The Electric Vehicle Routing and Overnight Charging Scheduling Problem on a Multigraph

Daniel Yamín[1] Guy Desaulniers[2] Jorge E. Mendoza[3]

[1]Centro para la Optimización y Probabilidad Aplicada (COPA), Departamento de Ingeniería Industrial, Universidad de los Andes, Bogotá, Colombia.

[2]Polytechnique Montréal and GERAD, Montreal, Quebec, Canada.

[3]HEC Montréal, Montreal, Quebec, Canada.

2023 Column Generation Workshop
May 17, 2023

Table of Contents

Table of Contents

The mE-VRSPTW

The Multigraph-based Electric Vehicle Routing and Overnight Charging Scheduling Problem with Time Windows (**mE-VRSPTW**)

The mE-VRSPTW

The Multigraph-based Electric Vehicle Routing and Overnight Charging Scheduling Problem with Time Windows (**mE-VRSPTW**)

Given

- Unlimited number of identical EVs with a given load capacity Q and energy capacity E, housed in a single depot
- Limited number B of identical chargers located at the depot
- Set of customers $\mathcal{C}$ with known demands q_i
- A time window $[\underline{t}_i, \overline{t}_i]$ for each customer $i \in \mathcal{C}$
- The road network
- Travel and service times, as well as energy consumptions

The mE-VRSPTW

Find vehicle routes and their preceding overnight charging schedule such that

- All customer demands are met
- Each customer is visited by a single vehicle
- Charging station capacity is not exceeded
- Each route starts and ends at the depot
- Each route is load-, time-, and energy-feasible
- Each route is scheduled for one continuous recharge in a single charger
- Total cost (distance) is minimized

Time Horizon

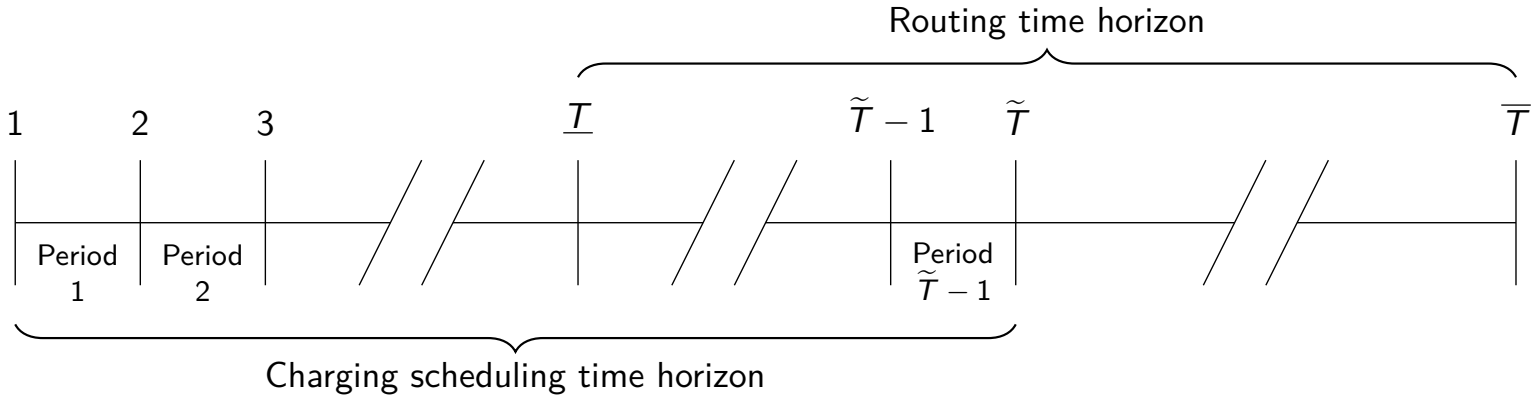


Figure 1: Routing and charging scheduling time horizons.

- $\mathcal{T} = \{1, \ldots, \underline{T}, \ldots, \widetilde{T} - 1\}$: **discrete charging timesteps**
- $\widetilde{T}$: **latest departure time** from the depot
- $[\underline{T}, \overline{T}]$: depot's time window
- Every vehicle initially (at time 1) has **zero energy**

Recharging Function

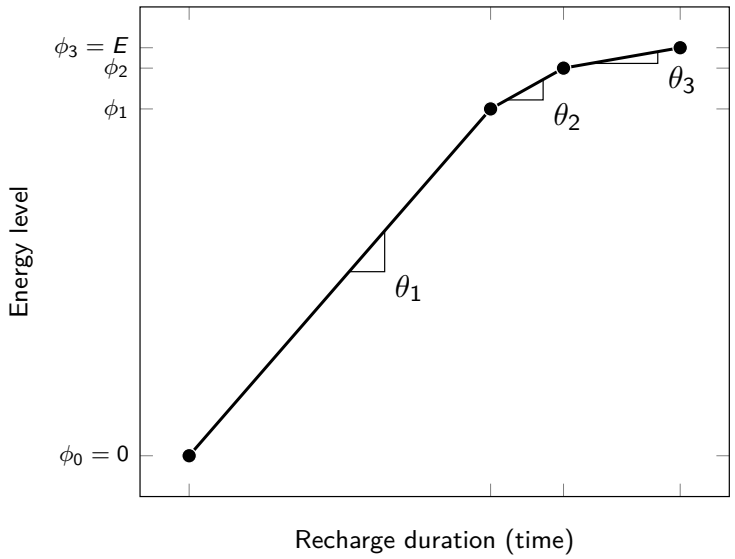


Figure 2: Piecewise-linear recharging function (Montoya et al., 2017).

- **Piecewise-linear function** f with a set $\mathcal{P} = \{1, \ldots, P\}$ of pieces
- **Energy levels**: $0 = \phi_0 < \phi_1 < \ldots < \phi_P = E$
- **Recharging rates** (energy per timestep): $\theta_1 > \theta_2 > \ldots > \theta_P > 0$

Charging Schedule

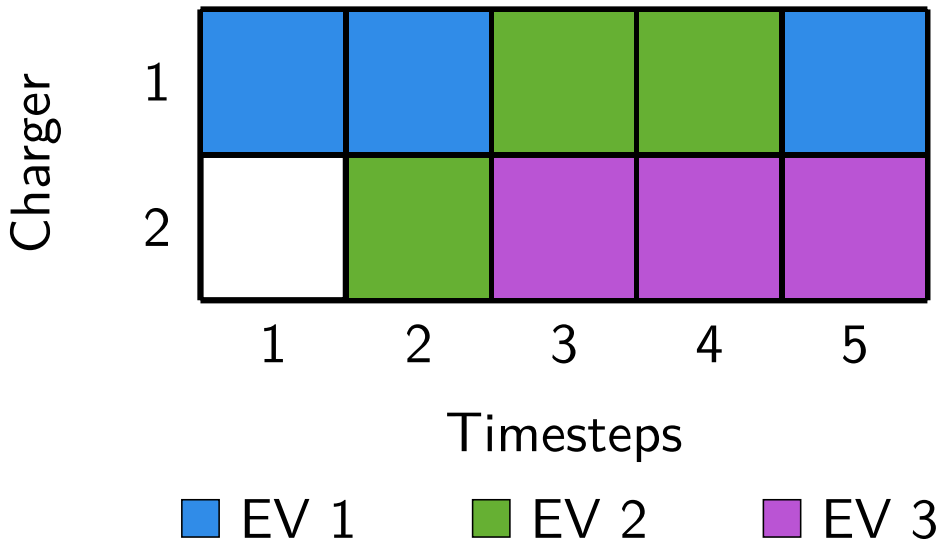


Figure 3: Infeasible charging schedule. Adapted from Lam et al. (2022).

- **Non-preemptively** and in **a single charger**
- **Sufficient energy** when departing from the depot

Notation

Vertex set $\mathcal{V} = \{0\} \cup \mathcal{C}$

Notation

Vertex set $\mathcal{V} = \{0\} \cup \mathcal{C}$

Arc multiset $\mathcal{A} \subseteq \bigcup_{(i,j)\in\mathcal{V}^2} \mathcal{A}_{(i,j)}$

- Alternative ways to travel: $\mathcal{A}_{(i,j)} = \{(i,j)^k \mid k = 1, \ldots, |\mathcal{A}_{(i,j)}|\}$
- Driving distance $c_{(i,j)^k}$
- Travel time $t_{(i,j)^k}$ (including service time at vertex i if any)
- Energy consumption $e_{(i,j)^k}$

Notation

Vertex set $\mathcal{V} = \{0\} \cup \mathcal{C}$

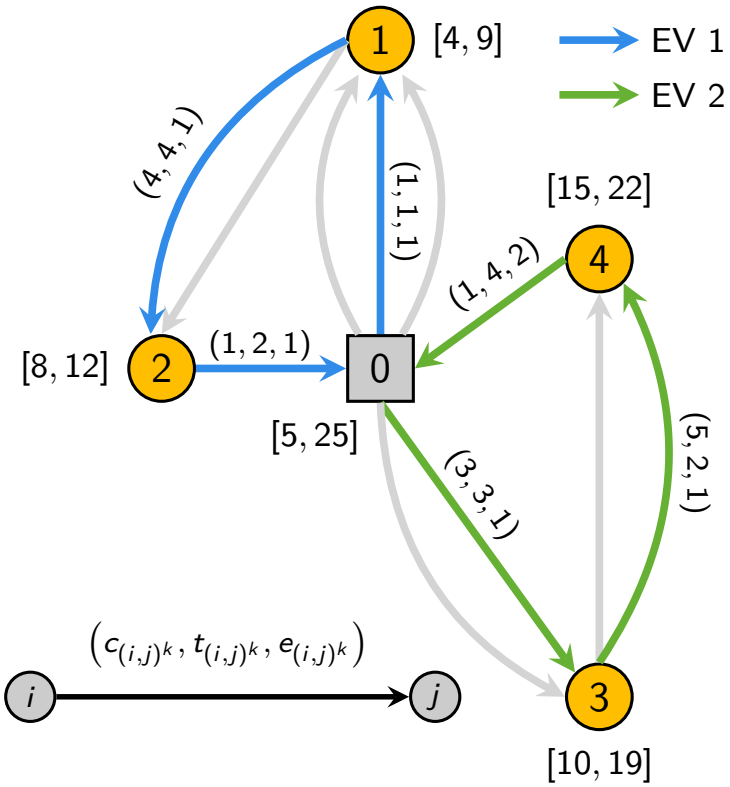
Arc multiset $\mathcal{A} \subseteq \bigcup_{(i,j)\in\mathcal{V}^2} \mathcal{A}_{(i,j)}$

- Alternative ways to travel: $\mathcal{A}_{(i,j)} = \{(i,j)^k \mid k = 1, \ldots, |\mathcal{A}_{(i,j)}|\}$
- Driving distance $c_{(i,j)^k}$
- Travel time $t_{(i,j)^k}$ (including service time at vertex i if any)
- Energy consumption $e_{(i,j)^k}$

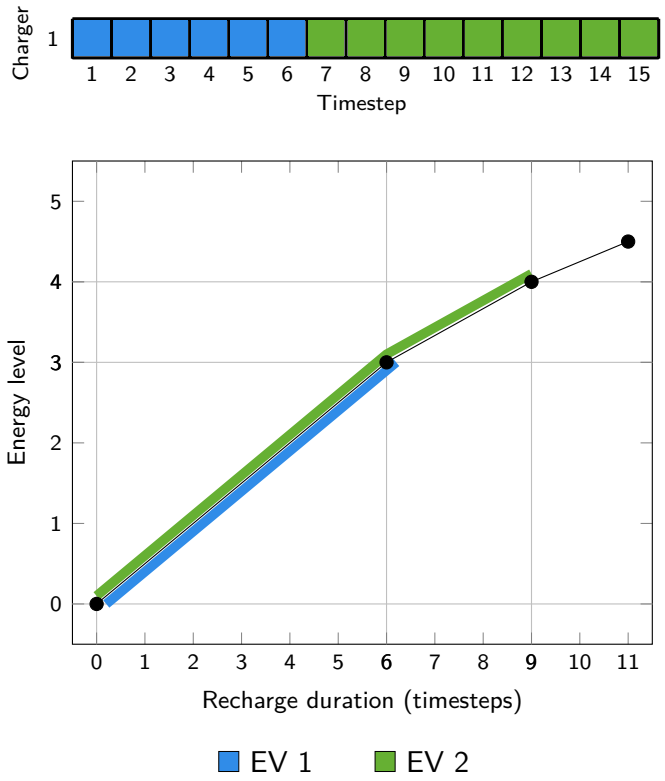
Multigraph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$

- Trade-off: consumed energy and the traveled distance

Illustrative Example



(a) Routing.



(b) Charging scheduling.

- $Q = 10$, $E = 5$, and $B = 1$
- $q_i = 5, i \in \mathcal{C}$

Set-Partitioning Formulation

$$
\begin{aligned}
\min \quad & \sum_{r \in \mathcal{R}} c_r \lambda_r & & & (1a)\\
\text{s.t.} \quad & \sum_{r \in \mathcal{R}} a_i^r \lambda_r = 1 & & \forall i \in \mathcal{C}, & (1b)\\
& \sum_{r \in \mathcal{R}} b_t^r \lambda_r \leq B & & \forall t \in \mathcal{T}, & (1c)\\
& \lambda_r \in \{0, 1\} & & \forall r \in \mathcal{R}. & (1d)
\end{aligned}
$$

Set-Partitioning Formulation

$$
\begin{aligned}
\min \quad & \sum_{r \in \mathcal{R}} c_r \lambda_r & & & (1a)\\
\text{s.t.} \quad & \sum_{r \in \mathcal{R}} a_i^r \lambda_r = 1 & & \forall i \in \mathcal{C}, & (1b)\\
& \sum_{r \in \mathcal{R}} b_t^r \lambda_r \leq B & & \forall t \in \mathcal{T}, & (1c)\\
& \lambda_r \in \{0, 1\} & & \forall r \in \mathcal{R}. & (1d)
\end{aligned}
$$

- (1a): minimize total **travel cost**
- (1b): each **customer is visited** by one EV
- (1c): meet **charging station capacity** each period

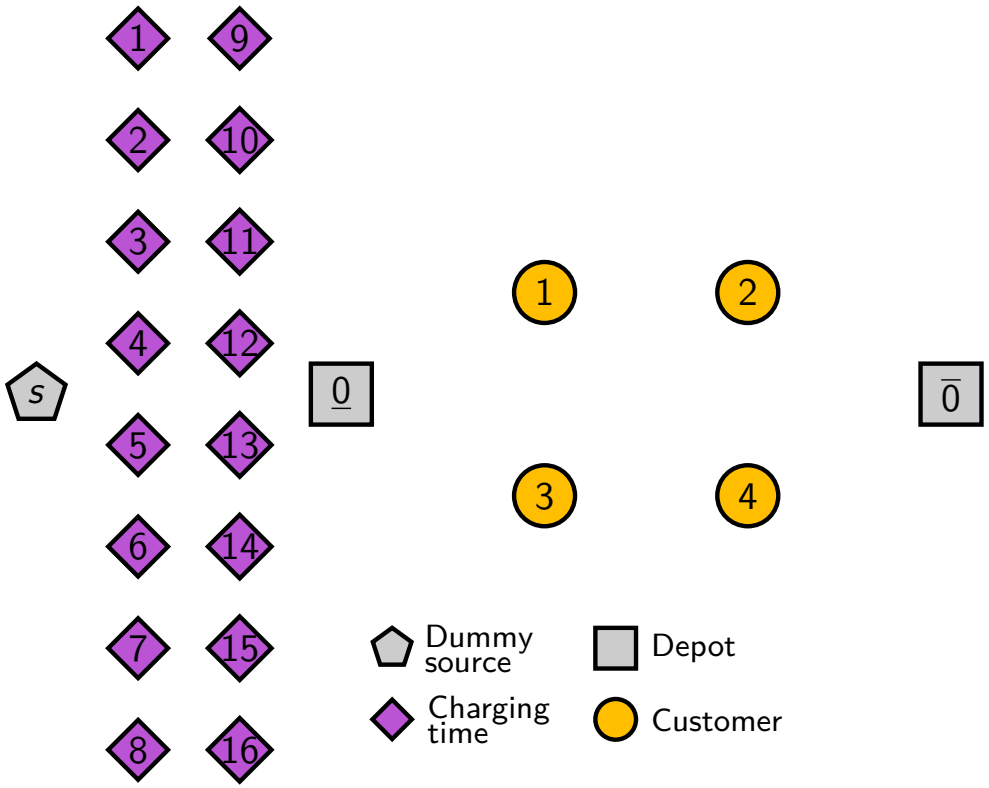
Set-Partitioning Formulation

$$
\begin{aligned}
\min \quad & \sum_{r \in \mathcal{R}} c_r \lambda_r & & & (1a)\\
\text{s.t.} \quad & \sum_{r \in \mathcal{R}} a_i^r \lambda_r = 1 & & \forall i \in \mathcal{C}, & (1b)\\
& \sum_{r \in \mathcal{R}} b_t^r \lambda_r \leq B & & \forall t \in \mathcal{T}, & (1c)\\
& \lambda_r \in \{0, 1\} & & \forall r \in \mathcal{R}. & (1d)
\end{aligned}
$$

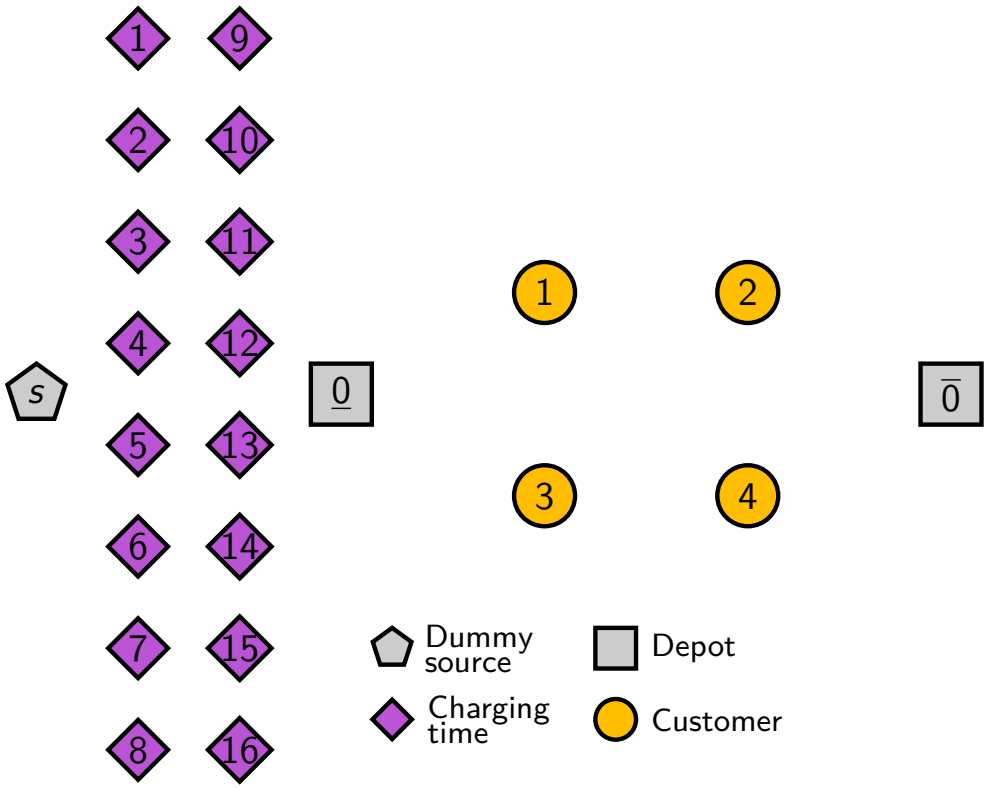
- (1a): minimize total **travel cost**
- (1b): each **customer is visited** by one EV
- (1c): meet **charging station capacity** each period
- $\alpha_i \in \mathbb{R}$: dual of (1b) for customer $i \in \mathcal{C}$
- $\beta_t \leq 0$: dual of (1c) for timestep $t \in \mathcal{T}$

Table of Contents

Pricing Problem Multigraph ($\mathcal{V}_{\mathrm{P}}$)

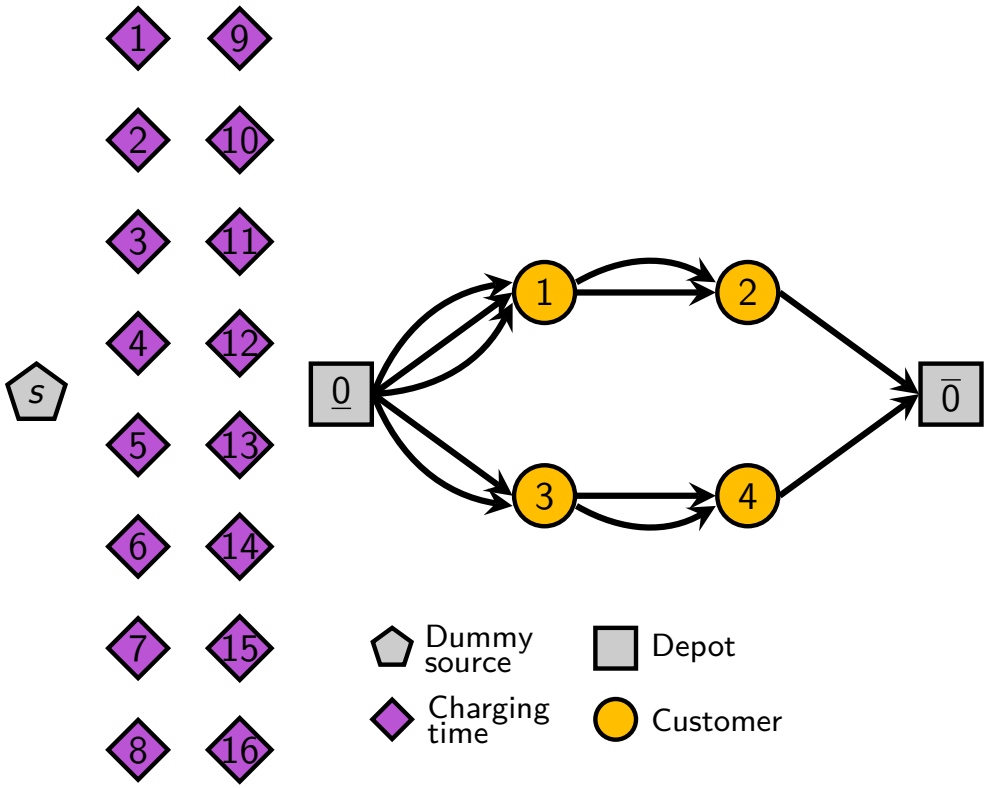


Pricing Problem Multigraph ($\mathcal{V}_\mathsf{P}$)



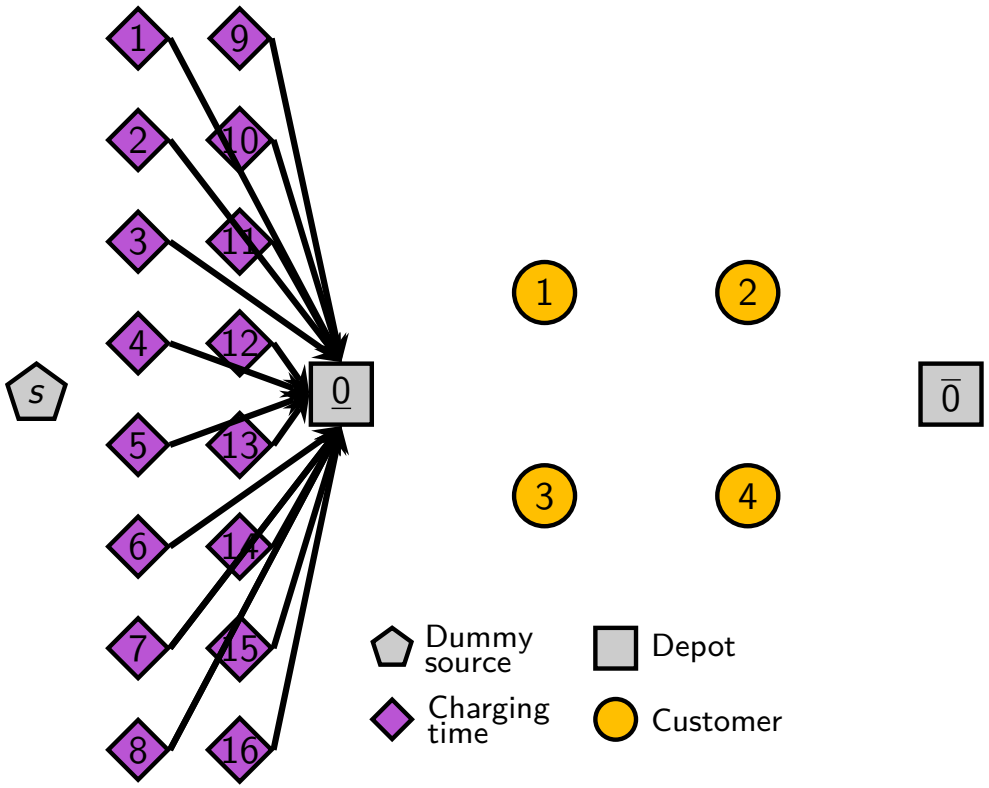
- $\mathcal{V}_\mathsf{P} = \{s\} \cup \mathcal{T} \cup \{\underline{0}\} \cup \mathcal{C} \cup \{\overline{0}\}$

Pricing Problem Multigraph ($\mathcal{A}$)



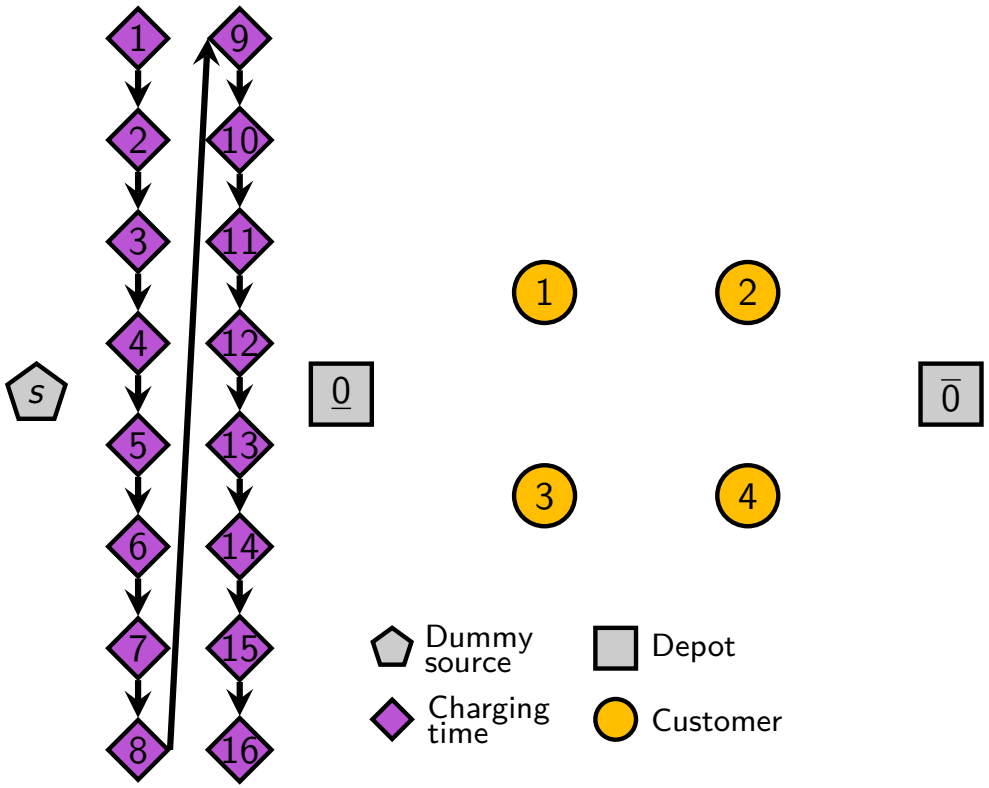
- $\mathcal{V}_{\mathsf{P}} = \{s\} \cup \mathcal{T} \cup \{\underline{0}\} \cup \mathcal{C} \cup \{\overline{0}\}$
- $\mathcal{A}_{\mathsf{P}} = \mathcal{A}$

Pricing Problem Multigraph ($\mathcal{A}_1$)



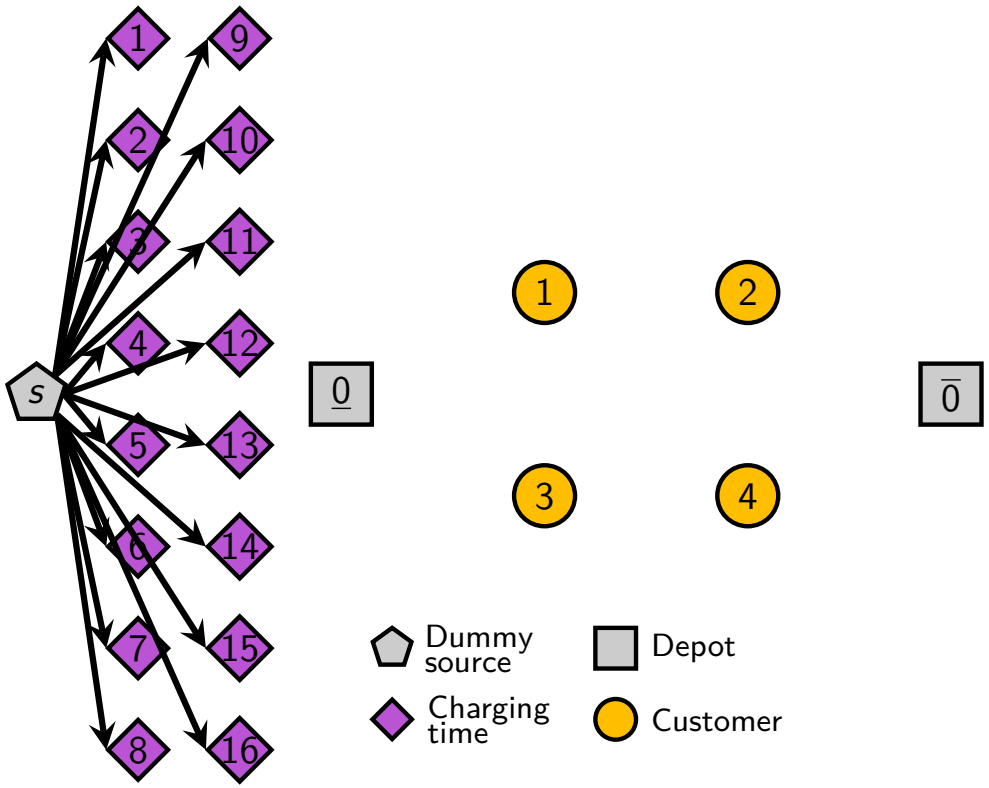
- $\mathcal{V}_{\mathsf{P}} = \{s\} \cup \mathcal{T} \cup \{\underline{0}\} \cup \mathcal{C} \cup \{\overline{0}\}$
- $\mathcal{A}_{\mathsf{P}} = \mathcal{A} \cup \mathcal{A}_1$

Pricing Problem Multigraph ($\mathcal{A}_2$)



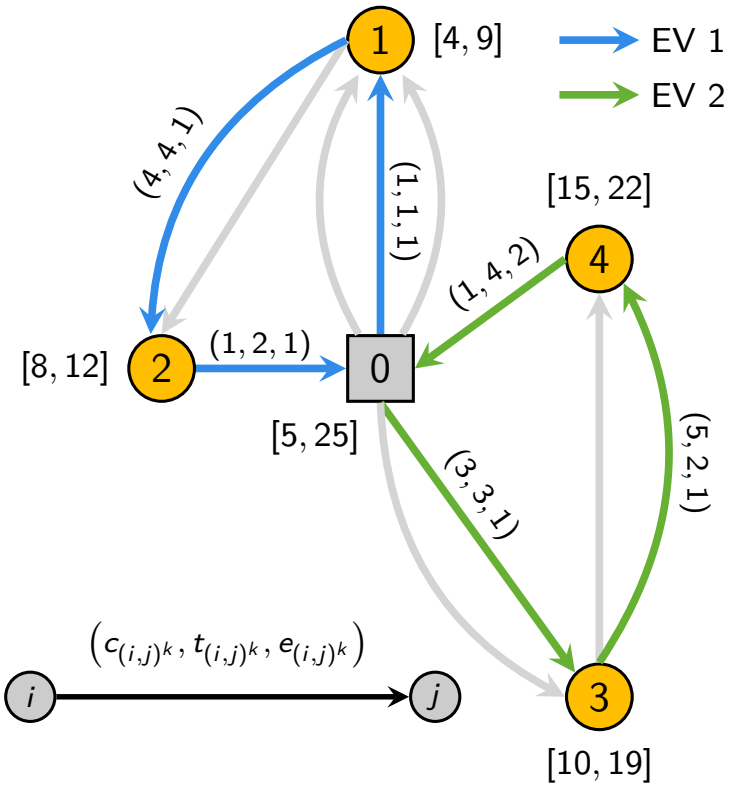
- $\mathcal{V}_{\mathsf{P}} = \{s\} \cup \mathcal{T} \cup \{\underline{0}\} \cup \mathcal{C} \cup \{\overline{0}\}$
- $\mathcal{A}_{\mathsf{P}} = \mathcal{A} \cup \mathcal{A}_1 \cup \mathcal{A}_2$

Pricing Problem Multigraph ($\mathcal{A}_3$)

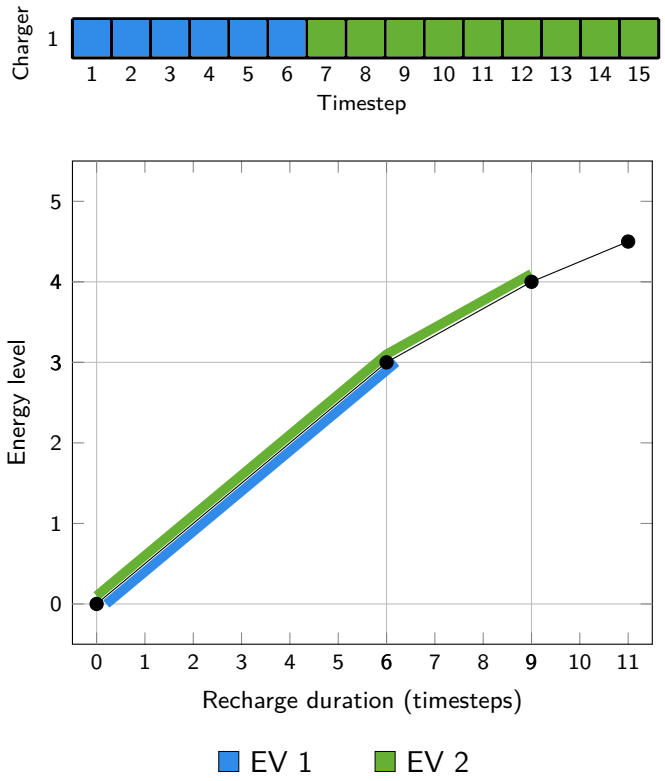


- $\mathcal{V}_{\mathsf{P}} = \{s\} \cup \mathcal{T} \cup \{\underline{0}\} \cup \mathcal{C} \cup \{\overline{0}\}$
- $\mathcal{A}_{\mathsf{P}} = \mathcal{A} \cup \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$

Illustrative Example



(a) Routing.



(b) Charging scheduling.

- $Q = 10$, $E = 5$, and $B = 1$
- $q_i = 5, i \in \mathcal{C}$

Pricing Problem Multigraph

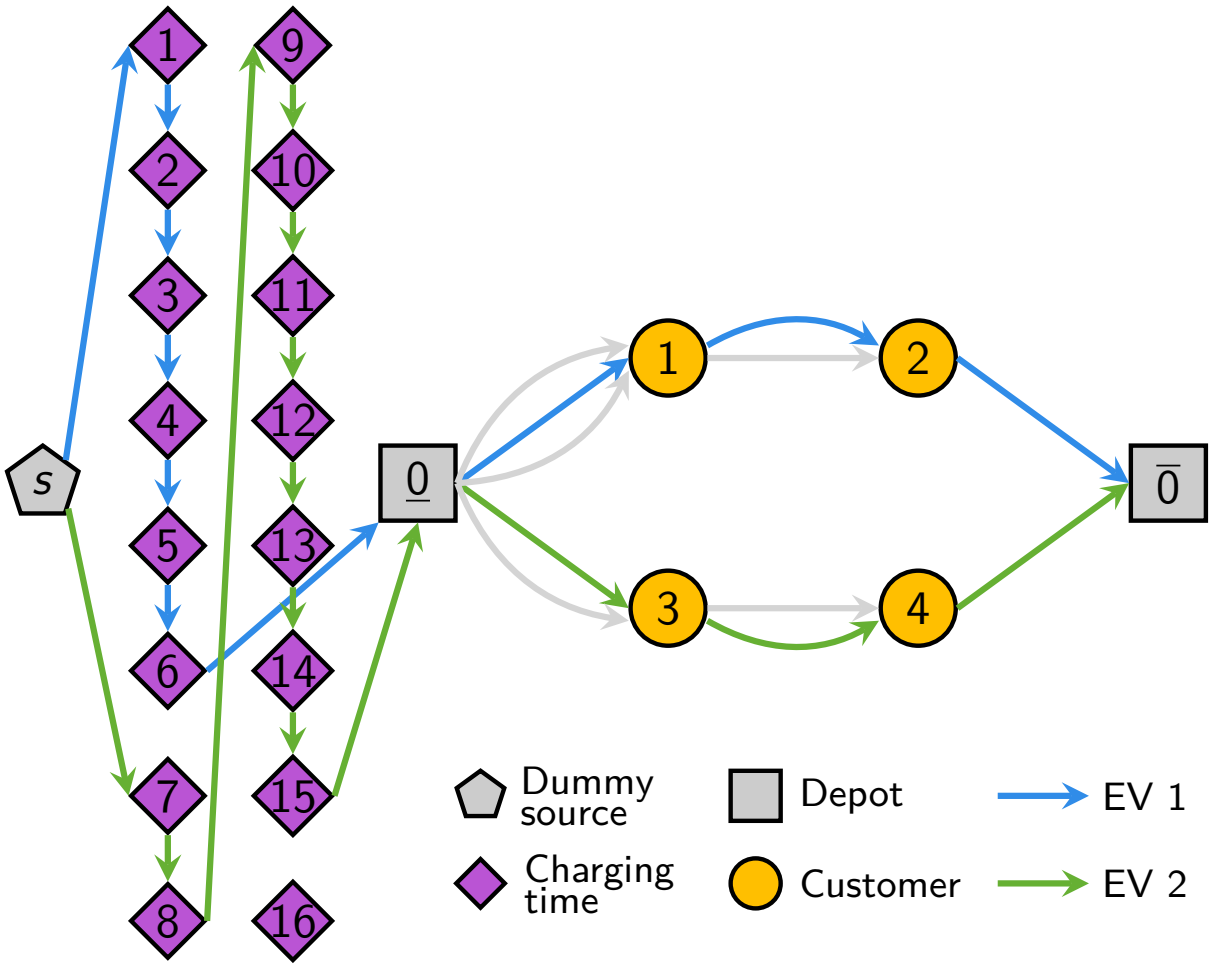


Figure 6: Illustrative pricing problem multigraph.

Pricing Problem Multigraph ($\mathcal{G}_\mathsf{P}$)

Multigraph $\mathcal{G}_\mathsf{P} = (\mathcal{V}_\mathsf{P}, \mathcal{A}_\mathsf{P})$

Vertex set $\mathcal{V}_\mathsf{P} = \{s\} \cup \mathcal{T} \cup \{\underline{0}\} \cup \mathcal{C} \cup \{\overline{0}\}$

Arc sets $\mathcal{A}_\mathsf{P} = \mathcal{A} \cup \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$

- **Modified arc cost**: for all $(i,j)^k \in \mathcal{A}_\mathsf{P}$

$$r_{(i,j)^k} = \begin{cases} c_{(i,j)^k} - \alpha_i, & (i,j)^k \in \mathcal{A}, \\ -\beta_i, & (i,j)^k \in \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3, \end{cases} \tag{2}$$

where $\alpha_{\underline{0}} = 0$ and $\beta_s = 0$ for notation conciseness

ng-SPPRC

A **path** from s to $\overline{0}$ passing through $\underline{0}$ defines a **route (and its charging schedule)**

We solve the ***ng*-SPPRC** as the PP (Baldacci et al., 2011)

ng-SPPRC

A **path** from s to $\overline{0}$ passing through $\underline{0}$ defines a **route (and its charging schedule)**

We solve the ***ng*-SPPRC** as the PP (Baldacci et al., 2011)

Backward labeling algorithm

- Starts at $\overline{0}$ at time $\overline{T}$
- Latest departure time from the depot (favoring more recharging time)
- Charging time required to perform the route

The Labeling Algorithm

A label ℓ_i encodes a **subpath** from $i \in \mathcal{V}_{\mathsf{P}}$ to $\overline{0}$:

- Cumulative **reduced cost** $r(\ell_i) \in \mathbb{R}$
- Remaining **load capacity** $q(\ell_i) \in \mathbb{Z}_+$
- Latest time for **starting service** $t(\ell_i) \in [\underline{T}, \overline{T}]$
- Remaining **energy capacity** $e(\ell_i) \in \mathbb{Z}_+$
- Number of **timesteps required to charge** $b(\ell_i) \in \mathbb{Z}_+$

The Labeling Algorithm

A label ℓ_i encodes a **subpath** from $i \in \mathcal{V}_{\mathsf{P}}$ to $\overline{0}$:

- Cumulative **reduced cost** $r(\ell_i) \in \mathbb{R}$
- Remaining **load capacity** $q(\ell_i) \in \mathbb{Z}_+$
- Latest time for **starting service** $t(\ell_i) \in [\underline{T}, \overline{T}]$
- Remaining **energy capacity** $e(\ell_i) \in \mathbb{Z}_+$
- Number of **timesteps required to charge** $b(\ell_i) \in \mathbb{Z}_+$
- A resource $\mathbb{I}_c^u(\ell_i) \in \{0, 1\}$ indicating whether customer $c \in \mathcal{C}$ is **unreachable**
- A resource $\mathbb{I}_c^{ng}(\ell_i) \in \{0, 1\}$ indicating whether visiting customer $c \in \mathcal{C}$ violates the *ng*-path **cycling restrictions**

Dominance Rules

Definition

In a **customer vertex** $i \in \mathcal{C}$, a label ℓ_i **dominates** another label ℓ'_i if the following conditions hold:

$$r(\ell_i) \leq r(\ell'_i), \tag{3a}$$
$$q(\ell_i) \geq q(\ell'_i), \tag{3b}$$
$$t(\ell_i) \geq t(\ell'_i), \tag{3c}$$
$$e(\ell_i) \geq e(\ell'_i), \tag{3d}$$
$$\max\{\mathbb{I}_c^u(\ell_i), \mathbb{I}_c^{ng}(\ell_i)\} \leq \max\{\mathbb{I}_c^u(\ell'_i), \mathbb{I}_c^{ng}(\ell'_i)\} \quad \forall c \in \mathcal{C}. \tag{3e}$$

Dominance Rules

Definition

In a **charging time vertex** $i \in \mathcal{T}$, a label ℓ_i **dominates** another label ℓ'_i if the following conditions hold:

$$r(\ell_i) \leq r(\ell'_i), \tag{4a}$$

$$b(\ell_i) \leq b(\ell'_i). \tag{4b}$$

Valid Inequalities

- **Rounded capacity inequality** (at the root node):

$$\sum_{r \in \mathcal{R}} \lambda_r \geq \left\lceil \sum_{i \in \mathcal{C}} q_i / Q \right\rceil$$

Valid Inequalities

- **Rounded capacity inequality** (at the root node):

$$\sum_{r \in \mathcal{R}} \lambda_r \geq \left\lceil \sum_{i \in \mathcal{C}} q_i / Q \right\rceil$$

- MP is (**dynamically**) strengthened with **subset-row cuts (SRCs)** defined over $S \subseteq \mathcal{C}, |S| = 3$ (Jepsen et al., 2008):

$$\sum_{r \in \mathcal{R}} \left\lfloor \frac{1}{2} \sum_{i \in S} a_i^r \right\rfloor \lambda_r \leq 1$$

Branching Rules

Hierarchical branching:

1. Number of vehicles used
2. Arc-flow variables
3. Consecutive flow through the depot source vertex

Branching Rules

Hierarchical branching:

1. Number of vehicles used
2. Arc-flow variables
3. Consecutive flow through the depot source vertex

Branching on **arc-flow variables** $x_{(i,j)^k}, (i,j)^k \in \mathcal{A}$:

- The branching is performed **locally**:

$$x_{(i,j)^k} = 0 \text{ and } x_{(i,j)^k} = 1$$

- We **remove routes** in $\mathcal{R}'$ that do not respect the imposed decisions
- We **remove arcs** in $\mathcal{A}_{\mathrm{P}}$ to forbid generating routes that do not respect the imposed decisions

Branching Rules

Branching on **arc-flow variables** $x_{(i,j)^1}, (i,j)^1 \in \mathcal{A}_1 \cup \mathcal{A}_3$:

- The branching is performed **globally**:

$$x_{(i,j)^1} \leq \lfloor \widetilde{B} \rfloor \text{ and } x_{(i,j)^1} \geq \lceil \widetilde{B} \rceil$$

- The corresponding **dual variable** is subtracted in $r_{(i,j)^1}$

Branching Rules

Branching on **arc-flow variables** $x_{(i,j)^1}, (i,j)^1 \in \mathcal{A}_1 \cup \mathcal{A}_3$:

- The branching is performed **globally**:

$$x_{(i,j)^1} \leq \lfloor \widetilde{B} \rfloor \text{ and } x_{(i,j)^1} \geq \lceil \widetilde{B} \rceil$$

- The corresponding **dual variable** is subtracted in $r_{(i,j)^1}$

Remarks

- Flow through arcs in $\mathcal{A}, \mathcal{A}_1$, and $\mathcal{A}_3$ is already an integer (flow conservation)
- Branching on arcs in $\mathcal{A}_3$: dual variable may penalize or reward the labels' reduced cost, depending on their charging start time
- Labels need to be extended even after they have completed their charge

Branching Rules

Current branching decisions:

- Guarantee **integer arc flows**
- But not necessarily guarantee **integer path flows**
- **Hint:** routes visiting the same customer arcs but with different charging schedules

Branching Rules

When none of the above decisions can be imposed, there exist two arcs, $(\underline{0}, j)^k \in \mathcal{A}$ and $(t, \underline{0})^1 \in \mathcal{A}_1$, such that their **consecutive flow** $m_{(\underline{0},j)^k,(t,\underline{0})^1}$ is fractional

$$m_{(\underline{0},j)^k,(t,\underline{0})^1} = \sum_{r \in \mathcal{R}} a^r_{(\underline{0},j)^k} \cdot b^r_t \cdot (1 - b^r_{t+1}) \cdot \lambda_r \tag{5}$$

Branching Rules

When none of the above decisions can be imposed, there exist two arcs, $(\underline{0}, j)^k \in \mathcal{A}$ and $(t, \underline{0})^1 \in \mathcal{A}_1$, such that their **consecutive flow** $m_{(\underline{0},j)^k,(t,\underline{0})^1}$ is fractional

$$m_{(\underline{0},j)^k,(t,\underline{0})^1} = \sum_{r \in \mathcal{R}} a^r_{(\underline{0},j)^k} \cdot b^r_t \cdot (1 - b^r_{t+1}) \cdot \lambda_r \tag{5}$$

Branching on the **consecutive flow** through the depot source vertex $\underline{0}$:

- The branching is performed **locally** (Desaulniers, 2010):

$$m_{(\underline{0},j)^k,(t,\underline{0})^1} = 0 \text{ and } m_{(\underline{0},j)^k,(t,\underline{0})^1} = 1$$

- We **remove routes** in $\mathcal{R}'$ that do not respect the imposed decisions
- The **labeling algorithm** is modified

Branching Rules

When none of the above decisions can be imposed, there exist two arcs, $(\underline{0}, j)^k \in \mathcal{A}$ and $(t, \underline{0})^1 \in \mathcal{A}_1$, such that their **consecutive flow** $m_{(\underline{0},j)^k,(t,\underline{0})^1}$ is fractional

$$m_{(\underline{0},j)^k,(t,\underline{0})^1} = \sum_{r \in \mathcal{R}} a^r_{(\underline{0},j)^k} \cdot b^r_t \cdot (1 - b^r_{t+1}) \cdot \lambda_r \tag{5}$$

Branching on the **consecutive flow** through the depot source vertex $\underline{0}$:

- The branching is performed **locally** (Desaulniers, 2010):

$$m_{(\underline{0},j)^k,(t,\underline{0})^1} = 0 \text{ and } m_{(\underline{0},j)^k,(t,\underline{0})^1} = 1$$

- We **remove routes** in $\mathcal{R}'$ that do not respect the imposed decisions
- The **labeling algorithm** is modified

It was **unnecessary** for our computational experiments

Table of Contents

Computational Results

- **Java OR library** (`jORLib`) and the **Java graph theory library** (`JGraphT`)
- **ILOG CPLEX solver** (version 22.1) for solving the RMPs
- Standard PC with an **Intel Core i7-8665U CPU @ 1.90GHz with 16GB of RAM** allocated to the memory heap size
- **Two-hour** computational time limit

Test Instances

- **112 instances** based on the well-known VRPTW benchmark of Solomon (1987)
- **Peugeot iOns**: $E = 16$ kilowatt-hours (kWh)
- **Minutes** as the time unit of discretization and **kilometers** as the distance unit
- EVs travel at a **fixed speed** of 60 km/h
- **Min-cost (and min-time) alternative** $(i,j)^1 \in \mathcal{A}$: $c_{(i,j)^1}$ is the *Euclidean distance* and $e_{(i,j)^1} = c_{(i,j)^1} \cdot 0.175$ kWh/km
- **Min-energy alternative** $(i,j)^2 \in \mathcal{A}$: $c_{(i,j)^2}$ is equal to the *Manhattan distance* and $e_{(i,j)^2} = c_{(i,j)^2} \cdot 0.125$ kWh/km
- (**Hopefully**) Available soon at www.vrp-rep.org (Mendoza et al., 2014)

Performance of the BPC Algorithm

Instance	C	Avg. # of altern.	B	LB	Cuts	Nodes	Time			Vehicles used	UB	Time in use (%)	Chargers in use (%)	Time at capacity (%)
							Total (s)	Master (%)	Pricing (%)					
R1		1.64	3	487.4	13.3	3.2	1.73	12.17	71.89	6.0	488.3	94.92	72.96	49.60
C1	25	1.61	1	198.2	10.6	1.7	11.01	3.47	92.30	3.2	198.6	75.27	75.27	75.27
RC1		1.53	3	445.4	59.0	2.5	4.44	11.81	78.34	5.0	446.6	100.00	75.08	47.40
R2		1.66	1	431.6	10.4	6.3	2.87	8.77	63.80	4.8	431.6	74.31	74.31	74.31
C2	25	1.62	1	272.5	43.1	1.5	57.50	2.78	94.21	3.0	272.5	37.93	37.93	37.93
RC2		1.53	1	419.4	39.8	7.3	4.20	16.35	63.42	4.6	419.4	87.19	87.19	87.19
R1		1.63	6	853.2	92.6	21.0	201.26	7.40	87.68	10.0	857.6	100.00	72.87	40.43
C1	50	1.68	2	370.3	21.2	1.4	53.03	3.65	93.53	5.2	371.0	92.08	76.85	61.62
RC1		1.48	4	798.3	108.5	2.3	559.96	2.67	96.42	7.6	800.1	99.59	74.48	54.10
R2		1.65	2	753.9	78.4	15.5	129.21	14.33	76.85	8.1	755.0	86.79	70.75	54.70
C2	50	1.68	1	482.5	80.5	2.0	221.91	6.67	91.08	5.4	482.5	34.21	34.21	34.21
RC2		1.49	2	699.9	33.8	2.5	26.10	8.93	84.14	6.4	699.9	87.72	66.90	46.09

Table 1: Summary of the BPC algorithm's performance.

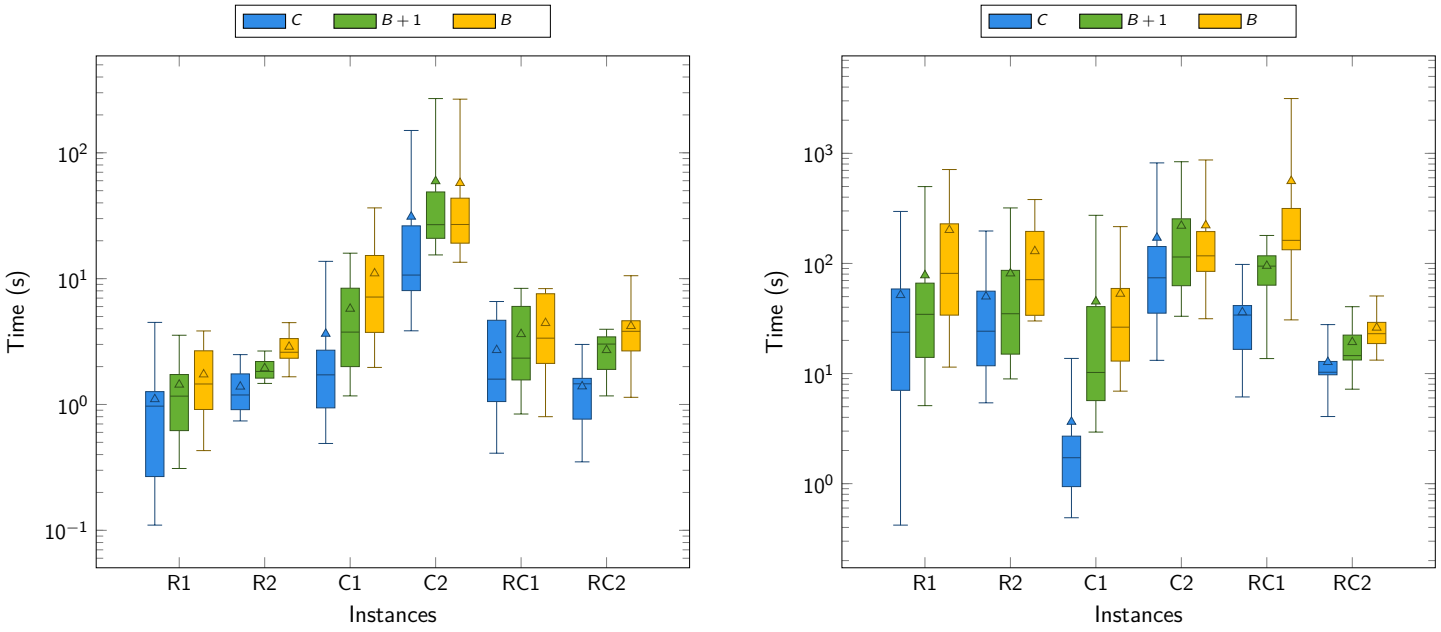
Performance of the BPC Algorithm

Instance	C	Avg. # of altern.	B	LB	Cuts	Nodes	Time			Vehicles used	UB	Time in use (%)	Chargers in use (%)	Time at capacity (%)
							Total (s)	Master (%)	Pricing (%)					
R1		1.64	3	487.4	13.3	3.2	1.73	12.17	71.89	6.0	488.3	94.92	72.96	49.60
C1	25	1.61	1	198.2	10.6	1.7	11.01	3.47	92.30	3.2	198.6	75.27	75.27	75.27
RC1		1.53	3	445.4	59.0	2.5	4.44	11.81	78.34	5.0	446.6	100.00	75.08	47.40
R2		1.66	1	431.6	10.4	6.3	2.87	8.77	63.80	4.8	431.6	74.31	74.31	74.31
C2	25	1.62	1	272.5	43.1	1.5	57.50	2.78	94.21	3.0	272.5	37.93	37.93	37.93
RC2		1.53	1	419.4	39.8	7.3	4.20	16.35	63.42	4.6	419.4	87.19	87.19	87.19
R1		1.63	6	853.2	92.6	21.0	201.26	7.40	87.68	10.0	857.6	100.00	72.87	40.43
C1	50	1.68	2	370.3	21.2	1.4	53.03	3.65	93.53	5.2	371.0	92.08	76.85	61.62
RC1		1.48	4	798.3	108.5	2.3	559.96	2.67	96.42	7.6	800.1	99.59	74.48	54.10
R2		1.65	2	753.9	78.4	15.5	129.21	14.33	76.85	8.1	755.0	86.79	70.75	54.70
C2	50	1.68	1	482.5	80.5	2.0	221.91	6.67	91.08	5.4	482.5	34.21	34.21	34.21
RC2		1.49	2	699.9	33.8	2.5	26.10	8.93	84.14	6.4	699.9	87.72	66.90	46.09

Table 1: Summary of the BPC algorithm's performance.

- **Maximum integrality** gap 1.86%!
- Relatively **small BB trees** (yet, enforcing integrality can be harder than in other VRPs)
- **Tight** charging schedules
- Solves **50-customer** instances

Impact of Charging Scheduling on Computational Time

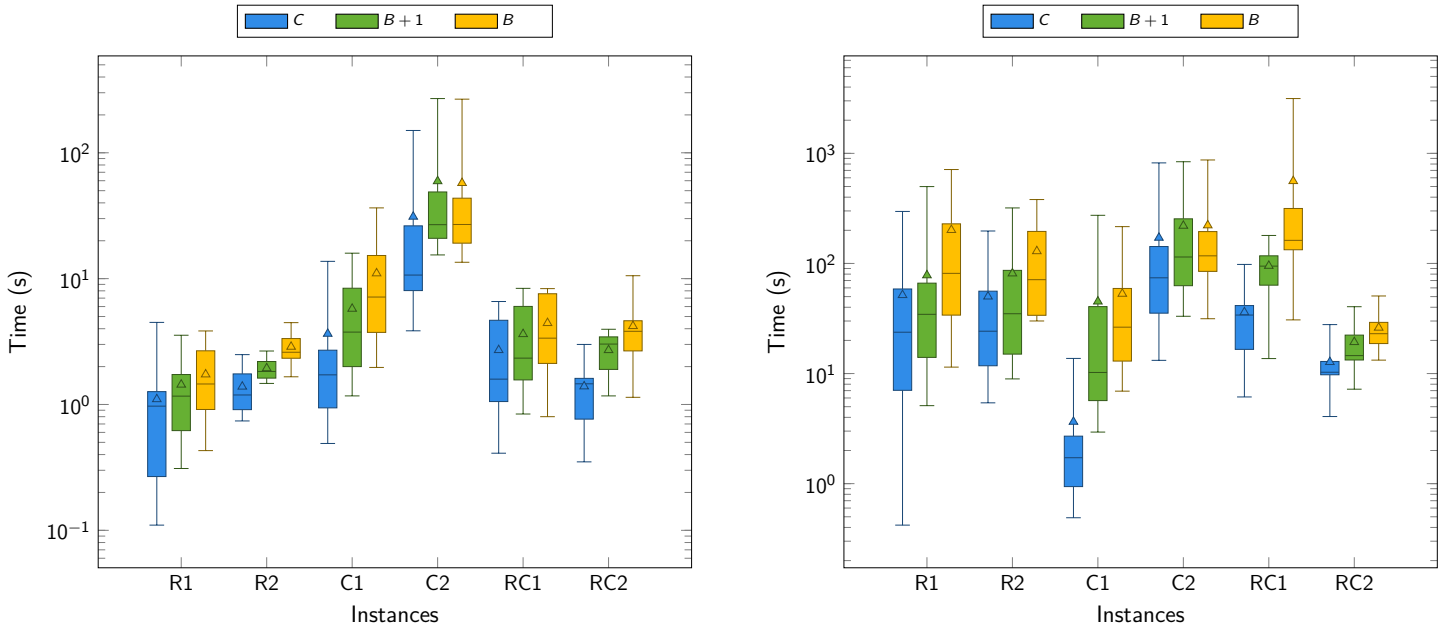


(a) 25-customer instances.

(b) 50-customer instances.

Figure 7: Impact of charging scheduling on computational time.

Impact of Charging Scheduling on Computational Time



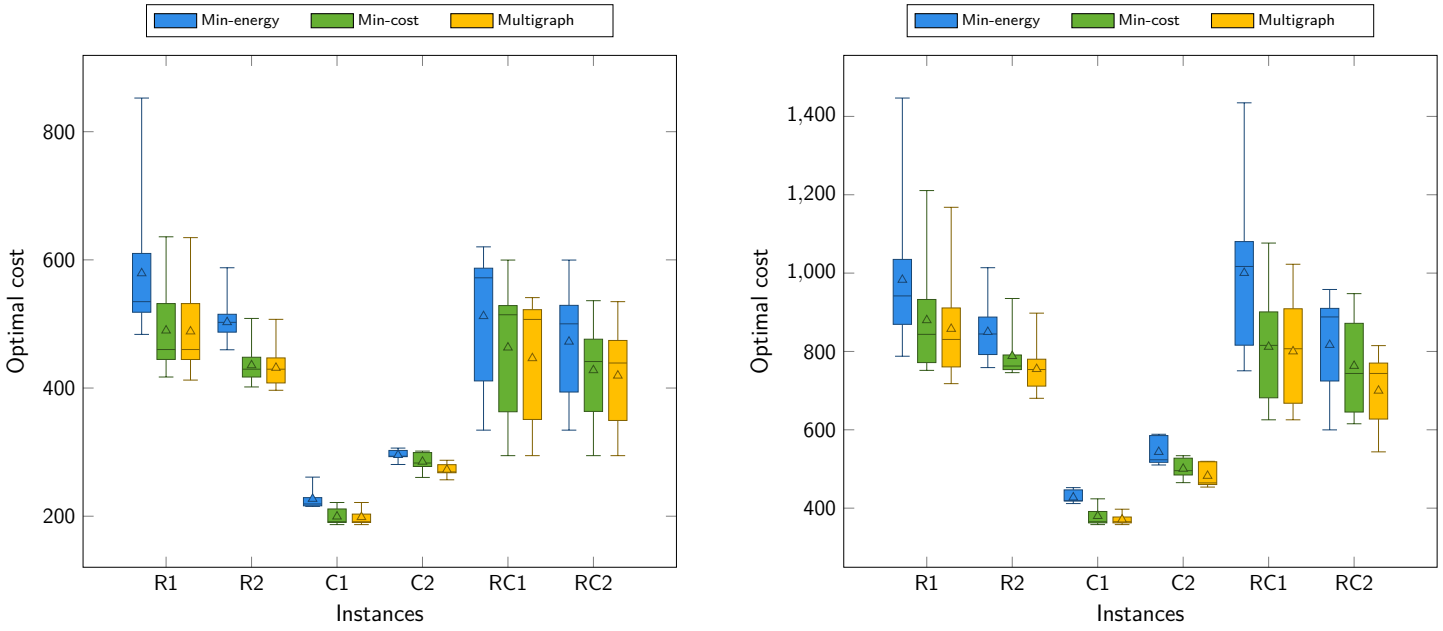
(a) **25-customer instances.**

(b) **50-customer instances.**

Figure 7: Impact of charging scheduling on computational time.

- Highly **variable** running times
- C chargers almost **3 times faster** than B chargers
- $B + 1$ chargers over **1.5 times faster** than B chargers

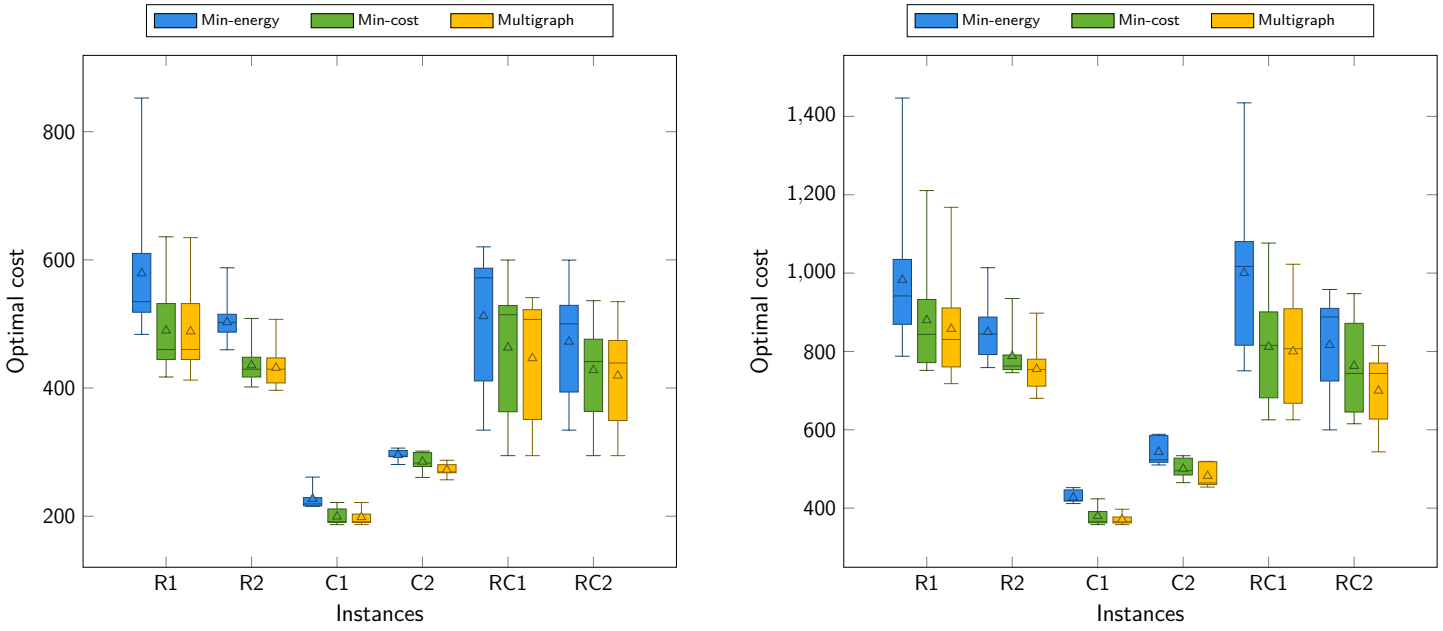
Impact of Multigraph Representation on Optimal Cost



(a) 25-customer instances. **(b) 50-customer instances.**

Figure 8: Impact of multigraph representation on optimal cost.

Impact of Multigraph Representation on Optimal Cost



(a) **25-customer instances.**

(b) **50-customer instances.**

Figure 8: Impact of multigraph representation on optimal cost.

- **Avg savings:** 12.96% (min-energy) and 2.87% (min-cost)
- **2.26 and 1.73 times faster** on min-energy and min-cost

Table of Contents

Conclusion

An **E-VRP** with piecewise-linear recharging and limited chargers

- EVs are charged at the depot
- Trade-off between conflicting resources

Conclusion

An **E-VRP** with piecewise-linear recharging and limited chargers

- EVs are charged at the depot
- Trade-off between conflicting resources

BPC algorithm that implements state-of-the-art techniques

- MP: capacity constraints of the depot's chargers
- PP: non-preemptive charge of the EVs
- *ng*-path relaxation, SRCs, and a specialized labeling algorithm
- Problem-specific dominance and branching rules

Conclusion

An **E-VRP** with piecewise-linear recharging and limited chargers

- EVs are charged at the depot
- Trade-off between conflicting resources

BPC algorithm that implements state-of-the-art techniques

- MP: capacity constraints of the depot's chargers
- PP: non-preemptive charge of the EVs
- *ng*-path relaxation, SRCs, and a specialized labeling algorithm
- Problem-specific dominance and branching rules

From a **computational perspective**

- Solves to optimality instances with 50 customers
- Impact of charging scheduling on computational time and multigraph representation on optimal cost

Conclusion

An **E-VRP** with piecewise-linear recharging and limited chargers

- EVs are charged at the depot
- Trade-off between conflicting resources

BPC algorithm that implements state-of-the-art techniques

- MP: capacity constraints of the depot's chargers
- PP: non-preemptive charge of the EVs
- *ng*-path relaxation, SRCs, and a specialized labeling algorithm
- Problem-specific dominance and branching rules

From a **computational perspective**

- Solves to optimality instances with 50 customers
- Impact of charging scheduling on computational time and multigraph representation on optimal cost

Thank you! Questions?

References I

Baldacci, R., Mingozzi, A., & Roberti, R. (2011). New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, *59*(5), 1269–1283.

Desaulniers, G. (2010). Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations Research*, *58*(1), 179–192.

Jepsen, M., Petersen, B., Spoorendonk, S., & Pisinger, D. (2008). Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, *56*(2), 497–511.

Lam, E., Desaulniers, G., & Stuckey, P. J. (2022). Branch-and-cut-and-price for the Electric Vehicle Routing Problem with Time Windows, Piecewise-Linear Recharging and Capacitated Recharging Stations. *Computers & Operations Research*, *145*, 105870.

References II

- Mendoza, J. E., Guéret, C., Hoskins, M., Lobit, H., Pillac, V., Vidal, T., & Vigo, D. (2014). VRP-REP: A vehicle routing community repository, in 'third meeting of the euro working group on vehicle routing and logistics optimization (VeRoLog). *Oslo (Norway)*, *56*, 77.
- Montoya, A., Guéret, C., Mendoza, J. E., & Villegas, J. G. (2017). The electric vehicle routing problem with nonlinear charging function. *Transportation Research Part B: Methodological*, *103*, 87–110.
- Morganti, E., & Browne, M. (2018). Technical and operational obstacles to the adoption of electric vans in France and the UK: An operator perspective. *Transport Policy*, *63*, 90–97.
- Pelletier, S., Jabali, O., Laporte, G., & Veneroni, M. (2017). Battery degradation and behaviour for electric vehicles: Review and numerical analyses of several models. *Transportation Research Part B: Methodological*, *103*, 158–187.

References III

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, *35*(2), 254–265.

Why Study This Problem?

Goods distribution in urban city centers: internal combustion engine vehicles (ICEVs) to commercial EVs (Pelletier et al., 2017)

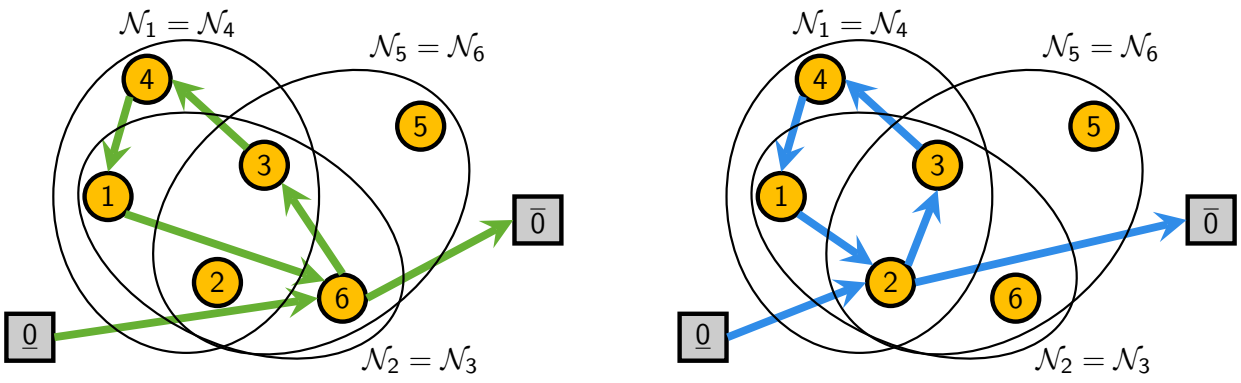
EVs alleviate some of the **unsustainable practices of present-day logistics**, but still have significant disadvantages over ICEVs

Researchers have investigated **two broad approaches**

1. Plan **recharging stops** along a route
2. Designing routes that can be completed with a **full battery load**

Operators prefer charging the fleet at **their facilities and overnight** (Morganti and Browne, 2018)

ng-path Relaxation



(a) **Feasible *ng*-path.** **(b)** **Infeasible *ng*-path.**

Figure 9: ***ng*-path relaxation.**

- Each customer $i \in \mathcal{C}$ has a neighborhood $\mathcal{N}_i \subseteq \mathcal{C}$ containing the $\Delta \in \mathbb{Z}_+$ closest customers to i (in terms of distance, time, or energy) and customer i itself.
- A *ng*-path allows a cycle starting and ending at vertex $j \in \mathcal{C}$ if and only if there exists a vertex $i \in \mathcal{C}$ in the cycle for which $j \notin \mathcal{N}_i$.
- Therefore, such a cycle is prohibited if and only if $j \in \mathcal{N}_i$ for every vertex $i \in \mathcal{C}$ it contains.

Label Extension

Algorithm 1: Label extension.

Input: ℓ_j, label representing a subpath starting at vertex j; $(i,j)^k \in \mathcal{A}_{\mathrm{P}}$, arc along which the extension is performed.

Output: ℓ_i, label representing a subpath starting at vertex i.

```
/* check whether the extension is feasible */
```

1 **if** $i \in \mathcal{C} \wedge \left(\mathbb{I}_i^u(\ell_j) = 1 \vee \mathbb{I}_i^{ng}(\ell_j) = 1\right)$ **then** `exit`

2 **if** $i \in \mathcal{T} \wedge j = \underline{0} \wedge \left(i < b(\ell_j) \vee i \geq t(\ell_j)\right)$ **then** `exit`

3 **if** $i = s \wedge b(\ell_j) > 0$ **then** `exit`

```
/* update the resource consumptions */
```

4 $r(\ell_i) \leftarrow r(\ell_j) + r_{(i,j)^k}$

5 $q(\ell_i) \leftarrow q(\ell_j) - q_i$

6 $t(\ell_i) \leftarrow \min\{t(\ell_j) - t_{(i,j)^k}, \bar{t}_i\}$

7 $e(\ell_i) \leftarrow e(\ell_j) - e_{(i,j)^k}$

8 **if** $i \in \mathcal{T}$ **then** $b(\ell_i) \leftarrow b(\ell_j) - 1$

9 **else** $b(\ell_i) \leftarrow \left\lceil f^{-1}\left(E - e(\ell_i)\right)\right\rceil$

```
/* check whether the extension is actually feasible */
```

10 **if** $t(\ell_i) < \underline{t}_i \vee e(\ell_i) < 0 \vee b(\ell_i) \geq t(\ell_i) \vee b(\ell_i) < 0$ **then** `exit`

```
/* mark customers that are unreachable by resource or cycling constraints */
```

11 **if** $i \in \mathcal{C}$ **then**

12 **for** $c \in \mathcal{C}$ **do**

```
/* unreachable customers */
```

13 $\mathbb{I}_c^u(\ell_i) \leftarrow \mathbb{I}_c^u(\ell_j)$

14 **if** $c \neq i \wedge \mathbb{I}_c^u(\ell_i) = 0 \wedge \left(q(\ell_i) - q_c < 0 \vee t(\ell_i) - \underline{t}_{(c,i)} < \underline{t}_c \vee e(\ell_i) - \underline{e}_{(c,i)} < 0\right)$ **then**

15 $\mathbb{I}_i^u(\ell_i) \leftarrow 1$

```
/* ng-path cycling restrictions */
```

16 **if** $c = i \vee \left(\mathbb{I}_c^{ng}(\ell_j) = 1 \wedge c \in \mathcal{N}_i\right)$ **then** $\mathbb{I}_c^{ng}(\ell_i) \leftarrow 1$

17 **else** $\mathbb{I}_c^{ng}(\ell_i) \leftarrow 0$

18 **return** ℓ_i