



FACULTY OF BUSINESS, ECONOMICS AND INFORMATION
SYSTEMS
UNIVERSITY OF PASSAU

BACHELOR THESIS

Philipp Stockerl

**Robust and Adaptive Path
Planning for Autonomous Vehicles
in Spatio-Temporal Cost Fields**

Chair of

Business Decisions & Data Science
Business and Economics Focus Management Science
Operations and Supply Chain Management

Supervisor of the Thesis: Prof. Dr. Marc Goerigk
Double Bachelor Programmes: Business Administration and Economics,
Information Systems

Passau, 23.01.2025

INFORMATION

Title: Robust and Adaptive Path Planning for Autonomous Vehicles
in Spatio-Temporal Cost Fields

Author: Philipp Stockerl

Institute: Chair of Business Decisions & Data Science,
Business and Economics Focus Management Science/
Operations and Supply Chain Management

Supervisor: Prof. Dr. Marc Goerigk

Keywords: Optimization, Management Science
Geostatistics, Modelling,
Optimization under Uncertainty,
Robotics, Path Planning

ABSTRACT

This thesis investigates minimum-cost path planning in spatio-temporal environments under uncertain edge costs. Our testbed is mobile-robot routing on directed graphs, with edge costs generated from synthetic spatio-temporal random fields via geostatistical methods. We compare two complementary approaches: (i) robust combinatorial optimization models for shortest paths, and (ii) an incremental shortest path heuristic that adapts after costs change. We address uncertainty both ex-ante (before executing a path) and ex-post (after cost shifts). We evaluate three options drawn from these two approaches and use the results to answer the research objectives and questions. Our research shows that...

TABLE OF CONTENTS

Notation	1
List of Tables.	3
List of Figures	4
CHAPTER 1	5
INTRODUCTION	
CHAPTER 2	9
CONCEPTUAL FOUNDATIONS	
2-1 Model and Data Driven Decision Support Systems.	9
2-2 Data Component: Geostatistical Data Generation	10
2-3 Model Component: Shortest-Path Planning Under Uncertainty	12
2-4 Robust Combinatorial Optimization	15
2-5 Incremental Search-Based Path-Planning.	19
CHAPTER 3	23
IMPLEMENTATION DETAILS	
3-1 Decision Support Systems.	23
3-2 Geostatistical Data Generation for Spatio-Temporal Cost Fields	23
3-3 Discrete Uncertainty Min–Max Model	25
3-4 Budgeted Uncertainty Min–Max Model	26
3-5 D* Lite	27
CHAPTER 4	30
EXPERIMENTAL RESULTS	
4-1 Baseline.	31
4-2 Experiment 1: Urban vs. Regional Scale	33

4-3 Experiment 2: Short vs. Long Exposure	34
4-4 Experiment 3: Rough vs. Smooth Terrain	35
4-5 Experiment 4: Temporal Stability	36
4-6 Experiment 5: Uncertainty Budgeted	37
4-7 Experiment 6: Number of Beacons	38

CHAPTER 5 39
CONCLUSION

CHAPTER 6 45
APPENDIX

6-1 Geostatistical Formulations.	45
--	----

Notation

Symbol	Description
Graphs and paths	
$G = (V, E)$	Directed graph; (V, E) notation used in robust models.
$G = (S, E)$	Directed graph; (S, E) notation used in D* Lite.
V, S	Vertex sets (robust optimization vs. D* Lite grid cells).
E	Set of directed edges (arcs).
i, j	Node indices for edge endpoints $(i, j) \in E$.
s, t	Start and target nodes in robust models.
$s_{\text{start}}, s_{\text{goal}}$	Current start and goal vertices in D* Lite.
$\delta^+(\cdot), \delta^-(\cdot)$	Successor and predecessor edge sets.
\mathcal{X}	Solution set (of feasible $s-t$ paths).
x_e	Binary decision variable indicating whether edge e is used.
L_{\min}	Minimum number of edges on an $s-t$ path (4-connected grid).
Scenarios and costs	
Ω, ω	Scenario index set and index.
\mathcal{T}, τ	Time index set and time index for STRF slices.
$ \Omega , \mathcal{T} $	Number of scenarios and time steps.
c_e^ω	Cost of edge e in scenario ω .
$c(i, j)$	Traversal cost of edge (i, j) in the current scenario (D* Lite).
\mathbf{c}^ω	Scenario cost vector.
$f_\tau(p, q)$	Scenario cost surface at time τ .
$Z(p, q, \tau)$	Spatio-temporal random field value.
Discrete robust model	
z	Worst-case path cost (discrete min–max model).
Budgeted uncertainty model	
\underline{c}_e	Nominal (average) edge cost.
d_e	Maximum deviation from nominal cost.
Γ	Budget of uncertainty.
π	Dual variable for the budget constraint.
ρ_e	Auxiliary deviation variable for edge e .
λ	Normalized budget factor, $\Gamma(\lambda) = \lambda \Gamma_{\max}$.
Γ_{\max}	Maximum budget, typically set to L_{\min} .

Continued on next page

Symbol	Description
D* Lite	
$g(s)$	Current shortest-path cost estimate from s to s_{goal} .
$\text{rhs}(s)$	One-step lookahead cost-to-go from s .
$h(i, j)$	Admissible heuristic estimate between vertices; also used as a metric distance in the geostatistical model (context by arguments).
$k(s) = (k_1(s), k_2(s))$	Lexicographic priority key for vertex s .
k_m	Heuristic shift variable due to start-vertex movement.
U	Priority queue of locally inconsistent vertices.
u	Vertex selected from U for expansion.
$\text{ComputeShortestPath}()$	Procedure that restores local consistency.
$\text{UpdateVertex}(s)$	Procedure that updates $\text{rhs}(s)$ and queue membership.
Geostatistical/SRF model	
$\mathbf{x} = (p, q, \tau)$	Coordinate vector in the 3D space–time domain.
p, q	Spatial coordinates along the x- and y-axes.
N_p, N_q	Grid dimensions in the spatial axes.
$\Delta p, \Delta q, \Delta \tau$	Coordinate increments in space and time.
ℓ_p, ℓ_q, ℓ_t	Spatial and temporal correlation lengths.
a_p, a_q, a_t	Anisotropy ratios.
σ^2	Field variance.
α	Stable-kernel exponent; also spatial scaling factor in experiments (context disambiguates).
β	Temporal scaling factor.
ν	Matérn smoothness parameter.
\mathbf{k}_i	Wave vectors in the spectral construction.
$Z_{1,i}, Z_{2,i}$	Standard normal coefficients in the spectral method.
$\gamma(r), C(\cdot), \rho(\cdot)$	Semi-variogram, covariance, and correlation functions.

List of Tables

2.1	Conceptual distinctions between nominal shortest-path planning, ex-ante robust optimization, and ex-post incremental search under cost uncertainty.	14
3.1	Notation for the discrete-scenario min–max shortest-path model	25
3.2	Notation for the budgeted-uncertainty min–max shortest-path model . .	26
3.3	Notation for the D* Lite incremental shortest-path algorithm	27
4.1	Baseline experimental configuration	31
4.2	Performance comparison under the baseline configuration.	31
4.3	Experiment 1: Parameter variations relative to the baseline configuration	33
4.4	Performance comparison across grid sizes	33
4.5	Experiment 2: Parameter variations relative to the baseline configuration	34
4.6	Performance comparison across exposure levels $ \mathcal{T} $	34
4.7	Experiment 4: Parameter variations relative to the baseline configuration	35
4.8	Performance comparison across spatial scaling factors α	35
4.9	Experiment 6: Parameter variations relative to the baseline configuration	36
4.10	Performance comparison across temporal scaling factors β	36
4.11	Experiment 7: Parameter variations relative to the baseline configuration	37
4.12	Performance comparison across budget levels λ	37
4.13	Experiment 8: Parameter variations relative to the baseline configuration	38
4.14	Performance comparison across beacon counts	38
6.1	Predefined covariance models in GSTools [MSZH22].	46

List of Figures

1.1	Spatial environmental data with sensor locations and inferred wind intensity over a two-dimensional domain.	5
1.2	Spatial cost realizations at two different time points over a fixed graph topology.	6
1.3	Top-down architecture of the conceptual decision support system used for experimental comparison, integrating spatio-temporal cost generation, robust optimization, incremental search-based planning, and a hybrid guidance variant within a unified pipeline.	8
2.1	Basic architecture of a model-driven decision support system	9
2.2	Difference of covariance models for same parameters visualized	11
2.3	Conceptual overview of the shortest-path problem under uncertainty. The underlying graph structure remains fixed, while traversal costs evolve over time. Robust combinatorial optimization and incremental search-based planning represent two distinct responses to uncertainty within the same problem setting.	15
2.4	Geometric interpretation of uncertainty sets in cost space. Left: discrete uncertainty represented by a finite set of scenario realizations. Right: continuous budgeted uncertainty defined by nominal costs and bounded deviations under a global uncertainty budget Γ	17
2.5	Visualization of D* Lite internals, illustrating backward search, heuristic guidance, and forward path execution. $s = u$ being the corresponding top node in U at this point.	21
4.1	Visualization of the baseline cost field for the first time frame.	32

CHAPTER 1

INTRODUCTION

Problem Setting Autonomous vehicle navigation admits a wide range of planning formulations, ranging from global route planning to local reactive control and hybrid approaches that combine multiple decision layers [KNK24, AK23]. Depending on the application, planners optimize objectives such as travel time, energy consumption, risk exposure, or mission goals under kinematic, safety, and environmental constraints [ASSM⁺22, GH23, ZM18]. Despite this diversity, many planning algorithms share a common structural core: they evaluate feasible paths on graph-based representations under additive cost models.

From a methodological perspective, these formulations can be viewed as instances of combinatorial optimization problems (COP) studied in operations research (OR). The shortest-path problem (SPP) provides the most elementary abstraction of this core and underlies a wide range of planning algorithms, including heuristic search, dynamic programming, and network flow formulations [WWW⁺25, GH24, QSW⁺23]. Even when planning problems extend beyond single-pair routing—for example, through timing constraints, risk measures, or vehicle dynamics—their algorithmic structure often still relies on modified shortest-path computations.

In real-world navigation tasks, however, the nominal assumptions of the classical SPP no longer hold. Traversal costs vary across space and time and are only partially observable at planning time. In unmanned aerial vehicle (UAV) applications, for instance, costs depend on environmental conditions such as wind fields, which directly affect travel time and operational risk along flight segments [RGRM24]. Forecasts and sensor data provide partial information, but future realizations remain uncertain and deviations from nominal predictions are unavoidable.

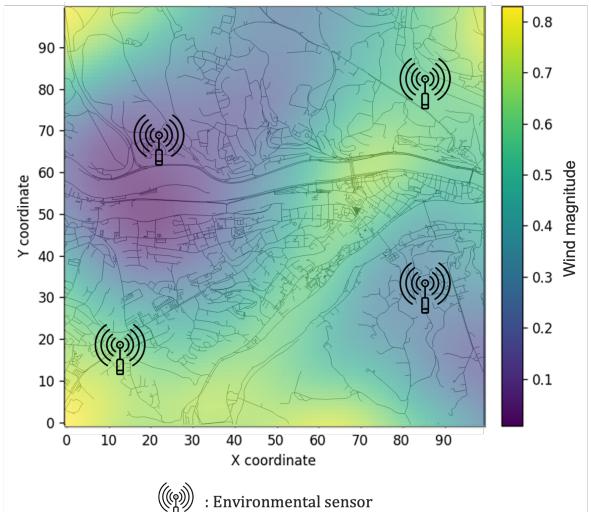


Figure 1.1: Spatial environmental data with sensor locations and inferred wind intensity over a two-dimensional domain.

Figure 1.1 shows how heterogeneous environmental information can be aggregated into spatial cost fields that serve as inputs to routing decisions. As costs evolve over time, the relative attractiveness of feasible paths changes rather than being blocked entirely [AS24], such that planning decisions based solely on nominal information may perform poorly or even fail. Figure 1.2 illustrates this effect by showing spatial cost realizations at two different time points over an identical graph structure. This motivates extensions of the classical shortest-path problem that explicitly account for uncertainty and temporal dynamics in traversal costs.

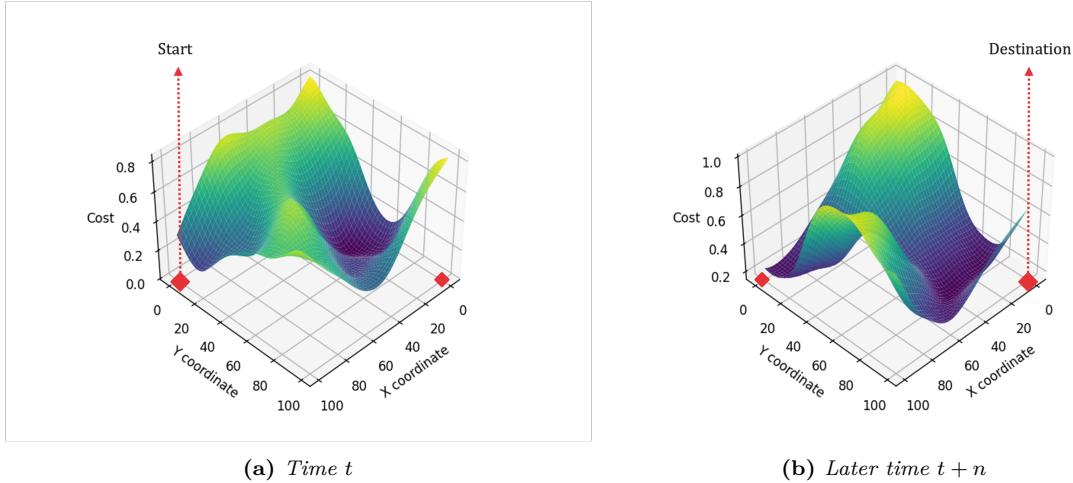


Figure 1.2: Spatial cost realizations at two different time points over a fixed graph topology.

Two dominant paradigms address shortest-path planning under such uncertainty by extending the classical model in fundamentally different ways:

Robust optimization (RO) addresses uncertainty *ex ante* by selecting complete paths prior to execution and protecting them against adverse cost realizations within explicitly defined uncertainty sets. Robust combinatorial shortest-path formulations optimize the worst-case path cost over such sets and increasingly derive uncertainty sets from data rather than expert choice [CDG19, GK25]. This approach adopts a global, strategic perspective on uncertainty and is particularly suited to planning problems in which route decisions must be fixed before execution and robust performance guarantees are required. Prominent application areas include vehicle routing problems (VRPs), where robustness is used to model application-specific uncertainty [RGRM24, ZZL23, YAB23]. Further extensions incorporate time-dependent decision making at multiple planning levels [GL21, ZLS⁺22] and address scalability challenges arising from large data sets [WWW⁺25]. Evaluation in this literature typically relies on scenario-based or bounded-deviation constructions to compare robustness across alternative formulations [CDG19]. These models provide the ex-ante reference solutions used for comparison in this work.

Incremental search-based planning addresses uncertainty *ex post* by revealing cost information during execution and repairing paths locally as changes occur. These

methods build on classical shortest-path and heuristic search principles, most notably Dijkstra’s algorithm [Dij59] and A* [HNR68], and extend them to dynamic environments through algorithms such as D* [Ste94], LPA* [KLF04], and D* Lite [KL02]. By reusing previous search effort, incremental planners avoid full recomputation and enable efficient replanning when edge costs change [BDG⁺24]. Recent research focuses on refining these foundations to improve replanning efficiency, path quality, and robustness under frequent cost updates, for example by incorporating safety margins, kinematic constraints, or adaptive heuristics [WQL⁺21, WLJ⁺22, HHW24, LLZ⁺24, XXG⁺25]. As a result, incremental search produces globally consistent paths while emphasizing execution-time adaptivity in dynamic and partially known environments [KSDS21]. This paradigm is therefore particularly well suited to mobile robotic systems, where tight integration with onboard sensing and frequent environment updates is required.

In essence, robust optimization emphasizes strategic, global path planning by selecting a complete route at planning time and protecting it against worst-case cost realizations, often assuming centralized computation and limited adaptivity during execution. Incremental search-based methods likewise aim to compute globally consistent paths, but rely on locally grounded exploration and adaptation, trading increased online computation for the ability to react to newly revealed cost information. As a result, incremental planners are particularly well suited to onboard execution in mobile robotic systems, where sensing and computation are tightly coupled. Although both paradigms address the same underlying routing task, they differ fundamentally in their information assumptions, degree of adaptivity, and evaluation criteria.

Research Objective The objective of this thesis is to develop and apply a controlled experimental framework that enables systematic comparisons between ex-ante robust planning and ex-post adaptive search under identical spatio-temporal uncertainty realizations. In addition, the thesis explores a hybrid planning approach in which robust optimization provides global, ex-ante guidance, while incremental search contributes local, ex-post adaptation during execution. Using this framework, we conduct a systematic experimental comparison of selected robust shortest-path formulations and an incremental replanning method under spatio-temporal cost uncertainty. Performance is evaluated with respect to solution quality, computational effort, and execution-time adaptivity in order to assess the practical implications of each paradigm for autonomous vehicle guidance.

Motivation A key motivation for this objective is that, despite extensive research in robust optimization and incremental search, there is little controlled, head-to-head comparison of these approaches under shared experimental conditions. Robust shortest-path formulations and incremental search methods are typically studied in isolation and rarely evaluated on identical graph structures, common spatio-temporal cost re-

alizations, and comparable performance metrics. This lack of a unified experimental setting limits our understanding of their relative strengths, weaknesses, and practical trade-offs for autonomous vehicle guidance on the same underlying cost fields.

Related work exists in global-local planning frameworks that combine long-horizon planning with local reactive control [GHF⁺24, LD25]. These methods typically decompose planning into a hierarchical structure, where a global planner computes a reference path and a local controller handles short-term disturbances and obstacle avoidance. While effective in practice, such approaches primarily focus on architectural integration and performance improvements within a single paradigm. Consequently, they do not provide a controlled comparison between ex-ante robustness and ex-post adaptivity under a common problem formulation, uncertainty model, and evaluation protocol.

Contribution The contributions of this thesis are fourfold: (i) the design of a reproducible experimental framework that enforces shared data, graph structures, and evaluation metrics across planning algorithms; (ii) a spatio-temporal cost generation process based on geostatistical random fields that are mapped consistently onto routing graphs; (iii) implementations of robust and incremental planning methods, including a hybrid guidance variant that combines ex-ante protection with ex-post adaptation; and (iv) a unified evaluation protocol for comparing solution cost, computational effort, and execution-time adaptivity. The experimental framework is implemented as a conceptual decision support system (DSS) that integrates spatio-temporal cost generation, robust combinatorial optimization models, and incremental search-based planning methods into a unified pipeline (Figure 1.3).

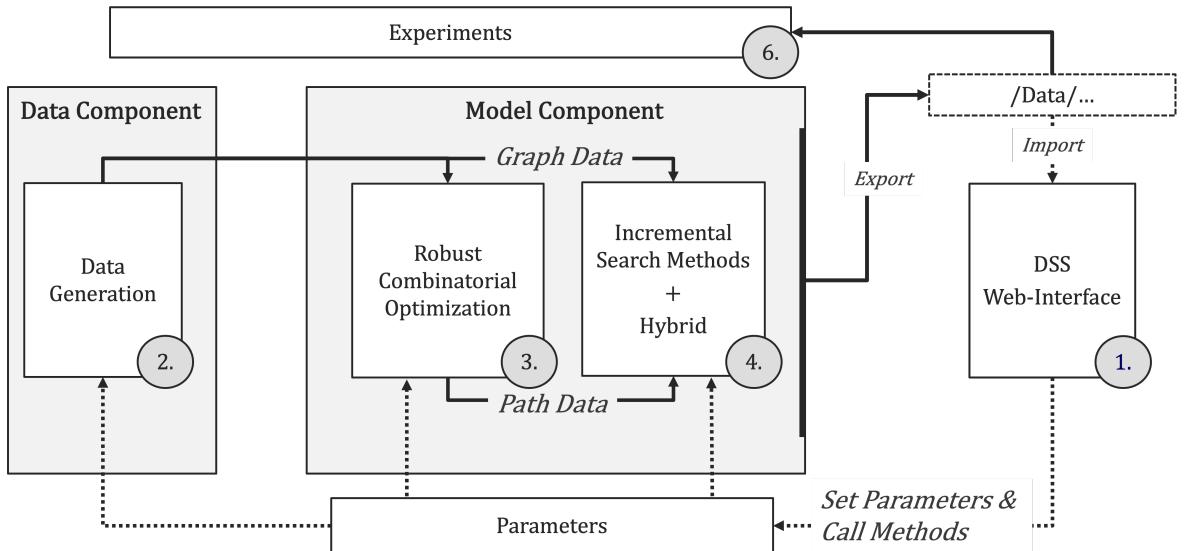


Figure 1.3: Top-down architecture of the conceptual decision support system used for experimental comparison, integrating spatio-temporal cost generation, robust optimization, incremental search-based planning, and a hybrid guidance variant within a unified pipeline.

CHAPTER 2

CONCEPTUAL FOUNDATIONS

2-1 Model and Data Driven Decision Support Systems

Decision Support Systems (DSS) are interactive, computer-based systems that support decision-making in complex and uncertain environments by integrating data, analytical models in user interfaces [FB22]. Rather than automating decisions, DSS structure decision-making by making assumptions explicit and enabling systematic comparison of alternatives.

At a functional level, DSS link data with analytical models. Input data are processed and transformed before being evaluated by models that represent the underlying decision problem. The resulting outputs—such as costs, feasibility indicators, or performance measures—are interpreted through visualization and interaction, forming the basis for model-driven decision support [Pow02].

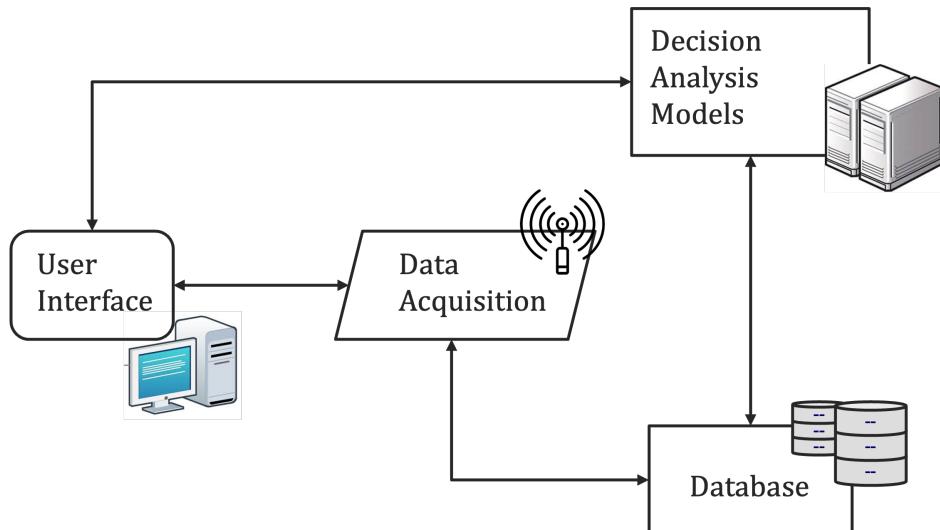


Figure 2.1: Basic architecture of a model-driven decision support system

DSS for Vehicle Routing Vehicle routing constitutes a prominent application area for decision support systems [RMA04, LRS21]. Routing decisions require selecting efficient paths while accounting for operational constraints, uncertain travel times, and

dynamic environmental conditions. As a result, vehicle routing DSS naturally integrate multiple analytical components.

A central role is played by model-driven DSS [PS07], which rely on formal models to represent routing decisions and evaluate alternatives. Such systems may incorporate deterministic optimization models, robust optimization models [GRT⁺16], heuristic algorithms, or simulation-based approaches [FIU⁺15]. Decision variables typically correspond to route choices or planning parameters, while performance measures include total cost, robustness, or computational effort. Environmental conditions act as exogenous inputs that influence outcomes but cannot be directly controlled.

Vehicle routing DSS often exhibit a strong data-driven component. When routing decisions depend on geographic or spatial information, this leads naturally to spatial DSS [Kee98], which integrate geographic representations such as maps or spatial cost fields. These representations enable decision makers to interpret routing decisions in their geographic context and to relate analytical results to real-world environments [UHV17]. In this thesis, spatial DSS concepts are combined with model-driven DSS to address routing problems under spatio-temporal uncertainty.

Model and Data Utilization The effectiveness of a DSS depends critically on the models and data it employs. Data quality, aggregation, and normalization directly influence the reliability of decision-relevant information, while the choice of analytical models determines which aspects of the decision problem can be analyzed and optimized. We employ a conceptual DSS for our autonomous vehicle routing problem as an interactive control surface that embeds a data component and model components.

2-2 Data Component: Geostatistical Data Generation

For the data component of our conceptual framework, we generate spatio-temporally correlated cost fields using geostatistical methods. These fields serve as controlled, reproducible inputs to all shortest-path models evaluated in this work. Since geostatistics is used purely as a data generation tool, we focus on implementation-relevant design choices rather than a detailed statistical treatment.

Using simulated data for model comparison is common practice when real-world data is scarce or difficult to control experimentally [MPW26, MSY⁺25]. Geostatistical frameworks provide a principled way to generate spatially and temporally correlated synthetic data while maintaining full control over variance, correlation length, and anisotropy [MSZH22]. In particular, we rely on the GSTools library, which offers reproducible implementations of second-order random field models widely used in

simulation studies [Rai19].

We model cost fields as realizations of a weakly stationary Gaussian random field defined on a three-dimensional space–time domain. Spatial and temporal dependence are introduced via a parametric covariance function, whose structure is fully determined by a normalized correlation kernel and a metric distance. Time is treated as an additional geometric dimension, resulting in temporally consistent spatial slices rather than independent scenarios.

GSTools provides a family of admissible correlation kernels with varying smoothness and compact support properties (see Table 6.1). While kernel choice influences local regularity, all kernels share a common structure in which dependence is governed by a single scalar distance. Directional anisotropy and spatio-temporal coupling are incorporated by defining this distance through a weighted metric.

Figure 2.2 illustrates the qualitative effect of different kernel choices under identical parameters. These visualizations demonstrate that kernel selection alone can induce substantially different spatial roughness, despite identical variance and correlation length. We further discuss kernel selection in Chapter 4.

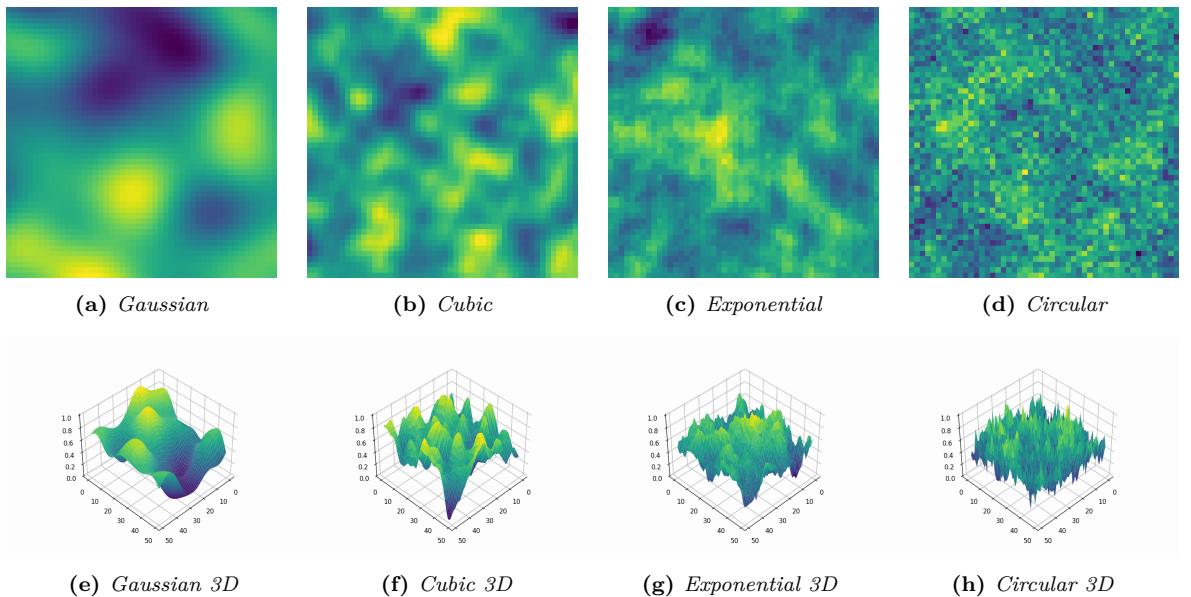


Figure 2.2: Difference of covariance models for same parameters visualized

The resulting spatio-temporal random fields constitute the data-generating basis for all experiments. Each temporal slice represents a spatial cost realization, and temporal correlation ensures consistency across scenarios. By discretizing spatial slices into directed graphs, the continuous geostatistical model is transformed into time-indexed shortest-path instances, enabling a controlled comparison of robust and adaptive planning methods under identical uncertainty assumptions

2-3 Model Component: Shortest-Path Planning Under Uncertainty

This section establishes a common problem formulation for shortest-path planning that serves as a conceptual foundation for the two planning paradigms studied in this thesis. It serves as a backbone for the model component of our conceptual DSS. We begin with the classical shortest-path problem under nominal assumptions and then introduce cost uncertainty, highlighting how violations of these assumptions lead to fundamentally different planning logics.

Although both paradigms address the same underlying routing problem on an identical graph structure, they differ in how this problem is represented and solved. Following the taxonomy of Hart et al. [HNR68], we distinguish between optimization-based formulations and heuristic search-based methods as two complementary perspectives on shortest-path planning. This distinction provides the organizing lens for the remainder of this chapter.

Shortest-path problem formulation We consider a directed graph $G = (V, A)$, where V denotes the set of vertices and $A \subseteq V \times V$ the set of directed arcs. Each arc $e = (i, j) \in A$ is associated with a nonnegative traversal cost c_e . A path P from a start vertex $s \in V$ to a destination vertex $t \in V$ is defined as an ordered sequence of arcs connecting s to t . The single-pair shortest-path problem (SPSP) consists of finding a path P^* that minimizes the total path cost

$$C(P) = \sum_{e \in P} c_e.$$

Such graph structures naturally represent spatial or spatio-temporal cost fields derived from simulated or real-world data (Figures 1.1 and 1.2). The shortest-path problem is inherently combinatorial, as it requires selecting a feasible sequence of discrete arcs from a finite set.

Nominal shortest-path problem In the classical (nominal) setting, traversal costs are assumed to be nonnegative, time-invariant, and fully known at planning time. Under these assumptions, both optimization-based and heuristic search-based approaches solve the same underlying combinatorial problem and are equivalent with respect to optimality, differing only in representation and solution strategy.

From an *optimization perspective*, the shortest-path problem is treated as a combinatorial optimization problem in which arcs constitute the ground set of decision elements. Following the general framework of combinatorial optimization [GH24], let

$S = A$ denote the finite ground set of directed arcs. Each arc $e \in A$ is associated with a nonnegative cost c_e , and a feasible solution corresponds to selecting a subset of arcs that forms an $s-t$ path. Using binary decision variables $x_e \in \{0, 1\}$ to signal whether a edge is selected or not, the nominal shortest-path problem can be formulated as

Nominal Combinatorial Shortest-Path Problem

$$\min \sum_{e \in A} c_e x_e \quad (2.1)$$

$$\text{s.t. } \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = b_i \quad \forall i \in V, \quad (2.2)$$

$$x_e \in \{0, 1\} \quad \forall e \in A, \quad (2.3)$$

where $b_s = 1$, $b_t = -1$, and $b_i = 0$ for all $i \in V \setminus \{s, t\}$. The flow-balance constraints (2.2) define the feasible set $\mathcal{X} \subseteq \{0, 1\}^{|A|}$ and encode the combinatorial structure of valid $s-t$ paths. Under nonnegative costs, any optimal solution corresponds to a simple path.

Heuristic search-based methods provide an algorithmic counterpart to this formulation. They interpret the graph as a discrete state space in which vertices represent states and arcs represent transitions. Paths are constructed implicitly through successive state expansions guided by cost estimates and heuristic information, rather than by explicitly selecting edges. The nominal shortest-path problem can equivalently be expressed as a discrete dynamic programming problem. Let $V^*(v)$ denote the optimal cost-to-go from node v to the destination t . Under deterministic, time-invariant, and fully known arc costs, V^* satisfies Bellman's optimality equation

Bellman's Optimality Equation

$$V^*(t) = 0, \quad V^*(v) = \min_{(v,u) \in A} \{c_{vu} + V^*(u)\} \quad \forall v \in V \setminus \{t\}. \quad (2.4)$$

This recursion formalizes optimal substructure: every subpath of an optimal path is itself optimal. Heuristic search algorithms such as A* exploit this structure by constructing the optimal path implicitly. A* maintains cost-to-come estimates $g(v)$ and expands nodes in increasing order of

$$f(v) = g(v) + h(v),$$

where $h(v)$ is an admissible and consistent heuristic estimate of the remaining cost from v to t . With an admissible, consistent heuristic, A* returns an optimal $s-t$

path. Thus, under nominal assumptions, optimization-based formulations and heuristic search methods solve the same combinatorial shortest-path problem and are equivalent with respect to optimality. A* serves as the nominal baseline; incremental methods such as LPA* and D* Lite reuse its heuristic structure and repair only affected nodes when edge costs change, rather than replanning from scratch (Table 2.1).

Conceptual Dimension	Nominal Shortest Path	Ex-Ante Robust Optimization	Ex-Post Incremental Search
Cost information	Deterministic, time-invariant, fully known	Uncertain but bounded via uncertainty sets	Revealed or updated during execution
Optimality principle	Bellman optimality holds globally	Bellman optimality generally fails under min–max objectives	Local consistency enforced incrementally
Representation	Explicit or implicit (equivalent)	Explicit edge-based decision variables	Implicit state-space exploration
Decision timing	Single planning step	Ex-ante path selection	Continuous replanning
Response to cost changes	Not applicable	No adaptation unless re-solved	Local repair
Primary objective	Minimum total cost	Worst-case protected cost	Adaptivity and replanning efficiency

Table 2.1: Conceptual distinctions between nominal shortest-path planning, ex-ante robust optimization, and ex-post incremental search under cost uncertainty.

Shortest-path planning under uncertainty In spatio-temporal routing environments, the nominal assumptions of time-invariant costs and full information no longer hold. We model uncertainty through time-varying traversal costs on a fixed graph. Let $k \in \{1, \dots, T\}$ index discrete planning or execution steps, and let c_e^k denote the cost of arc e at step k . The cost vector $c^k = (c_e^k)_{e \in A}$ represents a realization of a spatio-temporally correlated cost field. Conditional on a realization, costs remain deterministic and nonnegative, but future realizations are not known at planning time. At the time a path is planned, only partial information about future cost realizations is available, for example in the form of forecasts, bounds, or scenario samples. Additional cost information may become available during execution. Figure 2.3 illustrates how this uncertainty alters the decision context relative to the nominal setting. Uncertainty fundamentally affects the information structure and objective of the planning problem.

See [MAR⁺17, DP84] for additional SPP taxonomy. While each realized instance of the shortest-path problem satisfies Bellman’s optimality principle individually, robust

objectives may couple decisions across time or scenarios. As a result, global optimal substructure need not hold with respect to the robust planning objective, even though the underlying graph structure is unchanged.

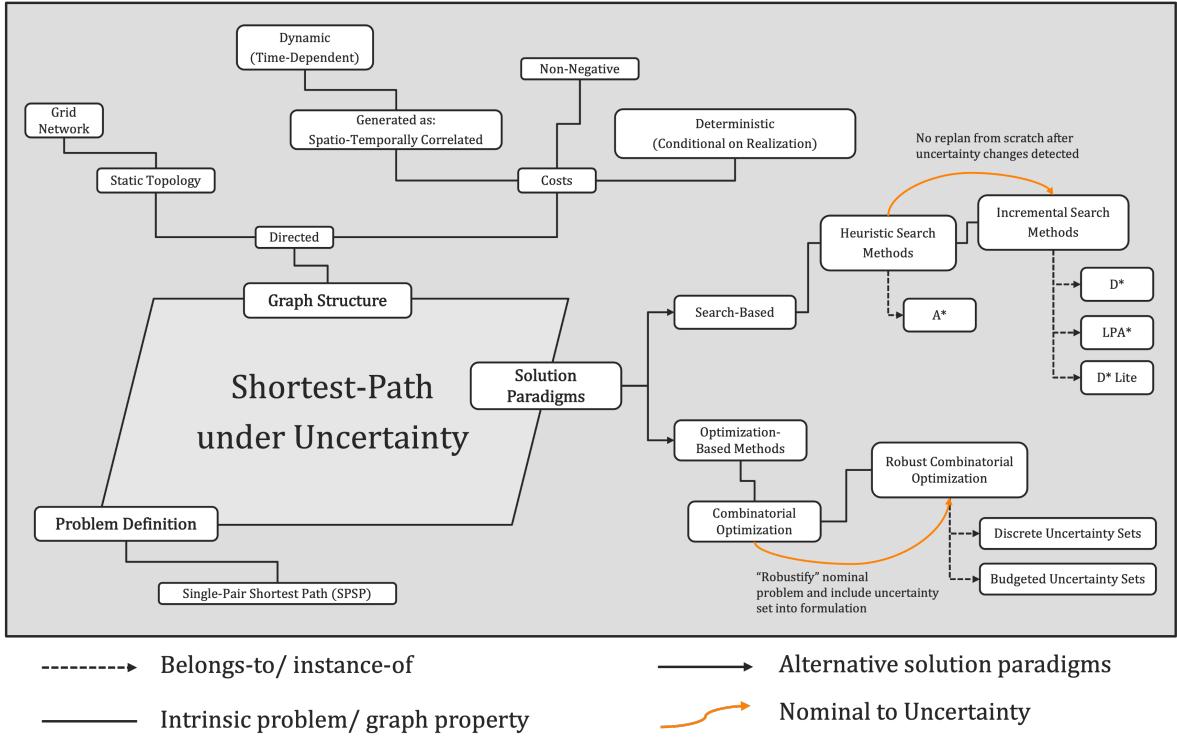


Figure 2.3: Conceptual overview of the shortest-path problem under uncertainty. The underlying graph structure remains fixed, while traversal costs evolve over time. Robust combinatorial optimization and incremental search-based planning represent two distinct responses to uncertainty within the same problem setting.

Two coherent planning paradigms emerge from this setting. Ex-ante robust optimization selects paths before execution and protects them against adverse cost realizations within prescribed uncertainty sets. Ex-post incremental search-based planning, in contrast, interleaves planning and execution by repairing paths locally as new cost information becomes available.

The following sections formalize these two paradigms in detail. Section 2-4 introduces robust combinatorial shortest-path models under ex-ante uncertainty, while Section 2-5 presents incremental search-based planning as an ex-post adaptive alternative.

2-4 Robust Combinatorial Optimization

Building on the nominal formulation in Section 2-3 for the shortest-path problem (Equation 2.3), robust optimization replaces fixed costs c with an uncertainty set \mathcal{U} and seeks a path $x \in \mathcal{X}$ that minimizes the worst-case cost over \mathcal{U} . We refer to [GH24]

for this section.

Robust Optimization and the Min–Max Principle In robust shortest-path planning, edge costs are not assumed to be known exactly at planning time. Instead, costs are assumed to belong to a prescribed uncertainty set $\mathcal{U} \subseteq \mathbb{R}_{\geq 0}^{|E|}$. The planner selects a feasible path, while the adversary selects a cost realization from \mathcal{U} that maximizes the cost of that path.

Formally, robust planning is based on the min–max criterion

$$\min_{x \in \mathcal{X}} \max_{\mathbf{c} \in \mathcal{U}} \sum_{(i,j) \in E} c_{ij} x_{ij},$$

where \mathcal{X} denotes the set of feasible s – t paths defined by the flow-balance constraints. This criterion guarantees worst-case protection against all cost realizations contained in \mathcal{U} . Alternative robustness concepts exist, but the min–max formulation provides the strongest feasibility guarantees and is therefore adopted throughout this thesis.

The choice of the uncertainty set \mathcal{U} determines the level of conservatism and the computational structure of the resulting model. In the following paragraphs, we introduce the uncertainty models used in this thesis. Their integration into fully specified shortest-path formulations is deferred to the Methodology chapter, where implementation details are discussed.

Uncertainty Sets The uncertainty set \mathcal{U} specifies which realizations of edge costs are considered plausible and directly determines the level of robustness and conservatism of the resulting solution. In the context of shortest-path planning, each component of a cost vector corresponds to a single directed edge of the underlying graph.

In this thesis, uncertainty sets are not inferred from observational data, but constructed deliberately from synthetic spatio-temporal random fields to enable controlled experimentation. Edge costs are generated as a three-dimensional array indexed over space and time, where each two-dimensional slice corresponds to a spatial cost realization at a given time step. Based on this representation, two complementary uncertainty models are derived and evaluated: discrete scenario-based uncertainty and continuous budgeted uncertainty. Figure 2.4 provides a geometric illustration of the two uncertainty models considered in this thesis.

Discrete Uncertainty Sets A discrete uncertainty set consists of a finite collection of cost scenarios

$$\mathcal{U} = \{\mathbf{c}^1, \dots, \mathbf{c}^K\},$$

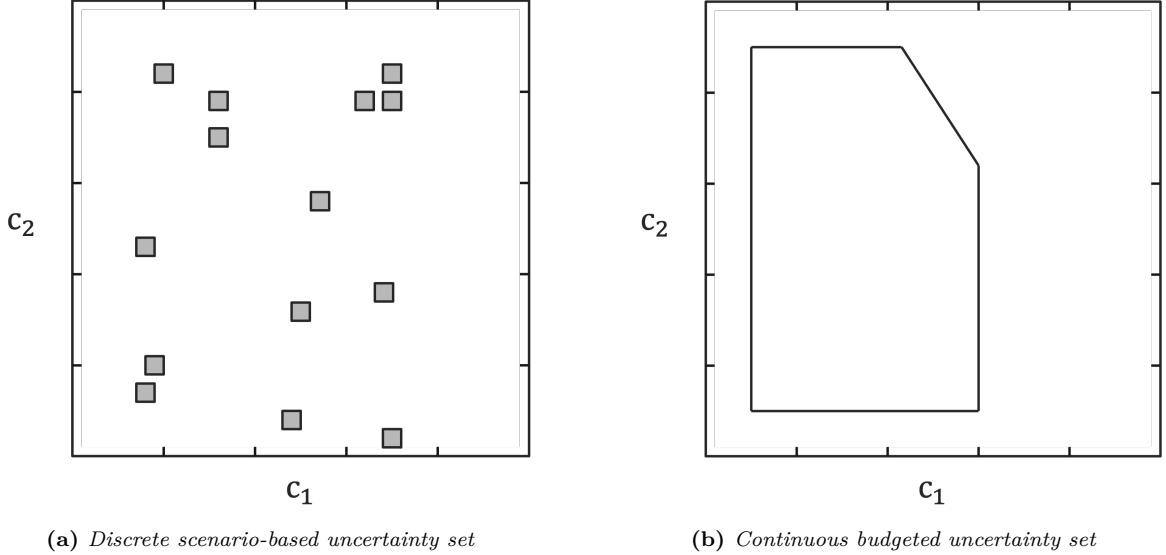


Figure 2.4: Geometric interpretation of uncertainty sets in cost space. Left: discrete uncertainty represented by a finite set of scenario realizations. Right: continuous budgeted uncertainty defined by nominal costs and bounded deviations under a global uncertainty budget Γ .

where each vector \mathbf{c}^k represents a complete realization of edge costs over the graph. As illustrated in Figure 2.4a, discrete uncertainty models represent uncertainty through a finite set of cost vectors, each corresponding to a complete scenario realization.

In the experimental setup of this thesis, discrete uncertainty arises naturally from the spatio-temporal random field representation. Each cost vector \mathbf{c}^k corresponds to a single two-dimensional time slice of the three-dimensional spatio-temporal random field array. That is, all edge costs within one scenario are taken from the same temporal snapshot, preserving spatial correlation while treating scenarios as mutually independent realizations.

As a result, the number of scenarios K directly corresponds to the number of time steps generated in the random field. Robust optimization with discrete uncertainty therefore evaluates candidate paths against all temporal realizations of the cost field considered in the experiment.

For discrete uncertainty sets, the min–max problem can be reformulated using an epigraph formulation:

Epigraph Reformulation for Discrete Uncertainty

$$\min \quad z \tag{2.5}$$

$$\text{s.t.} \quad z \geq \sum_{i \in [n]} c_i^k x_i \quad \forall k \in [K], \tag{2.6}$$

$$x \in \mathcal{X}. \tag{2.7}$$

This reformulation replaces the inner maximization by a finite set of linear constraints and yields a deterministic mixed-integer optimization problem.

Continuous Budgeted Uncertainty Sets Continuous budgeted uncertainty sets aggregate information across multiple cost scenarios into a compact representation consisting of nominal costs and bounded deviations. Rather than treating each scenario independently, this model summarizes the range of cost variation observed across time.

In this thesis, nominal and worst-case edge costs are derived by aggregating all scenario realizations of a given edge across time. For each directed edge, a nominal cost \underline{c}_i and a maximum deviation d_i are computed from the set of observed costs across scenarios. The resulting budgeted uncertainty set restricts the total amount of deviation that may occur simultaneously across edges by means of a global budget parameter Γ .

$$\mathcal{U} = \left\{ \mathbf{c} \mid c_i = \underline{c}_i + d_i \delta_i, 0 \leq \delta_i \leq 1, \sum_{i \in [n]} \delta_i \leq \Gamma \right\}.$$

In contrast to the discrete uncertainty set, the budgeted uncertainty set shown in Figure 2.4b defines a continuous polyhedral region around a nominal cost vector, where the total deviation across edges is limited by the budget parameter Γ .

For shortest-path planning, the budget parameter Γ admits an intuitive interpretation. It bounds the number or aggregate magnitude of edges along a path whose costs are allowed to deviate from their nominal values at the same time. Small values of Γ correspond to mild uncertainty and yield solutions close to the nominal shortest path, while larger values model increasingly adverse and coordinated cost deviations, resulting in more conservative paths.

Following standard duality arguments, the inner maximization problem can be dualized, yielding the following robust counterpart:

Robust Min–Max Formulation for Budgeted Uncertainty

$$\min_{x, \pi, \rho} \quad \sum_{i \in [n]} \underline{c}_i x_i + \Gamma \pi + \sum_{i \in [n]} \rho_i \tag{2.8}$$

$$\text{s.t.} \quad \pi + \rho_i \geq d_i x_i \quad \forall i \in [n], \tag{2.9}$$

$$x \in \mathcal{X}, \tag{2.10}$$

$$\pi \geq 0, \quad \rho_i \geq 0 \quad \forall i \in [n]. \tag{2.11}$$

In the experimental implementation, nominal costs and deviations for budgeted uncertainty are computed by aggregating all scenario realizations of each edge across time. Nominal costs are derived either as the minimum or the average observed cost per

edge, while worst-case costs are given by the maximum observed value. The deviation parameter is then defined as the difference between worst-case and nominal cost. A detailed description of this aggregation procedure and its implementation is provided in the Methodology chapter.

This formulation preserves the combinatorial structure of the shortest-path problem while incorporating controlled worst-case protection against uncertain edge costs. It constitutes the robust shortest-path model used for ex-ante planning in the experimental evaluation of this thesis.

Summary This section established the theoretical foundation for ex-ante shortest-path planning under uncertainty. Robust combinatorial optimization models uncertainty explicitly at planning time and yields solutions with worst-case performance guarantees. While discrete uncertainty enumerates extreme realizations explicitly, budgeted uncertainty replaces enumeration by a continuous envelope that bounds coordinated deviations across edges. In the following section, this approach is contrasted with search-based path-planning algorithms that address uncertainty *ex post* through adaptive replanning during execution.

2-5 Incremental Search-Based Path-Planning

Incremental search-based path-planning algorithms constitute the canonical *ex-post* response to uncertainty in shortest-path problems. Rather than committing to a fully specified path under assumed cost realizations, these methods maintain and update shortest-path information during execution as new cost information becomes available.

This execution-centric perspective is particularly well suited to environments in which traversal costs are revealed incrementally or change over time. Instead of modeling uncertainty explicitly at planning time, incremental search algorithms treat uncertainty as an information process and adapt decisions as costs are observed. As a result, they emphasize responsiveness, reuse of prior computation, and real-time feasibility under partial observability.

In contrast to robust combinatorial optimization, which selects a single path before execution and evaluates it against worst-case cost realizations, incremental search-based methods interleave planning and execution. Paths are repaired locally when costs change, allowing the planner to react to evolving conditions without recomputing solutions from scratch. This fundamental difference makes incremental search a natural *ex-post* counterpart to ex-ante robust optimization in the context of spatio-temporal cost uncertainty.

Incremental Heuristic Search as an Execution Model Building on the A* baseline introduced above, incremental search algorithms reuse heuristic search structure to enable efficient replanning under changing information. At their core, these methods maintain lower-bound estimates on remaining cost-to-go and exploit structural properties of shortest paths to reuse previous computation when costs are updated.

For incremental search methods such as D* Lite, a stronger and more general consistency condition is required. Rather than relating heuristic values across individual edges, D* Lite employs a heuristic of the form $h(s_1, s_2)$ that estimates the cost of a shortest path between two arbitrary vertices. Koenig and Likhachev express the corresponding consistency requirement as a triangle inequality with respect to true shortest-path costs:

$$h(s, s'') \leq c^*(s, s') + h(s', s'') \quad \forall s, s', s'' \in V,$$

where $c^*(s, s')$ denotes the optimal cost of a path from s to s' .

This formulation generalizes the classical edge-based consistency condition used by A* and directly supports D* Lite’s backward search and moving-start execution model. In grid-based graphs, the condition is typically satisfied by metric distance heuristics such as Manhattan or Euclidean distance.

Incremental heuristic search methods extend this framework by maintaining local consistency conditions between stored cost estimates and one-step lookahead values. Rather than discarding prior results when costs change, these methods repair only those parts of the search space that are affected by new information. Lifelong Planning A* (LPA*) formalizes this idea by introducing explicit consistency conditions that enable efficient reuse of previous search effort.

D* Lite as a Canonical Incremental Planner Dynamic A* (D*), originally introduced by Stentz, extends incremental search to navigation scenarios in which an agent moves through an initially unknown or changing environment. D* Lite preserves the execution model and optimality guarantees of D* while recasting its algorithmic structure within the simpler and more transparent framework of LPA*.

Rather than introducing new planning concepts, D* Lite unifies heuristic search, incremental repair, and execution-time adaptation into a single coherent algorithm. It eliminates specialized state classifications in favor of a consistent incremental shortest-path formulation, significantly reducing implementation complexity while retaining theoretical guarantees. As a result, D* Lite has become the preferred incremental search algorithm in both theory and practice.

In this thesis, D* Lite is used as the representative incremental search-based planner for the *ex-post* planning paradigm. The following paragraphs describe its internal state

representation, consistency mechanism, and execution model, which together enable efficient replanning under spatio-temporal cost uncertainty.

State Representation and Local Consistency Let $G = (S, E)$ be a directed graph with nonnegative edge costs. We consider a fixed goal s_{goal} and a start vertex s_{start} that changes as the agent moves. For each vertex $s \in V$, D* Lite maintains a cost-to-go estimate $g(s)$ and a one-step lookahead value

$$\text{rhs}(s) = \begin{cases} 0, & \text{if } s = s_{goal}, \\ \min_{s' \in \text{Succ}(s)} (c(s, s') + g(s')), & \text{otherwise.} \end{cases}$$

A vertex is *locally consistent* if $g(s) = \text{rhs}(s)$. Inconsistencies arise when edge costs change or when the agent moves, indicating that stored path information must be repaired. D* Lite restores optimality by incrementally re-establishing local consistency only where necessary.

Priority Queue and Lexicographic Keys All locally inconsistent vertices are stored in a priority queue. Each vertex is assigned a lexicographically ordered key

$$k(s) = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \min\{g(s), \text{rhs}(s)\} + h(s_{start}, s) + k_m \\ \min\{g(s), \text{rhs}(s)\} \end{bmatrix},$$

where the second component acts as a tie-breaker. This key structure generalizes the f - and g -value ordering of A* while enabling incremental updates without loss of correctness.

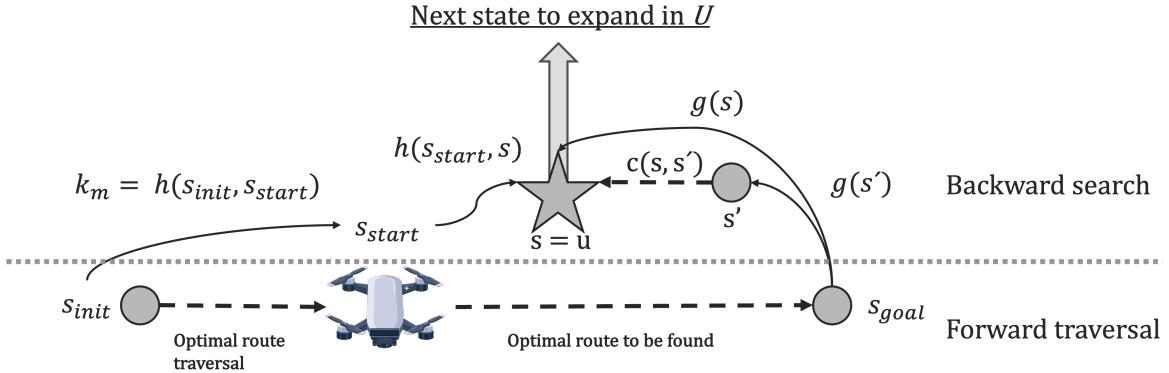


Figure 2.5: Visualization of D* Lite internals, illustrating backward search, heuristic guidance, and forward path execution. $s = u$ being the corresponding top node in U at this point.

Heuristic Shift and Moving Start As the agent moves, the start vertex changes and heuristic values become outdated. D* Lite avoids reordering the entire priority queue by introducing a heuristic shift variable k_m , which accumulates heuristic differences between successive start positions. This mechanism preserves lower-bound properties while maintaining relative queue ordering, enabling efficient replanning during execution.

Greedy Path Execution Once all relevant vertices are locally consistent, the stored cost-to-go values define a greedy execution policy. The agent follows successors that minimize $c(s, s') + g(s')$, yielding an optimal path under the current cost realization.

Role Within the Experimental Framework In this thesis, D* Lite serves as a representative ex-post planning approach. It processes cost information sequentially and adapts paths during execution, but does not provide worst-case guarantees. This contrasts directly with the ex-ante robust optimization models, enabling a controlled comparison between adaptivity and worst-case protection under spatio-temporal uncertainty. We include further details regarding implementation and consistency to Koenig and Likhachev's [KL02] pseudocode in chapter 4.

CHAPTER 3

IMPLEMENTATION DETAILS

3-1 Decision Support Systems

The experimental pipeline is implemented as a lightweight model- and data-driven DSS that orchestrates data generation, model execution, and result visualization. It provides a single control interface for running all experiments under identical parameter settings and producing comparable outputs (Figure 1.3). The DSS is implemented as a local Streamlit application; randomness is controlled through fixed seeds and explicit configuration parameters to ensure reproducibility.

3-2 Geostatistical Data Generation for Spatio-Temporal Cost Fields

This section links the implementation in the `STGRF` generator to the formulations in the appendix (Chapter 6). Each step below states the code action and its mathematical counterpart.

Step 1: Covariance model and kernel selection. A kernel from `GSTools` (e.g., Gaussian, Matérn, Stable) (see Table 6.1) is selected and parameterized by variance σ^2 , length scale ℓ , anisotropy ratios a_i , and (if applicable) kernel-specific parameters ν or α . In the implementation this corresponds to

```
Kernel(spatial_dim=3, temporal=False, var=..., len_scale=..., anis=...).
```

This instantiates the covariance model $C(r)$ in (6.2) with $\text{cor}(\cdot)$ from (6.3). The anisotropy ratios define effective correlation lengths $\ell_i = \ell a_i$ and enter the metric distance in (6.5), which in turn defines the spatio-temporal covariance in (6.6). Time is treated as a third spatial dimension (spatial dimension $d = 3$), consistent with $\mathbf{x} = (x, y, \tau)$.

Step 2: Structured space–time grid. The implementation constructs the coordinate arrays

$$x = (0, \Delta, \dots, (N - 1)\Delta), \quad y = (0, \Delta, \dots, (N - 1)\Delta), \quad \tau = (0, \dots, |\mathcal{T}| - 1),$$

where Δ is the cell size and N is the number of grid points per axis (`grid_size` in the implementation). These arrays define the structured grid on which the field is evaluated, and each grid point corresponds to a coordinate vector $\mathbf{x} = (x, y, \tau)$ in the formulation of $Z(\mathbf{x})$. Let

$$Z(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d,$$

denote a real-valued random field indexed over a d -dimensional domain.

Step 3: Spectral random field generation. Given the covariance model, `GSTools` constructs a stationary Gaussian random field via its spectral randomization method. The call `SRF(model, seed).structured([x, y, t])`, where t is the scenario index array τ , produces samples $Z(\mathbf{x})$ on the structured grid, implementing the generation in (6.7) for the covariance in (6.6).

Step 4: Global normalization. To ensure nonnegative and comparable costs across scenarios, the sampled 3D field is optionally rescaled as

$$\tilde{Z} = \frac{Z - \min Z}{\max Z - \min Z},$$

where the extrema are taken over the entire space–time grid.

Step 5: Scenario slices and graph mapping. For each scenario index τ , the 2D slice $f_\tau(x, y) := Z(x, y, \tau)$ is extracted as the spatial cost surface. The routing graph is a directed 4-connected grid; each grid point (x, y) is mapped to a node id $i = xN + y$. Each directed edge (i, j) receives the mean of its incident node values,

$$c(i, j) = \frac{1}{2} (f_\tau(x_i, y_i) + f_\tau(x_j, y_j)),$$

yielding a scenario-specific edge-cost graph derived directly from the realization $Z(\mathbf{x})$. We use `NetworkX` to generate our 4-connected graph.

Step 6: Export for downstream models. The implementation writes a global node file and one edge file per scenario, as well as the per-scenario field slices, enabling downstream shortest-path and robust optimization models to consume the same spatio-temporal realizations.

3-3 Discrete Uncertainty Min–Max Model

The sets V and E are instantiated from the data files (V from `nodes.csv`, E from `edges.csv` in scenario 000), and each scenario cost vector \mathbf{c}^ω is read from the corresponding `scenario_\omega/edges.csv`.

Table 3.1: Notation for the discrete-scenario min–max shortest-path model

Symbol	Description
$G = (V, E)$	Directed graph representing the routing environment
V	Set of nodes (vertices), indexed by node IDs
$E \subseteq V \times V$	Set of directed edges (arcs)
$s, t \in V$	Fixed start node s and target node t
$\Omega = \{0, 1, \dots, \Omega - 1\}$	Index set of cost scenarios
c_e^ω	Cost of edge $e \in E$ in scenario $\omega \in \Omega$
$\delta^+(v)$	Set of outgoing edges of node $v \in V$
$\delta^-(v)$	Set of incoming edges of node $v \in V$
$x_e \in \{0, 1\}$	Binary decision variable indicating whether edge e is used
$z \in \mathbb{R}$	Auxiliary variable representing the worst-case path cost

Discrete-Scenario Min–Max Shortest-Path Model

$$\min_{x, z} \quad z \tag{3.1}$$

$$\text{s.t.} \quad z \geq \sum_{e \in E} c_e^\omega x_e \quad \forall \omega \in \Omega, \tag{3.2}$$

$$\sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = \begin{cases} 1, & v = s, \\ -1, & v = t, \\ 0, & \text{otherwise,} \end{cases} \quad \forall v \in V, \tag{3.3}$$

$$x_e \in \{0, 1\} \quad \forall e \in E, \tag{3.4}$$

$$z \in \mathbb{R}. \tag{3.5}$$

The solution represents a path that is feasible for all scenarios and optimal with respect to the worst-case cost realization. While this formulation provides strong robustness guarantees, its size grows linearly with the number of scenarios and can become computationally demanding.

3-4 Budgeted Uncertainty Min–Max Model

Table 3.2: Notation for the budgeted-uncertainty min–max shortest-path model

Symbol	Description
$G = (V, E)$	Directed graph representing the routing environment
V	Set of nodes (vertices), indexed by node IDs
$E \subseteq V \times V$	Set of directed edges (arcs)
$s, t \in V$	Fixed start node s and target node t
$\Omega = \{0, 1, \dots, \Omega - 1\}$	Index set of scenarios used to construct the uncertainty set
c_e^ω	Cost of edge $e \in E$ in scenario $\omega \in \Omega$
\underline{c}_e	Nominal (average) cost of edge e , defined as $\underline{c}_e = \frac{1}{ \Omega } \sum_{\omega \in \Omega} c_e^\omega$
d_e	Maximum deviation of edge e from nominal cost, $d_e = \max_{\omega \in \Omega} c_e^\omega - \underline{c}_e$
Γ	Budget of uncertainty
$\delta^+(v)$	Set of outgoing edges of node $v \in V$
$\delta^-(v)$	Set of incoming edges of node $v \in V$
$x_e \in \{0, 1\}$	Binary decision variable indicating whether edge e is used
$\pi \in \mathbb{R}_+$	Dual variable associated with the uncertainty budget Γ
$\rho_e \in \mathbb{R}_+$	Auxiliary variable capturing edge-specific deviation cost

Budgeted-Uncertainty Min–Max Shortest-Path Model

$$\min_{x, \pi, \rho} \quad \sum_{e \in E} \underline{c}_e x_e + \Gamma \pi + \sum_{e \in E} \rho_e \quad (3.6)$$

$$\text{s.t.} \quad \pi + \rho_e \geq d_e x_e \quad \forall e \in E, \quad (3.7)$$

$$\sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = \begin{cases} 1, & v = s, \\ -1, & v = t, \\ 0, & \text{otherwise,} \end{cases} \quad \forall v \in V, \quad (3.8)$$

$$x_e \in \{0, 1\} \quad \forall e \in E, \quad (3.9)$$

$$\pi \geq 0, \rho_e \geq 0 \quad \forall e \in E. \quad (3.10)$$

3-5 D* Lite

Table 3.3: Notation for the D* Lite incremental shortest-path algorithm

Symbol	Description
$G = (S, E)$	Directed graph representing the planning environment
S	Set of vertices (grid cells)
$E \subseteq S \times S$	Set of directed edges between adjacent vertices
$s_{\text{start}} \in S$	Current start vertex (robot position)
$s_{\text{goal}} \in S$	Goal vertex (temporary or final destination)
$\delta^+(s)$	Set of successor vertices of $s \in S$
$\delta^-(s)$	Set of predecessor vertices of $s \in S$
$c(i, j)$	Traversal cost of directed edge $(i, j) \in E$, where $i, j \in S$
$g(s)$	Current estimate of the shortest-path cost from s to s_{goal}
$\text{rhs}(s)$	One-step lookahead estimate of the cost-to-go from s
$h(i, j)$	Admissible and consistent heuristic estimate between vertices $i, j \in S$
$k(s) = (k_1(s), k_2(s))$	Lexicographic priority key associated with vertex s
k_m	Heuristic shift variable accounting for movement of the start vertex
U	Priority queue of locally inconsistent vertices $s \in S$
$u \in S$	Temporary vertex selected from the priority queue U for expansion
$\Omega = \{0, \dots, \Omega - 1\}$	Index set of spatio-temporal cost scenarios
$c^\omega(i, j)$	Cost of edge $(i, j) \in E$ in scenario $\omega \in \Omega$
$\text{ComputeShortestPath}()$	Procedure that incrementally restores local consistency
$\text{UpdateVertex}(s)$	Procedure that updates $\text{rhs}(s)$ and queue membership

All theoretical properties and correctness guarantees of D* Lite are discussed in Chapter 2.

Modeling Differences. Compared to the reference formulation by Koenig and Likhachev [KL02], the following modeling assumptions are adopted:

- a 4-connected grid graph instead of 8-connectivity,
- initial edge costs are given by the first STRF realization rather than uniform default values,
- edges are never blocked; unfavorable regions are modeled exclusively via increased traversal costs.

These choices ensure structural consistency with the robust optimization models and isolate uncertainty effects to edge costs.

Scenario-Based Cost Evolution. The original D* Lite algorithm assumes static edge costs that may be revealed incrementally. To model spatio-temporal cost uncertainty, the implementation is extended to operate on a sequence of time-indexed cost maps derived from spatio-temporal random field (STRF) realizations.

During execution, the agent advances greedily according to locally consistent g -values. After a fixed number of movement steps, the active scenario index is advanced. Only those edges whose costs differ between consecutive scenarios are updated, and the standard D* Lite update rules are applied. Local consistency is restored incrementally via *ComputeShortestPath()*, thereby repairing the existing backward search rather than recomputing paths from scratch.

Coupling Robust Planning and Incremental Execution. To investigate the interaction between ex-ante robustness and ex-post adaptivity, D* Lite can optionally be guided by a robust offline shortest path. The robust path is reduced to a coarse sequence of intermediate waypoints, referred to as *beacons*, obtained by uniformly subsampling interior nodes.

Beacons act as temporary goals and are activated according to a scenario-based schedule or once reached. Whenever a new beacon becomes active, the D* Lite search state is fully reinitialized, deliberately discarding cross-goal reuse. This design enforces a clear separation between global guidance and local adaptation and serves as an experimental control mechanism rather than a new planning algorithm.

Algorithmic Structure and Pseudocode Mapping. The implementation follows the original D* Lite pseudocode by Koenig and Likhachev [KL02]. Core procedures are implemented without modification, while application-specific extensions are inserted at well-defined points in the control flow.

- **Initialize()** ({01–06}): initializes all g -values and rhs-values to infinity, sets the goal’s rhs to zero, and seeds the priority queue for backward search.
- **CalculateKey()** ({12–13}): computes the lexicographic priority key used to order vertices in the queue, incorporating the heuristic shift variable.
- **UpdateVertex()** ({07–09}): maintains queue consistency by inserting, updating, or removing vertices based on local consistency.
- **ComputeShortestPath()** ({10–28}): incrementally restores local consistency by propagating cost changes backward until the current start vertex is locally consistent.
- **Main()** ({29–48}): executes greedy forward motion, detects edge cost changes, applies updates, and triggers replanning as required.

The two application-specific extensions are embedded into the original control flow: (i) *ScenarioUpdate()*, which applies STRF-induced edge-cost changes incrementally during execution, and (ii) *ActivateNextBeacon()*, which implements milestone-based goal switching using robust offline paths.

Both extensions preserve the theoretical guarantees of D* Lite within each fixed-goal phase and are included solely to support controlled empirical comparison under spatio-temporal uncertainty.

CHAPTER 4

EXPERIMENTAL RESULTS

In our experiments, global path planners are evaluated with respect to solution quality and computational effort.

For each configuration, results are aggregated over runs using identical seeds across algorithms. We report the median realized costs and median runtime. Uncertainty is summarized via a non-parametric bootstrap 95% confidence interval of the median cost. Our CI is estimated by repeatedly resampling observed costs with replacement 1,000 times. Each resample is the same size as the original sample and we compute the median for each resample. We then take the 2.5th and 97.5th percentiles of those 1,000 medians as the 95% CI.

This combination provides a statistically sound and interpretable summary of algorithmic performance under stochastic spatio-temporal uncertainty.

Two-layer experiments:

- layer: run extremes to find out which parameters yield the most promising comparison results
- layer: re-run promising parameters in greater variability

Research Questions To achieve the stated research objective, this thesis addresses the following research questions through experimental evaluation:

- RQ1: How do ex-ante robust optimization and ex-post adaptive search-based planning methods compare in terms of path quality and realized traversal cost under identical spatio-temporal cost uncertainty?
- RQ2: What are the computational trade-offs between robust optimization models and incremental search methods when applied to shortest-path planning under uncertainty?
- RQ3: How does execution-time adaptivity influence the performance of planning approaches in environments with dynamically evolving traversal costs?
- RQ4: Can a hybrid planning approach that combines robust ex-ante guidance with incremental ex-post replanning improve performance compared to using either paradigm in isolation?

4-1 Baseline

These are the baseline parameters we test our models with and compare to. To keep our experiments reasonable and comparable, interesting parameters will be changed with the remaining parameters unchanged.

Table 4.1: Baseline experimental configuration

Parameter	Symbol/ Description	Value
Grid size	$N_p \times N_q$	20×20
Anisotropy	(a_p, a_q, a_t)	$(1.0, 1.0, 1.0)$
Number of scenarios	$ \mathcal{T} $	10
Spatial scaling factor	α	0.25
Temporal scaling factor	β	0.30
Gamma scaling factor	λ	0.10
Spatial length scales	(ℓ_p, ℓ_q)	$(\alpha N_p, \alpha N_q)$
Temporal length scale	ℓ_t	$\beta \mathcal{T} $
Variance	σ^2	1.0
Robustness budget	Γ	$\Gamma(\lambda) = \lambda \Gamma_{\max}$
Covariance model	—	Gaussian
Seed range	—	1–100
Milestone cap (D* Lite)	—	10

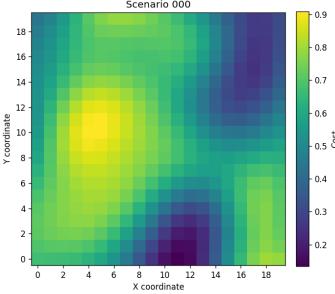
Algorithm	Cost (median)	95% CI	Runtime (ms, median)
Discrete Uncertainty	17.19	[16.55, 17.98]	287.55
Budgeted Uncertainty	17.04	[16.37, 17.35]	85.43
D* Lite	18.05	[17.38, 18.92]	230.96
D* Lite w. discrete guidance	16.28	[15.33, 17.00]	231.38
D* Lite w. budgeted guidance	15.33	[14.57, 16.12]	231.29

Table 4.2: Performance comparison under the baseline configuration.

Length Scale To ensure comparability across instance sizes, correlation lengths are defined relative to the extent of the modeled domain. Throughout all experiments, the spatial discretization is kept fixed, i.e., the cell size satisfies $\Delta p = \Delta q = 1$, and correlation lengths are therefore expressed directly in grid units. Spatial correlation lengths are scaled proportionally with grid size according to

$$\ell_p = \alpha N_p, \quad \ell_q = \alpha N_q,$$

where the scaling constant α is calibrated from a reference configuration. This construction keeps the ratio ℓ/N constant and thereby preserves the relative spatial smoothness of the generated cost fields across grid resolutions.



Many real-world planning and routing decisions must be made under conditions of uncertainty and dynamic change. Traversal costs may depend on environmental factors such as weather conditions, terrain properties, or sensor noise and may vary over both space and time.

This illustration shows how heterogeneous environmental sensor data can be spatially aggregated into a continuous cost field that serves as input for graph-based routing models.

Figure 4.1: Visualization of the baseline cost field for the first time frame.

Temporal correlation is defined relative to the scenario horizon $|\mathcal{T}|$ (with fixed time step). When $|\mathcal{T}|$ is fixed (e.g., $|\mathcal{T}| = 10$), the temporal correlation length ℓ_t is kept constant; if $|\mathcal{T}|$ changes, we set

$$\ell_t = \beta |\mathcal{T}|,$$

with β calibrated analogously, so that the ratio $\ell_t/|\mathcal{T}|$ remains constant.

The scaling constants are calibrated from the baseline configuration, yielding $\alpha = \ell_p^0/N_p^0 = \ell_q^0/N_q^0 = 0.25$ and $\beta = \ell_t^0/|\mathcal{T}|^0 = 0.3$.

Uncertainty Budget Since the budget parameter Γ is continuous in our implementation, we parameterize robustness via a normalized factor $\lambda \in [0, 1]$ and set

$$\Gamma(\lambda) = \lambda \Gamma_{\max}, \quad \Gamma_{\max} := L_{\min} = (N_p - 1) + (N_q - 1),$$

where L_{\min} is the minimum number of edges on an $s-t$ path in the 4-connected grid. Thus, $\lambda = 0$ yields the nominal model, while $\lambda = 1$ corresponds to full conservatism. We use $\lambda = 0.1$ as the baseline robustness level.

4-2 Experiment 1: Urban vs. Regional Scale

Parameter	Baseline	Tested values
Grid Size	20×20	$\{50 \times 50, 75 \times 75, 100 \times 100\}$

Table 4.3: Experiment 1: Parameter variations relative to the baseline configuration

By increasing the overall grid size, we also scale our spatial length scale (l_x, l_y) as well as our uncertainty budget Γ .

Table 4.4: Performance comparison across grid sizes

	Algorithm	Cost (median)	95% CI	Runtime (ms, median)
20*	Discrete Uncertainty	17.19	[16.55, 17.98]	287.55
	Budgeted Uncertainty	17.04	[16.37, 17.35]	85.43
	\times D* Lite	18.05	[17.38, 18.92]	230.96
	D* Lite w. Discrete guidance	16.28	[15.33, 17.00]	231.38
	D* Lite w. Budgeted guidance	15.33	[14.57, 16.12]	231.29
50	Discrete Uncertainty	44.61	[42.92, 45.97]	3160.54
	Budgeted Uncertainty	43.83	[42.71, 45.00]	3989.21
	\times D* Lite	47.10	[44.77, 48.91]	2011.42
	D* Lite w. Discrete guidance	42.21	[40.16, 44.06]	1951.31
	D* Lite w. Budgeted guidance	39.45	[37.56, 41.49]	1943.66
75	Discrete Uncertainty	68.07	[65.09, 69.42]	10354.67
	Budgeted Uncertainty	66.27	[64.08, 67.79]	41848.94
	\times D* Lite	70.81	[67.90, 74.09]	5985.83
	D* Lite w. Discrete guidance	63.25	[60.74, 65.59]	5873.51
	D* Lite w. Budgeted guidance	59.39	[56.82, 62.16]	5862.06
100	Discrete Uncertainty	90.90	[87.12, 92.82]	26568.95
	Budgeted Uncertainty	88.91	[85.00, 90.72]	61120.00
	\times D* Lite	94.80	[90.50, 100.33]	14442.30
	D* Lite w. Discrete guidance	85.25	[81.03, 87.82]	14037.08
	D* Lite w. Budgeted guidance	79.48	[75.90, 82.99]	14346.51

* Baseline configuration.

4-3 Experiment 2: Short vs. Long Exposure

In the short–long exposure experiment we vary the number of scenarios $|\mathcal{T}|$ while keeping the temporal length scale ℓ_t fixed.

Parameter	Baseline	Tested values
Number of Scenarios	10	{1, 5, 20, 50, 100}

Table 4.5: Experiment 2: Parameter variations relative to the baseline configuration

Table 4.6: Performance comparison across exposure levels $|\mathcal{T}|$

	Algorithm	Cost (median)	95% CI	Runtime (ms, median)
$ \mathcal{T} = 1$	Discrete Uncertainty	12.35	[11.58, 13.07]	80.74
	Budgeted Uncertainty	12.35	[11.58, 13.11]	39.52
	D* Lite	13.59	[12.64, 14.80]	40.46
	D* Lite w. Discrete guidance	12.35	[11.52, 13.11]	40.67
	D* Lite w. Budgeted guidance	12.35	[11.54, 13.10]	40.64
$ \mathcal{T} = 5$	Discrete Uncertainty	15.18	[14.11, 15.97]	167.74
	Budgeted Uncertainty	14.70	[14.17, 15.37]	74.12
	D* Lite	15.94	[15.17, 17.22]	123.71
	D* Lite w. Discrete guidance	14.56	[13.75, 15.10]	123.69
	D* Lite w. Budgeted guidance	13.78	[12.71, 14.35]	123.77
$ \mathcal{T} = 10^*$	Discrete Uncertainty	17.19	[16.55, 17.98]	287.55
	Budgeted Uncertainty	17.04	[16.37, 17.35]	85.43
	D* Lite	18.05	[17.38, 18.92]	230.96
	D* Lite w. Discrete guidance	16.28	[15.33, 17.00]	231.38
	D* Lite w. Budgeted guidance	15.33	[14.57, 16.12]	231.29
$ \mathcal{T} = 50$	Discrete Uncertainty	18.94	[18.57, 19.91]	1406.50
	Budgeted Uncertainty	19.12	[18.82, 19.58]	167.20
	D* Lite	18.99	[18.14, 19.57]	1064.51
	D* Lite w. Discrete guidance	17.64	[17.36, 17.95]	1066.09
	D* Lite w. Budgeted guidance	17.40	[16.90, 18.06]	1065.92
$ \mathcal{T} = 100$	Discrete Uncertainty	18.84	[18.17, 19.59]	2839.01
	Budgeted Uncertainty	19.80	[19.53, 20.18]	278.15
	D* Lite	19.28	[18.58, 19.93]	2011.47
	D* Lite w. Discrete guidance	19.00	[17.90, 19.56]	1997.88
	D* Lite w. Budgeted guidance	18.53	[17.49, 18.97]	2000.87

* Baseline configuration.

4-4 Experiment 3: Rough vs. Smooth Terrain

By varying the spatial scaling factor α , we change the spatial length scales and thus the terrain roughness.

Parameter	Baseline	Tested values
Spatial scaling factor α	0.25	{0.10, 0.50}

Table 4.7: Experiment 4: Parameter variations relative to the baseline configuration

Table 4.8: Performance comparison across spatial scaling factors α

	Algorithm	Cost (median)	95% CI	Runtime (ms, median)
$\alpha = 0.10$	Discrete Uncertainty	16.47	[16.03, 16.90]	281.89
	Budgeted Uncertainty	16.52	[16.15, 17.00]	77.05
	D* Lite	18.46	[18.11, 19.10]	231.20
	D* Lite w. Discrete guidance	15.49	[15.18, 16.03]	230.87
	D* Lite w. Budgeted guidance	15.28	[14.80, 16.00]	230.76
$\alpha = 0.25^*$	Discrete Uncertainty	17.19	[16.55, 17.98]	287.55
	Budgeted Uncertainty	17.04	[16.37, 17.35]	85.43
	D* Lite	18.05	[17.38, 18.92]	230.96
	D* Lite w. Discrete guidance	16.28	[15.33, 17.00]	231.38
	D* Lite w. Budgeted guidance	15.33	[14.57, 16.12]	231.29
$\alpha = 0.50$	Discrete Uncertainty	16.91	[15.75, 18.56]	271.63
	Budgeted Uncertainty	17.45	[16.90, 18.31]	83.14
	D* Lite	18.46	[16.22, 19.84]	228.03
	D* Lite w. Discrete guidance	16.75	[14.83, 17.98]	227.85
	D* Lite w. Budgeted guidance	14.97	[14.36, 16.68]	228.19

* Baseline configuration.

4-5 Experiment 4: Temporal Stability

By varying the temporal scaling factor β , we change the temporal length scale and thus the stability across scenarios.

Parameter	Baseline	Tested values
Temporal scaling factor β	0.30	{0.10, 0.60}

Table 4.9: Experiment 6: Parameter variations relative to the baseline configuration

Table 4.10: Performance comparison across temporal scaling factors β

	Algorithm	Cost (median)	95% CI	Runtime (ms, median)
$\beta = 0.10$	Discrete Uncertainty	18.69	[17.80, 19.55]	324.63
	Budgeted Uncertainty	18.47	[18.11, 19.00]	105.77
	D* Lite	18.97	[18.26, 19.93]	237.79
	D* Lite w. Discrete guidance	17.45	[16.53, 18.08]	237.79
	D* Lite w. Budgeted guidance	16.35	[15.82, 16.99]	237.63
$\beta = 0.30^*$	Discrete Uncertainty	17.19	[16.55, 17.98]	287.55
	Budgeted Uncertainty	17.04	[16.37, 17.35]	85.43
	D* Lite	18.05	[17.38, 18.92]	230.96
	D* Lite w. Discrete guidance	16.28	[15.33, 17.00]	231.38
	D* Lite w. Budgeted guidance	15.33	[14.57, 16.12]	231.29
$\beta = 0.60$	Discrete Uncertainty	15.32	[14.53, 16.13]	284.08
	Budgeted Uncertainty	14.89	[14.42, 15.33]	88.66
	D* Lite	16.09	[15.41, 17.25]	235.29
	D* Lite w. Discrete guidance	14.72	[13.83, 15.22]	235.18
	D* Lite w. Budgeted guidance	13.85	[12.98, 14.50]	234.96

* Baseline configuration.

4-6 Experiment 5: Uncertainty Budgeted

By varying the robustness budget scaling factor λ , we change the uncertainty budget $\Gamma = \lambda\Gamma_{\max}$.

Parameter	Baseline	Tested values
Gamma scaling factor λ	0.10	{0, 0.25, 0.50, 1.00}

Table 4.11: Experiment 7: Parameter variations relative to the baseline configuration

Table 4.12: Performance comparison across budget levels λ

	Algorithm	Cost (median)	95% CI	Runtime (ms, median)
$\lambda = 0.10^*$	Discrete Uncertainty	17.19	[16.60, 17.97]	291.86
	Budgeted Uncertainty	↓ 15.77 ↓	[15.18, 16.10]	↓ 49.30 ↓
	D* Lite	18.05	[17.38, 18.92]	232.22
	D* Lite w. Discrete guidance	16.28	[15.41, 17.00]	232.12
	D* Lite w. Budgeted guidance	15.24	[14.57, 15.95]	231.65
$\lambda = 0.10^*$	Discrete Uncertainty	17.19	[16.55, 17.98]	287.55
	Budgeted Uncertainty	17.04	[16.37, 17.35]	85.43
	D* Lite	18.05	[17.38, 18.92]	230.96
	D* Lite w. Discrete guidance	16.28	[15.33, 17.00]	231.38
	D* Lite w. Budgeted guidance	15.33	[14.57, 16.12]	231.29
$\lambda = 0.10^*$	Discrete Uncertainty	17.19	[16.60, 17.96]	287.01
	Budgeted Uncertainty	↑ 18.25 ↑	[17.74, 18.75]	↑ 433.59 ↑
	D* Lite	18.05	[17.38, 18.90]	235.27
	D* Lite w. Discrete guidance	16.28	[15.33, 17.05]	234.16
	D* Lite w. Budgeted guidance	15.59	[14.63, 16.28]	233.42
$\lambda = 0.50$	Discrete Uncertainty	17.19	[16.60, 17.98]	315.02
	Budgeted Uncertainty	19.73	[19.31, 20.32]	3415.74
	D* Lite	18.05	[17.38, 18.90]	313.06
	D* Lite w. Discrete guidance	16.28	[15.42, 17.05]	268.97
	D* Lite w. Budgeted guidance	15.75	[15.14, 16.53]	254.03
$\lambda = 1.00$	Discrete Uncertainty	17.19	[16.55, 17.97]	323.56
	Budgeted Uncertainty	21.64	[21.00, 22.02]	45773.08
	D* Lite	18.05	[17.36, 18.92]	333.87
	D* Lite w. Discrete guidance	16.28	[15.32, 17.07]	273.82
	D* Lite w. Budgeted guidance	15.75	[15.15, 16.59]	257.00

* Baseline configuration.

4-7 Experiment 6: Number of Beacons

In this experiment, we study the effect of milestone-based replanning frequency by varying the number of beacons used during execution. Increasing the number of beacons corresponds to more frequent replanning opportunities along the path.

Parameter	Baseline	Tested values
Number of beacons	5	{1, 5, 20}

Table 4.13: Experiment 8: Parameter variations relative to the baseline configuration

Table 4.14: Performance comparison across beacon counts

	Algorithm	Cost (median)	95% CI	Runtime (ms, median)
1 Beacon	Discrete Uncertainty	17.19	[16.60, 17.98]	282.84
	Budgeted Uncertainty	17.04	[16.46, 17.35]	85.14
	D* Lite	18.05	[17.38, 18.90]	229.52
	D* Lite w. Discrete guidance	16.80	[16.04, 17.65]	229.42
	D* Lite w. Budgeted guidance	16.35	[15.81, 17.35]	229.04
5* Beacons	Discrete Uncertainty	17.19	[16.48, 17.97]	276.43
	Budgeted Uncertainty	17.04	[16.40, 17.34]	83.11
	D* Lite	18.05	[17.36, 18.90]	226.48
	D* Lite w. Discrete guidance	15.92	[15.01, 16.72]	224.72
	D* Lite w. Budgeted guidance	15.39	[14.54, 16.28]	224.59
20 Beacons	Discrete Uncertainty	17.19	[16.60, 17.97]	279.12
	Budgeted Uncertainty	17.04	[16.40, 17.35]	83.22
	D* Lite	18.05	[17.38, 18.88]	227.46
	D* Lite w. Discrete guidance	16.45	[15.51, 17.14]	227.22
	D* Lite w. Budgeted guidance	15.57	[14.64, 16.29]	226.85

* Baseline configuration.

CHAPTER 5

CONCLUSION

Our experiments show that combining robust combinatorial optimization handling uncertainty via uncertainty sets can be used for mobile robot path finding to a certain degree.

One possible adaptation that we find worth looking into in future research is robust optimization for smaller scenarios/ windows. For example, imagine an environment with environmental sensors placed on "hotspots". We could map sensor data as costs on an unstructured grid and use this real time data as possible waypoints for a coupled-D*Lite approach. A robust combinatorial optimization method could be used beforehand to make out the best possible route using these possible waypoints to give a rough estimate route for D* Lite to then traverse through.

BIBLIOGRAPHY

- [AK23] Jaafar Ahmed Abdulsahib and Dheyaa Jasim Kadhim. Classical and Heuristic Approaches for Mobile Robot Path Planning: A Survey. *Robotics*, 12(4):93, June 2023.
- [AS24] Elnaz Azizi and Abbas Seifi. Shortest path network interdiction with incomplete information: A robust optimization approach. *Annals of Operations Research*, 335(2):727–759, April 2024.
- [ASSM⁺22] Amylia Ait Saadi, Assia Soukane, Yassine Meraïhi, Asma Benmes-saoud Gabis, Seyedali Mirjalili, and Amar Ramdane-Cherif. UAV Path Planning Using Optimization Approaches: A Survey. *Archives of Computational Methods in Engineering*, 29(6):4233–4284, October 2022.
- [BDG⁺24] Prathamesh Bagad, Prachi Dahatonde, Rushikesh Gagare, Ishwari Athare, and Gauri Ghule. Optimizing Pathfinding in Dynamic Environments: A Comparative Study of A* and D-Lite Algorithms. In *2024 IEEE Pune Section International Conference (PuneCon)*, pages 1–11, Pune, India, December 2024. IEEE.
- [CDG19] André Chassein, Trivikram Dokka, and Marc Goerigk. Algorithms and uncertainty sets for data-driven robust shortest path problems. *European Journal of Operational Research*, 274(2):671–686, April 2019.
- [Dij59] E W Dijkstra. A note on two problems in connexion with graphs. 1959.
- [DP84] Narsingh Deo and Chi-Yin Pang. Shortest-path algorithms: Taxonomy and annotation. *Networks*, 14(2):275–323, June 1984.
- [FB22] Joey G. Fernando and Myelinda Baldelovar. Decision Support System: Overview, Different Types and Elements. *TechnoareteTransactions on Intelligent Data Mining and Knowledge Discovery*, 2(2), May 2022.
- [FIU⁺15] M.P. Fanti, G. Iacobellis, W. Ukovich, V. Boschian, G. Georgoulas, and C. Stylios. A simulation based Decision Support System for logistics management. *Journal of Computational Science*, 10:86–96, September 2015.
- [GH23] Gopi Gugan and Anwar Haque. Path Planning for Autonomous Drones: Challenges and Future Directions. *Drones*, 7(3):169, February 2023.
- [GH24] Marc Goerigk and Michael Hartisch. *An Introduction to Robust Combinatorial Optimization: Concepts, Models and Algorithms for Decision*

Making under Uncertainty, volume 361 of *International Series in Operations Research & Management Science*. Springer Nature Switzerland, Cham, 2024.

- [GHF⁺24] Yang Gao, Qidong Han, Shuo Feng, Zhen Wang, Teng Meng, and Jingshuai Yang. Improvement and Fusion of D*Lite Algorithm and Dynamic Window Approach for Path Planning in Complex Environments. *Machines*, 12(8):525, August 2024.
- [GK25] Marc Goerigk and Jannis Kurtz. Data-driven prediction of relevant scenarios for robust combinatorial optimization. *Computers & Operations Research*, 174:106886, February 2025.
- [GL21] Marc Goerigk and Stefan Lendl. Robust Combinatorial Optimization with Locally Budgeted Uncertainty. *Open Journal of Mathematical Optimization*, 2:1–18, May 2021.
- [GRT⁺16] Chrysanthos E. Gounaris, Panagiotis P. Repoussis, Christos D. Tarantilis, Wolfram Wiesemann, and Christodoulos A. Floudas. An Adaptive Memory Programming Framework for the Robust Capacitated Vehicle Routing Problem. *Transportation Science*, 50(4):1239–1260, November 2016.
- [HHW24] Yong He, Ticheng Hou, and Mingran Wang. A new method for unmanned aerial vehicle path planning in complex environments. *Scientific Reports*, 14(1):9257, April 2024.
- [HNR68] Peter Hart, Nils 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.
- [Kee98] Peter B Keenan. Spatial decision support systems for vehicle routing. *Decision Support Systems*, 22(1):65–71, January 1998.
- [KL02] Sven Koenig and Maxim Likhachev. D* Lite. *AAAI-02 Proceedings*, 2002.
- [KLF04] Sven Koenig, Maxim Likhachev, and David Furcy. Lifelong Planning A*. *Artificial Intelligence*, 155(1-2):93–146, May 2004.
- [KNK24] Kornél Katona, Husam A. Neamah, and Péter Korondi. Obstacle Avoidance and Path Planning Methods for Autonomous Navigation of Mobile Robot. *Sensors*, 24(11):3573, June 2024.
- [KSDS21] Karthik Karur, Nitin Sharma, Chinmay Dharmatti, and Joshua E. Siegel. A Survey of Path Planning Algorithms for Mobile Robots. *Vehicles*, 3(3):448–468, August 2021.

- [LD25] Yaping Lu and Chen Da. Global and local path planning of robots combining ACO and dynamic window algorithm. *Scientific Reports*, 15(1):9452, March 2025.
- [LLZ⁺24] Xiaomei Li, Ye Lu, Xiaoyu Zhao, Xiong Deng, and Zhijiang Xie. Path planning for intelligent vehicles based on improved D* Lite. *The Journal of Supercomputing*, 80(1):1294–1330, January 2024.
- [LRS21] Philippe Lacomme, Gwénaël Rault, and Marc Sevaux. Integrated decision support system for rich vehicle routing problems. *Expert Systems with Applications*, 178:114998, September 2021.
- [MAR⁺17] Amgad Madkour, Walid G. Aref, Faizan Ur Rehman, Mohamed Abdur Rahman, and Saleh Basalamah. A Survey of Shortest-Path Algorithms, May 2017.
- [MPW26] Dmytro Matsypura, Yu Pan, and Hanzhao Wang. Learning Shortest Paths When Data is Scarce, January 2026.
- [MSY⁺25] Lingkai Meng, Yu Shao, Long Yuan, Longbin Lai, Peng Cheng, Xue Li, Wenyuan Yu, Wenjie Zhang, Xuemin Lin, and Jingren Zhou. Revisiting Graph Analytics Benchmark. *Proceedings of the ACM on Management of Data*, 3(3):1–28, June 2025.
- [MSZH22] Sebastian Müller, Lennart Schüler, Alraune Zech, and Falk Heße. GSTools v1.3: A toolbox for geostatistical modelling in Python. *Geoscientific Model Development*, 15(7):3161–3182, April 2022.
- [Pow02] Daniel J. Power. *Decision Support Systems: Concepts and Resources for Managers*. Quorum Books, Westport, Conn., 1. publ edition, 2002.
- [PS07] Daniel J. Power and Ramesh Sharda. Model-driven decision support systems: Concepts and research directions. *Decision Support Systems*, 43(3):1044–1061, April 2007.
- [QSW⁺23] Hongwei Qin, Shiliang Shao, Ting Wang, Xiaotian Yu, Yi Jiang, and Zonghan Cao. Review of Autonomous Path Planning Algorithms for Mobile Robots. *Drones*, 7(3):211, March 2023.
- [Rai19] Juste Raimbault. Second-order control of complex systems with correlated synthetic data. *Complex Adaptive Systems Modeling*, 7(1):4, December 2019.
- [RGRM24] Mohsen Roytvand Ghiasvand, Donya Rahmani, and Mohammad Moshref-Javadi. Data-driven robust optimization for a multi-trip truck-drone routing problem. *Expert Systems with Applications*, 241:122485, May 2024.

- [RMA04] Rubén Ruiz, Concepción Maroto, and Javier Alcaraz. A decision support system for a real vehicle routing problem. *European Journal of Operational Research*, 153(3):593–606, March 2004.
- [Ste94] Anthony Stentz. The D* Algorithm for Real-Time Planning of Optimal Traverses. 1994.
- [UHV17] Marlin W. Ulmer, Leonard Heilig, and Stefan Voß. On the Value and Challenge of Real-Time Information in Dynamic Dispatching of Service Vehicles. *Business & Information Systems Engineering*, 59(3):161–171, June 2017.
- [WLJ⁺22] Huanwei Wang, Shangjie Lou, Jing Jing, Yisen Wang, Wei Liu, and Tieming Liu. The EBS-A* algorithm: An improved A* algorithm for path planning. *PLOS ONE*, 17(2):e0263841, February 2022.
- [WQL⁺21] Huanwei Wang, Xuyan Qi, Shangjie Lou, Jing Jing, Hongqi He, and Wei Liu. An Efficient and Robust Improved A* Algorithm for Path Planning. *Symmetry*, 13(11):2213, November 2021.
- [WWW⁺25] Xuan Wu, Di Wang, Lijie Wen, Yubin Xiao, Chunguo Wu, Yuesong Wu, Chaoyu Yu, Douglas L. Maskell, and You Zhou. Neural Combinatorial Optimization Algorithms for Solving Vehicle Routing Problems: A Comprehensive Survey with Perspectives, April 2025.
- [XXG⁺25] Longyan Xu, Mao Xi, Ren Gao, Ziheng Ye, and Zaihan He. Dynamic path planning of UAV with least inflection point based on adaptive neighborhood A* algorithm and multi-strategy fusion. *Scientific Reports*, 15(1):8563, March 2025.
- [YAB23] Vincent F. Yu, Pham Tuan Anh, and Roberto Baldacci. A robust optimization approach for the vehicle routing problem with cross-docking under demand uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 173:103106, May 2023.
- [ZLS⁺22] Yunfan Zhang, Feng Liu, Yifan Su, Yue Chen, Zhaojian Wang, and Joao P. S. Catalao. Two-Stage Robust Optimization Under Decision Dependent Uncertainty. *IEEE/CAA Journal of Automatica Sinica*, 9(7):1295–1306, July 2022.
- [ZM18] Mohd. Nayab Zafar and J.C. Mohanta. Methodology for Path Planning and Optimization of Mobile Robots: A Review. *Procedia Computer Science*, 133:141–152, 2018.

- [ZZL23] Aobei Zhang, Ying Zhang, and Yanqiu Liu. Robust Optimization of Electric Vehicle Paths in Uncertain Environments. *IEEE Access*, 11:124240–124251, 2023.

CHAPTER 6

APPENDIX

6-1 Geostatistical Formulations

$$\gamma(r) = \sigma^2 \left(1 - \text{cor} \left(s \cdot \frac{r}{\ell} \right) \right) + n, \quad (6.1)$$

$$C(r) = \sigma^2 \text{ cor} \left(s \cdot \frac{r}{\ell} \right), \quad (6.2)$$

$$\rho(r) = \text{cor} \left(s \cdot \frac{r}{\ell} \right). \quad (6.3)$$

$$h = \sqrt{\sum_{i=1}^d \left(\frac{r_i}{\ell_i} \right)^2}, \quad (6.4)$$

$$h(x, y, \tau) = \sqrt{\left(\frac{\Delta p}{\ell_p} \right)^2 + \left(\frac{\Delta q}{\ell_q} \right)^2 + \left(\frac{\Delta \tau}{\ell_t} \right)^2}. \quad (6.5)$$

$$C_m(\mathbf{r}, \Delta \tau) = C \left(\sqrt{\sum_{i=1}^2 \left(\frac{r_i}{\ell_i} \right)^2 + \left(\frac{\Delta \tau}{\ell_t} \right)^2} \right), \quad (6.6)$$

where $\mathbf{r} = (x - x', y - y')$ and $\Delta \tau = \tau - \tau'$.

$$U(\mathbf{x}) = \sqrt{\frac{\sigma^2}{N}} \sum_{i=1}^N (Z_{1,i} \cos(\mathbf{k}_i \cdot \mathbf{x}) + Z_{2,i} \sin(\mathbf{k}_i \cdot \mathbf{x})), \quad (6.7)$$

where $Z_{1,i}, Z_{2,i} \sim \mathcal{N}(0, 1)$ are independent standard normal variables and $\mathbf{k}_i \in \mathbb{R}^3$ are wave vectors sampled from the spectral density associated with the covariance model 6.2.

Table 6.1: Predefined covariance models in GSTools [MSZH22].

Model	$\text{cor}(h)$	Source
Gaussian	$\exp(-h^2)$	Webster and Oliver (2007)
Exponential	$\exp(-h)$	Webster and Oliver (2007)
Stable	$\exp(-h^\alpha)$	Wackernagel (2003)
Matérn	$\frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{\nu} h)^\nu \cdot K_\nu(\sqrt{\nu} h)$	Rasmussen and Williams (2005)
Rational	$\left(1 + \frac{h^2}{\alpha}\right)^{-\alpha}$	Rasmussen and Williams (2005)
Cubic	$\left(1 - 7h^2 + \frac{35}{4}h^3 - \frac{7}{2}h^5 + \frac{3}{4}h^7\right) \quad (h < 1)$	Chilès and Delfiner (2012)
Linear	$(1 - h) \quad (h < 1)$	Webster and Oliver (2007)
Circular	$\frac{2}{\pi} \left(\cos^{-1}(h) - h\sqrt{1-h^2} \right) \quad (h < 1)$	Webster and Oliver (2007)
Spherical	$\left(1 - \frac{3}{2}h + \frac{1}{2}h^3\right) \quad (h < 1)$	Webster and Oliver (2007)
HyperSpherical	$\left(1 - h \cdot \frac{{}_2F_1\left(\frac{1}{2}, -\frac{d-1}{2}, \frac{3}{2}, h^2\right)}{{}_2F_1\left(\frac{1}{2}, -\frac{d-1}{2}, \frac{3}{2}, 1\right)}\right) \quad (h < 1)$	Matérn (1960)
SuperSpherical	$\left(1 - h \cdot \frac{{}_2F_1\left(\frac{1}{2}, -\nu, \frac{3}{2}, h^2\right)}{{}_2F_1\left(\frac{1}{2}, -\nu, \frac{3}{2}, 1\right)}\right) \quad (h < 1)$	Matérn (1960)
JBessel	$\Gamma(\nu + 1) \cdot \left(\frac{h}{2}\right)^{-\nu} \cdot J_\nu(h)$	Chilès and Delfiner (2012)
TPLSimple	$(1 - h)^\nu \quad (h < 1)$	Wendland (1995)
TPLGaussian	$H \cdot E_{1+H}(h^2)$	Di Federico and Neuman (1997)
TPLExponential	$2H \cdot E_{1+2H}(h)$	Di Federico and Neuman (1997)
TPLStable	$\frac{2H}{\alpha} \cdot E_{1+\frac{2H}{\alpha}}(h^\alpha)$	Müller et al. (2021a)

Formulas including the subscript ($h < 1$) are piecewise-defined functions being constantly zero for $h > 1$. Here, h is the non-dimensional distance, d is the dimension, $\Gamma(x)$ is the gamma function, $K_\nu(x)$ is the modified Bessel function of the second kind, $J_\nu(x)$ is the Bessel function of the first kind, ${}_2F_1(a, b, c, x)$ is the ordinary hypergeometric function, and $E_\nu(x)$ is the exponential integral function (Abramowitz et al., 1972). All other variables are shape parameters of the respective models.