# Calculating time-dependent travel times for VRPs

Oddvar Kloster

SINTEF ICT

# Goal

- Solving the Dynamic Shortest Path Problem
  - Dynamic = link travel times vary with time
  - Shortest = fastest/cheapest/…
- Relevant aspects of travelling in a road network should be modelled
- Should handle networks of size ≈ all of Norway
- Solver should be suitable as a module in a VRP solver

# VRP solver environment

- Either departure time or arrival time may be given
- The set of possible locations is known in advance
- The interval of possible departure/arrival times is (usually) known in advance
- Many identical requests
- - but still very many distinct requests
- Speed is essential
- Most requests are for time and cost only

# Road network modelling

- One way roads
- Turning restrictions
- Restrictions on whether vehicle may use road (possibly time-dependent)
- Time-dependent travel times
  - Gradual changes (e.g. rush hour)
  - Abrupt changes (e.g. ferry timetables)

# Computational Model

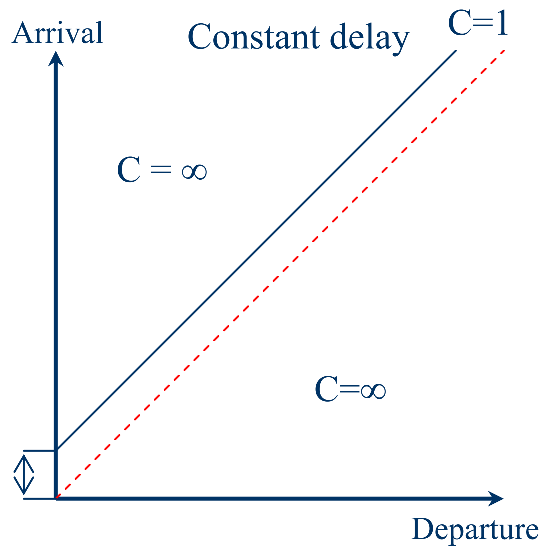
- Road network is transformed into a directed graph
  - One way roads
  - Turning restrictions
- Each edge has a cost function
  - Time-dependent cost and travel time
- Compute node labels (cost functions) by adapted Dijkstra
- Find shortest path by backtracking

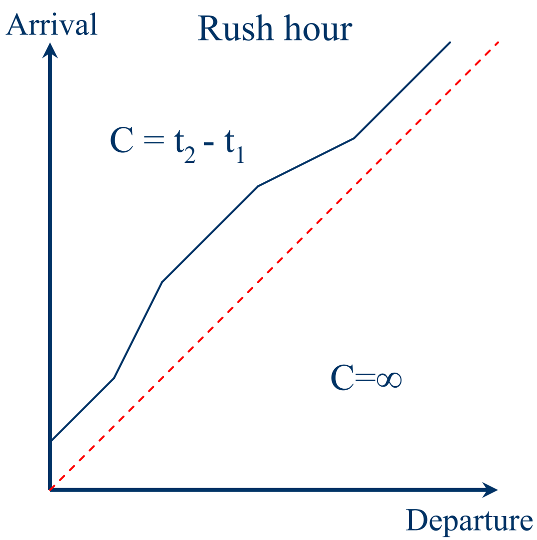
# Cost functions

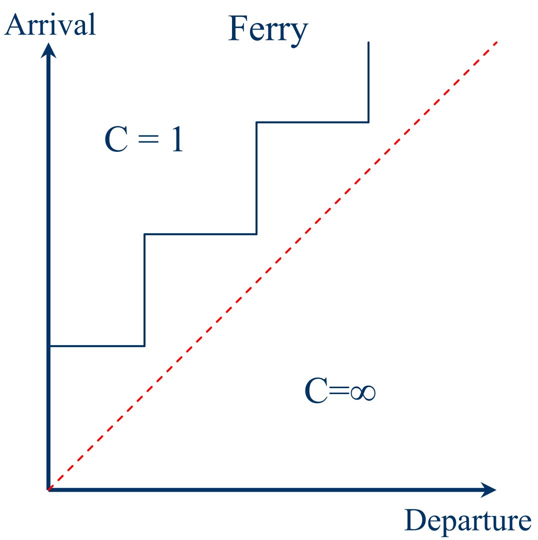
- "Normal" formulation: $c = f(t)$, $d = g(t)$

- Function $D^2 \rightarrow [0, \infty]$
- D: Time domain of interest
- $C(t_1, t_2)$ is the cost of travel when departing at time $t_1$ and arriving at time $t_2$
- $C(t_1, t_2) = \infty$: Illegal/impossible
- $t_2 < t_1 : C(t_1, t_2) = \infty$

# Cost function examples

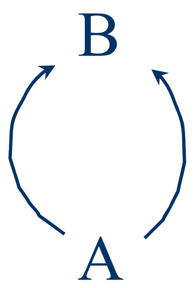


# Cost function operators

- min
  - Computes optimal cost function over alternative paths
  - Pointwise minimum




- +
  - Computes optimal cost function for concatenation of paths
  - Minimum of $C_{AB}(t_1, t_2) + C_{BC}(t_2, t_3)$ over $t_2$

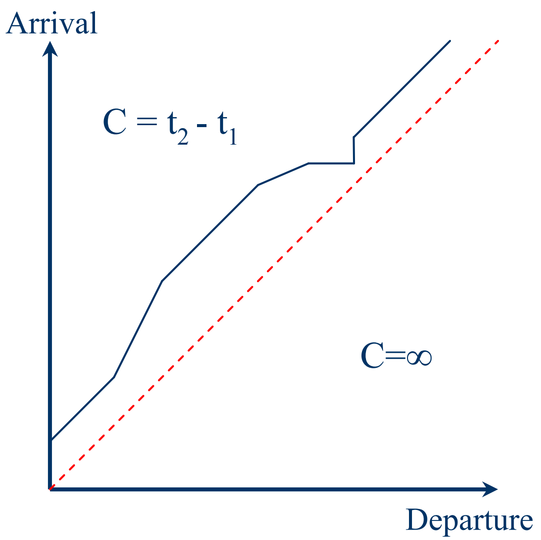

- Closed semiring

# Cost function form

- Functions $D^2 \rightarrow [0, \infty]$ in general are intractable
- Must choose a subspace such that
  - Functions can be represented
  - min and + are closed

- First version
  - $\tau(t) \geq t$, piecewise linear and monotonous

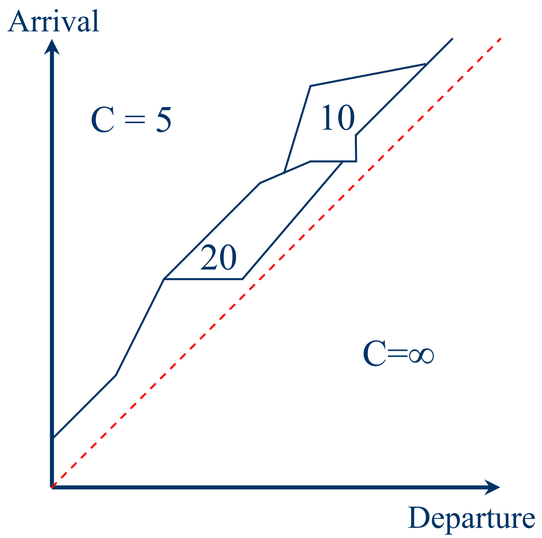
$$C(t_1, t_2) = \begin{cases} \infty & \text{if} \quad t_2 < \tau(t_1) \\ t_2 - t_1 & \text{if} \quad t_2 \geq \tau(t_1) \end{cases}$$



# Cost function form

- More advanced: several curves with different cost

- Computationally heavy
- Curves multiply
  - Eliminate dominated curves
  - Keep only fastest and cheapest
  - Approximate
- Points multiply
  - Eliminate redundant points
  - Approximate



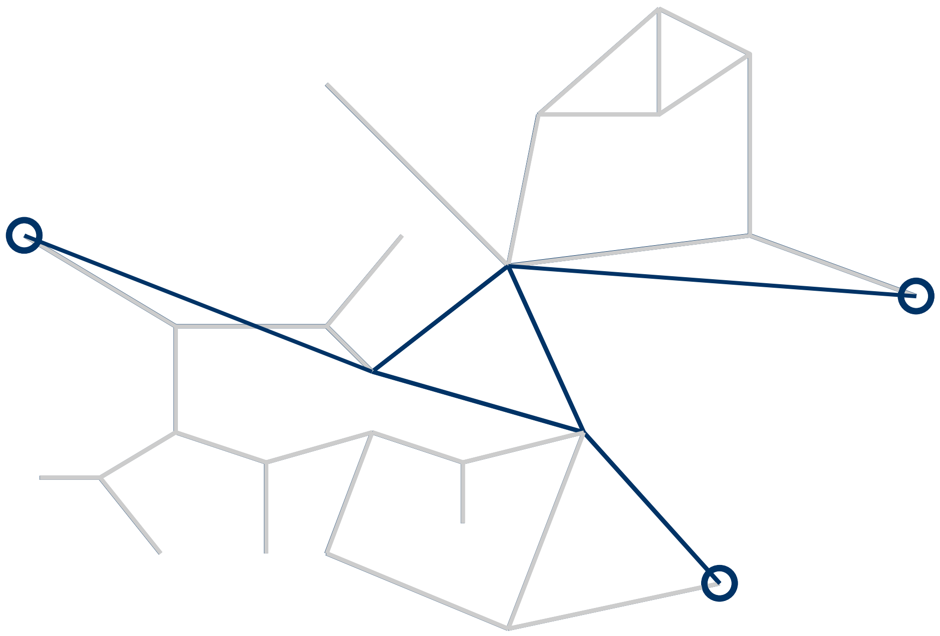
# Reducing network size

- Cost function operations are expensive
- Norwegian road network: >1M road links
- Hierarchical decomposition into subnetworks
- Only required networks are included in computation

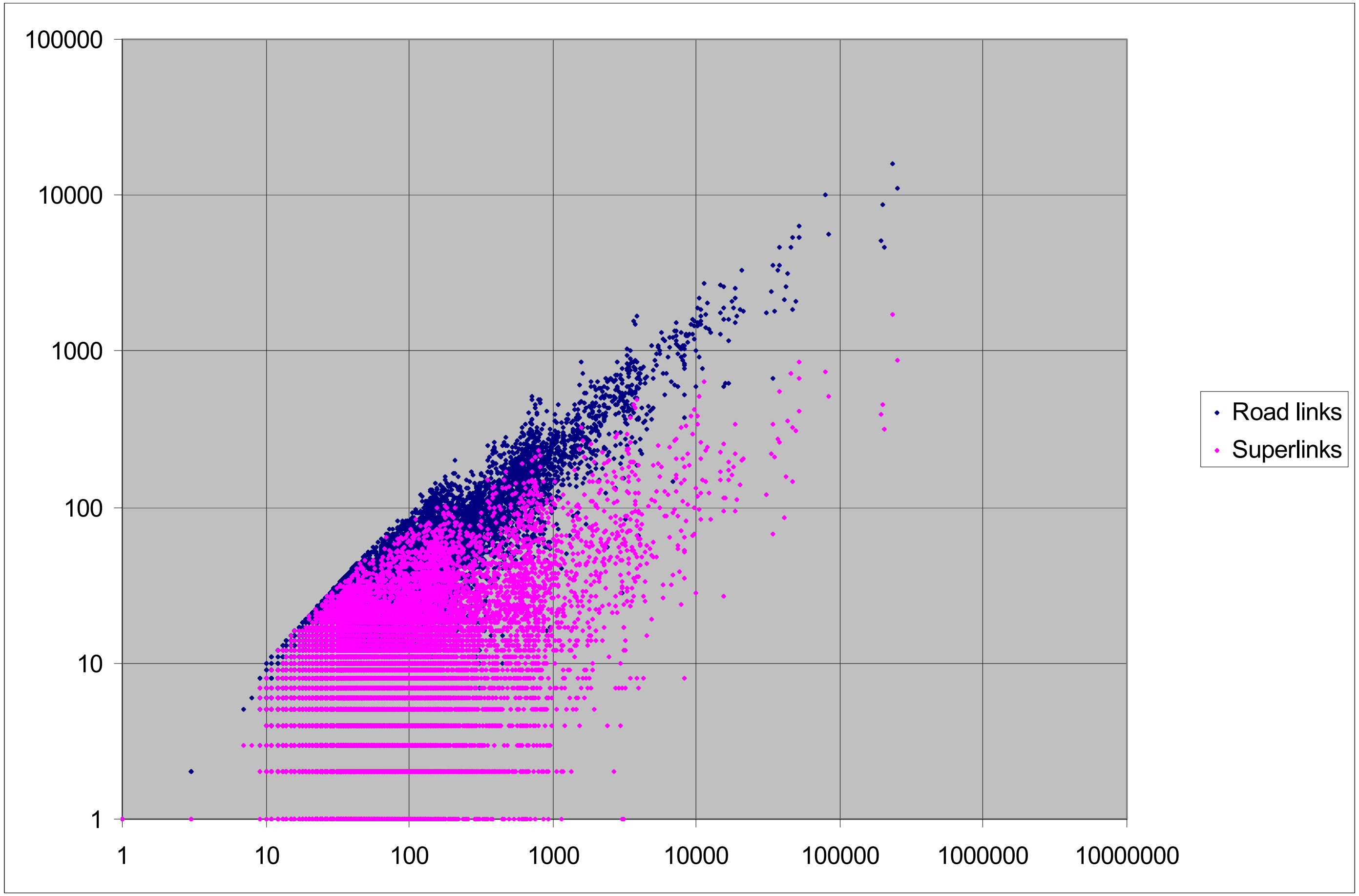
# Thoroughfares

- Network is partitioned into topologically compact regions
- If neither origin nor destination is in region, some roads *cannot* be part of optimal path
- Thoroughfare is remainder of network in region
- Computation uses full networks in origin and destination regions, and thoroughfare in other regions
- Does not affect optimal solution if done exactly
  - Tradeoff between exactness and preprocessing time

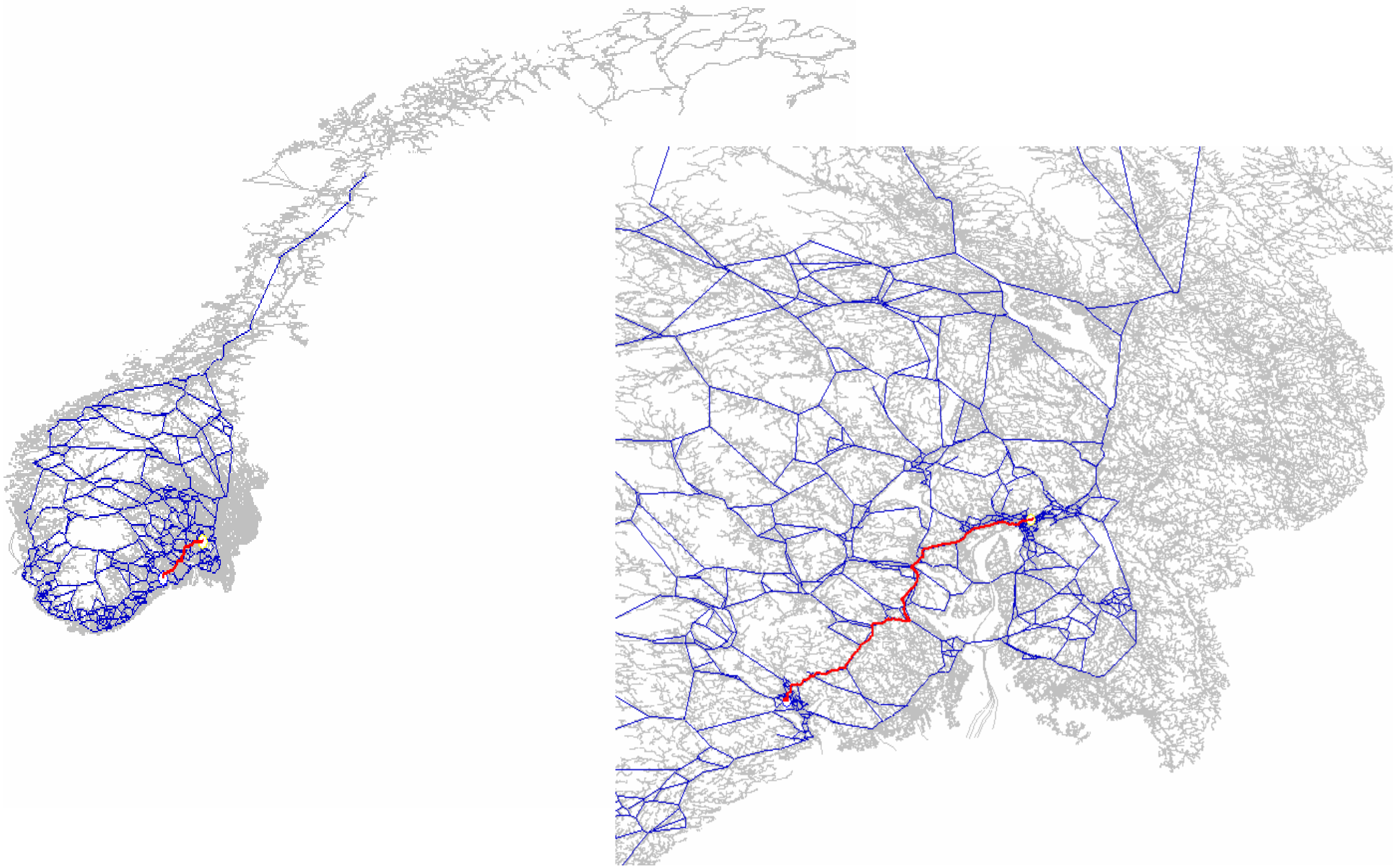
# Thoroughfare example



# Thoroughfare sizes



# Example computation network



# Architecture

Volatile, generated on demand

- Cost function cache
- Graph cache

---

Permanent on disk, loaded on demand

- Subnetwork hierarchy
- Road network database

# Demo

# Future work

- More realistic testing
  - Use real travel time data
  - Use in VRP
- Better cost function mechanics
  - Other cost function subspaces
  - Better time/space optimization