

Planning Under Uncertainty: Informative and Stochastic Path Planning via Gaussian Processes

by

Avraiem Iskandar

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2025

© Avraiem Iskandar 2025

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Autonomous systems operating in real-world environments often face uncertainty due to incomplete or noisy information about their surroundings. Effective planning under such conditions requires models that can represent uncertainty and support informed decision-making. This thesis addresses planning under uncertainty in environments characterized by spatially correlated stochastic processes. We model the agent’s belief over these environments using Gaussian Processes (GPs), enabling Bayesian inference from sparse observations by capturing spatial correlations.

We study two problems. The first is informative path planning (IPP) in continuous space, where an agent collects measurements along a trajectory to reduce uncertainty about an unknown spatial field, subject to budget and obstacle constraints. We propose a hierarchical framework that integrates global graph-based planning with local continuous trajectory optimization to improve scalability and performance.

The second problem is stochastic path planning in random fields (SPPRF), where an agent seeks to minimize expected traversal cost in an environment with uncertainty induced by a stochastic process. The agent maintains a GP-based belief over the cost field and adapts its plan through belief updates as new data becomes available.

Our solutions to these problems combine methods and tools from stochastic processes, path planning, and trajectory optimization. In summary, the contributions of this thesis extend the planning capabilities of autonomous agents in environments governed by spatially correlated uncertainty, encompassing both informative exploration and uncertainty-aware planning.

Acknowledgements

I'm incredibly grateful to my supervisors, Professor Stephen L. Smith and Professor Yash Vardhan Pant, for their constant support and guidance. I feel fortunate to have been part of their research groups. Their attention to detail and exceptional rigour have shaped how I approach research, and I've learned a great deal from our meetings.

I'd like to thank Professor William Melek and Professor Gennaro Notomista for their time in reviewing this thesis and for their valuable feedback.

I'm also thankful to Shamak Dutta for the clarity and perspective he brought throughout this journey. His thoughtful questions pushed me to think more critically, and his insights made a real difference in shaping this thesis. I'd also like to thank my collaborators at the National Research Council of Canada, Dr. Kevin Murrant and Robert Gash, for their valuable input on bridging the theory of my work with real-world applications.

I sincerely thank Marian, Amr, and Hisham for kindly reading the thesis and offering thoughtful suggestions.

One of the things I've valued most about my time at Waterloo is the friendships I've built along the way. I'm especially grateful to my colleagues in the Autonomous Systems Group and the Control, Logic, and Learning Group, particularly Ahmed Hamouda, Prajwal, Ninghan, Leroy, Rod, Saem, Praneeth, Megnath, Akash, Jack, Krishna, Steven, Barry, Murphy, and Arezoo, for their support, conversations, coffee, and shared laughs. It was also a joy to join the program alongside such a brilliant group of students, including Andrew Nasif, Tahmid, and Shechem. Learning and growing with them made the experience all the more rewarding.

I'm deeply thankful to my close friends, Andrew Atef, Abanoub Sedhom, Ahmed Tamer, Mostafa Kishar, Omar Elayat, Omar Elalem, Mohamed Ashraf, Joseph, Philopatear, and the unforgettable "Waterloo Strays", as well as many others. Your endless support and good company helped more than you know.

A special thank you goes to my second family in Canada, the families of Maged, Maryana, Ghada, and Eric. Your kindness and presence made this journey feel far less lonely, and the way you helped me adjust and settle here made all the difference.

And finally, to my dear parents and my amazing sister, thank you for the strength and boundless love that have carried me through. Even when nothing made sense, it was your love that kept me moving forward. Every step I take rests on the sacrifices you made for me, and every achievement is yours more than it is mine.

Dedication

To my parents, my sister, and my grandmother Angelle, your love is my anchor, even across different worlds.

Table of Contents

Author's Declaration	ii
Abstract	iii
Acknowledgements	iv
Dedication	v
List of Figures	ix
List of Tables	xii
List of Abbreviations	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Thesis Contributions	3
1.3 Thesis Outline	4
2 Literature Review	5
2.1 Informative Path Planning	5
2.2 Stochastic Path Planning	8

3	Background	10
3.1	Gaussian Process Regression	10
3.2	Estimation Error and Optimal Estimators	12
3.3	Spline Curves	13
3.3.1	Bézier Curves	14
3.3.2	B-spline Curves	14
3.3.3	Usage and Shared Properties	14
3.4	Graph Terminology	15
4	Hierarchical Informative Path Planning	16
4.1	Introduction	16
4.2	Contributions	17
4.3	Problem Formulation	18
4.4	Hierarchical Optimization Approach Outline	20
4.5	Graph-Based Planning Stage	21
4.6	Budget Allocation Stage	22
4.6.1	Informative Region Bounding	23
4.6.2	Max Coverage Problem for Local Budget Allocation	26
4.7	Trajectory Optimization Stage	28
4.8	Simulation Results	31
4.8.1	Comparisons in obstacle-free environments	33
4.8.2	Comparisons in obstacle-cluttered environments	36
4.8.3	Real World Data	38
4.8.4	Evaluation in a Smaller Environment	41
4.9	Summary	44

5	Stochastic Path Planning in Random Fields	45
5.1	Introduction	45
5.2	Contributions	46
5.3	Problem Formulation	46
5.4	Modelling	48
5.4.1	Stochastic Cost of an Arbitrary Curve	49
5.5	Methodology	50
5.5.1	Belief Graph with Gaussian Process (GP) Edge Costs	51
5.5.2	Adaptive Replanning Policy	52
5.6	Simulation Results	53
5.6.1	Uncertain Terrain Navigation	54
5.6.2	Real Data Setup	57
5.7	Summary	59
6	Conclusion	61
6.1	Summary	61
6.2	Future Work	62
	References	63

List of Figures

1.1	An autonomous ship (black) navigating around islands in Central Ontario, near Georgian Bay, toward a target location (green dot). The ship aims to collect informative measurements about the spatial dispersion of water pollution. Blue-outlined regions indicate areas of interest for monitoring contaminant concentration. Source: [25]	2
4.1	An instance of informative path planning in $\mathcal{M}_{\text{free}}$. The robot moves from start s (green circle) to goal g (orange circle) along two trajectories γ : a higher-budget (purple) and a shorter-budget (dark green) trajectory, each collecting $n = 8$ measurements (dots). Test points T (blue crosses) are locations of interest, and obstacles (light red) must be avoided. The longer path is able to take measurements closer to test points, while the shorter path is limited by its budget.	20
4.2	Hierarchical Informative Path Planning (IPP) Strategy	21
4.3	<i>Left:</i> Any trajectory starting at point u and ending at v with total length at most L is fully contained within the ellipse $\mathcal{E}(u, v, L)$, defined by foci u , v and major axis length L . <i>Right:</i> The converse does not hold: a trajectory that remains entirely within the ellipse can still exceed length L if it contains loops or detours.	25
4.4	The Minkowski sum of the ellipse $\mathcal{E}(u, v, L)$ and a ball of radius r_{kernel} . The resulting region represents all points within distance r_{kernel} from some point on the ellipse.	26
4.5	Example of two optimized Bézier curves in free space, both with 20 measurement points. The yellow trajectory is given a higher budget B than the red, allowing it to loop and better cover informative test points (blue). The underlying cost field is sampled from a known GP.	30

4.6	Posterior mean and variance maps for three measurement scenarios: a short-budget Bézier trajectory, a long-budget Bézier trajectory, and an ideal case with measurements taken directly at the test points. In all scenarios, variance decreases near measurement locations. The long-budget trajectory achieves lower overall uncertainty due to more effective measurement placement, while the ideal case represents the best achievable measurements. . .	32
4.7	Two B-spline trajectory optimization outcomes in an obstacle-rich environment with 12 control points. Both trajectories are subject to the same constraints, but the non-convex nature of the problem leads to different local solutions sensitive to problem parameters.	33
4.8	Comparison of continuous-space IPP using CasADi with IPOPT (ours) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [32].	35
4.9	Comparison between graph-based Mixed Integer Convex Program (MICP) and casadi-ipopt (ours) in terms of posterior uncertainty and runtime. . . .	36
4.10	Performance comparison in obstacle-cluttered environments.	38
4.11	Hierarchical IPP in the Canadian Arctic Archipelago. (a) The importance field I indicates regions of interest for reducing uncertainty. (b) Real ice cover data obtained from satellite observations [14]. (c)–(d) Planned paths using the hierarchical planner under low and high budget constraints, respectively, overlaid on the GP posterior mean. Obstacle boundaries are shown in black.	40
4.12	Posterior mean squared error maps under two different path budgets. A higher budget allows for more informative measurements, reducing estimation error. The error scale is small (on the order of 10^{-4}) due to the limited range of the measured quantity, as ice cover values lie between 0 and 1. . .	41
4.13	IPP over a synthetic nitrate concentration field generated near islands in Central Ontario, close to Georgian Bay. (a) The true concentration field and test points are used for evaluating informativeness. (b)–(d) Trajectories generated by different planners: hierarchical planning, CMA-ES, and CasADi-IPOPT, respectively, shown over the GP posterior mean. Obstacles are shown in black.	43

5.1	An agent navigates an unknown environment while collecting measurements from the ground-truth field. These observations are used to update its GP belief, represented by the posterior mean and variance maps. The belief is then transformed into a probability distribution over edge costs (<i>belief graph</i>) for planning.	51
5.2	Visualization of a spatially varying traversal cost field used to evaluate stochastic planning algorithms. The cost map represents an unknown terrain with varying difficulty. The agent does not observe the field directly, but instead receives noisy local measurements and updates its belief using a Gaussian Process model.	56
5.3	Comparison of different planning strategies in the low-information scenario with only two initial observations.	56
5.4	Comparison of different planners in the high-information scenario with 150 initial observations.	57
5.5	Comparison of different planning strategies for navigating an uncertain cost field.	58
5.6	Final GP posterior mean (left) and variance (right) of the cost field, conditioned on measurements collected by the adaptive planner.	59
5.7	Planning over real ice concentration data from the National Research Council Canada (NRC)[43, 44], along the southern Labrador coast. The figure compares different path planning strategies under varying levels of prior information, highlighting the impact of uncertainty on navigation decisions.	60

List of Tables

4.1	Frequently used notation	18
4.2	Estimation performance under different observation strategies.	41
4.3	Estimation performance in the synthetic water pollution scenario.	42

List of Abbreviations

BLUE Best Linear Unbiased Estimator [13](#)

CMA-ES Covariance Matrix Adaptation Evolution Strategy [x](#), [7](#), [29](#), [31](#), [33–35](#), [37](#), [39](#), [42](#), [43](#)

CTP Canadian Traveller Problem [8](#), [9](#), [45](#)

GP Gaussian Process [viii–xi](#), [2–4](#), [7](#), [9–11](#), [17–19](#), [21](#), [22](#), [26](#), [28–30](#), [40](#), [42](#), [43](#), [46](#), [48–55](#), [57](#), [59](#), [61](#), [62](#)

GPR Gaussian Process Regression [10](#)

IPP Informative Path Planning [ix](#), [x](#), [1–8](#), [10](#), [14](#), [16](#), [17](#), [20](#), [21](#), [29](#), [35](#), [40](#), [43](#), [44](#), [61](#), [62](#)

MICP Mixed Integer Convex Program [x](#), [31](#), [36](#), [37](#)

MMSE Minimum Mean Squared Error [12](#), [13](#)

NLP Nonlinear Programming [3](#), [17](#), [21](#), [29](#), [31](#), [36](#)

RIG Rapidly-exploring Information Gathering [7](#)

SPPRF Stochastic Path Planning in Random Fields [1–4](#), [10](#), [45](#), [46](#), [61](#), [62](#)

SSPPR Stochastic Shortest Path with Recourse [8](#), [9](#), [45](#)

Chapter 1

Introduction

1.1 Motivation

Consider the following two scenarios.

In the first scenario, illustrated in Figure 1.1, a ship is equipped with sensors to collect measurements of water pollution dispersion, represented by the concentration of contaminants across a region of interest [49, 5]. Ideally, the ship would follow a path that visits all the areas of interest to gather accurate measurements, resulting in the best possible estimate of the concentration field. However, such a path would be too long and consume excessive fuel. Instead, the ship must carefully plan a shorter path that collects informative data along the way, while staying within a strict travel budget. This trade-off between accurate estimation and limited resources defines the core challenge of the [Informative Path Planning \(IPP\)](#) problem.

In the second scenario, the ship is navigating through the Arctic, where ice floes cover most of the water surface, leaving no open-water routes that can be followed without interacting with ice [15, 87]. As a result, collisions with ice are unavoidable, and the ship must traverse ice-covered waters in a way that minimizes the overall collision cost. This cost increases with ice thickness or concentration, as thicker and denser ice poses a higher risk of hull damage and requires more fuel to break through [47]. A key challenge is that the full spatial distribution of ice is unknown. Due to sensing limitations and environmental variability, captains do not have access to complete information about ice conditions [12]. However, both ice thickness and concentration are known to exhibit spatial correlations [22, 39]. This scenario defines the [Stochastic Path Planning in Random Fields \(SPPRF\)](#)

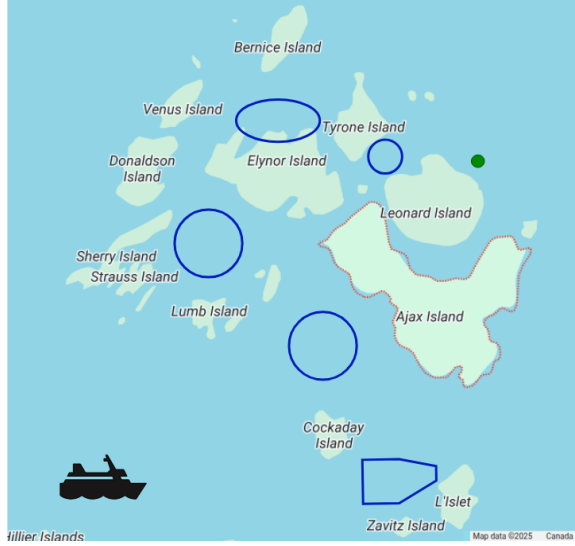


Figure 1.1: An autonomous ship (black) navigating around islands in Central Ontario, near Georgian Bay, toward a target location (green dot). The ship aims to collect informative measurements about the spatial dispersion of water pollution. Blue-outlined regions indicate areas of interest for monitoring contaminant concentration. Source: [25]

problem, in which the goal is to compute a *policy* that minimizes the expected cost of traversal in a spatially correlated environment with uncertainty, commonly referred to as a random field, while continuously updating decisions based on new observations gathered during navigation.

Although **IPP** and **SPPRF** have different goals, they are fundamentally connected through their reliance on spatially correlated stochastic processes. In **IPP**, the agent plans a path from a start to a goal while collecting measurements that reduce uncertainty about an unknown target phenomenon (e.g., temperature, salinity, or pollutant concentration), subject to a travel budget. In **SPPRF**, the agent navigates an environment where traversal costs are governed by an unknown random field (e.g., ice thickness, concentration, or wind speed), and must plan actions that minimize expected cost under uncertainty.

While **IPP** seeks to gather information for accurate estimation, **SPPRF** uses gathered information to make better decisions. In this sense, the two problems are dual. One uses planning to reduce uncertainty, the other uses reduced uncertainty to improve planning. Both problems are unified by a common modelling framework, where the underlying spatial process is treated as a **Gaussian Process (GP)** [83] that enables Bayesian inference and uncertainty quantification from sparse observations.

1.2 Thesis Contributions

This thesis addresses two closely related problems in planning under uncertainty, namely [IPP](#) in continuous space and [SPPRF](#) on graphs. The main contributions are summarized as follows.

1. **Hierarchical Informative Planning Framework:** We develop a hierarchical framework that combines global [IPP](#) on sparse graphs with local refinement through continuous optimization. The local planner formulates trajectory optimization as a [Nonlinear Programming \(NLP\)](#) using spline representations, allowing us to compute trajectories that respect obstacles and budget constraints. Compared to black-box optimizers, this formulation offers significantly improved computational efficiency while maintaining high-quality solutions. To further enhance planning performance, we introduce a heuristic budget allocation strategy that prioritizes the refinement of segments expected to yield the most informative measurements.
2. **Stochastic Motion Planning with Online Belief Updates:** We address [SP-PRF](#) in environments with spatially correlated stochastic cost fields, where costs are uncertain and partially revealed through observations along the path. We model the cost field using [GP](#) and map it to a probability distribution over graph edge costs, then dynamically update this distribution as new measurements are collected. This enables an adaptive replanning strategy that continuously updates the agent’s belief and adjusts plans accordingly.
3. **Simulation Studies and Real-World Validation:** We evaluate our hierarchical planner across simulated environments of varying obstacle complexity, showing the benefits of combining graph guidance with local refinement over purely continuous [IPP](#). For [SPPRF](#), we also provide simulations to compare the proposed strategy against other baseline methods. Finally, we validate both approaches on real-world geospatial data from Canadian coastal and Arctic regions, using satellite-derived fields and maps to simulate realistic monitoring conditions. Results show that our hierarchical planner outperforms all [IPP](#) baselines while maintaining competitive efficiency, and that our adaptive strategy achieves performance gains over baselines in [SPPRF](#).

1.3 Thesis Outline

The remainder of this thesis is structured as follows.

Chapter 2 presents a review of related work in informative path planning and stochastic path planning. We discuss key approaches to IPP, and connections to sensor placement and orienteering problems. Additionally, we discuss popular methods for stochastic path planning and highlight their limitations.

Chapter 3 reviews the key mathematical tools used throughout this work, including Gaussian processes, estimation theory, Bézier and B-spline curves, and graph terminology.

Chapter 4 formulates the IPP problem in continuous space and introduces a hierarchical framework that combines graph-based planning with local spline-based optimization. The approach is evaluated through simulations across both generic environments and realistic scenarios constructed from Canadian geographic and environmental data.

Chapter 5 formulates the SPPRF problem under spatially correlated uncertainty. A GP-based framework is used to model the unknown cost field and map it to a probability distribution over edge costs on a graph. We compare online and offline planning strategies through simulations conducted in both abstract environments and realistic scenarios constructed from Arctic data.

Chapter 6 concludes the thesis with final remarks and a discussion of possible future directions.

Chapter 2

Literature Review

In this chapter, we review related work from the literature on the two primary problems of interest: informative path planning and stochastic path planning.

2.1 Informative Path Planning

[IPP](#) lies at the intersection of sensing, planning, and optimization. It builds on early ideas from sensor placement, which focuses on maximizing information gain by choosing informative measurements under resource constraints, and from the orienteering problem, which addresses reward-maximizing routing within a travel budget. These foundations gave rise to [IPP](#) as a unified framework for actively collecting information through motion. Over time, [IPP](#) has expanded to include a variety of approaches, including discrete graph-based methods and continuous-space planners based on sampling or trajectory optimization. While our work does not aim to cover the full spectrum of [IPP](#), we draw connections to several of these threads. In the following subsections, we review relevant literature in sensor placement, orienteering, graph-based planning, and continuous-space planning.

Sensor Placement

The sensor placement problem is a classical problem in sensing and estimation, where the objective is to select a subset of sensor locations in a given environment to monitor an underlying spatial or spatio-temporal phenomenon. Typically, this involves maximizing some information-theoretic objective while satisfying constraints related to available resources

[42], heterogeneous deployment cost [48, 41], or spatial and temporal feasibility. A key result in sensor placement [42] utilizes the sub-modularity of mutual information to show that a greedy algorithm achieves a constant-factor approximation guarantee, making it an efficient and provably near-optimal solution for maximizing information under budget constraints. However, in [18], the authors demonstrate that estimation error, which we are interested in, is not a submodular function. As a result, the standard approximation guarantees associated with greedy algorithms do not necessarily apply.

While classical sensor placement focuses on choosing fixed sensor locations before deployment, more general formulations also consider scheduling and control. [20] proposes a mixed-integer optimization approach for sensor scheduling in linear dynamical systems, minimizing Kalman filter error covariance under polyhedral constraints. Their framework unifies placement, scheduling, and sensing-control co-design.

Orienteering

The orienteering problem [24] is a classical combinatorial optimization problem that arises in scenarios where a mobile agent must collect rewards by visiting a subset of locations, subject to a limited travel budget. It generalizes the travelling salesman problem by associating rewards with locations and relaxing the requirement to visit all nodes, and is therefore also NP-hard. Solving the orienteering problem involves a fundamental trade-off between path cost and reward acquisition, which makes it highly relevant to IPP, where a similar trade-off arises between travel cost and information gain. The Correlated Orienteering Problem [84, 85] extends the classical orienteering problem by incorporating spatial correlations in the reward function, making it well-suited for informative path planning. The environment is modelled as a time-invariant, spatially correlated Gaussian random field, and the problem is formulated as a mixed-integer Quadratic program that maximizes a quadratic utility function subject to a travel budget constraint.

Graph-based IPP

Several methods have been proposed to address the IPP problem on graphs. [9] used a branch and bound approach and exploited the monotonicity of variance reduction (the property that adding more measurements never increases the posterior variance) to derive an upper bound when exploring branches of the search tree. Their approach maintains the best solution found so far (i.e., the highest value of the objective) and prunes any branch whose upper bound is worse than this best value. This approach is solving for a globally

optimal solution, but is limited to small graphs. Other approaches have reformulated the optimization problem to jointly optimize both the measurement locations and the corresponding linear estimators using mixed-integer quadratic programs. This strategy was first introduced in the context of sensor scheduling and selection for Kalman filtering in [20], and later extended to informative path planning in [19], both within the linear estimation framework. More recently, [21] generalized this approach to GP by minimizing the trace of the expected squared estimation error covariance. However, this method remains restricted to small graphs and a limited number of target locations due to computational complexity. In [61], an approximate sequential approach based on dynamic programming was proposed. This method iteratively constructs a path by updating the GP posterior based on previously collected measurements. A convex relaxation of the informative path planning problem was also derived to serve as a lower bound, and the resulting optimality gap of the sequential approach was shown to be small. This approach is significantly more scalable than global optimization methods and can handle larger graphs efficiently, making it suitable for practical applications.

Continuous-space IPP

While graph-based methods can yield globally optimal solutions on the chosen graph, the quality of the solution heavily depends on the discretization and choice of vertices and edges. In sparse graphs, key regions may be missed, and in dense graphs, computational cost becomes prohibitive.

Sampling-based methods extend motion planning to continuous spaces while preserving scalability. **Rapidly-exploring Information Gathering (RIG)** [33], based on RRT* [36], supports submodular and time-varying objectives, and uses branch-and-bound pruning to ensure asymptotic optimality. However, RIG selects measurement locations without considering the trajectory during sampling.

Trajectory-parameterized methods address this by jointly optimizing the trajectory and information acquisition. [51] proposed a method using spline parameterization, similar to ours, but employed Bayesian Optimization to optimize the trajectory. They evaluated the accumulated value of information acquisition over a trajectory via line integrals, while we use finite measurement locations along the trajectory. Their formulation did not constrain the trajectory length, which is essential in IPP. [32] improved efficiency and performance by using the **Covariance Matrix Adaptation Evolution Strategy (CMA-ES)** [28], and introduced an adaptive replanning scheme. We compare our planner with this method because it shares similarities with our approach in terms of trajectory parameterization, finite measurement locations, and the inclusion of arc length constraints. We compare only against

their non-adaptive variant, which performs full offline trajectory optimization consistent with our setting. Furthermore, [64] expanded this framework by using informed initialization in the optimization process to achieve faster convergence. Additionally, [54] integrated obstacle considerations into continuous-space planning by treating collision penalties as part of the cost function. In contrast, our approach imposes obstacle avoidance as a hard constraint, rendering solutions that intersect obstacles infeasible.

Beyond estimation- and information-theoretic approaches, ergodic exploration offers a related but distinct perspective on continuous-space IPP. In this framework, trajectories are designed so that their time-averaged spatial statistics align with a prescribed density representing the spatial distribution of information [53, 56, 57]. Unlike IPP, which directly minimizes posterior uncertainty or maximizes information gain of a latent field, ergodic exploration assumes the density of information is known and seeks coverage with respect to this density. In this work, we focus on IPP by modelling correlated fields and defining informativeness through uncertainty reduction rather than time-averaged coverage.

2.2 Stochastic Path Planning

Classical path planning has been widely studied across robotics, artificial intelligence, and operations research. These methods operate under the assumption that the environment is represented as a graph with edge costs that are fixed and known in advance. In this deterministic setting, several well-established algorithms exist. Dijkstra’s algorithm [17] guarantees the shortest path in graphs with nonnegative edge weights, while A* [29] improves efficiency by using heuristics to guide the search without sacrificing optimality. To handle environments that are initially unknown, algorithms like D* [74] and D* Lite [40] support incremental replanning by efficiently updating the path as new information becomes available. However, they do not model edge costs or availability as random variables, nor do they maintain a belief over unobserved parts of the environment. As a result, they cannot reason about correlations, uncertainty, or the expected outcomes of future actions. This limitation has motivated the development of stochastic path planning models, which explicitly incorporate uncertainty and allow the agent to plan under partial information.

Path planning under uncertainty is a long-standing problem in the literature. Perhaps two of the early foundations of this problem are the **Canadian Traveller Problem (CTP)** introduced in [62] and the **Stochastic Shortest Path with Recourse (SSPPR)** proposed in [63]. The CTP involves planning in a graph where a subset of the edges might be blocked or open, but the agent does not know the state of the edge until reaching the endpoint. This problem is PSPACE-complete, and in cases where the blockage probabilities are given

in advance, it is P-hard. A limitation of this setting is that arc states are modelled as independent random variables, which prevents the agent from taking advantage of the history of observations it made about some of the arcs.

To address this limitation, an alternative version of the CTP was considered in [50], where the agent was initially provided with a set of random variables in the form of subgraphs from the original graph representing the environment. Additionally, the agent knows the probability associated with each subgraph. This approach enables the agent to infer the state of the unvisited edges using previous observations. In [77], the authors explored the same problem but did not assume that the agent has access to the set of subgraphs and their likelihood before deployment. Instead, the agent learns this information through executing repeated tours in the environment.

The SSPPR is a similar problem, where the agent plans in a graph with arc costs that are discrete random variables. These random variables could be independent or dependent. In the former case, the agent is initially given the discrete probability distribution of each edge cost, and in the latter case, the agent is given a set of finite realizations of the joint edge costs along with their probability distribution. Similar to the CTP, the agent can know the actual edge cost when reaching the endpoint of the edge.

The common shortcoming in all of the previously mentioned work is the assumption that a finite set of outcomes can represent the environment. While this may be applicable in certain practical scenarios, such as the factory model discussed in [50], it is often limited. This is because real-world environments are more complex, and the possibilities cannot be easily enumerated.

To address this issue, some work considered the edge costs to be modelled as continuous random variables. In [26], the authors assumed that the joint distribution of the arc costs follows a Gaussian distribution, which they stated could be estimated from real-world data. This approach could be useful in applications such as routing in stochastic networks, where the problem is inherently discrete. However, when dealing with complex environments exhibiting physical phenomena with spatial correlations, it may be more appropriate to represent the environment as a stochastic process.

One relevant study to this work is [16], where the authors used GP for aircraft navigation under uncertain wind conditions. However, it was not clear from their formulation how to transform continuous GP into stochastic arc costs on a graph. The planning method used was greedy in the sense that it didn't reason about future beliefs of the GP. [68] determined the reachable regions by converting GP to a graph with random arc costs, but the planning approach was only done once, in a greedy manner.

Chapter 3

Background

This chapter provides a concise review of the core mathematical concepts that underpin the methods developed in this thesis. The aim is to offer a self-contained reference for key ideas such as [Gaussian Process Regression \(GPR\)](#), optimal estimation, spline-based trajectory representations, and graph-theoretic terminology. These tools are used in the main chapters, Chapter 4 on Informative Path Planning and Chapter 5 on Stochastic Path Planning in Random Fields. Readers already familiar with these concepts are advised to skip or skim this chapter and refer back to it as needed.

3.1 Gaussian Process Regression

In this work, we use [GPR](#) to model unknown physical phenomena with spatial correlations. [GPR](#) provides a Bayesian nonparametric framework for modelling unknown spatial fields with both predictive mean and uncertainty estimates. This probabilistic structure is central to the two planning problems addressed in this thesis. In [IPP](#) (Chapter 4), [GPs](#) enable the selection of the most informative trajectories by allowing closed-form computation of the mean squared estimation error. In [SPPRF](#) (Chapter 5), [GPs](#) model traversal costs as random variables defined over space, yielding a belief over the cost that is updated with new observations. Importantly, their closure under linear operations allows the analytical computation of line integrals, which is critical for evaluating the probability distribution of random costs over paths. These features make [GPs](#) particularly well-suited for reasoning and decision-making under uncertainty. For a detailed review of [GPR](#) and [GP](#), we refer the reader to [\[83, 35\]](#).

Definition 1 (Stochastic Process, [37]). Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $\mathcal{X} \subseteq \mathbb{R}^d$ be an index set. A stochastic process is a collection of real-valued random variables $\{f(x)\}_{x \in \mathcal{X}}$, where each $f(x) : \Omega \rightarrow \mathbb{R}$ is a random variable defined on $(\Omega, \mathcal{F}, \mathbb{P})$.

Definition 2 (Positive Definite Kernel, [35]). A function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is called a positive definite kernel if it satisfies the following two conditions:

1. **Symmetry:** $k(x, y) = k(y, x)$ for all $x, y \in \mathcal{X}$.
2. **Positive Semi-Definiteness:** For any finite set of points $\{x_1, \dots, x_n\} \subset \mathcal{X}$ and any real numbers $c_1, \dots, c_n \in \mathbb{R}$,

$$\sum_{i=1}^n \sum_{j=1}^n c_i c_j k(x_i, x_j) \geq 0$$

Definition 3 (Gaussian Process, [35]). A stochastic process $f : \mathcal{X} \rightarrow \mathbb{R}$ is called a **GP** with mean function m and covariance kernel k , written $f \sim \mathcal{GP}(m, k)$, if for every finite collection $\{x_1, \dots, x_n\} \subset \mathcal{X}$, the vector $f_X = [f(x_1), \dots, f(x_n)]^\top$ is multivariate Gaussian:

$$f_X \sim \mathcal{N}(m_X, K_{XX}),$$

where $m_X = [m(x_1), \dots, m(x_n)]^\top$ and $(K_{XX})_{ij} = k(x_i, x_j)$. That is, $m(x) = \mathbb{E}[f(x)]$ is the pointwise mean function, and $k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$ is the covariance function.

Theorem 1 (Posterior Distribution in GP Regression, [35]). Let $f \sim \mathcal{GP}(m, k)$ and suppose we observe $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ with

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2). \quad (3.1)$$

Define $X = [x_1, \dots, x_n]$, $Y = [y_1, \dots, y_n]^\top$, $m_X = [m(x_1), \dots, m(x_n)]^\top$, and the covariance blocks of the Gram matrix are

$$K_{XX} = (k(x_i, x_j))_{i,j}, \quad K_{xX} = [k(x, x_1) \ \cdots \ k(x, x_n)], \quad K_{Xx} = K_{xX}^\top.$$

Then, for any test input x ,

$$f(x) \mid Y \sim \mathcal{N}(m_*(x), k_*(x, x)),$$

with posterior mean and variance

$$\begin{aligned} m_*(x) &= m(x) + K_{xX} (K_{XX} + \sigma_\epsilon^2 I)^{-1} (Y - m_X), \\ k_*(x, x) &= k(x, x) - K_{xX} (K_{XX} + \sigma_\epsilon^2 I)^{-1} K_{Xx}. \end{aligned} \quad (3.2)$$

For completeness, we include a short proof of this standard result.

Proof. From the definition of a Gaussian process, the joint distribution of the observed outputs Y and the latent value $f(x)$ is Gaussian:

$$\begin{bmatrix} Y \\ f(x) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_X \\ m(x) \end{bmatrix}, \begin{bmatrix} K_{XX} + \sigma_\epsilon^2 I & K_{Xx} \\ K_{xX} & k(x, x) \end{bmatrix}\right).$$

By the conditioning formula for multivariate Gaussians, the conditional distribution $f(x) | Y$ is Gaussian with mean and variance as in (3.2). The more general cross-covariance

$$k_*(x, x') = k(x, x') - K_{xX}(K_{XX} + \sigma_\epsilon^2 I)^{-1}K_{Xx'}$$

follows similarly by conditioning the joint Gaussian of $(Y, f(x), f(x'))$. $\square$

3.2 Estimation Error and Optimal Estimators

The primary objective in informative path planning is to select measurement locations that reduce uncertainty about an unknown quantity. Several information-theoretic criteria have been proposed to quantify measurement value, including mutual information, entropy reduction, and worst-case variance. In this work, we focus on minimizing the expected squared estimation error, which, under Gaussian assumptions, is equivalent to minimizing the trace of the posterior covariance matrix. The following results summarize key ideas from estimation theory that support this objective. For a more comprehensive treatment, we refer the reader to [38, 2].

Let X and Y be jointly Gaussian random vectors, where $X: \Omega \rightarrow \mathbb{R}^n$ and $Y: \Omega \rightarrow \mathbb{R}^m$. That is, X and Y are random variables taking values in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively, with known means and covariances:

$$\begin{aligned} \mathbb{E}[X] &= \mu_X, & \mathbb{E}[Y] &= \mu_Y, \\ \text{Cov}[X] &= \Sigma_{XX}, & \text{Cov}[Y] &= \Sigma_{YY}, & \text{Cov}[X, Y] &= \Sigma_{XY}. \end{aligned} \tag{3.3}$$

The [Minimum Mean Squared Error \(MMSE\)](#) estimator of X given Y is the conditional expectation:

$$\hat{X} := \mathbb{E}[X | Y], \tag{3.4}$$

which minimizes the expected squared error:

$$\mathbb{E} \left[\|X - \hat{X}\|^2 \right]. \tag{3.5}$$

The conditional covariance of X given Y is defined as

$$\begin{aligned}\text{Cov}[X | Y] &:= \mathbb{E} \left[(X - \hat{X})(X - \hat{X})^\top | Y \right] \\ &= \Sigma_{XX} - \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX},\end{aligned}\tag{3.6}$$

which represents the covariance of the estimation error.

Theorem 2 ([MMSE](#) is Equivalent to the Trace of Conditional Covariance). *Let X and Y be jointly Gaussian random vectors taking values in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. Then the minimum mean squared error of estimating X given Y is equal to the trace of the conditional covariance:*

$$\mathbb{E} \left[\|X - \hat{X}\|^2 \right] = \text{Tr}(\text{Cov}[X | Y]).\tag{3.7}$$

Proof. Let $E := X - \hat{X}$ denote the estimation error. Then

$$\mathbb{E}[\|E\|^2] = \mathbb{E}[E^\top E] = \text{Tr}(\mathbb{E}[EE^\top]) = \text{Tr}(\text{Cov}[X | Y]).$$

□

Remark 1. *The conditional covariance $\text{Cov}[X | Y]$ is deterministic and depends only on the prior variance and covariance. As a result, the expected estimation error can be evaluated offline, without the need to collect actual measurements.*

Remark 2. *In the Gaussian setting, the [MMSE](#) estimator coincides with the [Best Linear Unbiased Estimator \(BLUE\)](#), which takes the affine form*

$$\hat{X}_{lin} = \mu_X + \Sigma_{XY} \Sigma_{YY}^{-1} (Y - \mu_Y).$$

Thus, the optimal estimator under squared loss is affine in Y .

3.3 Spline Curves

Parametric curves are widely used to represent smooth trajectories in continuous space. We primarily use Bézier and B-spline curves, which provide differentiable representations and support constraint enforcement. While Bézier curves are intuitive and compact, B-splines offer greater flexibility and are better suited for cluttered environments due to their local control.

3.3.1 Bézier Curves

A degree- d Bézier curve with control points $\{P_0, \dots, P_d\}$ is defined as:

$$\gamma(t) = \sum_{i=0}^d B_i^d(t) P_i, \quad t \in [0, 1], \quad (3.8)$$

where the Bernstein basis polynomials are

$$B_i^d(t) = \binom{d}{i} (1-t)^{d-i} t^i. \quad (3.9)$$

The curve satisfies $\gamma(0) = P_0$, $\gamma(1) = P_d$.

3.3.2 B-spline Curves

B-splines generalize Bézier curves by combining piecewise polynomial segments using compactly supported basis functions. A degree- d B-spline curve with control points $\{P_0, \dots, P_n\}$ and knot vector $\{t_0, \dots, t_m\}$ is defined as:

$$\gamma(t) = \sum_{i=0}^n N_{i,d}(t) P_i, \quad (3.10)$$

with basis functions $N_{i,d}(t)$ given recursively by the Cox-de Boor formula:

$$\begin{aligned} N_{i,0}(t) &= \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \\ N_{i,d}(t) &= \frac{t - t_i}{t_{i+d} - t_i} N_{i,d-1}(t) + \frac{t_{i+d+1} - t}{t_{i+d+1} - t_{i+1}} N_{i+1,d-1}(t). \end{aligned} \quad (3.11)$$

We use clamped B-splines with uniform interior knots. This guarantees endpoint interpolation, i.e., $\gamma(0) = P_0$, $\gamma(1) = P_n$, and consistent parameterization.

3.3.3 Usage and Shared Properties

Bézier and B-spline curves are both used in our [IPP](#) framework (Chapter 4). Bézier curves are well-suited for less cluttered environments or shorter trajectories, as each control

point influences the entire curve. In contrast, B-splines offer local control, where each control point affects only a portion of the curve. This makes them preferable in cluttered environments, where local adjustments are often needed to avoid obstacles.

In terms of smoothness, the two curve types differ. A single Bézier curve is a polynomial and therefore infinitely differentiable (C^∞) with respect to its parameter. In contrast, a degree- d B-spline is only piecewise polynomial and typically achieves C^{d-1} continuity at interior knots, with reduced continuity when knots are repeated.

Despite these differences, both Bézier and B-spline curves share important geometric properties. Each point on the curve is a convex combination of the control points, formed using non-negative basis functions that sum to one (Equations 3.8–3.11). This guarantees that the curve lies entirely within the convex hull of its control points.

3.4 Graph Terminology

Many planning problems can be naturally modelled using graphs, which compactly represent connectivity between discrete configurations or locations. In motion planning, feasible transitions are typically captured using a *weighted graph*, which may be directed or undirected. Formally, a graph is defined as a pair $G = (V, E)$, where V is the set of vertices and $E \subseteq V \times V$ is the set of edges. A cost function $l : E \rightarrow \mathbb{R}_{\geq 0}$ assigns a nonnegative weight to each edge. An edge $(u, v) \in E$ represents a transition from vertex u to v , with an associated cost $l(u, v)$ that may reflect distance, time, energy, or other traversal penalties. Throughout this thesis, we use the term *path*¹ to denote a sequence of connected vertices from a start to a goal, possibly revisiting vertices. This allows flexibility in re-planning. With slight abuse of notation, the cost of a path $\mathcal{P} = (v_1, \dots, v_k)$ is then defined as

$$l(\mathcal{P}) := \sum_{i=1}^{k-1} l(v_i, v_{i+1}).$$

For any vertex $v \in V$, we define the set of outgoing edges as

$$\delta^+(v) = \{(v, u) \in E\}, \quad \text{and the set of incoming edges as} \quad \delta^-(v) = \{(u, v) \in E\}.$$

¹In standard graph theory, a *walk* allows repeated vertices, while a *path* is a walk with no repetitions. We adopt the term “path” more generally to denote any sequence of connected vertices, possibly with repetitions, for clarity and consistency.

Chapter 4

Hierarchical Informative Path Planning

4.1 Introduction

The [Informative Path Planning \(IPP\)](#) problem is a core challenge in robotics and autonomous systems. It involves planning a path for a mobile agent that collects measurements while operating under resource constraints, such as limited travel distance or energy. As the agent moves through the environment, it uses onboard sensors to observe an unknown spatial function representing a physical phenomenon (e.g., temperature or pollutant concentration). The goal is to select a path that enables the most accurate estimation of this function, typically by minimizing a metric such as the mean squared estimation error.

For instance, in Arctic exploration, a ship may be tasked with estimating sea-ice cover, but limited fuel prevents it from surveying the entire region. The challenge is to find a feasible path that provides the most useful measurements within the available budget. Although much research has focused on [IPP](#) for ocean monitoring [[73](#), [45](#), [32](#), [9](#), [8](#)], the problem arises in many other domains, including:

- Air quality and atmospheric pollution monitoring [[79](#), [13](#), [51](#)],
- Wildfire tracking and boundary estimation [[1](#), [72](#), [10](#), [4](#)],
- Terrain mapping using unmanned aerial vehicles (UAVs) [[64](#), [65](#), [76](#), [67](#)],
- Soil property estimation using unmanned ground vehicles (UGVs) [[69](#), [81](#)],

- Autonomous exploration by planetary rovers in space missions [70].

The IPP problem generalizes the Orienteering Problem [24] and is therefore NP-hard. Most prior work formulates it in discrete graph settings, using either approximate methods [61] or globally optimal solvers that scale only to small graphs [9, 19, 21]. These methods typically assume that measurements are acquired only at vertices, which does not reflect the continuous sensing capabilities of robotic platforms. While increasing graph density can approximate continuous measurements, it leads to scalability issues as resolution grows. To overcome this, some studies have investigated continuous-space trajectory optimization using splines [32, 64, 51], but often assume the absence of obstacles in the environment. This assumption does not hold in applications like water navigation, where trajectories must avoid land [11], shallow regions [30], or drifting ice floes [58, 15, 87]. We empirically show that while continuous optimization often outperforms graph solvers in obstacle-free settings, it can converge to poor local optima in obstacle-cluttered environments. To address this, we propose a hierarchical planning framework that combines the global guidance of graph-based planning with the local refinement capabilities of continuous trajectory optimization. This integration helps steer non-convex solvers toward more promising regions of the solution space.

A key challenge in IPP is that the informativeness of a new measurement depends on what has already been measured. This is due to spatial correlations in the underlying Gaussian Process (GP). For example, taking a measurement at one location also reduces uncertainty in nearby areas, so taking another nearby measurement may not add much new information. This leads to diminishing returns. Some objectives, like mutual information, show this behaviour and are submodular under certain conditions. In such cases, greedy algorithms can provide near-optimal solutions with performance guarantees [42]. In our work, we use a different objective, which is minimizing the mean squared estimation error, or the trace of the posterior covariance. This function is not submodular [34, 86], but it often shows diminishing returns in practice with GP models.

4.2 Contributions

In this chapter, we present a continuous-space formulation of the IPP problem using spline-based trajectory representations, which we cast as an Nonlinear Programming (NLP) solvable by interior-point methods. This approach yields locally optimal trajectories that satisfy obstacle and budget constraints while offering improved computational efficiency

compared to black-box optimizers. To address the limitations of purely continuous methods in cluttered environments, we introduce a hierarchical framework that combines global path planning on sparse graphs with local trajectory refinement. A pruning strategy is employed to eliminate irrelevant obstacles outside each segment’s refinement region, enhancing efficiency. To further improve performance, we propose an ellipse-based heuristic that estimates the informativeness of each segment and guides the allocation of the trajectory improvement budget. Finally, we demonstrate the effectiveness of our approach through simulations across environments of varying complexity and validate its applicability using real-world Arctic datasets.

4.3 Problem Formulation

Table 4.1: Frequently used notation

Symbol	Description
$\mathcal{M} \subset \mathbb{R}^d$	Environment domain
$\mathcal{M}_{\text{free}}$	Collision-free space
$f \sim \mathcal{GP}(0, k)$	Unknown spatial field modeled as a GP
$k(x, x')$	Squared exponential kernel
$T = \{\tau_1, \dots, \tau_m\}$	Set of test points
$\Sigma_{T X}$	Posterior covariance over test points
M	Weight matrix for test points
$J_M(\gamma)$	Objective for an arbitrary trajectory γ
$G = (V, E)$	Planning graph
$\mathcal{P}$	Selected path on the graph
$\mathcal{X}$	Binary edge selection vector
$\tilde{J}_M(\mathcal{X})$	Objective for graph-based plan
L_e	Budget assigned to edge e
$P^e = [P_0^e, \dots, P_r^e]$	Control points of spline for edge e
$\bar{J}_M(P^e)$	Objective over control points
$\gamma_e(t)$	Parametric spline segment for edge e

Let $\mathcal{M} \subset \mathbb{R}^d$ be a compact convex domain representing the environment, and let $\mathcal{M}_{\text{obs}} \subset \mathcal{M}$ denote the region occupied by static obstacles. Define the collision-free space as $\mathcal{M}_{\text{free}} := \mathcal{M} \setminus \mathcal{M}_{\text{obs}}$. An autonomous robot navigates through $\mathcal{M}_{\text{free}}$ to gather information

about an unknown scalar function $f : \mathcal{M} \rightarrow \mathbb{R}$, modeled as a GP:

$$f \sim \mathcal{GP}(0, k),$$

with zero mean and a positive semidefinite kernel $k : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}$. In this work, we use the squared exponential (SE) kernel:

$$k(x, x') = \sigma^2 \exp\left(-\frac{1}{2\ell^2} \|x - x'\|^2\right), \quad (4.1)$$

where σ^2 is the signal variance, ℓ is the length-scale parameter, and $x, x' \in \mathcal{M}$ are two input locations.

Remark 3. *The zero-mean prior is a standard assumption, since any nonzero mean can be subtracted from the data and added back after inference. Moreover, under the Gaussian assumption, the mean does not influence the covariance structure. This follows from Remark 1, which shows that the conditional covariance depends only on the prior covariance. Since our objective later relies on covariance properties, the zero-mean prior simplifies the formulation without loss of generality.*

Let $T = \{\tau_1, \dots, \tau_m\} \subset \mathcal{M}$ be a finite set of preselected test locations at which we wish to minimize the posterior uncertainty about f . The robot collects noisy measurements along its trajectory $\gamma : [0, 1] \rightarrow \mathcal{M}_{\text{free}}$, subject to length constraints. Observations are sampled at a finite set of points along γ , denoted by $X = \{x_1, \dots, x_n\} \subset \text{Im}(\gamma)$, where each measurement is given by:

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2) \quad (4.2)$$

We assume the noise terms ϵ_i are independent and identically distributed (i.i.d.) Gaussian random variables with zero mean and variance σ_ϵ^2 . Following Theorem 1, the GP posterior covariance matrix over T conditioned on X is given as:

$$\Sigma_{T|X} = K_{TT} - K_{TX}(K_{XX} + \sigma_\epsilon^2 I)^{-1}K_{XT}, \quad (4.3)$$

where K_{TT}, K_{TX}, K_{XX} are standard kernel Gram matrices evaluated at test and observation points. Let $M \in \mathbb{R}^{m \times m}$ be a fixed positive semidefinite weighting matrix encoding the relative importance of test points. Our goal is to minimize the weighted estimation error over the test set, and from Theorem 2 we know this is the weighted trace of posterior covariance.

$$J_M(\gamma) := \text{Tr}(M\Sigma_{T|X}), \quad (4.4)$$

The informative path planning problem in continuous space is then formally stated as:

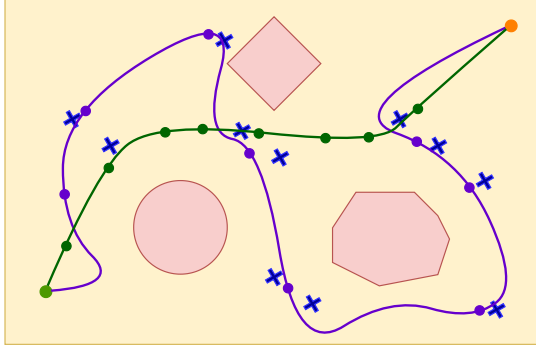


Figure 4.1: An instance of informative path planning in $\mathcal{M}_{\text{free}}$. The robot moves from start s (green circle) to goal g (orange circle) along two trajectories γ : a higher-budget (purple) and a shorter-budget (dark green) trajectory, each collecting $n = 8$ measurements (dots). Test points T (blue crosses) are locations of interest, and obstacles (light red) must be avoided. The longer path is able to take measurements closer to test points, while the shorter path is limited by its budget.

Problem 1 (IPP in Continuous Space). *Given a start point $s \in \mathcal{M}_{\text{free}}$, goal point $g \in \mathcal{M}_{\text{free}}$, trajectory length budget $B > 0$, weight matrix $M \succeq 0$, and test set $T \subset \mathcal{M}$, find a trajectory $\gamma : [0, 1] \rightarrow \mathcal{M}_{\text{free}}$ that is a solution to:*

$$\begin{aligned}
 & \min_{\gamma} \quad J_M(\gamma) \\
 & \text{subject to} \quad \gamma(0) = s, \quad \gamma(1) = g, \\
 & \quad \quad \quad \text{Length}(\gamma) \leq B
 \end{aligned} \tag{4.5}$$

Figure 4.1 illustrates the problem presented in this section.

4.4 Hierarchical Optimization Approach Outline

To solve Problem 1 efficiently in complex environments, we adopt a hierarchical optimization strategy that decomposes the planning task into three optimization stages: discrete graph-based planning, local budget allocation, and continuous trajectory improvement, as demonstrated in Figure 4.2. In the first stage, we identify a sequence of vertices on

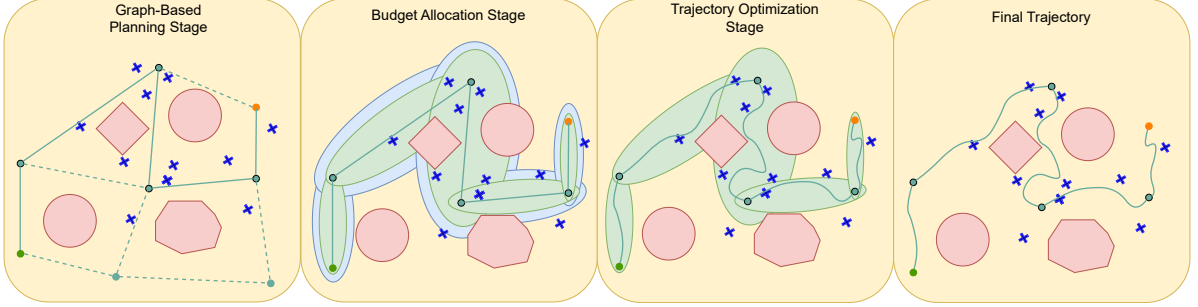


Figure 4.2: Hierarchical IPP Strategy

a pre-defined graph that minimizes the GP-based objective. This high-level path guides subsequent stages of the optimization process.

Using the global optimization method proposed in [21], we obtain a graph path that serves as a guiding initialization. In the second stage, we allocate the total trajectory budget across the path segments. This step is critical, as different segments offer varying potential for local information gain. To perform the allocation, we define a reachable region based on the selected vertices from the graph stage. The reachable region, together with the kernel hyperparameters, determines an informative region where test points are likely to be influenced by measurements along the segment. The allocation is determined by maximizing the number of test points covered within the union of these informative regions.

Finally, given the allocated budgets, we refine each path segment using a NLP formulation based on splines. This continuous trajectory optimization stage adjusts the path locally to improve the measurements. To ensure efficiency, optimization is restricted to the reachable region defined for each segment.

4.5 Graph-Based Planning Stage

We adopt a similar setup to [21]. Let $G = (V, E)$ be a directed graph, where $V \subset \mathcal{M}_{\text{free}}$ is a finite set of collision-free vertices, and $E \subseteq V \times V$ is a set of directed edges, each representing a collision-free line segment between two vertices. Each edge $e = (u, v) \in E$ is assigned a traversal cost

$$\text{Length}(e) := \|v - u\|. \quad (4.6)$$

Given start and goal vertices $s, g \in V$, we define a path $\mathcal{P} := (v_0, v_1, \dots, v_k)$ as a sequence of connected vertices from s to g , where $v_0 = s$, $v_k = g$, and each edge $(v_{i-1}, v_i) \in E$. We introduce binary decision variables $\mathcal{X} = \{\mathcal{X}_e\}_{e \in E}$, where $\mathcal{X}_e = 1$ if edge e is included in the selected path, and $\mathcal{X}_e = 0$ otherwise. The edge set corresponding to $\mathcal{P}$ is given by

$$E_{\mathcal{P}} := \{(v_{i-1}, v_i) \in E \mid 1 \leq i \leq k\}. \quad (4.7)$$

The vertices in $\mathcal{P}$ represent measurement locations for the GP model. Using the posterior covariance matrix from Equation 4.3, the GP posterior covariance over test points T , conditioned on measurements at $\mathcal{P}$, is denoted $\Sigma_{T|\mathcal{P}}$. We define the objective for the graph problem as:

$$\tilde{J}_M(\mathcal{X}) := \text{Tr}(M\Sigma_{T|\mathcal{P}}). \quad (4.8)$$

We then formulate the following optimization problem:

$$\min_{\mathcal{X} \in \{0,1\}^{|E|}} \tilde{J}_M(\mathcal{X}) \quad (4.9)$$

$$\text{subject to} \quad \sum_{e \in E} \text{Length}(e) \mathcal{X}_e \leq B, \quad (4.10)$$

$$\sum_{e \in \delta^+(v)} \mathcal{X}_e - \sum_{e \in \delta^-(v)} \mathcal{X}_e = \begin{cases} 1 & \text{if } v = s, \\ -1 & \text{if } v = g, \\ 0 & \text{otherwise,} \end{cases} \quad \forall v \in V. \quad (4.11)$$

Constraint (4.10) enforces that the total path length does not exceed the available budget B , and constraint (4.11) ensures a valid s - g path via flow conservation. Since the original objective is non-convex in the binary decision variables $\mathcal{X}$, we adopt a convex reformulation based on a linear estimator for posterior predictions [21]. This yields a mixed-integer convex program that provides a globally optimal high-level path on the graph. However, it is clear that using the graph formulation, measurements are taken only at vertex locations. In practice, robots collect data continuously along their path. To bridge this gap, we refine each selected edge using a smooth spline and solve a continuous optimization problem over the resulting trajectories. A key challenge is how to allocate the total budget B across these segments prior to local improvement.

4.6 Budget Allocation Stage

In this section, we introduce a heuristic method for distributing the total refinement budget across the segments obtained from the graph-based plan. Given a global budget B and a

path $\mathcal{P}$ from the high-level planner, our goal is to assign local budgets to each segment in that path in a way that prioritizes those estimated to contribute more to uncertainty reduction. Since measurements are collected along refined trajectories, we aim to identify which segments can potentially influence more test points. To guide this allocation, we estimate the set of test points that lie within the informative region of each segment under varying local budgets, and then solve an optimization problem that selects the budget allocation maximizing overall coverage of test points in these informative regions.

4.6.1 Informative Region Bounding

To identify which test points may be influenced by a trajectory segment, we derive geometric bounds on the region reachable by any curve of bounded length between two endpoints. These results form the foundation of our budget allocation strategy and allow us to conservatively approximate the informative region associated with each edge. By leveraging arc length inequalities, elliptical containment, and kernel decay properties, we identify a subset of test points that are informative under a given budget for refinement. We begin by showing that the arc length of a curve always upper bounds the sum of distances from any point on the curve to its endpoints.

Lemma 1 (Arc Length Bounds via Triangle Inequality). *Let $\gamma : [0, 1] \rightarrow \mathbb{R}^d$ be a piecewise continuously differentiable curve. We define the arc length from a to b as*

$$\text{Length}(a \rightarrow b) := \int_a^b \|\gamma'(t)\| dt.$$

In particular, $\text{Length}(\gamma) := \text{Length}(0 \rightarrow 1)$.

Then:

1. $\|\gamma(1) - \gamma(0)\| \leq \text{Length}(\gamma)$
2. *For any $t \in [0, 1]$, we have*

$$\|\gamma(t) - \gamma(0)\| + \|\gamma(1) - \gamma(t)\| \leq \text{Length}(\gamma)$$

Proof. 1. By the Fundamental Theorem of Calculus:

$$\gamma(1) - \gamma(0) = \int_0^1 \gamma'(t) dt.$$

Taking the norm and applying the triangle inequality for integrals:

$$\|\gamma(1) - \gamma(0)\| \leq \int_0^1 \|\gamma'(t)\| dt = \text{Length}(\gamma).$$

2. For any $t \in [0, 1]$, we apply the same argument:

$$\|\gamma(t) - \gamma(0)\| \leq \int_0^t \|\gamma'(s)\| ds = \text{Length}(0 \rightarrow t),$$

$$\|\gamma(1) - \gamma(t)\| \leq \int_t^1 \|\gamma'(s)\| ds = \text{Length}(t \rightarrow 1).$$

Summing both:

$$\|\gamma(t) - \gamma(0)\| + \|\gamma(1) - \gamma(t)\| \leq \text{Length}(\gamma).$$

□

To formalize the region reachable by length-constrained curves, we define the elliptical region characterized by two points and a length bound (i.e., foci points and major axis length).

Definition 4 (Elliptical Region). *Given two points $u, v \in \mathbb{R}^d$ and a length bound $L > 0$, define the elliptical region*

$$\mathcal{E}(u, v, L) := \{x \in \mathbb{R}^d \mid \|x - u\| + \|x - v\| \leq L\}.$$

This is the set of all points whose sum of distances to u and v is no greater than L . In two dimensions, this corresponds to an ellipse with foci u and v and major axis length L .

The following result implies that any length-constrained trajectory between two points lies entirely within a corresponding ellipse.

Lemma 2 (Elliptical Containment of Length-Bounded Trajectories). *Let $u, v \in \mathbb{R}^d$, and let $\gamma : [0, 1] \rightarrow \mathbb{R}^d$ be a curve such that $\gamma(0) = u$, $\gamma(1) = v$, and $\text{Length}(\gamma) \leq L$. Then for all $t \in [0, 1]$, we have:*

$$\gamma(t) \in \mathcal{E}(u, v, L).$$

Proof. From Lemma 1, for any $t \in [0, 1]$,

$$\|\gamma(t) - u\| + \|\gamma(t) - v\| \leq \text{Length}(\gamma) \leq L.$$

Hence, $\gamma(t) \in \mathcal{E}(u, v, L)$ for all $t \in [0, 1]$.

□

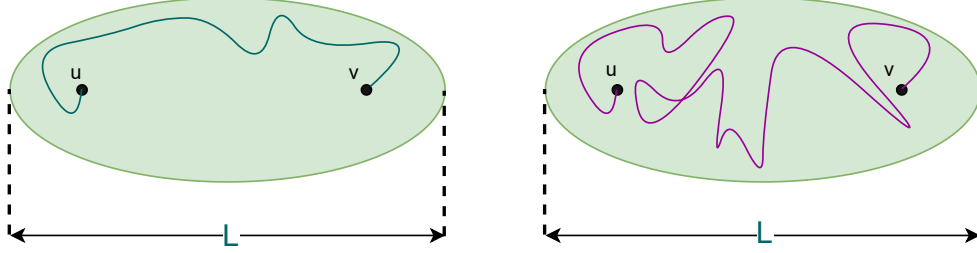


Figure 4.3: *Left:* Any trajectory starting at point u and ending at v with total length at most L is fully contained within the ellipse $\mathcal{E}(u, v, L)$, defined by foci u, v and major axis length L . *Right:* The converse does not hold: a trajectory that remains entirely within the ellipse can still exceed length L if it contains loops or detours.

Remark 4. *The converse does not hold: remaining within $\mathcal{E}(u, v, L)$ does not guarantee that a trajectory has length at most L . As illustrated by the right-hand trajectory in Figure 4.3, a curve may lie entirely inside the ellipse yet still exceed the length bound by following a trajectory that contains loops.*

To describe the region in which the squared exponential kernel has a non-negligible influence along a trajectory, we make use of the Minkowski sum between sets.

Definition 5 (Minkowski Sum). *Given two subsets $A, B \subset \mathbb{R}^d$, the Minkowski sum of A and B is defined as*

$$A \oplus B := \{a + b \mid a \in A, b \in B\}.$$

We now use this operation to express a bound on the region where measurements have a non-negligible impact on test points.

Lemma 3 (Kernel Influence Radius via Minkowski Sum). *Let $k(x, x') = \sigma^2 \exp\left(-\frac{\|x-x'\|^2}{2\ell^2}\right)$ be the squared exponential kernel. Fix a tolerance $\epsilon \in (0, \sigma^2)$, and define the kernel influence radius:*

$$r_{\text{kernel}} := \ell \sqrt{2 \log\left(\frac{\sigma^2}{\epsilon}\right)}.$$

Let γ be a curve satisfying $\text{Length}(\gamma) \leq L$, and let $\mathcal{E}(u, v, L)$ be the ellipse with foci $u = \gamma(0)$, $v = \gamma(1)$, and major axis length L . Then, for any point $x \notin \mathcal{E}(u, v, L) \oplus B(0, r_{\text{kernel}})$, we have:

$$k(x, \gamma(t)) < \epsilon \quad \text{for all } t \in [0, 1].$$

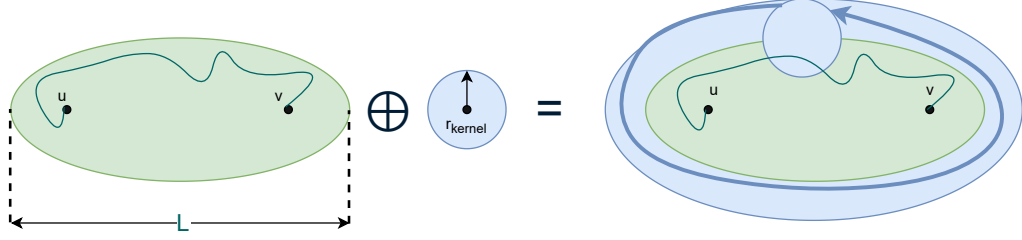


Figure 4.4: The Minkowski sum of the ellipse $\mathcal{E}(u, v, L)$ and a ball of radius r_{kernel} . The resulting region represents all points within distance r_{kernel} from some point on the ellipse.

Proof. From Lemma 2, we know $\gamma(t) \in \mathcal{E}(u, v, L)$ for all $t \in [0, 1]$. If $x \notin \mathcal{E}(u, v, L) \oplus B(0, r_{\text{kernel}})$, then:

$$\|x - \gamma(t)\| > r_{\text{kernel}} \quad \forall t.$$

Then:

$$k(x, \gamma(t)) = \sigma^2 \exp\left(-\frac{\|x - \gamma(t)\|^2}{2\ell^2}\right) < \sigma^2 \exp\left(-\frac{r_{\text{kernel}}^2}{2\ell^2}\right).$$

By definition of r_{kernel} , we have:

$$\exp\left(-\frac{r_{\text{kernel}}^2}{2\ell^2}\right) = \frac{\epsilon}{\sigma^2},$$

so:

$$k(x, \gamma(t)) < \sigma^2 \cdot \frac{\epsilon}{\sigma^2} = \epsilon.$$

□

Remark 5 (Negligible Effect of Distant Test Points). *Lemma 3 implies that any test point $\tau \in T$ lying outside the region $\mathcal{E}(u, v, L) \oplus B(0, r_{\text{kernel}})$ has negligible influence on the GP posterior, since its kernel value with respect to all trajectory points is less than ϵ . We therefore want to maximize the number of test points lying inside the union of all informative regions of all segments while allocating the budget.*

4.6.2 Max Coverage Problem for Local Budget Allocation

We now leverage the elliptical containment and kernel decay results to formulate a continuous optimization problem that allocates refinement lengths to each segment of the planned path. Recall that for an edge $e = (u, v) \in E_{\mathcal{P}}$, the informative region expands with the

allocated length L_e , as it determines the major axis of the ellipse $\mathcal{E}(u, v, L_e)$. By Lemma 2 and the kernel decay bound, the set of test points significantly influenced by edge e is contained in the Minkowski sum $\mathcal{E}(u, v, L_e) \oplus B(0, r_{\text{kernel}})$. To avoid computing this Minkowski sum explicitly, we instead use a conservative over-approximation:

$$\mathcal{R}_e := \{x \in \mathbb{R}^d \mid \|x - u\| + \|x - v\| \leq L_e + 2r_{\text{kernel}}\},$$

which enlarges the original ellipse by simply increasing its major axis by the kernel radius. Any test point $\tau_j \in T$ satisfying this inequality is guaranteed to lie within the true Minkowski sum. To enable smooth optimization, we introduce a differentiable approximation of whether a test point $\tau_j \in T$ is covered by $\mathcal{R}_e$. Let

$$d_{ej} := \|\tau_j - u\| + \|\tau_j - v\|, \quad (4.12)$$

$$s_{ej}(L_e) := \frac{1}{1 + \exp(\alpha(d_{ej} - L_e - 2r_{\text{kernel}}))}, \quad (4.13)$$

where $\alpha > 0$ controls the sharpness of the sigmoid. The score $s_{ej}(L_e) \in (0, 1)$ serves as a soft indicator for whether τ_j is covered by edge e . We then define the total coverage score for each test point as

$$z_j := 1 - \prod_{e \in E_{\mathcal{P}}} (1 - s_{ej}(L_e)), \quad (4.14)$$

which approximates whether τ_j is covered by at least one informative region.

The budget allocation problem is then formulated as:

$$\begin{aligned} & \underset{\{L_e\}}{\text{maximize}} && \sum_{j=1}^m M_{jj} z_j \\ & \text{subject to} && L_e \geq \|u - v\|, \quad \forall e \in E_{\mathcal{P}}, \\ & && \sum_{e \in E_{\mathcal{P}}} L_e \leq B. \end{aligned} \quad (4.15)$$

The output of this optimization problem is a set of budget lengths $\{L_e\}$ for each segment, which subsequently serve as constraints in the trajectory refinement stage. These budget lengths also correspond to the major axes of the ellipses $\mathcal{E}(u, v, L_e)$, as established by our earlier geometric results. This connection is not a result of coincidence, as increasing the budget for a segment enlarges its associated reachable region, thereby expanding the informative region and enabling influence over more test points. Consequently, the optimization problem we posed as a coverage maximization problem is, in essence, a method for distributing the total budget across the path segments.

4.7 Trajectory Optimization Stage

Given the ordered sequence of edges $E_{\mathcal{P}}$ corresponding to the graph path $\mathcal{P}$, and their associated local refinement budgets $\{L_e\}_{e \in E_{\mathcal{P}}}$, we refine each edge $e = (u, v) \in E_{\mathcal{P}}$ into a smooth parametric segment $\gamma_e : [0, 1] \rightarrow \mathcal{M}_{\text{free}}$. Each segment is optimized *independently*, with the goal of reducing the GP posterior uncertainty over the test set T , while respecting the assigned budget L_e and avoiding obstacles. We represent each segment as a degree- d spline curve defined by $r + 1$ control points $P^e = [P_0^e, \dots, P_r^e] \in \mathbb{R}^{d \times (r+1)}$. The segment is expressed as:

$$\gamma_e(t) = \sum_{i=0}^r \Psi_i(t) P_i^e, \quad t \in [0, 1], \quad (4.16)$$

where $\Psi_i(t)$ are fixed basis functions determined by the spline type. For example, Bézier curves use Bernstein polynomials, while B-splines use piecewise polynomial basis functions defined by a knot vector. To remain general, we refer to the basis abstractly as $\Psi_i(t)$. To ensure C^0 continuity, we clamp the spline endpoints

$$P_0^e = u, \quad P_r^e = v. \quad (4.17)$$

Only the interior control points $\{P_1^e, \dots, P_{r-1}^e\}$ are treated as decision variables. We then evaluate the objective by sampling each segment at N uniformly spaced parameter values $\{t_1, \dots, t_N\} \subset [0, 1]$, yielding the set of measurement locations:

$$X_{\text{traj}}^e := \{\gamma_e(t_i)\}_{i=1}^N. \quad (4.18)$$

Since the trajectory $\gamma_e(t)$ is expressed as a linear combination of the control points $\{P_i^e\}$ using fixed basis functions $\{\Psi_i(t)\}$, each sampled location $\gamma_e(t_i) \in X_{\text{traj}}^e$ is a function of the control points. As a result, from Equation 4.3 the GP posterior covariance of T conditioned on sampled trajectory measurements X_{traj}^e denoted as $\Sigma_{T|X_{\text{traj}}^e}$ is also a function of the control points. The objective is then defined as

$$\bar{J}_M(P^e) := \text{Tr}(M \Sigma_{T|X_{\text{traj}}^e}), \quad (4.19)$$

In order to enforce feasibility, we impose two constraints. Firstly, the trajectory arc length must satisfy:

$$\text{Length}(\gamma_e) \leq L_e. \quad (4.20)$$

Secondly, the trajectory must remain collision-free. Ideally, we require

$$\phi(\gamma_e(t)) \geq 0 \quad \text{for all } t \in [0, 1], \quad (4.21)$$

where $\phi : \mathcal{M} \rightarrow \mathbb{R}$ is a signed distance function satisfying $\phi(x) \geq 0$ if and only if $x \in \mathcal{M}_{\text{free}}$. This formulation is valid for convex obstacles, for which such a function can be computed efficiently. In practice, we enforce this condition at a finite set of observation points as:

$$\phi(\gamma_e(t_j)) \geq 0, \quad \forall j \in \{1, \dots, N_{\text{obs}}\}. \quad (4.22)$$

We thus formulate the trajectory optimization for each segment $e = (u, v)$ as the following NLP:

$$\begin{aligned} \min_{P^e} \quad & \bar{J}_M(P^e) \\ \text{subject to} \quad & \text{Length}(\gamma_e) \leq L_e, \\ & \phi(\gamma_e(t_j)) \geq 0, \quad \forall j \in \{1, \dots, N_{\text{obs}}\}, \end{aligned} \quad (4.23)$$

Remark 6 (Obstacle Pruning for Efficiency). *To reduce computational overhead, we exclude obstacles that do not intersect the elliptical region $\mathcal{E}(u, v, L_e)$ associated with a segment $e = (u, v)$. By Lemma 2, the refined trajectory is guaranteed to lie entirely within this region. As a result, such obstacles should not affect the optimization problem and can be safely ignored, reducing constraint evaluations and improving solver efficiency.*

Smoothness and Solver Compatibility. Second-order solvers such as IPOPT [80] require the objective and constraints in (4.23) to be at least twice continuously differentiable with respect to the optimization variables. This condition is satisfied under our formulation. Bézier curves are constructed as convex combinations of control points using Bernstein polynomial basis functions, which define a single polynomial over the full interval $[0, 1]$. As a result, Bézier curves are infinitely differentiable (C^∞) in both the parameter and the control points. B-spline curves, in contrast, are composed of multiple Bézier segments joined at knots. While they are also convex combinations of control points, their basis functions are only piecewise polynomial, and the resulting curve is typically C^{d-1} continuous, depending on the degree d and the knot spacing. For example, cubic B-splines are C^2 , which is sufficient for second-order optimization.

The GP posterior is smooth in the measurement locations due to the differentiability of the kernel and matrix operations. The arc length and obstacle avoidance constraints are continuous and piecewise differentiable, and may be smoothed if needed to improve optimization stability. Overall, the entire formulation is sufficiently smooth for practical use with IPOPT.

Remark 7. *Although many continuous-space IPP methods rely on derivative-free solvers such as Covariance Matrix Adaptation Evolution Strategy (CMA-ES), our approach sup-*

ports efficient gradient-based optimization through symbolic spline representations and closed-form GP expressions.

Figure 4.5 illustrates two optimized Bézier trajectories in free space, generated using the solution to the trajectory optimization problem (4.23). Each trajectory collects 20 measurements from a cost field sampled from a GP. The red trajectory is constrained by a smaller budget, limiting its reach and resulting in fewer measurements near test points. The yellow trajectory, with a larger budget, is able to loop and position measurements more effectively near test points. Figure 4.6 shows the corresponding posterior mean and variance maps obtained via the GP update equations (3.2). It also includes posterior maps generated from ideal measurements taken directly at the test points. The mean maps reflect predicted values of the underlying field, while the variance maps quantify uncertainty. As expected, variance decreases significantly near the locations where measurements were taken.

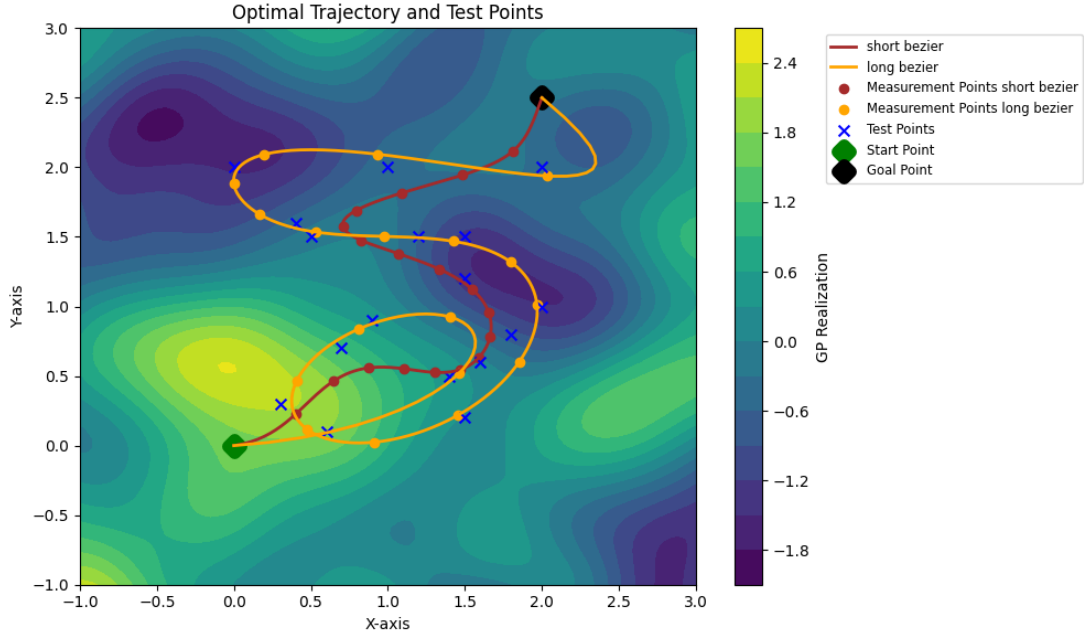


Figure 4.5: Example of two optimized Bézier curves in free space, both with 20 measurement points. The yellow trajectory is given a higher budget B than the red, allowing it to loop and better cover informative test points (blue). The underlying cost field is sampled from a known GP.

Figure 4.7 shows examples of optimized B-spline trajectories in an obstacle-cluttered

environment. B-splines are used in this setting instead of Bézier curves due to their local control property, as discussed in Section 3.3. In the left panel, the trajectory successfully navigates around obstacles while taking measurements close to test points. However, because the underlying problem is an (NLP), the solution may get trapped in local minima. This is illustrated in the right panel, where a slight change in the start location results in a suboptimal trajectory that fails to explore informative regions and instead wastes budget in less useful areas. This motivates the need to use our hierarchical framework instead of solely relying on continuous optimization methods.

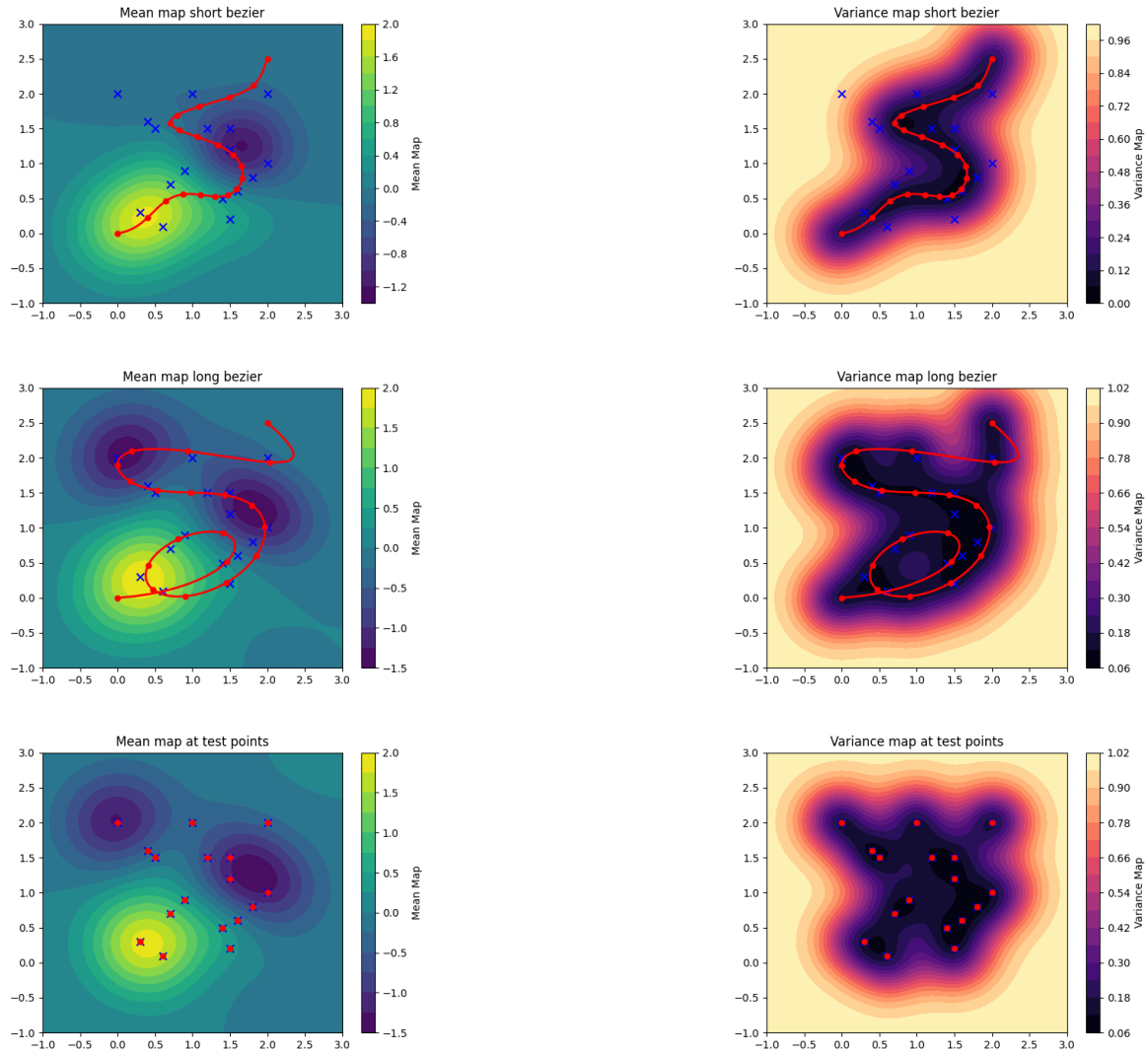
4.8 Simulation Results

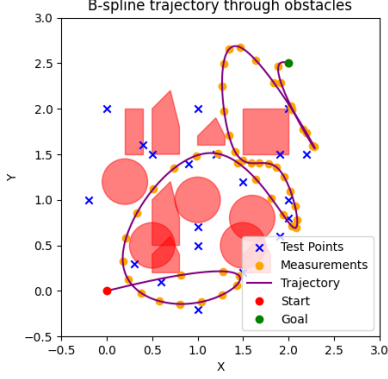
To evaluate our proposed method, we present results across a range of scenarios, including simulations in custom-built environments, experiments using real-world datasets, and simulations on real-world geographic maps. We specifically compare our approach against the continuous-space method introduced in [32], which uses CMA-ES as a black-box optimizer. Since CMA-ES does not support hard constraints, their original formulation incorporates budget constraints as penalty terms in the objective function. Obstacles were not considered in their work, so we extended the approach to include obstacle avoidance for a fair comparison. We use the Python package provided in [59] to implement CMA-ES planner in our experiments. To further assess performance, we compare against the graph-based planner proposed in [21], which formulates the problem as a Mixed Integer Convex Program (MICP) and is implemented using Gurobi 11.0 [27]. This same method is used as the graph-based stage planner in our hierarchical framework, as described in Section 4.5. For solving the budget allocation problem (Section 4.6.2) and the continuous trajectory optimization problem (Section 4.7), we use CasADi [3] together with the IPOPT solver. All experiments were run on a computer with an Intel Core i7-10700K CPU and 32GB RAM.

Although originally proposed to maximize mutual information under a soft budget constraint, the approach of [32] is objective-agnostic and uses black-box optimization. This flexibility allows us to adopt their structure and replace the objective with our variance-based criterion, while also incorporating obstacle and boundary penalties using the same penalty-based formulation. Specifically, we minimize

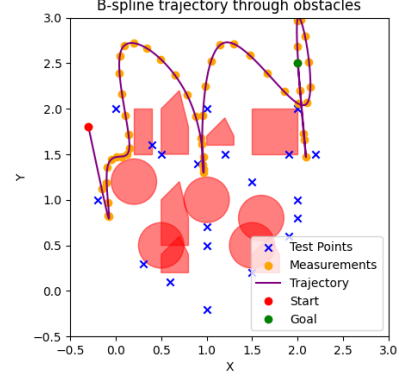
$$J(P) = \rho(B - \text{Length}(\gamma)) \cdot \rho(\phi_{\text{obs}}(\gamma)) \cdot \text{Tr}(M\Sigma_{T|X}), \quad (4.24)$$

where $\text{Length}(\gamma)$ denotes the length of the spline trajectory γ defined by control points P , and $\phi_{\text{obs}}(\gamma)$ is a penalty for obstacle violations, respectively. The smooth penalty function





(a) A successful trajectory in which the B-spline curve avoids obstacles and positions measurements close to informative test points.



(b) Suboptimal trajectory in which the optimizer converges to a poor local minimum, wasting budget in low-information areas.

Figure 4.7: Two B-spline trajectory optimization outcomes in an obstacle-rich environment with 12 control points. Both trajectories are subject to the same constraints, but the non-convex nature of the problem leads to different local solutions sensitive to problem parameters.

$\rho: \mathbb{R} \rightarrow \mathbb{R}_{>0}$ is defined as:

$$\rho(z) = \begin{cases} \exp\left(-\frac{z}{2\lambda^2}\right), & z < 0 \\ 1, & z \geq 0 \end{cases} \quad (4.25)$$

where $\lambda > 0$ is a user-defined smoothness parameter that controls the softness of the constraint. This formulation follows the constraint-handling approach of [32], enabling soft enforcement of budget and feasibility constraints within the CMA-ES framework.

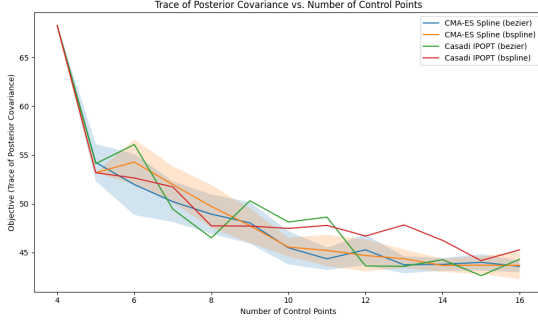
4.8.1 Comparisons in obstacle-free environments

We begin with comparisons in the obstacle-free setting.

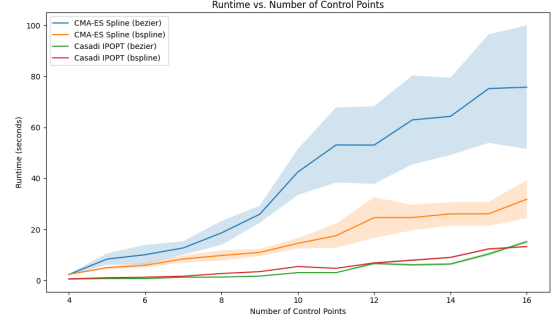
Setup 1. In the first experiment, we evaluate only the trajectory optimization stage proposed in Section 4.7, implemented using CasADi with IPOPT, against the offline version

of the continuous planner from [32], which uses CMA-ES. Both methods are tested using Bézier curves and degree-3 B-splines with a uniform knot vector. The environment is a 10×10 square, with kernel hyper-parameters set to length-scale $\ell = 1$, signal variance $\sigma^2 = 1$, and noise variance $\sigma_\epsilon^2 = 0.1$. We use $m = 100$ randomized test points with equal weighting, i.e., $M = I$, which is adopted in all experiments. Each trajectory produces $n = 20$ measurements. Start and goal points are fixed at the lower-left and upper-right corners of the domain, i.e., at the origin and $[10, 10]$, respectively. Both solvers are given a maximum of 2000 iterations, and the travel budget is set to $B = 40$. To evaluate scalability, we vary the number of control points from 4 (the minimum required to define a degree-3 B-spline) up to 16. For each configuration, we repeat the experiment 10 times using both solvers. This repetition is necessary due to the stochastic nature of CMA-ES, which produces different solutions for the same instance.

Results. As expected, increasing the number of control points generally improves the objective value for both methods, since more control points allow for better control over the trajectory shape. Overall, performance between the two approaches is comparable, with slight alternation in which method yields the better result. The key advantage of our approach is that it is deterministic, producing consistent results across runs, while the baseline exhibits higher variance, as reflected in the confidence intervals shown in Figure 4.8(left). In terms of runtime, our method is significantly faster and scales better with the number of control points, avoiding the steep runtime growth seen with CMA-ES (Figure 4.8(right)). This efficiency stems from our use of gradient-based optimization, which exploits the smoothness of both the objective and the constraints, in contrast to black-box methods like CMA-ES that rely on repeated sampling across a population of candidate solutions



(a) Objective value vs. number of control points.



(b) Runtime vs. number of control points.

Figure 4.8: Comparison of continuous-space IPP using CasADi with IPOPT (ours) and CMA-ES [32].

Setup 2. In the second experiment, we compare the trajectory optimization stage against a state-of-the-art graph planner [21], which formulates the problem as a mixed-integer convex program. We use an environment of size 10×10 , and randomize the kernel parameters: the length-scale ℓ is sampled between 0.8 and 1.5, the signal variance σ^2 between 0.1 and 2.0, and the noise variance σ_ϵ^2 between 0.05 and 0.1. The start and goal points are also randomized. For the graph planner, we consider two 4-connected grids with different discretizations. The coarse grid has a vertex spacing of 2, resulting in $6 \times 6 = 36$ vertices, while the fine grid uses spacing 1, yielding $11 \times 11 = 121$ vertices. Since the mixed-integer formulation operates on directed graphs, each undirected edge must be represented twice. The number of directed edges is given by:

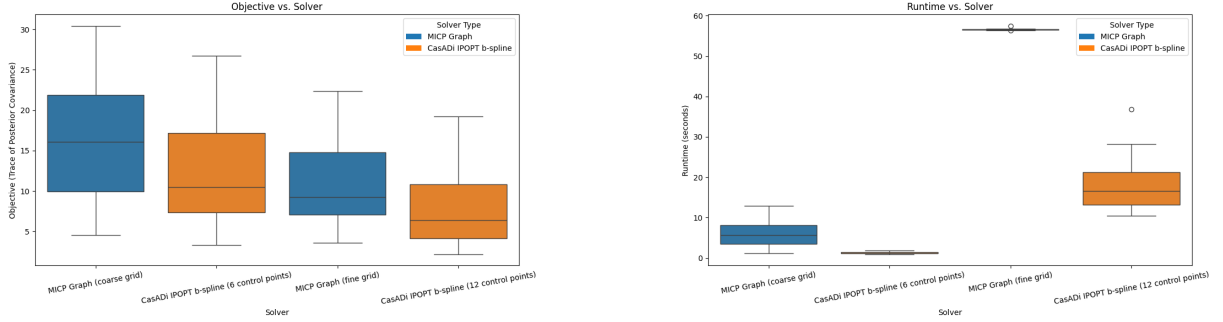
$$2 \times [(n-1) \cdot n + n \cdot (n-1)] = 4n(n-1),$$

where $n = L/\delta + 1$ is the number of vertices per side in a grid of size $L \times L$ with spacing δ . This results in 120 and 440 directed edges for the coarse and fine grids, respectively. Due to the increased combinatorial complexity in the fine setting, we limit the number of test points to $m = 25$, as this was the most complex configuration we could handle within a reasonable computational time.

For the trajectory optimization, we use B-spline curves and, similar to the graph planner, evaluate two versions: one with a low number of control points (6) and one with a higher number (12). The graph-based planner collects measurements only at the vertices it visits. After each run, we count the number of visited vertices and allow the trajectory to take the same number of measurements to ensure a fair comparison. This is important,

as the number of measurements significantly affects uncertainty reduction. Both planners are given a fixed travel budget of $B = 30$, and we run 25 different randomized runs.

Results. Our casadi-ipopt-based trajectory optimization outperforms the graph-based MICP planner in both estimation accuracy and computation time. In the fine-resolution setting, the 12-point spline achieves a lower median posterior trace (6.4) than the fine graph planner (9.3), and performs more than twice as well as the coarse graph planner (16.1). Even with just 6 control points, our method still outperforms the coarse graph in both objective and speed. In terms of runtime, the difference is even more apparent. The fine graph planner takes over 56 seconds per run, while our optimizer with 12 control points solves the problem in under 17 seconds. In the coarse case, our method is over four times faster, solving in about 1 second compared to nearly 6 seconds for the graph. These results, summarized in Figure 4.9, show that using smooth splines and gradient-based solvers not only improves performance but also makes the method more scalable and efficient than discrete graph-based approaches.



(a) Posterior trace with varying discretization.

(b) Runtime

Figure 4.9: Comparison between graph-based MICP and casadi-ipopt (ours) in terms of posterior uncertainty and runtime.

4.8.2 Comparisons in obstacle-cluttered environments

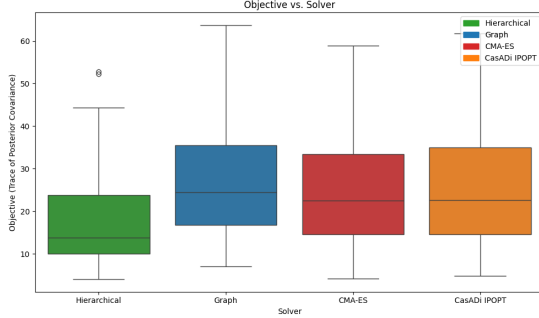
In the previous experiments, we compared solvers in obstacle-free environments. We now consider a more challenging, obstacle-cluttered scenario where pure trajectory optimization becomes less reliable due to the sensitivity of NLPs solvers to the geometry of the

environment. As demonstrated earlier, such solvers can easily become trapped in poor local minima. In this section, we evaluate our proposed hierarchical framework against two baselines: a graph-based MICP planner and a continuous-space solver using CMA-ES. We also include our CasADi-based solver with cubic B-splines as a standalone trajectory optimization baseline. The experimental setup follows the environment shown in Figure 4.7.

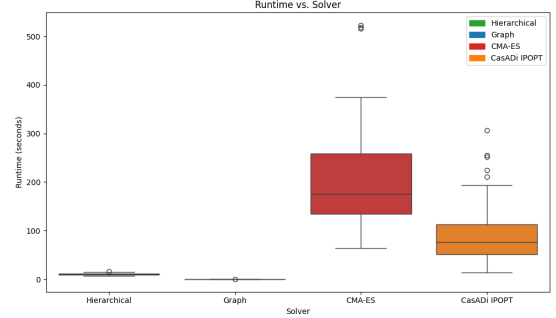
Setup 3. We use the same setup in Figure 4.7. The domain is bounded within $[-0.5, 3] \times [-0.5, 3]$ and contains twelve obstacles, comprising convex polygons and circles. A fixed graph with 11 vertices and 22 directed edges is used for both the hierarchical and graph-based planners. In each run, start and goal locations are randomly sampled from the graph vertices, ensuring a minimum separation of 1.0. We use a squared exponential kernel with length-scale $\ell = 0.35$, signal variance $\sigma^2 = 10$, and measurement noise $\sigma_\epsilon^2 = 0.1$. The planning budget is fixed at $B = 14$. We use a total of $m = 35$ test points, randomly generated. For fair comparison, all planners are constrained to use the same number of measurements per run. This number is determined dynamically based on the high-level plan produced by the hierarchical planner, which allocates a fixed number of measurements per segment. For the graph-based planner, we allow measurements to be placed along edges, even though optimization occurs only over vertices. Once the graph planner selects a path, measurements are uniformly distributed along the corresponding edges. This ensures a fair comparison, as the number of measurements directly affects posterior uncertainty, as discussed earlier.

Results. The proposed hierarchical planner outperforms all other methods in terms of both objective value and runtime. As shown in Figure 4.10, the hierarchical planner achieves the lowest mean objective (17.8), compared to 25.1 for CMA-ES, 25.8 for casadi-ipopt, and 27.4 for the graph MICP planner. It also has a smaller interquartile range, indicating more consistent performance.

The graph-based planner achieves the fastest runtime (mean of 0.27s), which is expected because it solves a combinatorial problem on a sparse graph. The hierarchical planner, while slower than the graph baseline, is significantly faster than both continuous-space solvers, CMA-ES (204.2s) and IPOPT (92.0s), with an average runtime of only 10.2s. Despite including a graph search stage, the hierarchical planner runs faster than standalone trajectory optimizers because it decomposes the problem into smaller subproblems and restricts trajectory optimization to elliptical regions. This reduces the number of obstacle constraints and speeds up convergence.



(a) Objective values across solvers.



(b) Runtime.

Figure 4.10: Performance comparison in obstacle-cluttered environments.

4.8.3 Real World Data

Setup 4. We evaluate our method on real-world sea-ice cover data in the Arctic, focusing on the Canadian Arctic Archipelago. The data is obtained from the ERA5 post-processed daily statistics on single levels dataset, provided by the Copernicus Climate Change Service through the Climate Data Store [14]. The dataset offers daily estimates of sea-ice cover, defined as the fraction of each grid cell covered by sea ice, based on satellite observations in April 2025. To obtain land and water boundaries, we use low-resolution vector geometry from the Cartopy library [55], which sources data from Natural Earth. Since both the ERA5 and land geometry data are in geographic coordinates (latitude–longitude), we reproject them onto a locally appropriate planar coordinate system (e.g., polar stereographic) to support accurate spatial modelling and path planning. Because our framework is limited to convex obstacles, we preprocess the landmasses using CGAL’s `Partition_2` package[31] to perform minimum convex decomposition. This converts non-convex polygons into 93 convex polygons, making them compatible with our constraint formulation. The planning scenario models an autonomous surface vessel navigating from a given start to a goal location to collect informative measurements of sea-ice cover along its trajectory. While sea-ice cover is already observable via satellite, other variables (e.g., ozone, salinity, ice thickness) often require on-site sensing for more accurate data. This scenario, therefore, still provides a realistic setting to demonstrate the applicability of our method. We use GPyTorch [23] to learn kernel hyperparameters by maximizing the marginal likelihood, with a squared exponential kernel.

In this scenario, the user provides an importance field I as shown in the top-left panel of Figure 4.11, which assigns higher values to regions of interest. This field is normalized into

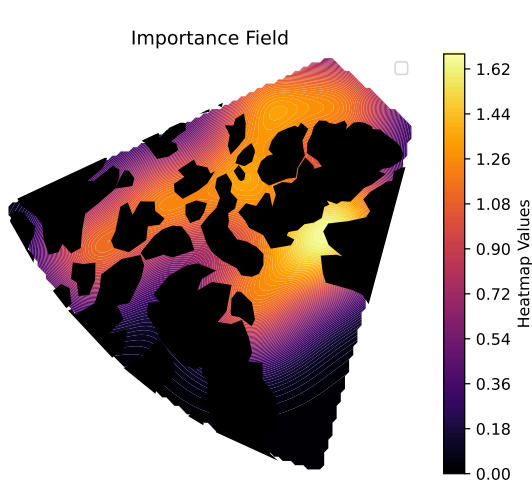
a probability density, from which $m = 30$ test points are sampled, constrained to lie in free space. Due to the large number of partitioned obstacles and the cluttered environment, pure trajectory optimization methods failed. CasADi-IPOPT terminated before finding a feasible solution, and CMA-ES returned a trajectory that collided with obstacles, with both being given 1000 iterations.

In contrast, our hierarchical optimization framework successfully generated feasible paths from start to goal while satisfying the budget and obstacle constraints and gathering informative measurements along the way. To evaluate estimation quality, we compare three strategies: (1) directly observing the test points, (2) using our planner with a low budget, and (3) using our planner with a higher budget. We assess performance using two metrics: RMSE at the test points, and an importance-weighted RMSE that incorporates the user-defined importance field.

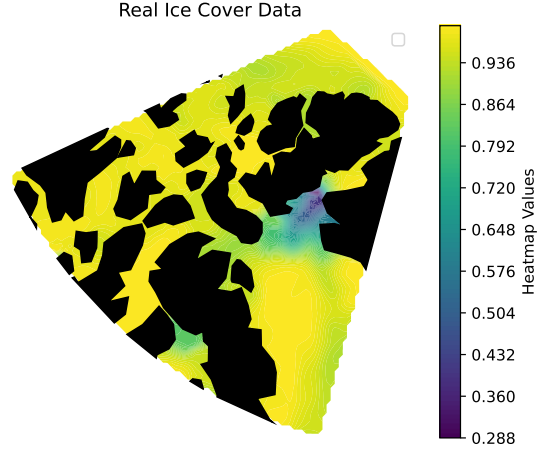
Importance-weighted RMSE. To evaluate approximation quality, we compute the root mean squared error weighted by a normalized importance distribution $\tilde{I}(x)$, where $\tilde{I}(x) = I(x)/\sum_j I(x_j)$. Letting $\mu(x)$ denote the posterior mean and $f(x)$ the ground truth, the metric is approximated over a grid $\{x_j\}_{j=1}^n$ as:

$$\text{Weighted RMSE} \approx \sqrt{\sum_{j=1}^n \tilde{I}(x_j) (\mu(x_j) - f(x_j))^2}. \quad (4.26)$$

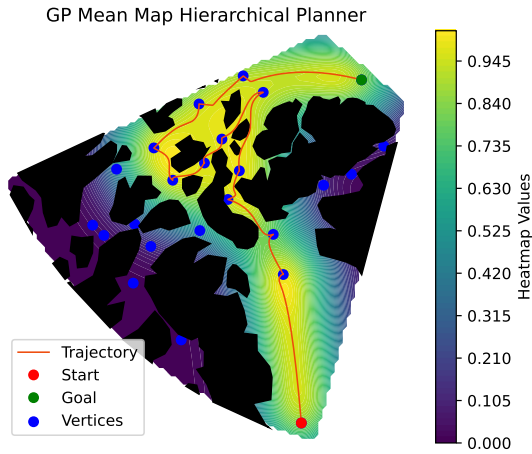
Results. Table 4.2 summarizes the estimation results under the different strategies. Direct test point observations yield the lowest RMSE at the test points, as expected, since the targets are observed directly (up to measurement noise). However, this approach is infeasible under the specified budget. Among the two planning strategies, increasing the budget significantly improves both RMSE and importance-weighted RMSE. Notably, the high-budget trajectory achieves a lower importance-weighted RMSE than direct test point observations, suggesting that it places measurements more effectively in regions that matter most to the user. Figure 4.11 demonstrates the real-data, and the trajectories produced by the hierarchical planner. Figure 4.12, shows the weighted mean squared error at each point.



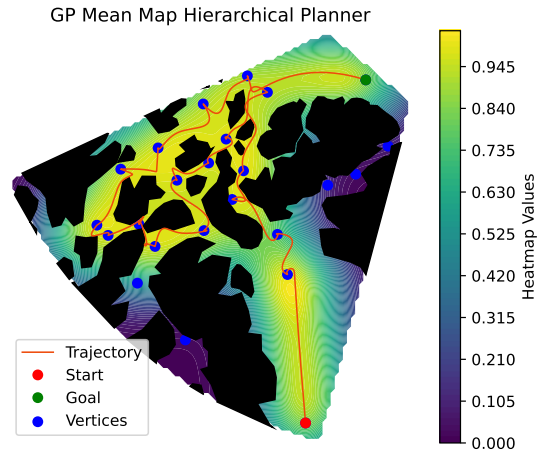
(a) Importance Field I



(b) Real Ice Cover from Satellite Imagery. Source:[14]



(c) Low budget hierarchical plan with GP posterior



(d) High budget hierarchical plan with GP posterior

Figure 4.11: Hierarchical IPP in the Canadian Arctic Archipelago. (a) The importance field I indicates regions of interest for reducing uncertainty. (b) Real ice cover data obtained from satellite observations [14]. (c)–(d) Planned paths using the hierarchical planner under low and high budget constraints, respectively, overlaid on the GP posterior mean. Obstacle boundaries are shown in black.

Table 4.2: Estimation performance under different observation strategies.

Method	RMSE at test points	Importance-weighted RMSE	Trace of posterior covariance
Direct test point observations	0.081	0.411	3.124
Hierarchical planner (low budget)	0.314	0.521	4.474
Hierarchical planner (high budget)	0.156	0.381	2.237

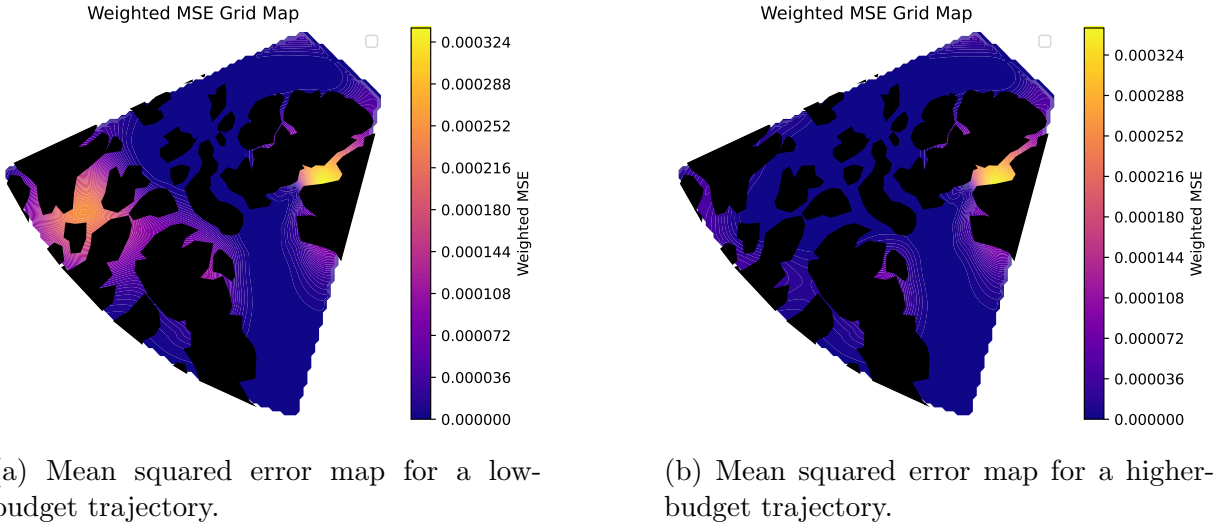


Figure 4.12: Posterior mean squared error maps under two different path budgets. A higher budget allows for more informative measurements, reducing estimation error. The error scale is small (on the order of 10^{-4}) due to the limited range of the measured quantity, as ice cover values lie between 0 and 1.

4.8.4 Evaluation in a Smaller Environment

While the Canadian Arctic Archipelago offers a compelling real-world testbed, its large geographical scale and harsh ice conditions make it impractical for most surface vessels, which would require icebreaker capabilities and long-range endurance to traverse such regions. To complement our main experiments, we therefore consider a smaller region with a lower number of obstacles.

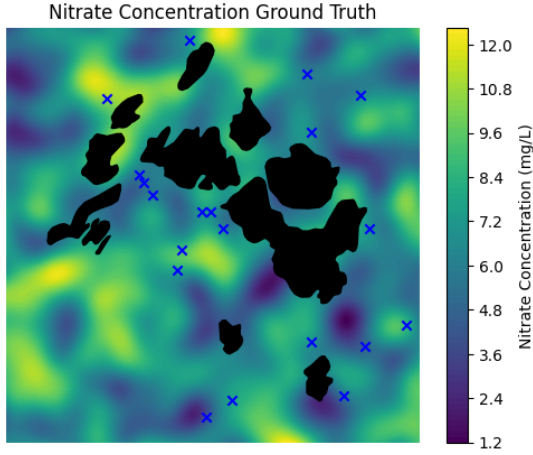
Setup 5 The environment is based on OpenStreetMap[60] coastline data for a region that includes islands in Central Ontario near Georgian Bay. The landmasses are enclosed using their convex hulls to simplify the environment and allow all planners to compute feasible trajectories. Unlike the Arctic case, no convex decomposition was required. Since real data at appropriate resolution is unavailable for this region, we generate synthetic environmental data using a GP with a squared exponential kernel. The resulting field represents the dispersion of water pollution, specifically the concentration of nitrates in mg/L. We also define a user importance field using a sum of unnormalized Gaussians to emphasize regions of interest.

All planners were given the same path length budget, and performance is reported using three metrics: RMSE at test points, importance-weighted RMSE, and the trace of the GP posterior covariance.

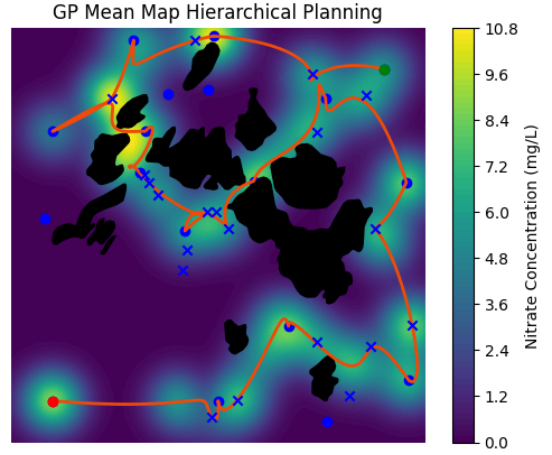
Results. Table 4.3 summarizes the results. The hierarchical planner again achieves the best estimation accuracy, both in standard and importance-weighted RMSE, while also being significantly faster than the other two methods. Figure 4.13 demonstrates the trajectories produced by each method.

Table 4.3: Estimation performance in the synthetic water pollution scenario.

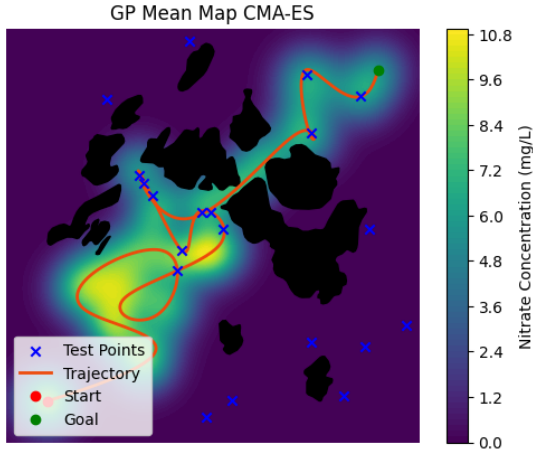
Method	RMSE at test points	Importance-weighted RMSE	Trace of posterior covariance	Time (s)
Hierarchical planner	2.13	4.51	13.68	18.64
CasADi-IPOPT	4.92	5.86	44.15	174.96
CMA-ES	4.79	5.71	37.12	282.33



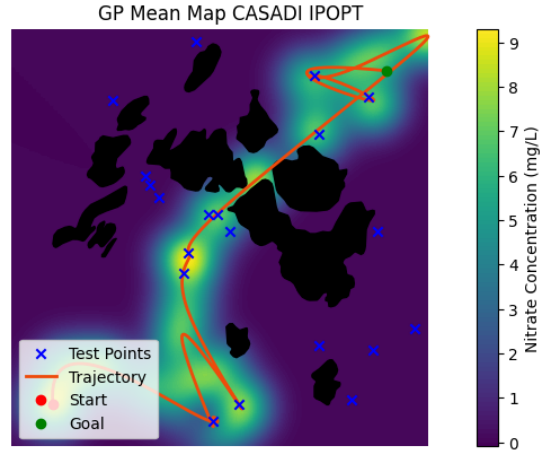
(a) Synthetic nitrate concentration field with test points.



(b) Planned trajectory using the hierarchical planner.



(c) Planned trajectory using the [CMA-ES](#) optimizer.



(d) Planned trajectory using CasADi with IPOPT.

Figure 4.13: [IPP](#) over a synthetic nitrate concentration field generated near islands in Central Ontario, close to Georgian Bay. (a) The true concentration field and test points are used for evaluating informativeness. (b)–(d) Trajectories generated by different planners: hierarchical planning, [CMA-ES](#), and CasADi-IPOPT, respectively, shown over the [GP](#) posterior mean. Obstacles are shown in black.

4.9 Summary

This chapter addresses the [IPP](#) problem in continuous space, where an agent must collect informative measurements of an unknown spatial field under resource constraints. We propose a hierarchical framework that combines graph-based planning with continuous spline trajectory optimization, enabling the agent to navigate complex environments with obstacles. A novel ellipse-based heuristic allocates trajectory budget across segments to prioritize informative regions. Simulations on both synthetic and real-world Arctic data demonstrate that our method consistently outperforms existing planners in estimation accuracy and runtime.

Chapter 5

Stochastic Path Planning in Random Fields

5.1 Introduction

In many robotics and autonomous systems applications, the environment is a priori unknown. In such settings, classical deterministic planning methods fall short because they fail to account for uncertainty in environmental variables. A common alternative is to model edge costs on a planning graph as random variables, as in the [Canadian Traveller Problem \(CTP\)](#) [62, 50, 77] and the [Stochastic Shortest Path with Recourse \(SSPPR\)](#) [63]. However, these approaches typically assume a discrete set of possible outcomes, which becomes impractical when the uncertainty arises from a continuous spatial process.

In this work, we consider the [Stochastic Path Planning in Random Fields \(SPPRF\)](#) problem, where uncertainty is induced by an underlying spatial field that directly influences traversal cost. Since such a field is continuous and its realizations cannot be enumerated, it is modelled as a stochastic process indexed by spatial coordinates. A motivating example is ship navigation through ice-covered waters, where captains of ice-breaking vessels must make routing decisions without full knowledge of the ice concentration or thickness. Even though these ships are capable of breaking ice, collisions with dense or thick ice remain hazardous and fuel-intensive, so navigators prefer routes through regions with lower values to minimize risk and energy use. Path planning in environments affected by random fields is ubiquitous across different domains. Beyond Arctic ship navigation, other motivating examples include:

- Flight planning in uncertain wind fields – Wind patterns affect fuel efficiency and flight time and exhibit strong spatial and temporal correlations.
- Autonomous underwater vehicle (AUV) navigation in ocean currents – Currents impact energy consumption and are correlated in both space and time.
- Agricultural robots in uncertain soil conditions – Soil moisture and density affect traversal effort and vary across the field.

In this work, we use a [Gaussian Process \(GP\)](#) to represent and update the agent’s belief over the underlying cost field based on real-time observations. As a nonparametric prior over continuous functions, a [GP](#) is particularly well-suited for modelling spatially correlated environmental variables. Unlike classical approaches that assign uncertainty directly to edge costs, we model the entire cost field as a stochastic process, reflecting the fact that the agent observes the field itself rather than the traversal costs on the edges. A key challenge in this formulation is deriving edge cost distributions from the [GP](#) posterior, enabling integration with existing graph-based stochastic planning methods. Specifically, given a sparse set of observations, how can we compute the induced distribution over edge costs to support informed path planning? This chapter focuses on planning paths through environments with uncertain, spatially varying traversal costs. The objective is to navigate from a start to a goal while minimizing the expected total cost.

5.2 Contributions

In this chapter, we address the [SPPRF](#) problem, where traversal costs are governed by a stochastic process modelled as a [GP](#) over continuous space. We introduce a framework for transforming the [GP](#) cost distribution in continuous space into a probability distribution over edge costs, forming what we call a belief graph. Leveraging this representation, we propose an adaptive planning approach that uses Dijkstra’s algorithm to compute the minimal expected-cost path given the current observations. As new observations are collected during traversal, the [GP](#) posterior and belief graphs are updated, allowing the planner to adaptively change the path in response to incoming information.

5.3 Problem Formulation

We consider an autonomous agent operating in an environment $\mathcal{M} \subset \mathbb{R}^d$. The environment is discretized into an undirected graph $G = (V, E)$, where $V \subset \mathcal{M}$ is a finite set of vertices

and $E \subseteq V \times V$ is a set of edges. Each edge $e = (u, v) \in E$ is geometrically associated with a continuous curve $\mathcal{C}_e \subset \mathcal{M}$. Traversal cost is determined by an unknown spatial field $f : \mathcal{M} \rightarrow \mathbb{R}$, representing quantities such as ice thickness, wind resistance, or ocean current strength. This field is modelled as a realization of a stochastic process. While the agent does not directly observe f , it may obtain noisy pointwise measurements during traversal.

Assumption 1 (Static Stochastic Cost Field). *The cost field f is unknown but time-invariant.*

Assumption 2 (Noisy Local Observations). *The agent can collect a noisy measurement of the cost field f at its current location $\mathbf{x} \in \mathcal{M}$.*

To evaluate the accumulated cost along a continuous path in a spatial domain, we use the concept of a line integral, which generalizes the idea of summing a scalar field along a curve. We recall its formal definition:

Definition 6 (Line Integral, [75]). *Let $\mathcal{C} \subset \mathbb{R}^d$ be a smooth curve parametrized by $\mathbf{x} : [a, b] \rightarrow \mathcal{C}$, and let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a scalar field. The line integral of f along $\mathcal{C}$ is defined as:*

$$\int_{\mathcal{C}} f(\mathbf{x}) ds := \int_a^b f(\mathbf{x}(t)) \|\dot{\mathbf{x}}(t)\| dt,$$

where $ds = \|\dot{\mathbf{x}}(t)\| dt$ denotes the differential arc length.

This integral enables the evaluation of accumulated values of a scalar field f along paths in arbitrary subsets of $\mathbb{R}^d$. We define the cost of traversing a curve $\mathcal{C} \subset \mathcal{M}$ as the line integral of the cost field:

$$T_{\mathcal{C}}(f) := \int_{\mathcal{C}} f(\mathbf{x}) ds \tag{5.1}$$

where ds denotes arc length, and the integral is a line integral in Definition 6. For a path $P = \{e_1, e_2, \dots, e_k\}$ from start node $s \in V$ to goal node $g \in V$, the total cost is:

$$T_P(f) := \sum_{i=1}^k T_{\mathcal{C}_{e_i}}(f) = \sum_{i=1}^k \int_{\mathcal{C}_{e_i}} f(\mathbf{x}) ds. \tag{5.2}$$

Since f is random, $T_P(f)$ is a random variable. The planning objective is to find a path that minimizes the expected total cost, and the problem is formally stated as:

Problem 2 (Stochastic Path Planning in Random Fields). *Given a stochastic process prior over the cost field f , a spatial graph $G = (V, E)$, and start and goal nodes $s, g \in V$, find a path $P \subseteq E$ from s to g that minimizes:*

$$\min_{P \in \mathcal{P}_{s \rightarrow g}} \mathbb{E}_f [T_P(f)], \quad (5.3)$$

where $\mathcal{P}_{s \rightarrow g}$ denotes the set of all feasible paths from s to g , and the expectation is taken over the distribution of f .

Remark 8. *Although the objective is expressed over static paths, solving the problem typically requires the agent to update its belief about the field as new observations are collected. This motivates the use of adaptive planning policies, which we introduce in the methodology section.*

5.4 Modelling

We model the unknown cost field $f : \mathcal{M} \rightarrow \mathbb{R}$, where $\mathcal{M} \subset \mathbb{R}^d$, as a realization of a GP with prior mean function $m_0(\cdot)$ and kernel $k(\cdot, \cdot)$, i.e. $f \sim \mathcal{GP}(m_0, k)$. The agent collects noisy observations over time. At time t , it has obtained a dataset $\mathcal{D}_{\leq t} = (\mathbf{X}_{\leq t}, \mathbf{Y}_{\leq t})$, where $\mathbf{X}_{\leq t} = [\mathbf{x}_1, \dots, \mathbf{x}_t]^\top \in \mathbb{R}^{t \times d}$ is the set of queried locations with $\mathbf{x}_i \in \mathcal{M}$, and $\mathbf{Y}_{\leq t} = [y_1, \dots, y_t]^\top \in \mathbb{R}^t$ are the corresponding noisy observations:

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2) \quad \text{i.i.d.} \quad (5.4)$$

Given this data, the posterior over f is a Gaussian process:

$$f \mid \mathcal{D}_{\leq t} \sim \mathcal{GP}(m_t(\cdot), k_t(\cdot, \cdot)),$$

with posterior mean and covariance functions:

$$\begin{aligned} m_t(\mathbf{x}) &= m_0(\mathbf{x}) + K(\mathbf{x}, \mathbf{X}_{\leq t}) [K(\mathbf{X}_{\leq t}, \mathbf{X}_{\leq t}) + \sigma_\epsilon^2 I]^{-1} (\mathbf{Y}_{\leq t} - m_0(\mathbf{X}_{\leq t})), \\ k_t(\mathbf{x}, \mathbf{x}') &= k(\mathbf{x}, \mathbf{x}') - K(\mathbf{x}, \mathbf{X}_{\leq t}) [K(\mathbf{X}_{\leq t}, \mathbf{X}_{\leq t}) + \sigma_\epsilon^2 I]^{-1} K(\mathbf{X}_{\leq t}, \mathbf{x}'), \end{aligned} \quad (5.5)$$

where $K(\mathbf{x}, \mathbf{X}_{\leq t}) \in \mathbb{R}^{1 \times t}$ is the vector of kernel evaluations between $\mathbf{x} \in \mathcal{M}$ and each point in $\mathbf{X}_{\leq t}$.

Remark 9. *Under Assumptions 1 and 2, the true cost function f is a fixed, time-invariant realization of a stochastic process, while observations are noisy and collected incrementally. As a result, the posterior mean $m_t(\cdot)$ and covariance $k_t(\cdot, \cdot)$ evolve with new data, reflecting the agent's improved belief. However, the underlying environment remains static, and any changes in predicted traversal cost stem from information gain, not environmental change.*

5.4.1 Stochastic Cost of an Arbitrary Curve

Let $\mathcal{C} \subset \mathcal{M}$ be a curve parameterized by a continuously differentiable injective function $\mathbf{x} : [a, b] \rightarrow \mathcal{C}$. We recall that the cost of traversing $\mathcal{C}$ is given by Equation 5.1:

$$T_{\mathcal{C}} = \int_{\mathcal{C}} f(\mathbf{x}) ds, \quad (5.6)$$

where $ds = \|\dot{c}(\tau)\| d\tau$ is the differential arc length. Since $f \sim \mathcal{GP}(m_t(\cdot), k_t(\cdot, \cdot))$, and each $f(x)$ is a random variable, the total cost $T_{\mathcal{C}}$ is itself a random variable.

Remark 10. *Since the cost field f is modeled as a stochastic process, the cost quantities $T_{\mathcal{C}}(f)$ and $T_P(f)$ are random variables. For conciseness, we omit the explicit dependence on f and write $T_{\mathcal{C}}$ and T_P throughout the chapter. This notation reflects their interpretation as random variables induced by the underlying cost field.*

Lemma 4. *At time $t \in \mathbb{N}_{\geq 0}$, the traversal cost $T_{\mathcal{C}}$ is a Gaussian random variable:*

$$T_{\mathcal{C}} \sim \mathcal{N}(\mu_{\mathcal{C}|t}, \sigma_{\mathcal{C}|t}^2),$$

where $\mu_{\mathcal{C}|t}$ is the expected cost (mean) and $\sigma_{\mathcal{C}|t}^2$ is the variance of the cost. They are given by:

$$\begin{aligned} \mu_{\mathcal{C}|t} &= \int_{\mathcal{C}} m_t(\mathbf{x}) ds, \\ \sigma_{\mathcal{C}|t}^2 &= \int_{\mathcal{C}} \int_{\mathcal{C}} k_t(\mathbf{x}, \mathbf{x}') ds ds'. \end{aligned} \quad (5.7)$$

Proof. We recall that $T_{\mathcal{C}} = \int_{\mathcal{C}} f(\mathbf{x}) ds$, and that $f \sim \mathcal{GP}(m_t, k_t)$. Since GPs are closed under linear operations [46], such as integration, $T_{\mathcal{C}}$ is a Gaussian random variable. Therefore, to characterize its distribution, it suffices to compute its mean and variance.

1) Mean. We begin with the expected value:

$$\mathbb{E}[T_{\mathcal{C}}] = \mathbb{E} \left[\int_{\mathcal{C}} f(\mathbf{x}) ds \right].$$

Since the integral is a linear operator and the GP mean function $m_t(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ is integrable over the curve $\mathcal{C}$, we apply linearity of expectation under the integral sign:

$$\mathbb{E}[T_{\mathcal{C}}] = \int_{\mathcal{C}} \mathbb{E}[f(\mathbf{x})] ds = \int_{\mathcal{C}} m_t(\mathbf{x}) ds.$$

2) Variance. To compute the variance, we use the fact that:

$$\sigma_{\mathcal{C}|t}^2 := \text{Var}[T_{\mathcal{C}}] = \mathbb{E} [(T_{\mathcal{C}} - \mathbb{E}[T_{\mathcal{C}}])^2].$$

Substituting the definition of $T_{\mathcal{C}}$, we get:

$$\text{Var}[T_{\mathcal{C}}] = \mathbb{E} \left[\left(\int_{\mathcal{C}} (f(\mathbf{x}) - m_t(\mathbf{x})) ds \right)^2 \right].$$

Expanding the square:

$$= \mathbb{E} \left[\int_{\mathcal{C}} \int_{\mathcal{C}} (f(\mathbf{x}) - m_t(\mathbf{x}))(f(\mathbf{x}') - m_t(\mathbf{x}')) ds ds' \right].$$

We then apply Fubini's theorem to exchange the order of expectation and integration:

$$= \int_{\mathcal{C}} \int_{\mathcal{C}} \mathbb{E} [(f(\mathbf{x}) - m_t(\mathbf{x}))(f(\mathbf{x}') - m_t(\mathbf{x}'))] ds ds'.$$

By definition of the [GP](#) posterior covariance function, we have:

$$\mathbb{E}[(f(\mathbf{x}) - m_t(\mathbf{x}))(f(\mathbf{x}') - m_t(\mathbf{x}'))] = k_t(\mathbf{x}, \mathbf{x}').$$

Thus:

$$\text{Var}[T_{\mathcal{C}}] = \int_{\mathcal{C}} \int_{\mathcal{C}} k_t(\mathbf{x}, \mathbf{x}') ds ds'.$$

This completes the proof. $\square$

Corollary 1. For any two curves $\mathcal{C}, \mathcal{C}' \subset \mathcal{M}$, the covariance of their traversal costs is:

$$\text{Cov}_t(T_{\mathcal{C}}, T_{\mathcal{C}'}) = \int_{\mathcal{C}} \int_{\mathcal{C}'} k_t(\mathbf{x}, \mathbf{x}') ds ds'.$$

Proof. Follows directly by applying the same reasoning as in the variance derivation, with the expectation taken over the product of deviations across two curves. $\square$

5.5 Methodology

We now describe how the [GP](#)-modelled cost field supports planning in uncertain environments. Our approach builds on the mathematical foundation established in Lemma 4, which shows that the traversal cost along any curve $\mathcal{C} \subset \mathcal{M}$ is a Gaussian random variable whose mean and variance are given by line integrals over the [GP](#) posterior.

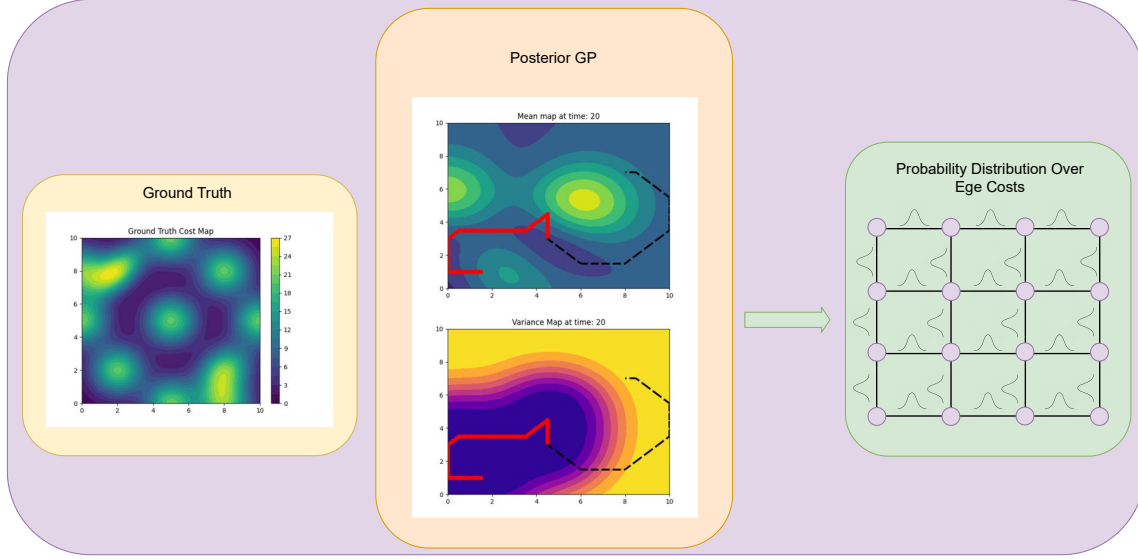


Figure 5.1: An agent navigates an unknown environment while collecting measurements from the ground-truth field. These observations are used to update its GP belief, represented by the posterior mean and variance maps. The belief is then transformed into a probability distribution over edge costs (*belief graph*) for planning.

5.5.1 Belief Graph with GP Edge Costs

To plan on a discretized spatial graph $G = (V, E)$, we associate each edge $e = (u, v) \in E$ with a continuous curve $\mathcal{C}_e \subset \mathcal{M}$. Using the results of Lemma 4, we define the *belief graph* at time t as:

$$G_t = (V, E, \mu_t^G, \sigma_t^G),$$

where the edge cost is represented by a Gaussian random variable:

$$T_{\mathcal{C}_e} \sim \mathcal{N}(\mu_t^G(e), \sigma_t^G(e, e)),$$

with

$$\mu_t^G(e) = \int_{\mathcal{C}_e} m_t(\mathbf{x}) ds, \quad (\text{expected cost via Lemma 4}) \quad (5.8)$$

$$\sigma_t^G(e, e') = \int_{\mathcal{C}_e} \int_{\mathcal{C}_{e'}} k_t(\mathbf{x}, \mathbf{x}') ds ds', \quad (\text{covariance via Corollary 1}). \quad (5.9)$$

These expressions allow us to compute the mean and covariance of any path cost T_P , where $P = \{e_1, \dots, e_k\}$, using:

$$\mu_t^P = \sum_{i=1}^k \mu_t^G(e_i), \quad (5.10)$$

$$\sigma_t^P = \sum_{i=1}^k \sum_{j=1}^k \sigma_t^G(e_i, e_j). \quad (5.11)$$

In this work, we use only the mean μ_t^G , since it is additive over edges, which allows us to use algorithms like Dijkstra’s for efficient planning. Including variance in the objective is much more challenging because variance is not additive, as each edge can be correlated with others. Therefore, computing the total requires considering all edge pairs. This quickly becomes expensive as the path gets longer. Nonetheless, variance is useful for quantifying risk in planning.

5.5.2 Adaptive Replanning Policy

Definition 7 (Policy). *A policy π is a decision-making rule that, at each time step t , selects the next edge to traverse based on the current vertex and the history of observations $\mathcal{D}_{\leq t} = (\mathbf{X}_{\leq t}, \mathbf{Y}_{\leq t})$.*

This policy constructs a path from start s to goal g adaptively, incorporating information gathered online. It relies on a [GP](#), which is iteratively updated as the agent collects new observations while traversing the environment. These observations form set $\mathcal{D}_{\leq t} = (\mathbf{X}_{\leq t}, \mathbf{Y}_{\leq t})$, which is used to condition the [GP](#) posterior at time t , as described in Equation (5.5).

At each time step t , the agent constructs a belief graph $G_t = (V, E, \mu_t^G, \sigma_t^G)$, where each edge $e \in E$ is assigned an expected cost $\mu_t^G(e)$ computed by integrating the [GP](#) posterior mean over the edge curve $\mathcal{C}_e$ (Equation (5.8)). Given this graph, the agent uses Dijkstra’s algorithm to compute the minimum-expected-cost path from its current vertex v_t to the goal vertex g . It then executes the first edge of that path, collects new observations along the traversed edge, and updates its dataset accordingly.

This feedback loop enables the agent to replan its path in response to newly gathered information, improving its estimate of the underlying cost field and making increasingly informed decisions. The Adaptive replanning policy is summarized in Algorithm 1.

Algorithm 1 Adaptive Replanning Policy π

- 1: **Input:** Graph $G = (V, E)$, start vertex $s \in V$, goal vertex $g \in V$
- 2: Initialize dataset $\mathcal{D}_{\leq 0} = (\mathbf{X}_{\leq 0}, \mathbf{Y}_{\leq 0})$ with initial observations
- 3: Set time $t \leftarrow 0$, and current vertex $v_t \leftarrow s$
- 4: **while** $v_t \neq g$ **do**
- 5: Update GP posterior $f \mid \mathcal{D}_{\leq t} \sim \mathcal{GP}(m_t, k_t)$ $\triangleright$ Eq. (5.5)
- 6: **for all** edges $e \in E$ **do**
- 7: Compute expected cost $\mu_t^G(e) \leftarrow \int_{\mathcal{C}_e} m_t(\mathbf{x}) ds$ $\triangleright$ Eq. (5.8)
- 8: **end for**
- 9: Use Dijkstra’s algorithm with edge weights $\mu_t^G(e)$ to compute:

$$\mathcal{P}^* \leftarrow \arg \min_{P \in \mathcal{P}_{v \rightarrow g}} \sum_{e \in P} \mu_t^G(e)$$

- 10: Let $e = (v_t, v_{t+1}) \in \mathcal{P}^*$ be the first edge
 - 11: Traverse $\mathcal{C}_e$, and collect noisy observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^r$
 - 12: Update dataset: $\mathcal{D}_{\leq t+1} \leftarrow \mathcal{D}_{\leq t} \cup \{(\mathbf{x}_i, y_i)\}_{i=1}^r$
 - 13: $t \leftarrow t + 1$
 - 14: **end while**
-

Computational Complexity. Let $n = |V|$ and $m = |E|$ denote the number of vertices and edges in the graph, respectively. At each time step t , the agent (i) updates the GP posterior using all previously collected observations, (ii) computes the expected cost $\mu_t^G(e)$ for each edge using Equation (5.8), and (iii) solves a shortest path problem using Dijkstra’s algorithm.

Assuming the agent collects a constant number of measurements per edge traversal, the dataset grows linearly with time, and the GP update requires $\mathcal{O}(t^3)$ operations due to matrix inversion. Computing expected edge costs via numerical integration over d discretization points per edge takes $\mathcal{O}(m \cdot d)$, while the most efficient implementation of Dijkstra’s algorithm adds $\mathcal{O}(m + n \log n)$ per step. The dominant per-step cost is thus $\mathcal{O}(t^3)$ from the GP update.

5.6 Simulation Results

To evaluate our approach and demonstrate its applicability, we conduct simulations on both synthetic cost fields and real Arctic datasets provided by the National Research Council of

Canada (NRC). We compare our adaptive replanning strategy against several baselines:

- **Certainty Equivalence Planner:** This baseline is similar to the adaptive planner but does not incorporate feedback. Given a prior GP and possibly some initial measurements, it computes the posterior distribution and uses it to evaluate the expected cost of each edge using Equation (5.8). A path from start to goal is then planned once, without updating the belief with future measurements. In essence, this represents an offline version of the adaptive policy. Comparing against this baseline highlights the benefit of incorporating real-time feedback during planning.
- **Shortest Path Planner:** A common baseline in path planning, this strategy ignores the stochastic cost field entirely and selects the shortest path based on Euclidean distance. It does not account for uncertainty or spatial variability in traversal cost. This comparison emphasizes how ignoring environmental uncertainty can lead to suboptimal routing decisions.
- **Random Planner:** In this baseline, edge costs are randomly sampled from a uniform distribution and used to compute a path via Dijkstra’s algorithm. Although unrelated to the actual cost field, it generates valid, non-looping paths with reasonable lengths. This helps assess whether our method outperforms uninformed or naïve strategies that disregard any cost structure.
- **Optimal Planner:** This oracle planner has access to the true underlying cost field and selects the minimum-cost path accordingly. While such knowledge does not exist in our setting, where the cost is inherently unknown, it serves as a lower bound on achievable performance. At best, other strategies can only match this planner.

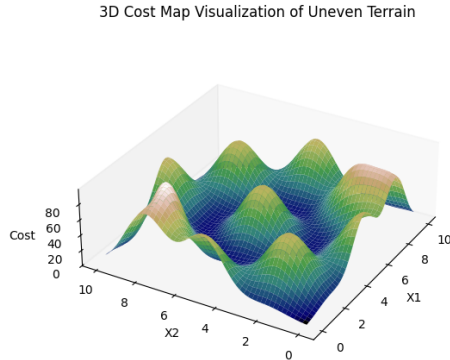
5.6.1 Uncertain Terrain Navigation

We begin our evaluation with a simulation scenario in which a robot must navigate an unknown, spatially varying terrain. The traversal cost is defined by a latent cost map, representing environmental difficulty. This cost is not directly observable by the robot, but may correlate with physical properties such as terrain elevation or surface friction, where higher values correspond to greater energy expenditure. An example of the cost landscape is shown in Figure 5.2, where the 3D surface (left) and corresponding heatmap (right) depict the underlying cost variability that the robot must learn and plan around.

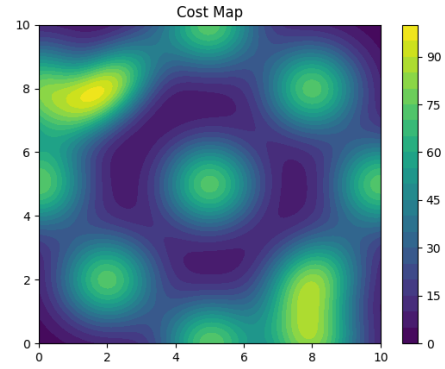
Setup 1. The environment is discretized into an 8-connected grid, with each cell spaced at a resolution of 0.5 units. This graph structure is used for all planners. The robot’s initial belief over the cost map is modelled using a GP prior with zero mean and a squared exponential kernel. To initialize the GP posterior, we consider two experimental setups: one with only 2 initial observations and another with 150. These observations are independently and identically distributed, corrupted by zero-mean Gaussian noise with variance 0.05. Both the certainty equivalence and adaptive planners are initialized with the same observations. In each experiment, the start and goal locations, as well as the initial observation locations, are randomly sampled, subject to a minimum separation threshold. Each planner returns a path, and we evaluate its actual cost and length using the true (ground-truth) cost map. We evaluate performance across 50 randomized runs for each condition, using the same true cost map but varying the start, goal, and initial observation locations. This allows us to assess the robustness of each planner under different configurations while isolating the effect of prior information.

Results. In the low-information setting with only 2 initial observations, the adaptive planner achieves a significantly lower mean cost (379.7) compared to the certainty equivalence baseline (461.0), highlighting the value of online replanning when prior knowledge is limited. The adaptive planner also outperforms uninformed strategies such as **shortest** (417.5) and **random** (402.5). Its performance remains above the optimal planner (234.5) due to the limited prior data available at initialization, which makes early decisions more exploratory and less informed. The adaptive planner also produces longer paths on average (16.7 versus 12.6), as it actively explores the environment and revises its plan in response to new measurements, leading to recourse decisions and consequently longer paths.

In the high-information setting with 150 initial observations, both the adaptive and certainty equivalence planners converge to near-optimal performance, with mean costs of 231.6 and 231.9, respectively, closely matching the optimal cost of 230.9. While the advantage of adaptivity is diminished in this case, both planners still leverage the GP model to capture the spatial correlations in the environment, which allows them to make informed decisions. This consistency with the true cost structure explains their strong performance. Path lengths are also more uniform across uncertainty-aware planners (approximately 13.4), whereas the **shortest** planner remains the most direct (11.4) but incurs a much higher cost (409.3) due to ignoring environmental variability. Figure 5.5 shows paths produced by each planner, and Figure 5.6 shows the final belief of the robot about the cost after traversing the adaptive planner’s path.

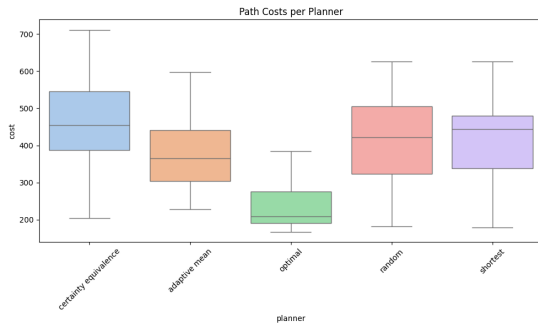


(a) 3D visualization of the generated traversal cost map over an unknown terrain.

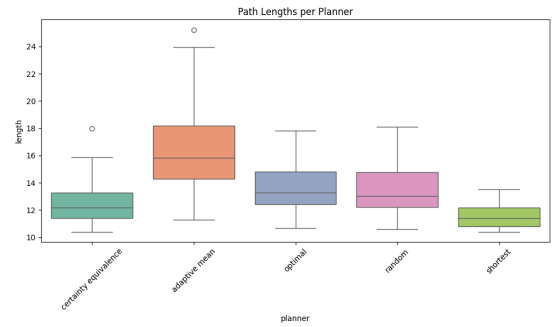


(b) 2D heatmap representation of the same traversal cost landscape.

Figure 5.2: Visualization of a spatially varying traversal cost field used to evaluate stochastic planning algorithms. The cost map represents an unknown terrain with varying difficulty. The agent does not observe the field directly, but instead receives noisy local measurements and updates its belief using a Gaussian Process model.

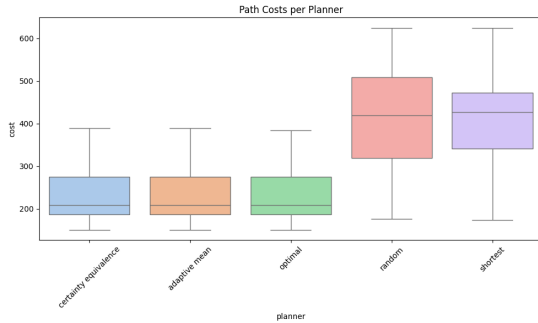


(a) Traversal cost for different planners using only two initial observations.

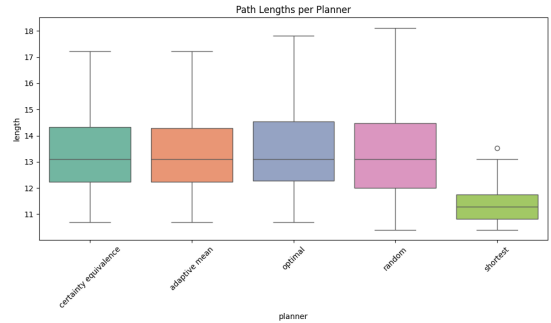


(b) Resulting path length for each planner in the same setting.

Figure 5.3: Comparison of different planning strategies in the low-information scenario with only two initial observations.



(a) Traversal cost for each planner with 150 initial observations.



(b) Resulting path length for each planner in the same setting.

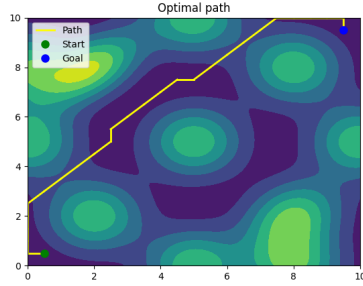
Figure 5.4: Comparison of different planners in the high-information scenario with 150 initial observations.

5.6.2 Real Data Setup

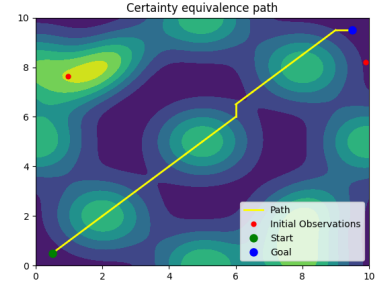
Setup 2 To evaluate planning under real-world uncertainty, we use data provided by the *National Research Council Canada (NRC)* [43, 44] representing ice concentration along the southern Labrador coast as depicted in Figure 5.7. This region exhibits significant spatial variability in ice concentration, making it well-suited for testing planners that rely on uncertain environmental information. We model traversal cost as a function of this concentration: areas with high ice concentration incur higher costs, reflecting their risk and reduced navigability for autonomous ships. The cost field is modelled using a GP with a squared exponential kernel, and learning hyperparameters using [23].

We focus on comparing three planners: the **optimal deterministic planner**, a **shortest-path baseline**, and an **adaptive planner** that replans based on updated beliefs. For the adaptive planner, we consider two scenarios with 10 and 50 initial GP observations, randomly sampled from a uniform distribution.

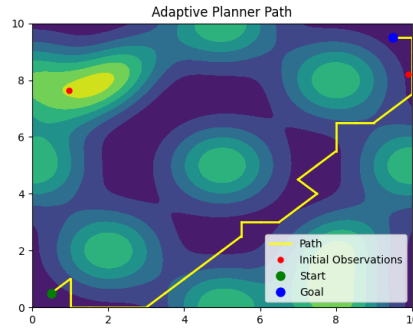
Results. As expected, the *optimal planner*, which has full access to the true cost map, achieves the lowest traversal cost of 1.21. The *shortest-path planner*, which ignores environmental cost and minimizes Euclidean distance, incurs a significantly higher cost of 5.03 due to navigating through high-concentration ice regions. The *adaptive planner with 10 initial measurements* incurs a cost of 3.49, as it initially explores two high-concentration



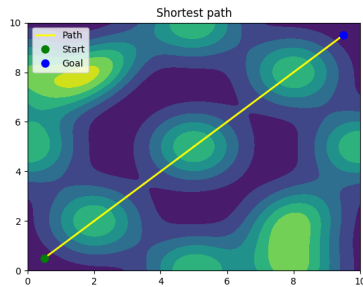
(a) Optimal planner with full knowledge of the cost field.



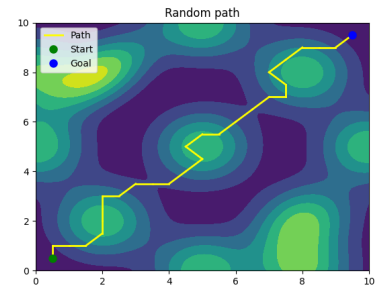
(b) Certainty equivalence planner using initial measurements.



(c) Adaptive planner that updates belief and replans based on new observations.

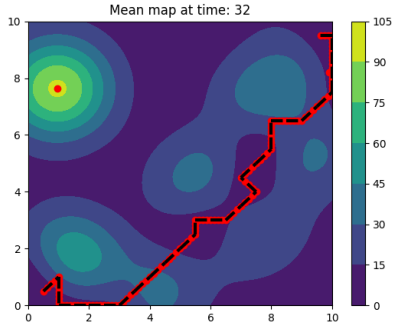


(d) Shortest path planner ignoring cost uncertainty.

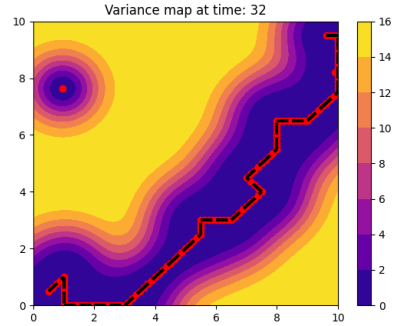


(e) Random path.

Figure 5.5: Comparison of different planning strategies for navigating an uncertain cost field.



(a) GP posterior mean of the cost field at final time step conditioned on the adaptive planner measurements.



(b) GP posterior variance at final time step conditioned on the adaptive planner measurements.

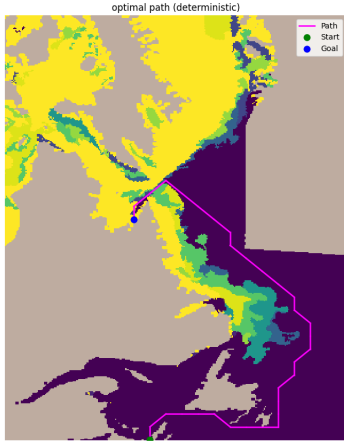
Figure 5.6: Final GP posterior mean (left) and variance (right) of the cost field, conditioned on measurements collected by the adaptive planner.

regions before forming a more accurate belief and rerouting. With 50 initial observations, the adaptive planner achieves a much lower cost of 1.73, reflecting the benefit of improved prior knowledge.

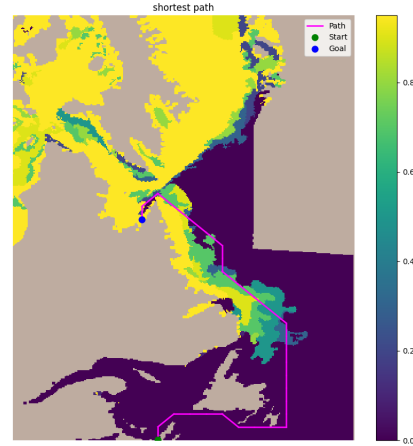
5.7 Summary

This chapter explores path planning in environments where traversal costs are unknown and spatially correlated, such as ship navigation in ice or robots traversing uncertain terrain. To model this, we use a [GP](#) to represent the cost field over space. As the agent moves, it gathers observations and updates its [GP](#) belief, enabling adaptive replanning through a belief graph framework. Simulations on both synthetic and real-world data demonstrate that this approach improves performance, particularly when prior knowledge is limited.

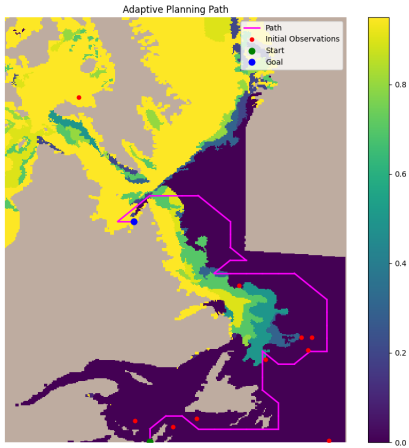
h



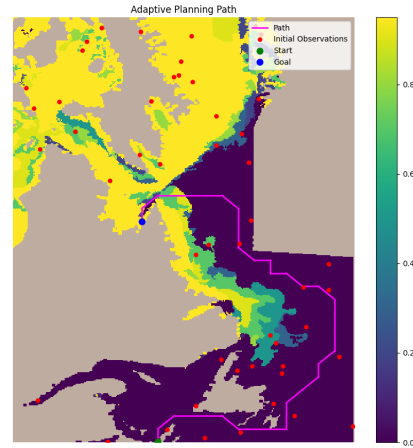
(a) Deterministic planner with full knowledge of the cost field (oracle baseline).



(b) Shortest path planner ignoring cost uncertainty.



(c) Adaptive planner with only 2 initial observations.



(d) Adaptive planner with 50 initial observations.

Figure 5.7: Planning over real ice concentration data from the National Research Council Canada (NRC)[43, 44], along the southern Labrador coast. The figure compares different path planning strategies under varying levels of prior information, highlighting the impact of uncertainty on navigation decisions.

Chapter 6

Conclusion

In this work, we addressed two key challenges in autonomous systems operating under uncertainty. The first is the problem of [Informative Path Planning \(IPP\)](#) in continuous space, where an agent must navigate complex environments under resource constraints to collect informative measurements and reduce uncertainty about an underlying phenomenon. The second is the [SPPRF](#) problem, where traversal costs are governed by an unknown stochastic process, and the objective is to find a path that minimizes expected cost.

6.1 Summary

In Chapter [4](#), we presented the continuous-space [IPP](#) problem and proposed a hierarchical planning framework composed of three stages. The first stage uses a graph-based [IPP](#) solver to identify an informative path. The second stage allocates the budget to each segment using geometric and information-based heuristics. In the final stage, each segment is refined via spline-based trajectory optimization. We demonstrated the effectiveness and efficiency of this approach through simulations on both synthetic and real-world datasets.

In Chapter [5](#), we addressed the [SPPRF](#) problem. We introduced a method for transforming the [GP](#) belief, formed from sequential observations, into a *belief graph*, a graph representation where edge costs are modelled as a joint Gaussian distribution. This structure enabled the design of an adaptive replanning policy that updates as new observations are collected. We evaluated the performance of this approach against several baselines in different simulated environments and real-world data, highlighting its ability to reduce traversal cost under uncertainty.

6.2 Future Work

While this work tackles two important challenges in path planning, several promising research directions remain open. Here, we highlight four extensions: adaptive [IPP](#), [IPP](#) via graphs of convex sets [\[52\]](#), non-myopic planning in stochastic environments, and [IPP](#) under stochastic budget constraints.

Adaptive [IPP](#). In our experiments, we either assumed that the [GP](#) hyperparameters were known or learned them from offline data. However, in real-world deployments, such data may be limited or unavailable, and the environment may differ from training conditions. This motivates adaptive [IPP](#) methods that update the [GP](#) online as new measurements are collected [\[21\]](#), allowing the planner to refine both its model and trajectory. Adaptivity is also critical when estimating task-specific features, such as level sets [\[71\]](#), where sensing must focus on specific regions of interest.

[IPP](#) via graphs of convex sets. Our high-level planner relies on a pre-defined, discretized graph. A promising alternative is to represent the environment as a graph of convex sets [\[52\]](#), where each node corresponds to a convex region of free space. This enables joint optimization over the discrete path and the continuous trajectory, without requiring a fixed graph structure.

Non-myopic policies for stochastic planning. Our current stochastic planner uses a myopic policy based on the current [GP](#) belief. While efficient, it does not account for how future observations may improve decision-making. Non-myopic policies, in contrast, reason about long-term information gain but are computationally intensive due to belief propagation. An interesting direction is to use deep learning to approximate future belief dynamics and enable tractable look-ahead planning.

[IPP](#) under stochastic budget constraints. Most [IPP](#) formulations assume a deterministic travel budget. In practice, traversal costs are often uncertain and must be inferred from data, as in the setting considered in our [SPPRF](#) framework. This motivates a unified approach where both the measured field and budget are stochastic. While one prior work considered this setting [\[66\]](#), it did not explore this aspect in depth. Developing methods to jointly address these uncertainties would improve robustness in real-world applications.

References

- [1] Rachit Aggarwal, Alexander Soderlund, Mrinal Kumar, and David J Grymin. Risk-aware path planning for unmanned aerial systems in a spreading wildfire. *Journal of Guidance, Control, and Dynamics*, 45(9):1692–1708, 2022.
- [2] Brian DO Anderson and John B Moore. *Optimal filtering*. Courier Corporation, 2005.
- [3] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 2018.
- [4] Rafael Bailon-Ruiz, Simon Lacroix, and Arthur Bit-Monnot. Planning to monitor wildfires with a fleet of uavs. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4729–4734. IEEE, 2018.
- [5] Alejandro Mendoza Barrionuevo, Samuel Yanes Luis, Daniel Gutiérrez Reina, and Sergio L Toral Marín. Informative deep reinforcement path planning for heterogeneous autonomous surface vehicles in large water resources. *IEEE Access*, 2024.
- [6] Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.
- [7] Patrick Billingsley. Stochastic processes. In *Probability and Measure*, chapter 7. Wiley, 3 edition, 1995.
- [8] Jonathan Binney, Andreas Krause, and Gaurav S Sukhatme. Informative path planning for an autonomous underwater vehicle. In *2010 IEEE International Conference on Robotics and Automation*, pages 4791–4796. IEEE, 2010.
- [9] Jonathan Binney and Gaurav S Sukhatme. Branch and bound for informative path planning. In *2012 IEEE international conference on robotics and automation*, pages 2147–2154. IEEE, 2012.

- [10] Jeremy Castagno, Matthew Romano, Prince Kuevor, and Ella Atkins. Multi-unmanned-aerial-vehicle wildfire boundary estimation using a semantic segmentation neural network. *Journal of Aerospace Information Systems*, 18(5):231–249, 2021.
- [11] Pengfei Chen, Yamin Huang, Eleonora Papadimitriou, Junmin Mou, and Pieter van Gelder. Global path planning for autonomous ship: A hybrid approach of fast marching square and velocity obstacles methods. *Ocean Engineering*, 214:107793, 2020.
- [12] Minjoo Choi, Hyun Chung, Hajime Yamaguchi, and Keisuke Nagakawa. Arctic sea route path planning based on an uncertain ice prediction model. *Cold Regions Science and Technology*, 109:61–69, 2015.
- [13] Taeyeong Choi and Grzegorz Cielniak. Adaptive selection of informative path planning strategies via reinforcement learning. In *2021 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2021.
- [14] Copernicus Climate Change Service, Climate Data Store. Era5 post-processed daily-statistics on single levels from 1940 to present. <https://doi.org/10.24381/cds.4991cf48>, 2024. Copernicus Climate Change Service (C3S) Climate Data Store (CDS), accessed July 2025.
- [15] Rodrigue De Schaetzen, Alexander Botros, Robert Gash, Kevin Murrant, and Stephen L Smith. Real-time navigation for autonomous surface vehicles in ice-covered waters. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1069–1075. IEEE, 2023.
- [16] Debadepta Dey, Andrey Kolobov, Rich Caruana, Ece Kamar, Eric Horvitz, and Ashish Kapoor. Gauss meets canadian traveler: shortest-path problems with correlated natural dynamics. In *AAMAS 2014*, pages 1101–1108, 2014.
- [17] Edsger W Dijkstra. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: his life, work, and legacy*, pages 287–290. 2022.
- [18] Shamak Dutta, Nils Wilde, and Stephen L Smith. An improved greedy algorithm for subset selection in linear estimation. In *2022 European Control Conference (ECC)*, pages 1067–1072. IEEE, 2022.
- [19] Shamak Dutta, Nils Wilde, and Stephen L Smith. Informative path planning in random fields via mixed integer programming. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 7222–7228. IEEE, 2022.

- [20] Shamak Dutta, Nils Wilde, and Stephen L Smith. A unified approach to optimally solving sensor scheduling and sensor selection problems in kalman filtering. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 1170–1176. IEEE, 2023.
- [21] Shamak Dutta, Nils Wilde, and Stephen L Smith. Informative path planning for active regression with gaussian processes via sparse optimization. *IEEE Transactions on Robotics*, 2025.
- [22] Gregory M Flato. Spatial and temporal variability of arctic ice thickness. *Annals of Glaciology*, 21:323–329, 1995.
- [23] Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.
- [24] Bruce L Golden, Larry Levy, and Rakesh Vohra. The orienteering problem. *Naval Research Logistics (NRL)*, 34(3):307–318, 1987.
- [25] Google Maps. Islands in Southern Ontario. <https://www.google.com/maps/@45.1912998,-80.1606578,15z/>, 2025. Accessed: July 11, 2025.
- [26] Hongliang Guo, Xuejie Hou, Zhiguang Cao, and Jie Zhang. Gp3: Gaussian process path planning for reliable shortest path in transportation networks. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):11575–11590, 2021.
- [27] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024.
- [28] Nikolaus Hansen. The cma evolution strategy: a comparing review. *Towards a new evolutionary computation: Advances in the estimation of distribution algorithms*, pages 75–102, 2006.
- [29] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [30] Hongwei He, Marc Mansuy, Jeroen Verwilligen, Guillaume Delefortrie, and Evert Lataire. Global path planning for inland vessels based on fast marching algorithm. *Ocean Engineering*, 312:119172, 2024.
- [31] Susan Hert. 2D polygon partitioning. In *CGAL User and Reference Manual*. CGAL Editorial Board, 6.0.1 edition, 2024.

- [32] Gregory Hitz, Enric Galceran, Marie-Ève Garneau, François Pomerleau, and Roland Siegwart. Adaptive continuous-space informative path planning for online environmental monitoring. *Journal of Field Robotics*, 34(8):1427–1449, 2017.
- [33] Geoffrey A Hollinger and Gaurav S Sukhatme. Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, 33(9):1271–1287, 2014.
- [34] Syed Talha Jawaid and Stephen L Smith. Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems. *Automatica*, 61:282–288, 2015.
- [35] Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.
- [36] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.
- [37] Ioannis Karatzas and Steven Shreve. Martingales, stopping times, and filtrations. In *Brownian Motion and Stochastic Calculus*, volume 113, chapter 1. Springer Science & Business Media, 2012.
- [38] Steven M Kay. *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.
- [39] Stefan Kern. Spatial correlation length scales of sea-ice concentration errors for high-concentration pack ice. *Remote Sensing*, 13(21):4421, 2021.
- [40] Sven Koenig and Maxim Likhachev. D* lite. In *Eighteenth national conference on Artificial intelligence*, pages 476–483, 2002.
- [41] Andreas Krause, Jure Leskovec, Carlos Guestrin, Jeanne VanBriesen, and Christos Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 134(6):516–526, 2008.
- [42] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(2), 2008.

- [43] Ivana Kubat, M Hossein Babaei, and Mohamed Sayed. Quantifying ice pressure conditions and predicting the risk of ship besetting. In *SNAME International Conference and Exhibition on Performance of Ships and Structures in Ice*, page D011S004R002. SNAME, 2012.
- [44] Ivana Kubat, David Watson, and Mohamed Sayed. Ice compression risks to shipping over canadian arctic and sub-arctic zones. In *OTC Arctic Technology Conference*, pages OTC–27348. OTC, 2016.
- [45] Xiaodong Lan and Mac Schwager. Rapidly exploring random cycles: Persistent estimation of spatiotemporal fields with multiple sensing robots. *IEEE Transactions on Robotics*, 32(5):1230–1244, 2016.
- [46] Markus Lange-Hegermann. Linearly constrained gaussian processes with boundary conditions. In *International Conference on Artificial Intelligence and Statistics*, pages 1090–1098. PMLR, 2021.
- [47] Ville Lehtola, Jakub Montewka, Floris Goerlandt, Robert Guinness, and Mikko Lensu. Finding safe and efficient shipping routes in ice-covered waters: A framework and a model. *Cold regions science and technology*, 165:102795, 2019.
- [48] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne Van-Briesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429, 2007.
- [49] Song Ma, Cunjia Liu, Christopher M Harvey, Richard Bucknall, and Yuanchang Liu. Adaptive informative path planning for active reconstruction of spatio-temporal water pollution dispersion using unmanned surface vehicles. *Applied Ocean Research*, 156:104458, 2025.
- [50] Ryan A MacDonald and Stephen L Smith. Active sensing for motion planning in uncertain environments via mutual information policies. *The International Journal of Robotics Research*, 38(2-3):146–161, 2019.
- [51] Roman Marchant and Fabio Ramos. Bayesian optimisation for informative continuous path planning. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6136–6143. IEEE, 2014.
- [52] Tobia Marcucci, Jack Umenberger, Pablo Parrilo, and Russ Tedrake. Shortest paths in graphs of convex sets. *SIAM Journal on Optimization*, 34(1):507–532, 2024.

- [53] George Mathew and Igor Mezić. Metrics for ergodicity and design of ergodic dynamics for multi-agent systems. *Physica D: Nonlinear Phenomena*, 240(4-5):432–442, 2011.
- [54] Ajith Anil Meera, Marija Popović, Alexander Millane, and Roland Siegwart. Obstacle-aware adaptive informative path planning for uav-based target search. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 718–724. IEEE, 2019.
- [55] Met Office. *Cartopy: a cartographic python library with a Matplotlib interface*. Exeter, Devon, 2010 - 2015.
- [56] Lauren M Miller and Todd D Murphey. Trajectory optimization for continuous ergodic exploration. In *2013 American Control Conference*, pages 4196–4201. IEEE, 2013.
- [57] Lauren M Miller, Yonatan Silverman, Malcolm A MacIver, and Todd D Murphey. Ergodic exploration of distributed information. *IEEE Transactions on Robotics*, 32(1):36–52, 2015.
- [58] Kevin Murrant, Robert Gash, and Jason Mills. Dynamic path following in ice-covered waters with an autonomous surface ship model. In *OCEANS 2021: San Diego–Porto*, pages 1–4. IEEE, 2021.
- [59] Masahiro Nomura and Masashi Shibata. cmaes : A simple yet practical python library for cma-es. *arXiv preprint arXiv:2402.01373*, 2024.
- [60] OpenStreetMap contributors. Planet dump retrieved from <https://planet.osm.org> . <https://www.openstreetmap.org>, 2017.
- [61] Joshua Ott, Mykel J Kochenderfer, and Stephen Boyd. Approximate sequential optimization for informative path planning. *Robotics and Autonomous Systems*, 182:104814, 2024.
- [62] Christos H Papadimitriou and Mihalis Yannakakis. Shortest paths without a map. *Theoretical Computer Science*, 84(1):127–150, 1991.
- [63] George H Polychronopoulos and John N Tsitsiklis. Stochastic shortest path problems with recourse. *Networks: An International Journal*, 27(2):133–143, 1996.
- [64] Marija Popović, Teresa Vidal-Calleja, Gregory Hitz, Jen Jen Chung, Inkyu Sa, Roland Siegwart, and Juan Nieto. An informative path planning framework for uav-based terrain monitoring. *Autonomous Robots*, 44(6):889–911, 2020.

- [65] Marija Popović, Teresa Vidal-Calleja, Gregory Hitz, Inkyu Sa, Roland Siegwart, and Juan Nieto. Multiresolution mapping and informative path planning for uav-based terrain monitoring. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1382–1388. IEEE, 2017.
- [66] Michael Quann, Lauro Ojeda, William Smith, Denise Rizzo, Matthew Castanier, and Kira Barton. An energy-efficient method for multi-robot reconnaissance in an unknown environment. In *2017 American Control Conference (ACC)*, pages 2279–2284. IEEE, 2017.
- [67] Michael Quann, Lauro Ojeda, William Smith, Denise Rizzo, Matthew Castanier, and Kira Barton. Ground robot terrain mapping and energy prediction in environments with 3-d topography. In *2018 Annual American Control Conference (ACC)*, pages 3532–3537. IEEE, 2018.
- [68] Michael Quann, Lauro Ojeda, William Smith, Denise Rizzo, Matthew Castanier, and Kira Barton. Chance constrained reachability in environments with spatially varying energy costs. *Robotics and Autonomous Systems*, 119:1–12, 2019.
- [69] Laurence Roberts-Elliott, Gautham P Das, and Grzegorz Cielniak. Kriging-variance-informed multi-robot path planning and task allocation for efficient mapping of soil properties. *Robotics*, 14(6):77, 2025.
- [70] Friedrich M Rockenbauer, Jaeyoung Lim, Marcus G Müller, Roland Siegwart, and Lukas Schmid. Traversing mars: Cooperative informative path planning to efficiently navigate unknown scenes. *IEEE Robotics and Automation Letters*, 2024.
- [71] Julius Rückin, Liren Jin, and Marija Popović. Adaptive informative path planning using deep reinforcement learning for uav-based active sensing. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4473–4479. IEEE, 2022.
- [72] Poorya Shobeiry, Ming Xin, Xiaolin Hu, and Haiyang Chao. Uav path planning for wildfire tracking using partially observable markov decision process. In *AIAA Scitech 2021 Forum*, page 1677, 2021.
- [73] Ryan N Smith, Mac Schwager, Stephen L Smith, Burton H Jones, Daniela Rus, and Gaurav S Sukhatme. Persistent ocean monitoring with underwater gliders: Adapting sampling resolution. *Journal of Field Robotics*, 28(5):714–741, 2011.

- [74] Anthony Stentz. Optimal and efficient path planning for partially-known environments. In *Proceedings of the 1994 IEEE international conference on robotics and automation*, pages 3310–3317. IEEE, 1994.
- [75] James Stewart. *Calculus: Early Transcendentals*. Cengage Learning, Boston, MA, 8th edition, 2015.
- [76] KN Tahar. Aerial terrain mapping using unmanned aerial vehicle approach. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39:493–498, 2012.
- [77] Florence Tsang, Tristan Walker, Ryan A MacDonald, Armin Sadeghi, and Stephen L Smith. Lamp: Learning a motion policy to repeatedly navigate in an uncertain environment. *IEEE transactions on robotics*, 38(3):1638–1652, 2021.
- [78] Vladyslav Usenko, Lukas Von Stumberg, Andrej Pangercic, and Daniel Cremers. Real-time trajectory replanning for mavs using uniform b-splines and a 3d circular buffer. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 215–222. IEEE, 2017.
- [79] Omar Velasco, João Valente, and Abeje Y Mersha. An adaptive informative path planning algorithm for real-time air quality monitoring using uavs. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1121–1130. IEEE, 2020.
- [80] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.
- [81] Jing Wang, Zhiqiang Zhai, Guangdong Xu, Ruoyu Zhang, Xin Zhang, and Rong Hu. Optimizing sampling points and path planning for soil monitoring in agricultural land. *Agronomy*, 14(12):2947, 2024.
- [82] Christopher Williams and Carl Rasmussen. Gaussian processes for regression. *Advances in neural information processing systems*, 8, 1995.
- [83] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.
- [84] Jingjin Yu, Mac Schwager, and Daniela Rus. Correlated orienteering problem and its application to informative path planning for persistent monitoring tasks. In *2014*

IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 342–349. IEEE, 2014.

- [85] Jingjin Yu, Mac Schwager, and Daniela Rus. Correlated orienteering problem and its application to persistent monitoring tasks. *IEEE Transactions on Robotics*, 32(5):1106–1118, 2016.
- [86] Haotian Zhang, Raid Ayoub, and Shreyas Sundaram. Sensor selection for optimal filtering of linear dynamical systems: Complexity and approximation. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 5002–5007. IEEE, 2015.
- [87] Ninghan Zhong, Alessandro Potenza, and Stephen L Smith. Autonomous navigation in ice-covered waters with learned predictions on ship-ice interactions. *arXiv preprint arXiv:2409.11326*, 2024.