is expanded. An experimental study that uses the ARA* algorithm for outdoor robot navigation and further details for the ARA* algorithm can be found here [17].

2.3.2 Efficiently Adaptive State Lattice (EASL)

When using a purely grid-based algorithm, the planned path involves sharp turns that do not necessarily incorporate the kinodynamic constraints of the robot. This problem is diminished by using a 3d grid instead of a 2d grid that incorporates yaw angles along with the x and y directions. However, such solutions only partially resolve the issue. That state lattice was created to resolve this issue. The idea of state lattices was introduced by Pivtoraiko et al., which created a search space that would efficiently encode only feasible motion plans. Figure 2.3 borrowed from [21] effectively showcases the generation of a space lattice. The state lattice resolves this issue by pre-computing the robot's motions, also called motion primitives, and uses these to construct a graph. State lattices use offline computing to form recombinant search spaces and thus enable the use of heuristic search. State lattices are defined by the use of a state mapping equation and a control set. The state mapping equation governs the transformation from the continuous states in the real world to the discrete nodes in the graph, while the control set represents the transitions from one node to another. Generating a feasible control set is challenging. If a continuous real-world space is expressed using an overly expressive control set, it may consume too many resources.

In contrast, a less expressive control set may not be able to handle complex environments. When the search space can not resolve a motion planning problem, a different representation of the real world is necessary to generate an alternate trajectory. The Adaptive State Lattice algorithm resolves this problem. This algorithm locally relaxes the state mapping equation, thus relaxing the local control set and better representing desirable regions of the path continuum. This adaptation takes place in an online fashion and is thus computationally expensive and limits the use of this algorithm. The efficiently adaptive state lattice (EASL) algorithm overcomes this limitation by limiting the set of feasible motions, thus allowing the precomputation of approximations of all motions that the adaptive state lattice algorithm could express. Thus the need for online trajectory generation is eliminated, and pre-computed swaths are used for evaluating edge costs. The swath represents the volume occupied by the robot during the execution of an edge action. This algorithm uses weighted A* to search for a feasible sequence of actions from the start node to the goal node on the state lattice graph.

2.3.3 Generalized Lazy Search (GLS)

Any typical search algorithm can be broken down into two parts, the first part is the search effort, and the second part is the edge evaluation. The search effort finds the shortest path from the start node to the end node, while the edge evaluation tests the feasibility of traversing of each edge. Algorithms like A* evaluate their paths as soon as the search effort for the path is completed. The process of edge evaluation is generally the more expensive of these two processes and can thus cause a bottleneck in computation. For instance, articulated robots that use roadmaps have few vertices, but their edge evaluation requires performing collision and

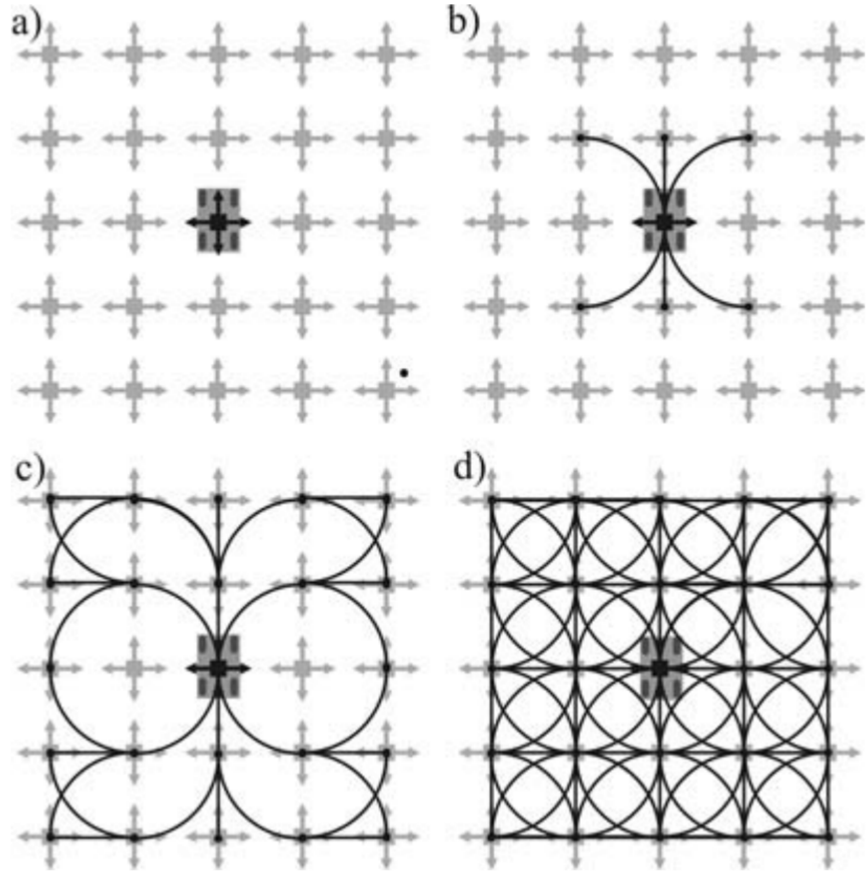


Figure 2.2: Constructing a space lattice for the Reeds-Shepp Car. Source: [21].

distance computations that often lead to large planning times, especially with complex robots and environments. This problem is tackled by the work conducted by Dellin et al. in [22]. In their proposed algorithm, the Lazy shortest path (LazySP), they evaluate the edges only for the shortest path. This increases the search effort since the algorithm must recompute the shortest path each time an edge is invalidated. The problem with this approach is that the search effort can become very expensive for large graphs or highly cluttered environments. The generalized lazy search algorithm solves this problem by algorithmically toggling between the search effort and edge evaluation. The GLS algorithm achieves this by using an Event module and a Selector module in their algorithm. The Event module defines the toggle between extending the lazy search tree and validating it, while the Selector module defines the strategy to select an edge along a sub-path to evaluate. This algorithm is described in greater detail in [16]. Figure 2.3 from [16] describes this framework.

2.3.4 Straight Line Global Planner (SLGP)

The straight-line global planner is a simple global planner that requires very little computation. However, this planner is incapable of recognizing optimal paths that require the robot to temporarily move away from the goal location. The straight-line global planner outputs a straight-line path from the start location to the goal location in the form of a global plan and ignores any obstacles that come in the global plan. This lack of planning from the global plan increases the computation effort on the local planner. However, such a planner can prove to be an ideal solution if the robot is not required to stray too far from the straight-line global plan in order for it to reach the goal location.

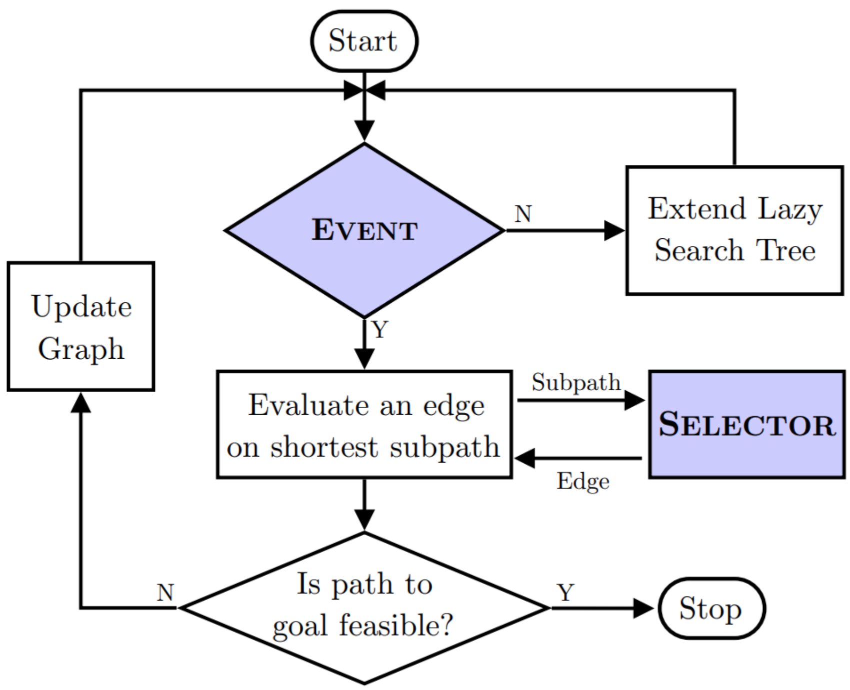


Figure 2.3: Framework for Generalized Lazy Search. Source: [16].

2.3.5 Model Predictive Path Integral (MPPI)

The model predictive path integral is a model predictive control algorithm that is very effective for control tasks in dynamic environments. Instead of optimizing a controller that would work well for the full-scale problem, this algorithm focuses on optimizing a simpler algorithm for a short planning horizon, thus making an appropriate decision based on the current conditions. It thus alternates between optimizing the control and applying its corresponding control in a manner that is very similar to online learning in which the learner makes a decision, suffers a cost, and based on this cost and the loss function, makes the next decision to minimize the accumulated costs also known as regret. The work by Wagener et al. [14] leverages this similarity to design MPC algorithms based on pre-existing powerful online learning algorithms. A standard MPC problem setup involves a state denoted as x_t a control denoted as u_t, and a transition map $f(x_t, u_t)$ that would give us the next state x_{t+1}. The state transition map is approximated in the form of a stochastic dynamic model $\hat{f}$, on which a control sequence $\hat{u}$ is applied for H time steps in order to obtain a predicted state trajectory $\hat{x_t}$ defined as $\hat{x_t}(\hat{x_t}, x_{t+1} \hat{....} x_{t\hat{+}H})$. The control sequence $\hat{u}$ is obtained by sampling a distribution π_θ that is parametrized by θ. The main objective of the MPC algorithm is to find θ_t that solves

$$min_{\theta \in \Theta} \hat{J}(\pi_\theta; x_t)$$

where $\hat{J}$ is the MPC objective, that predicts the long term performance of the system. On obtaining appropriate values for θ_t the sampled controls $\hat{u}$ are applied to the real dynamical system $f(x_t, u_t)$ by replacing u_t with $\hat{u_t}$. The problem of finding the control u_t can also be referred to as an optimal control problem when the control is open-loop. In the case of model predictive path integral control, the optimal control takes the form of a path integral. MPPI benefits from computing a weighted average of the sequences according to the desirability giving a smooth and low-cost trajectory. MPPI also does not need to back-propagate information during the process of optimization. This algorithm is introduced in the work of Williams et al. [23]. This work uses a dynamics model in the form of a linear function with 25 features for forward sampling. Furthermore, the initial model is trained from data obtained from human-controlled driving and fit using Bayesian linear regression. Finally, the online model is updated using a recursive Bayes filter.

2.3.6 Nonlinear Optimization (NLOPT)

The Nonlinear optimization approach is used in a local planner. This planner uses numerical linearization for trajectory generation. The numerical linearization inverts forward models of the vehicle. The forward models of the vehicles include the propulsion system, the suspension system, the wheel terrain interaction model, and a dynamics model. This planner uses a three-level architecture that separates trajectory generation, motion prediction, and vehicle simulation. Similar to MPPI, the trajectory generator generates a set of controls. These controls are subject to a set of state constraints and a set of differential equations that describe the system dynamics. The input to the system is the state boundary pair, the control parameters, and the vehicle model. The numerical method minimizes the constraint error by adjusting the control parameters. Next, motion is predicted using the vehicle model and numerical integration. This prediction forms feedback to the trajectory generator that modifies the control parameters until the terminal state of the forward simulated trajectory matches the target terminal state. Research by Howard et al. [13] demonstrates the use of this algorithm and shows the competency of this algorithm in challenging environments like rough terrain.

2.4 Metareasoning

Metareasoning is reasoning about reasoning. In order to build robots that compare with human-like performance, numerous research is conducted in the field of metalearning and metareasoning. While metalearning focuses on efficiently using computational resources and data, metareasoning addresses the problem of efficiently deploying computational resources. Based on these ideas, a rational agent should be aware of the value of computation and pursue computation with the highest value of computation.

Numerous works have shown the benefits of using metareasoning to improve autonomous systems. The work by Svegliato et al. [8] uses metareasoning for safe decision-making. It does by running task processes and safety processes in parallel. For instance, a planetary rover may run a crevice safety process, a dust storm safety process, and a rough terrain safety process in conjunction with a conflict resolver for arbitration. The safety processes recommend parameters for the robot while the conflict resolver selects the optimal parameters to be used for a given process. Unlike the work by Svegliato et al. [8] that focuses on altering parameters, this research focuses on analyzing the best planner combination for every situation the robot can face and using an optimal planner combination at all times. The work by Sung et al. [24] discusses using metareasoning for motion planning and focuses on anytime motion planners. Sung et al. use a metareasoner to determine when a motion planner should quit planning and start executing the plan. This metareasoner makes decisions based on the observed profile of the quality of the current best solution over time. If the profile is smooth, the metareasoner can extrapolate the future solution quality based on a model regressed from the history of solution qualities. The SATzilla program, created to solve propositional satisfiability problems, uses the idea of algorithm selection on cost-sensitive classification models. The solvers contributing to SATzilla are not necessarily the best-performing but use novel solution strategies to solve instances that would remain unsolved without them [25]. Like the approach taken by the SATzilla program, this research uses algorithm selection and uses it to select optimal motion planning algorithms.This research differs from [24] in its approach but focuses on the same topic of using metareasoning for motion planning.

Chapter 3: Methodology

3.1 Motion Planning

The motion planner used for this thesis is part of the ARL autonomy stack [11]. The task of motion planning has a hierarchical structure and is divided into a global planner and a local planner. The global planner computes an obstacle-free path from the start location to the goal location in the map frame, a coordinate frame that is fixed in reference to the world coordinate frame. The local path planner is used for collision avoidance. It computes short-horizon trajectories and works at a high update rate. Furthermore, it works on a separate inner loop and is required to meet a time threshold to keep up with the update rate. The robot performs evasive maneuvers; the planner determines when they are needed and plans the maneuvers. If the local planner fails to find a path, the system is stopped to ensure safety. The local planner computes the local plan by intersecting the global plan with the local map. After avoiding imminent collision, the robot may continue pursuing the original global plan; alternatively, it may recalculate a global plan better suited to the robot's current situation. The motion planning architecture consists of two more important modules, the navigation manager and the local controller. The navigation manager monitors task completion and triggers global re-planning when the current local plan is infeasible. Additionally, the local controller controls the robot's movement between updates from the local planner. Figure 3.1 shows the architecture of the entire

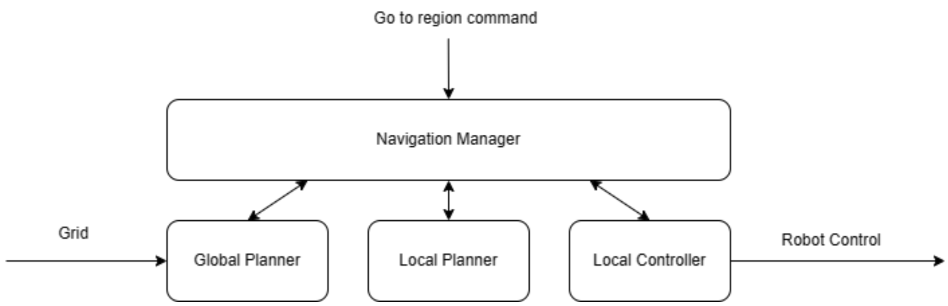


Figure 3.1: Architecture of the planning and execution of a *go_to* region command in the ARL autonomy stack

motion planning system.

This research uses eight combinations of global and local planners. All the possible combinations are listed in Table 3.1

Table 3.1: Matrix of Global and Local Planners

Local Planner/Global Planner	SBPL	EASL	GLS	SLGP
MPPI	SBPL-MPPI	EASL-MPPI	GLS-MPPI	SLGP-MPPI
NLOPT	SBPL-NLOPT	EASL-NLOPT	GLS-NLOPT	SLGP-NLOPT

3.2 Global Planners

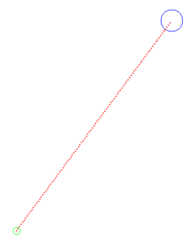
The purpose of the global planner is to provide a high-level plan that allows the robot to move from the start location to the end location while avoiding obstacles and meeting constraints like the robot's shape and kinematic limitations. The output from the global planner is the global plan which may change throughout the course of executing a mission. The global planner undergoes the process of planning or re-planning on receiving a trigger from the navigation manager and between fixed time intervals based on the speed with which the global planner can re-plan. After creating an initial plan, the global planner creates a new plan each time the local planner is not able to plan a path around an obstacle. This process might occur very often for global planners like the SLGP; for environments with multiple obstacles, a straight line from the robot's current location to the end goal can have obstacles the local planner can not circumvent.

On the other hand, if the robot has complete information about the environment and if there exists a feasible path from the start location to the end location, deterministic planners like SBPL are capable of planning an obstacle-free path from the start location to the end location without the local planner getting stuck and thus do not need to undergo the process of re-planning. However, the robot seldom has complete information about the environment and is required to construct a map of the environment while still running the mission. The global planner obtains information from the global costmap in the form of a grid on which the planning takes place.

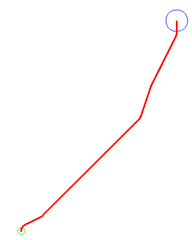
Figure 3.2 shows the paths planned by the different planners when faced with no obstacles. The green circle in the figures represents the start location, while the blue circle represents the end location. The red dots are a set of poses that the robot must follow in order to reach the end goal.

As is expected from the planners, the SLGP global plan is a straight line from the start location to the end location, the SBPL global plan contains sharp turns, the EASL global plan contains smooth curves since it uses a state lattice and the global plan from the GLS planner is a straight line path that would get updated on recognizing an obstacle.

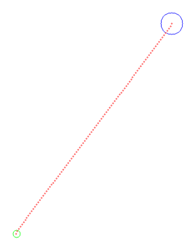
The robot's trajectory during a mission's completion might differ from the global plan. The final trajectory followed by the robot would be a piece-wise summation of the local plans generated using the global plan. Thus, despite the sharp turns in the SBPL plan, the final trajectory that the robot will follow will be smooth and will not involve any spin turns.



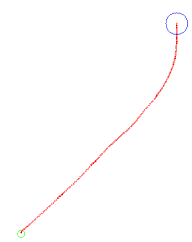
(a) SLGP global plan



(b) SBPL global plan



(c) GLS global plan



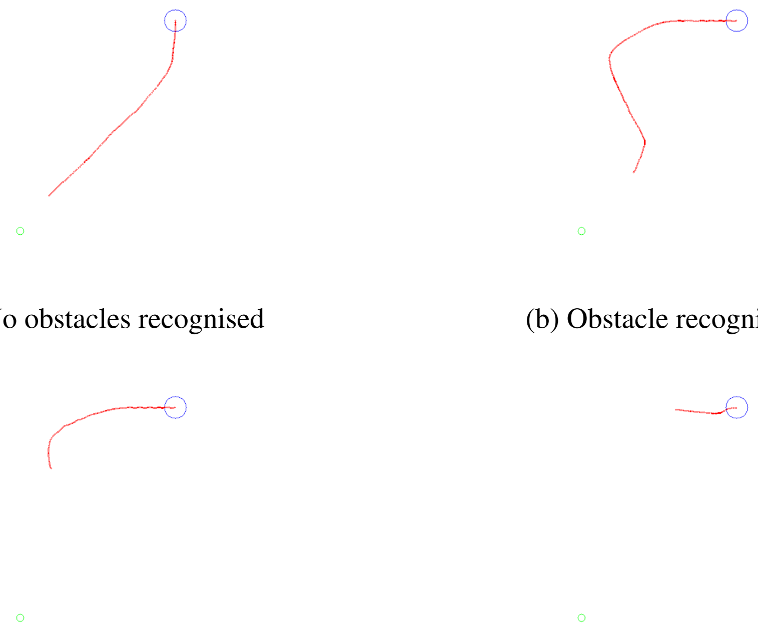
(d) EASL global plan

Figure 3.2: Global plans created by different planners when no obstacles are present in the environment. The green circle represents the start point, the blue circle represents the end point and the red dots are a set of poses.

Furthermore, Figure 3.3 shows the manner in which the global planner is updated for the EASL global planner when running test scenario 8. Details about the various test scenarios are discussed in Section 3.7. For example, in Figure 3.3a, the robot has not recognized any obstacles and plans a path to the end goal similar to Figure 3.2d. However, on recognizing an obstacle, it plans a new path as shown in Figure 3.3b.

3.3 Local Planner

The purpose of the local planner is to generate a trajectory that avoids obstacles in the robot's immediate surroundings. Figure 3.4 shows a sample of the local plan planned by the robot during the execution of a mission. The local planner runs at a constant frequency of 20hz



(a) No obstacles recognised

(b) Obstacle recognized

(c) Plan readjusted when close to obstacle

(d) Re-planning in fixed time intervals

Figure 3.3: Sequence of global planner updates while running test scenario 8.

for all the test scenarios and has a time horizon of 5 seconds. These values are tuned based on the robot's max speeds and ensure the local planner's smooth working.

The local planner takes the input from the global planner through the navigation manager in the form of a sequence of waypoints. The local planner also obtains information about the robot's immediate surroundings through the local costmap. The local planner may send commands to alter the global plan based on input from the surrounding. Furthermore, if the local planner fails to provide a feasible plan, the navigation manager may stop the run to ensure the robot's safety.

3.4 Global Costmap

The global costmap is a representation of the environment and consists of cells describing the cost or the difficulty of navigating through that cell. The costmap may also be called an

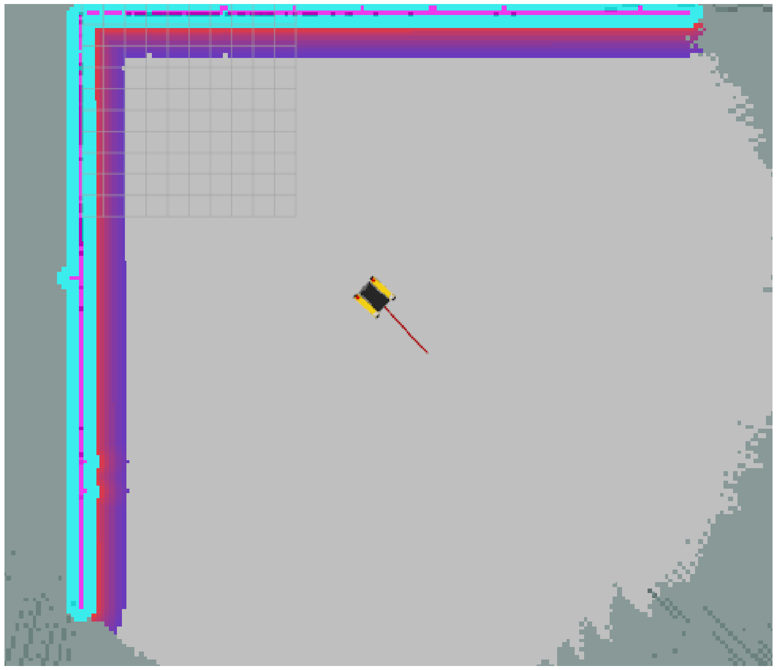


Figure 3.4: Local plan during the execution of a mission. The region in light blue are the obstacles, the area surrounding it is the inflated region created to avoid collision.

occupancy grid. Each cell of the costmap may be deemed free, occupied, or unknown. Figure 3.5 is a global costmap in which the free space is depicted in black, the occupied space in grey, and the unknown space in white. The costmap is an input to the planners that calculate a low-cost path from the start location to the end location. Similar to the global and local planners, the global costmap is updated at a lower frequency than the local costmap. The global costmap obtains information about the robot's surrounding based on information from the local costmap, which has a higher resolution and is more up-to-date. The global costmap integrates the local costmap information using a filtering algorithm.

Since the data used to construct the costmap is obtained from sensors which can be noisy and unreliable, the costmap is thus probabilistic in nature and assigns a probability distribution over each cell being occupied/closed or being free/clear. The clear probability threshold for the global costmap used during this research is 40. Thus, if the probability that a cell is clear is below 40%, the cell is assumed to be occupied by an obstacle.

Lastly, each occupied cell must be inflated based on the robot's inscribed radius. This is done to ensure that no part of the robot comes in contact with the obstacle when the plan is translated from the abstract costmap back to the real world. The inflation radius for the global costmap is set at 2 cells.

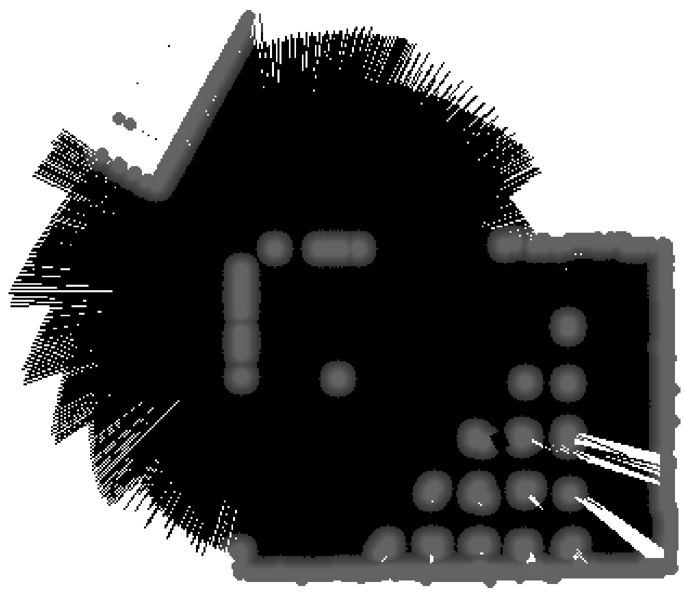


Figure 3.5: Global costmap from a run of test scenario 6 of test set 1. Free space is depicted in black, occupied space in grey and unknown space in white.

3.5 Local Costmap

Like the global costmap, the local costmap consists of cells that form an occupancy grid. Figure 3.6 shows a local costmap from test scenario 6 of test set 1. It runs at a higher frequency than the global costmap and holds the most up-to-date information about the surrounding cells. It receives data directly from the sensors and covers a much smaller area of the surroundings when compared to the global costmap. This way, the local planner can make faster decisions and find paths that avoid obstacles.

In the setup used for this research, the local planner updates the global planner using a moving average filter. This filter averages the values of the local costmap over time and is thus able to smooth out any noise in the data. This filtered data is integrated with the global costmap.

Data in the global costmap may not originate in the local costmap. For instance, Lidar data containing information about distant objects not in the robot's immediate vicinity is directly integrated with the global costmap and may never be seen by the local costmap.

The global costmap sacrifices the resolution of the data in order to store and access data about the complete environment. However, the local costmap is required to maintain high-resolution data in order to get a thorough understanding of the surrounding environment.

3.6 Navigation Manager

The navigation manager is at the heart of all the operations in the motion planning process. Details about the navigation manager can be found in Figure 3.7. The navigation manager connects the planners with each other. Furthermore, the navigation manager receives the *go_to* commands that provide the robot with a goal location. The navigation manager is also in charge of visualizing the planner's activities and the costmap; it does so by sending the necessary commands to an RViz panel. RViz is a graphical interface that allows the user to visualize the output of ROS topics. The RViz panel gives the user feedback about the planners and allows the

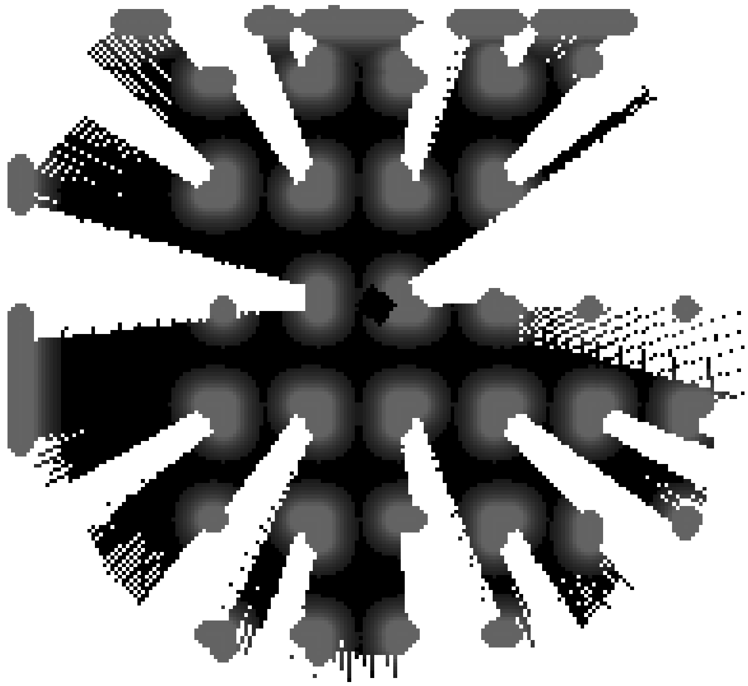


Figure 3.6: Local costmap from a run of test scenario 6 of test set 1. Free space is depicted in black, occupied space in grey and unknown space in white.

user to debug potential issues with the planner.

On receiving a *go_to* command, the navigation manager asks the global planner to compute or recompute a global plan. Alternatively, if the local planner cannot find a feasible local plan, the navigation manager may ask the global planner to recompute a plan. Lastly, the navigation manager may also cease the operation of both planners. It may do so when it repeatedly receives an infeasible plan from the global planner. The navigation manager waits for the global planner to re-plan 20 times without changing its location before it quits and shuts down both planners. The navigation manager may not receive a plan from the global planner on time. Under these conditions, the navigation manager shall output a timed-out error and wait for a new *go_to* command.

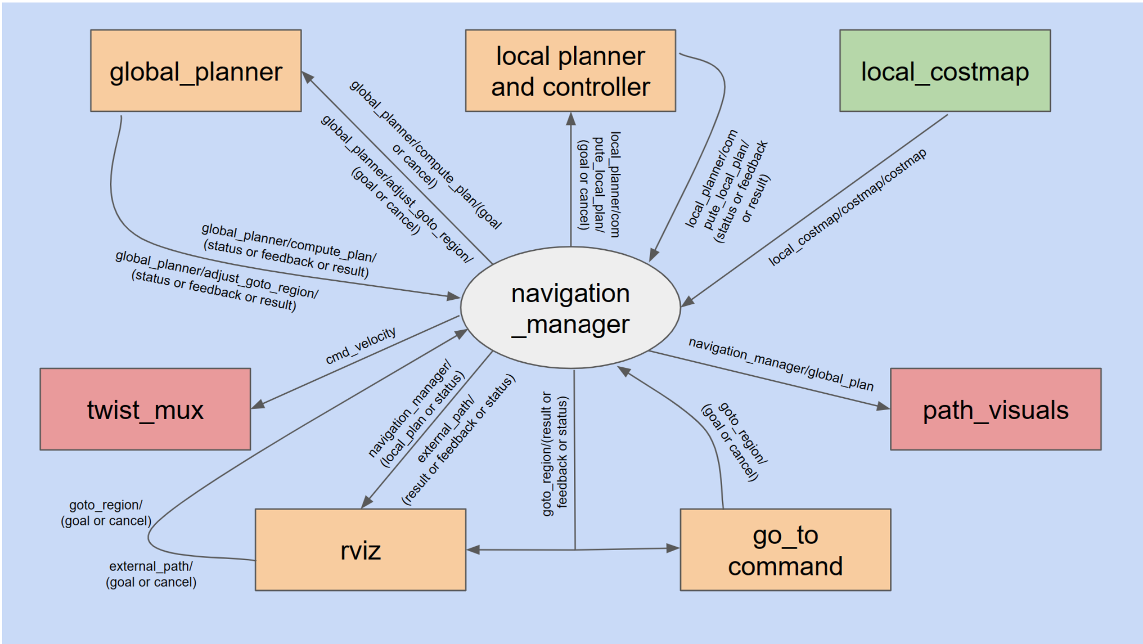
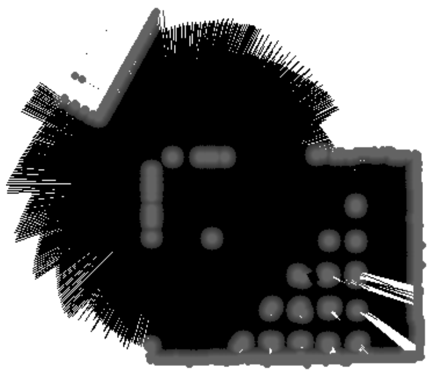


Figure 3.7: Interface diagram for the Navigation manager. The blocks in red are subscribers, green is publishers, and orange is both.

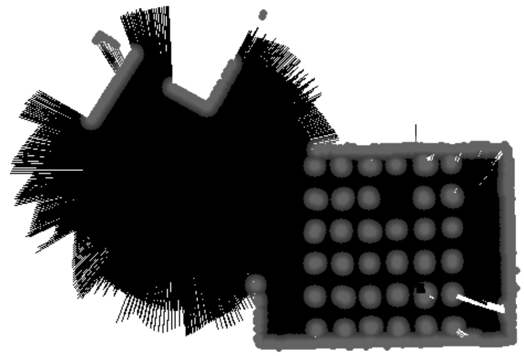
3.7 Generating Test Scenarios

All 8 motion planner combinations are tested under 2 test sets of 8 test scenarios each, thus reaching a total of 16 test scenarios. The obstacle placement for the 2 test sets is the same. The two test sets differ in the amount of time it takes for the global costmap to recognize the obstacles.

Both the test sets lead to different outcomes. The first test set can be compared to real-world scenarios in which an object moves into the robot's immediate vicinity from outside the robot's sensor's purview. For instance, these tests can be tantamount to scenarios in which a human has moved and stopped next to the robot vicinity from behind a blind spot, i.e., from behind a wall. This test set can also represent scenarios where the robot's sensors glitch. All scenarios in this set have cells in the local costmap that convert from free to occupied. Since the costmaps are probabilistic, introducing a new obstacle increases the probability that a cell is occupied. The



(a) Test set 1



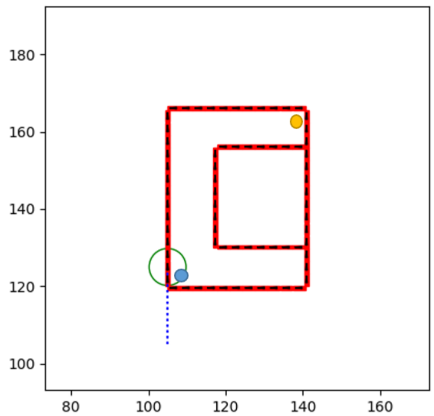
(b) Test set 1

Figure 3.8: Comparison of global costmaps for test scenario 6 in test set 1 and test set 2

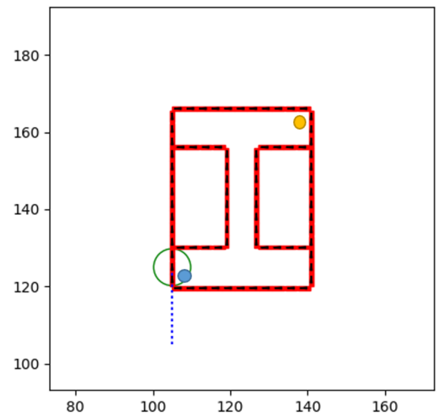
local costmap quickly adapts to these changes and updates its cells as occupied to accommodate the obstacle. However, the global costmap takes time to accommodate these changes from the local costmap. The rate at which the obstacles are integrated into the global costmap depends on the filtering algorithm.

For the second test set, the obstacles are spawned away from the robot, and the robot drives towards the obstacles before starting its mission. The obstacles are thus introduced to the robot one after another, similar to most real-world settings. For this test set, unknown cells are converted into occupied cells. Since there is no prior knowledge about these cells and the only information about these cells classify them as occupied, the global planner instantly updates these cells as occupied. Figure 3.8 shows a comparison of the global costmap for test scenario 6 in test set 1 and test set 2 to demonstrate the differences in the two test sets.

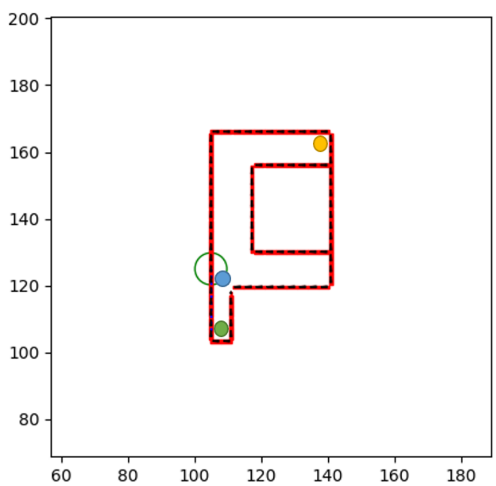
The two test cases lead to very different results. The first test case focuses more on the local planners and their ability to adapt to changes in the environment. The second test case has more information to work with and thus leads to fewer failures. In the second test case, the global planner's initial global plan can avoid most of the obstacles, reducing the local planner's computational effort and requiring the global plan to be corrected fewer times. Thus, there is significant difference in the robot's performance in the two test cases despite the obstacle arrangement being the same.



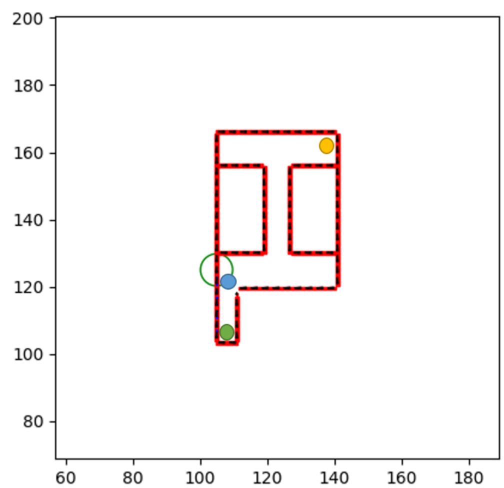
(a) Test scenario 2 in test set 1



(b) Test scenario 4 in test set 1



(c) Test scenario 2 in test set 1



(d) Test scenario 4 in test set 2

Figure 3.9: Comparison of test scenarios in test set 1 and test set 2. The blue dot is the start location, the orange dot is the end location. The green dot represents the robot spawn location for test set 2. The axes represent the position of the obstacles in Unity's coordinate frame.

Figure 3.9 shows a comparison between the test scenarios in the two test cases. The test scenarios for the second test case are the same as the first, apart from the extra road strip that the

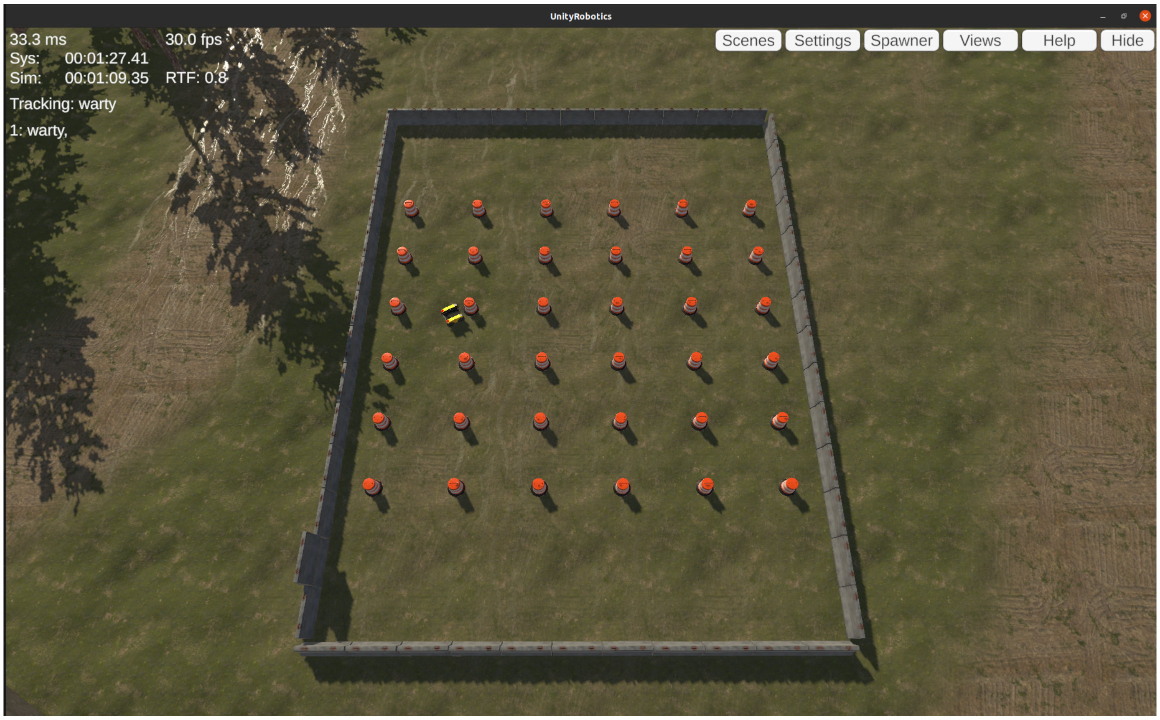


Figure 3.10: Test scenario 6 in the Unity environment.

robot must traverse before reaching the start location for the mission. The planning process only begins after the robot has reached the exact start location as the start location for test set 1. The start location is marked with a blue dot while the end location is marked with an orange dot. For test set 2 the robot traverses from the green dot to the blue dot before the test run starts. The robot faces the upwards direction during the start of each test scenario.

The various images in Figures 3.11, 3.12, 3.13, 3.14, 3.15 are obtained through Scenic. Scenic is a language-based scene-generation tool that allows one to create test scenarios in the form of written code. Once the JSON files created through Scenic are imported into the Unity environment, the robot is ready to start its mission. Figure 3.10 shows an example of the Unity environment after all the obstacles are spawned.

Each test scenario is built to replicate real-world situations the robot would face during its

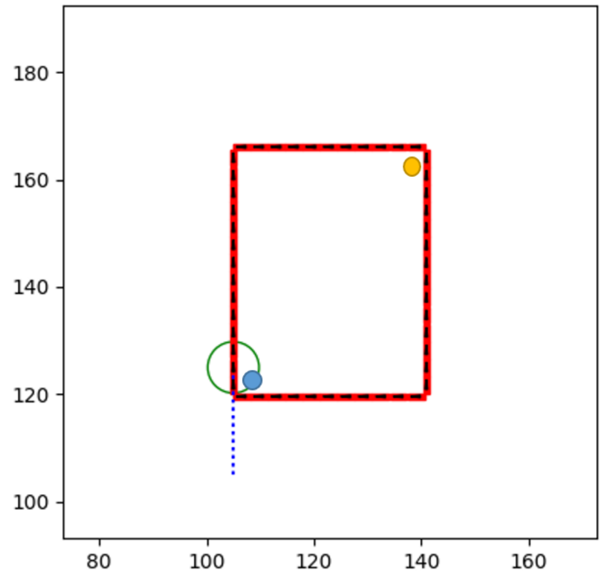


Figure 3.11: Test scenario 1. The blue dot is the start location, the orange dot is the end location. The axes represent the position of the obstacles in Unity's coordinate frame.

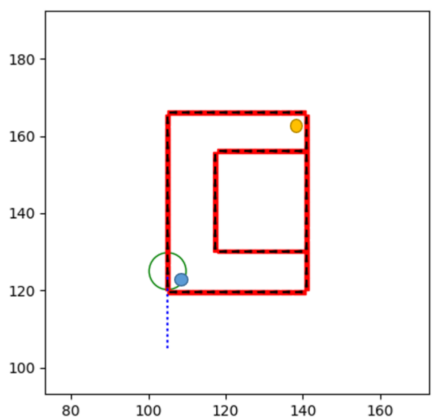
operation. A brief purpose behind each of the test scenarios is given in the following subsections. All the figures shown in the following subsections belong to test set 1. They are replicated for test set 2 as seen in figure 3.9.

3.7.1 Test scenario 1

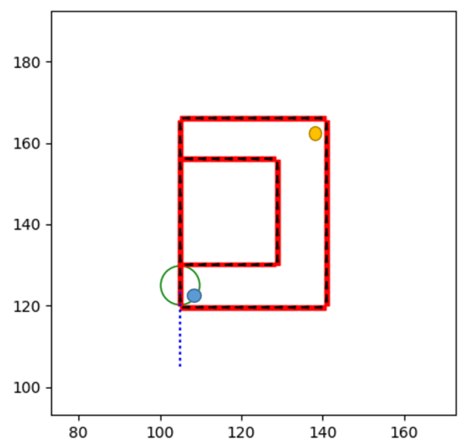
This test scenario consists of an open field, with the end location placed outside the purview of the local costmap. The approximate area covered by the local costmap can be seen in Figure 3.17. This size of the test scenario ensures that the global planner does not have complete knowledge of the environment and does not create a straight-line path to the end goal. This test scenario is meant to test the speed with which the robot can reach the end goal while also providing baseline values for calculating metrics like the CPU utility and the GPU utility. The test scenario 1 can be seen in Figure 3.11.

3.7.2 Test scenarios 2, 3 and 4

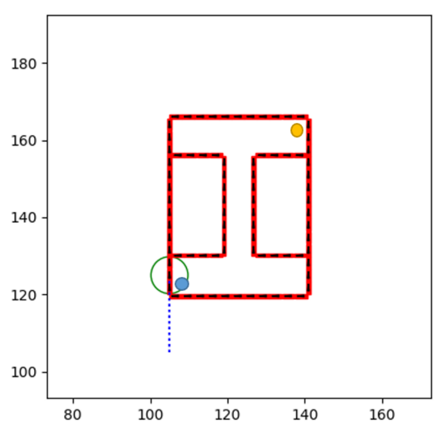
Test scenarios 2, 3 and 4 have a similar purpose; all three of these test scenarios try to replicate scenarios in which the robot is required to take turns in order for it to reach the end goal. Such a situation is similar to a city setting in which buildings are arranged in a grid-like pattern, and the robots must take 90-degree turns. Test scenarios 2, 3, and 4 require the robot to take 1,2 and 3 turns to reach the end goal.



(a) Test scenario 2



(b) Test scenario 3



(c) Test scenario 4

Figure 3.12: Test scenarios 2, 3 and 4. The blue dot is the start location, the orange dot is the end location. The axes represent the position of the obstacles in Unity's coordinate frame.

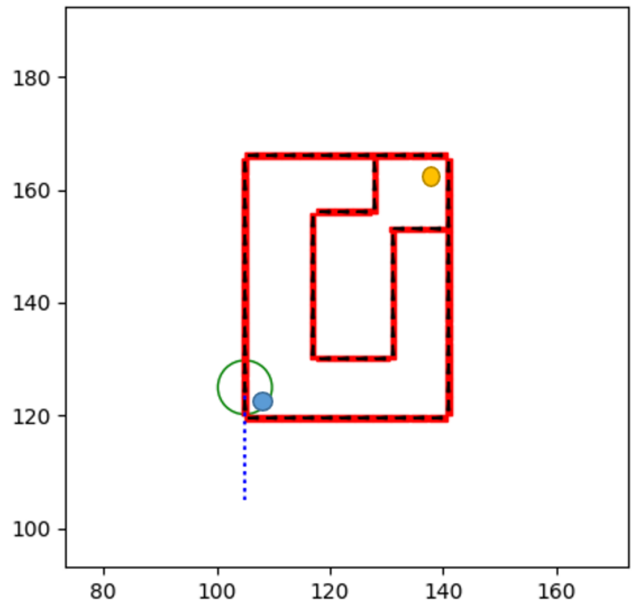


Figure 3.13: Test scenario 5. The blue dot is the start location, the orange dot is the end location. The axes represent the position of the obstacles in Unity's coordinate frame.

3.7.3 Test scenario 5

Test scenario 5 consists of obstacles placed in a manner such that it would be impossible for the robot to reach the end goal. This test scenario tests a planner combination's ability to explore the map and look for all potential solutions. The robot should take more time on this map as compared to the other maps. Moreover, the robot is capable of ending up in three states in this test scenario: failed, stuck, and timed out. Each planner combination may exit this test scenario in a different state.

3.7.4 Test scenario 6

Test scenario 6 replicates conditions similar to that of a forest with a regular distribution of trees, where obstacles are placed in a grid-like fashion with just enough space for the robot

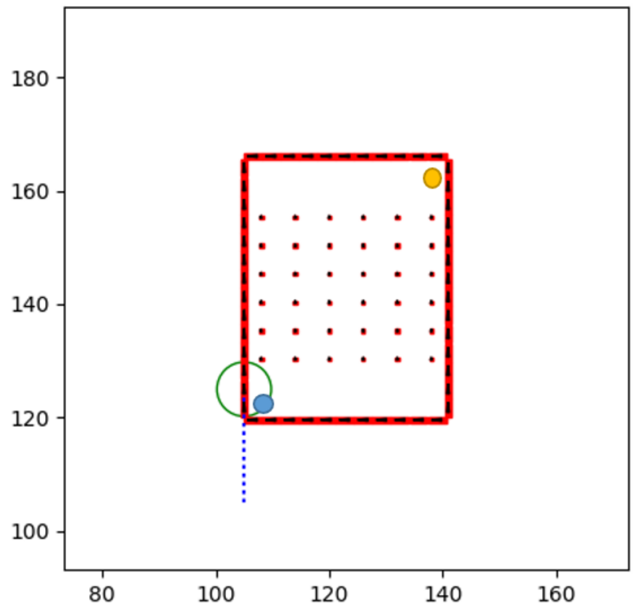
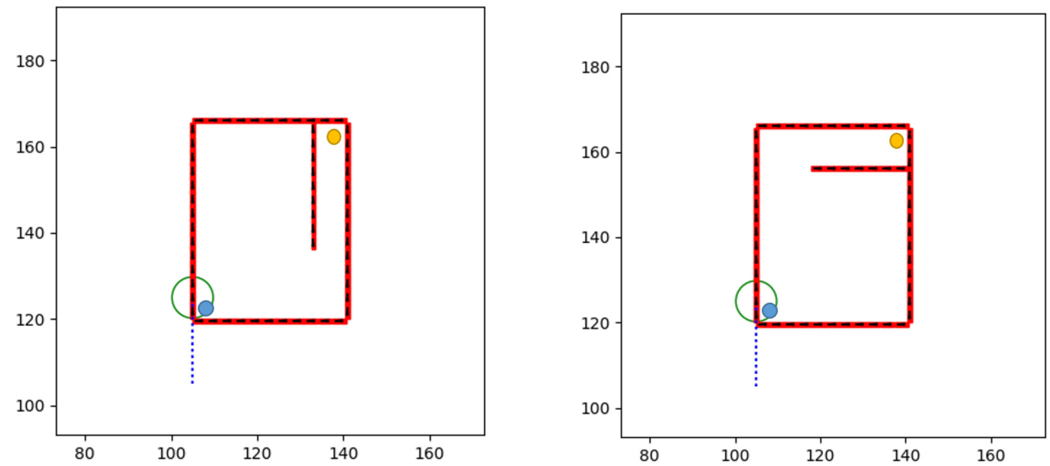


Figure 3.14: Test scenario 6. The blue dot is the start location, the orange dot is the end location.

to be able to move around them. The robot is required to re-plan each time a new obstacle is recognized in its path. The grid-like structure prevents the robot from being able to recognize all the obstacles at the same time.

3.7.5 Test scenarios 7 and 8

Test scenarios 7 and 8 are meant to replicate alleyways with small paths leading to the goal location. The paths are placed in such a fashion that the robot would recognize the need to use the alleyway only after it has reached the center of the map and has obtained sufficient information about the obstacle placement.



(a) Test scenario 7

(b) Test scenario 8

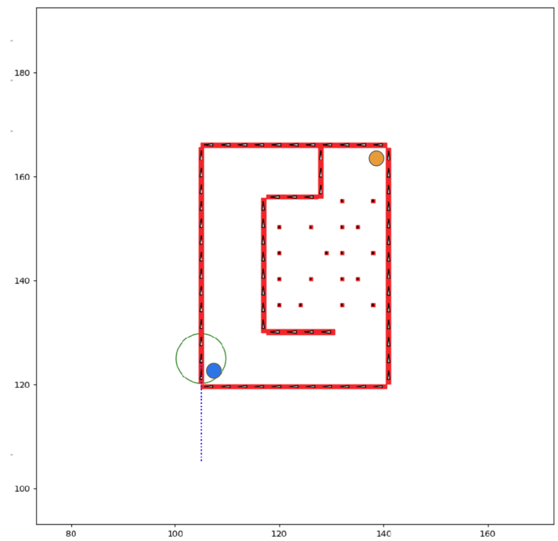
Figure 3.15: Test scenarios 7 and 8. The blue dot is the start location, the orange dot is the end location.

3.7.6 Test scenarios 9 and 10

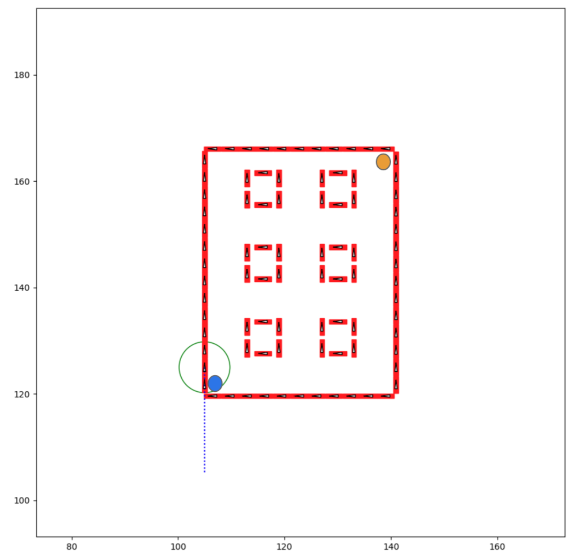
These two test scenarios are created to compare Policy 2 and Policy 3. These test scenarios were made once the metareasoning policies were developed and thus do not influence the process of creating the metareasoning policies. As a result, these test scenarios are directly referred to in Section 4.3 and shall not be referenced in the rest of Chapter 3.

3.8 Metareasoning

Metareasoning is the process of reasoning about reasoning. The application of metareasoning to motion planning requires the robot to evaluate the optimal planner combinations based on the condition the robot finds itself in. Learning a metareasoning policy can be broken down into four sub-problems. First, understanding the environment and the available data. Second, evaluating external information about the environment and internal information about the robot's working condition. Third, recognizing the most appropriate planner combination. Fourth, selecting



(a) Test scenario 9



(b) Test scenario 10

Figure 3.16: Test scenarios 9 and 10. The blue dot is the start location, the orange dot is the end location.

appropriate triggers for the robot to switch between the different planning algorithms. These four subproblems and the approach used to resolve them are described in greater detail in Sections 3.8.1, 3.8.2, 3.8.3, and 3.8.4.

3.8.1 Data Collection

It is of paramount importance for the metareasoner to use the least amount of memory and be computationally efficient in order for it to be able to contribute to the robot's performance instead of deteriorating it. Apart from writing efficient code, one of the critical factors in increasing the metareasoner's efficiency is choosing the appropriate data for the metareasoner to pay attention to and spend time evaluating. The metareasoner should use the least amount of data to develop a holistic understanding of the robot's situation. Thus, the data collection process focuses on assessing all available data, analyzing data, and synthesizing relevant information.

The first step in collecting data is to recognize all potential data sources. Focus is placed on using existing sources of data rather than generating new data. The following data sources were analysed for this research:

1. Global costmap
2. Local costmap
3. Global plan
4. Local plan
5. Forward camera images
6. Data from the IMU sensors
7. Goal location information
8. Navigation manager status

The global costmap is obtained in the form of a black-and-white image. The pixel intensities in these images represent the probability values of an obstacle being present in a given region. The global costmap provides complete information about the obstacle placement. However, two prominent issues were recognized, due to which the global costmap was not used for the metareasoner. First, the global planner does not provide the robot's location on the costmap. The image of the global costmap may be skewed depending on the obstacles surrounding the robot. Thus, it is necessary to fuse this data with other data sources to get a complete understanding of the robot's current situation. Second, the global costmap provides more data than is necessary for the robot to decide on the choice of planners. The robot requires local information to make decisions based on its current situation rather than the data provided by the global costmap, which is more suited for applications like planning future actions. For these reasons, this research has not focused on the use of a global costmap for metareasoning.

The use of a local costmap in place of a global costmap resolves the problem of surplus data, but it still requires additional information in the form of the robot's location. The local costmap is an ideal candidate for the metareasoner. The additional benefit of using a local costmap is the frequency with which it is published. This allows the metareasoner to respond quickly to the needs of the robot.

Lidar data is transmitted in the form of a point cloud and can thus be more computationally expensive to process. Furthermore, the costmaps are generated using the Lidar data. Using the Lidar data for the metareasoner would repeat computations on the Lidar data making the Lidar data a bad choice for the metareasoner.

The global plan and the local plan provide unique information about the robot's method of reasoning. The global, as well as the local plan are represented in the form of an array of poses. This data representation is lightweight and easy to process. Most of the global planners used in this research produce data in a recursive fashion. They use the current global plan to improve the next global plan. This makes this data source dynamic and unreliable. Furthermore, these data sources are incomplete and not sufficient for the robot to make a decision on the planner combinations.

The front camera images are a rich source of information. They are used in a number of different processes in the robot and are thus available in different forms, including a compressed version of the image and a semantically segmented version of the image. Furthermore, multiple image processing algorithms are readily available, which are optimized to run fast and without using a lot of memory. Moreover, the front camera images can provide a detailed description of

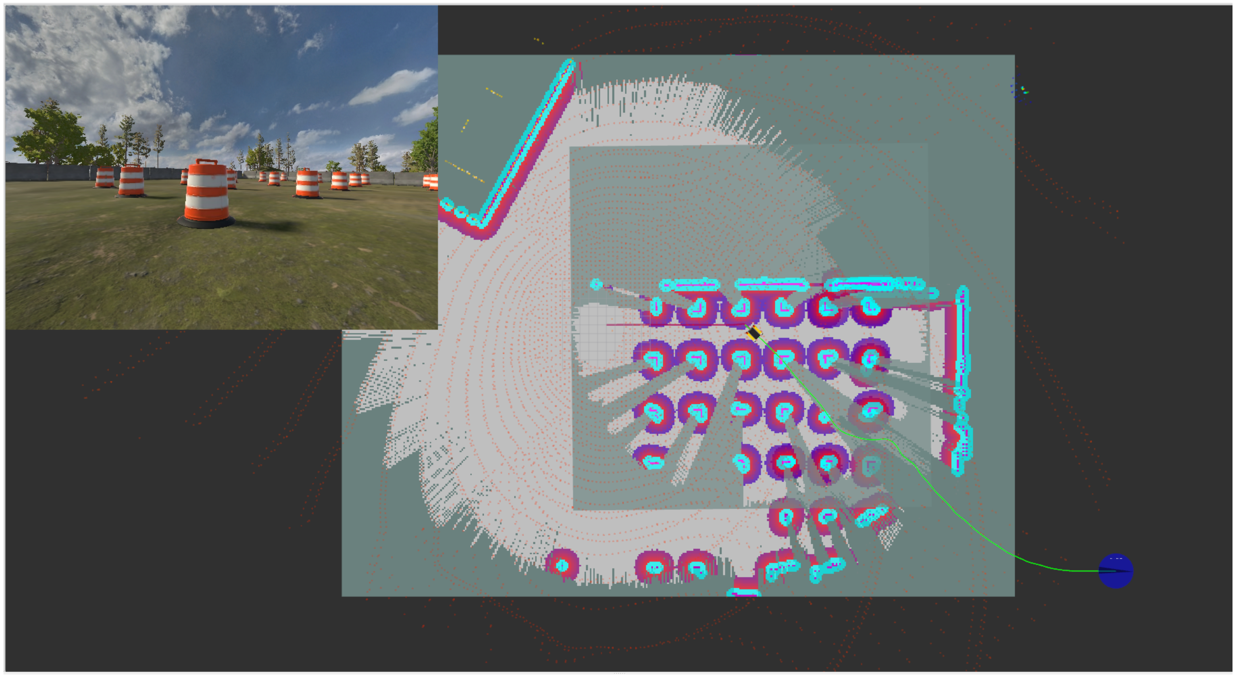


Figure 3.17: The image on the top left is the image from the robot's front camera. The dots surrounding the robot is Lidar information shown in the form of a point cloud. The bigger rectangle is the area covered by the global costmap. The small rectangle housed inside the bigger rectangle is the area covered by the local costmap. The portions in light blue are the obstacles, and the inflated region around the obstacle is shown in darker colors. The dark blue circle is the goal location. Lastly, the global plan is the green line from the robot to the goal location.

the obstacles. The only disadvantage of using images from the front camera is the loss of depth information that is provided through information sources like the local costmap.

Figure 3.17 shows all these different sources of information in a single image obtained through the RViz panel. In this figure, the robot is running test scenario 6.

The remaining data sources, like the IMU sensors, the goal location information, and the navigation manager status, are incomplete sources of information. They do not contain sufficient information for the metareasoner to make an informed decision.

Through this analysis, the front camera image and the goal location information have been chosen as the source of information to train the meta-reasoner. Two different approaches are used to manipulate the front camera image. Both these approaches are described in Section 3.8.7

Apart from the external sources of information, six parameters are selected to calculate the internal state of the robot. The average value of these six parameters is calculated over the span of an experimental run and stored in a tabular form. These results can be observed in A.1. The six parameters are as follows:

1. CPU Utilization
2. GPU Utilization
3. RAM Utilization
4. CPU Temperature
5. GPU Temperature

The system used to run the simulations has an Intel i9 11900H processor and runs an NVIDIA GeForce RTX 3050 Ti graphics card. While the utilization and temperature values may differ for a robot running a different CPU and GPU, the relative values for these parameters should be consistent. In order to maintain this consistency, all the experiments are conducted using the same system and under similar conditions.

3.8.2 Metrics

The internal input parameters for the metareasoner are also used as metrics to judge the metareasoners' performance. For example, an ideal metareasoner should be able to reduce the CPU and the GPU utilization, resulting in lower temperatures for both of them. Apart from these metrics, the success status of the test scenarios and the time of completion of the test scenarios are used as metrics to judge the performance of the individual planners and the meta-reasoning policies.

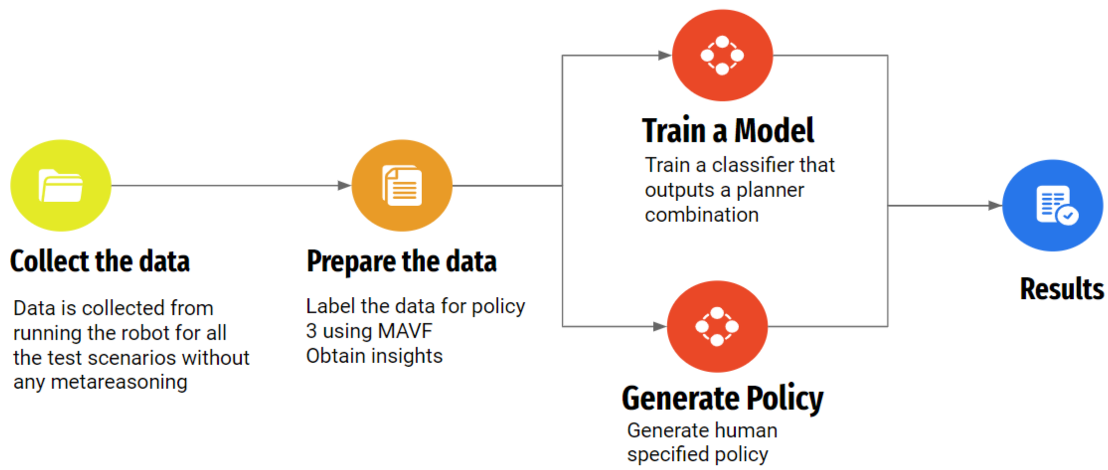


Figure 3.18: Process of choosing an optimal planner

3.8.3 Choosing an optimal planner combination

The data obtained from running the robot under the 16 different test scenarios are used to evaluate the optimal planner combination. This data influences the metareasoners' choice of planner combination. This data can also be used as a stand-alone piece of information to gain insights into the performance of different planner combinations. These insights are discussed in greater detail in Chapter 4.

The illustration in Figure 3.18 demonstrates the process of choosing a planner. The analysis process and the insights that lead to the policy are discussed in Sections 3.8.5, 3.8.6 and 3.8.7.

3.8.4 Trigger conditions

The trigger conditions are the conditions under which the metareasoner shall switch between the different planner combinations. Policy 1 and Policy 2 follow the same trigger conditions. The metalevel reads ROS topics to understand the working of the robot, particularly the ROS topic

COUNT of Success status	Global Planner	Local Planner							
	EASL		GLS		SBPL		SLGP		Grand Total
Test Num	MPPI	NLOPT	MPPI	NLOPT	MPPI	NLOPT	MPPI	NLOPT	
Test 1	5	4	5	5	5	5	5	5	39
Test 2	5						5		10
Test 3	5		1		3		5		14
Test 4	3		3		5		4		15
Test 6	5		3		5		4		17
Test 7	4	1	4	3	5	3			20
Test 8	4	5	1	4	5	3	1		23
Grand Total	**31**	**10**	**17**	**12**	**28**	**11**	**24**	**5**	**138**

Figure 3.19: Success count for Test set 1 without metareasoning

that publishes the navigation manager messages. The metareasoner triggers policy 1 and policy 2 when the navigation manager publishes one of the following messages: the local planner is stuck or the local plan is infeasible. Alternatively, the metareasoner also triggers the policies if the robot is stuck in the same location for more than 20 seconds. Policy 3 uses a time-based trigger and triggers an evaluation once every 5 seconds. The evaluation checks whether an optimal planner combination is being used for the current condition of the robot.

3.8.5 Policy 1

The first meta-reasoning policy is focused on increasing the performance of a motion planner using the least amount of computation. This policy is also a benchmark for the other two policies and tests the feasibility of using a meta-reasoner for motion planning. Figure 3.19 shows the number of successes for Test set 1. It can be observed that combinations with NLOPT work very poorly under these conditions. This low success rate can be attributed to the fact that the NLOPT local planner closely follows the global plan and does not stray away from the nominal path. The global plan created for data set 1 does not recognize all the obstacles at the start and thus requires the local planner to be more flexible with its functioning.

Policy 1 is designed with the intention of increasing the success rate while utilizing the least amount of computation effort. In order to design this meta-reasoning policy, a deeper understanding of the failure conditions is obtained by plotting the location of the failures. It is observed that most of the failures occur when the robot gets too close to an obstacle. This can be seen through Figure 3.20.

In order to resolve this issue, a logic-based meta-reasoning policy is defined that creates space between the robot and the obstacle and restarts the planning algorithms once the space is created. The metareasoner creates space by taking control over the robot and moving the robot away from the obstacle. This policy waits for output from the navigation manager that shall trigger the metareasoner to restart the planners. The exact trigger conditions are mentioned in Section 3.8.4.

The improvement in the success rate on using the metareasoning policy 1 are discussed in greater detail in Section 4.

3.8.6 Policy 2

Policy 2 for meta-reasoning builds on Policy 1 and focuses on reducing the computational load on the system while maintaining the system's success rate. Policy 2 and Policy 3 are designed with a greater emphasis on data set 2.

SLGP stands out as a planner that takes the least amount of computational effort, this can be observed in Figure 3.21. Moreover, the SLGP planner successfully runs on 6 out of the 8 test scenarios presented to it. While the planner has a considerably poor performance with NLOPT, the planner works better with MPPI and is also the fastest planner in test scenario 1. For these reasons, policy 2 is focused on leveraging the strengths of the SLGP global planner. The SLGP

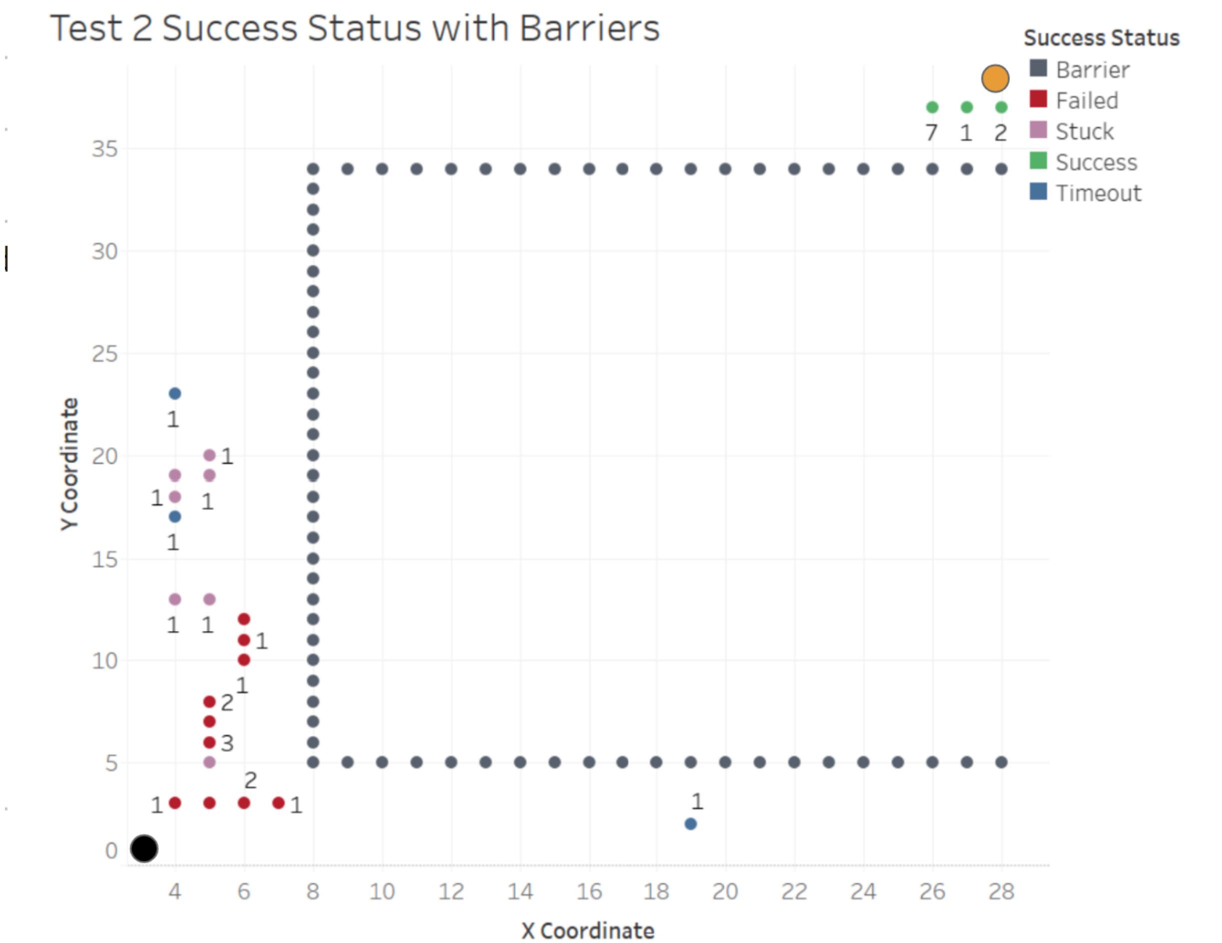


Figure 3.20: Test completion locations for Test set 1, Test scenario 2. The black dot is the start location, the orange dot is the end location. The grey dots are barriers, there is no space in between them. The red dots are failure locations, it can be noticed that they are clustered towards the barrier obstacles. The blue dots are instances where the robot timed out, their spread represents the amount the robot explored the map. The purple dots are instances where the robot was stuck. Lastly, the green dots are instances where the robot made it to the end goal, the number below the dot represents the number of times it succeeded.

AVERAGE of CPU Util	Global Planner			
Test Num	EASL	GLS	SBPL	SLGP
Test 1	51.134	55.702	49.240	48.452
Test 2	50.987	55.828	48.754	47.757
Test 3	51.336	53.877	49.015	48.284
Test 4	53.494	54.683	49.763	50.513
Test 5	59.959	55.220	58.282	47.748
Test 6	62.368	57.744	55.261	51.390
Test 7	55.669	57.420	48.863	51.754
Test 8	50.338	56.850	51.775	49.210
Grand Total	**54.410**	**55.916**	**51.369**	**49.391**

Figure 3.21: Average system utilization for Test set 2

AVERAGE of Completion time	Global Planner	Local Planner							
	EASL		GLS		SBPL		SLGP		Grand Total
Test Num	MPPI	NLOPT	MPPI	NLOPT	MPPI	NLOPT	MPPI	NLOPT	
Test 1	1:30:23	1:18:38	1:28:32	1:20:01	1:28:53	1:21:16	1:24:25	1:16:03	1:23:30
Test 2	1:37:46	1:29:24	2:00:32		1:36:21	1:30:17	1:38:54		1:35:32
Test 3	1:43:43	1:31:09	2:17:50		1:34:21	1:33:30	1:44:33		1:40:39
Test 4	1:39:22	1:29:16	1:41:46		1:40:49	1:26:50	1:40:34		1:37:09
Test 5	5:04:40				5:04:40				5:04:40
Test 6	1:57:16	1:40:07	2:03:55	1:12:48	1:42:40	1:42:20	1:56:48		1:48:09
Test 7	2:24:04	2:14:53	2:08:53		1:56:28	1:40:59			2:01:59
Test 8	1:47:19	1:41:16	2:17:34	1:53:11	1:40:34	1:36:22			1:46:16
Grand Total	**1:58:05**	**1:35:47**	**1:56:03**	**1:30:16**	**1:46:00**	**1:32:17**	**1:40:10**	**1:16:03**	**1:43:06**

Figure 3.22: Average Completion Time for Test set 2

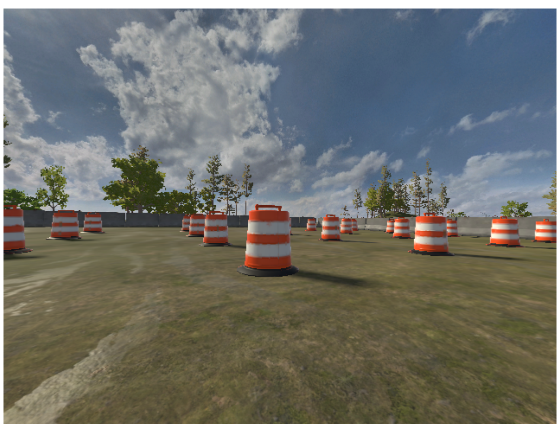
planner has a prominent downside in that it blindly follows a heuristic based on the Euclidean distance between the robot's current location and the end location. This causes the global planner to fail in test scenarios 7 and 8. To rectify this shortcoming, policy 2 switches between the SLGP global planner and the SBPL global planner. The SBPL global planner is deterministic and ensures that the robot will reach the end goal if there exists a feasible path from the robot's current location to the end goal. Furthermore, results from data set 2 highlight the benefits of using NLOPT. Planner combinations using NLOPT have faster run times. Table 3.22 illustrates the benefits of using NLOPT by depicting the average completion time for Test set 2. For these reasons, the robot switches between the SLGP MPPI planner combination to the SBPL NLOPT planner combination.

3.8.7 Policy 3

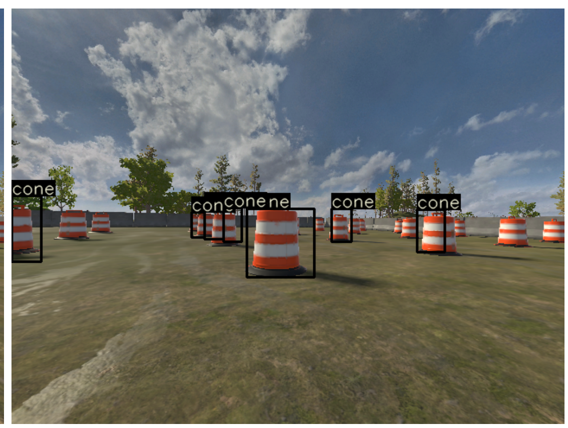
Policies 1 and 2 are rule-based policies, they are human-specified and rely on the user's insights. In order for a metareasoner to work in real-world scenarios, it needs to be self-sufficient. This requires the metareasoner to be able to learn the policy online. Furthermore, the metareasoner should be capable of storing its own experiences and using those experiences to train itself and update its policies. Policy 3 attempts to move from the paradigm of rule-based metareasoning to smart metareasoning policies that have the capacity to learn from data without human specification. In order to achieve these goals, Policy 3 builds a metareasoner based on a multi-input supervised learning model that uses neural networks for multi-class classification. This model is trained on data obtained by running the robot under the 8 test scenarios from Test set 2 without a metareasoner. The total amount of data used for the training of the model in approach 2 of Policy 3 is 35MB. The overall size of the learning model used in approach 2 of Policy 3 is 138MB. It is therefore possible for the robot to be able to store this data onboard. This model reduces the amount of human specification throughout the process by using MAVF analysis to label the data. Policy 3 takes care to use the least amount of data to train the neural networks, thus paving the path for future work in online learning of metareasoning policies. Two approaches have been attempted for policy 3. These approaches are outlined in Sections 3.8.7.1 and 3.8.7.2. Two new test scenarios are introduced to

3.8.7.1 Approach 1

It was established in Section 3.8.1 that the front camera images make an ideal data source for training the neural network. This approach uses the front camera images to detect the obstacles in the image and define the amount of *clutter* in the environment. This approach recognizes that the performance of the planners is dependent on the number of obstacles in the environment. For instance, if the environment is completely free of obstacles like in test scenario 1, the SLGP planner would make an ideal choice; the GLS algorithm reduces computational load by conducting edge evaluations only when it encounters an obstacle and can be used when there are a few obstacles, lastly, a highly cluttered environment with multiple obstacles can warrant the use of the SBPL and EASL planners. In order to test this hypothesis, an object detection model is trained to recognize cones. The amount of *clutter* in the environment is defined by the number of cones detected in the front camera image.



(a) Without Object detection



(b) With Object detection

Figure 3.23: Object detection in test scenario 6

Figure 3.23 shows an example of the implementation of this approach. An example of the same object detection model being used in a real-world scenario can be seen in Figure 3.24. This approach has many problems that are difficult to overcome. An object detection model used to detect the obstacles can potentially miss out on obstacles and thus directly affect the count of obstacles. In addition, The object detection model suffers from the problem of occlusion. Moreover, the model used for this approach is trained to detect cones, this model would have to be trained to detect multiple obstacles in order for it to succeed in real world scenarios. Lastly, there is no formal definition of obstacle clutter, making it challenging to quantify clutter and use it to evaluate the performance of planners. To avoid these problems, an alternate approach in the form of approach 2 is used for Policy 3.

(a) Without Object Detection

(b) With Object Detection

Figure 3.24: Object detection for real-world scenarios

3.8.7.2 Approach 2

Approach 2 tries to resolve the problems of Approach 1. It does so by transitioning from defining *clutter* to focusing on the amount of free space that the robot sees in the environment through the front camera image. The free space is the space that the robot can traverse in; it can be grass, gravel, or even asphalt. It is easier to recognize the free space in the environment than to recognize all the obstacles and quantify them. In order to obtain the amount of free space, the robot uses an image segmentation algorithm to separate the free space from the rest of the environment. Figure 3.27 shows the process of image segmentation. The segmented image is turned into a black-and-white image and compressed to reduce the amount of space that it occupies.

Multiple images are collected from each time the robot runs a test scenario without a metareasoner. Images are captured every 2 seconds and the number of images collected during each run depends on the time the robot takes to complete the run. This image data set is illustrated in Figure 3.25. Another data set called the metrics data set, stores the mission completion time, the system (CPU, GPU and RAM) utilization and the system temperatures (CPU and GPU) for each time the robot runs a test scenario without a metareasoner. Unlike the image data set that stores multiple images for each run of each test scenario, this data set has a single average value for each run of each test scenario. This data set is illustrated in Figure 3.26. A subset of the image data set is augmented with numerical data (robot orientation and distance with respect to the end goal) and is used to train a multi input classifier that outputs an optimal planner combination. This architecture can be seen in Figure 3.33.

An optimal planner combination is defined as the planner combination that best satisfies three mission objectives. These objectives are labelled MS1, MS2 and MS3 in Figure 3.29. The first mission objective is to minimize the mission completion time, the second objective is to reduce system utilization (CPU, GPU and RAM Utilization) and the third objective is to reduce the system temperature (CPU and GPU temperature).

Each test scenario has an optimal planner combination. Thus, there are a total of 8 optimal planner combinations. The optimal planner combination for each test scenario is identified by using a Multi Attribute Value Function (MAVF). The MAVF assigns a value to each planner combination and the planner combination with the highest overall value is the optimal planner combination. The inputs for test scenario 2 can be seen in Figures 3.28 and 3.29. For instance, to

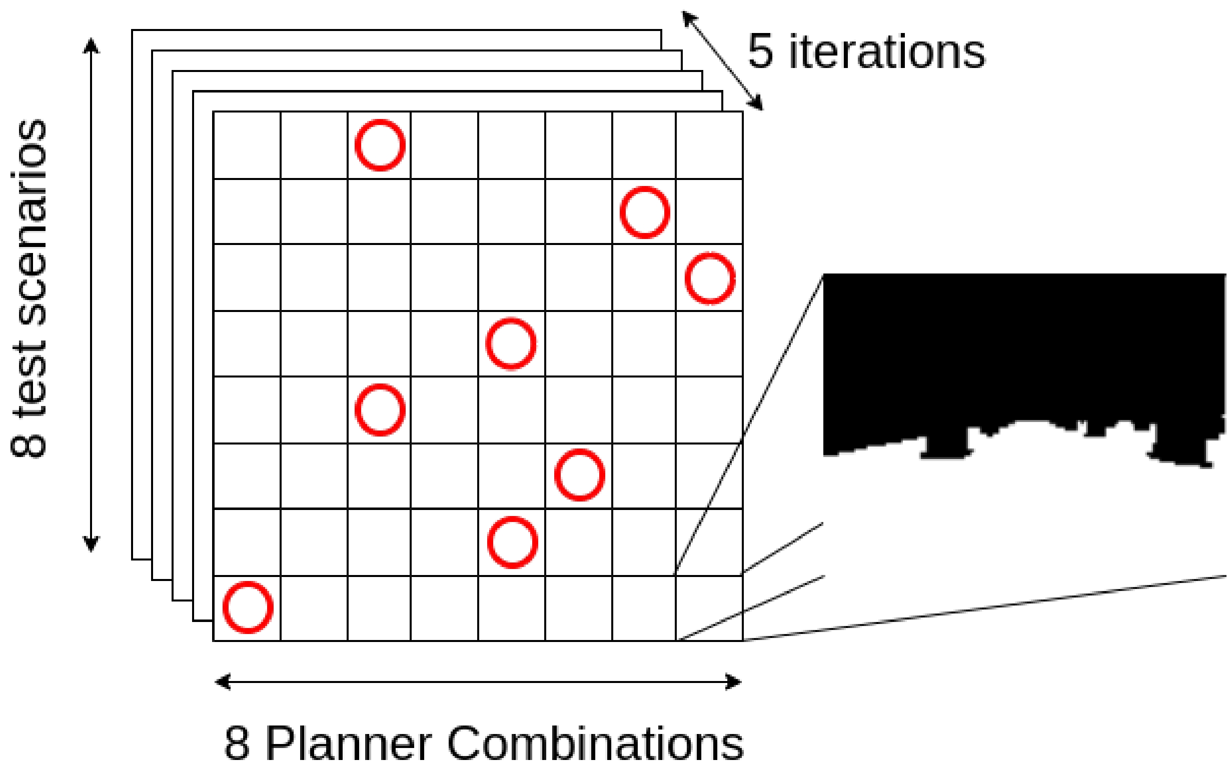


Figure 3.25: Illustration of the image data set used to train the metareasoning model in Policy 3.The red circles represent optimal planner combinations.

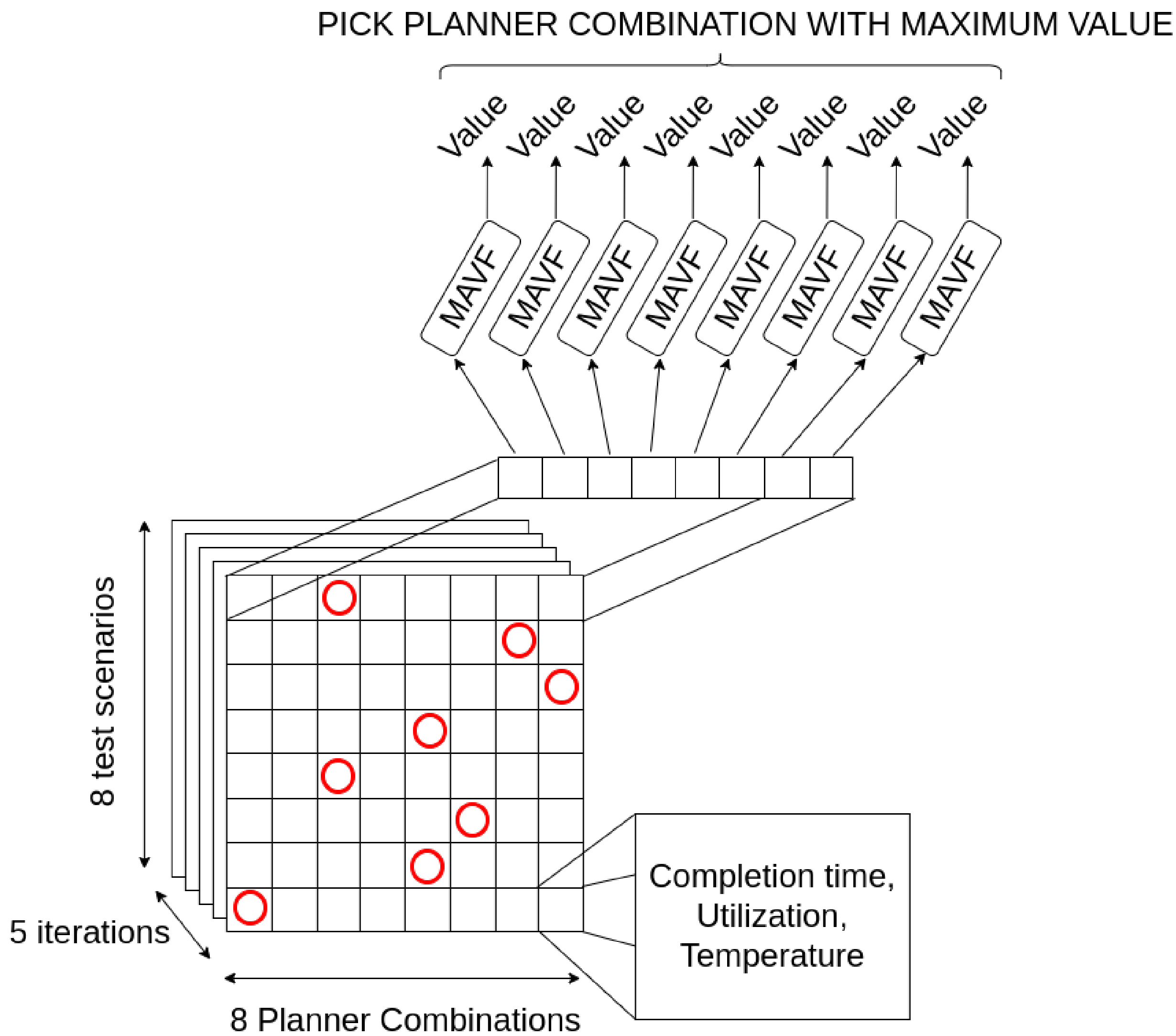
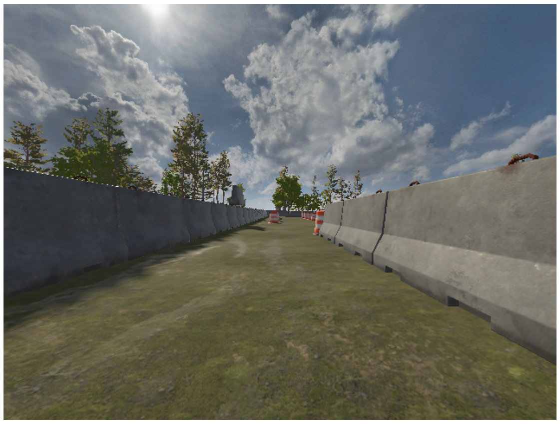
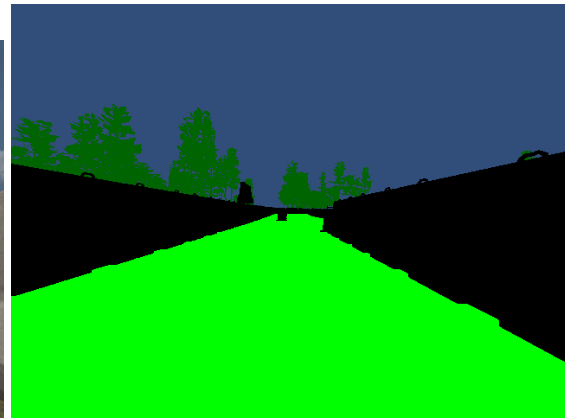


Figure 3.26: Illustration of the metric data set used to recognise the optimal planner combination for each test scenario.The red circles represent optimal planner combinations.



(a) Front camera image from the robot



(b) Semantic segmentation of the front camera image

(c) Comverting image into a black and white image and compressing the image

Figure 3.27: Image segmentation on the front camera image

calculate the value of the RAM utilization, where a lower utilization leads to a higher value the following formula is used:

$$value = \frac{(highest\_RAM\_utilization - RAM\_utilization)}{(highest\_RAM\_utilization - lowest\_RAM\_utilization)}$$

These values can be found in Figure 3.30. In order to obtain the total value (V_t), each value is multiplied by the weight and added together. (V_t) is obtained for each planner combination for each mission objective. The (V_t) for each planner combination under every mission objective and added to obatain the overall value of the planner combination. This process can be observed in Figure 3.31. The planner combination with the highest overall value is the optimal planner combination for the test scenario whose average values are the input in Figure 3.28. In Figure 3.31, all the values are color coded in the form of a heat map with green representing the highest value and red representing the lowest value. As a result we can recognize the planner combination with the highest value. In this example, the SBPL NLOPT planner combination is the optimal planner combination (planner combination with the highest value) for test scenario 6. The same method is used for every test scenario in order to obtain the 8 optimal planner combinations. Table 3.2 shows the optimal planner combination for each test scenario.

Table 3.2: Optimal planner combination for each test scenario

Test Scenario	Optimal Planner Combination
Test scenario 1	RDGP NLOPT
Test scenario 2	SBPL NLOPT
Test scenario 3	SBPL MPPI
Test scenario 4	SBPL NLOPT
Test scenario 5	RDGP MPPI
Test scenario 6	EASL NLOPT
Test scenario 7	EASL NLOPT
Test scenario 8	EASL NLOPT

The use of a MAVF reduces human decision-making and prevents bias towards a single mission objective. The MAVF allows the use of all the available data. The MAVF takes two inputs, the values from the metric data set and a ranking for each of these metrics. The input values are an average over 5 iterations for a single test scenario. Moreover, the ranking is indicative of the importance given to each metric. The ranking is provided by the user based on the mission objective. Alternatively, they may also be provided by a group of stakeholders.

Image data of these 8 optimal planner combinations form a subset of the image data set and are used to train a classifier. In order to demonstrate this subset, The 8 optimal planner combinations are marked as red circles in Figures 3.26 and 3.25.It is ensured that this subset of the image data set has equal number of images from each test scenario by randomly eliminating images from test scenarios that have more images than the test scenario with the least number of images. The image subset is roughly 20% of the complete image data set. Each image in this subset is labelled. The label is the planner combination the image belongs to. Hence, each time this classifier is faced with a situation similar to one of the test scenarios, it outputs the optimal planner combination. The image subset is divided into a training data, validation data and test data in a 70%,20% and 10% split.

Through a process of trial and error, it is observed that the image data is insufficient for the classifier to correctly choose a planner combination. Hence, the image data is augmented with two new data sources i.e. the robots orientation with respect to the goal position, illustrated in Figure 3.32 and the robots distance from the goal location. Both these data sources provide the robot with context for the front camera images. These data sources are referred to as numerical data in Figure 3.33. Hence the final model used for Policy 3 is a multi input classifier capable of taking numerical data as well as image data in the form of input and producing a choice of an optimal planner combination in the form of an output.

This model uses a multi-layer perceptron to process the numerical data and a convolutional neural network to process the image data. A summary of the architecture can be seen in Table 3.3. The total number of parameters for the model is 12,130,248. The model is run for a total of 15 epochs with a batch size of 16 to achieve a test accuracy of 89%. This accuracy can be

MEAN VALUES			Only change values in blue			
Design Options	Low => Good	Low => Good	Low => Good	Low => Good	Low => Good	Low => Good
Options	**Completion Time**	**GPU Utilization**	**CPU Utilization**	**RAM%**	**GPU Temp**	**CPU Temp**
SBPL_MPPI	118.00	31.27	50.38	11.52	68.58	82.47
SBPL_NLOPT	94.00	32.46	52.22	10.83	69.18	82.93
GLS_MPPI	115.00	32.48	55.72	10.39	69.37	82.84
GLS_NLOPT	84.00	31.55	56.03	9.45	68.17	81.40
EASL_MPPI	125.00	31.97	53.27	11.10	67.88	80.02
EASL_NLOPT	99.00	31.17	53.09	10.42	68.08	79.76
SLGP_MPPI	111.00	32.33	48.99	10.23	68.76	82.08
SLGP_NLOPT	68.00	30.74	49.43	9.08	66.53	79.65

Figure 3.28: First input to the MAVF in the form of the average values for test scenario two from Test set 2.

Mission ID	Mission Statement/Stakeholder Requirements	Completion Time	GPU Utilization	CPU Utilization	RAM%	GPU Temp	CPU Temp	Total
MS1	Finish mission in least amount of time	1	3	2	4	5	6	21
MS2	Minimum system utilization	3	2	1	4	5	6	21
MS3	Minimum system temperature	3	5	6	4	2	1	21

Mission ID	Mission Statement/Stakeholder Requirements	Completion Time	GPU Utilization	CPU Utilization	RAM%	GPU Temp	CPU Temp	Total
MS1	Finish mission in least amount of time	0.3300	0.1700	0.2500	0.1300	0.080	0.040	1.0000
MS2	Minimum system utilization	0.1700	0.2500	0.3300	0.1300	0.080	0.040	1.0000
MS3	Minimum system temperature	0.1700	0.0800	0.0400	0.1300	0.250	0.330	1.0000

Ranks	1	2	3	4	5	6	Total
ROD	0.33	0.25	0.17	0.13	0.08	0.04	1

Figure 3.29: Second input to the MAVF in the form of the ranking for each of the mission objectives

increased by increasing the number of images used for training, increasing the image resolution, and also by introducing new sources of information. Figure 3.34 shows the training progress of the model. These figures show that the validation accuracy curve starts to flatten around the 15 epoch and the model is fit to the data that we have provided it with.

A demonstration of the planner switching during the test runs is shown in Figure 3.35. Figure 3.35a shows the process of switching between two planners. Figure 3.35b depicts a situation where the robot is already running the optimal planner combination and does not require switching planners.

	Low => Good	Low => Good	Low => Good	Low => Good	Low => Good	Low => Good		
Weight ID	W1	W2	W3	W4	W5	W6		
ROD Weights	0.3300	0.1700	0.2500	0.1300	0.080	0.040		
	SVVF1	SVVF2	SVVF3	SVVF4	SVVF5	SVVF6	MAVF	
	V1	V2	V3	V4	V5	V6	Vt	Delta from 'best'
	0.123	0.695	0.803	0.000	0.278	0.140	0.387	0.129
	0.544	0.011	0.541	0.283	0.067	0.000	0.359	0.158
	0.175	0.000	0.044	0.463	0.000	0.027	0.130	0.386
	0.719	0.534	0.000	0.848	0.423	0.466	0.491	0.026
	0.000	0.293	0.392	0.172	0.525	0.887	0.248	0.269
	0.456	0.753	0.418	0.451	0.454	0.966	0.517	0.000
	0.246	0.086	1.000	0.529	0.215	0.259	0.442	0.075
	1.000	1.000	0.938	1.000	1.000	1.000	0.984	-0.468

Figure 3.30: Calculation of the value of each planner combination for mission objective 1 using swing weighting

Configuration	MS1	MS2	MS3	Overall
SBPL_MPPI	0.38725	0.45401	0.47469	1.31595
SBPL_NLOPT	0.35884	0.66959	0.82574	1.85417
GLS_MPPI	0.13021	0.47346	0.76526	1.36892
GLS_NLOPT	0.49098	0.41649	0.54259	1.45005
EASL_MPPI	0.24768	0.37102	0.12184	0.74053
EASL_NLOPT	0.51652	0.38075	0.23126	1.12853
SLGP_MPPI	0.44199	0.63000	0.34000	1.41199
SLGP_NLOPT	0.98438	0.37000	0.66000	2.01438
Best	SLGP_NLOPT	SBPL_NLOPT	SBPL_NLOPT	SBPL_NLOPT

Figure 3.31: Process of selecting the planner with the largest value

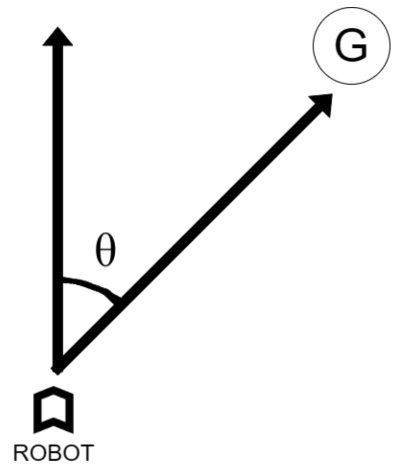


Figure 3.32: Calculating the robot's orientation with respect to the goal location. The circle with the letter G represents the goal location.

Table 3.3: Model summary

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 96, 128, 1)	0	
conv2d (Conv2D)	(None, 94, 126, 32)	320	input_1[0][0]
max_pooling2d (MaxPooling2D)	(None, 47, 63, 32)	0	conv2d[0][0]
flatten (Flatten)	(None, 94752)	0	max_pooling2d[0][0]
input_2 (InputLayer)	(None, 4)	0	
concatenate (Concatenate)	(None, 94756)	0	flatten[0][0]
			input_2[0][0]
dense (Dense)	(None, 128)	12128896	concatenate[0][0]
dense_1 (Dense)	(None, 8)	1032	dense[0][0]

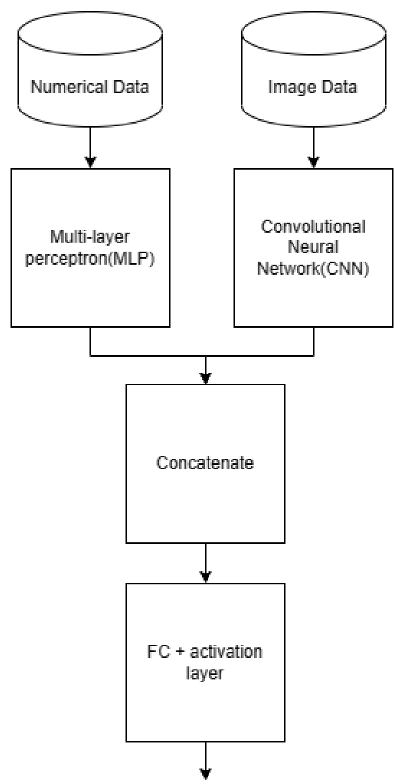
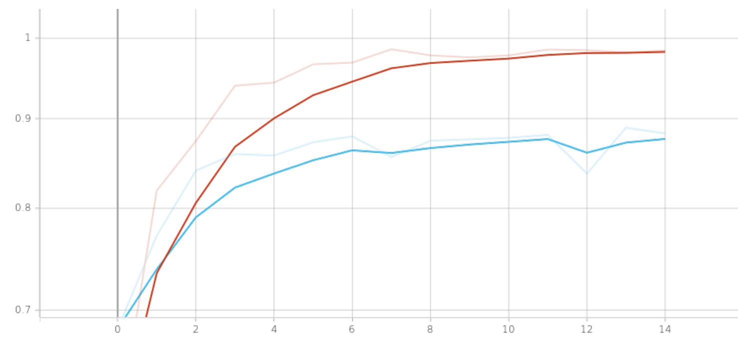
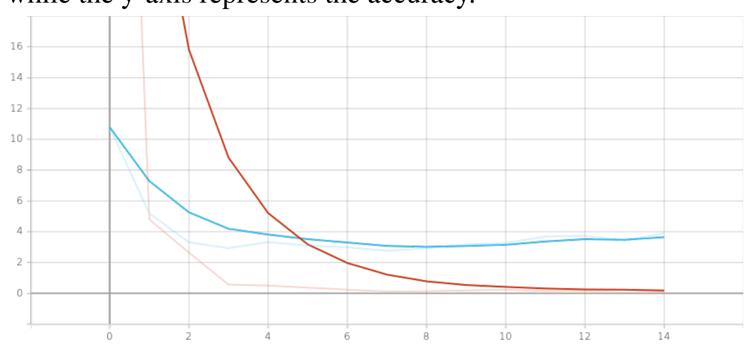


Figure 3.33: Architecture of the multi-input multi-class classifier used in Policy 3

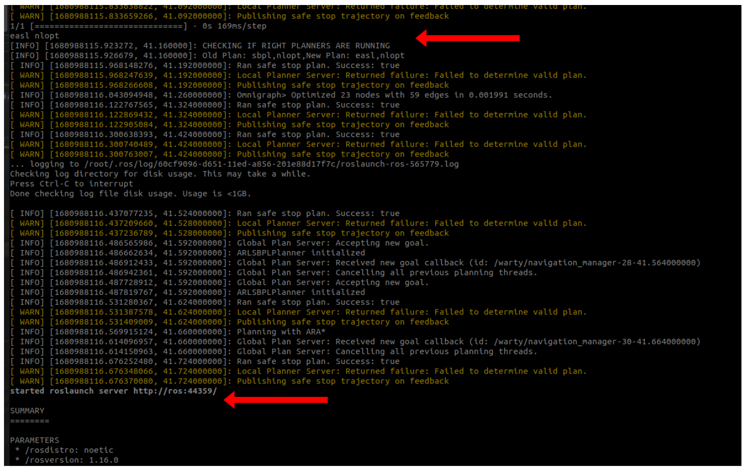


(a) Model training accuracy graph obtained through tensorboard. The X-axis represents the number of epochs while the y-axis represents the accuracy.

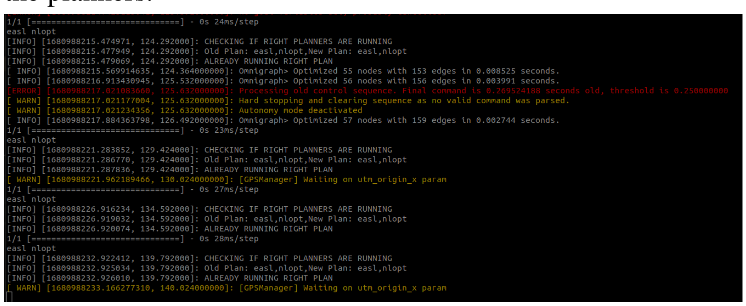


(b) Model training loss graph obtained through tensorboard. The X-axis represents the number of epochs while the y-axis represents the loss.

Figure 3.34: Model graphs obtained from tensorboard. The lines in red represent the training set while the lines in blue represent the validation set. The darker lines are smooth versions of the lighter lines.



(a) On obtaining inputs, the metareasoner decides that the robot is not running the optimal planner and switches the planners. The first arrow shows the part where the metareasoner makes the decision. The second arrow shows the actual switching of the planners.



(b) On obtaining inputs, the metareasoner decides that the robot is running the optimal planner combination, and thus, no switching takes place.

Figure 3.35: Process of switching planners

Chapter 4: Results

This chapter is divided into two subsections that focus on Test Set 1 and Test Set 2. Each subsection discusses the results of the experimental runs based on the four metrics discussed in Section 3.8.2. The 4 metrics are the success rate, completion time, systems utilization, and system temperature. In addition, to better understand the results, a more in-depth analysis of the results is conducted wherever the results differ from the initial hypotheses.

4.1 Test set 1

The robot has been run under the conditions for Test set 1, without any metareasoning policy, and for the metareasoning policy 1 and 2. For this test set, the obstacles are spawned near the robot. The cells in the robot's cost map are thus altered from an unoccupied state to an occupied state, thus leading to lower confidence in the presence of an obstacle in a given cell. It was hypothesized that Test Set 1 should lead to a lower success rate than Test Set 2. This hypothesis is proven correct and can be seen in Figure 4.1 and Figure 4.14. Furthermore, Policy 1 and Policy 2 are designed to increase the success rate of the planner combinations. As a result, the success rate increases from 43% without a meta reasoner to 47% for policy 1 and 60% for policy 2. These values are obtained considering that the total number of iterations without the meta reasoner and for policy 1 is 320, while the total number of iterations for policy 2 is 40.

T1 P0 Success Count									
	EASL		GLS		SBPL		SLGP		Grand Total
Test Num	MPPI	NLOPT	MPPI	NLOPT	MPPI	NLOPT	MPPI	NLOPT	
Test 1	5	4	5	5	5	5	5	5	39
Test 2	5						5		10
Test 3	5		1		3		5		14
Test 4	3		3		5		4		15
Test 6	5		3		5		4		17
Test 7	4	1	4	3	5	3			20
Test 8	4	5	1	4	5	3			22
Grand Total	31	10	17	12	28	11	23	5	137

Figure 4.1: Success Count for Test set 1 without metareasoning

T1 P1 Success Count									
	EASL		GLS		SBPL		SLGP		Grand Total
Test Num	MPPI	NLOPT	MPPI	NLOPT	MPPI	NLOPT	MPPI	NLOPT	
Test 1	5	5	5	5	5	5	5	5	40
Test 2	5						5		10
Test 3	5	1			3		5		14
Test 4	5		2		5		5		17
Test 6	5		2		5	1	2		15
Test 7	5	5	5	5	5	5			30
Test 8	5	5	5	2	5	5			27
Grand Total	35	16	19	12	28	16	22	5	153

Figure 4.2: Success count for Test set 1 with policy 1. These values are the combined average over both the local planners.

Test scenario 5 is missing from all the results since the robot can't succeed in test scenario 5. The NLOPT motion planner works very poorly without a meta reasoner. The success count for the NLOPT local planner has increased from 38 to 49 when using policy 1. Most of the failures for NLOPT occur when the robot gets too close to an obstacle, and the local planner can not find a feasible path. Policy 1 creates space between the robot and the obstacle, thus increasing the success rate. Similar to Policy 1, Policy 2 increases the space between the robot and the obstacle. However, policy 2 switches between the SLGP MPPI planner combination to the SBPL NLOPT planner combination.

The time of mission completion for all the test scenarios that succeeded in their missions can be seen in Figures 4.5, 4.6 and 4.7. Policies 1 and 2 have a higher mission completion time since they spend time rebooting and creating distance between the robot and the obstacle. Test scenario 7 and Test scenario 8 have significantly higher times for Policy 2. Conversely, the SLGP

T1 P2 Success Count	
Test Num	
Test 1	3
Test 2	5
Test 3	3
Test 4	5
Test 6	5
Test 7	1
Test 8	2
Grand Total	24

Figure 4.3: Success count for Test set 1 with policy 2

planner can not complete Test scenario 7 and Test scenario 8 without metareasoning or with Policy 1. The higher time for Policy 2 can be attributed to the robot moving to a corner of the map before failing with the SLGP planner and switching to the SBPL planner. This increases the time it takes for the SBPL planner to find a path to the end goal. Furthermore, this path must be corrected several times before leading the robot to the end goal. Figure 4.4 demonstrates the working of Policy 2 for Test Set 1. The green line represents the global plan, while the light blue areas are obstacles. The global plan passes through the obstacles because the obstacles are only recognized by the local cost map and not by the global cost map. When the robot approaches the obstacle, it recalculates the global plan to go around it.

Despite the initial hypothesis that using a metareasoner will decrease the system load, the CPU utilization is significantly higher when the robot runs Policy 1. This can be observed in Figure 4.9. Moreover, the CPU utilization is lower for Policy 2 than for Policy 1. This shows that CPU utilization is directly affected by the number of times the planners are changed. Policy 2 only changes the planner combination once, while there is no limitation on the number of times the planner combination can be changed with Policy 1.

The system temperatures follow the same general trends as the system utilization. Although

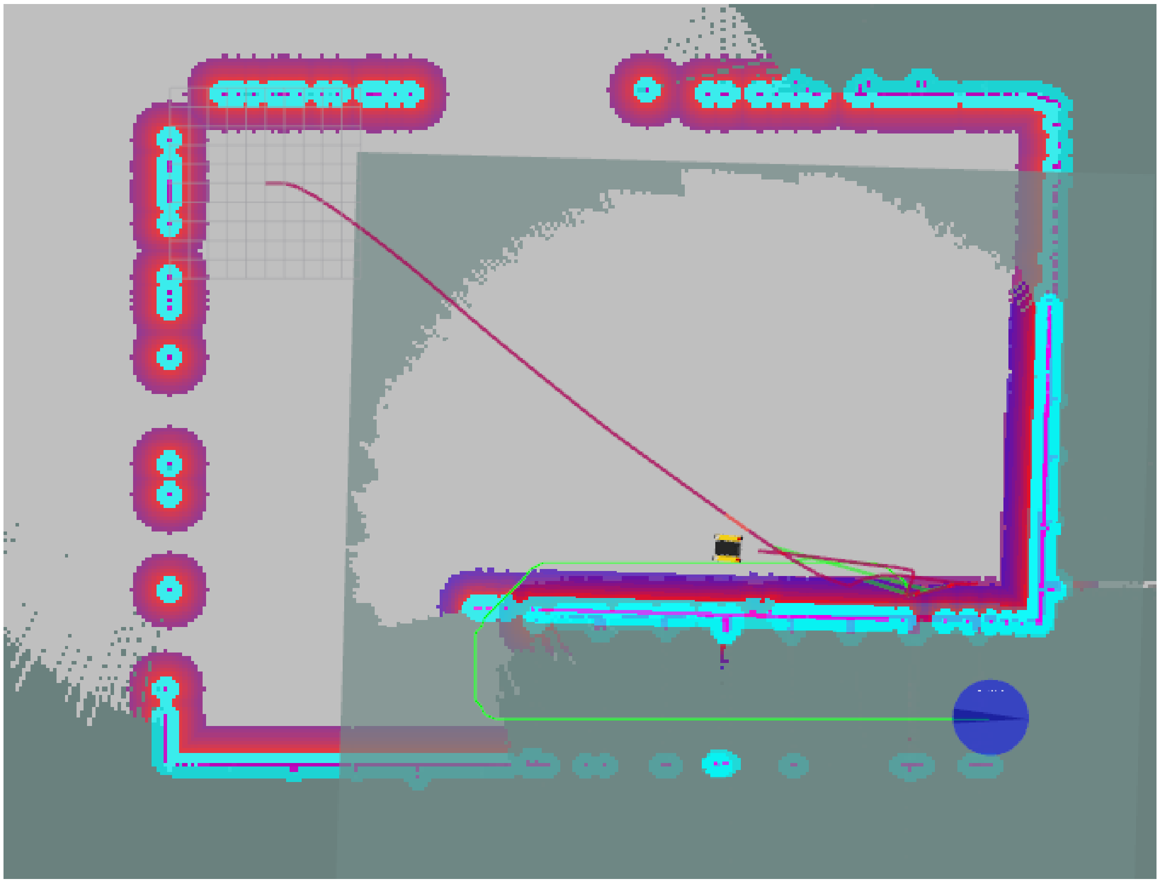


Figure 4.4: Robot movement in Test scenario 7 of Test set 1 without metareasoning

T1 P0 Success time				
Test Num	EASL	GLS	SBPL	SLGP
Test 1	1:00:41	0:51:33	0:55:16	0:52:43
Test 2	1:25:39			1:10:56
Test 3	1:21:47	3:11:04	2:26:20	1:17:18
Test 4	1:28:39	1:29:43	1:09:43	1:14:15
Test 6	1:41:18	1:40:12	1:26:30	1:40:53
Test 7	1:44:16	1:29:05	1:22:47	
Test 8	1:35:59	1:50:47	1:27:43	
Grand Total	1:26:22	1:24:37	1:20:26	1:10:19

Figure 4.5: Success Time for Test set 1 without metareasoning. These values are the combined average over every iteration run for each global planner without metareasoning.

T1 P1 Success Time				
Test Num	EASL	GLS	SBPL	SLGP
Test 1	1:12:28	1:01:54	1:03:53	1:00:34
Test 2	1:37:19			1:22:03
Test 3	1:24:11		1:38:52	1:39:07
Test 4	1:37:35	1:20:48	1:25:42	1:28:43
Test 6	1:55:34	1:52:34	1:33:48	1:47:52
Test 7	1:35:30	1:53:11	1:33:09	
Test 8	1:42:56	1:49:52	1:30:31	
Grand Total	1:33:28	1:33:46	1:25:32	1:20:24

Figure 4.6: Success Time for Test set 1 with policy 1. These values are the combined average over every iteration run for each global planner with metareasoning Policy 1.

T1 P2 Success Time	
Test Num	
Test 1	1:06:15
Test 2	1:25:17
Test 3	1:32:51
Test 4	1:25:22
Test 6	1:48:00
Test 7	2:17:32
Test 8	1:59:48
Grand Total	1:33:39

Figure 4.7: Success Time for Test set 1 with policy 2. These values are the combined average over 5 iterations run with the metareasoning policy 2.

T1 P0 System Load			
Test Num	AVERAGE of CPU Util	AVERAGE of RAM_%	AVERAGE of GPU_util
Test 1	53.40	8.83	33.84
Test 2	48.19	9.12	32.27
Test 3	42.59	8.18	28.33
Test 4	43.46	7.93	28.47
Test 5	47.20	9.17	31.93
Test 6	42.94	8.16	29.12
Test 7	51.34	9.42	31.22
Test 8	51.29	9.58	33.49
Grand Total	47.53	8.80	31.08

Figure 4.8: System Load for Test set 1 without metareasoning. These values are the combined average over every iteration run without a metareasoner.

T1 P1 System Load			
Test Num	AVERAGE of CPU Util	AVERAGE of RAM_%	AVERAGE of GPU_util
Test 1	54.752	9.178	35.189
Test 2	61.058	9.439	34.781
Test 3	63.329	10.860	30.968
Test 4	62.587	12.745	33.383
Test 5	65.673	14.456	33.254
Test 6	64.651	16.195	30.895
Test 7	68.194	9.789	29.748
Test 8	68.772	10.322	26.691
Grand Total	63.627	11.623	31.864

Figure 4.9: System Load for Test set 1 with policy 1. These values are the combined average over every iteration run with metareasoning policy 1.

T1 P2 System Load			
Test Num	AVERAGE of CPU Util	AVERAGE of RAM_%	AVERAGE of GPU_util
Test 1	53.576	11.934	52.119
Test 2	53.881	12.468	51.725
Test 3	52.921	12.677	54.624
Test 4	53.870	12.708	50.028
Test 5	53.319	13.685	55.138
Test 6	50.915	13.476	33.739
Test 7	48.551	12.777	31.454
Test 8	51.082	13.231	32.941
Grand Total	52.264	12.870	45.221

Figure 4.10: System Load for Test set 1 with policy 2. These values are the combined average over 5 iterations run with the metareasoning policy 2.

the system temperatures are the highest for Policy 1, the temperatures for Policy 2 are approximately 2 degrees higher than that for the system running without a meta reasoner.

4.2 Test set 2

The Test Set 2 is run without meta-reasoning and with three meta-reasoning policies. The first two metareasoning policies are rule-based, while the third meta-reasoning policy is smart and develops its own policy from past experiences. For Test Set 2, the obstacles are spawned away from the robot, and the robot drives toward the obstacles before starting its mission. The cells in the cost map turn from an unknown status to an occupied one; this gives the global cost map greater confidence in the obstacles. Consequently, the global plan avoids all the obstacles leading to a much higher success rate for Test Set 2 than for Test Set 1. The success count for Test Set 1 and Test Set 2 can be compared in Figures 4.1 and 4.14. The success rate for Test Set 1 without a metareasoner is 43% while that of Test Set 2 without a metareasoner is 52%. Similar to Test Set 1, the success rate increases when using a metareasoning policy. The success rate for

T1 P0 System Temperatures		
Test Num	AVERAGE of CPU_temp	AVERAGE of GPU_temp
Test 1	81.88	70.97
Test 2	76.31	66.36
Test 3	70.67	60.73
Test 4	68.99	60.54
Test 5	75.18	65.08
Test 6	71.17	61.07
Test 7	84.62	70.13
Test 8	83.86	70.13
Grand Total	76.57	65.61

Figure 4.11: System Temperatures for Test set 1 without metareasoning. These values are the combined average over every iteration run without a metareasoner.

T1 P1 System Temperatures		
Test Num	AVERAGE of CPU_temp	AVERAGE of GPU_temp
Test 1	84.472	74.511
Test 2	82.239	72.191
Test 3	80.831	69.990
Test 4	81.470	71.170
Test 5	82.638	71.204
Test 6	81.455	70.693
Test 7	85.156	70.469
Test 8	82.739	69.968
Grand Total	82.625	71.274

Figure 4.12: System Temperatures for Test set 1 with policy 1. These values are the combined average over every iteration run with metareasoning Policy 1.

T1 P2 System Temperatures		
Test Num	AVERAGE of CPU_temp	AVERAGE of GPU_temp
Test 1	75.103	65.318
Test 2	75.919	68.093
Test 3	76.336	68.168
Test 4	76.377	67.956
Test 5	77.476	68.781
Test 6	81.684	69.267
Test 7	82.053	69.576
Test 8	84.406	69.759
Grand Total	78.669	68.365

Figure 4.13: System Temperatures for Test set 1 with policy 2. These values are the combined average over 5 iterations run with the metareasoning policy 2.

policy 1 is 55%, for policy 2 is 67.5%, and for policy 3 is 65%. This increase in the success rate when using a metareasoner justifies the use of a metareasoner. It can be observed that the NLOPT local planner works better under Test Set 2 than Test Set 1. The success count is increased for all the planners when using Policy 1. The most observable differences occur in test scenarios 6, 7, and 8. These test scenarios are challenging for the robot since they include multiple obstacles and an alleyway the robot must pass to reach the end goal. These problems are alleviated when the robot creates space between itself and the obstacle and restarts the planners. Policy 2 for Test Set 2 has the highest success rate at 67.5%. If we ignore test scenario 5, since it is impossible to succeed in this test scenario, this rate rises to 77.1%. The planner selection for policy 2 has been decided based on human observations made when running the test scenarios without a metareasoner. Policy 3 makes its own planner decisions and shows comparable results to Policy 2. Moreover, there are multiple ways in which the performance of Policy 3 can be increased. These

T2 P0 Success Count									
	EASL		GLS		SBPL		SLGP		Grand Total
Test Num	MPPI	NLOPT	MPPI	NLOPT	MPPI	NLOPT	MPPI	NLOPT	
Test 1	5	5	5	5	5	5	4	5	39
Test 2	5	3			4	5	5		22
Test 3	5	4	1		5	5	4		24
Test 4	5	4	1		4	2	5		21
Test 6	3	4	3	1	5	1	3		20
Test 7	3	3	3		5	5			19
Test 8	4	5	1	3	3	4			20
Grand Total	30	28	14	9	31	27	21	5	165

Figure 4.14: Success Count for Test set 2 without metareasoning

T2 P1									
	EASL		GLS		SBPL		SLGP		Grand Total
Test Num	MPPI	NLOPT	MPPI	NLOPT	MPPI	NLOPT	MPPI	NLOPT	
Test 1	5	5	5	4	5	5	5	5	39
Test 2	5	5		1	4	5	5		25
Test 3	5	4	2	4	4	5	5		29
Test 4	4	2	3		5	5	4		23
Test 6	4	4	1		3	1	4	1	18
Test 7	4	3	1	2	5	3			18
Test 8	5	5	5		5	4			24
Grand Total	32	28	17	11	31	28	23	6	176

Figure 4.15: Success count for Test set 2 with Policy 1

are discussed in Section 3.8.7.

The completion time for Test Set 2 is higher than that for Test Set 1 because it needs to travel an extra patch of road and shift from manual control to control using the planner combinations. This accounts for an extra 25 seconds in the run time. On subtracting the extra 25 seconds from the results for Test Set 2, the overall performance of Test Set 2 is faster than that of Test Set 1. This is as expected since the robot does not have to undergo the re-planning process as often

T2 P2	
Test Num	
Test 1	5
Test 2	3
Test 3	4
Test 4	5
Test 6	4
Test 7	2
Test 8	4
Grand Total	27

Figure 4.16: Success count for Test set 2 with Policy 2

T2 P3	
Test Num	
Test 1	5
Test 2	1
Test 3	5
Test 4	4
Test 6	5
Test 7	3
Test 8	3
Grand Total	26

Figure 4.17: Success count for Test set 2 with Policy 3

in Test Set 2 as in Test Set 1. The success times for policy 1 are comparable to those without running a metareasoner. This shows that policy 1 did not undergo the process of rebooting as often as it did for Test set 1. Despite the initial hypothesis that policy 2 should run faster because it uses the SLGP global planner, the overall timings for policy 2 are comparable to the timings when running the EASL and SBPL. Policy 3 improves on these timings and has significantly lower timings. Figures 4.22, 4.23, 4.24, 4.25 provide deeper insights into these results. It can be seen in Figures 4.22 and 4.23 that without metareasoning and with Policy 1 the SLGP planner can not complete the test scenario 7. Policy 2 uses the SLGP planner since it is the fastest planner and takes the least computational load. This can be observed in Figure 4.24. On switching to the SBPL planner, the robot takes a turn and can recognize a path through the alleyway and to the goal. For these reasons, the robot can complete test scenario 7 when using policy 2. In Figure 4.25, it can be observed that the robot begins the run with the SBPL NLOPT planner. The decision-making process and why it chose the SBPL NLOPT planner combination can be observed in Figure 3.31. On approaching an obstacle, the robot switches the planners; this can be observed in Figure 4.25 as a kink in the otherwise smooth path that the robot has traced to reach the end goal. The planner the robot switches to is EASL MPPI which takes more computational

T2 P0 Success Time				
Test Num	EASL	GLS	SBPL	SLGP
Test 1	1:24:30	1:24:16	1:25:04	1:19:46
Test 2	1:34:33		1:33:03	1:38:54
Test 3	1:38:08	1:43:40	1:33:55	1:44:33
Test 4	1:34:52	1:36:16	1:36:09	1:40:34
Test 6	1:47:28	1:51:08	1:42:37	1:56:48
Test 7	2:19:29	2:13:19	1:48:44	
Test 8	1:43:57	1:54:59	1:37:18	
Grand Total	1:41:05	1:42:03	1:36:21	1:35:32

Figure 4.18: Success Time for Test set 2 without metareasoning. These values are the combined average over every iteration run for each global planner without a metareasoner.

effort but ensures that the robot shall reach the end goal. These images illustrate the benefit of using a meta-reasoner for motion planning in ground robots.

Despite improved success rate and completion times, it can be observed in Figures 4.26, 4.27, 4.28 and 4.29 that using a metareasoner causes an increase the system load. The results for Test Set 2 follow the same trends as Test Set 1. The marginally higher numbers for Test Set 2 can be attributed to a greater number of Test runs being successful for Test Set 2. Furthermore, Test Set 2 also marks a higher number of runs for test scenario 6, which has a notably higher system load since it has multiple obstacles and requires calculating plans numerous times.

Similar to Test Set 1, the system temperatures for Test Set 2 follow the same general trends as the system utilization. There is neither a significant spike in the system utilization nor the system temperatures when using policy 3. This validates the feasibility of transitioning from rule-based meta reasoners to smart meta reasoners capable of making their own decisions.

T2 P1 Success Time				
Test Num	EASL	GLS	SBPL	SLGP
Test 1	1:26:10	1:15:08	1:24:08	1:23:48
Test 2	1:35:36	2:06:20	1:33:25	1:38:22
Test 3	1:34:19	1:42:20	1:38:59	2:05:08
Test 4	1:34:23	1:56:56	1:33:08	1:37:04
Test 6	1:48:18	1:45:44	1:43:08	1:40:53
Test 7	2:22:56	2:16:33	1:58:53	
Test 8	1:39:18	2:04:42	1:33:44	
Grand Total	1:41:33	1:43:48	1:36:48	1:38:13

Figure 4.19: Success Time for Test set 2 with Policy 1. These values are the combined average over every iteration run for each global planner with metareasoning Policy 1.

T2 P2 Success Time	
Test Num	
Test 1	1:19:24
Test 2	1:34:57
Test 3	1:45:51
Test 4	1:38:36
Test 6	1:58:32
Test 7	2:10:52
Test 8	2:10:25
Grand Total	1:45:46

Figure 4.20: Success Time for Test set 2 with Policy 2. These values are the combined average over 5 iterations run with the metareasoning Policy 2.

T2 P3 Success Time	
Test Num	
Test 1	1:23:55
Test 2	1:30:44
Test 3	1:34:49
Test 4	1:26:02
Test 6	1:56:13
Test 7	1:38:28
Test 8	1:42:00
Grand Total	1:36:35

Figure 4.21: Success Time for Test set 2 with Policy 2. These values are the combined average over 5 iterations run with the metareasoning Policy 2.

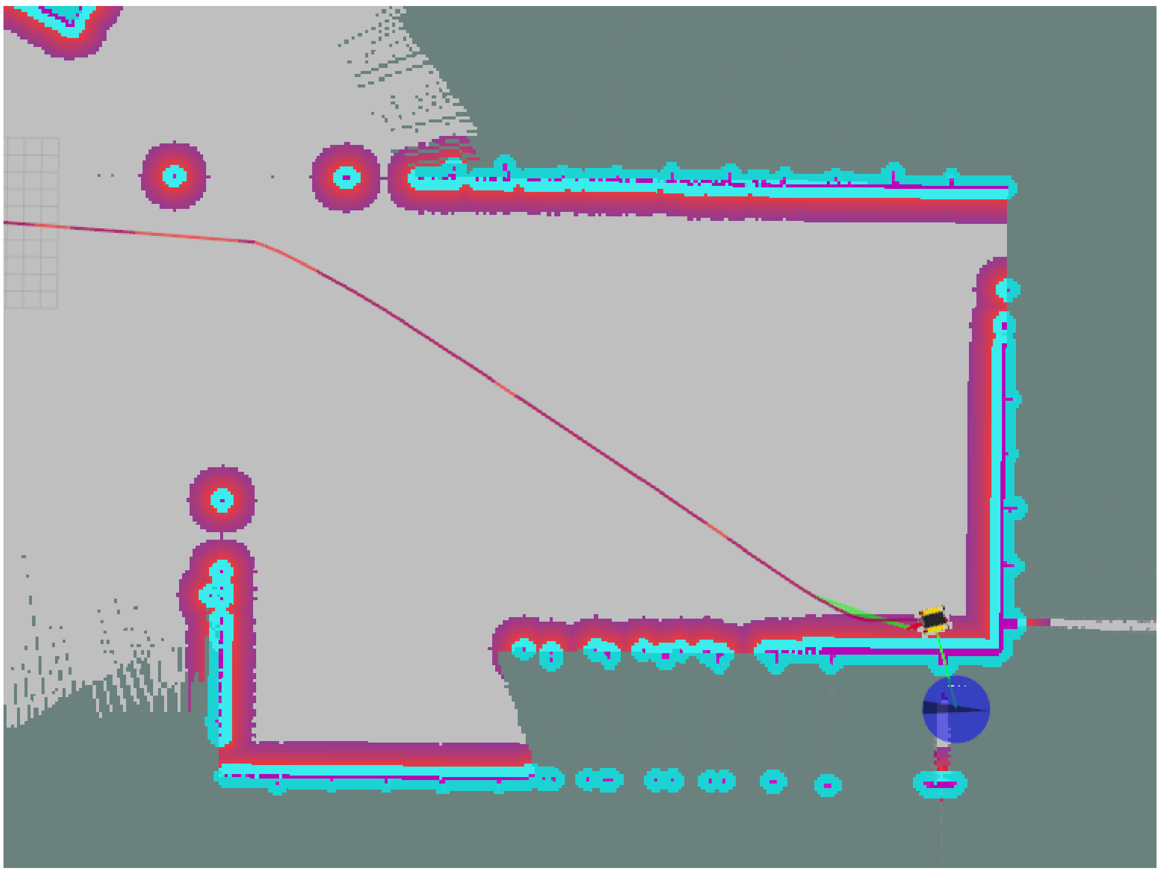


Figure 4.22: Robot movement in Test scenario 7 of Test set 2 without metareasoning

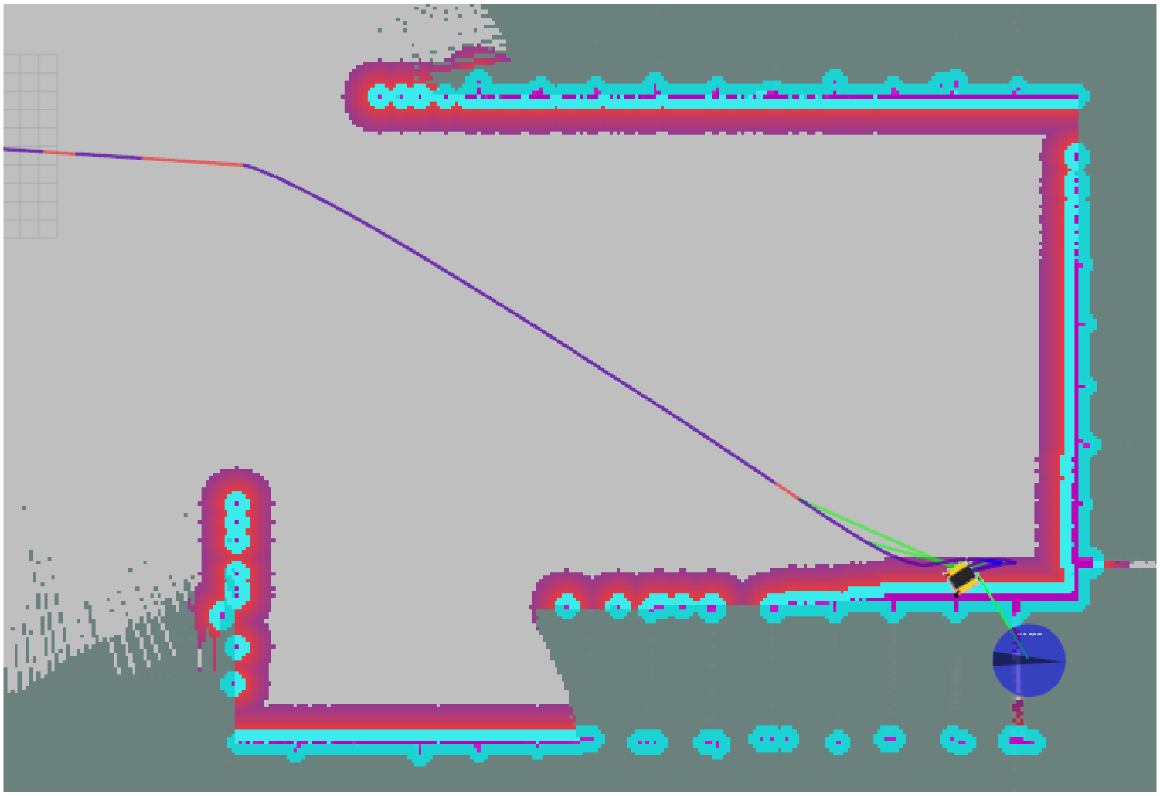


Figure 4.23: Robot movement in Test scenario 7 of Test set 2 with policy 1

4.3 Comparing Policy 2 and Policy 3

In order to compare Policy 2 and Policy 3, two new test scenarios were generated. These test scenarios are test scenarios 9 and 10. Test scenarios 9 and 10 can be observed in Figure 3.16. Policy 3 was not trained on images from these test scenarios. Hence, these results verify that Policy 2 as well as Policy 3 can be generalized. Complete results for Policy 2 and Policy 3 can be observed in Figures 4.34 and 4.35 respectively. Policy 3 has a success rate of 60% while Policy 2 has a success rate of 50%. This difference in the success rate signifies that Policy 3 generalizes to new test scenarios better than Policy 2. This can be attributed to the fact that Policy 2 is rule-based and works independently of the testing conditions. The metareasoner switches between the planners when they fail and is not concerned with the robot's situation when

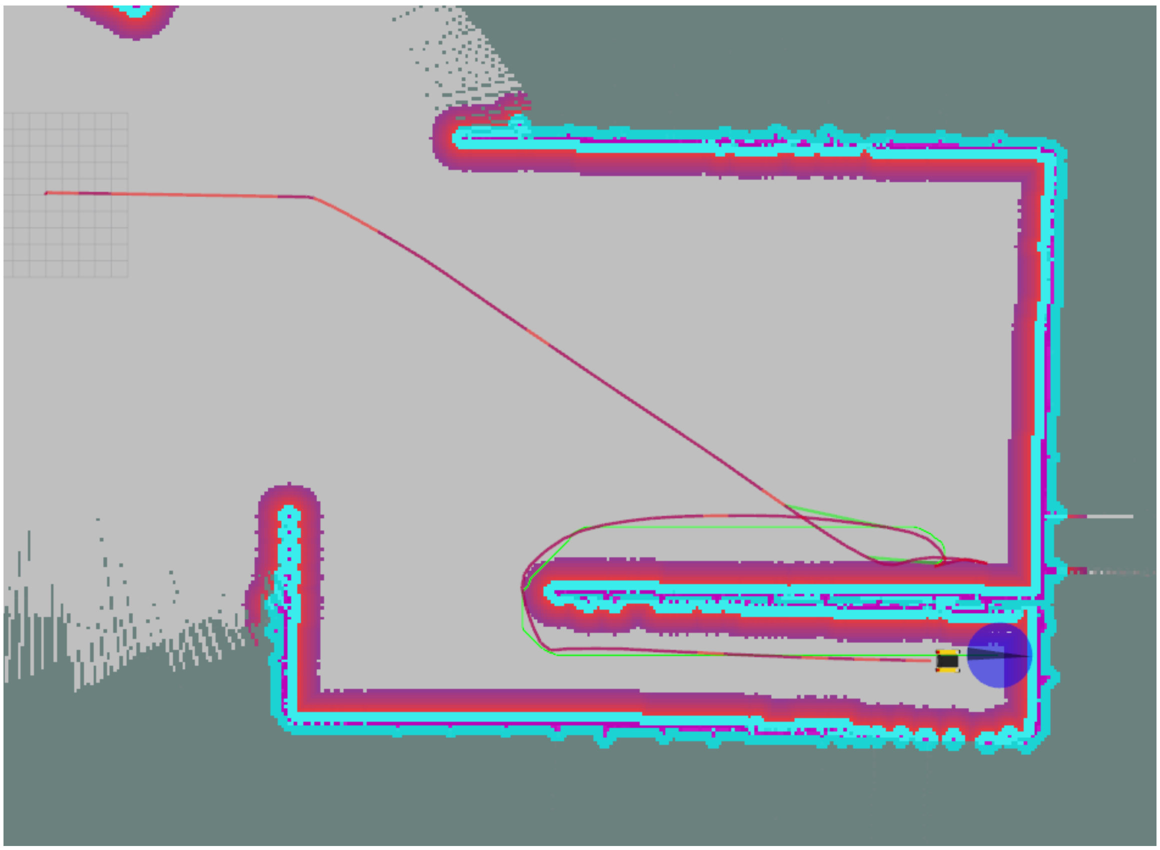


Figure 4.24: Robot movement in Test scenario 7 of Test set 2 with policy 2

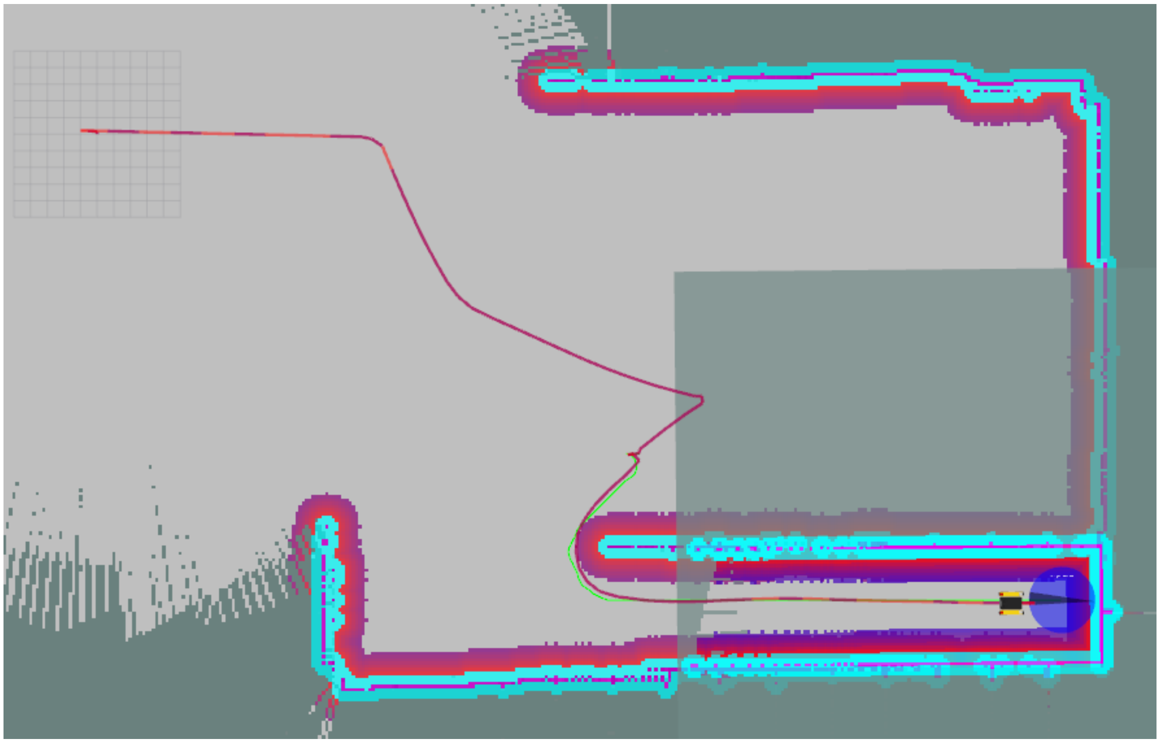


Figure 4.25: Robot movement in Test scenario 7 of Test set 2 with policy 3

T2 P0 System Load			
Test Num	AVERAGE of CPU Util	AVERAGE of RAM_%	AVERAGE of GPU_util
Test 1	51.132	9.979	32.090
Test 2	50.832	10.121	32.922
Test 3	50.628	10.017	30.575
Test 4	52.113	9.955	31.087
Test 5	55.302	10.927	32.093
Test 6	56.691	10.272	30.701
Test 7	53.426	10.729	31.985
Test 8	52.116	10.691	33.008
Grand Total	52.782	10.335	31.804

Figure 4.26: System Load for Test set 2 without metareasoning. These values are the combined average over every iteration run without metareasoning.

T2 P1 System Load			
Test Num	AVERAGE of CPU Util	AVERAGE of RAM_%	AVERAGE of GPU_util
Test 1	54.304	9.181	36.279
Test 2	53.128	9.541	33.360
Test 3	52.542	9.320	32.361
Test 4	63.674	9.557	30.929
Test 5	59.634	10.019	28.011
Test 6	69.446	10.012	26.392
Test 7	53.837	9.980	31.874
Test 8	72.308	9.974	31.505
Grand Total	59.859	9.698	31.339

Figure 4.27: System Load for Test set 2 with Policy 1. These values are the combined average over every iteration run with metareasoning Policy 1.

T2 P2 System Load			
Test Num	AVERAGE of CPU Util	AVERAGE of RAM_%	AVERAGE of GPU_util
Test 1	59.681	9.782	33.409
Test 2	59.045	10.014	29.489
Test 3	60.341	11.056	28.583
Test 4	61.117	11.099	33.836
Test 5	59.808	10.098	25.191
Test 6	62.030	11.951	32.285
Test 7	61.241	11.130	27.519
Test 8	60.237	11.721	28.548
Grand Total	60.437	10.856	29.858

Figure 4.28: System Load for Test set 2 with Policy 2. These values are the combined average over 5 iterations run with the metareasoning Policy 2.

T2 P3 System Load			
Test Num	AVERAGE of CPU Util	AVERAGE of RAM_%	AVERAGE of GPU_util
Test 1	61.206	10.507	31.828
Test 2	61.150	10.828	31.228
Test 3	58.990	11.284	30.464
Test 4	61.485	10.802	33.216
Test 5	61.953	11.227	32.414
Test 6	61.829	10.697	32.851
Test 7	59.963	11.110	33.575
Test 8	61.723	12.309	32.932
Grand Total	61.037	11.095	32.313

Figure 4.29: System Load for Test set 2 with Policy 3. These values are the combined average over 5 iterations run with the metareasoning Policy 3.

T2 P0 System Temperatures		
Test Num	AVERAGE of CPU_temp	AVERAGE of GPU_temp
Test 1	81.612	70.171
Test 2	80.784	69.549
Test 3	80.550	68.886
Test 4	76.376	67.408
Test 5	75.410	67.257
Test 6	75.301	66.831
Test 7	76.462	67.533
Test 8	79.037	68.515
Grand Total	78.189	68.268

Figure 4.30: System Temperatures for Test set 2 without metareasoning. These values are the combined average over every iteration run without metareasoning.

T2 P1 System Temperatures		
Test Num	AVERAGE of CPU_temp	AVERAGE of GPU_temp
Test 1	77.295	68.869
Test 2	80.713	69.981
Test 3	82.211	70.097
Test 4	80.626	69.874
Test 5	81.874	69.622
Test 6	80.191	69.260
Test 7	81.290	69.336
Test 8	81.227	69.995
Grand Total	80.679	69.629

Figure 4.31: System Temperatures for Test set 2 with Policy 1. These values are the combined average over every iteration run with metareasoning Policy 1.

T2 P2 System Temperatures		
Test Num	AVERAGE of CPU_temp	AVERAGE of GPU_temp
Test 1	79.639	69.780
Test 2	80.015	70.238
Test 3	80.326	70.260
Test 4	80.266	68.852
Test 5	79.909	68.542
Test 6	80.230	68.830
Test 7	79.770	68.868
Test 8	79.712	69.008
Grand Total	79.983	69.297

Figure 4.32: System Temperatures for Test set 2 with Policy 2. These values are the combined average over 5 iterations run with the metareasoning Policy 2.

T2 P3 System Temperatures		
Test Num	AVERAGE of CPU_temp	AVERAGE of GPU_temp
Test 1	80.879	70.638
Test 2	80.355	69.568
Test 3	78.906	68.984
Test 4	79.866	69.667
Test 5	80.509	69.732
Test 6	80.630	69.634
Test 7	80.570	68.233
Test 8	80.692	68.617
Grand Total	80.301	69.384

Figure 4.33: System Temperatures for Test set 2 with Policy 3. These values are the combined average over 5 iterations run with the metareasoning Policy 3.

the failure occurs. However, Policy 3 uses a smart metareasoner that avoids failure conditions by running the optimal planner combination for every situation the robot finds itself in. The CPU utilization, GPU utilization, RAM utilization, CPU temperatures and the GPU temperatures follow similar trends to the results obtained for the first 16 test scenarios. A summary of the results for test scenario 9 and test scenario 10 can be found in Figure 4.36.

Global Planner	Local Planner	Iter Num	Test Num	Completion time	Completion location	Success status	CPU Util	CPU_temp	RAM_%	GPU_util	GPU_temp
SLGP	MPPI	iter1	Test 9	2:39:20	26,37	Success	51.807	79.107	11.768	30.258	68.001
SLGP	MPPI	iter2	Test 9	1:02:20	-1,04	Stuck	50.217	78.164	10.612	29.570	66.746
SLGP	MPPI	iter3	Test 9	1:45:20	27,37	Success	58.818	80.109	11.171	28.332	69.004
SLGP	MPPI	iter4	Test 9	1:50:24	15,33	Failed	53.336	78.831	12.095	25.275	67.532
SLGP	MPPI	iter5	Test 9	2:22:32	23,19	Failed	49.945	79.165	10.500	43.676	67.500
SLGP	MPPI	iter1	Test 10	0:59:00	3,1	Failed	53.905	78.109	10.414	28.547	67.953
SLGP	MPPI	iter2	Test 10	0:57:24	1,2	Failed	53.589	77.797	10.483	23.719	65.719
SLGP	MPPI	iter3	Test 10	1:19:00	26,36	Success	54.486	79.621	10.825	32.031	68.000
SLGP	MPPI	iter4	Test 10	1:38:40	26,36	Success	51.699	77.767	10.981	37.459	67.999
SLGP	MPPI	iter5	Test 10	1:30:00	26,36	Success	56.107	81.734	10.410	29.150	69.133

Figure 4.34: Complete results for test scenario 9 and 10 when using Policy 2.

Global Planner	Local Planner	Iter Num	Test Num	Completion time	Completion location	Success status	CPU Util	CPU_temp	RAM_%	GPU_util	GPU_temp
SBPL	NLOPT	iter1	Test 9	1:30:28	14,33	Failed	64.472	82.131	10.687	35.501	73.359
SBPL	NLOPT	iter2	Test 9	2:46:20	25,36	Success	55.235	73.242	12.098	30.505	65.000
SBPL	NLOPT	iter3	Test 9	2:18:00	27,9	Failed	63.116	75.698	10.597	29.575	66.508
SBPL	NLOPT	iter4	Test 9	2:45:08	25,36	Success	53.723	71.478	11.920	29.153	64.157
SBPL	NLOPT	iter5	Test 9	2:52:20	26,36	Success	55.650	72.223	11.499	30.161	64.000
SBPL	NLOPT	iter1	Test 10	1:32:40	27,36	Success	55.679	72.393	10.591	26.818	63.990
SBPL	NLOPT	iter2	Test 10	0:42:40	4,0	Stuck	71.433	76.922	9.695	24.859	64.922
SBPL	NLOPT	iter3	Test 10	0:51:08	1,1	Failed	72.814	77.359	10.511	27.016	66.594
SBPL	NLOPT	iter4	Test 10	1:45:20	26,36	Success	54.222	73.879	11.145	30.988	64.124
SBPL	NLOPT	iter5	Test 10	1:40:08	27,36	Success	54.162	72.449	10.935	42.615	63.988

Figure 4.35: Complete results for test scenario 9 and 10 when using Policy 3.

	Success count	Avg completion time	Avg CPU Util	Avg CPU_temp	Avg RAM_%	Avg GPU_util	Avg GPU_temp
Policy 2	5	1:36:24	53.391	79.041	10.926	30.802	67.759
Policy 3	6	1:52:25	60.050	74.777	10.968	30.719	65.664

Figure 4.36: Summary of results for test scenario 9 and test scenario 10.

Chapter 5: Conclusion

Through a methodological approach, this thesis evaluates the performance of various global planner and local planner combinations on 16 test scenarios that can be combined to form a variety of real-world scenarios. This research results can be used to predict the performance of the various planner combinations under various real-world scenarios built from the 16 test scenarios. Moreover, the planners used for this research are the most current developments in motion planning and provide insights into an architecture involving a navigation manager and cost maps for motion planning. These insights include details about this architecture's shortcomings and strengths, focusing on switching between planning algorithms to achieve higher success rates. This study can be extended by creating more test scenarios and increasing the number of iterations to cover a wider variety of real-world scenarios and increase the data fidelity.

This thesis proposes three different approaches to developing a metareasoner for motion planning. The metareasoner developed through this research reasons the optimal global and local planner combination based on data accumulated by running the robot under the different test scenarios. Using a basic metareasoner that creates space between the robot and the obstacle and restarts the planners has been shown to improve the success rate while compromising the completion time, system utilization (CPU, GPU and RAM utilization), and system temperature (CPU and GPU temperatures). The subsequent two metareasoning policies are improvements on this approach. The first improvement is Policy 2. This data-driven policy uses the data from running the robots under different test scenarios. This data is used to make an informed decision about the choice of planners. Policy 2 shows improvement over policy 1 and increases the success rate from 47% to 60% for test set 1 and from 55% to 67.5% for test set 2. However, this process involves human specification making it inflexible. Furthermore, not all real world scenarios can be re-created as a combination of the 16 test scenarios used for this research. Policy 3 improves on Policy 2 by shifting from a rule-based metareasoner to a smart metareasoner capable of learning its own policy. Policy 3 uses a multi-input neural network-based metareasoner and stores the policy as the neural network's weights. Policy 3 has numerous benefits over Policy 2. This policy can learn from the experiences it collects from running different test scenarios. Since this policy uses supervised learning, the data needs to be labeled. Instead of relying on human decision-making to label the data, a MAVF is used to label the data. This makes labeling the data objective and free of human specification. The total size of the data used to train the model is 35MB. The low memory requirements of the data make it possible for the robot to store the data, thus making it possible for the robot to learn the meta-reasoning policy online.

In conclusion, this thesis shows the benefits of using a metareasoner for motion planning and successfully demonstrates three different metareasoning policies, each improving the robot's performance. In addition, this thesis also demonstrates a method to learn a metareasoning policy using data obtained through the robot's front camera and a multi-attribute variable function.

A.1 Appendix A

Table 1: Test set 1 without metareasoning

Global Planner	Local Planner	Iter Num	Test Num	End time	End location	Status	CPU Util	CPU temp	RAM	GPU util	GPU temp
EASL	MPPI	iter1	Test 1	1:11:20	27,36	Success	51.51	84.55	9.56	30.63	72.53
EASL	NLOPT	iter1	Test 1	1:01:40	28,38	Success	50.14	93.60	9.34	34.22	77.73
EASL	MPPI	iter2	Test 1	1:07:24	27,36	Success	52.83	78.31	9.66	28.01	70.48
EASL	NLOPT	iter2	Test 1	0:50:16	27,36	Success	53.52	85.80	9.44	31.45	71.20
EASL	NLOPT	iter3	Test 1	0:53:16	27,36	Success	54.03	82.47	9.45	35.16	73.18
EASL	MPPI	iter3	Test 1	1:06:19	27,36	Success	51.51	85.80	9.44	33.10	72.51
EASL	MPPI	iter4	Test 1	1:00:12	27,36	Success	54.03	84.36	9.95	44.15	74.33
EASL	NLOPT	iter4	Test 1	0:53:04	27,38	Success	55.09	81.89	9.44	33.10	71.29
EASL	MPPI	iter5	Test 1	1:02:36	27,36	Success	54.22	80.63	9.98	33.93	71.34
GLS	NLOPT	iter1	Test 1	0:48:04	27,36	Success	57.84	78.03	8.02	35.06	68.75
GLS	MPPI	iter1	Test 1	0:58:08	26,36	Success	57.90	80.73	8.69	30.56	69.95
GLS	NLOPT	iter2	Test 1	0:43:44	27,36	Success	57.69	78.59	8.03	37.69	68.75
GLS	MPPI	iter2	Test 1	0:49:04	27,36	Success	58.95	80.36	8.68	30.87	70.95
GLS	NLOPT	iter3	Test 1	0:48:20	27,36	Success	55.98	81.31	8.02	30.69	70.59
GLS	MPPI	iter3	Test 1	0:58:40	27,36	Success	59.64	80.47	8.76	31.36	70.66
GLS	NLOPT	iter4	Test 1	0:55:00	27,36	Success	58.28	79.58	8.16	29.80	69.66
GLS	MPPI	iter4	Test 1	0:56:40	26,36	Success	58.68	79.51	8.75	33.32	69.70
GLS	NLOPT	iter5	Test 1	0:48:28	27,36	Success	57.03	78.59	8.08	36.16	68.66
GLS	MPPI	iter5	Test 1	0:49:24	26,36	Success	57.67	79.91	8.61	31.16	70.02
SBPL	MPPI	iter1	Test 1	0:57:16	27,36	Success	48.12	98.90	9.41	37.38	77.68

SBPL	NLOPT	iter1	Test 1	0:58:20	28,36	Success	52.85	79.61	8.80	30.28	69.45
SBPL	MPPI	iter2	Test 1	0:54:04	26,36	Success	52.86	84.77	9.40	38.54	76.91
SBPL	NLOPT	iter2	Test 1	0:55:24	28,36	Success	53.24	78.05	8.85	30.23	69.02
SBPL	MPPI	iter3	Test 1	0:55:04	27,36	Success	52.41	80.53	9.25	33.63	72.76
SBPL	NLOPT	iter3	Test 1	0:52:40	27,36	Success	46.93	87.31	8.78	35.63	69.06
SBPL	MPPI	iter4	Test 1	0:49:00	27,36	Success	52.40	78.27	9.34	25.63	70.51
SBPL	NLOPT	iter4	Test 1	0:54:28	27,36	Success	55.29	78.13	8.80	36.08	69.02
SBPL	MPPI	iter5	Test 1	1:01:32	27,36	Success	50.66	86.71	9.36	37.22	73.77
SBPL	NLOPT	iter5	Test 1	0:54:48	27,36	Success	54.04	79.52	8.83	32.41	69.03
SLGP	NLOPT	iter1	Test 1	0:50:00	27,36	Success	52.46	76.00	7.98	35.00	69.03
SLGP	MPPI	iter1	Test 1	0:59:00	26,36	Success	45.68	89.28	8.61	34.13	73.31
SLGP	NLOPT	iter2	Test 1	0:43:20	27,36	Success	50.51	77.47	8.03	38.44	69.03
SLGP	MPPI	iter2	Test 1	1:01:32	27,36	Success	50.70	78.81	8.59	38.55	70.41
SLGP	NLOPT	iter3	Test 1	0:45:04	27,36	Success	52.80	75.05	8.14	38.45	66.94
SLGP	MPPI	iter3	Test 1	1:01:28	26,36	Success	51.59	76.42	8.60	29.14	69.69
SLGP	NLOPT	iter4	Test 1	0:55:20	27,36	Success	46.23	90.38	7.98	36.09	66.34
SLGP	MPPI	iter4	Test 1	0:51:56	26,36	Success	52.73	81.77	8.60	28.93	70.75
SLGP	NLOPT	iter5	Test 1	0:49:32	27,36	Success	50.27	82.63	8.07	40.47	73.16
SLGP	MPPI	iter5	Test 1	0:50:00	26,36	Success	54.19	79.08	8.70	33.28	69.84
EASL	MPPI	iter1	Test 2	1:26:20	26,37	Success	53.12	82.21	10.57	35.05	74.28
EASL	NLOPT	iter1	Test 2	0:36:48	5,7	Failed	44.58	81.75	8.55	31.50	68.13
EASL	MPPI	iter2	Test 2	1:19:32	28,37	Success	53.61	81.58	10.57	42.11	73.66
EASL	NLOPT	iter2	Test 2	0:40:32	5,6	Failed	45.89	81.50	8.44	33.50	68.38
EASL	MPPI	iter3	Test 2	1:16:48	28,37	Success	55.41	83.86	10.57	36.21	75.16
EASL	NLOPT	iter3	Test 2	0:37:00	5,6	Failed	45.63	83.38	8.50	28.63	67.38

EASL	MPPI	iter4	Test 2	1:33:16	26,37	Success	53.88	82.94	10.63	41.81	73.32
EASL	NLOPT	iter4	Test 2	0:38:44	5,7	Failed	46.38	76.13	8.50	29.88	68.25
EASL	MPPI	iter5	Test 2	1:32:20	27,37	Success	54.28	82.89	10.57	38.00	73.33
EASL	NLOPT	iter5	Test 2	0:36:04	5,7	Failed	46.64	79.13	8.34	41.00	68.25
GLS	NLOPT	iter1	Test 2	0:33:08	5,7	Failed	39.73	67.25	6.73	26.75	57.75
GLS	MPPI	iter1	Test 2	4:02:00	4,23	Timeout	52.21	88.87	11.79	45.21	74.00
GLS	NLOPT	iter2	Test 2	0:28:44	5,7	Failed	41.60	64.00	6.80	26.00	57.00
GLS	MPPI	iter2	Test 2	4:04:00	4,17	Timeout	59.89	85.73	11.39	36.03	73.11
GLS	NLOPT	iter3	Test 2	0:32:20	5,7	Failed	41.65	63.25	6.73	26.25	56.50
GLS	MPPI	iter3	Test 2	3:13:52	5,5	Stuck	59.08	85.06	11.00	33.99	73.03
GLS	NLOPT	iter4	Test 2	0:40:40	5,6	Failed	42.63	62.75	6.73	24.75	57.25
GLS	MPPI	iter4	Test 2	2:30:04	4,19	Stuck	59.45	83.18	10.58	35.11	73.50
GLS	NLOPT	iter5	Test 2	0:36:40	5,7	Failed	42.05	64.75	6.75	28.75	57.25
GLS	MPPI	iter5	Test 2	2:48:24	4,18	Stuck	50.63	90.61	10.95	36.52	74.78
SBPL	MPPI	iter1	Test 2	2:03:04	5,20	Stuck	49.92	84.45	11.81	38.19	72.31
SBPL	NLOPT	iter1	Test 2	0:33:16	6,3	Failed	43.00	64.25	7.20	27.25	54.75
SBPL	MPPI	iter2	Test 2	1:53:32	5,19	Stuck	54.01	79.01	11.76	33.40	71.04
SBPL	NLOPT	iter2	Test 2	0:36:44	7,3	Failed	42.83	67.50	7.15	29.00	57.25
SBPL	MPPI	iter3	Test 2	4:02:40	19,2	Timeout	54.17	83.12	13.36	35.02	71.63
SBPL	NLOPT	iter3	Test 2	0:32:20	5,3	Failed	41.70	65.25	7.18	29.50	55.50
SBPL	MPPI	iter4	Test 2	2:24:00	5,13	Stuck	55.47	82.33	12.57	34.82	71.50
SBPL	NLOPT	iter4	Test 2	0:29:20	4,3	Failed	42.18	66.50	7.18	26.75	55.75
SBPL	MPPI	iter5	Test 2	1:50:40	4,13	Stuck	50.72	85.82	11.98	33.70	73.03
SBPL	NLOPT	iter5	Test 2	0:31:36	6,3	Failed	42.70	67.25	7.20	25.00	56.25
SLGP	NLOPT	iter1	Test 2	0:38:44	6,11	Failed	37.28	70.25	6.78	24.50	57.00

SLGP	MPPI	iter1	Test 2	1:07:40	26,37	Success	53.26	81.69	9.64	40.23	73.01
SLGP	NLOPT	iter2	Test 2	0:34:56	5,8	Failed	39.95	63.75	6.80	30.50	57.25
SLGP	MPPI	iter2	Test 2	1:09:32	26,37	Success	53.37	83.92	9.73	32.67	73.46
SLGP	NLOPT	iter3	Test 2	0:32:56	5,8	Failed	40.38	63.00	6.80	25.25	56.50
SLGP	MPPI	iter3	Test 2	1:06:36	26,37	Success	53.34	81.04	9.79	28.90	73.48
SLGP	NLOPT	iter4	Test 2	0:38:28	6,12	Failed	38.53	65.00	6.75	26.75	57.00
SLGP	MPPI	iter4	Test 2	1:13:16	26,37	Success	54.95	82.56	9.71	31.51	73.24
SLGP	NLOPT	iter5	Test 2	0:39:16	6,10	Failed	38.53	63.50	6.78	25.50	57.00
SLGP	MPPI	iter5	Test 2	1:17:36	26,37	Success	53.10	81.29	9.78	35.26	73.09
EASL	MPPI	iter1	Test 3	1:15:16	27,37	Success	54.67	82.04	10.48	31.31	74.03
EASL	NLOPT	iter1	Test 3	0:29:24	1,2	Failed	37.28	72.50	7.30	36.75	59.00
EASL	MPPI	iter2	Test 3	1:24:52	27,36	Success	52.44	82.42	10.49	37.01	73.64
EASL	NLOPT	iter2	Test 3	0:28:44	1,2	Failed	38.98	68.50	7.28	25.75	58.50
EASL	MPPI	iter3	Test 3	1:21:00	27,36	Success	53.93	84.50	10.44	31.05	75.33
EASL	NLOPT	iter3	Test 3	0:32:40	1,2	Failed	38.18	72.00	7.28	24.00	59.25
EASL	MPPI	iter4	Test 3	1:19:20	27,36	Success	51.87	82.62	10.44	33.51	73.59
EASL	NLOPT	iter4	Test 3	0:35:12	1,2	Failed	38.63	68.50	7.28	30.75	58.50
EASL	MPPI	iter5	Test 3	1:28:28	27,38	Success	54.55	81.92	10.47	32.02	74.33
EASL	NLOPT	iter5	Test 3	0:28:20	1,3	Failed	39.53	68.00	7.33	25.00	58.50
GLS	NLOPT	iter1	Test 3	0:29:36	1,2	Failed	24.95	49.50	4.50	16.00	39.50
GLS	MPPI	iter1	Test 3	3:11:04	27,35	Success	56.44	88.63	10.77	31.27	74.83
GLS	NLOPT	iter2	Test 3	0:36:20	1,3	Failed	25.55	49.00	4.50	17.00	39.50
GLS	MPPI	iter2	Test 3	3:53:52	0,2	Stuck	56.17	91.68	11.57	37.06	75.44
GLS	NLOPT	iter3	Test 3	0:26:48	1,3	Failed	25.50	50.00	4.45	20.50	40.00
GLS	MPPI	iter3	Test 3	2:04:00	-1,3	Stuck	58.18	86.42	10.32	37.95	75.46

GLS	NLOPT	iter4	Test 3	0:35:40	1,3	Failed	25.60	50.00	4.45	19.00	39.00
GLS	MPPI	iter4	Test 3	1:49:04	0,3	Stuck	56.86	88.03	10.18	37.58	74.50
GLS	NLOPT	iter5	Test 3	0:25:00	1,3	Failed	25.20	49.00	4.50	18.50	39.50
GLS	MPPI	iter5	Test 3	2:16:08	13,4	Stuck	58.80	86.89	10.44	35.01	75.12
SBPL	MPPI	iter1	Test 3	3:17:00	27,36	Success	50.61	86.89	12.60	35.93	74.52
SBPL	NLOPT	iter1	Test 3	0:33:32	3,2	Failed	28.95	43.50	4.70	16.50	38.00
SBPL	MPPI	iter2	Test 3	4:08:00	12,2	Timeout	50.93	88.97	13.47	36.30	75.85
SBPL	NLOPT	iter2	Test 3	0:34:20	1,2	Failed	26.95	45.50	4.70	15.50	38.50
SBPL	MPPI	iter3	Test 3	1:55:00	27,36	Success	49.34	91.09	11.40	33.58	75.96
SBPL	NLOPT	iter3	Test 3	0:36:56	11,3	Failed	44.00	65.50	7.25	29.00	58.00
SBPL	MPPI	iter4	Test 3	2:07:00	27,36	Success	51.76	88.26	11.65	37.77	74.77
SBPL	NLOPT	iter4	Test 3	0:29:08	3,3	Failed	42.68	67.25	7.15	24.25	57.75
SBPL	MPPI	iter5	Test 3	1:52:40	13,2	Stuck	53.09	85.26	12.06	37.23	75.21
SBPL	NLOPT	iter5	Test 3	0:37:36	6,3	Failed	43.45	67.50	7.03	27.25	58.00
SLGP	NLOPT	iter1	Test 3	0:26:16	0,2	Failed	24.60	47.00	4.45	18.50	39.50
SLGP	MPPI	iter1	Test 3	1:08:48	27,36	Success	53.10	82.66	9.69	34.76	74.13
SLGP	NLOPT	iter2	Test 3	0:30:16	0,2	Failed	26.15	45.00	4.45	24.00	39.00
SLGP	MPPI	iter2	Test 3	1:10:36	27,36	Success	51.56	83.83	9.63	34.44	75.89
SLGP	NLOPT	iter3	Test 3	0:26:16	1,3	Failed	26.10	42.00	4.45	16.50	38.50
SLGP	MPPI	iter3	Test 3	1:25:48	27,36	Success	49.02	87.80	9.75	38.29	73.82
SLGP	NLOPT	iter4	Test 3	0:33:16	1,3	Failed	25.20	45.00	4.50	16.00	39.00
SLGP	MPPI	iter4	Test 3	1:22:00	27,36	Success	51.71	86.24	9.74	35.44	73.07
SLGP	NLOPT	iter5	Test 3	0:33:20	0,2	Failed	26.90	44.00	4.45	16.00	38.50
SLGP	MPPI	iter5	Test 3	1:19:16	27,36	Success	54.35	81.50	9.76	28.92	73.52
EASL	MPPI	iter1	Test 4	1:03:16	0,0	Stuck	50.56	82.36	10.12	33.09	74.04

EASL	NLOPT	iter1	Test 4	0:32:16	1,3	Failed	37.10	74.00	7.35	25.25	59.00
EASL	MPPI	iter2	Test 4	1:04:40	0,0	Stuck	49.88	84.07	10.15	33.66	75.63
EASL	NLOPT	iter2	Test 4	0:57:40	0,0	Stuck	49.74	84.39	9.67	29.18	75.63
EASL	MPPI	iter3	Test 4	1:26:52	26,37	Success	52.34	84.70	10.55	35.01	73.14
EASL	NLOPT	iter3	Test 4	0:31:48	1,2	Failed	38.90	68.25	7.23	25.50	57.75
EASL	MPPI	iter4	Test 4	1:30:16	26,37	Success	53.66	81.50	10.57	37.69	73.67
EASL	NLOPT	iter4	Test 4	0:30:16	1,3	Failed	38.48	69.00	7.28	30.00	58.00
EASL	MPPI	iter5	Test 4	1:28:48	26,37	Success	53.16	83.04	10.52	34.20	73.52
EASL	NLOPT	iter5	Test 4	0:27:20	1,2	Failed	39.40	70.00	7.30	26.50	57.75
GLS	NLOPT	iter1	Test 4	0:36:08	1,3	Failed	26.20	45.00	4.45	19.50	38.00
GLS	MPPI	iter1	Test 4	1:21:20	26,37	Success	59.37	85.35	9.74	34.67	76.21
GLS	NLOPT	iter2	Test 4	0:24:40	1,2	Failed	26.45	45.00	4.50	15.50	38.50
GLS	MPPI	iter2	Test 4	2:33:48	14,30	Stuck	56.46	86.87	10.75	35.29	74.25
GLS	NLOPT	iter3	Test 4	0:39:16	1,3	Failed	27.00	45.50	4.45	17.00	38.50
GLS	MPPI	iter3	Test 4	1:21:24	25,37	Success	57.99	84.52	9.74	33.99	74.04
GLS	NLOPT	iter4	Test 4	0:29:20	1,3	Failed	26.95	43.00	4.50	19.00	38.50
GLS	MPPI	iter4	Test 4	1:28:20	-3,1	Stuck	60.11	82.50	9.83	34.72	74.02
GLS	NLOPT	iter5	Test 4	0:33:20	1,3	Failed	26.70	44.50	4.50	18.00	38.50
GLS	MPPI	iter5	Test 4	1:46:24	26,37	Success	58.67	82.23	10.07	33.28	73.19
SBPL	MPPI	iter1	Test 4	1:14:04	26,37	Success	53.79	82.76	10.39	36.66	73.72
SBPL	NLOPT	iter1	Test 4	0:38:00	1,2	Failed	29.15	46.50	4.70	17.50	38.50
SBPL	MPPI	iter2	Test 4	1:10:36	26,37	Success	53.34	82.78	10.63	35.48	74.10
SBPL	NLOPT	iter2	Test 4	0:41:44	6,3	Failed	44.35	65.50	7.20	29.75	58.00
SBPL	MPPI	iter3	Test 4	1:11:40	26,37	Success	55.30	84.26	10.37	36.06	74.01
SBPL	NLOPT	iter3	Test 4	0:33:40	7,3	Failed	44.05	65.25	7.23	28.75	57.25

SBPL	MPPI	iter4	Test 4	1:07:00	26,37	Success	54.00	83.08	10.83	36.19	74.03
SBPL	NLOPT	iter4	Test 4	0:26:20	1,3	Failed	29.95	45.00	4.75	17.50	38.00
SBPL	MPPI	iter5	Test 4	1:05:16	26,37	Success	53.82	81.81	10.36	31.43	73.26
SBPL	NLOPT	iter5	Test 4	0:30:40	1,2	Failed	29.40	46.00	4.75	19.00	39.00
SLGP	NLOPT	iter1	Test 4	0:37:36	1,3	Failed	26.20	44.00	4.45	21.50	38.00
SLGP	MPPI	iter1	Test 4	1:15:40	26,37	Success	52.63	81.40	9.73	34.42	73.20
SLGP	NLOPT	iter2	Test 4	0:33:36	1,2	Failed	38.45	65.25	6.80	25.25	57.25
SLGP	MPPI	iter2	Test 4	1:15:20	26,37	Success	53.32	81.07	9.62	38.29	73.59
SLGP	NLOPT	iter3	Test 4	0:27:16	1,3	Failed	25.65	43.50	4.50	17.50	38.50
SLGP	MPPI	iter3	Test 4	1:11:20	26,37	Success	52.50	81.12	9.68	35.65	74.10
SLGP	NLOPT	iter4	Test 4	0:35:56	1,3	Failed	24.70	46.50	4.45	22.50	38.50
SLGP	MPPI	iter4	Test 4	1:14:40	26,37	Success	52.66	81.67	9.67	33.83	74.01
SLGP	NLOPT	iter5	Test 4	0:32:40	1,2	Failed	26.10	44.00	4.45	19.00	38.50
SLGP	MPPI	iter5	Test 4	1:08:12	0,0	Stuck	49.95	82.31	9.36	31.52	74.29
EASL	MPPI	iter1	Test 5	4:00:00	4,23	Timeout	60.52	85.25	12.65	36.24	75.49
EASL	NLOPT	iter1	Test 5	0:41:00	5,6	Failed	39.35	72.50	7.28	28.50	58.50
EASL	MPPI	iter2	Test 5	4:04:40	4,22	Timeout	61.78	85.50	12.66	44.41	74.04
EASL	NLOPT	iter2	Test 5	0:39:48	5,6	Failed	38.80	67.75	7.28	29.25	58.50
EASL	MPPI	iter3	Test 5	4:00:00	4,5	Timeout	56.47	84.86	12.37	39.75	75.46
EASL	NLOPT	iter3	Test 5	0:35:16	4,5	Failed	39.78	68.75	7.30	27.75	58.50
EASL	MPPI	iter4	Test 5	4:04:00	4,14	Timeout	58.51	84.68	12.35	36.97	74.61
EASL	NLOPT	iter4	Test 5	0:36:04	4,6	Failed	44.98	79.63	8.40	30.13	68.25
EASL	MPPI	iter5	Test 5	4:06:40	4,24	Timeout	57.32	86.85	12.26	37.80	75.67
EASL	NLOPT	iter5	Test 5	0:36:24	4,6	Failed	39.35	70.75	7.30	27.25	58.50
GLS	NLOPT	iter1	Test 5	0:32:20	5,7	Failed	28.00	44.50	4.45	16.00	38.00

GLS	MPPI	iter1	Test 5	4:06:40	4,18	Timeout	59.40	84.99	11.69	40.15	75.59
GLS	NLOPT	iter2	Test 5	0:30:20	5,7	Failed	43.73	66.75	6.78	22.50	57.25
GLS	MPPI	iter2	Test 5	3:48:36	5,10	Stuck	58.93	82.86	11.62	43.67	74.14
GLS	NLOPT	iter3	Test 5	0:30:00	5,7	Failed	41.25	64.25	6.73	23.50	57.25
GLS	MPPI	iter3	Test 5	2:29:40	4,16	Stuck	59.63	84.29	10.64	36.21	74.14
GLS	NLOPT	iter4	Test 5	0:33:56	5,7	Failed	42.20	66.00	6.75	27.50	57.75
GLS	MPPI	iter4	Test 5	4:06:16	4,18	Stuck	61.07	84.81	11.53	42.47	75.42
GLS	NLOPT	iter5	Test 5	0:41:36	5,7	Failed	43.75	65.00	6.75	26.25	57.75
GLS	MPPI	iter5	Test 5	4:13:20	4,22	Timeout	60.28	83.29	11.50	39.90	73.51
SBPL	MPPI	iter1	Test 5	4:06:40	24,25	Timeout	56.14	85.45	13.46	36.78	75.46
SBPL	NLOPT	iter1	Test 5	0:33:44	6,3	Failed	42.43	67.00	7.18	30.00	57.50
SBPL	MPPI	iter2	Test 5	2:43:44	4,6	Stuck	54.31	85.12	12.27	42.49	74.52
SBPL	NLOPT	iter2	Test 5	0:38:56	4,4	Failed	27.15	44.50	4.80	16.50	38.50
SBPL	MPPI	iter3	Test 5	4:02:40	23,12	Timeout	57.08	87.74	13.19	39.08	74.89
SBPL	NLOPT	iter3	Test 5	0:35:08	4,4	Failed	42.50	65.00	7.25	25.00	57.75
SBPL	MPPI	iter4	Test 5	1:17:40	4,9	Stuck	50.34	87.53	11.21	34.65	73.57
SBPL	NLOPT	iter4	Test 5	0:40:20	6,3	Failed	43.13	64.50	7.13	26.75	57.75
SBPL	MPPI	iter5	Test 5	4:00:00	18,2	Timeout	52.81	86.90	12.75	37.23	74.88
SBPL	NLOPT	iter5	Test 5	0:37:56	5,3	Failed	42.50	65.75	7.13	26.50	57.25
SLGP	NLOPT	iter1	Test 5	0:34:36	5,11	Failed	37.10	69.00	6.78	26.00	57.00
SLGP	MPPI	iter1	Test 5	1:39:44	15,36	Stuck	48.06	92.12	10.23	36.10	75.70
SLGP	NLOPT	iter2	Test 5	0:39:08	5,10	Failed	37.63	66.75	6.73	27.75	57.75
SLGP	MPPI	iter2	Test 5	2:09:32	15,35	Stuck	52.06	82.78	10.08	31.35	74.30
SLGP	NLOPT	iter3	Test 5	0:36:28	5,11	Failed	37.83	66.50	6.78	27.00	57.25
SLGP	MPPI	iter3	Test 5	1:41:32	15,36	Stuck	48.68	90.00	10.08	32.84	75.42

SLGP	NLOPT	iter4	Test 5	0:37:52	5,7	Failed	24.95	44.50	4.45	16.50	38.00
SLGP	MPPI	iter4	Test 5	2:12:44	15,35	Stuck	48.22	88.55	10.30	38.06	75.74
SLGP	NLOPT	iter5	Test 5	0:28:56	5,8	Failed	37.90	69.50	6.63	31.00	57.75
SLGP	MPPI	iter5	Test 5	2:07:44	15,34	Stuck	51.99	84.87	10.17	39.49	73.81
EASL	MPPI	iter1	Test 6	1:28:36	26,37	Success	53.19	80.74	10.68	47.29	73.78
EASL	NLOPT	iter1	Test 6	0:35:16	2,3	Failed	39.05	73.50	7.30	25.25	58.75
EASL	MPPI	iter2	Test 6	2:20:08	26,36	Success	48.94	87.53	11.20	30.34	74.85
EASL	NLOPT	iter2	Test 6	0:27:44	2,3	Failed	38.85	67.75	7.30	26.50	58.50
EASL	MPPI	iter3	Test 6	1:32:40	26,37	Success	52.70	84.17	10.72	33.54	73.33
EASL	NLOPT	iter3	Test 6	0:38:40	2,3	Failed	38.10	70.25	7.33	27.25	59.25
EASL	MPPI	iter4	Test 6	1:23:56	27,37	Success	54.58	81.35	10.67	36.58	74.34
EASL	NLOPT	iter4	Test 6	0:31:00	2,3	Failed	39.68	68.50	7.30	25.00	58.50
EASL	MPPI	iter5	Test 6	1:41:12	26,37	Success	49.77	84.20	10.75	32.44	73.83
EASL	NLOPT	iter5	Test 6	0:37:04	2,3	Failed	39.95	68.25	7.30	24.00	58.50
GLS	NLOPT	iter1	Test 6	0:26:40	2,4	Failed	29.30	45.00	4.50	17.00	38.50
GLS	MPPI	iter1	Test 6	1:01:56	0,0	Stuck	49.86	84.22	9.51	36.33	74.04
GLS	NLOPT	iter2	Test 6	0:30:32	2,5	Failed	28.70	44.50	4.55	19.00	39.00
GLS	MPPI	iter2	Test 6	1:22:12	2,3	Stuck	58.45	83.39	10.07	41.23	74.07
GLS	NLOPT	iter3	Test 6	0:34:52	8,13	Failed	50.30	75.13	7.98	29.63	66.88
GLS	MPPI	iter3	Test 6	1:50:28	25,37	Success	55.65	88.92	10.27	35.15	74.38
GLS	NLOPT	iter4	Test 6	0:29:12	2,4	Failed	28.75	45.00	4.55	15.50	38.50
GLS	MPPI	iter4	Test 6	1:34:40	27,36	Success	58.49	88.02	9.98	34.00	74.02
GLS	NLOPT	iter5	Test 6	0:38:16	2,4	Failed	29.05	45.00	4.55	15.50	38.50
GLS	MPPI	iter5	Test 6	1:35:28	26,37	Success	56.88	89.77	9.93	36.85	74.39
SBPL	MPPI	iter1	Test 6	1:25:00	26,37	Success	51.06	88.54	11.13	35.22	73.81
SBPL	NLOPT	iter1	Test 6	0:29:04	1,2	Failed	29.40	43.50	4.80	21.00	38.50

SBPL	MPPI	iter2	Test 6	1:39:28	27,36	Success	52.97	87.96	11.41	36.90	74.44
SBPL	NLOPT	iter2	Test 6	0:42:48	9,8	Failed	40.43	69.50	7.28	26.50	57.75
SBPL	MPPI	iter3	Test 6	1:14:28	26,36	Success	53.14	81.76	11.07	34.15	73.63
SBPL	NLOPT	iter3	Test 6	0:30:28	2,5	Failed	42.50	65.75	7.45	31.50	58.25
SBPL	MPPI	iter4	Test 6	1:28:36	26,36	Success	52.18	87.40	11.23	41.24	73.77
SBPL	NLOPT	iter4	Test 6	0:29:20	1,3	Failed	28.40	45.00	4.85	21.50	38.50
SBPL	MPPI	iter5	Test 6	1:25:00	26,37	Success	51.30	86.47	11.53	35.49	73.91
SBPL	NLOPT	iter5	Test 6	0:35:24	1,3	Failed	29.55	45.50	4.75	17.00	38.50
SLGP	NLOPT	iter1	Test 6	0:38:28	2,4	Failed	25.45	46.50	4.50	16.50	38.00
SLGP	MPPI	iter1	Test 6	1:05:32	0,0	Stuck	49.91	82.57	9.51	36.60	73.88
SLGP	NLOPT	iter2	Test 6	0:42:40	8,13	Failed	37.15	68.00	6.75	26.75	57.75
SLGP	MPPI	iter2	Test 6	1:51:04	25,37	Success	48.92	89.19	10.07	37.27	75.84
SLGP	NLOPT	iter3	Test 6	0:29:12	2,5	Failed	24.90	45.50	4.55	18.00	38.50
SLGP	MPPI	iter3	Test 6	1:17:52	26,37	Success	48.17	88.41	9.96	35.88	73.82
SLGP	NLOPT	iter4	Test 6	0:36:12	2,5	Failed	25.35	48.00	4.50	17.50	38.50
SLGP	MPPI	iter4	Test 6	1:40:40	26,38	Success	50.32	91.59	9.82	34.24	75.69
SLGP	NLOPT	iter5	Test 6	0:39:48	2,5	Failed	24.90	45.00	4.50	18.00	38.50
SLGP	MPPI	iter5	Test 6	1:53:56	27,36	Success	51.44	85.38	10.25	35.14	75.48
EASL	MPPI	iter1	Test 7	1:51:04	27,36	Success	47.68	82.62	10.27	28.95	69.29
EASL	NLOPT	iter1	Test 7	0:54:20	13,14	Failed	49.70	85.44	9.04	33.94	70.91
EASL	MPPI	iter2	Test 7	4:03:20	28,34	Timeout	57.85	83.34	11.79	37.29	70.01
EASL	NLOPT	iter2	Test 7	1:20:44	13,17	Failed	55.62	79.20	9.49	30.60	70.48
EASL	MPPI	iter3	Test 7	1:44:20	28,37	Success	63.99	79.75	10.18	26.97	68.15
EASL	NLOPT	iter3	Test 7	1:13:20	14,19	Failed	59.04	84.16	9.27	25.13	71.45
EASL	MPPI	iter4	Test 7	1:30:40	27,36	Success	47.39	85.53	10.67	31.21	68.93

EASL	NLOPT	iter4	Test 7	1:46:12	27,37	Success	51.97	82.66	9.73	30.99	68.17
EASL	MPPI	iter5	Test 7	1:49:04	27,36	Success	48.82	83.14	10.16	25.81	68.56
EASL	NLOPT	iter5	Test 7	1:01:40	13,19	Failed	54.39	85.36	9.10	35.31	73.02
GLS	NLOPT	iter1	Test 7	1:01:08	15,16	Failed	56.79	86.64	8.43	33.31	70.64
GLS	MPPI	iter1	Test 7	1:36:24	27,36	Success	56.10	85.00	9.43	31.39	69.17
GLS	NLOPT	iter2	Test 7	1:12:44	27,36	Success	55.42	84.36	8.71	30.03	70.32
GLS	MPPI	iter2	Test 7	2:03:32	14,30	Stuck	56.00	86.67	9.85	27.65	69.97
GLS	NLOPT	iter3	Test 7	1:45:04	26,36	Success	55.77	84.55	9.16	35.09	71.00
GLS	MPPI	iter3	Test 7	1:15:48	27,36	Success	54.96	84.73	9.04	30.45	70.75
GLS	NLOPT	iter4	Test 7	1:42:40	27,36	Success	57.03	84.19	8.93	31.34	70.17
GLS	MPPI	iter4	Test 7	1:26:40	27,36	Success	54.82	86.36	9.11	30.99	70.80
GLS	NLOPT	iter5	Test 7	0:59:40	14,14	Failed	53.02	77.75	8.02	35.13	65.63
GLS	MPPI	iter5	Test 7	1:24:16	27,36	Success	54.49	86.60	9.07	30.67	70.86
SBPL	MPPI	iter1	Test 7	1:27:28	27,36	Success	45.52	99.33	9.84	27.18	78.43
SBPL	NLOPT	iter1	Test 7	1:16:52	28,36	Success	49.84	83.43	9.91	33.21	70.13
SBPL	MPPI	iter2	Test 7	1:28:08	27,36	Success	45.49	99.73	9.94	32.20	76.80
SBPL	NLOPT	iter2	Test 7	1:12:44	27,36	Success	48.83	85.00	9.30	30.74	70.72
SBPL	MPPI	iter3	Test 7	1:24:16	27,36	Success	48.85	87.85	10.04	25.99	70.87
SBPL	NLOPT	iter3	Test 7	1:17:28	27,36	Success	51.14	85.11	9.14	31.98	70.44
SBPL	MPPI	iter4	Test 7	1:27:16	27,36	Success	48.37	84.17	9.86	33.44	70.86
SBPL	NLOPT	iter4	Test 7	1:02:32	14,15	Failed	50.33	79.75	8.88	39.69	67.13
SBPL	MPPI	iter5	Test 7	1:28:00	27,36	Success	49.49	84.62	9.70	30.27	70.64
SBPL	NLOPT	iter5	Test 7	1:08:28	16,13	Failed	53.08	81.53	8.98	32.81	68.75
SLGP	NLOPT	iter1	Test 7	0:58:00	15,21	Failed	46.47	81.44	8.03	30.69	65.63
SLGP	MPPI	iter1	Test 7	2:06:16	14,34	Stuck	48.70	84.36	9.86	31.74	71.00

SLGP	NLOPT	iter2	Test 7	1:02:40	15,25	Failed	46.76	81.38	8.38	29.94	68.56
SLGP	MPPI	iter2	Test 7	1:55:24	15,33	Stuck	47.04	85.86	9.62	33.21	70.03
SLGP	NLOPT	iter3	Test 7	0:59:44	15,23	Failed	46.92	78.97	8.41	32.44	68.78
SLGP	MPPI	iter3	Test 7	1:56:24	14,36	Stuck	47.46	85.98	9.67	30.42	71.00
SLGP	NLOPT	iter4	Test 7	0:58:12	15,21	Failed	45.61	79.94	8.08	34.50	66.88
SLGP	MPPI	iter4	Test 7	2:48:04	14,35	Stuck	47.34	85.46	9.98	29.08	70.87
SLGP	NLOPT	iter5	Test 7	1:07:56	15,23	Failed	47.38	85.84	8.27	26.94	69.28
SLGP	MPPI	iter5	Test 7	4:05:20	14,33	Timeout	48.08	87.13	11.32	30.25	70.16
EASL	MPPI	iter1	Test 8	1:14:08	0,0	Stuck	50.53	80.50	9.53	26.29	70.49
EASL	NLOPT	iter1	Test 8	1:37:44	28,37	Success	51.66	80.04	9.58	29.53	68.42
EASL	MPPI	iter2	Test 8	1:38:04	28,37	Success	49.64	83.36	10.08	32.28	69.46
EASL	NLOPT	iter2	Test 8	1:33:20	28,37	Success	51.20	82.80	9.63	30.46	68.34
EASL	MPPI	iter3	Test 8	1:37:08	26,37	Success	51.56	83.96	9.98	34.66	69.09
EASL	NLOPT	iter3	Test 8	1:37:00	28,37	Success	53.28	76.62	9.59	33.00	68.72
EASL	MPPI	iter4	Test 8	1:36:56	26,37	Success	50.58	85.08	10.06	33.37	68.80
EASL	NLOPT	iter4	Test 8	1:30:28	26,36	Success	53.29	80.64	9.59	31.03	68.18
EASL	MPPI	iter5	Test 8	1:36:00	26,37	Success	50.25	83.20	9.99	29.59	69.53
EASL	NLOPT	iter5	Test 8	1:37:08	26,37	Success	50.39	81.32	9.64	29.55	69.09
GLS	NLOPT	iter1	Test 8	1:25:12	6,31	Failed	54.02	87.20	8.83	26.63	70.79
GLS	MPPI	iter1	Test 8	4:04:00	23,11	Timeout	61.13	82.34	11.92	34.23	70.72
GLS	NLOPT	iter2	Test 8	1:35:44	27,36	Success	56.90	86.95	8.88	38.01	71.04
GLS	MPPI	iter2	Test 8	1:51:28	18,20	Stuck	61.06	79.27	9.71	35.49	69.20
GLS	NLOPT	iter3	Test 8	1:30:56	26,36	Success	56.26	85.25	8.98	35.14	70.96
GLS	MPPI	iter3	Test 8	2:02:48	21,11	Stuck	57.39	82.29	9.95	39.79	71.00
GLS	NLOPT	iter4	Test 8	1:26:12	26,38	Success	57.04	82.78	8.87	31.19	72.15

GLS	MPPI	iter4	Test 8	2:10:36	4,32	Stuck	57.50	82.58	9.89	32.95	70.38
GLS	NLOPT	iter5	Test 8	1:35:12	26,37	Success	56.05	86.56	8.88	33.11	71.01
GLS	MPPI	iter5	Test 8	3:05:52	26,37	Success	55.08	84.64	10.56	34.02	70.50
SBPL	MPPI	iter1	Test 8	1:33:20	27,36	Success	48.87	85.56	10.17	36.83	70.98
SBPL	NLOPT	iter1	Test 8	1:21:48	27,36	Success	49.34	83.62	9.82	27.31	70.96
SBPL	MPPI	iter2	Test 8	1:28:40	26,37	Success	48.33	85.72	9.99	35.97	70.98
SBPL	NLOPT	iter2	Test 8	1:25:04	27,36	Success	48.82	86.92	9.69	36.83	70.97
SBPL	MPPI	iter3	Test 8	1:27:04	26,37	Success	48.21	85.36	10.01	33.17	70.96
SBPL	NLOPT	iter3	Test 8	1:24:44	27,36	Failed	48.47	85.20	9.57	33.55	70.96
SBPL	MPPI	iter4	Test 8	1:27:24	26,37	Success	49.20	86.88	10.22	34.05	70.98
SBPL	NLOPT	iter4	Test 8	1:19:40	0,0	Stuck	46.46	88.11	8.35	33.39	70.38
SBPL	MPPI	iter5	Test 8	1:28:40	26,36	Success	49.05	84.12	10.06	37.37	70.46
SBPL	NLOPT	iter5	Test 8	1:29:44	27,36	Success	49.71	86.12	9.80	34.93	70.96
SLGP	NLOPT	iter1	Test 8	1:06:32	20,28	Failed	47.01	81.59	8.27	33.69	67.81
SLGP	MPPI	iter1	Test 8	1:59:24	25,29	Stuck	47.38	86.05	9.68	37.45	71.00
SLGP	NLOPT	iter2	Test 8	1:02:24	20,28	Failed	48.04	82.88	8.28	29.41	68.69
SLGP	MPPI	iter2	Test 8	2:19:48	25,29	Stuck	49.13	85.57	9.97	33.04	70.52
SLGP	NLOPT	iter3	Test 8	1:06:16	20,28	Failed	48.24	81.69	8.38	39.47	68.78
SLGP	MPPI	iter3	Test 8	2:57:40	28,28	Stuck	49.81	79.44	10.37	36.62	70.21
SLGP	NLOPT	iter4	Test 8	1:28:20	0,0	Stuck	46.32	87.56	8.34	27.48	70.86
SLGP	MPPI	iter4	Test 8	1:48:56	24,28	Stuck	47.75	85.87	9.55	36.54	70.99
SLGP	NLOPT	iter5	Test 8	1:09:24	20,28	Failed	49.17	83.44	8.44	36.94	69.38
SLGP	MPPI	iter5	Test 8	2:40:20	26,28	Failed	47.56	85.51	10.16	35.25	70.60

Table 2: Test set 1 with Policy 1

EASL	MPPI	iter1	Test 1	1:12:16	27,36	Success	51.562	89.859	10.117	34.145	80.117
EASL	NLOPT	iter1	Test 1	1:08:00	27,36	Success	59.139	86.949	9.581	36.844	76.371
EASL	MPPI	iter2	Test 1	1:14:44	27,37	Success	54.719	81.985	9.980	36.708	73.461
EASL	NLOPT	iter2	Test 1	1:10:12	27,36	Success	59.775	86.453	9.544	34.203	76.020
EASL	MPPI	iter3	Test 1	1:17:20	27,36	Success	51.525	87.797	9.967	37.094	78.557
EASL	NLOPT	iter3	Test 1	1:07:16	27,38	Success	59.979	83.824	9.531	35.602	74.828
EASL	MPPI	iter4	Test 1	1:17:36	27,37	Success	54.087	85.492	9.960	35.580	76.211
EASL	NLOPT	iter4	Test 1	1:10:08	27,36	Success	60.326	86.031	9.495	40.445	76.805
EASL	MPPI	iter5	Test 1	1:17:56	27,36	Success	59.124	83.795	10.075	37.790	74.229
EASL	NLOPT	iter5	Test 1	1:09:08	27,36	Success	60.782	84.156	9.538	36.211	75.727
GLS	NLOPT	iter1	Test 1	0:58:36	27,36	Success	57.422	83.563	8.513	41.156	73.844
GLS	MPPI	iter1	Test 1	1:06:24	27,36	Success	57.656	84.523	9.091	32.000	74.961
GLS	NLOPT	iter2	Test 1	0:54:00	27,36	Success	57.156	81.281	8.466	32.406	72.906
GLS	MPPI	iter2	Test 1	1:09:32	27,36	Success	58.303	86.004	9.200	34.949	74.855
GLS	NLOPT	iter3	Test 1	0:55:40	27,36	Success	57.997	83.969	8.453	36.063	73.719
GLS	MPPI	iter3	Test 1	1:02:56	26,36	Success	59.262	85.477	9.167	32.875	75.461
GLS	NLOPT	iter4	Test 1	0:57:40	27,36	Success	57.934	82.531	8.416	33.969	73.156
GLS	MPPI	iter4	Test 1	1:12:24	27,36	Success	57.847	85.188	9.194	33.227	74.711
GLS	NLOPT	iter5	Test 1	0:51:36	27,36	Success	58.269	84.656	8.491	35.719	73.719
GLS	MPPI	iter5	Test 1	1:10:16	26,36	Success	59.102	85.273	9.169	34.797	74.703
SBPL	MPPI	iter1	Test 1	1:00:00	26,36	Success	46.725	98.125	9.883	32.484	79.672
SBPL	NLOPT	iter1	Test 1	1:06:12	27,36	Success	51.834	81.219	8.988	34.469	73.844
SBPL	MPPI	iter2	Test 1	1:11:52	27,36	Success	50.673	87.977	9.655	34.594	73.797
SBPL	NLOPT	iter2	Test 1	1:08:00	27,36	Success	57.661	83.922	8.977	32.719	73.984

SBPL	MPPI	iter3	Test 1	1:13:24	27,36	Success	52.231	83.348	9.904	34.203	74.031
SBPL	NLOPT	iter3	Test 1	0:59:12	28,36	Success	56.815	85.383	9.162	31.047	74.586
SBPL	MPPI	iter4	Test 1	1:01:32	27,36	Success	53.541	82.914	9.717	52.945	74.555
SBPL	NLOPT	iter4	Test 1	0:57:36	28,36	Success	52.247	82.219	9.078	34.813	73.313
SBPL	MPPI	iter5	Test 1	1:01:00	27,36	Success	53.858	83.398	9.787	35.570	74.711
SBPL	NLOPT	iter5	Test 1	1:00:00	27,36	Success	54.714	85.813	8.970	40.492	74.055
SLGP	NLOPT	iter1	Test 1	0:54:00	27,36	Success	48.631	81.188	8.156	31.188	70.938
SLGP	MPPI	iter1	Test 1	1:11:40	27,36	Success	51.670	82.914	9.222	32.758	74.047
SLGP	MPPI	iter2	Test 1	1:01:52	27,36	Success	52.015	84.992	9.197	30.844	77.219
SLGP	NLOPT	iter2	Test 1	0:51:32	27,36	Success	48.788	78.750	8.113	29.625	70.938
SLGP	MPPI	iter3	Test 1	1:08:20	27,36	Success	53.436	84.461	9.152	36.391	73.484
SLGP	NLOPT	iter3	Test 1	0:59:48	27,36	Success	48.644	80.688	8.131	34.563	70.938
SLGP	MPPI	iter4	Test 1	1:11:48	26,36	Success	51.870	85.180	9.195	35.172	74.086
SLGP	NLOPT	iter4	Test 1	0:54:40	27,36	Success	49.200	79.688	8.188	34.063	70.938
SLGP	MPPI	iter5	Test 1	1:01:24	26,36	Success	51.837	85.117	9.250	32.758	74.055
SLGP	NLOPT	iter5	Test 1	0:50:40	27,36	Success	51.728	82.781	8.444	35.094	72.906
EASL	MPPI	iter1	Test 2	1:35:32	26,37	Success	68.760	86.068	10.461	38.460	75.031
EASL	NLOPT	iter1	Test 2	0:52:12	5,6	Failed	82.591	87.594	9.425	32.563	73.406
EASL	MPPI	iter2	Test 2	1:34:28	26,37	Success	64.105	85.699	10.221	39.113	74.514
EASL	NLOPT	iter2	Test 2	0:46:20	5,6	Failed	85.709	85.063	9.306	30.781	73.625
EASL	MPPI	iter3	Test 2	1:49:36	26,37	Success	70.642	86.814	10.411	41.370	75.031
EASL	NLOPT	iter3	Test 2	0:52:40	5,6	Failed	88.241	86.156	9.497	30.406	73.531
EASL	MPPI	iter4	Test 2	1:31:16	26,37	Success	74.980	85.638	10.289	35.811	75.500
EASL	NLOPT	iter4	Test 2	0:45:40	5,7	Failed	90.756	86.219	9.397	43.875	73.625
EASL	MPPI	iter5	Test 2	1:35:44	27,37	Success	80.425	88.612	10.476	39.646	75.500

EASL	NLOPT	iter5	Test 2	0:48:00	5,6	Failed	92.500	86.219	9.538	30.313	73.594
GLS	NLOPT	iter1	Test 2	0:56:36	5,7	Failed	51.488	78.875	7.913	30.375	67.000
GLS	MPPI	iter1	Test 2	1:37:28	4,18	Failed	58.936	83.340	10.059	40.382	74.997
GLS	NLOPT	iter2	Test 2	0:49:28	5,10	Failed	55.144	79.500	8.563	30.688	71.438
GLS	MPPI	iter2	Test 2	1:32:08	5,24	Failed	62.297	83.628	10.073	38.012	74.386
GLS	NLOPT	iter3	Test 2	0:44:00	5,7	Failed	53.400	77.375	7.925	28.875	66.875
GLS	MPPI	iter3	Test 2	1:33:24	5,20	Failed	59.154	84.040	10.006	40.867	75.005
GLS	NLOPT	iter4	Test 2	0:49:20	5,7	Failed	49.625	74.625	7.913	29.500	66.625
GLS	MPPI	iter4	Test 2	1:36:48	4,27	Failed	61.913	84.341	9.952	34.449	74.506
GLS	NLOPT	iter5	Test 2	0:55:28	5,7	Failed	49.738	75.125	7.925	32.000	66.875
GLS	MPPI	iter5	Test 2	1:29:44	5,23	Failed	61.823	84.256	9.891	34.638	75.024
SBPL	MPPI	iter1	Test 2	1:23:00	4,15	Failed	61.797	85.762	10.625	37.757	74.224
SBPL	NLOPT	iter1	Test 2	0:50:32	5,7	Failed	47.588	77.625	8.325	31.750	67.375
SBPL	MPPI	iter2	Test 2	1:29:40	5,15	Failed	59.260	85.027	11.019	44.727	74.247
SBPL	NLOPT	iter2	Test 2	0:46:36	5,6	Failed	53.525	73.875	8.463	28.500	66.875
SBPL	MPPI	iter3	Test 2	1:38:00	5,17	Failed	60.859	83.384	10.953	35.716	74.031
SBPL	NLOPT	iter3	Test 2	0:44:08	5,6	Failed	51.450	76.625	8.250	29.625	66.875
SBPL	MPPI	iter4	Test 2	1:40:16	4,20	Failed	58.032	84.161	11.235	34.311	74.068
SBPL	NLOPT	iter4	Test 2	0:51:40	6,4	Failed	57.781	79.438	9.350	33.750	71.438
SBPL	MPPI	iter5	Test 2	1:28:08	4,17	Failed	59.957	83.833	10.798	36.933	74.531
SBPL	NLOPT	iter5	Test 2	0:45:48	5,6	Failed	53.663	74.625	8.400	32.375	66.875
SLGP	NLOPT	iter1	Test 2	0:43:20	5,8	Failed	46.388	80.625	7.763	33.250	67.375
SLGP	MPPI	iter1	Test 2	1:24:36	26,37	Success	53.197	84.316	9.741	35.541	73.537
SLGP	MPPI	iter2	Test 2	1:21:08	26,37	Success	52.813	83.493	9.777	38.090	74.253
SLGP	NLOPT	iter2	Test 2	0:52:56	5,10	Failed	51.500	78.750	8.581	33.063	70.813

SLGP	MPPI	iter3	Test 2	1:18:16	26,37	Success	52.410	82.236	9.814	34.691	74.063
SLGP	NLOPT	iter3	Test 2	0:47:56	5,8	Failed	46.950	77.750	7.925	30.875	66.875
SLGP	MPPI	iter4	Test 2	1:22:20	26,37	Success	53.611	81.854	9.792	38.828	74.044
SLGP	NLOPT	iter4	Test 2	0:59:08	5,17	Failed	57.253	81.953	9.077	33.750	74.109
SLGP	MPPI	iter5	Test 2	1:23:56	26,37	Success	52.901	82.290	9.817	33.773	74.259
SLGP	NLOPT	iter5	Test 2	0:57:32	5,13	Failed	49.175	82.750	8.631	31.813	71.688
EASL	MPPI	iter1	Test 3	1:19:32	27,36	Success	97.295	89.700	10.637	29.374	77.000
EASL	NLOPT	iter1	Test 3	0:46:00	1,3	Failed	96.088	87.000	9.409	39.438	73.531
EASL	MPPI	iter2	Test 3	1:20:56	27,36	Success	97.343	89.663	10.591	24.911	76.945
EASL	NLOPT	iter2	Test 3	0:50:32	2,3	Failed	98.194	88.875	10.011	27.750	75.313
EASL	MPPI	iter3	Test 3	1:24:52	27,36	Success	97.464	90.014	10.580	28.374	76.953
EASL	NLOPT	iter3	Test 3	0:57:40	2,3	Failed	96.272	87.219	9.828	29.844	73.625
EASL	MPPI	iter4	Test 3	1:15:00	27,36	Success	98.671	90.870	10.687	27.654	76.954
EASL	NLOPT	iter4	Test 3	0:55:36	1,3	Failed	96.322	88.156	9.659	44.063	74.344
EASL	MPPI	iter5	Test 3	1:25:44	27,36	Success	98.430	90.477	10.436	26.862	76.734
EASL	NLOPT	iter5	Test 3	1:39:00	29,37	Success	99.570	92.687	11.198	25.369	77.932
GLS	NLOPT	iter1	Test 3	0:45:48	1,3	Failed	48.475	77.500	10.113	24.875	67.000
GLS	MPPI	iter1	Test 3	1:25:24	11,2	Failed	64.631	85.219	12.184	35.140	75.115
GLS	NLOPT	iter2	Test 3	0:45:08	1,3	Failed	43.225	65.000	8.625	27.750	58.000
GLS	MPPI	iter2	Test 3	1:26:04	13,1	Failed	60.515	83.787	12.248	35.983	74.985
GLS	NLOPT	iter3	Test 3	1:01:16	5,3	Failed	57.988	79.938	10.850	34.250	71.438
GLS	MPPI	iter3	Test 3	1:24:36	13,2	Failed	62.916	86.150	12.433	38.762	75.000
GLS	NLOPT	iter4	Test 3	0:46:04	1,3	Failed	42.475	65.250	8.600	27.750	57.750
GLS	MPPI	iter4	Test 3	1:26:20	14,2	Failed	63.454	84.588	12.279	38.059	75.114
GLS	NLOPT	iter5	Test 3	0:50:52	1,3	Failed	45.125	65.500	8.600	33.250	57.750
GLS	MPPI	iter5	Test 3	1:10:44	6,3	Failed	61.937	85.465	12.215	38.910	75.375

SBPL	MPPI	iter1	Test 3	1:40:20	27,36	Success	45.551	97.830	13.939	29.675	75.500
SBPL	NLOPT	iter1	Test 3	0:44:04	3,3	Failed	51.300	78.125	10.588	29.250	67.625
SBPL	MPPI	iter2	Test 3	1:28:40	27,36	Success	53.762	82.599	13.242	24.833	73.315
SBPL	NLOPT	iter2	Test 3	0:46:28	2,3	Failed	47.150	66.750	9.150	29.000	57.000
SBPL	MPPI	iter3	Test 3	1:22:00	15,2	Failed	62.637	83.073	13.127	33.753	74.402
SBPL	NLOPT	iter3	Test 3	0:43:36	3,3	Failed	52.125	74.375	10.588	28.375	66.625
SBPL	MPPI	iter4	Test 3	1:29:00	22,4	Failed	60.361	85.248	13.312	32.147	74.257
SBPL	NLOPT	iter4	Test 3	0:45:44	10,3	Failed	51.200	74.625	10.788	26.500	66.875
SBPL	MPPI	iter5	Test 3	1:47:36	27,36	Success	50.353	89.843	13.178	34.649	73.814
SBPL	NLOPT	iter5	Test 3	0:42:24	2,3	Failed	53.113	77.750	10.575	25.250	66.500
SLGP	NLOPT	iter1	Test 3	0:51:56	1,3	Failed	39.700	69.750	8.625	30.750	57.500
SLGP	MPPI	iter1	Test 3	1:49:36	27,36	Success	49.863	88.561	12.438	32.877	73.782
SLGP	NLOPT	iter2	Test 3	0:45:00	1,3	Failed	42.450	64.250	8.600	25.750	58.000
SLGP	MPPI	iter2	Test 3	1:52:40	27,36	Success	53.316	81.477	12.389	33.265	73.133
SLGP	MPPI	iter3	Test 3	1:20:44	27,36	Success	54.535	82.465	12.216	30.877	74.044
SLGP	NLOPT	iter3	Test 3	0:41:36	2,3	Failed	43.675	64.750	8.675	26.750	57.250
SLGP	MPPI	iter4	Test 3	1:28:16	27,36	Success	52.175	84.118	12.125	34.562	74.097
SLGP	NLOPT	iter4	Test 3	0:46:20	1,3	Failed	44.825	66.000	8.625	29.750	57.750
SLGP	MPPI	iter5	Test 3	1:44:20	28,36	Success	53.317	82.334	12.396	33.861	73.534
SLGP	NLOPT	iter5	Test 3	0:52:52	2,3	Failed	45.375	66.250	8.625	28.500	57.750
EASL	MPPI	iter1	Test 4	1:47:32	26,37	Success	99.772	91.266	10.939	30.577	77.000
EASL	NLOPT	iter1	Test 4	1:12:52	2,3	Failed	99.486	90.988	11.343	25.055	76.699
EASL	MPPI	iter2	Test 4	1:47:32	26,37	Success	99.772	91.266	10.939	30.577	77.000
EASL	NLOPT	iter2	Test 4	0:38:52	1,3	Failed	78.138	73.375	8.613	24.375	62.250
EASL	MPPI	iter3	Test 4	1:44:40	26,37	Success	64.987	79.344	10.593	34.243	68.563

EASL	NLOPT	iter3	Test 4	1:09:56	2,3	Failed	99.404	91.770	11.460	28.871	76.699
EASL	MPPI	iter4	Test 4	1:01:48	26,37	Success	99.591	92.047	11.688	46.920	77.312
EASL	NLOPT	iter4	Test 4	1:09:56	2,3	Failed	99.404	91.770	11.460	28.871	76.699
EASL	MPPI	iter5	Test 4	1:46:24	25,37	Success	59.000	78.042	10.803	27.992	68.117
EASL	NLOPT	iter5	Test 4	1:12:52	2,3	Failed	99.486	90.988	11.343	25.055	76.699
GLS	NLOPT	iter1	Test 4	1:28:16	6,2	Stuck	55.525	81.774	13.721	39.456	74.249
GLS	MPPI	iter1	Test 4	1:23:40	26,37	Success	59.810	83.800	14.276	34.248	74.999
GLS	NLOPT	iter2	Test 4	1:43:28	-1,0	Stuck	53.225	88.762	13.796	37.800	74.329
GLS	MPPI	iter2	Test 4	1:17:56	26,37	Success	57.877	83.495	14.282	44.949	75.001
GLS	NLOPT	iter3	Test 4	0:55:40	2,3	Failed	55.213	77.250	11.913	29.000	68.625
GLS	MPPI	iter3	Test 4	1:40:12	12,10	Failed	64.382	86.795	14.592	34.996	75.016
GLS	NLOPT	iter4	Test 4	0:48:04	1,3	Failed	40.325	75.000	10.200	23.250	59.750
GLS	MPPI	iter4	Test 4	1:29:44	15,10	Failed	60.822	86.674	14.351	46.660	75.954
GLS	NLOPT	iter5	Test 4	1:40:04	-1,0	Stuck	50.421	88.050	13.775	36.893	75.140
GLS	MPPI	iter5	Test 4	1:47:20	15,14	Failed	62.707	85.320	14.822	37.260	75.000
SBPL	MPPI	iter1	Test 4	1:24:56	26,37	Success	54.493	81.587	15.120	35.037	73.996
SBPL	NLOPT	iter1	Test 4	0:53:08	6,3	Failed	60.831	83.125	13.819	31.875	71.313
SBPL	MPPI	iter2	Test 4	1:31:04	26,36	Success	54.048	84.915	15.380	39.663	74.022
SBPL	NLOPT	iter2	Test 4	0:48:44	5,3	Failed	57.781	79.000	13.238	35.938	71.438
SBPL	MPPI	iter3	Test 4	1:29:40	26,37	Success	55.720	86.631	15.733	34.284	75.255
SBPL	NLOPT	iter3	Test 4	1:15:48	1,2	Stuck	55.458	82.314	13.824	36.631	73.994
SBPL	MPPI	iter4	Test 4	1:24:00	26,36	Success	54.217	83.878	15.353	34.737	74.069
SBPL	NLOPT	iter4	Test 4	0:47:08	4,3	Failed	53.638	78.125	12.738	33.250	68.250
SBPL	MPPI	iter5	Test 4	1:18:48	24,38	Success	54.993	83.071	15.163	41.302	75.006
SBPL	NLOPT	iter5	Test 4	0:49:28	3,3	Failed	54.075	73.875	12.350	33.750	65.625

SLGP	NLOPT	iter1	Test 4	0:50:12	2,3	Failed	41.575	64.500	10.200	28.000	57.750
SLGP	MPPI	iter1	Test 4	1:29:24	27,36	Success	54.255	82.791	14.164	37.509	74.100
SLGP	NLOPT	iter2	Test 4	0:39:08	1,3	Failed	44.175	62.750	10.100	25.500	56.500
SLGP	MPPI	iter2	Test 4	1:29:20	26,37	Success	54.365	85.192	14.327	33.250	76.199
SLGP	MPPI	iter3	Test 4	1:30:08	26,37	Success	53.449	83.971	14.257	37.053	74.116
SLGP	NLOPT	iter3	Test 4	0:44:08	1,3	Failed	45.500	64.000	10.200	28.750	57.250
SLGP	MPPI	iter4	Test 4	1:24:20	26,37	Success	52.876	82.383	14.379	36.309	74.064
SLGP	NLOPT	iter4	Test 4	0:45:36	1,3	Failed	46.300	63.500	10.125	26.250	57.250
SLGP	MPPI	iter5	Test 4	1:30:24	26,37	Success	53.570	83.179	14.230	34.436	74.233
SLGP	NLOPT	iter5	Test 4	0:42:24	1,3	Failed	42.800	62.250	10.200	24.750	57.250
EASL	MPPI	iter1	Test 5	1:50:56	4,16	Failed	97.029	83.472	10.987	36.448	70.500
EASL	NLOPT	iter1	Test 5	1:02:04	5,6	Failed	99.263	82.992	10.286	22.586	69.234
EASL	MPPI	iter2	Test 5	2:27:20	4,21	Failed	72.669	79.075	11.635	29.849	68.770
EASL	NLOPT	iter2	Test 5	0:54:48	5,10	Failed	99.104	83.717	10.538	25.191	68.834
EASL	MPPI	iter3	Test 5	1:54:08	4,20	Failed	96.828	83.989	11.532	37.506	70.000
EASL	NLOPT	iter3	Test 5	1:45:16	4,24	Failed	99.246	83.312	10.771	38.060	70.016
EASL	MPPI	iter4	Test 5	2:00:56	4,14	Failed	86.018	82.344	11.148	35.095	69.032
EASL	NLOPT	iter4	Test 5	1:07:24	5,7	Failed	99.091	83.523	10.165	29.797	69.477
EASL	MPPI	iter5	Test 5	1:00:12	4,22	Stuck	99.890	84.356	12.051	27.084	70.921
EASL	NLOPT	iter5	Test 5	1:07:28	5,6	Failed	98.266	82.406	10.266	25.734	68.906
GLS	NLOPT	iter1	Test 5	1:31:44	11,2	Failed	60.052	84.579	16.160	35.188	74.063
GLS	MPPI	iter1	Test 5	1:27:40	4,23	Failed	61.384	84.665	16.681	39.295	75.058
GLS	NLOPT	iter2	Test 5	0:54:48	5,7	Failed	47.963	87.250	13.950	34.125	69.750
GLS	MPPI	iter2	Test 5	1:24:48	5,19	Failed	62.630	87.526	16.943	40.711	76.581
GLS	NLOPT	iter3	Test 5	0:54:00	5,7	Failed	50.700	78.250	13.863	28.500	67.750

GLS	MPPI	iter3	Test 5	1:13:52	5,12	Failed	64.158	85.341	16.731	39.161	75.025
GLS	NLOPT	iter4	Test 5	0:50:56	5,7	Failed	56.225	83.438	14.969	35.000	71.438
GLS	MPPI	iter4	Test 5	1:32:12	2,8	Failed	64.736	84.902	16.771	35.015	75.011
GLS	NLOPT	iter5	Test 5	0:53:52	5,8	Failed	50.588	75.875	13.963	30.500	66.875
GLS	MPPI	iter5	Test 5	1:26:00	5,23	Failed	64.638	87.422	16.753	44.421	77.157
SBPL	MPPI	iter1	Test 5	1:22:25	4,12,24	Failed	60.652	84.353	17.232	35.099	76.314
SBPL	NLOPT	iter1	Test 5	0:56:00	5,4	Failed	49.638	81.250	14.388	32.125	69.875
SBPL	NLOPT	iter2	Test 5	0:50:48	6,4	Failed	57.950	82.625	15.250	31.188	70.813
SBPL	MPPI	iter2	Test 5	1:26:08	4,17	Failed	61.644	83.052	16.969	32.650	75.049
SBPL	NLOPT	iter3	Test 5	0:46:00	5,3	Failed	57.950	83.500	15.500	32.250	71.688
SBPL	MPPI	iter3	Test 5	1:26:08	5,10	Failed	64.523	84.528	17.323	36.998	76.136
SBPL	NLOPT	iter4	Test 5	0:44:40	5,5	Failed	42.925	74.750	12.100	26.500	59.250
SBPL	MPPI	iter4	Test 5	1:45:28	4,17	Failed	60.115	85.311	17.203	35.220	74.065
SBPL	NLOPT	iter5	Test 5	0:44:04	5,5	Failed	51.013	80.125	14.413	30.250	69.500
SBPL	MPPI	iter5	Test 5	1:22:24	4,17	Failed	60.664	87.028	17.416	38.728	76.656
SLGP	NLOPT	iter1	Test 5	0:52:24	5,8	Failed	46.425	78.125	13.875	33.000	66.500
SLGP	MPPI	iter1	Test 5	1:44:44	15,34	Failed	58.246	91.078	16.785	35.894	74.503
SLGP	NLOPT	iter2	Test 5	0:48:44	5,8	Failed	49.713	79.250	13.950	27.750	66.875
SLGP	MPPI	iter2	Test 5	1:49:44	16,33	Failed	57.845	84.507	16.997	32.084	75.167
SLGP	NLOPT	iter3	Test 5	1:16:48	-0,01	Stuck	51.386	83.168	15.861	38.723	74.043
SLGP	MPPI	iter3	Test 5	1:51:28	15,34	Failed	61.168	84.719	17.098	37.193	74.071
SLGP	MPPI	iter4	Test 5	1:45:04	15,33	Failed	57.960	84.193	16.930	35.867	74.582
SLGP	NLOPT	iter4	Test 5	0:53:20	5,8	Failed	41.725	67.750	11.875	27.250	57.750
SLGP	MPPI	iter5	Test 5	1:44:32	15,36	Failed	55.877	82.394	16.956	32.483	74.290
SLGP	NLOPT	iter5	Test 5	0:48:08	5,8	Failed	49.013	75.375	13.963	29.625	66.625

EASL	MPPI	iter1	Test 6	1:51:16	27,37	Success	99.921	84.031	11.648	23.806	70.985
EASL	NLOPT	iter1	Test 6	1:17:40	8,17	Failed	99.962	84.512	10.570	34.304	70.991
EASL	MPPI	iter2	Test 6	1:46:00	27,38	Success	55.233	99.988	10.930	28.278	79.930
EASL	NLOPT	iter2	Test 6	0:43:12	3,3	Failed	83.794	75.500	9.313	23.125	63.500
EASL	MPPI	iter3	Test 6	1:49:00	27,37	Success	70.706	79.927	10.870	23.475	69.503
EASL	NLOPT	iter3	Test 6	0:40:48	3,4	Failed	87.863	75.438	9.381	20.563	63.688
EASL	MPPI	iter4	Test 6	2:01:16	27,24	Success	98.921	83.031	14.648	24.806	72.586
EASL	NLOPT	iter4	Test 6	0:58:00	10,8	Failed	91.955	83.570	10.090	30.063	70.461
EASL	MPPI	iter5	Test 6	2:10:20	27,37	Success	79.568	81.207	11.361	22.550	68.248
EASL	NLOPT	iter5	Test 6	0:42:40	2,4	Failed	90.138	76.563	9.375	26.625	63.688
GLS	NLOPT	iter1	Test 6	0:48:12	1,3	Failed	32.500	45.500	9.400	19.000	38.500
GLS	MPPI	iter1	Test 6	1:33:20	16,23	Failed	62.600	83.413	19.572	34.390	75.029
GLS	NLOPT	iter2	Test 6	0:47:36	2,4	Failed	91.544	76.938	9.531	22.188	63.750
GLS	MPPI	iter2	Test 6	1:49:00	26,37	Success	60.736	83.458	19.680	36.456	74.066
GLS	NLOPT	iter3	Test 6	0:51:52	8,14	Failed	59.128	84.438	18.391	31.313	74.094
GLS	MPPI	iter3	Test 6	1:18:40	7,14	Failed	67.251	84.284	19.576	35.194	75.016
GLS	NLOPT	iter4	Test 6	0:53:08	2,4	Failed	46.225	66.750	14.075	25.500	57.750
GLS	MPPI	iter4	Test 6	1:28:16	12,13	Failed	65.076	83.430	19.596	38.046	75.123
GLS	NLOPT	iter5	Test 6	0:50:24	7,4	Failed	47.375	65.250	14.075	28.000	57.750
GLS	MPPI	iter5	Test 6	1:56:08	26,37	Success	60.683	83.148	19.934	35.294	74.004
SBPL	NLOPT	iter1	Test 6	0:50:20	9,8	Failed	55.038	83.938	18.356	29.125	72.188
SBPL	MPPI	iter1	Test 6	1:44:00	27,36	Success	56.872	81.890	21.313	33.071	74.032
SBPL	NLOPT	iter2	Test 6	0:54:20	9,8	Failed	54.863	74.750	17.125	28.375	66.625
SBPL	MPPI	iter2	Test 6	1:32:36	26,37	Success	57.042	83.322	20.403	42.122	73.632
SBPL	NLOPT	iter3	Test 6	0:57:48	15,13	Failed	59.766	84.328	19.475	37.125	74.109

SBPL	MPPI	iter3	Test 6	1:48:00	27,37	Success	65.130	78.132	10.646	27.130	69.767
SBPL	NLOPT	iter4	Test 6	1:02:00	15,13	Failed	55.969	82.500	18.925	32.094	74.094
SBPL	MPPI	iter4	Test 6	1:21:52	26,37	Success	55.749	82.850	20.742	36.846	74.061
SBPL	NLOPT	iter5	Test 6	1:24:00	27,36	Success	57.021	81.716	20.055	35.385	73.556
SBPL	MPPI	iter5	Test 6	1:32:20	26,37	Success	56.525	82.791	20.264	36.524	74.018
SLGP	NLOPT	iter1	Test 6	0:55:04	3,8	Failed	47.425	80.625	16.500	31.250	67.000
SLGP	MPPI	iter1	Test 6	1:48:08	25,37	Success	49.785	90.104	19.718	33.602	75.375
SLGP	NLOPT	iter2	Test 6	0:59:08	4,8	Failed	55.488	80.875	17.675	32.813	71.688
SLGP	MPPI	iter2	Test 6	1:48:12	14,24	Failed	57.223	87.018	19.774	34.215	74.371
SLGP	NLOPT	iter3	Test 6	1:10:08	8,13	Failed	53.205	93.938	18.669	35.203	78.125
SLGP	MPPI	iter3	Test 6	1:47:36	27,36	Success	54.267	82.516	19.681	28.331	73.258
SLGP	MPPI	iter4	Test 6	1:28:16	16,23	Failed	59.781	83.524	19.552	35.752	74.068
SLGP	NLOPT	iter4	Test 6	1:07:08	8,14	Failed	60.174	86.641	18.920	36.484	75.789
SLGP	MPPI	iter5	Test 6	1:39:48	15,23	Failed	63.684	87.604	19.610	35.120	76.328
SLGP	NLOPT	iter5	Test 6	0:56:20	9,13	Failed	59.859	88.750	18.381	32.250	76.969
EASL	MPPI	iter1	Test 7	2:13:00	27,37	Success	76.434	84.221	10.782	27.119	68.459
EASL	NLOPT	iter1	Test 7	1:53:32	26,56	Success	70.083	84.523	10.333	22.018	70.308
EASL	MPPI	iter2	Test 7	1:57:16	26,37	Success	69.223	82.199	10.368	30.328	70.399
EASL	NLOPT	iter2	Test 7	1:20:20	27,36	Success	75.504	84.228	10.328	22.823	70.002
EASL	MPPI	iter3	Test 7	2:17:08	26,37	Success	75.899	82.198	10.873	30.732	70.063
EASL	NLOPT	iter3	Test 7	1:22:12	27,35	Success	70.698	84.710	10.365	23.757	70.002
EASL	MPPI	iter4	Test 7	1:44:12	27,35	Success	68.235	84.890	10.889	27.889	68.223
EASL	NLOPT	iter4	Test 7	1:04:16	27,36	Success	69.829	84.237	10.233	22.382	70.230
EASL	MPPI	iter5	Test 7	1:02:32	27,35	Success	68.991	84.443	10.443	27.998	68.432
EASL	NLOPT	iter5	Test 7	1:00:36	26,56	Success	70.627	84.792	10.133	22.720	70.308

GLS	NLOPT	iter1	Test 7	2:11:56	27,36	Success	68.115	83.918	9.470	30.860	70.496
GLS	MPPI	iter1	Test 7	2:02:44	27,36	Success	67.472	84.836	9.834	31.655	69.997
GLS	NLOPT	iter2	Test 7	1:38:32	27,36	Success	71.262	82.129	9.028	30.578	70.274
GLS	MPPI	iter2	Test 7	2:08:28	27,35	Success	67.709	84.172	9.856	29.604	70.124
GLS	NLOPT	iter3	Test 7	1:53:24	27,36	Success	60.832	86.408	9.145	30.127	70.979
GLS	MPPI	iter3	Test 7	2:07:04	27,36	Success	68.246	84.679	9.233	30.479	70.210
GLS	NLOPT	iter4	Test 7	1:56:32	26,36	Success	69.246	82.759	9.226	32.876	70.479
GLS	MPPI	iter4	Test 7	1:47:04	27,36	Success	68.323	84.877	9.594	31.479	70.412
GLS	NLOPT	iter5	Test 7	1:32:24	28,35	Success	68.797	83.369	9.083	34.590	70.931
GLS	MPPI	iter5	Test 7	1:33:40	27,36	Success	66.078	83.764	9.528	23.182	70.861
SBPL	NLOPT	iter1	Test 7	1:27:24	27,36	Success	61.852	86.090	9.556	29.211	70.723
SBPL	MPPI	iter1	Test 7	1:49:12	28,36	Success	58.624	98.810	10.605	31.633	80.950
SBPL	NLOPT	iter2	Test 7	1:36:00	28,36	Success	64.669	84.697	9.373	32.330	69.945
SBPL	MPPI	iter2	Test 7	1:34:20	27,36	Success	60.440	88.701	9.957	29.361	71.609
SBPL	NLOPT	iter3	Test 7	1:30:20	27,36	Success	63.442	85.047	9.468	34.520	69.758
SBPL	MPPI	iter3	Test 7	1:27:00	27,36	Success	70.319	87.227	10.120	32.209	70.986
SBPL	NLOPT	iter4	Test 7	1:30:56	27,36	Success	72.245	85.832	9.812	32.417	70.934
SBPL	MPPI	iter4	Test 7	1:36:36	27,36	Success	59.610	85.671	10.159	30.817	70.930
SBPL	NLOPT	iter5	Test 7	1:23:44	26,37	Success	61.436	83.797	9.681	27.930	70.352
SBPL	MPPI	iter5	Test 7	1:36:00	27,36	Success	59.567	87.773	10.001	27.465	70.861
SLGP	NLOPT	iter1	Test 7	1:12:48	15,19	Failed	62.684	85.641	8.703	40.156	69.359
SLGP	MPPI	iter1	Test 7	2:09:00	14,33	Failed	67.433	85.862	9.970	30.960	70.788
SLGP	NLOPT	iter2	Test 7	1:15:20	15,19	Failed	72.306	84.703	8.742	29.281	68.906
SLGP	MPPI	iter2	Test 7	2:02:08	14,33	Failed	67.000	84.829	9.886	32.033	70.679
SLGP	NLOPT	iter3	Test 7	1:12:44	15,20	Failed	73.661	84.859	8.742	33.875	70.063

SLGP	MPPI	iter3	Test 7	2:16:16	15,33	Failed	75.346	82.433	10.132	31.969	70.029
SLGP	NLOPT	iter4	Test 7	1:29:16	15,24	Failed	62.286	84.609	8.899	29.352	69.984
SLGP	MPPI	iter4	Test 7	2:19:44	15,32	Failed	75.537	85.712	10.191	30.523	70.998
SLGP	MPPI	iter5	Test 7	2:04:48	15,32	Failed	75.730	86.109	10.033	29.144	70.109
SLGP	NLOPT	iter5	Test 7	1:21:08	15,25	Failed	71.966	86.500	8.782	31.539	69.609
EASL	MPPI	iter1	Test 8	2:04:16	26,37	Success	69.561	84.384	10.697	30.523	69.250
EASL	NLOPT	iter1	Test 8	1:46:48	26,36	Success	69.516	84.032	10.387	22.742	70.008
EASL	MPPI	iter2	Test 8	1:38:00	26,37	Success	69.549	84.395	10.789	21.296	69.875
EASL	NLOPT	iter2	Test 8	1:46:12	26,37	Success	69.718	83.981	10.366	23.150	70.008
EASL	MPPI	iter3	Test 8	1:47:52	27,37	Success	69.546	83.871	10.845	40.863	69.997
EASL	NLOPT	iter3	Test 8	1:30:28	26,37	Success	69.259	84.021	10.387	24.066	70.003
EASL	MPPI	iter4	Test 8	1:50:44	26,37	Success	69.382	84.007	10.780	24.822	70.000
EASL	NLOPT	iter4	Test 8	1:36:40	26,37	Success	69.620	84.112	10.420	24.754	69.750
EASL	MPPI	iter5	Test 8	1:43:04	26,37	Success	69.549	83.914	10.811	22.393	69.250
EASL	NLOPT	iter5	Test 8	1:25:16	28,38	Success	69.654	84.721	10.441	25.223	70.004
GLS	NLOPT	iter1	Test 8	1:42:40	25,57	Stuck	69.572	84.372	9.800	22.760	69.750
GLS	MPPI	iter1	Test 8	1:59:48	26,37	Success	69.334	84.768	10.233	27.444	70.000
GLS	NLOPT	iter2	Test 8	1:46:00	22,0	Stuck	69.687	83.512	10.050	30.718	69.047
GLS	MPPI	iter2	Test 8	1:45:40	26,37	Success	69.683	83.720	10.295	22.304	69.124
GLS	NLOPT	iter3	Test 8	1:40:04	7,32	Stuck	69.442	84.270	9.853	21.750	69.750
GLS	MPPI	iter3	Test 8	1:55:12	26,37	Success	69.674	84.125	10.392	25.027	69.040
GLS	NLOPT	iter4	Test 8	1:49:00	26,36	Success	69.282	84.477	10.078	24.690	69.500
GLS	MPPI	iter4	Test 8	1:49:48	26,37	Success	69.358	84.909	10.391	27.085	70.000
GLS	NLOPT	iter5	Test 8	1:37:52	27,37	Success	69.529	84.137	9.795	22.602	69.187
GLS	MPPI	iter5	Test 8	1:51:44	26,36	Success	69.550	83.796	10.373	29.180	69.250
SBPL	NLOPT	iter1	Test 8	1:38:20	27,36	Success	69.366	84.755	10.444	33.348	69.250

SBPL	MPPI	iter1	Test 8	1:30:12	26,36	Success	69.546	83.929	10.874	21.350	70.000
SBPL	NLOPT	iter2	Test 8	1:27:40	27,36	Success	69.413	84.044	10.393	25.925	70.001
SBPL	MPPI	iter2	Test 8	1:37:20	26,36	Success	69.392	84.000	11.066	40.663	69.094
SBPL	NLOPT	iter3	Test 8	1:25:56	27,36	Success	69.668	84.820	10.572	40.640	69.490
SBPL	MPPI	iter3	Test 8	1:38:24	26,36	Success	69.770	84.090	10.918	24.208	69.187
SBPL	NLOPT	iter4	Test 8	1:26:12	27,36	Success	69.645	84.071	10.785	25.628	69.187
SBPL	MPPI	iter4	Test 8	1:27:44	26,37	Success	69.367	84.177	10.888	24.598	69.812
SBPL	NLOPT	iter5	Test 8	1:20:44	27,36	Success	69.575	84.090	10.510	21.154	70.000
SBPL	MPPI	iter5	Test 8	1:32:42	27,36	Success	69.547	83.929	10.974	21.324	70.219
SLGP	NLOPT	iter1	Test 8	1:14:32	22,29	Failed	69.453	84.107	9.608	31.275	69.997
SLGP	MPPI	iter1	Test 8	1:36:28	26,29	Failed	69.571	84.502	10.240	22.613	69.937
SLGP	NLOPT	iter2	Test 8	1:03:52	21,28	Failed	69.657	85.221	9.560	24.385	70.006
SLGP	MPPI	iter2	Test 8	1:40:08	26,28	Failed	69.661	84.184	10.445	28.102	69.023
SLGP	NLOPT	iter3	Test 8	0:53:44	15,43	Failed	38.910	24.329	8.115	23.112	81.213
SLGP	MPPI	iter3	Test 8	1:32:48	21,29	Failed	69.641	84.911	10.476	30.663	69.500
SLGP	NLOPT	iter4	Test 8	1:07:36	21,28	Failed	69.384	84.929	9.541	25.293	70.026
SLGP	MPPI	iter4	Test 8	1:25:00	24,29	Failed	69.816	83.877	10.283	32.628	69.969
SLGP	NLOPT	iter5	Test 8	1:21:32	22,29	Failed	69.427	84.578	9.671	31.468	69.996
SLGP	MPPI	iter5	Test 8	1:33:20	23,29	Failed	69.605	83.505	10.337	25.864	70.000

Table 3: Test set 1 with Policy 2

Global Planner	Local Planner	Iter Num	Test Num	End time	End location	Status	CPU Util	CPU temp	RAM	GPU util	GPU temp
SLGP	MPPI	iter1	Test 1	1:12:56	27,36	Failed	53.650	73.250	11.400	50.000	62.375
SLGP	MPPI	iter2	Test 1	1:05:52	26,35	Failed	53.780	74.220	12.340	49.231	61.790
SLGP	MPPI	iter3	Test 1	1:01:48	27,36	Success	53.671	79.328	12.028	51.293	67.723
SLGP	MPPI	iter4	Test 1	1:04:20	26,36	Success	54.115	73.395	11.977	53.828	67.227
SLGP	MPPI	iter5	Test 1	1:12:36	26,36	Success	52.661	75.320	11.927	56.242	67.477
SLGP	MPPI	iter1	Test 2	1:27:40	26,38	Success	53.581	75.874	12.254	51.852	67.935
SLGP	MPPI	iter2	Test 2	1:39:52	25,37	Success	54.475	76.970	13.561	51.452	68.747
SLGP	MPPI	iter3	Test 2	1:19:36	26,37	Success	54.428	75.188	12.177	50.789	67.950
SLGP	MPPI	iter4	Test 2	1:20:56	26,37	Success	53.115	75.428	12.163	54.417	67.901
SLGP	MPPI	iter5	Test 2	1:18:20	26,37	Success	53.807	76.136	12.186	50.115	67.932
SLGP	MPPI	iter1	Test 3	1:38:08	27,36	Success	52.670	77.435	12.177	54.939	67.975
SLGP	MPPI	iter2	Test 3	1:24:32	27,36	Success	53.005	75.019	12.178	54.253	67.975
SLGP	MPPI	iter3	Test 3	1:09:40	12,3	Stuck	53.092	75.617	12.279	54.505	67.896
SLGP	MPPI	iter4	Test 3	2:11:40	4,2	Stuck	53.440	77.437	14.480	54.667	69.000
SLGP	MPPI	iter5	Test 3	1:35:52	27,36	Success	52.400	76.173	12.273	54.757	67.996
SLGP	MPPI	iter1	Test 4	1:27:36	25,38	Success	54.088	76.510	12.107	50.742	67.950
SLGP	MPPI	iter2	Test 4	1:32:16	26,36	Success	55.155	75.790	13.148	49.656	67.980
SLGP	MPPI	iter3	Test 4	1:25:04	26,36	Success	54.101	75.006	13.119	47.331	67.979
SLGP	MPPI	iter4	Test 4	1:21:04	25,37	Success	51.891	78.642	12.166	50.745	67.901
SLGP	MPPI	iter5	Test 4	1:20:48	26,37	Success	54.117	75.936	12.997	51.668	67.971
SLGP	MPPI	iter1	Test 5	1:55:52	15,35	Stuck	52.105	78.547	12.455	57.261	67.999
SLGP	MPPI	iter2	Test 5	3:20:40	5,12	Stuck	53.931	79.946	15.147	56.637	69.866

SLGP	MPPI	iter3	Test 5	1:45:32	15,36	Stuck	53.150	75.907	12.496	48.657	67.999
SLGP	MPPI	iter4	Test 5	1:34:40	15,35	Stuck	53.225	74.947	12.273	54.106	67.982
SLGP	MPPI	iter5	Test 5	5:03:20	4,7	Timeout	54.183	78.033	16.056	59.027	70.059
SLGP	MPPI	iter1	Test 6	1:31:00	27,36	Success	53.375	77.670	13.571	42.505	67.004
SLGP	MPPI	iter2	Test 6	1:37:20	26,37	Success	51.916	80.083	13.499	28.785	66.716
SLGP	MPPI	iter3	Test 6	1:41:00	25,37	Success	49.305	81.898	12.433	30.370	71.472
SLGP	MPPI	iter4	Test 6	2:16:00	27,36	Success	51.322	81.436	13.962	31.327	70.464
SLGP	MPPI	iter5	Test 6	1:54:40	27,37	Success	48.656	87.332	13.916	35.705	70.681
SLGP	MPPI	iter1	Test 7	1:10:40	0,0	Stuck	45.505	80.531	11.664	30.359	69.891
SLGP	MPPI	iter2	Test 7	2:17:32	27,36	Success	47.727	84.698	14.269	32.029	68.791
SLGP	MPPI	iter3	Test 7	1:55:44	21,36	Stuck	47.767	83.254	12.677	31.695	69.771
SLGP	MPPI	iter4	Test 7	2:13:20	21,35	Stuck	52.128	79.652	12.881	35.049	70.233
SLGP	MPPI	iter5	Test 7	1:36:52	20,36	Stuck	49.630	82.132	12.393	28.136	69.194
SLGP	MPPI	iter1	Test 8	1:42:20	24,29	Failed	54.563	83.315	13.120	34.571	70.733
SLGP	MPPI	iter2	Test 8	2:09:40	26,37	Success	50.940	82.427	14.143	34.228	68.591
SLGP	MPPI	iter3	Test 8	1:10:48	0,1	Stuck	45.459	88.859	11.614	36.922	69.938
SLGP	MPPI	iter4	Test 8	1:27:00	24,29	Failed	54.150	82.916	13.306	27.922	70.108
SLGP	MPPI	iter5	Test 8	1:49:56	26,36	Success	50.298	84.511	13.971	31.063	69.427

Table 4: Test set 2 without metareasoning

Global Planner	Local Planner	Iter Num	Test Num	End time	End location	Status	CPU Util	CPU temp	RAM	GPU util	GPU temp
EASL	MPPI	iter1	Test 1	1:36:44	26,36	Success	53.831	74.487	10.582	29.229	67.805
EASL	NLOPT	iter1	Test 1	1:17:36	26,36	Success	51.116	79.218	10.125	29.399	71.188
EASL	MPPI	iter2	Test 1	1:33:28	26,36	Success	49.313	81.885	10.572	33.056	70.487
EASL	NLOPT	iter2	Test 1	1:19:16	25,38	Success	52.885	80.336	10.012	27.387	72.325
EASL	MPPI	iter3	Test 1	1:27:20	26,36	Success	49.073	83.867	10.577	43.261	71.000
EASL	NLOPT	iter3	Test 1	1:18:40	26,36	Success	51.476	80.083	10.114	30.053	70.313
EASL	MPPI	iter4	Test 1	1:30:44	26,36	Success	49.713	82.532	10.654	28.981	70.234
EASL	NLOPT	iter4	Test 1	1:20:40	26,38	Success	48.954	82.532	10.249	35.913	70.431
EASL	MPPI	iter5	Test 1	1:23:40	26,37	Success	55.148	74.734	10.578	34.707	68.846
EASL	NLOPT	iter5	Test 1	1:16:56	26,36	Success	49.829	84.709	10.152	30.005	70.059
GLS	MPPI	iter1	Test 1	1:29:12	25,36	Success	54.489	81.547	9.784	25.679	69.931
GLS	NLOPT	iter1	Test 1	1:20:40	25,36	Success	55.614	83.320	9.332	30.008	69.727
GLS	MPPI	iter2	Test 1	1:30:08	26,36	Success	55.177	81.688	9.758	30.120	70.557
GLS	NLOPT	iter2	Test 1	1:23:32	25,36	Success	57.185	80.105	9.304	28.523	70.160
GLS	MPPI	iter3	Test 1	1:29:16	25,36	Success	55.338	80.684	9.837	29.853	69.994
GLS	NLOPT	iter3	Test 1	1:15:40	25,37	Success	56.778	79.785	9.285	28.703	69.719
GLS	MPPI	iter4	Test 1	1:21:40	25,36	Success	55.809	79.700	9.858	24.604	70.180
GLS	NLOPT	iter4	Test 1	1:18:00	25,36	Success	56.245	82.633	9.309	31.406	69.852
GLS	MPPI	iter5	Test 1	1:32:24	25,36	Success	55.100	81.196	9.808	29.693	69.866
GLS	NLOPT	iter5	Test 1	1:22:12	25,36	Success	55.290	81.484	9.338	27.266	70.727
SBPL	NLOPT	iter1	Test 1	1:22:40	25,36	Success	50.018	81.883	10.172	28.773	69.863
SBPL	MPPI	iter1	Test 1	1:26:04	27,36	Success	46.859	96.273	10.364	37.625	77.279

SBPL	NLOPT	iter2	Test 1	1:20:08	25,36	Success	50.010	79.840	10.350	48.347	69.474
SBPL	MPPI	iter2	Test 1	1:30:40	27,36	Success	48.573	80.839	10.461	30.424	69.932
SBPL	NLOPT	iter3	Test 1	1:17:48	26,36	Success	49.502	80.879	10.445	32.660	69.924
SBPL	MPPI	iter3	Test 1	1:23:00	26,36	Success	47.992	78.097	10.808	31.379	69.558
SBPL	NLOPT	iter4	Test 1	1:20:44	26,36	Success	50.761	80.068	10.301	38.725	69.756
SBPL	MPPI	iter4	Test 1	1:27:12	26,36	Success	48.873	79.802	10.772	26.859	69.683
SBPL	NLOPT	iter5	Test 1	1:25:00	26,36	Success	50.931	83.098	10.140	35.268	70.002
SLGP	MPPI	iter1	Test 1	1:28:40	25,36	Success	48.349	84.010	9.753	50.133	69.861
SLGP	NLOPT	iter1	Test 1	1:20:36	25,36	Success	48.897	82.883	9.305	31.402	69.602
SLGP	MPPI	iter2	Test 1	1:25:40	25,36	Success	48.272	83.595	9.826	29.836	69.932
SLGP	NLOPT	iter2	Test 1	1:14:48	25,36	Success	48.614	80.152	9.336	32.723	69.977
SLGP	MPPI	iter3	Test 1	1:25:40	-0,03	Stuck	45.431	81.703	9.562	27.898	69.727
SLGP	NLOPT	iter3	Test 1	1:19:08	25,36	Success	50.117	85.422	9.295	35.457	69.699
SLGP	MPPI	iter4	Test 1	1:19:52	25,36	Success	48.522	79.424	9.743	31.088	69.863
SLGP	NLOPT	iter4	Test 1	1:12:40	25,36	Success	49.438	81.031	9.295	32.883	69.758
SLGP	MPPI	iter5	Test 1	1:23:28	25,36	Success	48.298	83.718	9.877	33.394	70.057
SLGP	NLOPT	iter5	Test 1	1:13:04	25,36	Success	48.582	83.844	9.308	32.570	69.727
SBPL	MPPI	iter5	Test 1	1:37:28	27,36	Success	48.883	81.382	10.803	28.330	69.779
EASL	MPPI	iter1	Test 2	1:44:40	25,38	Success	51.886	81.982	10.741	28.821	69.977
EASL	NLOPT	iter1	Test 2	1:26:04	25,37	Success	51.499	77.433	10.325	31.304	70.500
EASL	MPPI	iter2	Test 2	1:34:04	23,37	Success	49.584	79.463	10.832	36.793	68.961
EASL	NLOPT	iter2	Test 2	1:34:16	18,22	Failed	49.739	83.910	10.254	33.100	71.001
EASL	MPPI	iter3	Test 2	1:31:56	24,37	Success	50.797	84.872	10.682	34.829	70.031
EASL	NLOPT	iter3	Test 2	1:25:56	25,37	Success	51.240	78.393	9.806	33.099	70.724
EASL	MPPI	iter4	Test 2	1:41:00	24,37	Success	50.078	82.175	10.692	33.069	70.371

EASL	NLOPT	iter4	Test 2	1:25:12	19,21	Failed	53.035	75.332	10.184	30.931	70.128
EASL	MPPI	iter5	Test 2	1:37:12	26,37	Success	50.681	80.510	10.683	28.175	70.125
EASL	NLOPT	iter5	Test 2	1:35:32	24,37	Success	51.327	80.486	10.179	35.853	70.250
GLS	MPPI	iter1	Test 2	2:39:20	13,2	Stuck	56.924	80.865	11.021	32.686	70.237
GLS	NLOPT	iter1	Test 2	1:50:00	5,24	Failed	57.496	80.170	9.870	37.940	70.499
GLS	MPPI	iter2	Test 2	1:25:16	7,3	Stuck	56.153	82.426	9.948	40.172	70.900
GLS	NLOPT	iter2	Test 2	1:00:32	5,4	Failed	53.694	75.875	8.663	27.625	66.563
GLS	MPPI	iter3	Test 2	2:15:08	19,57	Stuck	57.750	78.888	10.595	38.932	69.305
GLS	NLOPT	iter3	Test 2	1:13:52	5,24	Failed	54.575	83.906	9.354	32.988	70.500
GLS	MPPI	iter4	Test 2	2:00:32	19,32	Failed	55.504	81.521	10.148	33.366	69.500
GLS	NLOPT	iter4	Test 2	0:58:24	6,4	Failed	53.928	80.031	8.959	33.406	67.813
GLS	MPPI	iter5	Test 2	1:54:00	3,21	Stuck	56.535	82.073	10.235	33.567	70.506
GLS	NLOPT	iter5	Test 2	1:09:56	5,12	Failed	55.725	81.922	9.274	33.219	70.719
SBPL	NLOPT	iter1	Test 2	1:35:00	24,37	Success	48.698	81.931	10.936	33.454	69.978
SBPL	MPPI	iter2	Test 2	1:38:04	24,36	Success	50.011	83.630	11.186	35.659	69.988
SBPL	NLOPT	iter2	Test 2	1:32:36	24,37	Success	49.714	79.634	10.804	33.142	69.964
SBPL	MPPI	iter3	Test 2	1:34:20	25,36	Success	48.125	84.380	11.076	32.824	69.987
SBPL	NLOPT	iter3	Test 2	1:25:24	25,37	Success	49.515	81.146	10.805	31.921	69.932
SBPL	MPPI	iter4	Test 2	1:34:12	24,37	Success	48.611	81.577	10.836	35.066	69.986
SBPL	NLOPT	iter4	Test 2	1:25:00	26,36	Success	48.199	81.832	10.840	33.040	69.962
SBPL	MPPI	iter5	Test 2	1:39:28	24,37	Success	47.597	82.654	10.959	32.277	69.712
SBPL	NLOPT	iter5	Test 2	1:33:24	25,36	Success	49.374	83.774	10.370	36.746	69.932
SLGP	MPPI	iter1	Test 2	1:39:52	24,37	Success	47.089	81.800	9.944	34.789	69.733
SLGP	NLOPT	iter1	Test 2	1:23:28	-1,02	Stuck	45.666	81.328	9.172	33.531	69.719
SLGP	MPPI	iter2	Test 2	1:40:08	24,37	Success	48.541	83.199	9.943	29.570	69.983

SLGP	NLOPT	iter2	Test 2	1:08:40	5,8	Failed	48.281	75.313	8.675	28.813	65.688
SLGP	MPPI	iter3	Test 2	1:35:24	23,38	Success	47.168	82.345	9.892	32.205	69.957
SLGP	NLOPT	iter3	Test 2	1:04:28	5,7	Failed	48.500	76.750	8.675	26.688	66.563
SLGP	MPPI	iter4	Test 2	1:40:48	24,37	Success	46.793	82.411	9.881	34.999	69.984
SLGP	NLOPT	iter4	Test 2	1:05:52	5,7	Failed	48.419	74.125	8.700	31.250	65.625
SLGP	MPPI	iter5	Test 2	1:38:16	24,37	Success	48.517	81.127	9.869	29.968	70.111
SLGP	NLOPT	iter5	Test 2	0:58:04	5,7	Failed	48.600	76.875	8.700	29.750	66.563
SBPL	MPPI	iter1	Test 2	5:05:20	5,26	Timeout	47.698	83.307	11.135	31.318	69.969
EASL	MPPI	iter1	Test 3	1:38:44	25,38	Success	49.344	82.352	10.684	29.297	69.395
EASL	NLOPT	iter1	Test 3	0:58:04	6,2	Failed	51.859	80.844	9.653	33.281	69.375
EASL	MPPI	iter2	Test 3	1:45:20	26,36	Success	48.461	84.860	10.667	31.701	68.579
EASL	NLOPT	iter2	Test 3	1:34:20	26,36	Success	55.562	73.896	10.316	32.571	68.346
EASL	MPPI	iter3	Test 3	1:44:20	26,36	Success	50.128	83.119	10.833	26.752	70.035
EASL	NLOPT	iter3	Test 3	1:18:16	26,36	Success	53.840	79.087	10.265	27.175	72.909
EASL	MPPI	iter4	Test 3	1:38:40	26,36	Success	48.995	81.048	10.769	27.994	70.074
EASL	NLOPT	iter4	Test 3	1:36:00	26,36	Success	49.822	79.877	10.345	25.724	69.546
EASL	MPPI	iter5	Test 3	1:51:32	24,38	Success	50.621	79.560	10.984	30.986	69.054
EASL	NLOPT	iter5	Test 3	1:36:00	26,38	Success	54.725	73.389	9.690	31.319	68.461
GLS	MPPI	iter1	Test 3	2:52:00	18,3	Stuck	56.467	78.463	10.842	26.996	68.983
GLS	NLOPT	iter1	Test 3	1:01:20	4,3	Failed	52.450	75.750	8.675	29.250	66.125
GLS	MPPI	iter2	Test 3	1:21:00	-1,03	Stuck	46.331	83.641	9.565	30.934	69.941
GLS	NLOPT	iter2	Test 3	0:58:40	3,3	Failed	52.338	77.813	8.725	30.063	65.938
GLS	MPPI	iter3	Test 3	1:54:04	7,2	Stuck	59.623	77.640	10.421	26.317	69.500
GLS	NLOPT	iter3	Test 3	1:01:20	3,2	Failed	51.888	76.688	8.675	29.375	65.625
GLS	MPPI	iter4	Test 3	1:58:12	5,2	Stuck	56.017	83.677	10.381	36.109	70.874

GLS	NLOPT	iter4	Test 3	0:55:36	3,3	Failed	54.000	77.500	8.700	38.063	65.625
GLS	MPPI	iter5	Test 3	1:43:40	26,36	Success	55.676	83.171	10.042	30.054	69.996
GLS	NLOPT	iter5	Test 3	1:06:32	3,3	Failed	53.981	78.063	8.675	29.563	66.563
SBPL	MPPI	iter1	Test 3	1:28:00	25,36	Success	49.151	85.897	11.038	31.873	69.856
SBPL	NLOPT	iter1	Test 3	1:36:40	26,36	Success	49.327	81.597	10.578	27.330	70.216
SBPL	MPPI	iter2	Test 3	1:41:24	25,37	Success	48.392	84.888	11.043	32.554	70.108
SBPL	NLOPT	iter2	Test 3	1:27:56	26,36	Success	49.113	79.628	10.592	27.784	69.682
SBPL	MPPI	iter3	Test 3	1:31:20	24,37	Success	48.863	82.351	10.836	30.092	69.920
SBPL	NLOPT	iter3	Test 3	1:35:40	25,36	Success	49.259	83.929	10.962	29.038	70.403
SBPL	MPPI	iter4	Test 3	1:39:00	26,36	Success	47.663	82.942	10.735	29.769	69.991
SBPL	NLOPT	iter4	Test 3	1:29:44	26,36	Success	49.992	79.589	10.640	25.792	69.964
SBPL	MPPI	iter5	Test 3	1:32:00	26,36	Success	48.918	84.803	11.178	36.015	69.983
SBPL	NLOPT	iter5	Test 3	1:37:28	26,36	Success	49.475	84.094	10.672	36.733	69.966
SLGP	MPPI	iter1	Test 3	1:47:40	25,36	Success	47.680	82.693	9.920	32.032	69.996
SLGP	NLOPT	iter1	Test 3	0:56:20	2,2	Failed	48.369	77.938	8.675	28.125	65.625
SLGP	MPPI	iter2	Test 3	1:53:56	26,36	Success	47.347	83.212	10.284	29.436	70.030
SLGP	NLOPT	iter2	Test 3	0:55:32	2,2	Failed	47.825	78.188	8.675	30.438	66.563
SLGP	MPPI	iter3	Test 3	1:25:36	10,2	Stuck	48.453	81.819	9.963	34.081	70.217
SLGP	NLOPT	iter3	Test 3	1:06:08	1,2	Failed	48.419	75.125	8.681	27.250	65.563
SLGP	MPPI	iter4	Test 3	1:37:16	26,36	Success	49.192	81.361	9.989	46.438	70.031
SLGP	NLOPT	iter4	Test 3	1:00:56	1,2	Failed	49.006	79.438	8.600	31.063	66.750
SLGP	MPPI	iter5	Test 3	1:39:20	25,36	Success	47.484	83.843	9.979	24.869	69.991
SLGP	NLOPT	iter5	Test 3	1:03:56	2,2	Failed	49.063	78.250	8.713	28.750	65.625
EASL	MPPI	iter1	Test 4	1:39:04	26,37	Success	53.380	70.076	10.781	30.245	65.155
EASL	NLOPT	iter1	Test 4	1:33:44	26,37	Success	53.468	71.154	10.360	37.872	66.016

EASL	MPPI	iter2	Test 4	1:39:48	23,37	Success	52.941	70.432	10.881	29.059	66.063
EASL	NLOPT	iter2	Test 4	1:30:04	25,37	Success	53.115	70.615	10.171	28.830	66.012
EASL	MPPI	iter3	Test 4	1:45:04	24,37	Success	54.180	71.237	10.768	30.786	66.003
EASL	NLOPT	iter3	Test 4	1:29:56	25,37	Success	53.455	72.741	10.290	30.667	66.068
EASL	MPPI	iter4	Test 4	1:40:04	23,37	Success	52.888	71.440	10.686	29.055	65.750
EASL	NLOPT	iter4	Test 4	1:23:20	25,37	Success	54.472	71.701	10.279	31.477	66.244
EASL	MPPI	iter5	Test 4	1:32:48	25,37	Success	52.998	71.912	10.760	43.248	65.507
EASL	NLOPT	iter5	Test 4	0:54:56	7,3	Failed	54.047	69.625	9.653	31.313	65.000
GLS	MPPI	iter1	Test 4	1:47:16	7,3	Stuck	59.610	75.337	10.280	28.539	69.421
GLS	NLOPT	iter1	Test 4	0:54:08	3,3	Failed	52.688	75.625	8.794	30.438	65.625
GLS	MPPI	iter2	Test 4	1:24:16	4,3	Stuck	55.963	83.930	10.033	34.168	70.403
GLS	NLOPT	iter2	Test 4	1:13:20	-2,1	Failed	57.334	81.141	9.302	33.852	69.453
GLS	MPPI	iter3	Test 4	1:36:16	25,37	Success	56.124	79.241	9.994	33.038	70.366
GLS	NLOPT	iter3	Test 4	1:09:24	7,3	Failed	53.863	80.406	9.088	31.000	67.813
GLS	MPPI	iter4	Test 4	1:22:56	2,3	Stuck	55.980	81.914	9.887	35.891	70.900
GLS	NLOPT	iter4	Test 4	0:59:12	3,3	Failed	53.050	77.313	8.769	32.813	65.625
GLS	MPPI	iter5	Test 4	1:20:40	-0,03	Stuck	46.969	81.863	9.662	32.582	69.723
GLS	NLOPT	iter5	Test 4	1:18:24	11,5	Failed	55.250	80.078	9.241	32.094	69.391
SBPL	MPPI	iter1	Test 4	1:41:04	24,37	Success	49.356	84.756	11.332	28.897	69.983
SBPL	NLOPT	iter1	Test 4	1:21:48	-0,02	Stuck	46.548	81.371	9.363	27.484	69.727
SBPL	MPPI	iter2	Test 4	1:36:08	25,37	Success	49.778	81.035	11.112	26.039	70.761
SBPL	NLOPT	iter2	Test 4	1:23:00	-1,03	Stuck	46.133	83.117	9.170	30.605	69.723
SBPL	MPPI	iter3	Test 4	1:46:20	24,37	Success	48.750	86.485	11.533	31.111	69.991
SBPL	NLOPT	iter3	Test 4	1:22:44	25,36	Success	50.672	79.497	10.444	44.742	69.979
SBPL	MPPI	iter4	Test 4	1:18:04	-1,02	Stuck	45.748	82.504	9.559	30.086	69.723

SBPL	NLOPT	iter4	Test 4	1:30:56	25,36	Success	48.989	80.141	10.593	29.481	69.932
SBPL	MPPI	iter5	Test 4	1:39:44	24,37	Success	48.730	81.934	11.430	32.125	69.983
SBPL	NLOPT	iter5	Test 4	1:11:16	15,16	Failed	62.927	80.445	10.395	28.063	69.445
SLGP	MPPI	iter1	Test 4	1:34:00	25,37	Success	48.788	82.192	9.994	28.487	69.990
SLGP	NLOPT	iter1	Test 4	0:54:36	2,2	Failed	47.744	73.688	8.713	29.000	64.063
SLGP	MPPI	iter2	Test 4	1:45:28	24,38	Success	50.190	73.619	10.070	27.172	68.803
SLGP	NLOPT	iter2	Test 4	0:50:20	2,2	Failed	52.400	66.000	8.650	34.250	60.875
SLGP	MPPI	iter3	Test 4	1:38:08	25,37	Success	52.167	70.579	10.079	31.208	65.000
SLGP	NLOPT	iter3	Test 4	1:01:36	2,2	Failed	48.550	73.813	8.725	31.875	63.250
SLGP	MPPI	iter4	Test 4	1:41:52	25,37	Success	48.778	82.470	10.020	31.042	69.737
SLGP	NLOPT	iter4	Test 4	1:02:36	3,2	Failed	51.994	66.188	8.656	25.563	61.875
SLGP	MPPI	iter5	Test 4	1:43:24	24,37	Success	53.049	72.052	10.039	23.459	66.012
SLGP	NLOPT	iter5	Test 4	0:52:40	1,2	Failed	51.475	65.375	8.650	25.813	60.938
EASL	MPPI	iter1	Test 5	5:04:40	-2,2	Timeout	64.136	74.093	13.421	31.392	67.064
EASL	NLOPT	iter1	Test 5	2:30:36	26,25	Stuck	58.682	72.142	10.799	27.423	66.059
EASL	MPPI	iter2	Test 5	5:04:40	27,57	Timeout	63.386	73.392	13.370	31.834	66.939
EASL	NLOPT	iter2	Test 5	2:29:08	23,23	Stuck	58.577	73.353	10.796	35.433	67.036
EASL	MPPI	iter3	Test 5	2:51:20	26,25	Stuck	60.729	71.904	11.598	27.576	66.000
EASL	NLOPT	iter3	Test 5	2:32:00	26,25	Failed	58.736	73.322	10.923	23.382	66.250
EASL	MPPI	iter4	Test 5	2:54:52	25,26	Stuck	60.099	72.593	11.590	38.061	66.112
EASL	NLOPT	iter4	Test 5	0:58:04	7,3	Failed	54.419	72.531	9.688	34.875	65.000
EASL	MPPI	iter5	Test 5	4:21:44	15,57	Stuck	62.376	74.564	12.783	32.758	68.346
EASL	NLOPT	iter5	Test 5	2:32:44	26,25	Failed	58.451	72.343	10.893	24.402	67.001
GLS	MPPI	iter1	Test 5	2:30:40	29,22	Stuck	56.945	76.949	11.033	30.548	69.460
GLS	NLOPT	iter1	Test 5	1:04:28	20,3	Failed	54.598	80.313	9.267	35.281	67.922

GLS	MPPI	iter2	Test 5	2:48:44	21,3	Stuck	55.532	75.930	11.154	37.663	69.624
GLS	NLOPT	iter2	Test 5	1:18:40	-0,02	Stuck	46.039	86.402	9.264	38.211	69.699
GLS	MPPI	iter3	Test 5	1:58:48	8,3	Stuck	59.605	74.945	10.358	36.753	68.312
GLS	NLOPT	iter3	Test 5	1:28:56	21,3	Failed	54.223	80.356	9.559	30.175	69.925
GLS	MPPI	iter4	Test 5	2:10:12	5,19	Stuck	56.352	78.509	10.687	34.428	68.867
GLS	NLOPT	iter4	Test 5	3:03:40	5,14	Failed	58.176	82.157	10.948	30.731	68.999
GLS	MPPI	iter5	Test 5	1:39:44	4,19	Stuck	57.644	80.050	10.273	38.235	68.654
GLS	NLOPT	iter5	Test 5	1:03:20	5,3	Failed	53.081	75.438	8.763	26.688	64.688
SBPL	MPPI	iter1	Test 5	2:22:20	25,16	Failed	55.095	71.312	11.964	28.162	65.109
SBPL	NLOPT	iter1	Test 5	1:16:28	17,57	Failed	59.590	72.606	10.896	32.298	66.396
SBPL	MPPI	iter2	Test 5	4:42:12	22,25	Stuck	58.615	67.993	14.229	28.794	64.969
SBPL	NLOPT	iter2	Test 5	2:11:48	28,25	Failed	61.241	71.289	11.635	34.917	65.266
SBPL	NLOPT	iter3	Test 5	2:12:32	27,24	Failed	60.751	71.032	11.583	32.495	65.000
SBPL	MPPI	iter4	Test 5	2:41:20	22,24	Failed	56.263	72.514	12.426	26.747	65.001
SBPL	NLOPT	iter4	Test 5	2:03:40	23,23	Failed	60.456	70.372	11.431	34.601	65.125
SBPL	MPPI	iter5	Test 5	2:55:52	25,11	Failed	57.303	69.636	12.393	28.522	65.001
SBPL	NLOPT	iter5	Test 5	2:53:00	8,2	Failed	58.750	75.836	12.440	31.958	69.749
SLGP	MPPI	iter1	Test 5	1:55:56	15,35	Stuck	46.299	81.920	10.242	36.787	69.999
SLGP	NLOPT	iter1	Test 5	1:02:12	5,7	Failed	48.069	75.000	8.713	30.563	65.625
SLGP	MPPI	iter2	Test 5	1:59:32	15,36	Stuck	47.163	80.673	10.293	27.085	69.995
SLGP	NLOPT	iter2	Test 5	0:56:08	5,7	Failed	48.125	76.000	8.769	35.000	65.625
SLGP	MPPI	iter3	Test 5	2:09:16	16,34	Stuck	47.894	80.612	10.430	28.019	69.906
SLGP	NLOPT	iter3	Test 5	0:57:16	5,8	Failed	47.538	75.500	8.769	30.938	66.375
SLGP	MPPI	iter4	Test 5	2:08:12	15,35	Stuck	46.588	81.999	10.283	46.427	69.735
SLGP	NLOPT	iter4	Test 5	1:00:08	5,9	Failed	47.475	79.313	9.056	34.625	68.000
SLGP	MPPI	iter5	Test 5	2:28:48	15,36	Stuck	48.206	80.955	10.774	25.508	70.000

SLGP	NLOPT	iter5	Test 5	1:18:36	-0,03	Stuck	50.119	70.520	9.265	27.914	66.074
SBPL	MPPI	iter3	Test 5	5:04:40	15,58	Timeout	54.756	70.020	14.321	36.514	65.379
SBPL	MPPI	iter1	Test 6	1:39:20	25,36	Success	53.422	72.189	11.540	30.258	65.123
EASL	MPPI	iter1	Test 6	1:44:04	26,37	Success	54.169	70.847	10.790	44.611	65.039
EASL	NLOPT	iter1	Test 6	1:50:20	25,37	Success	51.880	81.879	10.595	31.657	67.978
EASL	MPPI	iter2	Test 6	1:57:00	26,37	Success	53.742	71.278	10.942	38.956	65.256
EASL	NLOPT	iter2	Test 6	1:38:04	26,36	Success	48.697	81.247	10.621	28.207	68.566
EASL	MPPI	iter3	Test 6	1:11:56	8,4	Stuck	99.540	81.128	10.387	27.326	69.999
EASL	NLOPT	iter3	Test 6	1:38:24	26,38	Success	50.956	78.617	10.648	26.144	67.171
EASL	MPPI	iter4	Test 6	2:10:44	28,36	Success	63.181	73.997	11.067	25.006	66.912
EASL	NLOPT	iter4	Test 6	0:52:56	7,3	Failed	51.563	79.625	9.906	34.656	68.063
EASL	MPPI	iter5	Test 6	1:33:08	21,2	Stuck	99.626	79.766	10.612	24.369	70.000
EASL	NLOPT	iter5	Test 6	1:33:40	25,37	Success	50.323	79.396	10.579	37.235	67.924
GLS	MPPI	iter1	Test 6	2:00:44	24,37	Success	55.763	82.026	10.392	25.924	69.742
GLS	NLOPT	iter1	Test 6	1:12:48	25,36	Success	58.569	72.884	9.462	29.832	64.903
GLS	MPPI	iter2	Test 6	1:30:32	10,8	Stuck	55.379	78.077	9.988	27.454	69.561
GLS	NLOPT	iter2	Test 6	1:13:04	14,20	Failed	59.402	71.990	9.400	31.721	65.555
GLS	MPPI	iter3	Test 6	2:16:40	25,36	Success	56.281	81.770	10.660	29.171	68.520
GLS	NLOPT	iter3	Test 6	0:58:20	10,8	Failed	59.159	70.516	9.111	35.672	63.703
GLS	MPPI	iter4	Test 6	1:45:12	14,26	Stuck	57.660	75.496	10.369	33.231	69.070
GLS	NLOPT	iter4	Test 6	0:56:04	8,3	Failed	57.466	67.469	9.006	27.469	63.469
GLS	MPPI	iter5	Test 6	1:54:20	25,37	Success	57.722	73.312	10.431	37.658	68.070
GLS	NLOPT	iter5	Test 6	1:12:48	20,30	Failed	60.040	72.081	9.438	24.930	65.810
SBPL	MPPI	iter2	Test 6	1:37:40	26,36	Success	56.377	72.394	11.431	28.827	65.996
SBPL	NLOPT	iter1	Test 6	1:00:28	9,3	Failed	53.559	77.594	10.097	28.375	67.813

SBPL	MPPI	iter3	Test 6	1:42:00	25,36	Success	50.259	78.145	11.724	30.335	66.680
SBPL	NLOPT	iter2	Test 6	1:14:00	20,18	Failed	63.511	80.977	10.970	30.262	69.789
SBPL	MPPI	iter4	Test 6	1:50:40	26,36	Success	55.050	75.291	12.038	29.409	68.872
SBPL	NLOPT	iter3	Test 6	1:42:20	26,36	Success	56.203	77.935	11.265	31.750	68.366
SBPL	MPPI	iter5	Test 6	1:43:40	25,35	Success	54.234	73.815	11.832	30.636	68.330
SBPL	NLOPT	iter4	Test 6	1:07:40	13,12	Failed	54.416	81.672	10.502	33.422	69.453
SBPL	NLOPT	iter5	Test 6	1:08:36	13,8	Failed	55.578	81.266	10.633	29.656	70.063
SLGP	MPPI	iter1	Test 6	2:09:04	24,37	Success	53.794	72.081	10.608	24.788	65.750
SLGP	NLOPT	iter1	Test 6	1:00:16	3,3	Failed	53.091	68.906	8.909	27.438	62.750
SLGP	MPPI	iter2	Test 6	1:51:48	24,37	Success	52.075	70.539	10.121	29.333	64.999
SLGP	NLOPT	iter2	Test 6	0:55:00	4,3	Failed	53.253	70.125	8.925	25.594	63.938
SLGP	MPPI	iter3	Test 6	1:17:36	-1,05	Stuck	50.053	67.438	9.577	25.094	63.723
SLGP	NLOPT	iter3	Test 6	1:11:24	-0,03	Stuck	49.970	71.273	9.169	38.594	64.867
SLGP	MPPI	iter4	Test 6	1:49:32	23,38	Success	52.857	70.473	10.068	26.081	65.693
SLGP	NLOPT	iter4	Test 6	1:01:08	2,3	Failed	47.881	81.813	8.713	32.813	65.000
SLGP	MPPI	iter5	Test 6	1:07:56	-1,03	Stuck	51.053	68.840	9.577	45.137	63.738
SLGP	NLOPT	iter5	Test 6	1:05:00	3,3	Failed	49.869	75.875	8.769	29.000	67.000
SBPL	MPPI	iter1	Test 7	1:52:32	26,36	Success	47.016	85.359	11.277	33.111	69.120
EASL	MPPI	iter1	Test 7	2:49:28	26,38	Success	53.899	70.531	11.699	38.912	65.759
EASL	NLOPT	iter1	Test 7	2:14:00	26,36	Success	53.085	70.805	10.897	28.669	66.024
EASL	MPPI	iter2	Test 7	2:16:20	24,38	Success	68.219	74.352	11.455	25.819	67.000
EASL	NLOPT	iter2	Test 7	1:16:24	20,34	Failed	59.673	72.416	10.282	42.810	65.952
EASL	MPPI	iter3	Test 7	1:02:20	-1,03	Stuck	50.722	69.793	10.459	33.730	65.020
EASL	NLOPT	iter3	Test 7	1:18:24	20,37	Failed	59.438	73.677	10.388	33.628	65.960
EASL	MPPI	iter4	Test 7	2:06:24	26,38	Success	53.154	73.835	11.223	31.049	66.512

EASL	NLOPT	iter4	Test 7	2:11:20	26,36	Success	53.410	72.519	10.803	24.579	66.759
EASL	MPPI	iter5	Test 7	3:34:48	26,36	Stuck	51.873	71.717	12.097	28.922	65.204
EASL	NLOPT	iter5	Test 7	2:19:20	27,37	Success	53.214	71.819	11.049	29.910	66.191
GLS	MPPI	iter1	Test 7	2:10:08	20,16	Stuck	54.726	80.506	10.694	34.436	68.887
GLS	NLOPT	iter1	Test 7	2:24:12	21,37	Failed	57.126	80.116	10.602	27.881	68.243
GLS	MPPI	iter2	Test 7	2:14:00	26,36	Success	56.029	79.704	10.596	25.951	68.125
GLS	NLOPT	iter2	Test 7	1:37:00	23,10	Failed	57.312	80.370	9.942	35.527	70.049
GLS	MPPI	iter3	Test 7	2:23:52	26,36	Success	55.620	80.032	11.139	38.904	69.024
GLS	NLOPT	iter3	Test 7	2:39:00	20,16	Failed	69.639	79.809	10.879	27.898	69.523
GLS	MPPI	iter4	Test 7	1:55:36	23,8	Stuck	54.974	79.391	10.534	23.899	68.121
GLS	NLOPT	iter4	Test 7	1:49:12	21,36	Failed	54.607	80.151	9.990	31.336	68.914
GLS	MPPI	iter5	Test 7	2:02:04	26,36	Success	54.301	81.062	10.569	34.842	68.125
GLS	NLOPT	iter5	Test 7	1:55:08	-1,38	Failed	59.866	76.013	10.266	31.949	68.326
SBPL	MPPI	iter2	Test 7	1:43:00	27,36	Success	48.073	81.197	11.516	26.445	69.425
SBPL	NLOPT	iter1	Test 7	1:42:24	26,36	Success	50.259	81.283	10.827	27.446	69.272
SBPL	MPPI	iter3	Test 7	1:39:40	26,36	Success	48.704	80.335	11.361	33.779	68.996
SBPL	NLOPT	iter2	Test 7	1:45:28	26,36	Success	49.052	80.296	11.140	28.589	69.037
SBPL	MPPI	iter4	Test 7	1:44:48	26,36	Success	48.024	78.962	11.295	33.095	69.994
SBPL	NLOPT	iter3	Test 7	1:44:24	26,36	Success	50.759	80.781	11.413	33.743	69.105
SBPL	MPPI	iter5	Test 7	2:42:20	26,36	Success	48.351	81.650	13.029	28.495	67.929
SBPL	NLOPT	iter4	Test 7	1:29:36	25,36	Success	49.363	80.550	11.076	27.511	69.294
SBPL	NLOPT	iter5	Test 7	1:43:04	26,36	Success	49.028	81.855	11.076	46.566	69.343
SLGP	MPPI	iter1	Test 7	2:15:44	20,36	Stuck	49.066	79.272	10.734	25.853	68.641
SLGP	NLOPT	iter1	Test 7	1:19:40	21,32	Failed	52.220	69.785	9.514	30.457	64.730
SLGP	MPPI	iter2	Test 7	1:45:04	20,36	Stuck	50.658	76.930	10.383	37.588	68.378

SLGP	NLOPT	iter2	Test 7	1:19:48	21,33	Failed	53.105	70.766	9.501	31.496	64.434
SLGP	MPPI	iter3	Test 7	1:49:00	20,36	Stuck	50.640	75.102	10.370	28.674	68.185
SLGP	NLOPT	iter3	Test 7	1:19:20	21,31	Failed	53.166	70.930	9.451	30.438	65.621
SLGP	MPPI	iter4	Test 7	1:58:20	20,36	Stuck	52.058	71.250	10.385	31.664	66.017
SLGP	NLOPT	iter4	Test 7	1:05:40	21,30	Failed	53.148	72.039	9.527	36.039	65.480
SLGP	MPPI	iter5	Test 7	1:50:24	20,36	Stuck	52.538	70.674	10.358	49.400	65.755
SLGP	NLOPT	iter5	Test 7	1:17:00	-1,04	Stuck	50.938	70.836	9.364	28.359	64.867
SBPL	MPPI	iter5	Test 8	1:46:40	21,9	Stuck	48.814	81.565	11.473	29.397	69.006
SBPL	MPPI	iter1	Test 8	1:36:04	25,36	Success	54.722	71.145	11.094	33.617	64.998
EASL	MPPI	iter1	Test 8	1:52:36	24,37	Success	49.766	85.043	11.196	31.955	68.774
EASL	NLOPT	iter1	Test 8	1:39:40	26,37	Success	49.430	78.532	10.671	51.923	67.081
EASL	MPPI	iter2	Test 8	1:46:00	26,37	Success	48.194	78.791	11.142	32.729	67.895
EASL	NLOPT	iter2	Test 8	1:39:20	24,37	Success	50.452	80.655	10.671	31.525	67.173
EASL	MPPI	iter3	Test 8	1:41:32	26,37	Success	51.697	77.280	11.067	27.332	67.585
EASL	NLOPT	iter3	Test 8	1:40:56	25,37	Success	51.805	80.734	10.590	28.254	67.164
EASL	MPPI	iter4	Test 8	1:49:08	24,37	Success	50.151	80.501	11.045	27.715	68.963
EASL	NLOPT	iter4	Test 8	1:40:20	24,37	Success	54.333	72.906	10.589	28.957	66.850
EASL	MPPI	iter5	Test 8	1:12:36	-1,03	Stuck	45.723	82.855	10.480	31.242	69.859
EASL	NLOPT	iter5	Test 8	1:46:04	24,37	Success	51.829	76.558	10.642	29.165	67.430
GLS	MPPI	iter1	Test 8	2:22:32	9,29	Stuck	57.123	81.959	11.217	33.700	69.350
GLS	NLOPT	iter1	Test 8	1:57:56	24,38	Success	57.652	74.369	10.181	31.096	68.891
GLS	MPPI	iter2	Test 8	2:00:24	24,37	Success	57.591	77.033	10.580	36.452	67.164
GLS	NLOPT	iter2	Test 8	1:33:04	6,29	Failed	55.925	80.805	9.859	29.822	71.200
GLS	MPPI	iter3	Test 8	2:01:28	7,27	Stuck	56.125	79.551	10.836	32.076	68.524
GLS	NLOPT	iter3	Test 8	1:29:04	9,32	Failed	55.091	80.959	9.722	33.205	70.683

GLS	MPPI	iter4	Test 8	2:34:44	17,21	Stuck	56.475	80.559	11.080	33.739	69.005
GLS	NLOPT	iter4	Test 8	1:43:16	25,37	Success	57.679	74.028	10.092	34.202	68.621
GLS	MPPI	iter5	Test 8	1:59:08	5,30	Stuck	57.914	81.845	10.784	35.957	69.515
GLS	NLOPT	iter5	Test 8	1:58:20	24,37	Success	56.921	77.345	10.157	37.067	69.148
SBPL	MPPI	iter2	Test 8	1:38:52	24,37	Success	52.844	69.818	11.094	30.135	63.992
SBPL	NLOPT	iter1	Test 8	1:36:28	26,35	Success	53.198	77.129	11.392	33.456	67.897
SBPL	MPPI	iter3	Test 8	1:40:40	24,36	Success	53.792	69.407	11.200	33.460	64.996
SBPL	NLOPT	iter2	Test 8	1:34:32	25,36	Success	51.852	79.011	11.060	29.319	69.234
SBPL	NLOPT	iter3	Test 8	1:33:20	26,36	Success	48.724	79.884	10.830	30.390	69.988
SBPL	MPPI	iter4	Test 8	1:15:44	16,14	Failed	50.491	79.848	10.874	32.848	68.719
SBPL	NLOPT	iter4	Test 8	1:25:44	9,39	Failed	53.436	80.861	10.816	40.372	69.557
SBPL	NLOPT	iter5	Test 8	1:41:08	26,36	Success	49.879	82.723	11.062	31.574	69.980
SLGP	MPPI	iter1	Test 8	3:10:12	23,29	Stuck	50.306	81.427	11.533	33.610	69.000
SLGP	NLOPT	iter1	Test 8	1:10:52	19,28	Failed	49.313	81.883	9.569	30.969	68.961
SLGP	MPPI	iter2	Test 8	1:55:04	23,29	Stuck	50.032	78.884	10.604	32.338	69.935
SLGP	NLOPT	iter2	Test 8	1:11:16	20,28	Failed	48.639	82.172	9.570	36.336	68.773
SLGP	MPPI	iter3	Test 8	3:39:52	25,28	Stuck	48.494	80.567	11.690	32.391	68.994
SLGP	NLOPT	iter3	Test 8	1:17:00	19,28	Failed	48.905	81.133	9.557	31.516	68.961
SLGP	MPPI	iter4	Test 8	2:15:32	23,28	Stuck	48.593	82.047	10.842	34.936	69.758
SLGP	NLOPT	iter4	Test 8	1:21:24	19,28	Failed	49.976	79.961	9.519	35.852	68.586
SLGP	MPPI	iter5	Test 8	1:50:48	25,28	Stuck	48.633	80.689	10.564	36.696	69.862
SLGP	NLOPT	iter5	Test 8	1:18:40	20,28	Failed	48.577	84.078	9.570	28.781	69.219

Table 5: Test set 2 with Policy 1

EASL	MPPI	iter1	Test 1	1:23:48	26,36	Success	50.313	81.270	9.865	30.774	69.584
EASL	NLOPT	iter1	Test 1	1:16:32	26,36	Success	54.077	79.044	9.508	29.321	70.907
EASL	MPPI	iter2	Test 1	1:27:08	26,36	Success	53.607	77.116	9.931	28.741	68.592
EASL	NLOPT	iter2	Test 1	1:19:24	26,36	Success	53.593	78.715	9.431	29.367	70.961
EASL	MPPI	iter3	Test 1	1:33:08	26,37	Success	53.973	76.323	9.887	34.046	69.611
EASL	NLOPT	iter3	Test 1	1:24:48	25,36	Success	53.710	77.272	9.452	31.703	69.743
EASL	MPPI	iter4	Test 1	1:36:40	26,36	Success	53.810	75.992	9.883	30.051	68.977
EASL	NLOPT	iter4	Test 1	1:23:20	26,36	Success	54.937	75.997	9.458	26.211	69.809
EASL	MPPI	iter5	Test 1	1:33:24	26,37	Success	54.220	75.721	9.887	33.751	69.727
EASL	NLOPT	iter5	Test 1	1:23:32	26,36	Success	54.343	76.308	9.352	27.561	69.838
GLS	MPPI	iter1	Test 1	1:15:00	25,36	Success	59.598	77.206	8.963	53.835	68.458
GLS	NLOPT	iter1	Test 1	1:13:32	-0,04	Stuck	50.144	72.762	8.368	48.613	67.609
GLS	MPPI	iter2	Test 1	1:15:00	25,36	Success	59.138	75.952	9.028	53.911	68.573
GLS	NLOPT	iter2	Test 1	1:14:12	25,36	Success	60.718	76.734	8.521	59.174	68.789
GLS	MPPI	iter3	Test 1	1:14:52	25,36	Success	59.671	75.130	8.969	54.383	68.432
GLS	NLOPT	iter3	Test 1	1:17:40	25,36	Success	61.073	75.582	8.547	54.963	68.322
GLS	MPPI	iter4	Test 1	1:23:32	25,36	Success	60.074	74.538	9.006	55.444	68.825
GLS	NLOPT	iter4	Test 1	1:13:52	25,36	Success	60.729	75.814	8.519	58.883	67.584
GLS	MPPI	iter5	Test 1	1:15:08	25,36	Success	59.604	75.907	8.957	53.423	68.895
GLS	NLOPT	iter5	Test 1	1:07:00	25,36	Success	59.376	72.436	8.525	28.293	65.717
SBPL	MPPI	iter1	Test 1	1:22:44	25,36	Success	52.270	74.261	10.129	20.618	66.614
SBPL	NLOPT	iter1	Test 1	1:20:00	26,36	Success	55.369	75.356	9.483	28.220	66.901
SBPL	MPPI	iter2	Test 1	1:26:36	25,36	Success	52.183	73.260	9.854	27.106	66.775
SBPL	NLOPT	iter2	Test 1	1:18:48	26,36	Success	56.564	73.029	8.856	25.767	66.436

SBPL	MPPI	iter3	Test 1	1:28:44	26,36	Success	53.743	73.399	9.863	29.354	66.709
SBPL	NLOPT	iter3	Test 1	1:16:36	25,36	Success	55.122	72.261	8.980	27.190	66.647
SBPL	MPPI	iter4	Test 1	1:24:20	26,36	Success	53.537	73.339	9.643	23.678	66.459
SBPL	NLOPT	iter4	Test 1	1:30:16	26,36	Success	55.603	76.321	8.928	55.416	67.974
SBPL	MPPI	iter5	Test 1	1:27:16	25,36	Success	55.147	73.133	9.930	34.834	66.012
SBPL	NLOPT	iter5	Test 1	1:26:00	26,36	Success	57.243	74.384	9.151	60.455	67.948
SLGP	MPPI	iter1	Test 1	1:24:16	25,36	Success	49.405	83.709	9.026	27.613	71.785
SLGP	NLOPT	iter1	Test 1	1:24:36	25,36	Success	49.104	83.281	8.588	35.211	70.258
SLGP	MPPI	iter2	Test 1	1:30:56	26,37	Success	51.869	79.217	9.138	25.229	69.682
SLGP	NLOPT	iter2	Test 1	1:24:36	25,36	Success	48.640	84.625	8.637	26.719	70.723
SLGP	MPPI	iter3	Test 1	1:27:20	24,36	Success	47.629	87.023	9.018	27.521	71.049
SLGP	NLOPT	iter3	Test 1	1:18:20	25,36	Success	53.620	79.602	8.650	27.109	70.684
SLGP	MPPI	iter4	Test 1	1:19:24	25,36	Success	48.534	85.977	9.021	27.336	70.924
SLGP	NLOPT	iter4	Test 1	1:22:16	25,36	Success	51.956	77.703	8.604	35.680	70.660
SLGP	MPPI	iter5	Test 1	1:32:20	25,36	Success	48.206	82.230	9.018	32.408	70.861
SLGP	NLOPT	iter5	Test 1	1:13:56	25,36	Success	49.690	83.891	8.686	31.246	70.723
EASL	MPPI	iter1	Test 2	1:39:32	24,37	Success	53.075	76.178	10.020	32.293	68.868
EASL	NLOPT	iter1	Test 2	1:30:16	24,37	Success	54.204	76.730	9.573	34.362	69.736
EASL	MPPI	iter2	Test 2	1:40:00	24,37	Success	50.098	81.716	9.983	38.530	68.923
EASL	NLOPT	iter2	Test 2	1:23:08	25,37	Success	50.925	79.908	9.488	36.904	69.188
EASL	MPPI	iter3	Test 2	1:45:00	26,37	Success	51.051	82.361	9.954	31.019	69.898
EASL	NLOPT	iter3	Test 2	1:34:00	25,37	Success	50.830	80.387	9.494	34.356	69.096
EASL	MPPI	iter4	Test 2	1:34:40	24,37	Success	53.085	81.545	9.983	32.359	68.341
EASL	NLOPT	iter4	Test 2	1:34:24	25,37	Success	52.901	81.222	9.541	33.422	69.096
EASL	MPPI	iter5	Test 2	1:44:56	24,37	Success	53.427	75.242	9.945	24.193	70.271

EASL	NLOPT	iter5	Test 2	1:30:00	26,38	Success	53.431	78.222	9.472	32.184	68.719
GLS	MPPI	iter1	Test 2	2:44:00	24,2	Stuck	57.475	81.359	10.433	33.661	70.125
GLS	NLOPT	iter1	Test 2	1:37:00	8,33	Failed	59.953	79.806	8.918	35.827	70.960
GLS	MPPI	iter2	Test 2	1:59:32	24,2	Stuck	56.330	81.545	9.709	33.682	70.094
GLS	NLOPT	iter2	Test 2	1:25:44	18,57	Failed	62.642	77.970	8.843	30.392	69.811
GLS	MPPI	iter3	Test 2	1:27:48	4,5	Stuck	55.683	81.035	9.247	40.243	70.869
GLS	NLOPT	iter3	Test 2	1:59:52	27,57	Failed	62.559	80.040	9.253	36.339	69.937
GLS	MPPI	iter4	Test 2	1:26:20	7,3	Stuck	57.786	79.102	9.078	31.096	71.166
GLS	NLOPT	iter4	Test 2	1:17:48	8,3	Failed	58.238	83.914	8.570	28.180	70.453
GLS	MPPI	iter5	Test 2	1:26:00	7,3	Stuck	55.823	84.658	9.235	35.063	70.900
GLS	NLOPT	iter5	Test 2	2:06:20	24,37	Success	56.177	83.320	9.264	37.948	70.937
SBPL	MPPI	iter1	Test 2	1:43:08	25,36	Success	49.674	82.507	10.187	29.423	70.983
SBPL	NLOPT	iter1	Test 2	1:30:40	24,37	Success	52.915	76.133	10.151	28.434	69.817
SBPL	MPPI	iter2	Test 2	1:30:40	24,37	Success	49.948	81.820	10.479	37.945	70.960
SBPL	NLOPT	iter2	Test 2	1:35:16	25,36	Success	53.698	78.341	10.028	34.002	70.316
SBPL	MPPI	iter3	Test 2	1:30:23	26,36	Success	48.700	82.898	12.874	31.072	70.094
SBPL	NLOPT	iter3	Test 2	1:34:00	25,36	Success	54.639	76.047	9.764	28.900	69.733
SBPL	MPPI	iter4	Test 2	1:22:20	-0,01	Stuck	49.761	78.078	8.840	28.816	70.660
SBPL	NLOPT	iter4	Test 2	1:24:04	26,36	Success	53.224	76.055	10.150	31.373	69.875
SBPL	MPPI	iter5	Test 2	1:38:48	25,36	Success	52.045	79.007	10.662	35.579	68.358
SBPL	NLOPT	iter5	Test 2	1:33:44	24,37	Success	49.833	82.715	9.957	38.928	70.958
SLGP	MPPI	iter1	Test 2	1:36:00	24,37	Success	47.223	82.690	9.188	36.463	70.936
SLGP	NLOPT	iter1	Test 2	1:12:04	5,8	Failed	52.408	82.996	8.648	32.852	70.203
SLGP	MPPI	iter2	Test 2	1:42:28	24,37	Success	50.482	79.937	9.188	36.827	68.917
SLGP	NLOPT	iter2	Test 2	1:16:48	5,15	Failed	53.606	82.098	8.638	31.484	70.664

SLGP	MPPI	iter3	Test 2	1:36:40	24,37	Success	47.732	82.306	9.232	32.806	70.983
SLGP	NLOPT	iter3	Test 2	1:23:28	5,15	Failed	53.920	86.391	8.634	32.074	70.723
SLGP	MPPI	iter4	Test 2	1:36:20	24,37	Success	47.920	82.248	9.184	36.318	70.612
SLGP	NLOPT	iter4	Test 2	1:08:08	5,8	Failed	52.266	83.281	8.281	33.063	68.625
SLGP	MPPI	iter5	Test 2	1:40:20	24,37	Success	47.557	84.037	9.232	35.123	70.049
SLGP	NLOPT	iter5	Test 2	1:07:36	5,7	Failed	51.869	82.688	8.334	30.875	68.375
EASL	MPPI	iter1	Test 3	1:41:08	26,36	Success	48.594	81.992	10.043	34.959	68.885
EASL	NLOPT	iter1	Test 3	1:27:40	26,36	Success	53.921	76.230	9.489	30.977	69.477
EASL	MPPI	iter2	Test 3	1:38:48	26,36	Success	48.989	81.897	9.967	45.078	68.919
EASL	NLOPT	iter2	Test 3	1:24:28	26,38	Success	53.195	77.685	9.543	30.534	68.600
EASL	MPPI	iter3	Test 3	1:43:28	26,36	Success	53.001	75.699	9.973	31.803	68.373
EASL	NLOPT	iter3	Test 3	1:12:20	27,36	Success	59.310	80.750	9.272	36.016	71.078
EASL	MPPI	iter4	Test 3	1:38:40	25,38	Success	50.427	82.127	9.973	36.748	69.463
EASL	NLOPT	iter4	Test 3	1:36:24	26,36	Success	61.796	82.800	9.591	30.877	70.933
EASL	MPPI	iter5	Test 3	1:45:52	26,36	Success	48.935	81.341	9.990	30.440	69.880
EASL	NLOPT	iter5	Test 3	1:03:20	6,2	Failed	71.128	80.906	9.213	34.266	69.156
GLS	MPPI	iter1	Test 3	2:07:00	25,36	Success	54.629	82.338	9.642	45.348	70.250
GLS	NLOPT	iter1	Test 3	1:33:04	25,36	Success	55.711	82.115	8.783	30.284	70.969
GLS	MPPI	iter2	Test 3	1:20:40	-1,03	Stuck	45.622	84.719	8.774	29.738	70.715
GLS	NLOPT	iter2	Test 3	1:26:16	26,35	Success	55.432	81.758	8.758	33.988	70.736
GLS	MPPI	iter3	Test 3	1:42:40	2,2	Stuck	57.724	80.177	9.466	34.792	68.826
GLS	NLOPT	iter3	Test 3	1:09:24	6,3	Failed	56.517	82.781	8.511	34.188	69.656
GLS	MPPI	iter4	Test 3	1:54:52	28,36	Success	56.817	82.738	9.440	30.214	68.920
GLS	NLOPT	iter4	Test 3	1:39:48	26,36	Success	57.716	82.313	8.865	30.178	70.981
GLS	MPPI	iter5	Test 3	1:44:04	8,3	Stuck	56.589	81.389	9.392	35.835	69.521
GLS	NLOPT	iter5	Test 3	1:33:00	26,36	Success	55.181	82.701	8.865	33.203	70.980

SBPL	MPPI	iter1	Test 3	1:41:52	26,36	Success	48.522	83.094	10.251	31.813	70.432
SBPL	NLOPT	iter1	Test 3	1:27:44	26,35	Success	52.270	84.841	9.265	26.574	70.904
SBPL	MPPI	iter2	Test 3	1:38:20	26,36	Success	49.408	83.546	10.456	30.227	70.983
SBPL	NLOPT	iter2	Test 3	1:48:16	26,36	Success	50.917	81.647	9.311	29.953	70.995
SBPL	MPPI	iter3	Test 3	1:39:36	26,36	Success	48.649	81.275	10.086	34.205	70.272
SBPL	NLOPT	iter3	Test 3	1:39:00	26,35	Success	52.434	82.497	9.344	34.043	70.981
SBPL	MPPI	iter4	Test 3	1:14:00	-0,02	Stuck	45.038	86.668	8.914	27.625	70.715
SBPL	NLOPT	iter4	Test 3	1:38:44	26,36	Success	52.546	85.386	9.376	32.987	70.863
SBPL	MPPI	iter5	Test 3	1:42:20	26,36	Success	48.314	86.690	10.142	28.272	70.983
SBPL	NLOPT	iter5	Test 3	1:35:00	26,36	Success	52.546	83.940	9.680	27.670	70.802
SLGP	MPPI	iter1	Test 3	1:40:04	26,36	Success	48.405	83.982	9.261	31.226	68.855
SLGP	NLOPT	iter1	Test 3	1:09:32	3,3	Failed	51.706	86.000	8.044	33.625	70.250
SLGP	MPPI	iter2	Test 3	1:41:48	26,35	Success	47.324	82.685	9.184	36.532	70.991
SLGP	NLOPT	iter2	Test 3	0:57:40	2,3	Failed	52.353	79.250	8.331	31.094	69.813
SLGP	MPPI	iter3	Test 3	2:08:00	26,35	Success	48.314	83.265	9.494	27.718	70.508
SLGP	NLOPT	iter3	Test 3	1:00:48	2,3	Failed	52.553	83.750	8.281	33.219	70.781
SLGP	MPPI	iter4	Test 3	2:06:04	27,36	Success	48.102	82.784	9.384	25.252	70.999
SLGP	NLOPT	iter4	Test 3	1:03:40	3,3	Failed	51.166	79.719	8.306	29.938	68.781
SLGP	MPPI	iter5	Test 3	2:49:44	25,36	Success	48.153	82.450	9.889	27.578	70.139
SLGP	NLOPT	iter5	Test 3	1:09:04	2,3	Failed	51.738	80.531	8.234	35.438	68.500
EASL	MPPI	iter1	Test 4	1:38:32	24,37	Success	70.321	79.796	10.142	24.669	70.327
EASL	NLOPT	iter1	Test 4	1:07:24	5,3	Failed	54.731	78.148	9.401	31.609	70.445
EASL	MPPI	iter2	Test 4	1:12:52	-1,04	Stuck	46.296	79.824	9.675	28.637	70.164
EASL	NLOPT	iter2	Test 4	0:54:12	6,3	Failed	59.698	80.695	9.282	31.672	70.461
EASL	MPPI	iter3	Test 4	1:40:36	26,38	Success	72.072	83.211	10.148	29.008	70.994

EASL	NLOPT	iter3	Test 4	1:31:12	26,37	Success	79.221	83.427	9.773	27.341	70.999
EASL	MPPI	iter4	Test 4	1:34:00	25,37	Success	70.044	81.177	10.135	34.154	70.013
EASL	NLOPT	iter4	Test 4	0:56:16	5,1	Failed	64.206	78.859	9.277	24.906	68.922
EASL	MPPI	iter5	Test 4	1:32:36	25,37	Success	67.738	83.278	10.130	28.394	70.187
EASL	NLOPT	iter5	Test 4	1:29:20	25,37	Success	88.426	81.961	9.867	26.581	70.066
GLS	MPPI	iter1	Test 4	1:28:40	27,38	Success	73.770	83.053	9.469	33.074	70.920
GLS	NLOPT	iter1	Test 4	1:12:24	3,2	Failed	56.395	79.367	8.798	26.109	69.578
GLS	MPPI	iter2	Test 4	1:39:16	15,14	Stuck	52.225	78.759	9.480	38.825	70.998
GLS	NLOPT	iter2	Test 4	1:12:04	7,2	Failed	56.311	79.781	8.698	33.063	69.453
GLS	MPPI	iter3	Test 4	2:21:56	25,37	Success	72.106	82.016	10.151	32.931	70.769
GLS	NLOPT	iter3	Test 4	1:09:36	6,3	Failed	58.273	79.316	8.748	30.707	69.727
GLS	MPPI	iter4	Test 4	2:00:12	25,37	Success	72.492	78.779	9.743	40.401	69.313
GLS	NLOPT	iter4	Test 4	1:04:08	7,2	Failed	54.069	76.906	8.403	32.875	68.063
GLS	MPPI	iter5	Test 4	1:06:40	-1,05	Stuck	53.930	79.512	9.216	30.922	69.727
GLS	NLOPT	iter5	Test 4	1:29:28	12,57	Failed	57.693	78.249	9.217	32.850	70.174
SBPL	MPPI	iter1	Test 4	1:36:20	24,37	Success	68.661	82.747	10.257	41.663	70.079
SBPL	NLOPT	iter1	Test 4	1:34:16	24,36	Success	69.811	82.705	10.217	27.430	69.991
SBPL	MPPI	iter2	Test 4	1:30:28	24,37	Success	68.788	82.811	10.340	30.447	70.241
SBPL	NLOPT	iter2	Test 4	1:22:08	24,37	Success	71.284	81.426	10.431	30.825	70.840
SBPL	MPPI	iter3	Test 4	1:30:48	23,38	Success	69.369	80.991	10.643	30.498	69.991
SBPL	NLOPT	iter3	Test 4	1:41:20	25,37	Success	69.544	82.154	10.409	31.402	69.996
SBPL	MPPI	iter4	Test 4	1:28:20	23,37	Success	68.922	82.160	10.373	30.521	69.991
SBPL	NLOPT	iter4	Test 4	1:31:20	25,36	Success	69.688	83.556	10.352	29.293	69.966
SBPL	MPPI	iter5	Test 4	1:38:20	24,37	Success	69.346	81.814	10.309	26.777	69.991
SBPL	NLOPT	iter5	Test 4	1:38:04	25,36	Success	70.571	81.834	10.179	31.792	69.991

SLGP	MPPI	iter1	Test 4	1:51:20	25,37	Success	69.612	82.513	9.558	47.522	70.979
SLGP	NLOPT	iter1	Test 4	1:07:32	4,3	Failed	50.775	76.938	8.419	28.594	68.063
SLGP	MPPI	iter2	Test 4	1:27:20	24,37	Success	68.468	84.172	9.441	27.054	70.085
SLGP	NLOPT	iter2	Test 4	1:03:28	3,3	Failed	49.419	77.000	8.475	34.906	67.813
SLGP	MPPI	iter3	Test 4	1:31:20	24,37	Success	68.145	81.234	9.366	30.893	70.029
SLGP	NLOPT	iter3	Test 4	1:02:24	4,3	Failed	50.222	77.969	8.469	32.469	67.813
SLGP	MPPI	iter4	Test 4	1:38:16	25,37	Success	68.720	85.012	9.366	25.759	70.679
SLGP	NLOPT	iter4	Test 4	1:02:56	3,3	Failed	49.563	76.875	8.469	26.625	68.313
SLGP	MPPI	iter5	Test 4	1:16:16	-1,04	Stuck	45.482	78.625	8.966	26.520	70.477
SLGP	NLOPT	iter5	Test 4	1:00:16	3,3	Failed	50.559	76.375	8.491	27.438	68.313
EASL	MPPI	iter1	Test 5	0:55:08	-1,08	Stuck	56.295	81.375	9.859	24.504	69.727
EASL	NLOPT	iter1	Test 5	2:17:28	11,2	Failed	61.995	82.048	10.723	31.779	69.992
EASL	MPPI	iter2	Test 5	2:44:40	28,26	Stuck	60.162	80.833	11.417	29.426	69.844
EASL	NLOPT	iter2	Test 5	0:47:44	-0,06	Failed	64.054	81.516	9.515	36.945	69.570
EASL	MPPI	iter3	Test 5	2:23:28	27,25	Stuck	60.819	82.347	11.205	22.946	69.066
EASL	NLOPT	iter3	Test 5	1:48:28	22,17	Failed	68.890	81.718	10.632	28.847	69.031
EASL	MPPI	iter4	Test 5	2:43:40	24,25	Stuck	60.918	81.916	11.330	28.954	69.515
EASL	NLOPT	iter4	Test 5	0:37:00	-0,10	Failed	67.772	82.203	9.511	21.969	69.406
EASL	MPPI	iter5	Test 5	4:33:00	27,58	Stuck	61.821	82.452	12.643	24.106	69.039
EASL	NLOPT	iter5	Test 5	0:48:28	-0,09	Failed	67.753	81.344	9.611	21.766	68.938
GLS	MPPI	iter1	Test 5	1:49:20	4,27	Stuck	60.096	81.342	10.078	29.932	69.998
GLS	NLOPT	iter1	Test 5	1:02:44	8,3	Failed	59.841	82.156	8.805	28.875	69.195
GLS	MPPI	iter2	Test 5	1:11:20	5,3	Stuck	61.173	81.833	9.472	23.825	69.982
GLS	NLOPT	iter2	Test 5	1:09:12	-1,07	Stuck	56.643	81.953	8.766	24.477	69.715
GLS	MPPI	iter3	Test 5	1:01:20	-1,08	Stuck	60.250	82.125	9.270	30.020	69.723

GLS	NLOPT	iter3	Test 5	1:04:44	-1,07	Stuck	56.125	81.680	8.793	23.488	69.602
GLS	MPPI	iter4	Test 5	1:06:24	7,2	Stuck	60.240	82.420	9.385	26.447	69.983
GLS	NLOPT	iter4	Test 5	0:55:04	-0,07	Failed	58.469	81.078	8.744	23.234	68.891
GLS	MPPI	iter5	Test 5	0:53:16	-1,08	Stuck	60.361	82.273	9.332	26.004	69.727
GLS	NLOPT	iter5	Test 5	1:29:28	16,56	Failed	59.881	82.495	9.284	27.767	69.999
SBPL	MPPI	iter1	Test 5	2:17:00	26,18	Failed	58.915	82.675	11.592	35.121	69.500
SBPL	NLOPT	iter1	Test 5	2:12:12	29,2	Failed	61.837	82.365	11.121	27.768	70.000
SBPL	MPPI	iter2	Test 5	2:39:20	28,25	Stuck	57.997	81.995	12.265	34.389	69.977
SBPL	NLOPT	iter2	Test 5	1:03:12	-1,07	Stuck	55.934	82.633	8.787	23.547	69.770
SBPL	MPPI	iter3	Test 5	1:08:36	-1,08	Stuck	55.921	82.219	9.095	26.516	69.719
SBPL	NLOPT	iter3	Test 5	2:28:48	11,58	Failed	61.636	81.798	10.970	31.305	70.000
SBPL	MPPI	iter4	Test 5	1:02:40	-1,07	Stuck	57.006	82.426	9.182	31.684	69.867
SBPL	NLOPT	iter4	Test 5	1:24:04	5,27	Failed	60.379	82.178	10.496	41.977	69.999
SBPL	MPPI	iter5	Test 5	5:05:00	25,1	Failed	58.518	80.776	14.072	27.548	69.266
SBPL	NLOPT	iter5	Test 5	2:13:24	23,25	Failed	59.125	82.174	10.896	21.139	69.937
SLGP	MPPI	iter1	Test 5	1:40:36	15,36	Stuck	57.265	81.655	9.841	37.011	69.999
SLGP	NLOPT	iter1	Test 5	1:06:00	5,14	Failed	58.140	82.812	8.963	32.479	69.822
SLGP	MPPI	iter2	Test 5	1:51:08	15,34	Stuck	56.938	81.865	9.721	22.710	69.875
SLGP	NLOPT	iter2	Test 5	1:03:56	5,4	Failed	57.657	82.305	8.806	23.586	69.445
SLGP	MPPI	iter3	Test 5	1:41:00	15,36	Stuck	57.500	81.290	9.752	25.814	70.000
SLGP	NLOPT	iter3	Test 5	0:56:28	-1,08	Stuck	56.691	80.977	8.766	23.633	69.676
SLGP	MPPI	iter4	Test 5	2:04:48	15,35	Stuck	57.168	81.726	10.139	31.166	69.187
SLGP	NLOPT	iter4	Test 5	0:59:32	5,4	Failed	57.813	80.750	8.855	30.859	69.430
SLGP	MPPI	iter5	Test 5	2:23:16	15,35	Stuck	57.133	81.750	10.261	26.939	69.003
SLGP	NLOPT	iter5	Test 5	0:57:16	6,4	Failed	58.226	81.492	8.793	29.953	69.453

EASL	MPPI	iter1	Test 6	1:21:36	-1,10	Stuck	49.206	79.520	10.004	24.488	69.730
EASL	NLOPT	iter1	Test 6	1:42:56	25,36	Success	99.739	82.484	10.372	29.918	69.188
EASL	MPPI	iter2	Test 6	1:52:00	24,37	Success	99.303	82.477	10.642	32.135	69.996
EASL	NLOPT	iter2	Test 6	0:52:08	-0,11	Failed	48.144	77.750	9.697	25.594	68.906
EASL	MPPI	iter3	Test 6	1:41:28	25,37	Success	53.278	71.800	10.867	31.296	65.254
EASL	NLOPT	iter3	Test 6	1:51:16	26,36	Success	99.808	82.455	10.277	39.751	69.438
EASL	MPPI	iter4	Test 6	1:58:08	24,37	Success	99.514	81.984	11.081	21.190	69.388
EASL	NLOPT	iter4	Test 6	1:47:28	27,37	Success	99.599	82.451	10.190	21.623	69.031
EASL	MPPI	iter5	Test 6	1:49:24	24,37	Success	48.890	80.850	10.971	30.962	68.667
EASL	NLOPT	iter5	Test 6	1:43:40	25,36	Success	99.800	82.246	10.248	24.395	69.344
GLS	MPPI	iter1	Test 6	1:56:24	4,6	Stuck	49.701	79.559	10.340	23.435	69.000
GLS	NLOPT	iter1	Test 6	1:02:52	-0,15	Failed	49.664	78.934	9.380	27.495	69.964
GLS	MPPI	iter2	Test 6	0:46:24	-1,11	Stuck	49.389	79.277	9.516	25.332	69.730
GLS	NLOPT	iter2	Test 6	0:53:00	8,2	Failed	49.406	79.019	9.337	23.092	69.983
GLS	MPPI	iter3	Test 6	1:07:04	8,3	Stuck	49.698	79.565	9.672	23.553	69.724
GLS	NLOPT	iter3	Test 6	0:44:40	-0,08	Failed	48.030	78.250	8.939	32.906	68.906
GLS	MPPI	iter4	Test 6	1:08:48	7,5	Stuck	49.697	79.463	9.694	23.232	69.984
GLS	NLOPT	iter4	Test 6	1:01:56	7,2	Stuck	49.619	79.498	9.368	21.642	69.374
GLS	MPPI	iter5	Test 6	1:45:44	25,36	Success	99.720	81.453	10.273	23.756	69.640
GLS	NLOPT	iter5	Test 6	0:47:40	-0,10	Stuck	48.942	79.262	9.176	22.309	69.227
SBPL	MPPI	iter1	Test 6	1:42:40	26,36	Success	99.374	82.585	11.961	28.581	69.008
SBPL	NLOPT	iter1	Test 6	0:56:16	-1,11	Stuck	49.251	78.879	9.835	27.711	69.051
SBPL	MPPI	iter2	Test 6	0:45:40	-1,10	Stuck	49.117	78.547	9.509	23.273	69.164
SBPL	NLOPT	iter2	Test 6	0:48:40	-0,09	Failed	48.225	78.313	9.556	33.828	68.922
SBPL	MPPI	iter3	Test 6	1:59:56	26,36	Success	99.689	82.409	11.723	23.636	69.000

SBPL	NLOPT	iter3	Test 6	0:40:00	-0,08	Failed	49.574	79.619	9.524	22.723	69.740
SBPL	MPPI	iter4	Test 6	1:27:08	24,37	Success	99.364	82.232	11.663	23.498	69.004
SBPL	NLOPT	iter4	Test 6	0:50:32	-1,09	Failed	48.198	78.266	9.241	23.063	68.781
SBPL	MPPI	iter5	Test 6	0:52:16	-1,11	Stuck	49.343	79.941	9.414	31.816	68.977
SBPL	NLOPT	iter5	Test 6	1:42:48	25,36	Success	99.731	81.992	11.319	22.273	68.969
SLGP	MPPI	iter1	Test 6	1:44:20	25,36	Success	99.473	82.481	9.970	36.694	69.086
SLGP	NLOPT	iter1	Test 6	1:01:28	8,9	Failed	49.289	79.210	9.180	22.522	69.966
SLGP	MPPI	iter2	Test 6	1:37:40	24,37	Success	99.583	81.891	10.142	31.239	69.045
SLGP	NLOPT	iter2	Test 6	1:37:44	25,36	Success	99.783	82.000	9.586	20.393	69.281
SLGP	MPPI	iter3	Test 6	1:24:04	15,22	Stuck	49.559	78.688	9.959	24.634	69.035
SLGP	NLOPT	iter3	Test 6	1:16:24	14,19	Failed	49.240	79.578	9.335	35.215	69.937
SLGP	MPPI	iter4	Test 6	1:46:40	24,38	Success	99.601	81.994	10.089	26.486	69.993
SLGP	NLOPT	iter4	Test 6	0:59:08	8,8	Failed	49.378	78.782	9.229	21.393	69.935
SLGP	MPPI	iter5	Test 6	1:38:00	25,37	Success	99.496	82.468	9.953	21.113	69.064
SLGP	NLOPT	iter5	Test 6	1:01:24	8,9	Failed	49.443	79.461	9.233	27.489	69.964
EASL	MPPI	iter1	Test 7	3:03:36	26,35	Stuck	61.037	81.378	10.830	38.492	69.965
EASL	NLOPT	iter1	Test 7	2:21:40	26,38	Success	48.878	82.865	10.021	31.300	68.926
EASL	MPPI	iter2	Test 7	2:29:12	26,35	Success	47.357	84.326	10.773	31.772	70.000
EASL	NLOPT	iter2	Test 7	1:38:12	27,38	Success	65.985	79.924	9.839	29.115	69.298
EASL	MPPI	iter3	Test 7	2:38:40	26,36	Success	48.556	83.733	10.881	27.389	69.814
EASL	NLOPT	iter3	Test 7	1:37:04	21,37	Failed	68.854	82.264	9.766	32.073	69.554
EASL	MPPI	iter4	Test 7	2:34:20	26,38	Success	48.826	84.673	10.692	37.763	69.936
EASL	NLOPT	iter4	Test 7	2:23:24	26,36	Success	70.038	76.627	10.148	29.461	68.148
EASL	MPPI	iter5	Test 7	2:35:04	26,35	Success	48.449	85.717	10.706	32.582	68.987
EASL	NLOPT	iter5	Test 7	1:46:04	21,37	Stuck	79.691	81.956	10.066	25.260	70.001

GLS	MPPI	iter1	Test 7	2:09:28	23,8	Stuck	59.147	76.627	9.818	28.675	70.781
GLS	NLOPT	iter1	Test 7	1:18:16	-1,03	Stuck	45.541	84.547	8.467	34.566	69.758
GLS	MPPI	iter2	Test 7	1:21:40	-1,04	Stuck	45.713	83.254	8.778	30.836	69.727
GLS	NLOPT	iter2	Test 7	2:24:36	26,36	Success	56.368	83.480	9.550	29.247	69.972
GLS	MPPI	iter3	Test 7	2:04:16	19,17	Stuck	57.363	82.254	9.814	37.528	69.999
GLS	NLOPT	iter3	Test 7	1:16:04	-0,03	Stuck	46.609	85.688	8.478	33.211	69.852
GLS	MPPI	iter4	Test 7	3:08:36	19,33	Stuck	58.719	79.840	10.867	39.618	68.651
GLS	NLOPT	iter4	Test 7	2:11:48	26,36	Success	60.125	74.661	9.624	24.453	68.197
GLS	MPPI	iter5	Test 7	2:13:16	26,36	Success	54.380	83.523	9.661	38.128	69.248
GLS	NLOPT	iter5	Test 7	2:04:48	19,35	Stuck	47.398	80.488	9.398	30.015	68.952
SBPL	MPPI	iter1	Test 7	1:49:28	26,36	Success	59.480	78.558	10.568	26.360	70.218
SBPL	NLOPT	iter1	Test 7	2:07:24	25,36	Success	54.760	73.237	11.281	31.860	65.160
SBPL	MPPI	iter2	Test 7	2:06:04	27,38	Success	52.391	74.131	11.049	35.123	67.000
SBPL	NLOPT	iter2	Test 7	1:49:08	25,36	Success	50.178	84.848	10.098	33.174	69.997
SBPL	MPPI	iter3	Test 7	1:48:48	26,36	Success	53.814	73.051	10.528	30.096	65.995
SBPL	NLOPT	iter3	Test 7	1:42:12	26,36	Success	49.687	81.283	10.520	32.091	69.532
SBPL	MPPI	iter4	Test 7	3:05:08	26,36	Success	54.553	74.026	11.700	28.769	66.514
SBPL	NLOPT	iter4	Test 7	2:18:00	21,30	Failed	60.257	76.883	11.438	29.791	68.868
SBPL	MPPI	iter5	Test 7	1:22:52	27,36	Success	53.361	74.017	11.508	50.875	68.000
SBPL	NLOPT	iter5	Test 7	1:22:48	21,36	Failed	56.299	84.212	9.881	26.972	69.932
SLGP	MPPI	iter1	Test 7	1:54:04	20,36	Stuck	47.469	82.552	9.685	26.013	70.145
SLGP	NLOPT	iter1	Test 7	1:30:40	21,35	Failed	49.811	83.326	8.998	32.693	69.932
SLGP	MPPI	iter2	Test 7	2:00:36	20,37	Stuck	48.166	84.466	9.789	32.721	70.006
SLGP	NLOPT	iter2	Test 7	1:17:08	-1,04	Stuck	46.120	84.270	8.487	26.027	69.727
SLGP	MPPI	iter3	Test 7	1:52:08	20,36	Stuck	48.847	84.661	9.482	38.293	70.121

SLGP	NLOPT	iter3	Test 7	1:24:28	21,31	Failed	51.054	84.639	8.829	35.197	70.367
SLGP	MPPI	iter4	Test 7	1:57:32	20,37	Stuck	47.327	83.795	9.682	24.110	70.842
SLGP	NLOPT	iter4	Test 7	1:29:00	21,30	Failed	52.257	83.422	8.796	29.096	71.111
SLGP	MPPI	iter5	Test 7	2:13:36	20,36	Stuck	47.648	86.736	9.842	31.021	70.031
SLGP	NLOPT	iter5	Test 7	1:21:52	21,31	Failed	50.980	81.672	8.853	33.197	70.176
EASL	MPPI	iter1	Test 8	1:34:44	25,37	Success	79.150	81.636	10.568	30.994	70.033
EASL	NLOPT	iter1	Test 8	1:35:16	27,37	Success	80.002	83.145	10.078	32.538	70.016
EASL	MPPI	iter2	Test 8	1:40:56	24,37	Success	78.521	82.522	10.471	28.112	70.000
EASL	NLOPT	iter2	Test 8	1:40:12	26,37	Success	79.206	83.014	10.098	32.325	70.750
EASL	MPPI	iter3	Test 8	1:33:56	24,37	Success	79.449	81.936	10.485	33.969	70.265
EASL	NLOPT	iter3	Test 8	1:49:16	25,37	Success	78.711	83.290	10.082	30.243	70.314
EASL	MPPI	iter4	Test 8	1:48:24	24,37	Success	79.021	81.960	10.517	44.099	70.008
EASL	NLOPT	iter4	Test 8	1:33:00	25,37	Success	80.102	82.828	10.047	34.661	70.105
EASL	MPPI	iter5	Test 8	1:40:04	26,37	Success	79.538	81.996	10.485	27.561	70.004
EASL	NLOPT	iter5	Test 8	1:37:08	25,37	Success	79.997	82.670	10.114	32.608	70.287
GLS	MPPI	iter1	Test 8	2:20:00	25,37	Success	83.190	82.313	10.532	27.153	70.000
GLS	NLOPT	iter1	Test 8	1:43:40	19,16	Stuck	65.128	79.454	9.729	32.986	69.750
GLS	MPPI	iter2	Test 8	1:51:24	24,37	Success	84.080	80.929	9.871	29.735	70.031
GLS	NLOPT	iter2	Test 8	1:34:20	17,57	Failed	65.033	80.175	9.429	31.868	69.998
GLS	MPPI	iter3	Test 8	1:26:44	27,36	Success	83.908	83.573	9.816	25.072	69.996
GLS	NLOPT	iter3	Test 8	2:02:24	6,32	Failed	63.328	79.419	9.740	31.172	69.875
GLS	MPPI	iter4	Test 8	2:15:32	24,38	Success	83.405	81.857	10.403	24.100	70.016
GLS	NLOPT	iter4	Test 8	1:01:28	-0,06	Stuck	56.584	79.242	8.855	29.434	69.727
GLS	MPPI	iter5	Test 8	2:29:48	24,37	Success	83.407	82.542	10.470	27.938	70.000
GLS	NLOPT	iter5	Test 8	1:02:44	-1,05	Stuck	57.245	80.430	8.780	24.871	69.734
SBPL	MPPI	iter1	Test 8	1:20:48	26,36	Success	77.210	80.563	9.209	25.363	69.715

SBPL	NLOPT	iter1	Test 8	1:31:08	25,36	Success	79.637	83.308	10.512	27.726	69.995
SBPL	MPPI	iter2	Test 8	1:40:00	25,37	Success	79.867	82.385	10.841	32.830	70.124
SBPL	NLOPT	iter2	Test 8	1:30:20	25,36	Success	83.254	82.232	10.684	31.410	69.998
SBPL	MPPI	iter3	Test 8	1:38:36	24,37	Success	79.396	82.051	10.780	29.466	69.905
SBPL	NLOPT	iter3	Test 8	1:30:20	25,36	Success	80.307	81.865	10.237	28.027	69.989
SBPL	MPPI	iter4	Test 8	1:43:40	24,37	Success	79.930	83.865	11.023	45.708	70.000
SBPL	NLOPT	iter4	Test 8	1:34:00	25,36	Success	79.889	81.945	10.160	30.847	69.996
SBPL	MPPI	iter5	Test 8	1:34:48	24,36	Success	80.108	82.607	10.644	32.158	69.999
SBPL	NLOPT	iter5	Test 8	1:04:12	-0,06	Stuck	57.700	80.191	8.780	27.402	69.727
SLGP	MPPI	iter1	Test 8	1:45:16	22,29	Stuck	58.663	79.355	9.871	36.202	69.999
SLGP	NLOPT	iter1	Test 8	1:19:28	20,29	Failed	60.928	79.521	9.124	30.292	69.932
SLGP	MPPI	iter2	Test 8	2:05:00	23,29	Stuck	58.540	78.110	10.183	30.736	70.000
SLGP	NLOPT	iter2	Test 8	1:14:36	21,29	Failed	62.414	78.881	9.104	39.843	69.932
SLGP	MPPI	iter3	Test 8	1:41:56	23,29	Stuck	59.246	79.977	10.015	30.901	69.968
SLGP	NLOPT	iter3	Test 8	1:09:04	-1,06	Stuck	58.064	79.816	8.866	29.305	69.711
SLGP	MPPI	iter4	Test 8	1:36:00	24,28	Stuck	59.351	79.070	9.964	31.180	70.000
SLGP	NLOPT	iter4	Test 8	1:16:44	20,29	Failed	60.267	79.235	9.212	24.865	69.966
SLGP	MPPI	iter5	Test 8	1:43:08	22,29	Stuck	58.115	79.040	10.067	36.395	70.000
SLGP	NLOPT	iter5	Test 8	1:20:24	21,29	Failed	60.431	80.142	9.124	48.098	69.932

Table 6: Test set 2 with Policy 2

Global Planner	Local Planner	Iter Num	Test Num	End time	End location	Status	CPU Util	CPU temp	RAM	GPU util	GPU temp
SLGP	MPPI	iter1	Test 1	1:17:20	25,36	Success	60.285	79.648	9.819	28.065	70.014
SLGP	MPPI	iter2	Test 1	1:17:08	25,36	Success	59.394	79.424	9.731	41.459	69.036
SLGP	MPPI	iter3	Test 1	1:21:44	25,36	Success	59.056	80.195	9.731	33.460	69.965
SLGP	MPPI	iter4	Test 1	1:19:20	25,36	Success	59.936	79.059	9.794	33.128	69.920
SLGP	MPPI	iter5	Test 1	1:21:28	25,36	Success	59.735	79.867	9.831	30.933	69.966
SLGP	MPPI	iter1	Test 2	1:40:44	24,37	Success	59.796	79.259	10.940	27.531	69.749
SLGP	MPPI	iter2	Test 2	1:36:08	24,37	Success	59.415	80.281	9.885	26.738	70.492
SLGP	MPPI	iter3	Test 2	1:12:20	-1,05	Stuck	57.951	79.816	9.465	31.223	70.223
SLGP	MPPI	iter4	Test 2	1:02:00	-0,06	Stuck	57.793	80.406	9.519	27.320	69.727
SLGP	MPPI	iter5	Test 2	1:28:00	24,37	Success	60.271	80.312	10.260	34.633	70.999
SLGP	MPPI	iter1	Test 3	1:34:28	26,36	Success	59.909	81.749	11.171	24.590	70.827
SLGP	MPPI	iter2	Test 3	1:42:28	26,36	Success	59.036	80.438	10.885	24.572	70.615
SLGP	MPPI	iter3	Test 3	1:53:40	26,36	Success	59.487	80.600	11.474	27.004	70.996
SLGP	MPPI	iter4	Test 3	1:52:48	25,36	Success	61.299	78.469	11.434	34.622	69.032
SLGP	MPPI	iter5	Test 3	1:09:08	8,3	Failed	61.976	80.375	10.317	32.129	69.832
SLGP	MPPI	iter1	Test 4	1:42:44	24,37	Success	61.933	79.125	11.230	30.900	69.998
SLGP	MPPI	iter2	Test 4	1:32:28	24,37	Success	61.135	81.875	11.441	35.272	68.500
SLGP	MPPI	iter3	Test 4	1:41:52	24,37	Success	59.979	79.020	10.071	30.039	69.001
SLGP	MPPI	iter4	Test 4	1:35:04	24,37	Success	61.583	81.250	11.457	27.943	68.749
SLGP	MPPI	iter5	Test 4	1:40:52	24,37	Success	60.955	80.059	11.295	45.026	68.012
SLGP	MPPI	iter1	Test 5	1:43:12	15,36	Stuck	59.585	79.434	10.059	26.222	68.812
SLGP	MPPI	iter2	Test 5	1:47:40	15,36	Stuck	59.706	79.777	10.258	27.887	68.461

SLGP	MPPI	iter3	Test 5	0:57:32	-1,06	Stuck	57.865	79.519	9.465	26.703	67.723
SLGP	MPPI	iter4	Test 5	1:01:28	-0,05	Stuck	59.222	79.065	9.464	24.426	68.715
SLGP	MPPI	iter5	Test 5	1:35:44	15,36	Stuck	62.661	81.750	11.246	20.717	68.999
SLGP	MPPI	iter1	Test 6	1:43:16	25,36	Success	61.518	78.963	11.767	35.950	68.980
SLGP	MPPI	iter2	Test 6	1:54:04	25,36	Success	62.077	80.998	11.826	22.632	69.000
SLGP	MPPI	iter3	Test 6	1:15:00	8,3	Stuck	63.804	80.762	11.455	28.569	68.295
SLGP	MPPI	iter4	Test 6	2:02:20	26,35	Success	61.911	81.379	12.223	43.088	69.000
SLGP	MPPI	iter5	Test 6	2:14:28	24,37	Success	60.839	79.047	12.482	31.188	68.875
SLGP	MPPI	iter1	Test 7	1:44:04	19,37	Stuck	62.893	78.813	11.451	26.931	68.718
SLGP	MPPI	iter2	Test 7	2:08:24	26,36	Success	60.593	79.119	12.269	22.844	69.008
SLGP	MPPI	iter3	Test 7	2:13:20	26,36	Success	60.610	79.766	11.876	21.915	68.875
SLGP	MPPI	iter4	Test 7	1:43:00	20,37	Stuck	60.772	81.441	10.045	32.765	68.745
SLGP	MPPI	iter5	Test 7	1:37:20	20,35	Stuck	61.336	79.710	10.010	33.141	68.996
SLGP	MPPI	iter1	Test 8	2:00:36	25,36	Success	60.876	79.188	12.045	28.710	69.006
SLGP	MPPI	iter2	Test 8	2:06:24	25,36	Success	60.001	79.078	11.981	28.403	69.000
SLGP	MPPI	iter3	Test 8	1:48:32	25,28	Stuck	60.247	81.282	10.280	29.751	69.000
SLGP	MPPI	iter4	Test 8	2:21:00	25,35	Success	60.113	79.961	12.188	27.714	69.035
SLGP	MPPI	iter5	Test 8	2:13:40	25,37	Success	59.946	79.054	12.112	28.160	69.000

Table 7: Test set 2 with Policy 3

Global Planner	Local Planner	Iter Num	Test Num	End time	End location	Status	CPU Util	CPU temp	RAM	GPU util	GPU temp
SBPL	NLOPT	iter1	Test 1	1:18:44	25,36	Success	56.828	79.648	10.258	33.984	75.773
SBPL	NLOPT	iter2	Test 1	1:27:52	26,36	Success	69.778	81.324	11.070	33.309	70.172
SBPL	NLOPT	iter3	Test 1	1:21:56	24,37	Success	61.024	82.195	10.036	27.842	70.205
SBPL	NLOPT	iter4	Test 1	1:17:44	26,36	Success	58.807	80.359	10.177	35.592	67.915
SBPL	NLOPT	iter5	Test 1	1:33:20	27,36	Success	59.592	80.867	10.993	28.413	69.127
SBPL	NLOPT	iter1	Test 2	1:17:04	5,28	Failed	59.731	80.259	10.796	32.246	69.789
SBPL	NLOPT	iter2	Test 2	1:19:28	5,28	Failed	61.204	81.281	10.735	30.766	69.230
SBPL	NLOPT	iter3	Test 2	1:10:00	5,27	Failed	64.268	80.816	10.807	28.316	70.004
SBPL	NLOPT	iter4	Test 2	1:30:44	25,37	Success	58.372	78.406	10.965	31.371	69.310
SBPL	NLOPT	iter5	Test 2	1:22:16	5,30	Failed	62.174	81.012	10.836	33.439	69.508
SBPL	NLOPT	iter1	Test 3	1:32:12	25,36	Success	57.742	79.749	11.691	29.648	69.984
SBPL	NLOPT	iter2	Test 3	1:42:20	26,37	Success	58.778	78.438	11.020	30.400	69.025
SBPL	NLOPT	iter3	Test 3	1:33:04	26,36	Success	59.567	78.500	10.988	27.921	68.586
SBPL	NLOPT	iter4	Test 3	1:34:28	26,36	Success	59.072	79.469	11.029	37.278	68.604
SBPL	NLOPT	iter5	Test 3	1:32:00	25,36	Success	59.793	78.375	11.690	27.071	68.722
SBPL	NLOPT	iter1	Test 4	1:22:56	25,34	Success	63.755	80.125	10.895	29.633	70.535
SBPL	NLOPT	iter2	Test 4	1:11:24	26,36	Success	61.703	77.875	10.594	34.594	69.641
SBPL	NLOPT	iter3	Test 4	1:16:08	25,34	Success	60.156	81.020	10.667	33.430	69.766
SBPL	NLOPT	iter4	Test 4	1:53:40	25,36	Success	59.401	79.250	11.214	35.269	68.986
SBPL	NLOPT	iter5	Test 4	1:05:56	9,3	Failed	62.408	81.059	10.639	33.156	69.406
SBPL	NLOPT	iter1	Test 5	2:23:40	12,2	Stuck	61.027	80.434	11.657	33.426	68.792
SBPL	NLOPT	iter2	Test 5	1:42:00	14,32	Failed	66.738	80.777	11.154	32.366	69.887

SBPL	NLOPT	iter3	Test 5	2:26:08	26,26	Failed	59.580	80.519	11.582	34.323	68.924
SBPL	NLOPT	iter4	Test 5	1:17:08	5,25	Failed	62.269	81.065	10.808	33.676	71.301
SBPL	NLOPT	iter5	Test 5	1:30:00	14,33	Failed	60.149	79.750	10.933	28.280	69.755
SBPL	NLOPT	iter1	Test 6	2:05:44	27,35	Success	65.858	80.963	10.614	34.984	69.641
SBPL	NLOPT	iter2	Test 6	1:52:36	24,34	Success	62.016	80.998	10.595	26.547	69.969
SBPL	NLOPT	iter3	Test 6	1:55:08	26,36	Success	59.776	80.762	10.816	35.293	69.877
SBPL	NLOPT	iter4	Test 6	1:52:36	26,36	Success	60.108	80.379	10.820	34.307	69.541
SBPL	NLOPT	iter5	Test 6	1:55:00	24,37	Success	61.388	80.047	10.639	33.125	69.141
SBPL	NLOPT	iter1	Test 7	1:37:32	25,36	Success	59.713	79.813	11.058	43.486	68.234
SBPL	NLOPT	iter2	Test 7	1:38:20	25,36	Success	61.138	81.119	11.164	37.733	68.102
SBPL	NLOPT	iter3	Test 7	1:39:32	26,36	Success	59.082	79.766	12.427	27.449	68.033
SBPL	NLOPT	iter4	Test 7	1:28:04	21,29	Failed	61.473	81.441	10.841	33.031	68.252
SBPL	NLOPT	iter5	Test 7	1:20:08	21,29	Failed	58.412	80.710	10.061	26.176	68.543
SBPL	NLOPT	iter1	Test 8	1:44:48	25,36	Success	64.521	80.188	12.278	34.068	69.005
SBPL	NLOPT	iter2	Test 8	1:19:40	25,29	Failed	64.726	81.078	12.222	29.200	67.948
SBPL	NLOPT	iter3	Test 8	1:36:28	25,36	Success	62.310	80.282	12.469	26.374	68.346
SBPL	NLOPT	iter4	Test 8	1:44:44	25,36	Success	58.721	80.961	12.962	38.410	68.921
SBPL	NLOPT	iter5	Test 8	1:17:00	23,29	Failed	58.338	80.954	11.615	36.609	68.865

Bibliography

[1] J.-C. Latombe, “Motion planning: A journey of robots, molecules, digital actors, and other artifacts,” The International Journal of Robotics Research, vol. 18, no. 11, pp. 1119–1128, 1999.

[2] T. Lozano-Pérez and M. A. Wesley, “An algorithm for planning collision-free paths among polyhedral obstacles,” Communications of the ACM, vol. 22, no. 10, pp. 560–570, 1979.

[3] P. Hart, N. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100–107, 1968.

[4] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp. 566–580, 1996.

[5] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” Proceedings. 1985 IEEE International Conference on Robotics and Automation.

[6] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. TR 98-11 Computer Science Dept., Iowa State University. http://lavalle.pl/papers/Lav98c.pdf, Oct 1998.

[7] S. J. Russell and E. Wefald, “chapter 2,” in Do the right thing: Studies in limited rationality, MIT Press, 1991, pp. 31–50.

[8] J. Svegliato, C. Basich, S. Saisubramanian, and S. Zilberstein, “Metareasoning for safe decision making in Autonomous Systems,” 2022 International Conference on Robotics and Automation (ICRA), 2022.

[9] J. Ortiz-Haro, E. Karpas, M. Toussaint, and M. Katz, "Conflict-directed diverse planning for logic-geometric programming," Proceedings of the International Conference on Automated Planning and Scheduling, vol. 32, pp. 279–287, 2022.

[10] A. Bhatia, J. Svegliato, S. B. Nashed, and S. Zilberstein, "Tuning the hyperparameters of anytime planning: A metareasoning approach with deep reinforcement learning," Proceedings of the International Conference on Automated Planning and Scheduling, vol. 32, pp. 556–564, 2022.

[11] "Collaborative Research Alliances," DEVCOM Army Research Laboratory. [Online]. Available: https://www.arl.army.mil/cras/sara-cra/sara-overview/. [Accessed: 21-Mar-2023].

[12] "Wiki," ros.org. [Online]. Available: http://wiki.ros.org/sbpl. [Accessed: 22-Mar-2023].

[13] T. M. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," The International Journal of Robotics Research, vol. 26, no. 2, pp. 141–166, 2007.

[14] N. Wagener, C.-an Cheng, J. Sacks, and B. Boots, "An online learning approach to model predictive control," Robotics: Science and Systems XV, 2019.

[15] B. Hedegaard, E. Fahnestock, J. Arkin, A. Menon, and T. M. Howard, "Discrete optimization of adaptive state lattices for iterative motion planning on Unmanned Ground Vehicles," 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021.

[16] A. Mandalika, S. Choudhury, O. Salzman, and S. Srinivasa, "Generalized lazy search for robot motion planning:interleaving search and edge evaluation via event-based toggles," Proceedings of the International Conference on Automated Planning and Scheduling, vol. 29, pp. 745–753, 2021.

[17] M. Likhachev, G. Gordon and S. Thrun, "ARA*: Anytime A* with Provable Bounds on Sub-Optimality," Proceedings of Advances in Neural Information Processing Systems 16(NIPS), 2003, pp. 258-265.

[18] "Robot operating system," ROS. [Online]. Available: https://www.ros.org/. [Accessed: 31-Mar-2023

[19] "Warthog unmanned ground vehicle robot - clearpath," Clearpath Robotics, 20-Dec-2022. [Online]. Available: https://clearpathrobotics.com/warthog-unmanned-ground-vehicle

[20] E. Khanmirza, M. Haghbeigi, M. Nazarahari, and S. Doostie, “A comparative study of deterministic and probabilistic mobile robot path planning algorithms,” 2017 5th RSI International Conference on Robotics and Mechatronics (ICRoM), 2017.

[21] M. Pivtoraiko and A. Kelly, “Constrained motion planning in discrete state spaces,” Springer Tracts in Advanced Robotics, pp. 269–280.

[22] C. Dellin and S. Srinivasa, “A unifying formalism for shortest path problems with expensive edge evaluations via lazy best-first search over paths with edge selectors,” Proceedings of the International Conference on Automated Planning and Scheduling, vol. 26, pp. 459–467, 2016.

[23] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, “Aggressive driving with model predictive path integral control,” 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016.

[24] Y. Sung, L. P. Kaelbling, and T. Lozano-Perez, “Learning when to quit: Meta-reasoning for motion planning,” 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021.

[25] L. Xu, F. Hutter, H. Hoos, and K. Leyton-Brown, “Evaluating component solver contributions to portfolio-based algorithm selectors,” Theory and Applications of Satisfiability Testing – SAT 2012, pp. 228–241, 2012.