

Philipp Stockerl

Matr. No 107829

Chair of Business Decisions & Data Science

Prof. Dr. Marc Goerigk



Faculty of Business Administration  
and Economics

---

## Bachelor's Thesis Proposal: Static and Dynamic Vehicle Routing under Uncertainty

### 1 RESEARCH QUESTION

How much does dynamic adaptation (D\* Lite) improve cost efficiency and resilience compared to a static robust plan under spatially correlated uncertainty?

### 2 GAUSSIAN RANDOM FIELD

Creates Gaussian Random Field map that acts as data for comparison. The generator produces synthetic environments for further testing.

- Forecast edges: map used by the robust model
- Actual edges: map used by D\* Lite

Both maps are derived from the same Gaussian Random Field, so they are statistically correlated but not identical. The idea is that the robust, static model pre-plans the route using forecast data. The actual map contains data that the vehicle's sensors are going to pick up. Theoretically this should mimic a weather forecast versus real atmospheric conditions scenario.

- temporal sequence
- spatio-temporal uncertainty model

### Modeling Interpretation of Forecast and Actual GRF Maps

The Gaussian Random Field (GRF) generator produces two spatially correlated cost maps, denoted as the *forecast field* and the *actual field*, which represent different levels of environmental knowledge available to the planner. Both maps originate from the same base GRF and are normalized to a comparable cost range [1, 10] to ensure numerical consistency across models.

The **actual field** corresponds to the true cost surface  $c_i^{\text{actual}}$ , i.e., the underlying spatial reality as experienced by the dynamic path-planning algorithm (D\* Lite). In contrast, the **forecast field** represents a smoothed version of the same GRF obtained by applying a Gaussian blur with parameter  $\sigma = \text{BLUR\_SIGMA}$ . This blurring process systematically removes high-frequency local variations, thereby mimicking the loss of spatial detail that occurs in real-world forecasting, e.g., through weather prediction or low-resolution sensing.

The **uncertainty** between the forecast and actual environments is not artificially added but instead *emerges naturally* from this difference in spatial resolution. Specifically, for each edge  $i$ , the local deviation is computed as

$$d_i = |c_i^{\text{actual}} - c_i^{\text{forecast}}|,$$

which quantifies the discrepancy between predicted and realized costs. These deviations  $d_i$  serve as the uncertainty parameters in the BertsimasSim robust optimization model, where the planner anticipates that up to  $\Gamma$  of these deviations may simultaneously worsen the travel cost.

This formulation ensures that the two maps remain statistically comparable: both encode nominal costs  $c_i$ , but the forecast field reflects an *imperfect estimate* of the true environment. Consequently, the blurring operation itself functions as the uncertainty-generating mechanism, introducing spatially correlated forecast errors without modifying the nominal cost structure or artificially inflating costs.

## Nominal Cost Derivation from the Gaussian Random Field

The Gaussian Random Field (GRF)  $Z(x, y)$  provides the continuous spatial foundation for cost generation. Each grid point  $(x, y)$  represents an environmental condition (e.g., wind resistance, terrain difficulty, or congestion intensity) modeled as a realization of a zero-mean, stationary Gaussian process with variance  $\sigma^2$  and correlation length  $l$ :

$$Z(x, y) \sim \mathcal{GP}(0, k(r)), \quad \text{with} \quad k(r) = \sigma^2 \exp\left(-\frac{r^2}{2l^2}\right),$$

where  $r$  denotes the Euclidean distance between two locations. This formulation yields spatially correlated, smooth cost landscapes in which nearby regions exhibit similar values.

To obtain operationally meaningful cost data, the GRF values are linearly rescaled into a normalized cost range  $[1, 10]$ , defining the **nominal cost**  $c_i$  associated with each node or edge:

$$c_i = 1 + 9 \cdot \frac{Z_i - \min(Z)}{\max(Z) - \min(Z)}.$$

Here, lower values correspond to favorable traversal conditions, while higher values represent costly or risky regions. This normalization ensures comparability across different random field realizations and maintains consistent magnitude scales for optimization.

Two distinct nominal cost fields are then derived from the same base GRF:

- **Actual cost field**  $c_i^{\text{actual}}$ : obtained directly from the normalized GRF, preserving full local variation. It represents the true, fine-grained environment encountered by the D\* Lite path-planning algorithm.
- **Forecast cost field**  $c_i^{\text{forecast}}$ : generated by applying a Gaussian blur (with parameter  $\sigma_{\text{blur}}$ ) to the base GRF before normalization. The smoothing operation removes small-scale features, simulating the coarser spatial resolution of predictive models such as weather forecasts.

The difference between these two correlated but non-identical fields gives rise to the element-wise deviation term used in the robust optimization model:

$$d_i = |c_i^{\text{actual}} - c_i^{\text{forecast}}|.$$

Thus, the nominal costs  $c_i$  are statistically grounded quantities derived from the GRF, while the blurring process naturally introduces spatially structured uncertainty without artificial noise injection.

### 3 ROBUST MIN-MAX OPTIMIZATION UNDER BUDGET UNCERTAINTY

In operations planning route planning often has to take place under uncertain conditions. To account for discrepancies a certain threshold has to be considered. In this case possible deviation from the initial route occurs because of unpredictable changes in weather conditions. Although forecasting methods are able to pick up bad weather cells, their actual impact on route budget can vary.

$$\min_{x \in \mathcal{X}} \max_{c \in \mathcal{U}} \sum_{i \in [n]} c_i x_i \quad (1.0)$$

$$= \min \sum_{i \in [n]} c_i x_i + \Gamma \pi + \sum_{i \in [n]} \rho_i \quad (1.1)$$

$$\text{s.t.} \quad \pi + \rho_i \geq d_i x_i \quad \forall i \in [n] \quad (2)$$

$$x \in \mathcal{X} \quad (3)$$

$$\pi \geq 0 \quad (4)$$

$$\rho_i \geq 0 \quad \forall i \in [n] \quad (5)$$

$x_i$  Decision variable indicating whether edge  $i$  is included in the route ( $x_i = 1$  if edge  $i$  is chosen, 0 otherwise).

$c_i$  Nominal (forecast) cost associated with edge  $i$ , obtained from the Gaussian Random Field representing the predicted environment.

$d_i$  Maximum deviation of edge  $i$  from its nominal cost, defining the upper bound of uncertainty ( $c_i \in [c_i, c_i + d_i]$ ).

$\Gamma$  Budget of uncertainty controlling how many edges may simultaneously take on their worst-case deviations. Higher  $\Gamma$  leads to more conservative (robust) solutions.

$\pi$  Shared protection variable that captures the common level of protection against the  $\Gamma$  worst deviations. It represents the dual value of the uncertainty budget constraint in the BertsimasSim model.

$\rho_i$  Individual excess variable for edge  $i$ , representing the amount by which its deviation exceeds the shared protection level  $\pi$ . Only activated for edges that belong to the worst-case subset.

The data for the robust model is explained in the following table.

The nominal problem can be formulated as following:

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & \sum_{i \in [n]} c_i x_i \\ \text{s.t.} \quad & x \in \mathcal{X} \end{aligned} \quad (1)$$

### 4 INCREMENTAL PATH REPLANNING

Table 1: Mapping between GRF-generated data and robust optimization model inputs

<b>Generated File / Variable</b>	<b>Model Parameter</b>	<b>Description and Creation Process</b>
<code>nodes.csv</code>	Network nodes ( $V$ )	Each node represents a grid point in the Gaussian Random Field (GRF). Created by iterating over a $GRID\_SIZE \times GRID\_SIZE$ lattice with coordinates $(x, y)$ and unique node IDs.
<code>edges_forecast.csv</code>	Nominal cost $c_i$ and deviation $d_i$	Each edge $(u, v)$ connects adjacent nodes in 4-neighbour topology. <i>Nominal cost</i> $c_i$ is taken from the <code>forecast_field.npy</code> (blurred GRF). <i>Deviation</i> $d_i$ is the absolute difference between the forecast and actual field values at node $u$ , i.e. $d_i =  a_u - f_u $ . This represents uncertainty between predicted and real local conditions.
<code>edges_actual.csv</code>	Realized cost $c_i^{\text{actual}}$	Contains the same edge structure as the forecast but with the <i>actual</i> GRF values assigned as cost. Used to evaluate realized travel cost after the robust solution or D* Lite path is computed.
<code>forecast_field.npy</code>	Forecast cost surface (nominal map)	A blurred version of the base GRF produced using a Gaussian filter ( $\sigma = BLUR\_SIGMA$ ). Represents the spatially smoothed forecast that serves as input to the robust model.
<code>actual_field.npy</code>	Actual cost surface (realized map)	Identical to the base GRF (no white noise added). Represents the true underlying cost environment used by the D* Lite algorithm.
SEED	Reproducibility control	Global random seed ensuring identical GRF layouts between forecast and actual maps. Changing this seed produces a new synthetic environment while maintaining statistical comparability.