The Vehicle Routing Problem with Discrete Split Delivery and Time Windows

Ilaria Vacca & Matteo Salani

Transport and Mobility Laboratory, EPFL, Switzerland

9th Swiss Transport Research Conference (STRC)

September 11, 2009
Outline

- Problem description and applications
- MILP formulation and Column Generation approach
- Branch & Price algorithm
- Computational experiments
- Conclusion
The Vehicle Routing Problem (VRP)

Given

- set of customers with demands to be served (within time windows: VRPTW);
- set of capacitated vehicles available at a single depot;
The Vehicle Routing Problem (VRP)

Objective

- design the optimal minimum-cost routes for vehicles,
- such that every customer is visited exactly once.
VRP with Split Delivery (SDVRP)

Splittable demand

- demand can be split and thus served by more than one vehicle;
- customers can be visited more than once.
VRP with Discrete Split Delivery (DSDVRP)

Nakao & Nagamochi (2007); Ceselli, Righini & Salani (2009)

- variant of VRP with split delivery;
- demand is discretized;
- the demand of each customer is represented by a set of items;
- items are grouped and delivered in orders (combinations of items);
- demand can be split but items cannot;
- service times depend on the quantity delivered;
- some combinations of items are not allowed.
Field Technician Scheduling Problem

Xu & Chiu (2001)

Problem description

- different types of jobs which require different skills;
- each technician is specialized in a field with certain skills;
- time windows on job starting and completion;
- assignment problem (jobs to technicians) + scheduling problem, where the duration of a job depends on the assignment.

Objective

- maximize the number of jobs completed within a time frame.
Problem description

- Tactical Berth Allocation Plan (TBAP): assignment and scheduling of ships to berths;
- Quay-Cranes (QC) assignment: a QC profile (number of QCs per shift) is assigned to each ship;
- feasible profiles can vary in length (number of shifts dedicated to the ship) and in size (number of QCs dedicated to the ship in each active shift);
- time windows on ship arrival and on berth availabilities.

Objective

- maximize the value of chosen profiles.
To each subset is associated:
- a demand fraction
- a service time
Modeling the Discrete Split Delivery VRPTW

- \( G = (V, E) \) complete graph with \( V = \{0\} \cup N \);
- \((c_{ij}, t_{ij})\) : cost and travel time of arc \((i, j) \in E\);
- \( N \) : set of customers \( \{1, ..., n\} \); node \( \{0\} \) represents the depot;
- \( K \) : set of vehicles with identical capacity \( Q \);
- \( R \) : set of items; \( R = \bigcup_{i \in N} R_i \), \( R_i \cap R_j = \emptyset \ \forall i \neq j, \ i, j \in N \);
- \( C \) : set of combinations of items; \( C = \bigcup_{i \in N} C_i \), \( C_i \cap C_j = \emptyset \ \forall i \neq j, \ i, j \in N \);
- \( e_{cr}^r \) : 1 if item \( r \in R \) belongs to combination \( c \in C \);
- \( t_c \) : service time of combination \( c \in C \) such that \( \max_{r \in R} e_{cr}^r t_r \leq t_c \leq \sum_{r \in R} e_{cr}^r t_r \);
- \( q_c \) : size of combination \( c \in C \); \( q_c = \sum_{r \in R} e_{cr}^r t_r \);
- \([a_i, b_i]\) : time window for customer \( i \in N \).
Modeling the Discrete Split Delivery VRPTW

Decision variables

- \( x_{ij}^k \) binary: 1 if arc \((i, j) \in E\) is used by vehicle \( k \in K \), 0 otherwise;
- \( y_c^k \) binary: 1 if vehicle \( k \in K \) delivers combination \( c \in C \), 0 otherwise;
- \( T_{ki}^k \geq 0 \) : time when vehicle \( k \in K \) arrives at customer \( i \in N \).

Objective function

- minimize the total traveling costs:
  \[ z^* = \min \sum_{k \in K} \sum_{(i,j) \in E} c_{ij} x_{ij}^k \]

Constraints

- flow and precedence constraints;
- demand-satisfaction constraints;
- time-windows constraints;
- capacity constraints.
Modeling the Discrete Split Delivery VRPTW

Flow and linking constraints

\[
\sum_{j \in V} x_{0j}^k = 1 \quad \forall k \in K, (1)
\]

\[
\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{ji}^k = 0 \quad \forall k \in K, \forall i \in V, (2)
\]

\[
\sum_{j \in V} x_{ij}^k = \sum_{c \in C_i} y_c^k \quad \forall k \in K, \forall i \in N, (3)
\]

Covering constraints

\[
\sum_{k \in K} \sum_{c \in C} e_c^r y_c^k = 1 \quad \forall r \in R, (4)
\]

\[
\sum_{c \in C_i} y_c^k \leq 1 \quad \forall k \in K, \forall i \in N, (5)
\]
Modeling the Discrete Split Delivery VRPTW

**Precedence constraints**

\[ T^k_i + \sum_{c \in C_i} t_c y^k_c + t_{ij} - T^k_j \leq (1 - x^k_{ij})M \quad \forall k \in K, \forall i \in N, \forall j \in V, \]  

(6)

\[ T^k_i - t_{0i} \geq (1 - x^k_{0i})M \quad \forall k \in K, \forall i \in N, \]  

(7)

**Time windows**

\[ T^k_i \geq a_i \sum_{j \in V} x^k_{ij} \quad \forall k \in K, \forall i \in N, \]  

(8)

\[ T^k_i + \sum_{c \in C_i} t_c y^k_c \leq b_i \sum_{j \in V} x^k_{ij} \quad \forall k \in K, \forall i \in N, \]  

(9)

**Capacity constraints**

\[ \sum_{c \in C} q_c y^k_c \leq Q \quad \forall k \in K. \]  

(10)
Dantzig-Wolfe reformulation

- $P$: set of feasible routes;
- $c_p$: cost of route $p \in P$;
- $e^r_p$: binary parameter equal to 1 if item $r \in R$ is delivered in route $p \in P$;
- $\lambda_p$: binary decision variable equal to 1 if route $p \in P$ is chosen;

**Master problem**

\[
\begin{align*}
\text{min} & \quad \sum_{p \in P} c_p \lambda_p \\
\sum_{p \in P} e^r_p \lambda_p &= 1 \quad \forall r \in R, \\
\sum_{p \in P} \lambda_p &\leq |K|, \\
\lambda_p &\geq 0 \quad \forall p \in P.
\end{align*}
\]
Column generation scheme

Pricing subproblem

\[ p^* = \arg\min_{p \in P} \{ \tilde{c}_p \} = \arg\min_{p \in P} \{ c_p - \sum_{r \in R} \pi_r e^r_p - \pi_0 \} \tag{15} \]

where \( \pi_r \) are the dual variables associated to constraints (12) and \( \pi_0 \) is the dual variable associated to constraint (13).

Column generation

- if \( \tilde{c}_{p^*} < 0 \), then column \( p^* \) is added to the (restricted) master problem;
- otherwise, the current master problem solution is proven to be optimal.
Column generation scheme

Remarks

- the pricing subproblem is an Elementary Resource Constrained Shortest Path Problem (ERCSPP);
- the underlying network has one node for each order and transit time \((t_{ij} + t_c)\);
  Ceselli et al. 2009: one node for each item.
Branch & Price for the DSDVRPTW

- column generation approach;
- exact algorithm based on Branch&Price;
- pricing solved using bi-directional dynamic programming (Righini & Salani, 2006);
- branching rules:
  1. total number of vehicles;
  2. number of vehicles visiting a customer;
  3. flow on arcs.
- no additional cuts at master level.
Instances

- derived from Solomon's data sets R1, C1 and RC1 for the VRPTW;
- \( N = 25, 50 \) customers;
- \( Q = 30, 50, 100 \);
- the demand of each customer is discretized in 12 items;
- we generated 3 scenarios:

<table>
<thead>
<tr>
<th>scenario</th>
<th>orders</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3</td>
<td>full, 50-50%</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>full, 50-50%, 75-25%</td>
</tr>
<tr>
<td>C</td>
<td>7</td>
<td>full, 50-50%, 75-25%, 90-10%</td>
</tr>
<tr>
<td>O</td>
<td>1</td>
<td>full order (unsplitted case)</td>
</tr>
</tbody>
</table>
### Computational results: 25 customers

<table>
<thead>
<tr>
<th>class</th>
<th>nb_inst</th>
<th>Q</th>
<th>nb_solved</th>
<th>t</th>
<th>nb_solved</th>
<th>t</th>
<th>nb_solved</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td>B</td>
<td></td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>12</td>
<td>30</td>
<td>12</td>
<td>87</td>
<td>10</td>
<td>694</td>
<td>6</td>
<td>1554</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td></td>
<td>11</td>
<td>342</td>
<td>6</td>
<td>463</td>
<td>5</td>
<td>522</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td></td>
<td>9</td>
<td>16</td>
<td>10</td>
<td>129</td>
<td>9</td>
<td>551</td>
</tr>
<tr>
<td>C1</td>
<td>9</td>
<td>50</td>
<td>9</td>
<td>273</td>
<td>0</td>
<td>x</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td></td>
<td>3</td>
<td>947</td>
<td>0</td>
<td>x</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>RC1</td>
<td>8</td>
<td>30</td>
<td>8</td>
<td>317</td>
<td>0</td>
<td>x</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td></td>
<td>8</td>
<td>222</td>
<td>2</td>
<td>1542</td>
<td>0</td>
<td>x</td>
</tr>
</tbody>
</table>
Computational results: 50 customers

<table>
<thead>
<tr>
<th>Class</th>
<th>nb_inst</th>
<th>$Q$</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>nb_solved</td>
<td>t</td>
<td>nb_solved</td>
</tr>
<tr>
<td>R1</td>
<td>12</td>
<td>30</td>
<td>1</td>
<td>3011</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>1</td>
<td>1527</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>2</td>
<td>120</td>
<td>2</td>
</tr>
<tr>
<td>RC1</td>
<td>8</td>
<td>50</td>
<td>7</td>
<td>723</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>1</td>
<td>1953</td>
<td>0</td>
</tr>
</tbody>
</table>
Summing up

- time limit: 1 hour;

- 69% (A), 32% (B) and 23% (C) instances solved for 25 customers;

- 14% (A), 2% (B) and 1% (C) instances solved for 50 customers;

- difficulty increases with the number of customers and with the number of orders;

- split deliveries are more frequent with small values of $Q$;

- in some cases, split deliveries not only decrease the total traveling costs but also allow to save one vehicle.
Conclusions

- finding optimal solutions is difficult, already with a small number of orders;
- only a limited number of instances with 50 customers could be solved;
- the bottleneck is the pricing problem:
  - the underlying network is huge (one node per each order!)
  - how to efficiently handle this feature of the problem?
Thanks for your attention!