Branch-price-and-cut algorithms for electric vehicle routing problems with time windows

Guy Desaulniers
École Polytechnique de Montréal and GERAD, Canada

Co-authors: Fausto Errico, Stefan Irnich, Michael Schneider

Route 2014
Snekkersten, Denmark, June 4, 2014

Warning

This is the last talk of the workshop so I will skip some technical stuff to make it shorter

and maybe less boring

Warning

This is the last talk of the workshop so I will skip some technical stuff to make it shorter

and maybe less boring

Outline

Outline

Vehicle routing problem with time windows (VRPTW)

Definition

- Given
  - A single depot with identical capacitated vehicles
  - A set of customers with known demands and time windows
- Find vehicle routes such that
  - All customer demands are met, each by a single vehicle
  - Each route starts and ends at the depot
  - Each route satisfies vehicle capacity and time windows
  - Total cost (distance) is minimized

Electric VRPTW (EVRPTW)

Definition

- Same as VRPTW except
  - Battery capacity and recharging stations
    - Battery consumption is a linear function of traveled distance
    - Recharging time is a linear function of the recharged quantity
    - Battery fully recharged overnight at depot
    - No recharging costs
  - Maximum route length (in distance traveled)

Four variants: SF, SV, MF, MV

- S: Single recharge allowed per route
- M: Multiple recharges allowed per route
- F: Full recharges only
- V: Variable recharges allowed

Electric VRPTW (EVRPTW)

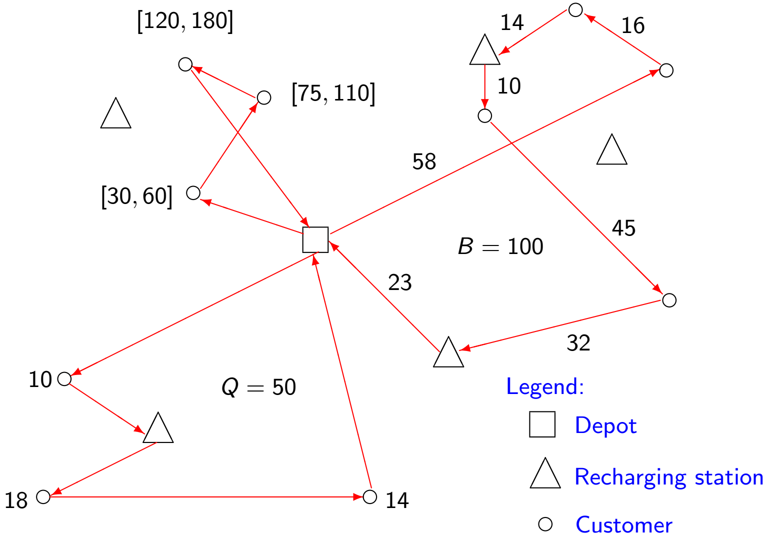
Definition

- Same as VRPTW except
  - Battery capacity and recharging stations
    - Battery consumption is a linear function of traveled distance
    - Recharging time is a linear function of the recharged quantity
    - Battery fully recharged overnight at depot
    - No recharging costs
  - Maximum route length (in distance traveled)

Four variants: SF, SV, MF, MV

- S: Single recharge allowed per route
- M: Multiple recharges allowed per route
- F: Full recharges only
- V: Variable recharges allowed

Example of a solution



Literature

- Conrad and Figliozzi (2011)
  - Battery capacity with recharging possibility at certain customers
  - Fixed recharging time
  - Local search heuristic
- Erdogan and Miller-Hooks (2012)
  - Green VRP
  - No time windows, nor vehicle capacity
  - Limited vehicle autonomy with recharging stations
  - Fixed recharging time
  - Exact MIP approach and local search heuristic
- Schneider, Stenger, Goeke (2014)
  - Battery capacity with recharging stations
  - Recharging time depends on recharged quantity
  - Full recharges only
  - Variable neighborhood search combined with tabu search

Our goal is to develop exact algorithms for solving four variants of the EVRPTW: SF. SV, MF, and MV

Outline

A path-flow model

A feasible route

- Starts and ends at the depot
- Does not visit a customer more than once
- Respects
  - Vehicle capacity
  - Time windows
  - Maximum length
  - Battery capacity
    - Visits recharging stations if needed
    - At most one recharge in S variants
    - Full recharges in F variants

A path-flow model (cont'd)

Notation

N: Set of customers

Ω: Set of feasible elementary $o - d$ paths in G (elementarity required only at customers)

c_p: Cost of feasible path $p \in \Omega$

a_{pi}: Binary parameter equal to 1 if customer $i \in N$ is visited in path $p \in \Omega$

θ_p: Binary path-flow variable equal to 1 if path $p \in \Omega$ is selected

A path-flow model (cont'd)

$$\min \sum_{p \in \Omega} c_p \theta_p \qquad (1)$$ total cost

$$\text{s.t.} \sum_{p \in \Omega} a_{pi} \theta_p = 1, \qquad \forall\, i \in N \qquad (2)$$ visit each customer

$$\theta_p \in \{0, 1\}, \qquad \forall\, p \in \Omega \qquad (3)$$ binary requirements

Remarks

- Route feasibility rules are implicitly taken into account in the definition of Ω
- Same model as for the VRPTW. Differs only by the definition of set Ω.

A path-flow model (cont'd)

$$\min \sum_{p \in \Omega} c_p \theta_p \qquad (1) \quad \text{total cost}$$

$$\text{s.t.} \sum_{p \in \Omega} a_{pi} \theta_p = 1, \qquad \forall\, i \in N \qquad (2) \quad \text{visit each customer}$$

$$\theta_p \in \{0, 1\}, \qquad \forall\, p \in \Omega \qquad (3) \quad \text{binary requirements}$$

Remarks

- Route feasibility rules are implicitly taken into account in the definition of Ω
- Same model as for the VRPTW. Differs only by the definition of set Ω.

Outline

Branch-price-and-cut

Branch-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

- Master problem corresponds to the linear relaxation of the above model
- One subproblem that corresponds to an elementary shortest path problem with resource constraints

Subproblem

Defined on a directed network $G = (N, A)$

- Node set $N = C \cup R \cup \{o, d\}$
  - C: Set of customers
  - R: Set of recharging stations
  - $\{o, d\}$: Depot at start and end of the day
- Arc set $A \subset N \times N$
  - From o to each node in $C \cup R$
  - From each node in $C \cup R$ to d
  - From each node i in $C \cup R$ to each node j in $C \cup R$ if $i \neq j$ and j can be visited immediately after i according to time windows, vehicle capacity, and battery capacity

Subproblem definition

There is one subproblem (single depot and identical vehicles):

$$\min_{p \in \Omega} \quad \bar{c}_p$$

where $\bar{c}_p$ is the reduced cost of variable θ_p:

$$\bar{c}_p = c_p - \sum_{i \in N} a_{pi} \pi_i = \sum_{(i,j) \in p} (c_{ij} - \pi_i) = \sum_{(i,j) \in p} \bar{c}_{ij}$$

and π_i, $i \in N$, are the dual variables associated with constraints (2) and $\bar{c}_{ij} = c_{ij} - \pi_i$ is the "reduced cost" of arc (i,j) (assuming $\pi_i = 0$ if $i \notin N$)

Subproblem definition (cont'd)

- Subproblem corresponds to an elementary shortest path problem with resource constraints (ESPPRC) on network G
- Resources are required to impose
  - Vehicle capacity
  - Time windows
  - Battery capacity
  - Maximum route length
  - Elementarity
- Battery consumption can be reset (fully or partially) at recharging stations
- Recharging time depends on recharged quantity

Labeling algorithm

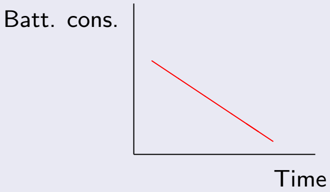
- ESPPRC can be solved by a labeling algorithm
  - Partial paths are represented by multi-dimensional resource vectors, called labels
  - From node o, labels are propagated forwardly through network G using resource extension functions (REFs)
  - Resource windows are checked at each node to discard infeasible partial paths
  - A dominance rule is applied to discard unpromising labels

Labeling for F variants

- A label contains the following components
  - Reduced cost
  - Load
  - Route length
  - Time
  - Battery consumption
  - Number of recharges (for SF variant)
- Standard REFs except on arcs leaving a node in R
  - Recharging time is computed and added
  - Battery consumption must be reset to 0
- Standard dominance rule (lower battery consumption yields lower recharging time)

Labeling for V variants

- A label contains the same components except that
  - Battery consumption is a function of the time once a recharge station has been visited



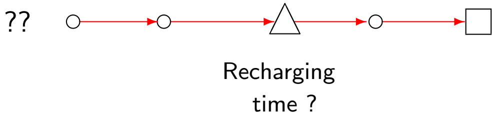
- REFs must be adapted to extend these dependent resources
  - Can be truncated from the left if waiting occurs before a TW
  - Must be truncated from the right if TW upper bound is exceeded
- Adapted dominance rule that compares line segments

Bidirectional labeling for F variants

- When extending labels backward
  - Consider all previous components
  - Recharging time at a recharging station depends on unknown battery consumption
  - Requires two additional components:
    - Available time for next recharge
    - Unavoidable waiting time before next recharge
- Adapted backward REFs and dominance rule

Bidirectional labeling for F variants

- When extending labels backward
  - Consider all previous components
  - Recharging time at a recharging station depends on unknown battery consumption
  - Requires two additional components:
    - Available time for next recharge
    - Unavoidable waiting time before next recharge
- Adapted backward REFs and dominance rule



Bidirectional labeling for V variants

- Backward labeling similar to the variant F case except that you must consider again battery consumption as a function of time
- Adapted REFs and dominance rule

Other acceleration techniques

- *ng*-paths (Baldacci et al., 2011)
  - Allow certain cycles
  - Less restrictive dominance rule (increased number of dominated labels)
  - Possibly weaker lower bounds but shorter CPU times for the subproblems
- Heuristic pricing first (network with less arcs and dominance on a subset of the customer resources)

Valid inequalities

Two families of inequalities

- Two-path inequalities (Kohl et al., 1999)
  - S: Subset of customers for which it can be proven that it cannot be serviced by a single vehicle
  - Cut for S imposes a minimum number of two vehicles entering S
  - Separation by enumeration of subsets S that considers time windows and battery capacity
- Subset-row inequalities (Jepsen et al., 2008)
  - S: subset of three customers
  - Cut for S imposes a maximum of one route visiting at least two customers in S
  - An additional resource for each cut is required in the subproblem

Branching

Branch on

- Arc flow
- Total number of vehicles (to come)
- Total number of recharges (to come)

Branching

Branch on

- Arc flow
- Total number of vehicles (to come)
- Total number of recharges (to come)

Outline

Instances

Derived from Solomon's instances (as in Schneider, 2012)

- R1, C1, RC1 instances only (relatively narrow adjusted time windows)
- A few customers moved closer to the depot randomly to ensure feasibility for the S variants
- Instances with 25, 50 and 100 customers (total: 29×3)
- In each instance, the same 21 recharging stations
- Battery capacity allows to travel 60% of the average route length in the VRPTW solution
- Full recharge from empty takes about 3 times the service time at a customer

Computational experiments

- Performed on a processor Intel Core i7-4770 CPU @ 3.40 GHz
- One core used
- One-hour time limit

Main results (preliminary)

Instances	Variant SF				Variant MF			
	No. Solved	Avg. Time (s)	Avg. No. Veh.	Avg. No. Rch./Veh	No. Solved	Avg. Time (s)	Avg. No. Veh.	Avg. No. Rch./Veh
25 cust.	29	4	6.4	0.70	29	24	6.1	0.89
50 cust.	26	> 530	9.7	0.73	28	> 330	9.2	0.90
100 cust.	10	$> 2,526$	15.1	0.60	14	$> 2,282$	15.6	0.84
all	65	$> 1,020$	9.1	0.69	71	> 879	9.4	0.88

Instances	Variant SV				Variant MV			
	No. Solved	Avg. Time (s)	Avg. No. Veh.	Avg. No. Rch./Veh	No. Solved	Avg. Time (s)	Avg. No. Veh.	Avg. No. Rch./Veh
25 cust.	29	3	6.4	0.70	29	4	6.1	0.94
50 cust.	27	> 368	9.2	0.76	27	> 403	8.9	1.04
100 cust.	10	$> 2,497$	15.1	0.58	12	$> 2,559$	15.1	1.05
all	66	> 956	8.8	0.71	68	> 989	8.8	1.00

Cost comparison between variants

Instances	SF to SV		MF to MV	
	No. Common	Avg. Cost Decrease (%)	No. Common	Avg. Cost Decrease (%)
25 cust.	29	0.9	29	1.6
50 cust.	26	1.1	27	2.0
100 cust.	8	0.8	11	2.1
all	63	1.0	67	1.8

Instances	SF to MF		SV to MV	
	No. Common	Avg. Cost Decrease (%)	No. Common	Avg. Cost Decrease (%)
25 cust.	29	1.6	29	2.3
50 cust.	25	2.9	26	3.5
100 cust.	9	1.5	7	2.6
all	63	2.1	62	2.8

Bidirectional vs monodirectional labeling

	MF: Mono. to Bid.		
Instances	No. Common	Avg. Time Decrease (%)	Add. Solved
25 cust.	29	18.1	0
50 cust.	26	36.9	2
100 cust.	9	44.8	5
all	64	29.5	7

	MV: Mono. to Bid.		
Instances	No. Common	Avg. Time Decrease (%)	Add. Solved
25 cust.	29	35.0	0
50 cust.	26	51.8	1
100 cust.	5	62.6	7
all	60	44.6	8

Outline

Conclusions

- We presented the first exact algorithms for different variants of the EVRPTW
- Work in progress: to come
  - Other branching rules
  - Fast local search heuristic to generate columns
  - Configuration of strategies and parameter values
  - Tests on instances in the 200 series (wide time windows)

Thank you!

Questions?

Conclusions

- We presented the first exact algorithms for different variants of the EVRPTW
- Work in progress: to come
  - Other branching rules
  - Fast local search heuristic to generate columns
  - Configuration of strategies and parameter values
  - Tests on instances in the 200 series (wide time windows)

Thank you!

Questions?

Conclusions

- We presented the first exact algorithms for different variants of the EVRPTW
- Work in progress: to come
  - Other branching rules
  - Fast local search heuristic to generate columns
  - Configuration of strategies and parameter values
  - Tests on instances in the 200 series (wide time windows)

Thank you!

Questions?