Vehicle routing problems with road-network information

Dominique Feillet

Mines Saint-Étienne and LIMOS, CMP Georges Charpak, F-13541 Gardanne, France

ORBEL - Liège, February 1, 2018
Vehicle Routing Problems

Given a complete graph $G = (V, A)$ with $V = \{0, \ldots, n\}$

- 0 is a depot where is available a fleet of vehicles of capacity $Q$
- nodes $\{1, \ldots, n\}$ are customers with a delivery demand $q_i$

Given costs $c_{ij}$ on arcs

Find a set of vehicle routes that serve all customers at a minimal total cost
We call graph $G$ customer-based graph

Arcs in $G$ represent best paths in the original road-network
THE TRUCK DISPATCHING PROBLEM*

G. B. DANTZIG AND J. H. RAMSER

The paper is concerned with the optimum routing of a fleet of gasoline delivery trucks between a bulk terminal and a large number of service stations supplied by the terminal. The shortest routes between any two points in the system are given and a demand for one or several products is specified for a number of stations within the distribution system. It is desired to find a way to assign stations to trucks in such a manner that station demands are satisfied and total mileage covered by the fleet is a minimum. A procedure based on a linear programming formulation is given for obtaining a near optimal solution. The calculations may be readily performed by hand or by an automatic digital computing machine. No practical applications of the method have been made as yet. A number of trial problems have been calculated, however.

First paper published on the VRP

The first VRP

Figure: Geographical Information System (GIS)?

The Rand Mc Nally road atlas (1958)
Trend 1: A lot of papers on urban distribution (city logistics)
VRPs nowadays

Trend 2: Accurate data

- Geographic Information Systems (openStreetMap...)
- Traffic information (historical / real-time)
- Real-time monitoring
VRPs nowadays

**Trend 3:** Complex organizations / models

- Time constraints
- Multiple trips
- Multiple echelons (synchronization)
- Electric vehicles (range anxiety / recharging)
- Dynamic problem...
PART I: New issues

1. Model granularity
2. Complex attributes
3. Multiple attributes

All these issues show the limits of the customer-based graph

PART II: Methodology

1. Multigraph
2. Road-network graph
In the context of urban delivery, the distance between customers is often limited and the detail of operations (parking...) at customers becomes important.

Typical size of a parcel delivery tour

- 40 customers
- $\leq 5$ minutes per customer, including service and traveling

The classical model implicitly assumes:

- A unique and available parking location
- “Independence” of successive arcs in a tour
Model granularity

A unique and available parking location?

In practice:

- parking in cities is complex
- several parking locations are possible
- some booking systems start being developed

A unique and available parking location?
A unique and available parking location?
A unique and available parking location?
A unique and available parking location?

Diagram showing different routes:
- **Driving** pathways:
  - From A to B via a straight line.

- **Walking** pathways:
  - Possible detours or connections not explicitly shown in the diagram.
A unique and available parking location?
Model granularity

“Independence of successive arcs in a tour”?  

Parking selection implies dependence between the ingoing and the outgoing arcs

This dependence also exists when some roads are subject to fees

Some complications could arise with:

- Complex cost functions / constraints (fuel consumption minimization, congestion charges, etc.)
- Additional decisions
  - Breaks (driver working hour regulation)
  - Speed (speed optimization)

Not possible / not efficient (?) to precompute paths.
Complex cost functions / constraints

Illustration: Electric vehicle routing with stochastic energy consumption

Energy consumption $\mathcal{N}(8, 6)$

Best path?

Energy consumption $\mathcal{N}(10, 2)$
Complex attributes

Complex cost functions / constraints

Illustration: Electric vehicle routing with deterministic energy consumption depending on street segment slopes

Energy consumption 8 with peak at 14

Best path?

Energy consumption 10 with peak at 12
Additional decisions: breaks (driver working hour regulation)

15 km

<table>
<thead>
<tr>
<th>Time intervals (min)</th>
<th>Speed (km/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0, 20[</td>
<td>30</td>
</tr>
<tr>
<td>[20, 40[</td>
<td>15</td>
</tr>
<tr>
<td>[40, 60]</td>
<td>30</td>
</tr>
</tbody>
</table>

Break time: 20 minutes
Additional decisions: breaks (driver working hour regulation)

<table>
<thead>
<tr>
<th>Time intervals (min)</th>
<th>Speed (km/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0, 20[</td>
<td>30</td>
</tr>
<tr>
<td>[20, 40[</td>
<td>15</td>
</tr>
<tr>
<td>[40, 60]</td>
<td>30</td>
</tr>
</tbody>
</table>

Break time: 20 minutes

Break at customer 1:

⇒ Customer 2 reached at time 60
2 Complex attributes

Additional decisions: breaks (driver working hour regulation)

<table>
<thead>
<tr>
<th>Time intervals (min)</th>
<th>Speed (km/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0, 20]</td>
<td>30</td>
</tr>
<tr>
<td>[20, 40]</td>
<td>15</td>
</tr>
<tr>
<td>[40, 60]</td>
<td>30</td>
</tr>
</tbody>
</table>

Break time: 20 minutes

Break at customer 2:

⇒ Customer 2 reached at time 40, break finished at time 60
2 Complex attributes

Additional decisions: breaks (driver working hour regulation)

<table>
<thead>
<tr>
<th>Time intervals (min)</th>
<th>Speed (km/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0, 20]</td>
<td>30</td>
</tr>
<tr>
<td>[20, 40]</td>
<td>15</td>
</tr>
<tr>
<td>[40, 60]</td>
<td>30</td>
</tr>
</tbody>
</table>

Break time: 20 minutes

Break optimized on the road-network:

⇒ Customer 2 reached at time 50

In more complex networks, the path between customers 1 and 2 might even depend on the break time.

Also, the break time influences the previous / following parts of the route.

It is even possible that no solution exists with breaks at customers 1 or 2.
Additional decisions: Speed

It is assumed that the decision-maker can control driver’s speed to limit fuel consumption / pollution

Travel time / fuel consumption / pollution simple functions of the speed? No!

Max speed: 50 km/h  90 km/h  30 km/h
It is assumed that the decision-maker can control driver’s speed to limit fuel consumption / pollution.

Travel time / fuel consumption / pollution simple functions of the speed? No!
Additional decisions: Speed

It is assumed that the decision-maker can control driver’s speed to limit fuel consumption / pollution.

Travel time / fuel consumption / pollution simple functions of the speed? No!

Max speed: 50 km/h  90 km/h  30 km/h
Additional decisions: Speed

In addition:

- Depending on this speed different paths will be followed
- The decision-maker might modify the speed at any node in the road-network

Examples of attributes:
- Distance.
- Travel time (not necessarily strongly correlated with distance).
- Energy consumption (electric vehicle), pollution, robustness, sightseeing, danger, tolls...

The best path is not necessarily the same for each attribute!
3 Multiple attributes

(a) Min-cost graph

(b) Min-time graph

Legend
- Depot
- Customer
- Path 0→1: d = 729 & t = 2.4
- Path 0→2: d = 820 & t = 5.1
- Path 1→0: d = 801 & t = 5.6
- Path 1→2: d = 846 & t = 6.7
- Path 2→0: d = 600 & t = 4.9
- Path 2→1: d = 951 & t = 5.4

(946, 6.7)

(1590, 5.9)

(distance, time)
Some authors tried to evaluate numerically the consequences


Experiments show important increases of solution costs when using a customer-based graph, that can often exceed 10 %
Outline of the presentation

PART I: New issues

1. Model granularity
2. Complex attributes
3. Multiple attributes

All these issues show the limits of the customer-based graph

PART II: Methodology

1. Multigraph
2. Road-network graph
Outline of the presentation

Illustration with the VRP with Time Windows (VRPTW)

- Standard problem with 2 attributes: cost (distance) and time

Methodology

1. Model the road network with a multigraph.
   - One node is introduced for each customer, depot and other points of interest.
   - An arc is introduced for every efficient path between two nodes.

2. Apply directly solution methods on the road network.

In both cases, contrary to a customer-based graph, no information is lost.
Main issues

1. How to construct the multigraph? Size?
2. How to adapt exact solution schemes
   ▶ in multigraphs?
   ▶ in road-network graph?
3. Multigraph vs road-network graph?
4. How to adapt heuristic solution schemes
   ▶ in multigraphs?
   ▶ (in road-network graphs?)
Construction of the multigraph

Involve multi-objective shortest path problems: NP-hard

(a) 5437 nodes / 100 customers
(b) 19500 nodes / 100 customers

Algorithm 1 NaiveAlgorithm(s)

1: \( \mathcal{L} \leftarrow \{ (s, 0, 0) \} \) // label definition: (last vertex, distance, time)
2: repeat
3: Select \( L = (i, d, t) \in \mathcal{L} \)
4: \( \mathcal{L} \leftarrow \mathcal{L} \setminus \{L\} \)
5: for all \( j \) successor of \( i \) do
6: \( L' = (j, d + d_{ij}, t + t_{ij}) \)
7: InsertWithDominance(\( L', \mathcal{L} \))
8: // \( L_1 \prec L_2 \iff i_1 = i_2 \text{ and } t_1 \leq t_2 \text{ and } d_1 \leq d_2 \)
9: end for
10: until \( \mathcal{L} = \emptyset \)

Execute NaiveAlgorithm for each \( s \in V \)
Construction of the multigraph

Improvements:

- Implement a multi-objective multi-destination A* to guide the search:
  
  Select the label that minimizes the detour in distance among all destinations

- Stop the search once the key of the selected label is greater than the maximal detour among all destinations

Other improvements with Time Windows

- Consider only reachable customer nodes
- Only nodes that are apt to lead to a feasible path to a destination node should be considered
Construction of the multigraph

Improvements:

- Implement a multi-objective multi-destination A* to guide the search:
  
  Select the label that minimizes the detour in distance among all destinations

- Stop the search once the key of the selected label is greater than the maximal detour among all destinations

Other improvements with Time Windows

- Consider only reachable customer nodes
- Only nodes that are apt to lead to a feasible path to a destination node should be considered
Construction of the multigraph

(a) 5437 nodes / 100 customers
   Naive Algorithm: 5 seconds
   Multi-A*: 2 seconds

(b) 19500 nodes / 100 customers
   Naive Algorithm: 400 seconds
   Multi-A*: 30 seconds

≤ 4 arcs in parallel on average
Exact solution schemes
Multigraph

Branch-and-Price algorithms can easily be generalized

- **Master problem:** set partitioning problem
- **Pricing problem:** Elementary Shortest Path Problem with Resource Constraints on multigraph
  
  Solved using an adapted labelling algorithm: a label at some node is extended to all outgoing arcs
- **Branching rules:** standard branching rules

Exact solution schemes
Multigraph

Computational results: Real instances

(a) 5437 nodes / 100 customers
(b) 19500 nodes / 100 customers
## Exact solution schemes

### Multigraph

### Computational results: Real instances

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>C</th>
<th>#</th>
<th>CPU(s)</th>
<th>CPU(s)</th>
<th>CPU(s)</th>
<th>gap cost (%)</th>
<th>gap cost (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>min-cost graph</strong></td>
<td><strong>min-time graph</strong></td>
<td><strong>multigraph</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a)</td>
<td>5437</td>
<td>25</td>
<td>1</td>
<td>0.2</td>
<td>0.3</td>
<td>1.7</td>
<td>-3.4</td>
<td>-7.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.8</td>
<td>-8.0</td>
<td>-6.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>1</td>
<td>1.9</td>
<td>4.3</td>
<td>13.4</td>
<td>-2.3</td>
<td>-5.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>2.9</td>
<td>3.7</td>
<td>18.8</td>
<td>-1.6</td>
<td>-4.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>1</td>
<td>23.4</td>
<td>21.2</td>
<td>131.4</td>
<td>-0.5</td>
<td>-5.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>311.6</td>
<td>11.4</td>
<td>73.4</td>
<td>-0.7</td>
<td>-5.9</td>
<td></td>
</tr>
<tr>
<td>(b)</td>
<td>19500</td>
<td>25</td>
<td>1</td>
<td>1.5</td>
<td>0.2</td>
<td>1.2</td>
<td>-6.6</td>
<td>-10.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0.2</td>
<td>0.4</td>
<td>1.1</td>
<td>-1.7</td>
<td>-8.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>1</td>
<td>4.0</td>
<td>12.6</td>
<td>22.7</td>
<td>-0.1</td>
<td>-9.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>10.4</td>
<td>13.6</td>
<td>13.3</td>
<td>-2.3</td>
<td>-8.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>1</td>
<td>599.1</td>
<td>1372.3</td>
<td>174.1</td>
<td>-10.5</td>
<td>-4.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>55.4</td>
<td>23.6</td>
<td>102.6</td>
<td>-0.9</td>
<td>-4.4</td>
<td></td>
</tr>
</tbody>
</table>
Column generation (fractional solution)

- The master problem is not modified
- The pricing algorithm is applied in the road-network graph
  - many nodes
  - a few arcs from each node
- The service is elementary but not the routes: crossroad nodes or arcs can be traversed many times...
- When extending a label to a customer, two labels are generated: one with service, one without service

Branch-and-price (integer solution)

- A fractional solution can be supported by an integer flow
- No simple way to deal with it

However it “never” happens (unlike to what is happening in arc routing problems)
Exact solution schemes
Road network

Branch-and-price (integer solution)

- A fractional solution can be supported by an integer flow
- No simple way to deal with it

However it “never” happens (unlike to what is happening in arc routing problems)
Exact solution schemes
Road network

Branch-and-price:

- When the flow is fractional: branch on arc flow
- When the flow is integer:
  - **branch 1:** enumerate all the feasible routes in the subgraph supported by the flow and solve by IP
  - **branch 2:** impose to use an arc not in the subgraph supported by the flow

\[
\sum_{(i,j) \in A \setminus \tilde{A}} \sum_{r \in \Omega} b_{ijr} x_r \geq 1
\]

Multigraph versus road-network Experiments

| $|V_{RN}|$ | $|A_{RN}|$ | $|C|$ | $CPU_{MG}$ | $CPU_{RN}$ | $CPU_{RN} / CPU_{MG}$ |
|---|---|---|---|---|---|
| 5437 | 10181 | 5 | 1 | 0.0 | 5.5 | 117 |
| | | | 2 | 0.1 | 4.2 | 64 |
| | | | 3 | 0.0 | 4.3 | 95 |
| | | | 4 | 0.1 | 8.8 | 149 |
| | | | 5 | 0.0 | 14.8 | 330 |
| 10 | | 1 | 0.1 | 11.9 | 128 |
| | | | 2 | 0.1 | 7.1 | 90 |
| | | | 3 | 0.1 | 6.0 | 87 |
| | | | 4 | 0.1 | 11.6 | 135 |
| | | | 5 | 0.1 | 25.6 | 337 |
| 25 | | 1 | 0.2 | 56.9 | 283 |
| | | | 2 | 0.2 | 51.3 | 252 |
| | | | 3 | 0.2 | 35.1 | 213 |
| | | | 4 | 0.4 | 111.8 | 285 |
| | | | 5 | 0.2 | 81.0 | 463 |
| 50 | | 1 | 1.0 | 113.5 | 114 |
| | | | 2 | 3.3 | - | - |
| | | | 3 | 2.1 | 147.6 | 70 |
| | | | 4 | 1.0 | 252.0 | 244 |
| | | | 5 | 17.4 | - | - |

---: instances not solved in 7200 seconds
Local search operations (e.g., an insertion, a removal) imply reoptimizing the selection of arcs.
Arc selection is NP-hard

It can be managed by dynamic programming


Incremental techniques can be implemented to accelerate the method

Initially the best arcs are selected via dynamic programming (backward + forward)

Labels are stored

When a move is applied, these labels are used
Heuristic solution schemes
Multigraph

1. Initially the best arcs are selected via dynamic programming (backward + forward)
2. Labels are stored
3. When a move is applied, these labels are used
1. Initially the best arcs are selected via dynamic programming (backward + forward)
2. Labels are stored
3. When a move is applied, these labels are used
Heuristic solution schemes
Multigraph

1. Initially the best arcs are selected via dynamic programming (backward + forward)
2. Labels are stored
3. When a move is applied, these labels are used
Conclusions

Recent VRPs often involve

- Urban distribution
- Accurate data
- Complex organization / models

Customer-based graphs often fail modeling these VRPs with accuracy because of

- Model precision (granularity)
- Complex attributes
- Multiple attributes
Conclusions

Recent VRPs often involve

- Urban distribution
- Accurate data
- Complex organization / models

Customer-based graphs often fail modeling these VRPs with accuracy because of

- Model precision (granularity)
- Complex attributes
- Multiple attributes
Conclusions

The number of papers investigating these issues is very limited... ...even if it has grown a lot recently!

Replacing the customer-based graph with a multigraph seems efficient, but is not always possible (or easy).

Replacing the customer-based graph with a road-network graph is not tractable yet.

Still a lot to do!

Conclusions

The number of papers investigating these issues is very limited... ...even if it has grown a lot recently!

Replacing the customer-based graph with a multigraph seems efficient, but is not always possible (or easy).

Replacing the customer-based graph with a road-network graph is not tractable yet.

Still a lot to do!

Conclusions

The number of papers investigating these issues is very limited... ...even if it has grown a lot recently!

Replacing the customer-based graph with a multigraph seems efficient, but is not always possible (or easy).

Replacing the customer-based graph with a road-network graph is not tractable yet.

Still a lot to do!

Conclusions

The number of papers investigating these issues is very limited... ...even if it has grown a lot recently!

Replacing the customer-based graph with a multigraph seems efficient, but is not always possible (or easy).

Replacing the customer-based graph with a road-network graph is not tractable yet.

Still a lot to do!

Vehicle routing problems with road-network information

Dominique Feillet
Mines Saint-Étienne and LIMOS,
CMP Georges Charpak,
F-13541 Gardanne, France

ORBEL - Liège, February 1, 2018