New route formulations for the Split-Delivery VRP

Ruslan Sadykov

Inria Centre at the University of Bordeaux

Joint work with Isaac Balster (Inria), Teobaldo Bulhôes (UFPB, Brazil), and Pedro Munari (UFSC, Brazil)

Workshop on the Inventory Routing Problem
Ecole des Ponts Paristech, May 20, 2022
From standard CVRP to SDVRP’s

Classic Capacitated Vehicle Routing Problem - CVRP

▶ Objective: minimise routing costs.

Split delivery variants – SDVRP’s

▶ The single visit requirement for customers is relaxed.
▶ Each client can now be visited by one or more vehicles.
Practical motivation

Instance (a) with $Q = 5$. The cost is 24 (with 3 vehicles) for the CVRP (b) and 18 (with 2 vehicles) for the SDVRP. Source: [Archetti and Speranza, 2012].

- Routing savings can reach up to 50% [Archetti et al., 2006].
Research motivation

CG-based approaches and BCP algorithms:

- Feillet, Dejax, Gendreau and Gueguen (2006)
- Moreno, De Aragão and Uchoa (2010)
- Desaulniers (2010)
- Archetti, Bouchard and Desaulniers (2011)
- Archetti, Bianchessi and Speranza (2011)
- Munari and Savelsbergh (2020)

BC algorithms (current state-of-the-art):

- Archetti, Bianchessi and Speranza (2014)
- Ozbaygin, Karasan and Yaman (2018)
- Bianchessi and Irnich (2019)
- Gouveia, Leitner and Ruthmair (2021)
- Munari and Savelsbergh (2022)
Research motivation

CG-based approaches and BCP algorithms:

Feillet, Dejax, Gendreau and Gueguen (2006)
Moreno, De Aragão and Uchoa (2010)
Desaulniers (2010)
Archetti, Bouchard and Desaulniers (2011)
Archetti, Bianchessi and Speranza (2011)
Munari and Savelsbergh (2020)

BC algorithms (current state-of-the-art):

Archetti, Bianchessi and Speranza (2014)
Ozbaygin, Karasan and Yaman (2018)
Bianchessi and Irnich (2019)
Gouveia, Leitner and Ruthmair (2021)
Munari and Savelsbergh (2022)

The SDVRP has a similar structure to the IRP!
Current BCP algorithms for the SDVRP’s

- Based on extreme delivery patterns [Desaulniers, 2010]
- Pricing problem is harder than the standard RCSPP
- To use the standard RCSPP solver [Sadykov et al., 2021], we need to discretize delivery quantities
- Are there route formulations which allow us to use the standard RCSPP solver without full discretization?
Base formulation for SDVRP’s

- \( \mathcal{C} \) — set of customers
- \( \mathcal{R} \) — set of elementary [and time-feasible] routes.
- \( c^r \) — cost of route \( r \in \mathcal{R} \)
- \( h_{rS} = 1 \) iff route \( r \in \mathcal{R} \) enters subset \( S \subseteq \mathcal{C} \) of customers
- \( \theta_r \) — number of vehicles which follow route \( r \in \mathcal{R} \) (variable)

\[(F0): \quad \text{Min} \quad \sum_{r \in \mathcal{R}} c^r \theta_r, \]

\[\text{s.t.} \quad \sum_{r \in \mathcal{R}} h_{rS} \theta_r \geq \left\lfloor \sum_{i \in S} \frac{d_i}{Q} \right\rfloor, \quad \forall S \subseteq \mathcal{C}, \]

\[\theta_r \in \mathbb{Z}_+, \quad \forall r \in \mathcal{R}. \]

- Constraints are strong \( k \)-path inequalities [Baldacci et al., 2008, Archetti et al., 2011].
- No information about delivery quantities in route variables!
Flow graph $\mathcal{F}(\tilde{\mathcal{R}})$ to show correctness of (F0)

$\tilde{\mathcal{R}}$ is the set of routes in the solution of (F0)
An example of flow graph $\mathcal{F}(\tilde{R})$

Customers $C = \{1, 2, 3, 4, 5\}$ with demands $d = \{10, 20, 30, 40, 10\}$, and vehicle capacity $Q = 30$.

$\tilde{R} = \left\{ r_1 = \{0, 1, 2, 3, 6\}, r_2 = \{0, 2, 3, 6\}, r_3 = \{0, 4, 5, 6\} \right\}$
Checking feasibility with $\mathcal{F}(\tilde{R})$

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

$$\text{max-flow} = 90 < \sum_{i \in C} d_i$$

$$\sum_{r \in \tilde{R}} h_{r,\{4,5\}} \theta_r (1) < \left\lceil \sum_{i \in \{4,5\}} d_i/Q \right\rceil \left\lceil 50/30 \right\rceil = 2$$

$\Rightarrow$ strong $k$-path inequality for $S = \{4,5\}$ is violated
Checking feasibility with $\mathcal{F}(\tilde{\mathcal{R}})$ (II)

$\tilde{\mathcal{R}}^* = \left\{ r_1 = \{0, 1, 2, 3, 6\}, r_2 = \{0, 2, 3, 6\}, r_3 = \{0, 4, 6\}, r_4 = \{0, 4, 5, 6\} \right\}$

flow-max $= 110 = \sum_{i \in C} d_i$
A dominance rule for optimal solutions

Divide arc capacities in $\mathcal{F}(\tilde{\mathcal{R}})$ by $\bar{q} = \gcd(Q, d_1, d_2, \ldots, d_n)$.

**Dominance rule:** There exists an optimal solution in which all delivery quantities in all routes are multiples of $\bar{q}$.
Strengthened formulation (F2)

- $R'$ — set of all resource-feasible routes (but not necessarily elementary)
- $D_i = \{\bar{q}, 2\bar{q}, \ldots, d_i\}$ — possible delivery quantities to $i \in C$.
- $b_{iF}^r = b_{i,d_i}^r$ — # of times $r \in R'$ delivers full demand to $i \in C$.
- $b_{iP}^r = \sum_{q \in D_i \setminus \{d_i\}} b_{iq}^r$ — # of times $r \in R'$ delivers partial demand to $i$.

(F2) : Objective and all constraints in (F0)

$$\sum_{r \in R'} (2b_{iF}^r + b_{iP}^r) \theta_r \geq 2, \quad \forall i \in C. \quad (*)$$

(*) is a special case of strong minimum number of vehicles (SVM) constraints from [Archetti et al., 2011].
Pricing problem for formulation (F2)

Example: \( i = 4, \ d_4 = 40, \ \bar{q} = 10. \)

- Arrows incoming to nodes \( i \) with delivery \( q \notin \{ \bar{q}, d_i \} \) can be removed without compromising correctness
- Their removal does not weaken formulation (F2)
A family of formulations (FK)

A valid inequality for a customer \( i \in C \)

\[
\sum_{r \in R'} \sum_{q \in D_i} (q b_{iq}^r) \theta_r \geq d_i, \quad (*)
\]

where \( b_{iq}^r \) is the # of times \( r \in R' \) visits \( i \in C \) delivering \( q \in D_i \).

Given \( K < d_i / \bar{q} \), after Chvátal-Gomory rounding with multiplier \( \frac{K-1}{d_i-\epsilon} \):

\[
\sum_{r \in R'} \sum_{q \in D_i} \sum_{k=1}^{K} (b_{iq}^r g_{iq}^k k) \geq K, \quad (**)
\]

where \( g_{iq}^k = 1 \) iff \( \frac{(k-1)d_i}{k-1} \leq q < \frac{kd_i}{k-1} \).

(FK) : Objective and all constraints in (F0)

Inequalities (*) \( \forall i \in C : K \geq d_i / \bar{q} \)

Inequalities (**) \( \forall i \in C : K < d_i / \bar{q} \)
**From partial to full discretisation: illustration**

- Number of incoming arcs for vertices $i \in C$ in the pricing for $(FK)$ is at most $K$.
- Full discretisation formulation ($FK_{\text{max}}$), $K_{\text{max}} = \max_{i \in C} \left\{ \frac{d_i}{\bar{q}} \right\}$.

### Partial discretisation

- $2 \leq K < \frac{d_i}{\bar{q}}$
- $d_i/(K - 1) \leq K - 1$
- $k = \{1, 2, \ldots, K\}$
- $\left\lceil \frac{(k-1)d_i}{K-1}, \frac{kd_i}{K-1} \right\rceil$

### Full discretisation

Example: $d_i = 40, \bar{q} = 5, \frac{d_i}{\bar{q}} = 8$

<table>
<thead>
<tr>
<th>$D_i$</th>
<th>$D_i$</th>
<th>$D_i$</th>
<th>$D_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${5, 10, 20, 30, 40}$</td>
<td>${5, 10, 20, 25, 35, 40}$</td>
<td>${5, 10, 15, 20, 30, 35, 40}$</td>
<td>${5, 10, 15, 20, 25, 30, 35, 40}$</td>
</tr>
<tr>
<td>$</td>
<td>D_i</td>
<td>= 5$</td>
<td>$</td>
</tr>
</tbody>
</table>
Valid inequalities

\( x^r_{ij} \) — # of times \( r \in R' \) follows arc \((i, j) \in A, i, j \in C \cap \{0\} \).

- Rounded capacity inequalities:

\[
\sum_{r \in R} \sum_{(i,j) \in A: \{|i,j\} \cap S| = 1} x^r_{ij} \theta_r \geq 2 \left[ \sum_{i \in S} d_i / Q \right], \quad \forall S \subseteq C.
\]

- 3-row subset-row packing inequalities:

\[
\sum_{r \in R} \left[ \sum_{i \in S} \sum_{q \in D_i: q > d_i / 2} \frac{1}{2} b^r_{iq} \right] \theta_r \leq 1, \quad \forall S \subseteq C, |S| = 3.
\]
Valid inequalities (II)

- 3-row subset-row covering inequalities:

\[
\sum_{r \in R} \left[ \sum_{i \in S} \sum_{q \in D_i: q > 0} \frac{1}{2} b_{iq} \right] \theta_r \geq 2, \quad \forall S \subseteq C, \ |S| = 3.
\]

- Limited memory technique ([Pecin et al., 2017]) is used for all non-robust cuts.
Implementation

- C++ libraries BaPCod [Sadykov and Vanderbeck, 2021] and VRPSolver extension [Pessoa et al., 2020] are used to leverage all the latest advances on exact solution of the classic CVRP
- VRPSolver is extended with
  - separation procedures for strong $k$-path inequalities
  - covering sets (to support limited-memory Chvátal-Gomory rank-1 covering cuts and strong $k$-path inequalities in the pricing)
- Branching on arcs and Ryan-and-Foster branching
Computational evaluation

Instance sets

- **SDVRPTW** – 504 test instances, derived from 56 classic Solomon’s VRPTW instances, having \( n = \{25, 50, 100\} \) and \( Q = \{30, 50, 100\} \).

- **SDVRP** – 352 test instances, 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).

Initial upper bounds

- We use an ILS-based matheuristic proposed by [Alvarez and Munari, 2022] to generate initial upper bounds.
Comparison of formulations (FK)

Root node results for all SDVRPTW instances with $n = 50$. 

- C, RC. $Q = 30$
- C, RC. $Q = 50$
- C, RC. $Q = 100$

- R. $Q = 30$
- R. $Q = 50$
- R. $Q = 100$
Comparison with the state-of-the-art on the SDVRPTW

<table>
<thead>
<tr>
<th>$n$</th>
<th>Benchmark run – 3600s</th>
<th></th>
<th></th>
<th></th>
<th>Long run – 18000s</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$(FK_{\text{max}})$</td>
<td>MS22</td>
<td>BI19</td>
<td>A11</td>
<td>$(F2)$</td>
<td>$(FK_{\text{max}})$</td>
<td>Best $(F2, FK_{\text{max}})$</td>
</tr>
<tr>
<td>25</td>
<td>168</td>
<td>168</td>
<td>168</td>
<td>168</td>
<td>168</td>
<td>168</td>
<td>168 (0)</td>
</tr>
<tr>
<td>50</td>
<td>152 (27)</td>
<td>123</td>
<td>104</td>
<td>86</td>
<td>136</td>
<td>168</td>
<td>168 (40)</td>
</tr>
<tr>
<td>100</td>
<td>54 (48)</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>24</td>
<td>55</td>
<td>56 (50)</td>
</tr>
<tr>
<td>{50, 100}</td>
<td>206</td>
<td>127</td>
<td>109</td>
<td>94</td>
<td>160</td>
<td>223</td>
<td>224 (90)</td>
</tr>
<tr>
<td>{25, 50, 100}</td>
<td>374 (75)</td>
<td>295</td>
<td>277</td>
<td>262</td>
<td>328</td>
<td>391</td>
<td>392 (90)</td>
</tr>
</tbody>
</table>

OI average Gap (%) 1.66 - - - 3.02 1.56 1.57

MS22: Munari and Savelsbergh (2022)
BI19: Bianchessi and Irnich (2019)
A11: Archetti et al. (2011)

- Formulation $(FK_{\text{max}})$ finds 374 optimal solutions, 75 for the first time, within one hour benchmark tests.
- Formulations $(F2)$ and $(FK_{\text{max}})$ all together find 392 optimal solutions, 90 for the first time, within five hours.
Comparison with the state-of-the-art on the SDVRP

Formulation \((FK)\), \(K = \min(K_{\text{max}}, 10)\)

<table>
<thead>
<tr>
<th>Tests</th>
<th>Model or reference – test set size</th>
<th>Opt</th>
<th>Opt*</th>
<th>LB*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark run – 7200s</td>
<td>(FK) (MH) – 352</td>
<td>94 (88†)</td>
<td>10 (6†)</td>
<td>121 (53†)</td>
</tr>
<tr>
<td></td>
<td>Munari and Savelsbergh (2022) – 224†</td>
<td>85</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Gouveia et al. (2021) – 352</td>
<td>106</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Long run – 18000s</td>
<td>(FK) (MH) – 352</td>
<td>112 (106†)</td>
<td>14 (10†)</td>
<td>130 (53†)</td>
</tr>
<tr>
<td></td>
<td>(FK) (BKS) – 352</td>
<td>121 (115†)</td>
<td>19 (15†)</td>
<td>134 (53†)</td>
</tr>
<tr>
<td></td>
<td>Best of long runs – 352</td>
<td>123 (117†)</td>
<td>20 (16†)</td>
<td>136 (54†)</td>
</tr>
</tbody>
</table>

† number of corresponding instances in the reduced test set considered in Munari and Savelsbergh (2022).

- Formulation \((FK)\) finds 94 (88) optimal solutions, 10 (6) for the first time, within two hours benchmark tests.
- Our best results overall account for 123 (117) optimal solutions, 20 (16) for the first time, within five hours.
Conclusions

▶ A new family of partially discretised route formulations ($FK$) for SDVRP’s.
▶ A new dominance rule ($\bar{q}$) for optimal SDVRP’s solutions.
▶ Experimentally ($FK$) becomes stronger with $K \uparrow$
▶ BCP algorithm is the new state-of-the-art for the SDVRPTW

Perspectives

▶ Our BCP algorithm can be easily extended to other variants such as multiple depots [Gouveia et al., 2021], heterogeneous fleet [Belfiore and Yoshizaki, 2009], using the generic VRPSolver model.
▶ Further strengthening of formulation ($FK_{\max}$) requires a generalized RCSPP solver for the pricing
▶ We are bad for at finding good primal solutions!
▶ Extension to inventory and/or production routing problems?


