Aggregation based on road topologies for large scale VRPs

Eivind Nilssen, SINTEF ICT

Oslo, June 12-14 2008

Outline

- Motivation and background
- Aggregation
- Some results
- Conclusion

Motivation

- Companies with very large rich VRPs
  - Renovation
  - Distribution of newspapers
- Commercial software unable to handle these huge instances
  - Memory issues
  - Run time issues

Background

- *Edge* research project
  - Consortium between SINTEF, university labs, and industrial partners in Norway and Finland
  - Improve logistics efficiency, handle huge VRP instances
- *Effect* research project
  - Consortium between some of the *Edge* partners, and some new. Ongoing, continuation of some of the work in *Edge*
- *Spider* commercial software
  - Iterated Local Search, Variable Neighbourhood Search
  - Multiple time windows and capacity dimensions
  - Pickup, delivery, direct, and single visit orders
  - Bulk orders with compartment allocation
  - Work time regulations
  - In-homogenous fleet
  - Manual locking of plan subsets
  - etc.

Aggregation

- Combines a large set of original customer orders into fewer *aggregate orders*
- Search for optimal routes between the aggregates (problem size reduction)
- The aggregate order can represent e.g. a geographical cluster, being located at the center of the cluster
- For Euclidian space problems
  - K-means clustering algorithm

Aggregation

- K-means clustering not so good for real-world problems based on road networks

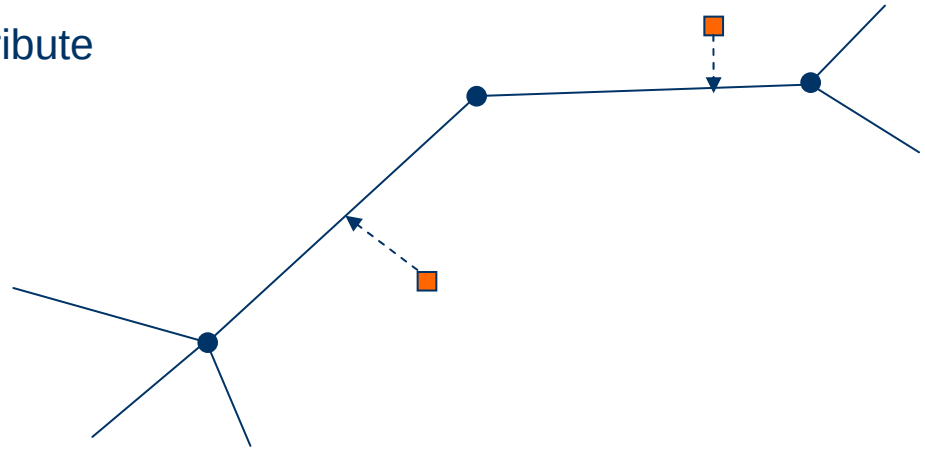
- Must use the structure imposed by the road network when constructing aggregate orders

SPIDER road networks

- Road links
- Junctions connecting 2 or more road links
- Road topology
  - Static travel times
  - Dynamic travel times
- Travellers (vehicles, bicyclists, pedestrians, …)
- Point locations
- Edge locations (reversible)
  - Reversed by search operators reversing sequences
- Cache layer (N^2 distance matrix)

Locations

- Orders typically have a pickup location and a delivery location
- Locations are snapped onto the nearest road link
  - Offroad distance
    - Travellers with different offroad speed
  - Side of road
    - Zigzag travel
      - Road attribute

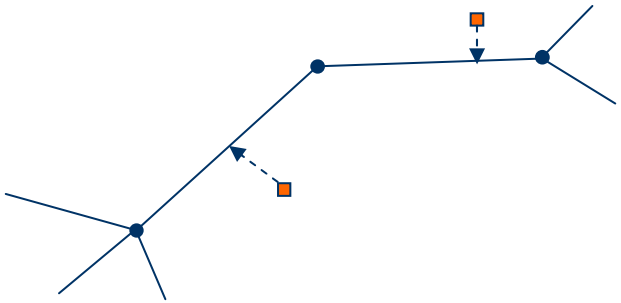


Aggregation based on road topology

- Aggregate order
  - Sequence of orders from the original problem
- Two neighbouring orders in the aggregate should have a small travel time with regard to the road topology
- Would like to read the entire problem in one go
- Must aggregate before we cache locations and calculate the distance matrix
  - N^2 is too large for the available memory

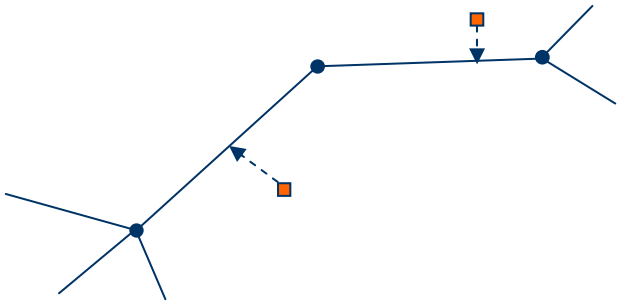
Aggregation based on road topology (2)

- Simple aggregation algorithm
  - Read orders, create locations without caching
    - still snaps locations onto the road network
  - Map orders to lists, one for each snapped road link name
  - Sort orders within the same snapped road link, based on distance to junction
  - Form aggregate orders, with sequence given by each list
  - Create reversible edge location for each aggregate order
  - Cache edge locations, calculate n^2 distance matrix



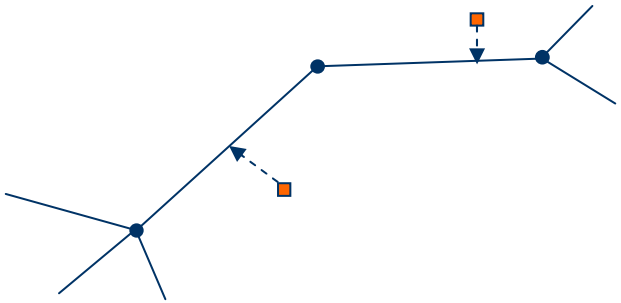
Aggregation based on road topology (3)

- Some extra processing due to road data format
  - Many consecutive junctions with only 2 road links. These road links may or may not have the same road name
  - → combine neighbouring order lists. Proceed until reaching a branching junction

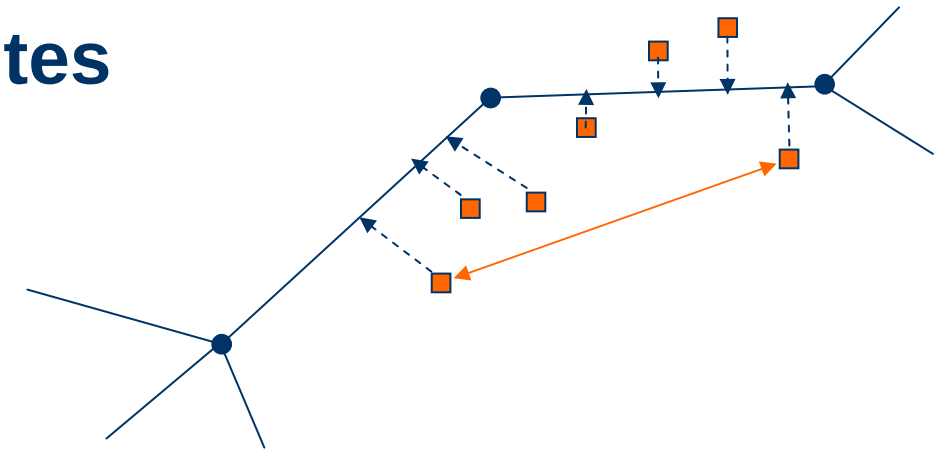


Aggregation based on road topology (4)

- Zigzag travel
  - If road allows zigzag travel: one aggregate order
  - If road prohibits zigzag travel: two aggregates – one for each side of the road
- Pickup orders, delivery orders
  - One aggregate for each type



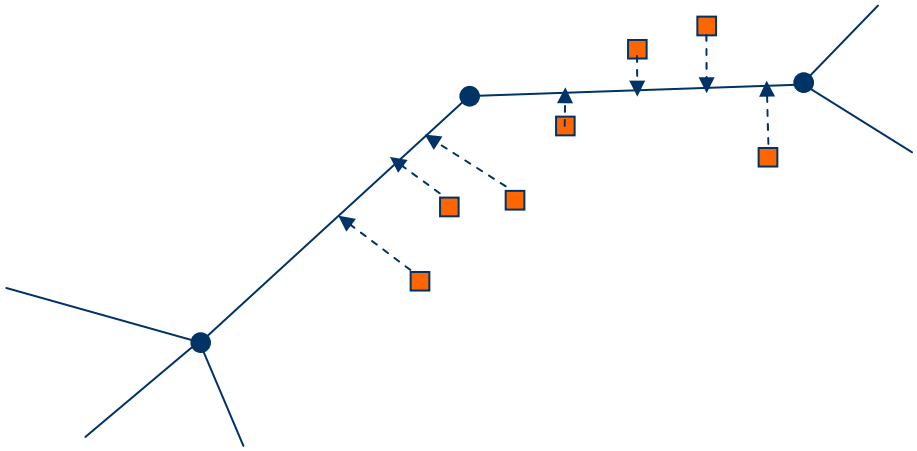
Aggregate order attributes



- Location
  - Reversible edge location from first to last order in aggregate
- Order size
  - sum of individual sizes
- Service time
  - Sum of order service times and the time spent travelling in between
    - includes offroad travel time
  - value dependent on traveller and planned edge location direction
- Cost attributes and compatibility constraints …

Aggregate order attributes (2)

- Can break aggregate formation if
  - Size exceeds aggregate threshold
  - Service time exceeds aggregate threshold
  - Original order is flagged to be excluded from aggregation

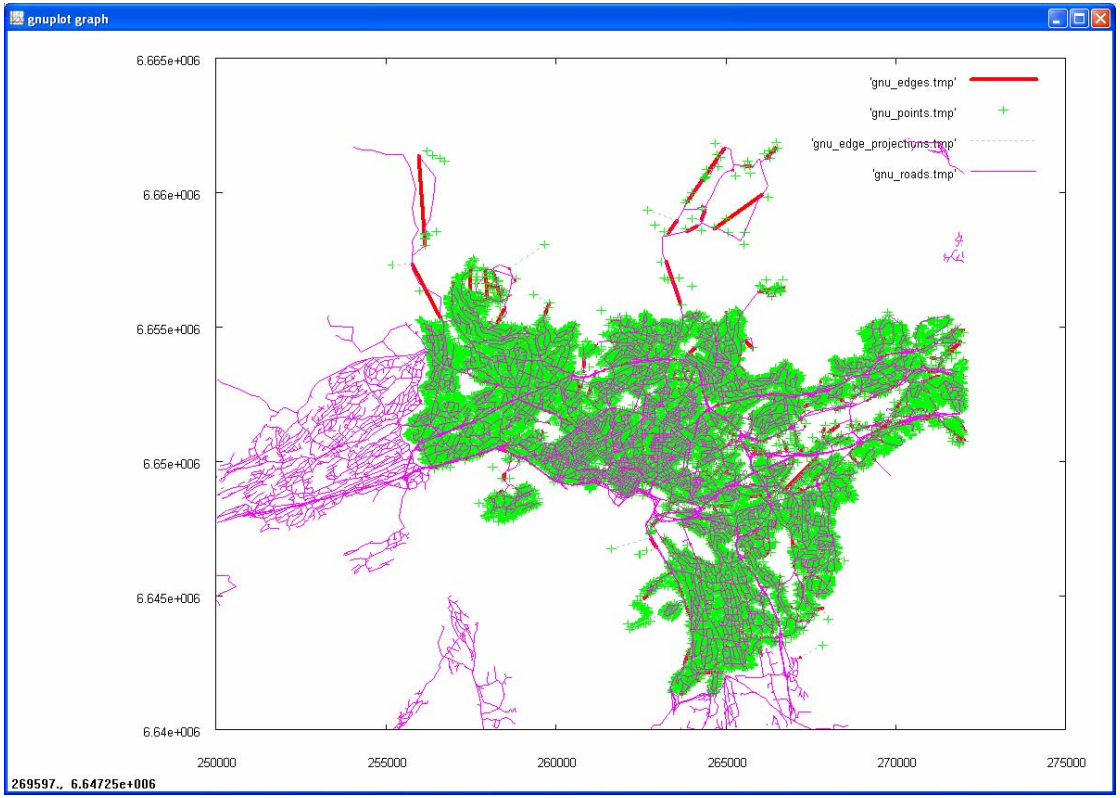


VRP solving with aggregate orders

- End user wants to see original orders only
- Formed aggregate orders are normal Spider orders
  - → Can use existing VRP solver with little extra modifications
- Formed set of aggregates is smaller in size
  - More efficient operators
  - Can have larger instance in memory
- VRP solver finds routes for aggregate orders. This is transformed into a plan for the original orders
  - For each tour in aggregate plan: create similar tour using the lists of original orders
  - Respect edge location direction when adding original orders
- Possibly combine aggregate and original orders, and edge locations plus point locations

Results

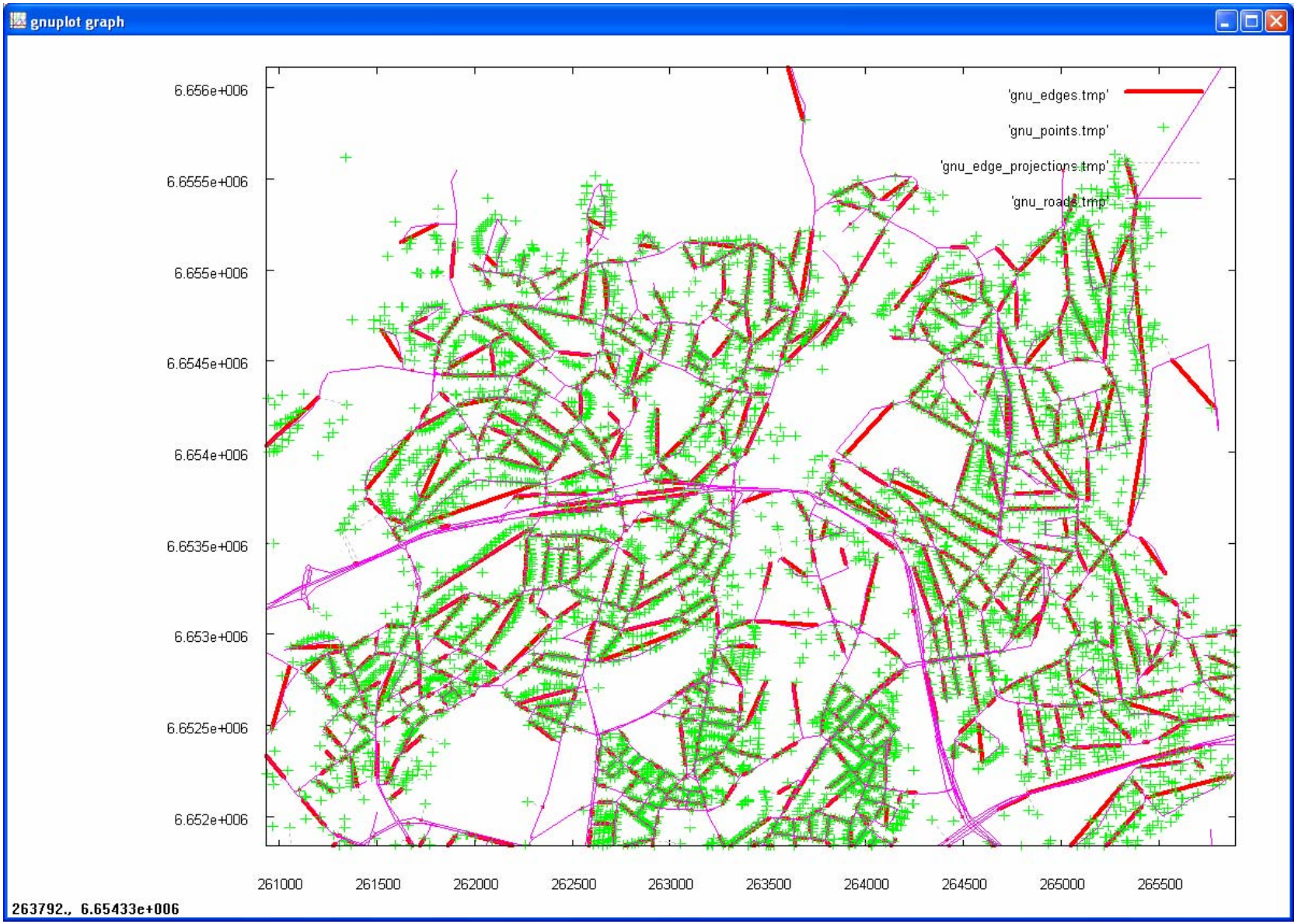
- Test case: Newspaper delivery in Oslo. >33.000 orders with unique locations (*Distribution Innovation modules*)



Results

- 33200 orders
  - 1640 seconds to read in and snap to road network
- 9300 unique aggregates after initial aggregation
- 5600 aggregate orders after further reduction
  - Aggregation takes 580 seconds
  - Reduction factor 6
- Aggregate with most orders has 181 orders

Ring 3 east


gnuplot graph
'gnu_edges.tmp'
'gnu_points.tmp'
'gnu_edge_projections.tmp'
'gnu_roads.tmp'
6.656e+006
6.6555e+006
6.655e+006
6.6545e+006
6.654e+006
6.6535e+006
6.653e+006
6.6525e+006
6.652e+006
261000
261500
262000
262500
263000
263500
264000
264500
265000
265500
263792., 6.65433e+006

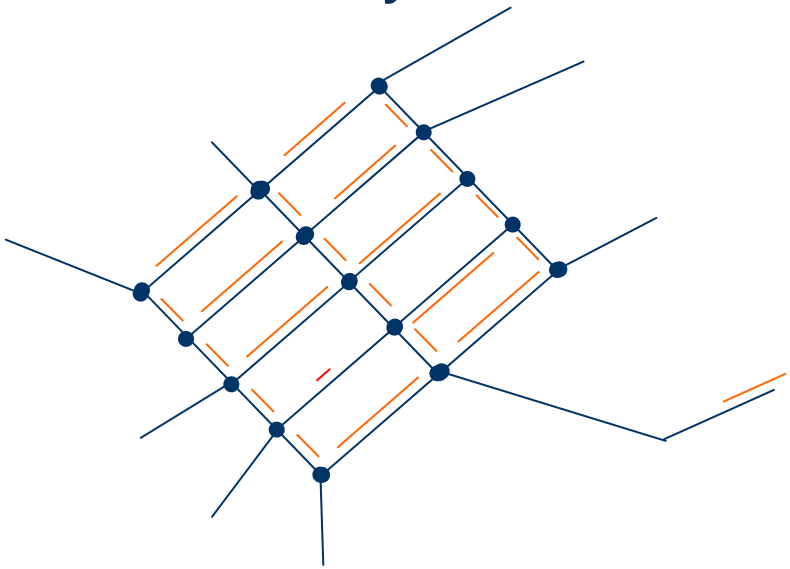
Grünerløkka

Results

- Optimization results …

Possible enhancements

- Memory reductions for city networks



- Example: 22 aggregate orders
  - 44 locations from the 22 edge locations
  - Move locations to junctions (sharing)
  - 15 locations if moved to the junctions
    → saves memory, application can handle larger case

Possible enhancements (2)

- Detect ”tree branches” in the road network, and represent area with a single aggregate order

Conclusions

- Implemented aggregation based on road topology
- Fewer locations enables us to calculate the cached travel times matrix
- Reduction factor of 6 for a real-world newspaper delivery problem
- Must test more on optimization with aggregates

Thank you for your time