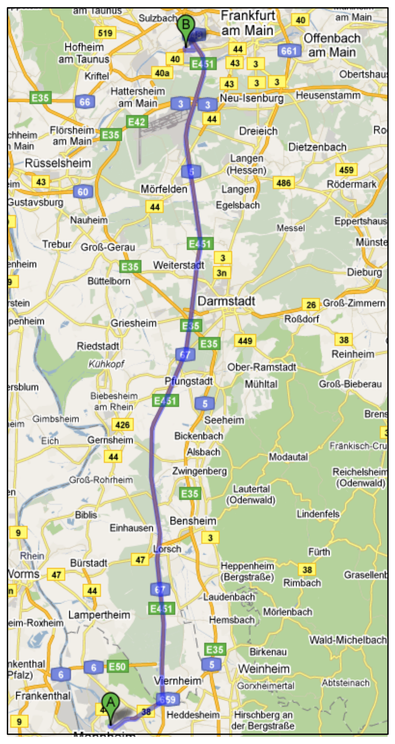
# Automatic Generation of Route Sketches

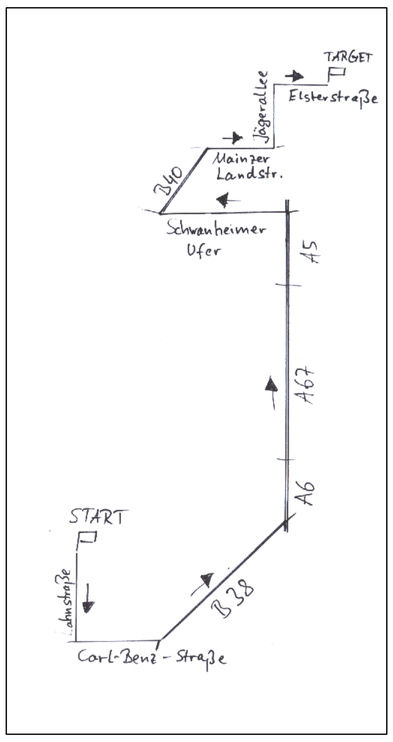
Andreas Gemsa[1], Martin Nöllenburg[1,2], Thomas Pajor[1] and Ignaz Rutter[1].

[1]Institute of Theoretical Informatics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

[2]Department of Computer Science, University of California, Irvine, USA



Google Maps [Mannheim to Frankfurt]




Hand drawn route sketch [Mannheim to Frankfurt]

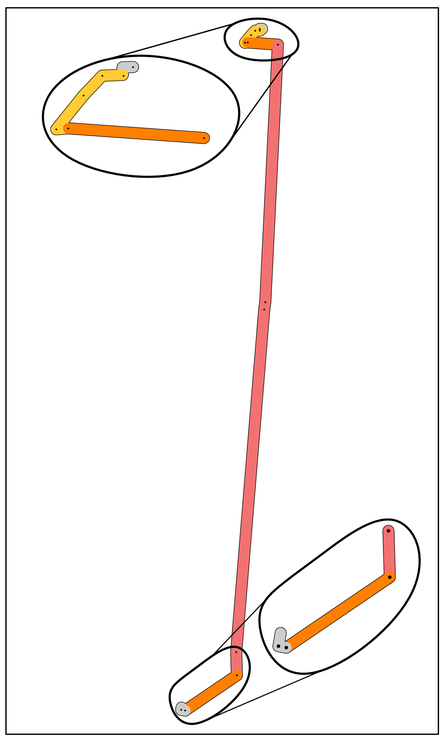
**Our Goal:**

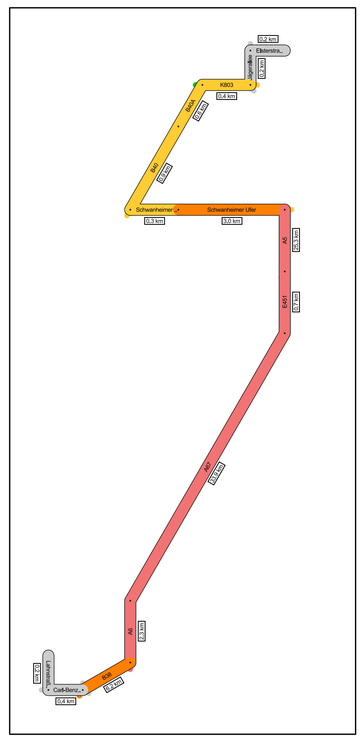
Provide *easy to read* route sketches

- reduce drawing complexity
- preserve the user's mental map
- every important edge is visible

**Applications:**

- travel directions on a piece of paper
- present multiple alternative routes





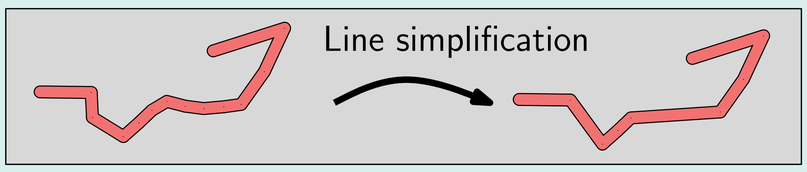
Simplified input route and automtically generated route sketch with our approach [Mannheim to Frankfurt].

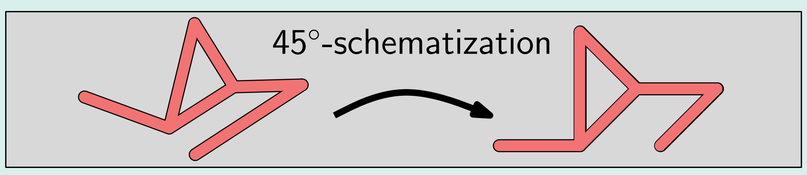
## Model

**Reducing the drawing complexity**

Exact geographic details are not necessary: simplify the input while maintaining the overall shape using the Douglas-Peucker algorithm.

Use only a small set of admissible edge slopes, namely $\mathcal{C}_d = \{z \cdot 90°/d \mid z \in \mathbb{Z}\}$ for a fixed $d \geq 1$, e.g., $d = 2$ for multiples of 45°.

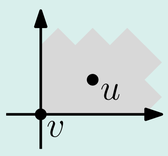



**Maintaining the mental map**

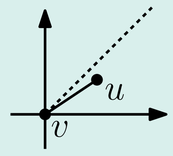
Maintain the *orthogonal order*, i.e., the top-bottom and left-right relationship between all vertex pairs.

The orthogonal order is preserved if $u$ is embedded to the upper right of $v$.



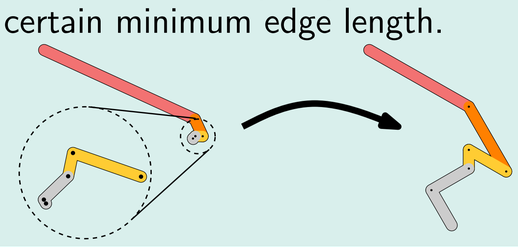

For every edge use the edge slope closest to its original slope if possible.

The orthogonal order allows for embedding $uv$ with 0°, 45° and 90°. An angle of 45° is preferable as it is closest to the original slope.




**Ensuring visibility of important edges**

Enforce that every edge has a certain minimum edge length.



**Route sketch generation problem**

Given a path $P$ with a plane embedding, an integer $d$ for angular resolution and a minimum edge length $\ell_{\min}$.

Generate a route sketch such that

- the orthogonal order of all vertex pairs is maintained,
- each edge slope is in $\mathcal{C}_d$, and
- each edge has length at least $\ell_{\min}$.

**Theorem:** The route sketch generation problem is NP-hard.

## Algorithm

Compute Shortest Path(s)

Simplify Graph

Schematize with MIP

## Heuristic Improvements

**Preserving Length Order:**

To retain some information about distances, we may require that the input length order of the edges is preserved.

**Relaxation of the Orthogonal Order:**

Preserving the orthogonal order for distant vertices is less important. For pairs of vertices whose distance in one coordinate is at least one third of the extent in that coordinate we do not preserve their order in the respective other coordinate.

## Experiments

1000 random shortest path queries on the German road network:

**Rectilinear Drawing:**

51.3% infeasible instances

107.36ms avg. running time

**Octilinear Drawing:**

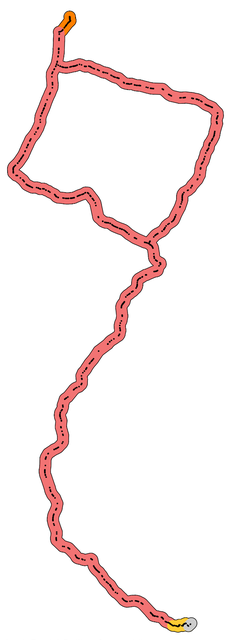
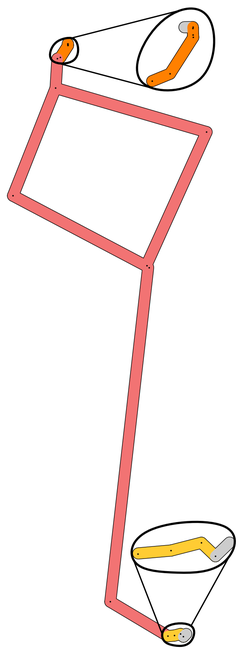
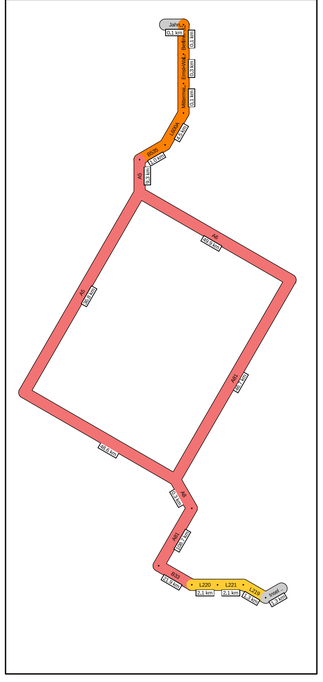
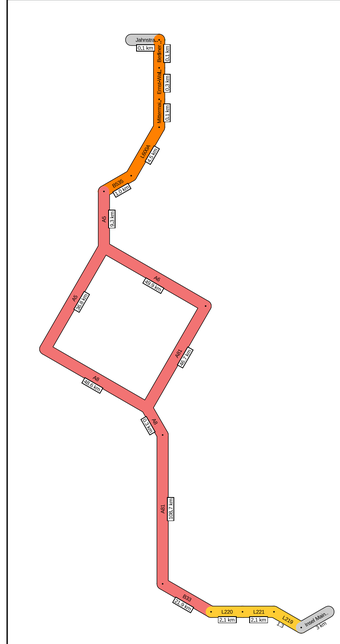
0.7% infeasible instances

363.49ms avg. running time

Single AMD Opteron 2218@2.6 Ghz, 2x1 MiB L2 Cache, 16GiB RAM, Linux 2.6.27.23, GCC 4.3.2, optimization level 3, Gurobi Solver 3.0.1

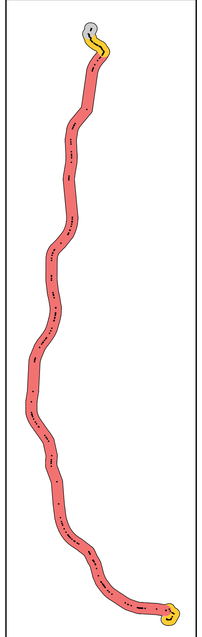
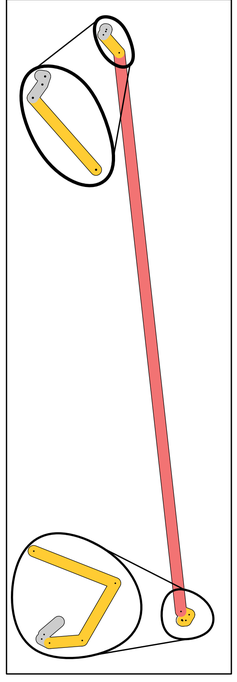
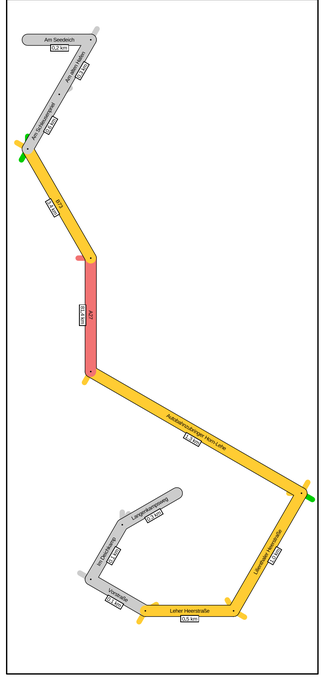
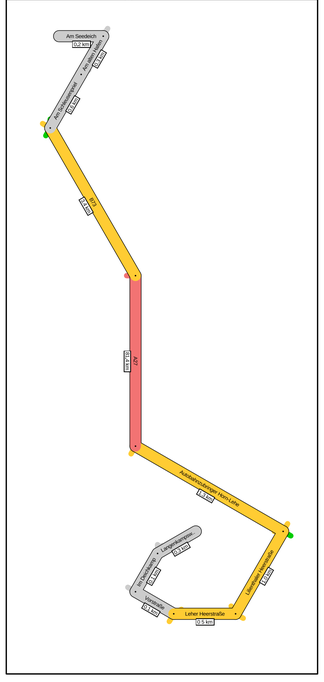
Route: Heidelberg - Konstanz

Routes through: Karlsruhe/Heilbronn

| Input Routes: | Simplified Routes: | Schematized by MIP | Schematized by MIP with both heuristics |
|---|---|---|---|
|  |  |  |  |

We thank Daniel Delling for providing us with sensible alternative routes.

Route: Bremen - Cuxhaven

| Input Route: | Simplified Route: | Schematized by MIP | Schematized by MIP with length order heuristic |
|---|---|---|---|
|  |  |  |  |