An exact algorithm for the discrete split delivery VRP with time windows

Matteo Salani & Ilaria Vacca

Transport and Mobility Laboratory, EPFL, Switzerland

23rd EURO European Conference on Operational Research

Bonn, Germany

July 6, 2009

Outline

- Problem description and applications
- MILP formulation and Column Generation approach
- Branch & Price algorithm
- Computational experiments
- Conclusion

VRP with Discrete Split Delivery

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

Problem description

- variant of VRP with split delivery;
- each customer demand is represented by a set of items which are delivered by orders (combination of items);
- demand can be split (discretized) but items cannot;
- some combinations of items are not allowed because of incompatibilities between items and vehicles, items and locations, etc.

Objective

- Minimize the total travel costs.

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 (2009)

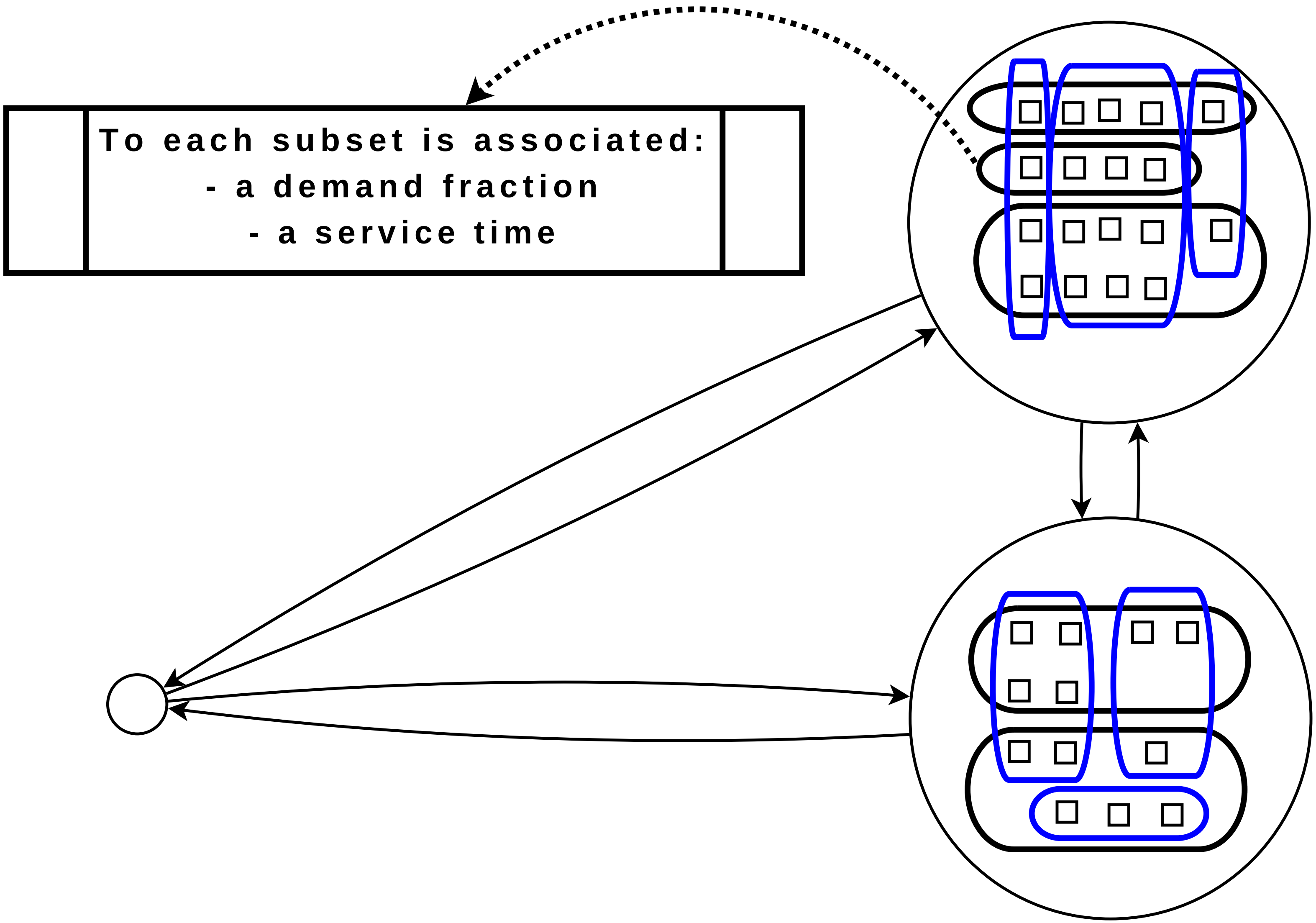
Problem description

- 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 of DSDVRPTW



VRP with Discrete Split Delivery

- $G = (V, E)$ complete graph with $V = \{0\} \cup N, \;\; (c_{ij}, t_{ij}) \;\; \forall (i,j) \in E$;
- N : set of customers $\{1, ..., n\}$;
- K : set of vehicles (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 $r \in R$ is in combination $c \in C$;
- t_c : service time of combination $c \in C$ with $t_c \leq \sum_{r \in c} t_r, \;\; t_c \geq t_r \;\; \forall r \in c$;
- q_c : size of combination $c \in C$;
- $[a_i, b_i]$: time window for customer $i \in N$.

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

VRP with Discrete Split Delivery

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

VRP with Discrete Split Delivery

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

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

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

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

VRP with Discrete Split Delivery

Precedence constraints

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

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

Time windows

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

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

Capacity constraints

$$\sum_{c \in C} q_c y_c^k \leq Q \qquad \forall k \in K, \quad (10)$$

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Column generation for DSDVRPTW

Master problem

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

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

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

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

where:

- P : set of feasible sequences;
- e_p^r : 1 if item $r \in R$ is delivered in sequence $p \in P$ and 0 otherwise;
- c_p : cost of sequence $p \in P$.

Column generation for DSDVRPTW

Pricing problem

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

Network formulation

- one node for each order (Ceselli et al. 2009: one node for each item);
- elementary resource constrained shortest path problem (ERCSPP).

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

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: vehicles first, arcs next;
- No additional cuts at master level.

Computational results

- Instances derived from Solomon R1 data set for the VRPTW;
- N = 25 customers;
- Demand of each customer is discretized as follows:

scenario	items	orders	description
split_50	12	3	full; 50% (2)
split_33	12	6	full; 50% (2); 33% (3)
split_random	10 (avg)	21 (avg)	full; single(10); random (10)
unsplit	12	1	full

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

$N = 25,\ Q = 200$ (no initialization)

instance	z^*	K^*	unsplit (sec)	split_50 (sec)	split_33 (sec)	split_random (sec)
r101	617.1	8	0.31	3.66	91.07	>1h
r102	547.1	7	1.78	140.79	>1h	>1h
r103	454.6	5	1.43	64.16	>1h	>1h
r104	416.9	4	2.71	235.54	>1h	>1h
r105	530.5	6	0.76	11.01	296.14	>1h
r106	465.4	3	4.63	455.13	>1h	>1h
r107	424.3	4	2.65	637.58	>1h	>1h
r108	397.3	4	12.35	>1h	>1h	>1h
r109	441.3	5	1.91	167.21	>1h	>1h
r110	444.1	4	5.04	636.59	>1h	>1h
r111	428.8	5	8.21	1085.29	>1h	>1h
r112	393.0	4	20.16	>1h	>1h	>1h

$N = 25, \; Q = 200$ (with initialization "unsplit")

instance	z^*	K^*	unsplit (sec)	split_50 (sec)	split_33 (sec)	split_random (sec)
r101	617.1	8	0.31	0.78	1.27	0.75
r102	547.1	7	1.78	8.40	35.65	75.22
r103	454.6	5	1.43	2.42	3.47	3.37
r104	416.9	4	2.71	9.08	16.58	21.72
r105	530.5	6	0.76	0.79	0.98	2.19
r106	465.4	3	4.63	34.54	101.15	878.44
r107	424.3	4	2.65	6.81	13.28	20.53
r108	397.3	4	12.35	1878.39	>1h	>1h
r109	441.3	5	1.91	0.82	>1h	3.65
r110	444.1	4	5.04	1104.66	>1h	132.91
r111	428.8	5	8.21	3156.03	>1h	237.18
r112	393.0	4	20.16	>1h	>1h	>1h

Computational results

- no savings in terms of vehicles and total costs;
- increased complexity, higher computational time;
- symmetry doesn't help (split_33 vs split_random);
- remark on Solomon's instances:
  - average demand per customer (13) $<<$ vehicle's capacity (200);
  - average number of customers per vehicle: 5.

- additional tests on scenarios with smaller capacities:
  - $Q =$ 2, 3 and 4 times the average customer's demand;
  - in order to make the comparison with the unsplit case: $Q \geq$ max demand.

$N = 25,\ Q = 29$ (with initialization "unsplit")

instance	UNSPLIT z^*	UNSPLIT K^*	UNSPLIT time (sec)	SPLIT_50 z	SPLIT_50 K	SPLIT_50 time (sec)	SPLIT_33 z	SPLIT_33 K	SPLIT_33 time (sec)
r101	803.6	13	0.28	803.6	13	6.33	803.6	13	1h
r102	793.5	13	0.60	**792.3**	**13**	**70.34**	793.5	13	1h
r103	766.1	12	0.56	766.1	12	9.90	766.1	12	1054.92
r104	766.1	12	0.77	766.1	12	20.29	766.1	12	1888.39
r105	788.8	13	1.04	**787.7**	**12**	**1h**	788.8	12	1h
r106	778.7	12	0.85	778.7	12	77.73	778.7	12	1h
r107	754.6	12	0.54	754.6	12	8.36	754.6	12	1334.72
r108	754.6	12	0.52	754.6	12	10.88	754.6	12	1397.00
r109	754.6	12	0.42	754.6	12	6.91	754.6	12	817.47
r110	759.2	12	0.48	759.2	12	30.86	759.2	12	845.45
r111	754.6	12	0.47	754.6	12	8.82	754.6	12	1456.56
r112	754.6	12	0.77	754.6	12	13.15	754.6	12	1832.00

$N = 25,\ Q = 39$ (with initialization "unsplit")

	UNSPLIT			SPLIT_50			SPLIT_33		
instance	z^*	K^*	time (sec)	z	K	time (sec)	z	K	time (sec)
r101	698.1	10	0.32	698.1	10	1h	698.1	10	270.38
r102	656.1	9	4.69	656.1	9	117.73	656.1	9	1h
r103	650.4	9	4.81	650.4	9	1659.35	650.4	9	1h
r104	650.4	9	4.61	**644.9**	**9**	**1h**	650.4	9	1h
r105	663.7	9	0.83	663.7	9	29.63	663.7	9	232.66
r106	648.5	9	3.55	648.5	9	596.32	648.5	9	1h
r107	645.1	9	6.77	645.1	9	1h	645.1	9	1h
r108	641.8	9	6.78	641.8	9	1h	641.8	9	1h
r109	654.2	10	5.60	**652.0**	**9**	**1h**	654.2	10	1h
r110	646.0	9	5.69	**640.2**	**9**	**1096.19**	646.0	9	1h
r111	640.8	9	5.65	**639.2**	**9**	**1357.77**	640.8	9	1h
r112	655.0	9	8.23	655.0	9	1h	655	9	1h

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

$N = 25,\ Q = 52$ (with initialization "unsplit")

instance	UNSPLIT			SPLIT_50			SPLIT_33		
	z^*	K^*	time (sec)	z	K	time (sec)	z	K	time (sec)
r101	630.7	9	0.67	630.7	9	8.81	630.7	9	6.05
r102	580.7	8	>1h	580.7	8	497.84	588.4	8	1h
r103	534.3	7	2.12	**528.4**	**7**	**139.02**	534.3	7	1h
r104	527.3	7	6.69	**521.4**	**7**	**1h**	527.3	7	1h
r105	580.5	8	2.02	580.5	8	43.96	580.5	8	795.47
r106	539.5	7	3.46	539.5	7	344.35	539.5	7	1h
r107	527.7	7	4.66	527.7	7	1210.55	527.7	7	1h
r108	521.6	7	4.57	521.6	7	1h	521.6	7	1h
r109	524.6	7	2.43	524.6	7	298.61	524.6	7	1h
r110	536.7	7	4.18	**533.4**	**7**	**1005.51**	536.7	7	1h
r111	521.6	7	4.52	521.6	7	1181.08	521.6	7	1h
r112	515.8	7	2.37	515.8	7	1h	515.8	7	1h

Conclusions

- still not much saving in terms of vehicles and total costs;
- higher computational time;
- increased complexity due to the pricing problem:
  - the underlying network is huge (one node per each order);
  - how to efficiently handle this feature of the problem?
- need of further tests.

Thanks for your attention!