Branch-and-Price-and-Cut for the Split Delivery Vehicle Routing Problem with Time Windows

Guy Desaulniers

École Polytechnique de Montréal and GERAD

Column Generation 2008
Aussois, France

Outline

Outline

1. Introduction
   - Problem definition
   - Literature review
   - Main difficulty with branch-and-price for SDVRPTW
2. Formulation
   - Master problem
   - Subproblem
3. Branch-and-price-and-cut method
   - Column generation
   - Cutting and branching
4. Computational results
5. Conclusions

Vehicle Routing Problem with Time Windows (VRPTW)

- Given
  - unlimited number of identical vehicles with a given capacity, housed in a single depot
  - set of customers with known demands
  - a time window for each customer
- Find vehicle routes such that
  - all customer demands are met
  - each customer is visited by a single vehicle
  - each route starts and ends at the depot
  - each route satisfies vehicle capacity and time windows
  - total distance is minimized

Vehicle Routing Problem with Time Windows (VRPTW)

- Given
  - unlimited number of identical vehicles with a given capacity, housed in a single depot
  - set of customers with known demands
  - a time window for each customer
- Find vehicle routes such that
  - all customer demands are met
  - each customer is visited by a single vehicle
  - each route starts and ends at the depot
  - each route satisfies vehicle capacity and time windows
  - total distance is minimized

The Split Delivery VRPTW (SDVRPTW)

Same as the VRPTW except

- several vehicles can visit each customer
- demand can be split (split deliveries)
- demand can be greater than vehicle capacity

SDVRP was introduced by Dror and Trudeau (1989, 1990)

SDVRPTW was studied first by Frizzell and Giffin (1995)

Literature on the SDVRP

Not well-studied problem

- Close to 10 heuristics (2 based on branch-and-price)
- A few exact methods
  - Branch-and-cut method by Dror et al. (1994)
  - Cutting plane method by Belenguer et al. (2000)
  - Dynamic programming method by Lee et al. (2006)
  - Iterative two-stage method by Jin et al. (2007)

Literature on the SDVRPTW

Very few papers

- Construction and improvement heuristics by Frizzell and Giffin (1995) and Mullaseril et al. (1997)
- One exact branch-and-price method by Gendreau et al. (2006)

Branch-and-price

Leading methodology for the VRPTW

- Each column in the master problem corresponds to a feasible route
- Subproblem is an elementary shortest path problem with resource constraints

Main difficulty with branch-and-price for SDVRPTW

Delivered quantities are decision variables

- Branch-and-price heuristics of Sierksma and Tijssen (1998) and Jin et al. (2008)
  - For a given route, determining the delivered quantities in the subproblem is the linear relaxation of a bounded knapsack problem
  - Maximum of one split customer per route
  - All other customers receive a full delivery
  - Integrality requirements on the master problem dynamic columns (not valid for an exact method)

Main difficulty with branch-and-price for SDVRPTW

Delivered quantities are decision variables

- Branch-and-price heuristics of Sierksma and Tijssen (1998) and Jin et al. (2008)
  - For a given route, determining the delivered quantities in the subproblem is the linear relaxation of a bounded knapsack problem
  - Maximum of one split customer per route
  - All other customers receive a full delivery
  - Integrality requirements on the master problem dynamic columns (not valid for an exact method)

Main difficulty with branch-and-price for SDVRPTW

- Exact branch-and-price method of Gendreau et al. (2006)
    - Subproblem generates only routes
    - Quantities are decided in the master problem
    - Exponential numbers of quantity variables and constraints in the master problem (depend on number of generated routes)

Our contribution

New branch-and-price method

- Subproblem is an elementary shortest path problem with resource constraints, combined with the linear relaxation of a bounded knapsack problem
- Maximum of one split customer per route
- Other customers receive a zero or a full delivery
- Integrality requirements not on the master problem dynamic columns
- Convex combinations of these columns can yield routes with multiple split deliveries
- Specialized dynamic programming algorithm for the subproblem

Our contribution

New branch-and-price method

- Subproblem is an elementary shortest path problem with resource constraints, combined with the linear relaxation of a bounded knapsack problem
- Maximum of one split customer per route
- Other customers receive a zero or a full delivery
- Integrality requirements not on the master problem dynamic columns
- Convex combinations of these columns can yield routes with multiple split deliveries
- Specialized dynamic programming algorithm for the subproblem

Our contribution

New branch-and-price method

- Subproblem is an elementary shortest path problem with resource constraints, combined with the linear relaxation of a bounded knapsack problem
- Maximum of one split customer per route
- Other customers receive a zero or a full delivery
- Integrality requirements not on the master problem dynamic columns
- Convex combinations of these columns can yield routes with multiple split deliveries
- Specialized dynamic programming algorithm for the subproblem

Our contribution

New branch-and-price method

- Subproblem is an elementary shortest path problem with resource constraints, combined with the linear relaxation of a bounded knapsack problem
- Maximum of one split customer per route
- Other customers receive a zero or a full delivery
- Integrality requirements not on the master problem dynamic columns
- Convex combinations of these columns can yield routes with multiple split deliveries
- Specialized dynamic programming algorithm for the subproblem

Outline

Integer master problem

$$\text{Minimize} \quad \sum_{r\in\mathcal{R}} \sum_{w\in\mathcal{W}_r} c_r \theta_{rw}$$

subject to :

$$\sum_{r\in\mathcal{R}} \sum_{w\in\mathcal{W}_r} \delta_{iw}\theta_{rw} \geq d_i, \qquad \forall\, i \in \mathcal{N}$$

$$\sum_{r\in\mathcal{R}} \sum_{w\in\mathcal{W}_r} a_{ir}\theta_{rw} \geq k_i^C, \qquad \forall\, i \in \mathcal{N}$$

$$\sum_{r\in\mathcal{R}\setminus\{0\}} \sum_{w\in\mathcal{W}_r} \theta_{rw} = H,$$

$$H \in \left[k^C(\mathcal{N}), |\mathcal{F}|\right], \ \text{integer},$$

Integer master problem (cont'd)

$$\theta_{rw} \geq 0, \qquad \forall\, r \in \mathcal{R}, w \in \mathcal{W}_r$$

$$\sum_{r \in \mathcal{R}} \sum_{w \in \mathcal{W}_r} x_{ijr}\theta_{rw} = y_{ij}, \qquad \forall\, (i,j) \in \mathcal{A}$$

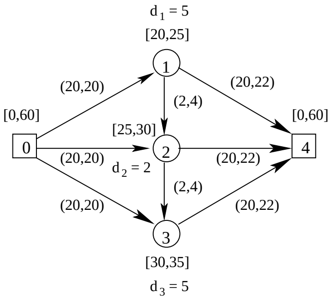
$$\sum_{r \in \mathcal{R}} \sum_{w \in \mathcal{W}_r} b_{ij\ell r}\theta_{rw} = z_{ij\ell}, \qquad \forall\, (i,j,\ell) \in \mathcal{B}$$

$$y_{ij} \;\; \text{binary}, \qquad \forall\, (i,j) \in \mathcal{A}(\mathcal{N})$$

$$y_{ij} \;\; \text{integer}, \qquad \forall\, (i,j) \in \mathcal{A} \setminus \mathcal{A}(\mathcal{N})$$

$$z_{ij\ell} \;\; \text{binary}, \qquad \forall\, (i,j,\ell) \in \mathcal{B}.$$

A small example with $n = 3$ customers and $Q = 4$



Optimal solution is (delivered quantities in parenthesis)

- 0-1(4)-4 with a flow of 1, 0-3(4)-4 with a flow of 1
- 0-1(2)-2(2)-3(0)-4 with a flow of 0.5
- 0-1(0)-2(2)-3(2)-4 with a flow of 0.5

Subproblem

$$\textit{Minimize} \quad \sum_{(i,j)\in\mathcal{A}} (c_{ij} - \alpha_i - \nu_{ij}) x_{ij} - \sum_{i\in\mathcal{N}} \pi_i \delta_i - \beta - \sum_{(i,j,\ell)\in\mathcal{B}} \mu_{ij\ell}\zeta_{ij\ell}$$

subject to :

$$\sum_{i\in\mathcal{N}} x_{0,i} = 1,$$

$$\sum_{j\in\mathcal{V}^+(i)} x_{ij} - \sum_{j\in\mathcal{V}^-(i)} x_{ji} = 0, \qquad \forall\, i \in \mathcal{N}$$

$$x_{ij}(s_i + t_{ij} - s_j) \leq 0, \qquad \forall\, (i,j) \in \mathcal{A}$$

$$e_i \leq s_i \leq l_i, \qquad \forall\, i \in \mathcal{V}$$

Subproblem (cont'd)

$$\sum_{i \in \mathcal{N}} \delta_i \leq Q,$$

$$0 \leq \delta_i \leq \bar{d}_i \sum_{j \in \mathcal{V}^+(i)} x_{ij}, \qquad \forall\, i \in \mathcal{N}$$

$$x_{ij} \in \{0,1\}, \qquad \forall\,(i,j) \in \mathcal{A}$$

$$x_{ij} + x_{j\ell} \leq \zeta_{ij\ell} + 1 \leq \frac{x_{ij} + x_{j\ell}}{2} + 1, \qquad \forall\,(i,j,\ell) \in \mathcal{B}$$

$$\zeta_{ij\ell} \in \{0,1\}, \qquad \forall\,(i,j,\ell) \in \mathcal{B}.$$

Assuming no ζ variables, the subproblem is an elementary shortest path problem with time windows and vehicle capacity, combined with the linear relaxation of a bounded knapsack problem

Outline

1. Introduction
   - Problem definition
   - Literature review
   - Main difficulty with branch-and-price for SDVRPTW
2. Formulation
   - Master problem
   - Subproblem
3. Branch-and-price-and-cut method
   - Column generation
   - Cutting and branching
4. Computational results
5. Conclusions

Branch-and-price-and-cut

- Column generation used to compute lower bounds
- Cutting planes added to strengthen linear relaxations
- Branching used to derive integer solutions

Column generation

- Standard column generation
- Label-setting algorithm for solving the subproblem
- Accelerating strategies

Label-setting algorithm

For the VRPTW

- Label E
  - C : reduced cost
  - S : time
  - L : total quantity delivered
  - V^i : customer i reachable or not
- Extension functions : e.g., $S_j = \max\{e_j, S_i + t_{ij}\}$
- Dominance : E_1 dominates E_2 if
  - $C_1 \leq C_2$
  - $S_1 \leq S_2$
  - $L_1 \leq L_2$
  - $V_1^i \leq V_2^i, \forall\, i \in \mathcal{N}$

Label-setting algorithm

Not applicable because

- delivered quantities are decision variables
- reduced cost and load resource are functions of these quantities

New algorithm for the SDVRPTW

- When extending a label along an arc, up to three labels can be created : zero, split, and full deliveries
- New binary resource to limit the number of split deliveries
- Reduced cost is a linear function of the total quantity delivered
- Dominance procedure must compare such functions

Label-setting algorithm

Not applicable because

- delivered quantities are decision variables
- reduced cost and load resource are functions of these quantities

New algorithm for the SDVRPTW

- When extending a label along an arc, up to three labels can be created : zero, split, and full deliveries
- New binary resource to limit the number of split deliveries
- Reduced cost is a linear function of the total quantity delivered
- Dominance procedure must compare such functions

Label-setting algorithm (cont'd)

Label E

- C : reduced cost excluding cost of split delivery (if any)
- S : time
- L : total quantity delivered in full deliveries
- V^i : customer i reachable or not
- P : split delivery done or not
- Δ : maximum quantity that can be delivered in the split delivery (if any)
- Π : unit dual price for the split delivery (if any)

- Adapted extension functions

Label-setting algorithm (cont'd)

Label E

- C : reduced cost excluding cost of split delivery (if any)
- S : time
- L : total quantity delivered in full deliveries
- V^i : customer i reachable or not
- P : split delivery done or not
- Δ : maximum quantity that can be delivered in the split delivery (if any)
- Π : unit dual price for the split delivery (if any)

- Adapted extension functions

Label-setting algorithm (cont'd)

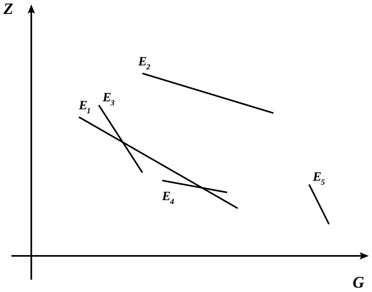
E_1 dominates E_2 if

- E_1 and E_2 are associated with the same node
- all feasible extensions of E_2 are also feasible for E_1
- cost of every feasible extension of E_2 is greater than or equal to the cost of a similar extension of E_1

Label-setting algorithm (cont'd)

Cost of a label $E = (C, S, L, V^i, P, \Delta, \Pi)$ is

$$Z(G) = C - \Pi(G - L) \quad \text{for } G \in [L, L + \Delta]$$



E_1 can dominate E_2, but not the other labels

Label-setting algorithm (cont'd)

E_1 can dominate E_2 if

- $S_1 \leq S_2$
- $L_1 \leq L_2$
- $V_1^i \leq V_2^i, \forall\, i \in \mathcal{N}$
- $P_1 \leq P_2$
- $C_1 - \Delta_1 \Pi_1 \leq C_2 - \Delta_2 \Pi_2$
- $C_1 - (L_2 - L_1)\Pi_1 \leq C_2$
- $C_1 - (L_2 + \Delta_2 - L_1)\Pi_1 \leq C_2 - \Delta_2 \Pi_2$

Accelerating strategies

1. Initial columns : dedicated trips $0 - i - n + 1$
2. If $\pi_j = 0$, then only zero deliveries at j
3. Bounded bidirectional search and decremental search space (Righini and Salani, 2006, 2007, Boland et al., 2006)
4. Heuristic column generator (omit $V_1^i \leq V_2^i, \forall\, i \in \mathcal{N}$, in the dominance rule)

Accelerating strategies

1. Initial columns : dedicated trips $0 - i - n + 1$
2. If $\pi_j = 0$, then only zero deliveries at j
3. Bounded bidirectional search and decremental search space (Righini and Salani, 2006, 2007, Boland et al., 2006)
4. Heuristic column generator (omit $V_1^i \leq V_2^i, \forall i \in \mathcal{N}$, in the dominance rule)

Accelerating strategies

1. Initial columns : dedicated trips $0 - i - n + 1$
2. If $\pi_j = 0$, then only zero deliveries at j
3. Bounded bidirectional search and decremental search space (Righini and Salani, 2006, 2007, Boland et al., 2006)
4. Heuristic column generator (omit $V_1^i \leq V_2^i, \forall i \in \mathcal{N}$, in the dominance rule)

Accelerating strategies

1. Initial columns : dedicated trips $0 - i - n + 1$
2. If $\pi_j = 0$, then only zero deliveries at j
3. Bounded bidirectional search and decremental search space (Righini and Salani, 2006, 2007, Boland et al., 2006)
4. Heuristic column generator (omit $V_1^i \leq V_2^i, \forall\, i \in \mathcal{N}$, in the dominance rule)

Cutting

- k-path inequalities (Kohl et al., 1999)
  - $k_{\mathcal{U}} = \max\{k_{\mathcal{U}}^{C}, k_{\mathcal{U}}^{T}\}$, where
    - $k_{\mathcal{U}}^{C} = \lceil \sum_{i \in \mathcal{U}} \frac{d_i}{Q} \rceil$: minimum according to vehicle capacity
    - $k_{\mathcal{U}}^{T}$: minimum according to time windows (1 or 2)
- Arc-flow inequalities (Gendreau et al, 2006)
  - Maximum flow of 1 on arcs (i, j) and (j, i) for $i, j \in \mathcal{N}$

Cutting

- k-path inequalities (Kohl et al., 1999)
  - $k_{\mathcal{U}} = \max\{k_{\mathcal{U}}^C, k_{\mathcal{U}}^T\}$, where
    - $k_{\mathcal{U}}^C = \lceil \sum_{i \in \mathcal{U}} \frac{d_i}{Q} \rceil$: minimum according to vehicle capacity
    - $k_{\mathcal{U}}^T$: minimum according to time windows (1 or 2)
- Arc-flow inequalities (Gendreau et al, 2006)
  - Maximum flow of 1 on arcs (i, j) and (j, i) for $i, j \in \mathcal{N}$

Branching

In order of priority, we branch on

- number of vehicles used (H)
- number of vehicles visiting a customer $(\sum_j y_{ij})$
  - add y_{ij} and corresponding constraint in the master problem
- number of vehicles on an arc (y_{ij})
  - add y_{ij} and corresponding constraint in the master problem
- number of vehicles on two consecutive arcs $(z_{ij\ell})$
  - add $z_{ij\ell}$ and corresponding constraint in the master problem
  - modify the subproblem algorithm

Outline

Instances

Modified VRPTW instances of Solomon (1987)

- Allow split deliveries
- Solomon : 56 instances with 100 customers (6 classes)
- 2×56 other instances taking the first 25 and 50 customers
- $Q = 30, 50, 100$
- Total of 504 instances
- Same instances as in Gendreau et al. (2006)

Maximum CPU time = 1 hour, 2.8GHz PC

Linear relaxation results

Computational time

- increases with number of customers
- decreases with capacity Q
- Slower increase than with the method of Gendreau et al. (2006) who used a 1.6GHz PC

Example (C1 instances with $Q = 30$)

Gendreau et al. : 0.3, 8, 304 seconds for $n = 25, 50, 100$
Our method : 0.4, 3, 17 seconds for $n = 25, 50, 100$

Integer solution results

n	class	Q	Nb inst	numbers solved	numbers veh	numbers splits	gap (%) lp	gap (%) lp+cuts	numbers cuts	numbers nodes	times (s) lp	times (s) cuts	times (s) total
25	R1	30	12	12	12.0	3.8	2.1	0.3	42.1+1.2	709.5	<1	2	117
		50	12	12	7.3	1.0	1.6	0.1	25.3+0.0	5.6	1	26	30
		100	12	12	5.1	0.1	0.5	0.2	2.5+0.0	3.9	1	8	11
25	C1	30	9	4	16.0	5.0	2.9	0.3	58.3+23.5	15918.0	<1	1	1439
		50	9	9	10.0	1.8	1.8	<0.1	25.8+0.1	3.2	1	3	7
		100	9	8	5.0	0.0	1.9	1.0	25.0+1.6	123.8	2	43	254
25	RC1	30	8	8	18.0	7.0	2.5	<0.1	86.1+3.5	1422.5	<1	<1	268
		100	8	8	6.0	0.4	0.8	<0.1	11.6+0.0	1.3	1	<1	2
25	R2	30	11	11	12.0	3.5	2.4	0.5	49.1+1.8	1220.6	3	2	462
		50	11	11	7.0	1.0	1.5	0.1	21.6+0.0	17.0	10	4	25
		100	11	9	4.0	0.3	1.6	0.8	8.5+0.7	57.7	73	19	300
25	C2	30	8	4	16.0	6.0	1.4	0.2	53.0+ 9.5	2563.5	<1	<1	429
		50	8	3	10.0	2.0	1.3	0.4	30.3+13.3	4342.0	<1	4	1410
		100	8	8	5.0	0.6	0.8	0.1	12.6+0.0	5.3	11	7	40
25	RC2	30	8	7	18.0	7.0	1.8	<0.1	82.6+2.0	1055.8	1	<1	465
		100	8	8	6.0	0.4	0.8	<0.1	11.5+0.0	1.9	14	<1	19

Maximum CPU time = 1 hour, 2.8GHz PC

Integer solution results (cont'd)

			Nb	numbers			gap (%)		numbers		times (s)		
n	class	Q	inst	solved	veh	splits	lp	lp+cuts	cuts	nodes	lp	cuts	total
50	R1	50	12	1	15.0	3.0	2.3	0.4	28.0+3.0	155.0	<1	81	91
		100	12	5	9.8	0.4	1.2	0.9	8.6+1.2	346.2	2	426	1145
50	RC1	50	8	8	20.0	3.9	0.6	<0.1	34.1+0.1	1.1	3	<1	10
		100	8	8	10.0	0.6	0.9	0.1	17.9+0.1	8.0	11	<1	34
50	R2	100	11	1	8.0	1.0	0.8	0.7	20.0+1.0	109.0	107	1214	1806
50	C2	50	8	1	18.0	6.0	1.3	0.2	73.0+12.0	2001.0	4	7	1522
50	RC2	50	8	8	20.0	4.9	0.6	<0.1	33.3+0.1	1.5	16	<1	37
		100	8	8	10.0	0.8	1.0	0.1	16.4+0.1	14.3	244	<1	384
100	R1	100	12	1	20.0	0.0	0.1	<0.1	7.0+0.0	5.0	2	13	17

Maximum CPU time = 1 hour,   2.8GHz PC

Integer solution results (cont'd)

Remarks

- Gap decreases with capacity Q (need split deliveries)
- Cycling increases with capacity Q
- Large gaps for C1 and C2 instances

- k-path inequalities are useful
- Arc-flow inequalities are not useful

- Gendreau et al. (2006) solved 27 instances (1.6GHz PC)
- We solved 175 instances (2.8GHz PC)

Outline

Conclusions

In this paper

- Novel decomposition
- New subproblem type
- New label-setting algorithm
- Relatively good results

We can do better

- Accelerating strategies for solving the subproblem
- Heuristics for solving the subproblem
- Other valid inequalities to reduce gaps