

May 16-19, 2023, Montréal, Canada

Strengthened route-based formulations and a branch-cut-andprice algorithm for split delivery vehicle routing problems



Outline

Split delivery vehicle routing problems;

Outline

Split delivery vehicle routing problems;

A new property and the flow graph;

Outline

- Split delivery vehicle routing problems;
- A new property and the flow graph;
- A new family of route formulations based on partial discretizations;

Outline

- Split delivery vehicle routing problems;
- A new property and the flow graph;
- A new family of route formulations based on partial discretizations;
- Computational results;

Outline

- Split delivery vehicle routing problems;
- A new property and the flow graph;
- A new family of route formulations based on partial discretizations;

- Computational results;
- Conclusions.

Split delivery vehicle routing problems

 $| d_i$: demand $| s_i$: service time $| [e_i, l_i]$: time window $| t_i$: travel time $| c_i$: travel cost |



Split delivery vehicle routing problems

 $| d_i$: demand $| s_i$: service time $| [e_i, l_i]$: time window $| t_i$: travel time $| c_i$: travel cost |



 A customer can be served by more than one vehicle, if necessary or beneficial (reductions of up to 50% in the cost);

Split delivery vehicle routing problems

 $| d_i$: demand $| s_i$: service time $| [e_i, l_i]$: time window $| t_i$: travel time $| c_i$: travel cost |



Additionally to defining the routes, we have to decide how much each vehicle delivers to each customer.

Split delivery vehicle routing problems

 $| d_i$: demand $| s_i$: service time $| [e_i, l_i]$: time window $| t_i$: travel time $| c_i$: travel cost |



Multiple visits to customers create modeling and algorithmic challenges: What are the actual routes here? (NP-hard!)

- Compact formulations:
 - Two-index vehicle flow formulations (arc-based variables only), can only provide *relaxations*: load transfer between vehicles;
 - The best performing compact models are based on three-index variables, in which the indices represent two subsequent arcs (not vehicle indexed) [Munari and Savelsbergh, 2022];

- Compact formulations:
 - Two-index vehicle flow formulations (arc-based variables only), can only provide *relaxations*: load transfer between vehicles;
 - The best performing compact models are based on three-index variables, in which the indices represent two subsequent arcs (not vehicle indexed) [Munari and Savelsbergh, 2022];
- Branch-and-cut methods (state-of-the-art so far)
 - They are based on two-index (relaxed) formulations [Archetti et al., 2014; Bianchessi and Irnich, 2019; Gouveia et al., 2023] and combinations of two- and three-index formulations [Munari and Savelsbergh, 2022];

- Branch(-cut)-and-price methods
 - Mullaseril and Dror (1996): CG-based approach based on the replication of customers;

- Branch(-cut)-and-price methods
 - Mullaseril and Dror (1996): CG-based approach based on the replication of customers;
 - Feillet et al. (2006): routes in the subproblem, delivery quantities in the master (huge number of columns and constraints);

- Branch(-cut)-and-price methods
 - Mullaseril and Dror (1996): CG-based approach based on the replication of customers;
 - Feillet et al. (2006): routes in the subproblem, delivery quantities in the master (huge number of columns and constraints);
 - Desaulniers (2010), Archetti, Bouchard, and Desaulniers (2011):
 - routes and extreme delivery patterns in the subproblem, delivery quantities as convex combinations in the master problems;
 - extreme delivery pattern: a visit delivers nothing, the full demand, or a fraction of the demand (but for at most one customer in the route);

- Branch(-cut)-and-price methods
 - Mullaseril and Dror (1996): CG-based approach based on the replication of customers;
 - Feillet et al. (2006): routes in the subproblem, delivery quantities in the master (huge number of columns and constraints);
 - Desaulniers (2010), Archetti, Bouchard, and Desaulniers (2011):
 - routes and extreme delivery patterns in the subproblem, delivery quantities as convex combinations in the master problems;
 - extreme delivery pattern: a visit delivers nothing, the full demand, or a fraction of the demand (but for at most one customer in the route);
 - Archetti, Bianchessi and Speranza (2011): addressed the SDVRP replicating each customer node d_i times;

- Branch(-cut)-and-price methods
 - Mullaseril and Dror (1996): CG-based approach based on the replication of customers;
 - Feillet et al. (2006): routes in the subproblem, delivery quantities in the master (huge number of columns and constraints);
 - Desaulniers (2010), Archetti, Bouchard, and Desaulniers (2011):
 - routes and extreme delivery patterns in the subproblem, delivery quantities as convex combinations in the master problems;
 - extreme delivery pattern: a visit delivers nothing, the full demand, or a fraction of the demand (but for at most one customer in the route);
 - Archetti, Bianchessi and Speranza (2011): addressed the SDVRP replicating each customer node d_i times;
 - Ceselli et al. (2009) and Moreno et al. (2010): path-based formulations for the SDVRP.

Properties

Properties

For a feasible instance with integer demands and vehicle capacity as well as costs and travel times that satisfy the triangle inequality, there exists an optimal solution with the following properties [(Dror and Trudeau, 1990; Feillet et al. 2006; Desaulniers 2010; Archetti, Bouchard and Desaulniers, 2011]:

1. Two routes have at most one customer in common;

Properties

- 1. Two routes have at most one customer in common;
- 2. Routes are elementary, i.e., a route never visits a node more than once;

Properties

- 1. Two routes have at most one customer in common;
- 2. Routes are elementary, i.e., a route never visits a node more than once;
- 3. Arcs between customers are traversed at most once;

Properties

- 1. Two routes have at most one customer in common;
- 2. Routes are elementary, i.e., a route never visits a node more than once;
- 3. Arcs between customers are traversed at most once;
- 4. If arc (i,j) is traversed, then arc (j,i) is not;

Properties

- 1. Two routes have at most one customer in common;
- 2. Routes are elementary, i.e., a route never visits a node more than once;
- 3. Arcs between customers are traversed at most once;
- 4. If arc (i,j) is traversed, then arc (j,i) is not; and
- 5. Delivery quantities are positive integers.

Strengthened route-based formulations and a branch-cut-and-price algorithm for SDVRPs

Pedro Munari [munari@dep.ufscar.br], Column Generation 2023, May 16-19, 2023 - Montreal, Canada

The flow graph $\mathfrak{F}(\tilde{\mathfrak{R}})$

C: set of customers;

The flow graph $\mathcal{F}(\tilde{\mathcal{R}})$

C: set of customers;

The flow graph $\mathfrak{F}(\tilde{\mathfrak{R}})$

C: set of customers;

•
$$\mathcal{F}(\tilde{\mathcal{R}}) = \{0\} \cup \tilde{\mathcal{R}} \cup \mathcal{C} \cup \{n+1\};$$

The flow graph $\mathcal{F}(\tilde{\mathcal{R}})$

C: set of customers;

•
$$\mathcal{F}(\tilde{\mathcal{R}}) = \{0\} \cup \tilde{\mathcal{R}} \cup \mathcal{C} \cup \{n+1\};$$

- ▶ A₁(Ĩ): connects the source with each of the route nodes in Ĩ. The capacity of these arcs is Q.
- A₂(𝔅): connects the route nodes in 𝔅 with customer nodes in 𝔅: arc (r, i) belongs to A₂(𝔅) if and only if route r ∈ 𝔅 visits customer i ∈ 𝔅. The capacity of these arcs is ∞.
- A₃(𝔅): connects each customer node to the sink. The capacity of an arc (*i*, *n* + 1) ∈ A₃(𝔅) is *d_i*.

The flow graph $\mathfrak{F}(\tilde{\mathfrak{R}})$



Strengthened route-based formulations and a branch-cut-and-price algorithm for SDVRPs

Pedro Munari [munari@dep.ufscar.br], Column Generation 2023, May 16-19, 2023 - Montreal, Canada

The flow graph $\mathfrak{F}(\tilde{\mathfrak{R}})$ - An example

•
$$\tilde{\mathfrak{R}} = \{r_1 = \{0, 1, 2, 3, 6\}, r_2 = \{0, 2, 3, 6\}, r_3 = \{0, 4, 5, 6\}\};$$

•
$$C = \{1, 2, 3, 4, 5\}, d = \{10, 20, 30, 40, 10\}, Q = 30.$$



The flow graph $\mathcal{F}(\tilde{\mathcal{R}})$ - An observation

Observation 1. A set ℜ of time-feasible routes forms a feasible SDVRPTW solution iff the maximum flow *f* in graph 𝔅(ℜ) has a value of ∑_{i∈𝔅} *d_i*. In such a case, values *f_a*, *a* ∈ 𝔅, correspond to the delivery quantities for every route in ℜ to each customer in 𝔅.

The flow graph $\mathcal{F}(\tilde{\mathcal{R}})$ - Checking feasibility

•
$$\tilde{\mathfrak{R}} = \{r_1 = \{0, 1, 2, 3, 6\}, r_2 = \{0, 2, 3, 6\}, r_3 = \{0, 4, 5, 6\}\};$$

• $C = \{1, 2, 3, 4, 5\}, d = \{10, 20, 30, 40, 10\}, Q = 30.$



• The max-flow in $\mathcal{F}(\tilde{\mathcal{R}})$ tells us if $\tilde{\mathcal{R}}$ forms a feasible solution.

The flow graph $\mathcal{F}(\tilde{\mathcal{R}})$ - Checking feasibility

- $\tilde{\mathbb{R}}^* = \{r_1 = \{0, 1, 2, 3, 6\}, r_2 = \{0, 2, 3, 6\}, r_3 = \{0, 4, 6\}, r_4 = \{0, 4, 5, 6\}\};$
- $C = \{1, 2, 3, 4, 5\}, d = \{10, 20, 30, 40, 10\}, Q = 30.$



• The max-flow in $\mathcal{F}(\tilde{\mathcal{R}})$ tells us if $\tilde{\mathcal{R}}$ forms a feasible solution.

The flow graph $\mathfrak{F}(\tilde{\mathfrak{R}})$ - Greatest common divisor

- All arc capacities are integer (other than ∞): there is an integer max-flow;
- Let $\bar{q} = gcd(Q, d_1, d_2, \dots, d_n)$: we divide all arc capacities by this value



A new property...

- 1. Two routes have at most one node in common;
- 2. Routes are elementary, i.e., a route never visits a node more than once;
- 3. Arcs between customers are traversed at most once;
- 4. If arc (i, j) is traversed, then arc (j, i) is not;
- 5. Delivery quantities are positive integers;

A new property...

- 1. Two routes have at most one node in common;
- 2. Routes are elementary, i.e., a route never visits a node more than once;
- 3. Arcs between customers are traversed at most once;
- 4. If arc (i, j) is traversed, then arc (j, i) is not;
- 5. Delivery quantities are positive integers; and
- 6. Let $\bar{q} = gcd(Q, d_1, d_2, \dots, d_n)$. All delivery quantities are multiples of \bar{q} .

Base formulation (F0)

 R: set of all feasible SDVRPTW routes satisfying Property 6; (it includes nonelementary routes.)
Base formulation (F0)

- R: set of all feasible SDVRPTW routes satisfying Property 6; (it includes nonelementary routes.)
- ▶ h_{rS} : binary parameter indicating if route *r* enters subset $S \subseteq C$;
- θ_r : integer variable indicating the number of vehicles that follows route *r*;

Base formulation (F0)

- R: set of all feasible SDVRPTW routes satisfying Property 6; (it includes nonelementary routes.)
- ▶ h_{rS} : binary parameter indicating if route *r* enters subset $S \subseteq C$;
- θ_r : integer variable indicating the number of vehicles that follows route *r*;

(F0): Min
$$\sum_{r \in \mathcal{R}} c^r \theta_r$$
, (1)
s.t. $\sum_{r \in \mathcal{R}} h_{rS} \theta_r \ge \left[\sum_{i \in S} d_i / Q \right]$, $\forall S \subseteq \mathcal{C}$, (2)
 $\theta_r \in \mathbb{Z}^+$, $\forall r \in \mathcal{R}$. (3)

Base formulation (F0)

- R: set of all feasible SDVRPTW routes satisfying Property 6; (it includes nonelementary routes.)
- ▶ h_{rS} : binary parameter indicating if route *r* enters subset $S \subseteq C$;
- θ_r : integer variable indicating the number of vehicles that follows route *r*;

F0): Min
$$\sum_{r \in \mathcal{R}} c^r \theta_r$$
, (1)
s.t. $\sum_{r \in \mathcal{R}} h_{rS} \theta_r \ge \left[\sum_{i \in S} d_i / Q \right]$, $\forall S \subseteq \mathcal{C}$, (2)
 $\theta_r \in \mathbb{Z}^+$, $\forall r \in \mathcal{R}$. (3)

 (2): strong k-path inequalities [Baldacci, Christofides and Mingozzi, 2008; Archetti, Bouchard and Desaulniers, 2011].

Base formulation (F0)

- Constraints (2) suffice to define the set of feasible solutions;
- ℜ(θ
 = {r₁, r₂, ..., r_{|ℜ(θ)|}}: set of routes corresponding to an integer solution θ;

Base formulation (F0)

- Constraints (2) suffice to define the set of feasible solutions;

Theorem 1. If an integer solution $\bar{\theta}$ satisfies Constraints (2), then set $\tilde{\mathbb{R}}(\bar{\theta})$ of routes forms a feasible SDVRPTW solution.

Base formulation (F0)

- Constraints (2) suffice to define the set of feasible solutions;

Theorem 1. If an integer solution $\bar{\theta}$ satisfies Constraints (2), then set $\tilde{\mathbb{R}}(\bar{\theta})$ of routes forms a feasible SDVRPTW solution.

Corolary 1. Constraints (2) can be exactly separated for integer solutions $\bar{\theta}$ of Formulation (F0) in polynomial time by using the flow graph $\mathcal{F}(\tilde{\mathcal{R}}(\bar{\theta}))$ and finding a minimum cut in it.

The flow graph $\mathcal{F}(\tilde{\mathcal{R}})$ - Checking feasibility

•
$$\tilde{\mathfrak{R}} = \{r_1 = \{0, 1, 2, 3, 6\}, r_2 = \{0, 2, 3, 6\}, r_3 = \{0, 4, 5, 6\}\};$$

• $C = \{1, 2, 3, 4, 5\}, d = \{10, 20, 30, 40, 10\}, Q = 30.$



• The max-flow in $\mathcal{F}(\tilde{\mathcal{R}})$ tells us if $\tilde{\mathcal{R}}$ forms a feasible solution.

Formulation (F1)

Formulation (F0) is correct even without any information about delivery quantities in routes r ∈ ℜ;

Formulation (F1)

- Formulation (F0) is correct even without any information about delivery quantities in routes r ∈ ℜ;
- Although it seems good, it may result in an infinite number of routes;

Formulation (F1)

- Formulation (F0) is correct even without any information about delivery quantities in routes r ∈ ℜ;
- Although it seems good, it may result in an infinite number of routes;
- Even when finite, Formulation (F0) is weak :(

Formulation (F1)

- Formulation (F0) is correct even without any information about delivery quantities in routes r ∈ ℜ;
- Although it seems good, it may result in an infinite number of routes;
- Even when finite, Formulation (F0) is weak :(

Formulation (F1):

- $\blacktriangleright D_i = \{\bar{q}, 2\bar{q}, \ldots, d_i\};$
- $\mathcal{R}^1 \subseteq \mathcal{R}$: set of routes that has length of at most Q/\bar{q} .

Formulation (F2)

But we are still missing information about the delivery quantities in the master problem!

Formulation (F2)

- But we are still missing information about the delivery quantities in the master problem!
- \blacktriangleright b_{iq}^r : number of times r makes a visit to i and delivers quantity q;

Formulation (F2)

- But we are still missing information about the delivery quantities in the master problem!
- \blacktriangleright b_{iq}^r : number of times r makes a visit to i and delivers quantity q;
- We start with Full and Partial deliveries: $b_{iF}^r = b_{id_i}^r$;

Formulation (F2)

- But we are still missing information about the delivery quantities in the master problem!
- \blacktriangleright b_{iq}^r : number of times r makes a visit to i and delivers quantity q;
- We start with Full and Partial deliveries: $b_{iF}^r = b_{id_i}^r$; $b_{iP}^r = \sum_{q \in D_i \setminus \{0, d_i\}} b_{iq}^r$.

Formulation (F2)

- But we are still missing information about the delivery quantities in the master problem!
- \blacktriangleright b_{iq}^r : number of times r makes a visit to i and delivers quantity q;

► We start with Full and Partial deliveries: $b_{iF}^r = b_{id_i}^r$; $b_{iP}^r = \sum_{q \in D_i \setminus \{0, d_i\}} b_{iq}^r$.

$$\begin{array}{ll} \text{(F2):} & \text{Min} & \sum_{r \in \mathcal{R}} c^r \theta_r, \\ & \text{s.t.} & \sum_{r \in \mathcal{R}} h_{rS} \theta_r \geq \left[\sum_{i \in S} d_i / \mathcal{Q} \right], & \forall S \subseteq \mathcal{C}, \\ & \sum_{r \in \mathcal{R}} (2b_{i\mathsf{F}}^r + b_{i\mathsf{P}}^r) \theta_r \geq 2, & \forall i \in \mathcal{C}. \\ & \theta_r \in \mathbb{Z}^+, & \forall r \in \mathcal{R}. \end{array}$$

Formulation (F2)

- But we are still missing information about the delivery quantities in the master problem!
- \blacktriangleright b_{iq}^r : number of times r makes a visit to i and delivers quantity q;

• We start with Full and Partial deliveries: $b_{iF}^r = b_{id_i}^r$; $b_{iP}^r = \sum_{q \in D_i \setminus \{0, d_i\}} b_{iq}^r$.

$$\begin{array}{lll} \text{(F2):} & \text{Min} & \sum_{r \in \mathcal{R}} c^r \theta_r, \\ & \text{s.t.} & \sum_{r \in \mathcal{R}} h_{rS} \theta_r \geq \left\lceil \sum_{i \in S} d_i / \mathcal{Q} \right\rceil, & \forall S \subseteq \mathcal{C}, \\ & & \sum_{r \in \mathcal{R}} (2b_{i\mathsf{F}}^r + b_{i\mathsf{P}}^r) \theta_r \geq 2, & \forall i \in \mathfrak{C}. \\ & & \theta_r \in \mathbb{Z}^+, & \forall r \in \mathcal{R}. \end{array}$$

 (4): special case of the strong minimum number of vehicles inequalities used by Archetti, Bouchard, and Desaulniers (2011).

A family of route-based formulations (FK)

From partial to full discretization: we extend the possible delivery quantities per customer to any integer $K \ge 2$;

A family of route-based formulations (FK)

- ► From partial to full discretization: we extend the possible delivery quantities per customer to any integer K ≥ 2;
- Increasing the value of K renders the formulation stronger, possibly at the expense of slower generation of route variables;

A family of route-based formulations (FK)

- From partial to full discretization: we extend the possible delivery quantities per customer to any integer $K \ge 2$;
- Increasing the value of K renders the formulation stronger, possibly at the expense of slower generation of route variables;
- Recall that $D_i = \{\bar{q}, 2\bar{q}, \dots, d_i\}$. The following constraints are valid:

$$\sum_{r \in \mathcal{R}} \sum_{q \in D_i} q b_{iq}^r \theta_r \ge d_i, \quad \forall i \in \mathcal{C}.$$
 (5)

A family of route-based formulations (FK)

- From partial to full discretization: we extend the possible delivery quantities per customer to any integer $K \ge 2$;
- Increasing the value of K renders the formulation stronger, possibly at the expense of slower generation of route variables;
- Recall that $D_i = \{\bar{q}, 2\bar{q}, \dots, d_i\}$. The following constraints are valid:

$$\sum_{r \in \mathfrak{R}} \sum_{q \in D_i} q b_{iq}^r \theta_r \ge d_i, \quad \forall i \in \mathfrak{C}.$$
 (5)

C(K) = {i ∈ C : Kq̄ < d_i}: contains each customer i ∈ C to which K deliveries of size q̄ are not enough to satisfy demand d_i;

A family of route-based formulations (FK)

- ► From partial to full discretization: we extend the possible delivery quantities per customer to any integer K ≥ 2;
- Increasing the value of K renders the formulation stronger, possibly at the expense of slower generation of route variables;
- Recall that $D_i = \{\bar{q}, 2\bar{q}, \dots, d_i\}$. The following constraints are valid:

$$\sum_{r \in \mathfrak{R}} \sum_{q \in D_i} q b_{iq}^r \theta_r \ge d_i, \quad \forall i \in \mathfrak{C}.$$
 (5)

- C(K) = {i ∈ C : Kq̄ < d_i}: contains each customer i ∈ C to which K deliveries of size q̄ are not enough to satisfy demand d_i;
- For $i \in \mathcal{C}(K)$, we multiply (5) by $(K-1)/(d_i \epsilon)$, where $0 < \epsilon \ll \overline{q}$;

A family of route-based formulations (FK)

We apply Chvátal-Gomory rounding on both sides and obtain:

$$\sum_{r \in \mathcal{R}} \sum_{q \in D_i} \left\lceil \frac{(K-1)qb_{iq}^r}{d_i - \epsilon} \right\rceil \theta_r \ge \left\lceil \frac{(K-1)d_i}{d_i - \epsilon} \right\rceil.$$
(6)

A family of route-based formulations (FK)

We apply Chvátal-Gomory rounding on both sides and obtain:

$$\sum_{r \in \mathcal{R}} \sum_{q \in D_i} \left\lceil \frac{(K-1)qb_{iq}^r}{d_i - \epsilon} \right\rceil \theta_r \ge \left\lceil \frac{(K-1)d_i}{d_i - \epsilon} \right\rceil.$$
(6)

• Since $d_i/(d_i - \epsilon) \approx 1$ and b_{iq}^r is integer:

$$\sum_{r \in \mathcal{R}} \sum_{q \in D_i} b_{iq}^r \left\lceil \frac{(K-1)q}{d_i - \epsilon} \right\rceil \theta_r \ge K.$$
(7)

A family of route-based formulations (FK)

We apply Chvátal-Gomory rounding on both sides and obtain:

$$\sum_{r \in \mathfrak{R}} \sum_{q \in D_i} \left\lceil \frac{(K-1)qb_{iq}^r}{d_i - \epsilon} \right\rceil \theta_r \ge \left\lceil \frac{(K-1)d_i}{d_i - \epsilon} \right\rceil.$$
(6)

• Since $d_i/(d_i - \epsilon) \approx 1$ and b_{iq}^r is integer:

$$\sum_{r \in \mathcal{R}} \sum_{q \in D_i} b_{iq}^r \left\lceil \frac{(K-1)q}{d_i - \epsilon} \right\rceil \theta_r \ge K.$$
(7)

► Given K and delivery quantity q ∈ D_i, the rounded-up coefficient is a step function that assumes integer values k from 1 to K;

A family of route-based formulations (FK)

We apply Chvátal-Gomory rounding on both sides and obtain:

$$\sum_{r \in \mathfrak{R}} \sum_{q \in D_i} \left\lceil \frac{(K-1)qb_{iq}^r}{d_i - \epsilon} \right\rceil \theta_r \ge \left\lceil \frac{(K-1)d_i}{d_i - \epsilon} \right\rceil.$$
(6)

Since $d_i/(d_i - \epsilon) \approx 1$ and b_{iq}^r is integer:

$$\sum_{r \in \mathcal{R}} \sum_{q \in D_i} b_{iq}^r \left\lceil \frac{(K-1)q}{d_i - \epsilon} \right\rceil \theta_r \ge K.$$
(7)

- ► Given K and delivery quantity q ∈ D_i, the rounded-up coefficient is a step function that assumes integer values k from 1 to K;
- For K = 2: 1, if $q < d_i$; and 2, if $q = d_i$ [the same as (4) used in (F2)];

A family of route-based formulations (FK)

We apply Chvátal-Gomory rounding on both sides and obtain:

$$\sum_{r \in \mathfrak{R}} \sum_{q \in D_i} \left\lceil \frac{(K-1)qb_{iq}^r}{d_i - \epsilon} \right\rceil \theta_r \ge \left\lceil \frac{(K-1)d_i}{d_i - \epsilon} \right\rceil.$$
(6)

Since $d_i/(d_i - \epsilon) \approx 1$ and b_{iq}^r is integer:

$$\sum_{r \in \mathcal{R}} \sum_{q \in D_i} b_{iq}^r \left\lceil \frac{(K-1)q}{d_i - \epsilon} \right\rceil \theta_r \ge K.$$
(7)

- ► Given K and delivery quantity q ∈ D_i, the rounded-up coefficient is a step function that assumes integer values k from 1 to K;
- For K = 2: 1, if $q < d_i$; and 2, if $q = d_i$ [the same as (4) used in (F2)];
- For K = 3: 1, if $q < d_i/2$; 2, if $d_i/2 \le q < d_i$; and 3, if $q = d_i$.

A family of route-based formulations (FK)

► For an arbitrary *K* and delivery quantity
$$q \in D_i$$
:

$$\left[\frac{(K-1)q}{(d_i - \varepsilon)}\right] = k, \text{ if } \frac{(k-1)d_i}{K-1} \le q < \frac{kd_i}{K-1}, \text{ for } k = 1, \dots, K;$$

A family of route-based formulations (FK)

► For an arbitrary *K* and delivery quantity
$$q \in D_i$$
:
 $\left\lceil \frac{(K-1)q}{(d_i - \varepsilon)} \right\rceil = k$, if $\frac{(k-1)d_i}{K-1} \le q < \frac{kd_i}{K-1}$, for $k = 1, \dots, K$;

► Using this observation, we define the binary parameter g_{iq}^k that assumes the value of 1 if and only if $\frac{(k-1)d_i}{K-1} \le q < \frac{kd_i}{K-1}$ and obtain:

$$\sum_{r \in \mathcal{R}} \sum_{q \in D_i} \sum_{k=1}^{K} b_{iq}^r g_{iq}^k k \theta_r \ge K.$$
(8)

A family of route-based formulations (FK)

► For an arbitrary *K* and delivery quantity
$$q \in D_i$$
:
 $\left[\frac{(K-1)q}{(d_i - \varepsilon)}\right] = k$, if $\frac{(k-1)d_i}{K-1} \le q < \frac{kd_i}{K-1}$, for $k = 1, \dots, K$;

► Using this observation, we define the binary parameter g^k_{iq} that assumes the value of 1 if and only if (k-1)d_i/K-1 ≤ q < kd_i/K-1 and obtain:

$$\sum_{r \in \mathcal{R}} \sum_{q \in D_i} \sum_{k=1}^{K} b_{iq}^r g_{iq}^k k \theta_r \ge K.$$
(8)

We defined the sets of possible delivery quantities to each customer:

$$D_i(K) = \bigcup_{k=1}^K \left\{ \min\{l\bar{q}\} : l\bar{q} \in \left[\frac{(k-1)d_i}{K-1}, \frac{kd_i}{K-1}\right), \ l \in \mathbb{N} \right\} \quad \text{if } i \in \mathcal{C}(K),$$

A family of route-based formulations (FK)

► For an arbitrary *K* and delivery quantity
$$q \in D_i$$
:
 $\left[\frac{(K-1)q}{(d_i - \varepsilon)}\right] = k$, if $\frac{(k-1)d_i}{K-1} \le q < \frac{kd_i}{K-1}$, for $k = 1, \dots, K$;

► Using this observation, we define the binary parameter g^k_{iq} that assumes the value of 1 if and only if (k-1)d_i/K-1 ≤ q < kd_i/K-1 and obtain:

$$\sum_{r \in \mathcal{R}} \sum_{q \in D_i} \sum_{k=1}^{K} b_{iq}^r g_{iq}^k k \theta_r \ge K.$$
(8)

We defined the sets of possible delivery quantities to each customer:

$$D_i(K) = \bigcup_{k=1}^K \left\{ \min\{l\bar{q}\} : l\bar{q} \in \left[\frac{(k-1)d_i}{K-1}, \frac{kd_i}{K-1}\right), l \in \mathbb{N} \right\} \quad \text{if } i \in \mathcal{C}(K),$$
$$D_i(K) = \{l\bar{q} : \forall l = 1, \dots, d_i/\bar{q}\} \quad \text{if } i \in \mathcal{C} \setminus \mathcal{C}(K).$$

A family of route-based formulations (FK)



A family of route-based formulations (FK)



A family of route-based formulations (FK)

$$\begin{array}{ll} (\mathsf{FK}): & \mathsf{Min} & \sum_{r \in \mathcal{R}} c^r \theta_r, \\ & \mathsf{s.t.} & \sum_{r \in \mathcal{R}} h_{rS} \theta_r \geq \left[\sum_{i \in S} d_i / \mathcal{Q} \right], & \forall S \subseteq \mathcal{C}, \\ & & \sum_{r \in \mathcal{R}} \sum_{q \in D_i} \sum_{k=1}^K b_{iq}^r g_{iq}^k k \theta_r \geq K, & \forall i \in \mathfrak{C}(K), \\ & & \sum_{r \in \mathcal{R}} \sum_{q \in D_i} q b_{iq}^r \theta_r \geq d_i, & \forall i \in \mathfrak{C} \setminus \mathfrak{C}(K), \\ & & \theta_r \in \mathbb{Z}^+, & \forall r \in \mathcal{R}. \end{array}$$

 $\blacktriangleright \ \mathcal{R}^{K} = \left\{ r \in \mathcal{R} : b_{iq}^{r} = 0, \ \forall i \in \mathcal{C}, \ \forall q \notin D_{i}(K) \right\};$

► There exists an optimal solution $\bar{\theta}$ to the linear relaxation of Formulation (FK) such that $\bar{\theta}_r = 0$ for all $r \notin \Re^K$.

22

A family of route-based formulations (FK)

For $K_{\max} = \max_{i \in \mathcal{C}} d_i / \bar{q}$, Formulation (F K_{\max}) is *fully discretized*: all possible delivery quantities from Property 6 are considered ($\mathcal{R}^K = \mathcal{R}$).

A family of route-based formulations (FK)

- For K_{max} = max_{i∈C} d_i/q̄, Formulation (FK_{max}) is *fully discretized*: all possible delivery quantities from Property 6 are considered (ℝ^K = ℝ).
 - Strong k-path inequalities are redundant for this formulation but still useful as cutting planes.
A family of route-based formulations (FK)

- For K_{max} = max_{i∈C} d_i/q̄, Formulation (FK_{max}) is *fully discretized*: all possible delivery quantities from Property 6 are considered (ℝ^K = ℝ).
 - Strong k-path inequalities are redundant for this formulation but still useful as cutting planes.
- Formulations (FK) with $K > K_{max}$ are equivalent to (F K_{max}).

A family of route-based formulations (FK)

- For K_{max} = max_{i∈C} d_i/q̄, Formulation (FK_{max}) is *fully discretized*: all possible delivery quantities from Property 6 are considered (ℜ^K = ℜ).
 - Strong k-path inequalities are redundant for this formulation but still useful as cutting planes.
- Formulations (FK) with $K > K_{max}$ are equivalent to (F K_{max}).
- Formulations (FK) with $K < K_{max}$ are partially discretized.

A family of route-based formulations (FK)

- For K_{max} = max_{i∈C} d_i/q̄, Formulation (FK_{max}) is *fully discretized*: all possible delivery quantities from Property 6 are considered (ℝ^K = ℝ).
 - Strong k-path inequalities are redundant for this formulation but still useful as cutting planes.
- Formulations (FK) with $K > K_{max}$ are equivalent to (F K_{max}).
- Formulations (FK) with $K < K_{max}$ are partially discretized.
 - The exact separation of strong k-path inequalities (2) is necessary to ensure the feasibility of integer solutions.

Branch-cut-and-price: Main components

▶ Rounded capacity inequalities (RCI); Limited-memory subset-row packing inequalities [valid for formulations (FK) with K ≥ 1]; Limited-memory subset-row covering inequalities; Limited-memory strong *k*-path inequalities;

- ▶ Rounded capacity inequalities (RCI); Limited-memory subset-row packing inequalities [valid for formulations (FK) with K ≥ 1]; Limited-memory subset-row covering inequalities; Limited-memory strong *k*-path inequalities;
- ▶ We branch on customer edges ≫ depot edges ≫ number of used vehicles. If none is fractional, we do Ryan&Foster branching [Desaulniers, 2010]. We use a multiphase strong branching procedure.

- ▶ Rounded capacity inequalities (RCI); Limited-memory subset-row packing inequalities [valid for formulations (FK) with K ≥ 1]; Limited-memory subset-row covering inequalities; Limited-memory strong *k*-path inequalities;
- ▶ We branch on customer edges ≫ depot edges ≫ number of used vehicles. If none is fractional, we do Ryan&Foster branching [Desaulniers, 2010]. We use a multiphase strong branching procedure.
- Automatic dual price smoothing stabilization [Pessoa et al., 2018];

- ▶ Rounded capacity inequalities (RCI); Limited-memory subset-row packing inequalities [valid for formulations (FK) with K ≥ 1]; Limited-memory subset-row covering inequalities; Limited-memory strong *k*-path inequalities;
- ▶ We branch on customer edges ≫ depot edges ≫ number of used vehicles. If none is fractional, we do Ryan&Foster branching [Desaulniers, 2010]. We use a multiphase strong branching procedure.
- Automatic dual price smoothing stabilization [Pessoa et al., 2018];
- Elementary route enumeration based on the primal-dual gap;

- ▶ Rounded capacity inequalities (RCI); Limited-memory subset-row packing inequalities [valid for formulations (FK) with K ≥ 1]; Limited-memory subset-row covering inequalities; Limited-memory strong *k*-path inequalities;
- ▶ We branch on customer edges ≫ depot edges ≫ number of used vehicles. If none is fractional, we do Ryan&Foster branching [Desaulniers, 2010]. We use a multiphase strong branching procedure.
- Automatic dual price smoothing stabilization [Pessoa et al., 2018];
- Elementary *route enumeration* based on the primal-dual gap;
- Before launching our BCP algorithm, we run a *matheuristic* and the value of the best solution (plus a small epsilon) is then used as the *initial upper bound* in the BCP algorithm.

Branch-cut-and-price: Pricing subproblem

- Resource constrained shortest path problem in multi-graph $G'(K) = (\mathcal{V}, \mathcal{A}'(K))$ (capacity and time-window constrained);
- ► Every arc (*i*, *j*) in the original graph G is replaced by multiple arcs (*i*, *j*, *q*), *q* ∈ *D_j*(*K*), between nodes *i* and *j* in G'(*K*);
- The resource-feasible paths in $\mathcal{G}'(K)$ are the routes in \mathcal{R}^K ;

Branch-cut-and-price: Pricing subproblem

- Resource constrained shortest path problem in multi-graph $G'(K) = (\mathcal{V}, \mathcal{A}'(K))$ (capacity and time-window constrained);
- ► Every arc (*i*, *j*) in the original graph G is replaced by multiple arcs (*i*, *j*, *q*), *q* ∈ *D_j*(*K*), between nodes *i* and *j* in G'(*K*);
- The resource-feasible paths in $\mathcal{G}'(K)$ are the routes in \mathcal{R}^K ;
- We use the bucket-graph based bidirectional labeling algorithm of Sadykov, Uchoa, and Pessoa (2021), with ng-path relaxation;
- Every label *L* represents a partial path $\mathfrak{G}'(K)$, which is either forward (starting from node 0) or backward (starting from node n + 1).

Branch-cut-and-price: Pricing subproblem

▶ The reduced cost $\bar{c}_{(i,j,q)}$ of each arc $(i,j,q) \in \mathcal{A}'(K)$ is

$$\bar{c}_{(i,j,q)} = c_{ij} - \sum_{\substack{o \in O:\\ (i,j) \in \delta(S^o)}} \bar{\rho}_o - \begin{cases} 0, & \text{if } K < 2, \\ \bar{k}(j,q,K) \cdot \bar{\pi}_j, & \text{if } K \ge 2 \text{ and } j \in \mathbb{C}(K), \\ q \bar{\pi}_j, & \text{if } K \ge 2 \text{ and } j \in \mathbb{C} \setminus \mathbb{C}(K). \end{cases}$$

- ▶ $\bar{k}(j,q,K)$ is equal to the value *k* that satisfies $\frac{(k-1)d_j}{K-1} \leq q < \frac{kd_j}{K-1}$;
- O: set of active RCIs, with S^o ⊆ C defining the rounded capacity inequality o ∈ O with dual value p̄_o > 0;
- Let $\delta(S^{\circ})$ also be the set of arcs in \mathcal{A} which have exactly one node in S° .

Computational results

- The BCP algorithm was coded in C++ on top of the generic BCP library BaPCod [Sadykov and Vanderbeck, 2021] with its VRPSolver extension [Pessoa et al., 2020];
- We use the IBM CPLEX Optimization Studio 20.1 as LP and MIP solver;
- Server with processor 2.6GHz Cascade Lake Intel Xeon Skylake Gold 6240 with 36 cores and 196 GB of RAM (up to 36 instances were run in parallel on each node, each run using a single core).

Computational results: Benchmark instances

► SDVRPTW: Adapted Solomon's VRPTW instances, having n = {50, 75, 100} and Q = {30, 50, 100} [total: 56 × 3 × 3 = 504];

Computational results: Benchmark instances

- ► SDVRPTW: Adapted Solomon's VRPTW instances, having n = {50, 75, 100} and Q = {30, 50, 100} [total: 56 × 3 × 3 = 504];

Computational results: Benchmark instances

- ► SDVRPTW: Adapted Solomon's VRPTW instances, having n = {50, 75, 100} and Q = {30, 50, 100} [total: 56 × 3 × 3 = 504];
- Thus, the total number of SDVRPTW instances is 705;

Computational results: Benchmark instances

- ► SDVRPTW: Adapted Solomon's VRPTW instances, having n = {50, 75, 100} and Q = {30, 50, 100} [total: 56 × 3 × 3 = 504];
- ▶ New instances based on those with $\bar{q} = 10$ (all instances in class C, and instances in class RC with 50 customers). We add a random integer value in [-3, 3] to the demand of every customer $i \in C$ such that $d_i \in [1.2d_{\min}, 0.8d_{\max}]$; We denote these new instances as 50P, 75P, and 100P [total: 201 instances].
- Thus, the total number of SDVRPTW instances is 705;
- SDVRP: We perform 352 tests derived from 88 instances (S, SD, eil, p), limiting, or not, the size of the fleet (LF/UF) and rounding, or not, distances (LF-r/UF-r) [total: 88 × 4 = 352].

Computational results: Comparing formulations (FK)

Root node results for SDVRPTW instances with n = 50





Computational results: Comparing valid inequalities

SDVRPTW – F2 and FK_{max} (one-hour time limit)

BCP variant	Opt	Geom. time(s)	Time(s)	Root gap(%)	Final gap(%)		
BCPmax _{all-SKPI-SRCI}	361	360.40	1837.16	0.87	0.72		
BCPmax _{all-SKPI}	358	362.56	1865.11	0.86	0.72		
BCPmax _{all-SRCI}	368	354.53	1820.09	0.84	0.70		
BCPmax _{all}	363	349.39	1828.03	0.82	0.69		
BCP2 _{all-SRCI}	273	734.92	2243.48	2.42	2.00		
BCP2 _{all}	273	730.12	2238.13	2.39	1.93		

- SKPI and SRCI contribute to reducing root and final gaps (on average).
- Strong k-path inequalities SKPI increase the number of optimally solved instances.
- Subset-row covering inequalities SRCI does not allow us to solve more instances to optimality.

Computational results: SDVRPTW

SDVRPTW – F2 and FK_{max}

n	Benchmark run – 3600s					Long run – 18000s				
	(FK_{\max})	MS22	BI19	ABD11		(F2)	(FK_{\max})	Best(F2, FK _{max})		
25	168	168	168	168		168	168	168 (0)		
50	152 (27)	123	104	86		136	168	168 (40)		
100	54 (48)	4	5	8		24	55	56 (50)		
{50, 100}	206	127	109	94		160	223	224 (90)		
{25, 50, 100}	374 (75)	295	277	262		328	391	392 (90)		

MS22: Munari and Savelsbergh (2022); BI19: Bianchessi and Irnich (2019); A11: Archetti, Bouchard and Desaulniers (2011).

- Formulation (*FK*_{max}) finds 374 optimal solutions, 75 for the first time, within one hour benchmark tests.
- Formulations (F2) and (FK_{max}) all together find 392 optimal solutions, 90 for the first time, within five hours.

Computational results: SDVRP

Three variants of our BCP algorithm on the full test set of SDVRP instances: BCP2_{all-SRCI}, BCP10_{all-SRCI} and BCPmax_{all-SRCI} (time limit of 2 hours);

Computational results: SDVRP

Three variants of our BCP algorithm on the full test set of SDVRP instances: BCP2_{all-SRCI}, BCP10_{all-SRCI} and BCPmax_{all-SRCI} (time limit of 2 hours);

Class #	#	LF-r		LF			UF-r			UF			
"		Opt	LB*	Opt*	Opt	LB*	Opt*	Opt	LB*	Opt*	Opt	LB*	Opt*
eil	11	8	5	2	7	6	2	9	5	3	7	6	2
р	42	3	16	1	2	9	0	2	16	0	2	7	0
S	14	3	2	0	4	1	0	3	5	0	4	2	0
SD	21	12	9	1	13	9	1	12	9	1	11	9	1
Total	88	26	32	4	26	25	3	26	35	4	24	24	3

Best overall performance: BCP10_{all-SRCI};

102 optimally solved, 14 for the first time.

We improve 116 best known LB's (approximately 50% of open instances).

Conclusions

- We proposed a new family of partially discretized route-based formulations (FK) for split delivery vehicle routing problems;
- K is the maximum number of different delivery quantities allowed when visiting a customer. In the fully discretized formulation (FK_{max}), all possible delivery quantities are considered;
- We propose a property that provides a minimum delivery quantity based on customer demand and vehicle capacity. This property is likely to benefit other formulations, as well as other exact and heuristic approaches in the literature;
- To effectively solve the formulations, we have designed a BCP algorithm that resorts to new and state-of-the-art algorithmic improvements.

Conclusions

- On SDVRPTW instances: the BCP based on (FK_{max}) achieves best overall performance; (F2) is more efficient for instances with long routes and large value K_{max};
- ▶ On SDVRP instances: (F10) is the best, and (FK_{max}) is the second best;
- The proposed formulations and BCP algorithm establish a new state-of-the-art for the SDVRPTW, and are highly competitive with the best approach in the literature for the SDVRP.
- Future research topics include trying different versions of the proposed formulations, possibly including delivery quantities explicitly in the master problem (e.g. using flow variables); using other families of valid inequalities; addressing related problem variants.





http://www.dep.ufscar.br/munari

Balster, I.; Bulhões, T.; Munari, P.; Sadykov, R. A new family of route formulations for split delivery vehicle routing problems. Technical Report 8918, Operations Research Group, Production Engineering Department, Federal University of São Carlos. 2022. http://www.optimization-online.org/DB_HTML/2022/05/8918.html