The Vehicle Routing Problem with Discrete Split Deliveries and Time Windows

Ilaria Vacca

Transport and Mobility Laboratory, EPFL

joint work with Matteo Salani

Séminaire du 3ème cycle romand de Recherche Opérationnelle

Zinal - January 18, 2010

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

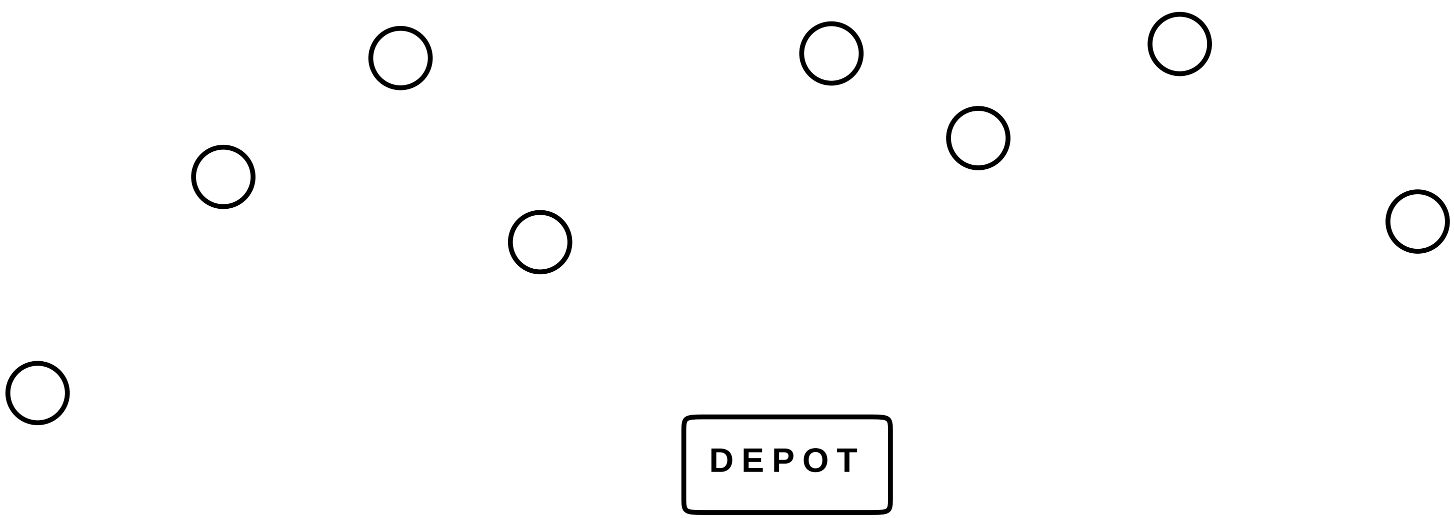
Outline

- Problem description and applications
- MILP formulation and Column generation
- Branch-and-price algorithm
- Computational results
- Conclusions and further research

The Vehicle Routing Problem (VRP)

Given

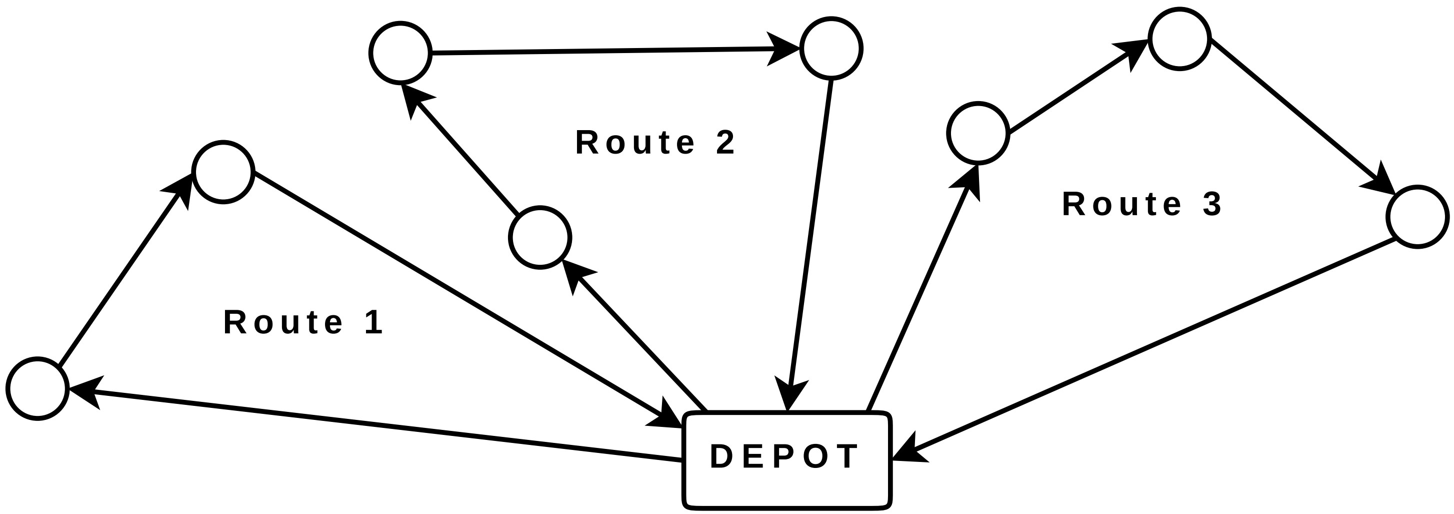
- set of customers with demands to be served (within time windows: VRPTW);
- set of capacitated vehicles available at a single depot;



The Vehicle Routing Problem (VRP)

Objective

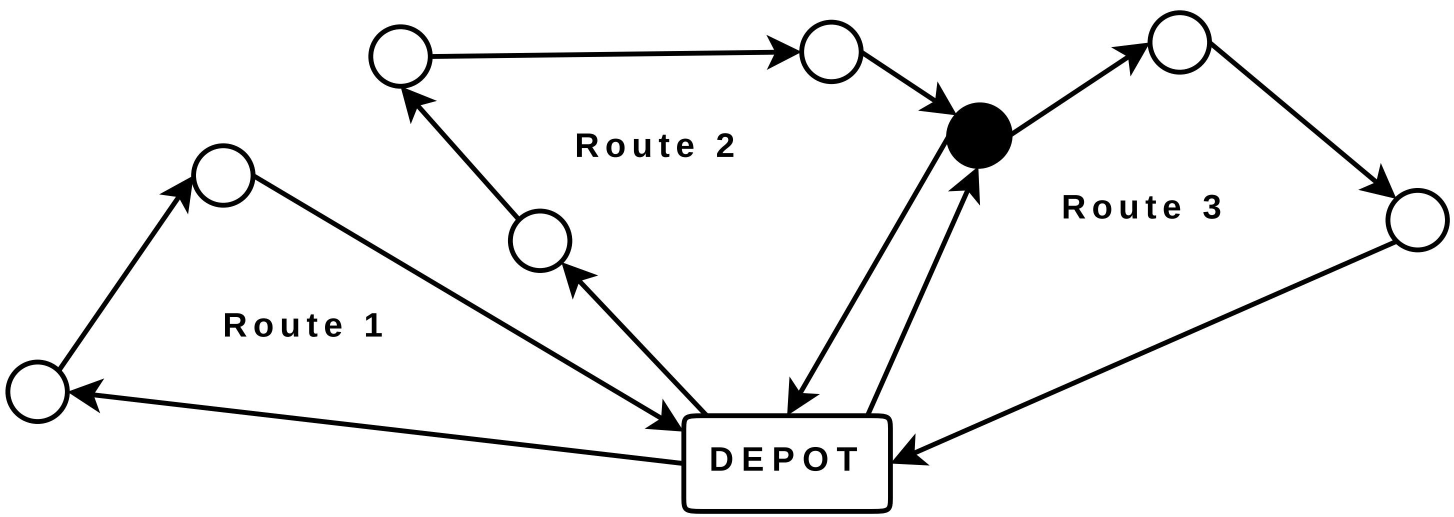
- design the optimal minimum-cost routes for vehicles,
- such that every customer is visited exactly once.



VRP with Split Deliveries (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 Deliveries (DSDVRP)

Demand is discretized in items

- variant of VRP with (continuous) split deliveries;
- the demand of each customer is represented by a set of items;
- demand can be split but items cannot.

Demand is delivered in combinations of items

- items are grouped and delivered in combinations (subsets of items);
- for every customer, feasible combinations of items are predefined and known;
- some combinations of items are not allowed.

Sierksma & Tijssen (1998); Nakao & Nagamochi (2007); Ceselli et al. (2009).

DSDVRP with Time Windows (DSDVRPTW)

Service time is quantity-dependent

- we define *service time* the time needed to perform the delivery once the vehicle is arrived at customer's location;
- in our model, service time depends on the quantity delivered;
- in particular, a specific service time is associated to every feasible combination of items;
- we relax the usual assumption of constant service times.

Known properties

Property 1. The SDVRPTW is NP-Hard.

- Property 1 holds for DSDVRPTW.

Property 2. There exists an optimal solution of the SDVRPTW in which no two routes have more than one split demand in common.

- Property 2 doesn't hold for DSDVRPTW (because of discrete demand and quantity-dependant service times).

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.

TBAP with QC assignment in container terminals

Giallombardo, Moccia, Salani & Vacca (2010)

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.

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_c^r : 1 if item $r \in R$ belongs to combination $c \in C$;
- t_c : service time of combination $c \in C$ such that $\max_{r \in R} e_c^r t^r \leq t_c \leq \sum_{r \in R} e_c^r t_r$;
- q_c : size of combination $c \in C$; $q_c = \sum_{r \in R} e_c^r t^r$;
- $[a_i, b_i]$: time window for customer $i \in N$.

Arc-flow formulation

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;
- $T_i^k \geq 0$: 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.

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Arc-flow formulation

Flow and linking constraints

$$\sum_{j \in V} x_{0j}^k = 1 \qquad \forall k \in K, \tag{1}$$

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{ji}^k = 0 \qquad \forall k \in K, \forall i \in V, \tag{2}$$

$$\sum_{j \in V} x_{ij}^k = \sum_{c \in C_i} y_c^k \qquad \forall k \in K, \forall i \in N, \tag{3}$$

Covering constraints

$$\sum_{k \in K} \sum_{c \in C} e_c^r y_c^k = 1 \qquad \forall r \in R, \tag{4}$$

$$\sum_{c \in C_i} y_c^k \leq 1 \qquad \forall k \in K, \forall i \in N, \tag{5}$$

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Arc-flow formulation

Precedence constraints

$$T_i^k + \sum_{c \in C_i} t_c y_c^k + t_{ij} - T_j^k \leq (1 - x_{ij}^k)M \qquad \forall k \in K, \forall i \in N, \forall j \in V, \tag{6}$$

$$T_i^k - t_{0i} \geq (1 - x_{0i}^k)M \qquad \forall k \in K, \forall i \in N, \tag{7}$$

Time windows

$$T_i^k \geq a_i \sum_{j \in V} x_{ij}^k \qquad \forall k \in K, \forall i \in N, \tag{8}$$

$$T_i^k + \sum_{c \in C_i} t_c y_c^k \leq b_i \sum_{j \in V} x_{ij}^k \qquad \forall k \in K, \forall i \in N, \tag{9}$$

Capacity constraints

$$\sum_{c \in C} q_c y_c^k \leq Q \qquad \forall k \in K. \tag{10}$$

Dantzig-Wolfe reformulation

- P : set of feasible routes (constraints (1)-(3) + (5)-(10));
- c_p : cost of route $p \in P$;
- e_p^r : binary parameter equal to 1 if item $r \in R$ is delivered in route $p \in P$;
- λ_p : binary decision variable equal to 1 if route $p \in P$ is chosen;

Master problem (path formulation)

$$\min \sum_{p \in P} c_p \lambda_p \tag{11}$$

$$\sum_{p \in P} e_p^r \lambda_p = 1 \qquad \forall r \in R, \tag{12}$$

$$\sum_{p \in P} \lambda_p \leq |K|, \tag{13}$$

$$\lambda_p \geq 0 \qquad \forall p \in P. \tag{14}$$

Column generation scheme

The so-called **Restricted Master Problem** (RMP) is repeatedly solved on a subset of variables λ, which otherwise would be an exponential number.

At each iteration of column generation we add profitable variables not yet in the formulation by solving the *pricing subproblem*.

Pricing subproblem

Find the route (column) p with the minimum reduced cost $\tilde{c}_p$:

$$p^* = \arg\min_{p \in P}\{\tilde{c}_p\} = \arg\min_{p \in P}\{c_p - \sum_{r \in R} \pi_r e_p^r - \pi_0\} \tag{15}$$

where π_r are the dual variables associated to constraints (12) and π_0 is the dual variable associated to constraint (13).

Column generation scheme

Generation of columns

- solve the pricing subproblem (identify the min-red-cost column);
- 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.

Remarks

- the pricing subproblem is an Elementary Resource Constrained Shortest Path Problem solved via dynamic programming;
- the underlying network has one node for each combination and transit time equal to $(t_{ij} + t_c)$.

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Branch & Price for the DSDVRPTW

- exact algorithm based on branch-and-bound;
- column generation at each node of the search tree;
- pricing solved using bi-directional dynamic programming;
- branching rules:
  1. total number of vehicles;
  2. number of vehicles visiting a customer;
  3. flow on one arc;
  4. flow on two consecutive arcs.
- cutting: 2-path inequalities at the root node.

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

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:

scenario	combinations	description
A	3	full, 50-50%;
B	5	full, 50-50%, 75-25%;
C	7	full, 50-50%, 75-25%, 90-10%;
O	1	full (unsplit case).

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Computational results: $n = 25$ customers

			A		B		C	
class	nb_inst	Q	nb_solved	t	nb_solved	t	nb_solved	t
R1	12	30	12	7	12	75	8	466
		50	12	6	12	60	12	430
		100	12	9	12	41	12	113
C1	9	30	4	1108	0	x	0	x
		50	9	37	4	2137	0	x
		100	7	706	4	705	2	1876
RC1	8	30	2	1988	0	x	0	x
		100	8	3	8	11	8	35

Time limit: 1 hour.

Computational results: $n = 50$ customers

			A		B		C	
class	nb_inst	Q	nb_solved	t	nb_solved	t	nb_solved	t
R1	12	30	1	1010	0	x	0	x
		50	3	1572	1	385	0	x
		100	3	1035	2	167	2	535
RC1	8	50	7	54	6	902	0	x
		100	8	529	6	809	3	2832

Time limit: 1 hour.

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Optimal solutions for class RC1, $n = 50$, $Q = 50$

id	O			A			B			C		
	z_{IP}	veh	t	z_{IP}	veh	t	z_{IP}	veh	t	z_{IP}	veh	t
rc101	1713.2	20	0	**1708.9**	20	13	**1708.3**	20	594	x		
rc102	1704.3	20	0	**1700.5**	20	62	**1700.5**	20	1938	x		
rc103	1703.4	20	1	**1696.8**	20	37	**1696.8**	20	427	x		
rc104	1702.2	20	1	**1696.7**	20	54	**1696.7**	20	677	x		
rc105	1703.9	20	0	**1700.1**	20	73	**1700.1**	20	1132	x		
rc107	1704.1	20	1	**1698.6**	20	58	x			x		
rc108	1702.2	20	2	**1696.7**	20	83	**1696.7**	20	645	x		

Summing up

- 76% (A), 60% (B) and 48% (C) instances solved for 25 customers;
- 25% (A), 17% (B) and 6% (C) instances solved for 50 customers;
- 96% (R1), 37% (C1) and 36% (RC1) instances solved for 25 customers;
- 11% (R1), 0% (C1) and 42% (RC1) instances solved for 50 customers;
- difficulty increases with the number of customers and combinations;
- 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 few combinations;
- 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 combination!)
  - how to efficiently handle this feature of the problem?

Ongoing work

New framework: two-stage column generation

- methodology to accelerate an overall B&P algorithm via generation of compact formulation variables;
- useful when the compact formulation exhibits a large number of variables;
- useful to detect sub-optimal compact formulation variables.

Application to the DSDVRPTW

- we start with a subset of combinations (corresponding to a subset of y_c variables);
- we compute a bound of the reduced cost for variables y_c not yet considered;
- we add the most profitable variable and we iterate.

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Thanks for your attention!

More details on **transp-or.epfl.ch** :

M. Salani and I. Vacca, *Branch-and-price for the Vehicle Routing Problem with Discrete Split Deliveries and Time Windows*, Technical report TRANSP-OR 091224.