Quickly Compute Exact Shortest Paths with Changing Weights

**Problem:** Classical approaches too slow (text book approaches need several seconds)

**Challenges:**
- Traffic congestion constantly perturbates driving speeds.
- Use-specific needs. For example:
  - Trucks drive slower on the freeway.
  - Some people want to avoid highways.
- Huge network with millions of intersections.

**Requirements:**
- For Europe graph:
  - Path query must be fast \( \approx 1 \text{ ms} \)
  - Edge weights changeable in \( \approx 1 \text{ sec} \)
  - Preprocessing can be slow (Map updates are rare)

**Step 1: Preprocessing**

- Compute Nested Dissection Order using for example FlowCutter
- Compute Chordal Supergraph / Customizable Contraction Hierarchy
- Nodes ordered bottom to top according to order

**Step 2: Customization**

**Objective:**
- Introduce / Exchange weights

**Algorithm:**
- Enumerate all triangles bottom to top
- Path along triangle top always longer than along triangle bottom
- Lower triangle inequality

**Step 3: Path Query**

- Bidirectional Graph Search between source and target
  - Only follow upward edges
- Searches meet at high node
- Can use Elimination Tree instead of Dijkstra’s algorithm

---

**References:**
- Open source implementation: https://github.com/RoutingKit/RoutingKit