Non-Elementary Formulations for the Single Vehicle Routing Problem with Deliveries and Selective Pickups

Bruno Petrato Bruck   Manuel Iori

Department of Science and Methods for Engineering (DISMI)
University of Modena and Reggio Emilia

Topics

1. Introduction
   - The problem
   - Motivation
   - Literature
2. Formulations
   - Relaxed Non-Elementary Formulation
   - Benders-Based Non-Elementary Formulation
3. Exact algorithms for combined demand
   - Throw Away
   - 2 Step
   - Minimal Extended Network
4. Computational experiments
5. Conclusions

The problem

- The *single vehicle routing problem with deliveries and selective pickups* (SVRPDSP) is one of the so-called *one-to-many-to-one single vehicle pickup and delivery problems*
- In the SVRPDSP
  - Deliveries are mandatory
  - Pickups are optional
    - Each pickup performed generates a certain revenue

The problem

- In case a customer has both demands
    - Two different visits
    - One visit
        - only the delivery
        - both demands simultaneously
- Partial fulfillments of demands are not allowed
- Objective is to minimize the total routing cost
    - Travel cost − Revenue generated

Motivation

- Arises in the so-called *reverse logistics* domain
- Common in several practical contexts
- Beverage distributors
  - The vehicle delivers full drink bottles to customers and retrieves empty bottles to the depot

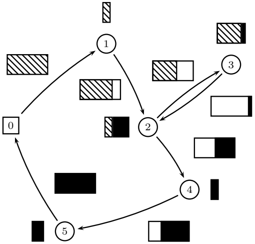
- Logistic service companies
- Electronic and battery manufacturers
  - Some countries already have some policies regarding waste disposal
  - Manufacturers are being held responsible for picking up broken and used products

Motivation

- The SVRPDSP is a generalization of some other *one-to-many-to-one single vehicle pickup and delivery problems*.
- For instance
  - Linehauls and Backhauls
  - Mixed load

Motivation

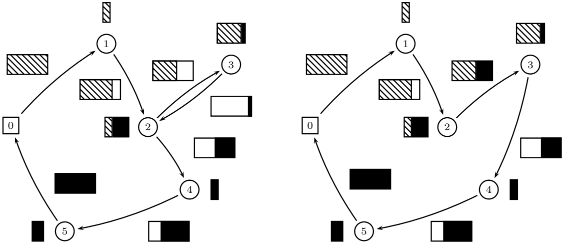
- The SVRPDSP is a generalization of some other *one-to-many-to-one single vehicle pickup and delivery problems*.
- For instance
  - Linehauls and Backhauls
  - Mixed load



(a) backhauls

Motivation

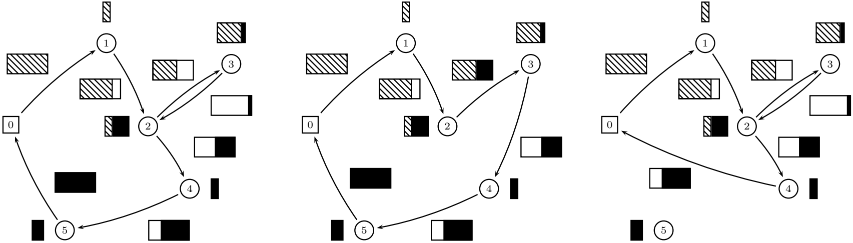
- The SVRPDSP is a generalization of some other *one-to-many-to-one single vehicle pickup and delivery problems*.
- For instance
  - Linehauls and Backhauls
  - Mixed load



(a) backhauls (b) mixed deliveries

Motivation

- The SVRPDSP is a generalization of some other *one-to-many-to-one single vehicle pickup and delivery problems*.
- For instance
  - Linehauls and Backhauls
  - Mixed load



(a) backhauls (b) mixed deliveries (c) selective pickups

Single-demand vs Combined-demand

Two SVRPDSP cases are known in the literature:

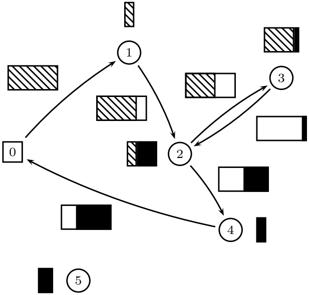
- *single-demand* (SD): customers have one type of demand, either delivery or pickup
- *combined-demand* (CD): may contain any type of customer

These two versions are, up to a certain extent, interchangeable:

- CD generalizes SD
- CD can be transformed into SD by *duplicating* combined customers, obtaining what we call the *extended network*

Single-demand vs Combined-demand

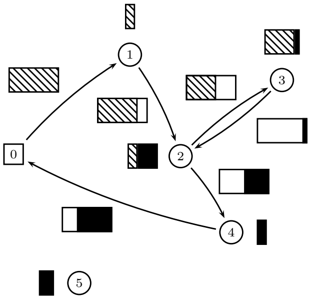
Original network → extended network



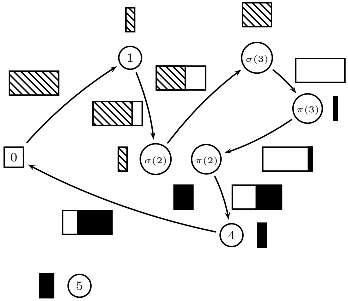
(c) selective pickups

Single-demand vs Combined-demand

Original network → extended network



(c) selective pickups



(c) selective pickups - extended network

Prior Work

As far as we know all SVRPDSP algorithms refer either to the extended network or to an equivalent scenario

Prior Work

As far as we know all SVRPDSP algorithms refer either to the extended network or to an equivalent scenario

In terms of exact algorithms:

- Süral and Bookbinder (*Networks*, 2003): MILP formulation
- Gribkovskaia, Laporte and Shyshou (C&OR, 2008): MILP formulation
- Gutiérrez-Jarpa, Marianov and Obreque (*IIE Transactions*, 2009): branch-and-cut algorithm

Prior Work

As far as we know all SVRPDSP algorithms refer either to the extended network or to an equivalent scenario

In terms of exact algorithms:

- Süral and Bookbinder (*Networks*, 2003): MILP formulation
- Gribkovskaia, Laporte and Shyshou (C&OR, 2008): MILP formulation
- Gutiérrez-Jarpa, Marianov and Obreque (*IIE Transactions*, 2009): branch-and-cut algorithm

In terms of heuristic to our knowledge the best ones are

- Coelho *et al* (*Electronic Notes in Discrete Mathematics*, 2012): General Variable Neighborhood Search
- Bruck, Santos and Arroyo (*CEC*, 2012): Hybrid Evolutionary Algorithm

Prior Work

Pros and cons of working on the Extended Network

- Advantage: optimal solutions are Elementary. Thus, many classical results from the literature can be reused
- Disadvantage: network is doubled in size

Prior Work

Pros and cons of working on the Extended Network

- Advantage: optimal solutions are Elementary. Thus, many classical results from the literature can be reused
- Disadvantage: network is doubled in size

In our work we consider instead *non-Elementary* solutions on the original problem network

Relaxed Non-Elementary Formulation

- Before getting into the Relaxed Non-Elementary Formulation that works on the original network

Relaxed Non-Elementary Formulation

- Before getting into the Relaxed Non-Elementary Formulation that works on the original network
- Let
    - V → the set of customers + depot (0)
    - A → the set of arcs
    - D → set of delivery customers
    - P → set of pickup customers
    - PD → set of combined demand customers

Two-commodity Elementary (TCE) formulation

- x_{ij} = 1 if vehicle travels along arc (i, j), 0 otherwise
- y_j = 1 if pickup of j is performed, 0 otherwise
- f^d_{ij} = flow of delivery commodity
- f^p_{ij} = flow of pickup commodity

(11)

Two-commodity Elementary (TCE) formulation

$$\min\ z_{TCE} = \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{j\in P} r_j(1 - y_j) \qquad (1)$$

- x_{ij} = 1 if vehicle travels along arc (i, j), 0 otherwise
- y_j = 1 if pickup of j is performed, 0 otherwise
- f^d_{ij} = flow of delivery commodity
- f^p_{ij} = flow of pickup commodity

(11)

Two-commodity Elementary (TCE) formulation

$$\min\ z_{TCE} = \sum\nolimits_{(i,j)\in A} c_{ij}x_{ij} + \sum\nolimits_{j\in P} r_j(1-y_j) \qquad (1)$$

$$\sum\nolimits_{i\in V} x_{ij} = 1 \qquad \forall\, j \in D \cup \{0\} \quad (2)$$

$$\sum\nolimits_{i\in V} x_{ij} = y_j \qquad \forall\, j \in P \quad (3)$$

- x_{ij} = 1 if vehicle travels along arc (i,j), 0 otherwise
- y_j = 1 if pickup of j is performed, 0 otherwise
- f_{ij}^d = flow of delivery commodity
- f_{ij}^p = flow of pickup commodity

(11)

Two-commodity Elementary (TCE) formulation

$$\min\ z_{TCE} = \sum\nolimits_{(i,j)\in A} c_{ij}x_{ij} + \sum\nolimits_{j\in P} r_j(1-y_j) \qquad (1)$$

$$\sum\nolimits_{i\in V} x_{ij} = 1 \qquad \forall\, j \in D \cup \{0\} \quad (2)$$

$$\sum\nolimits_{i\in V} x_{ij} = y_j \qquad \forall\, j \in P \quad (3)$$

$$\sum\nolimits_{i\in V} (x_{ij} - x_{ji}) = 0 \qquad \forall\, j \in V \quad (4)$$

(11)

- x_{ij} = 1 if vehicle travels along arc (i,j), 0 otherwise
- y_j = 1 if pickup of j is performed, 0 otherwise
- f^d_{ij} = flow of delivery commodity
- f^p_{ij} = flow of pickup commodity

Two-commodity Elementary (TCE) formulation

$$\min\ z_{TCE} = \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{j\in P} r_j(1-y_j) \qquad (1)$$

$$\sum_{i\in V} x_{ij} = 1 \qquad \forall\, j \in D \cup \{0\} \quad (2)$$

$$\sum_{i\in V} x_{ij} = y_j \qquad \forall\, j \in P \quad (3)$$

$$\sum_{i\in V} (x_{ij} - x_{ji}) = 0 \qquad \forall\, j \in V \quad (4)$$

$$f^d_{ij} + f^p_{ij} \le Qx_{ij} \qquad \forall\, (i,j) \in A \quad (5)$$

$$(11)$$

- x_{ij} = 1 if vehicle travels along arc (i, j), 0 otherwise
- y_j = 1 if pickup of j is performed, 0 otherwise
- f^d_{ij} = flow of delivery commodity
- f^p_{ij} = flow of pickup commodity

Two-commodity Elementary (TCE) formulation

$$\min \ z_{TCE} = \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{j\in P} r_j(1-y_j) \qquad (1)$$

$$\sum_{i\in V} x_{ij} = 1 \qquad \forall\, j \in D \cup \{0\} \quad (2)$$

$$\sum_{i\in V} x_{ij} = y_j \qquad \forall\, j \in P \quad (3)$$

$$\sum_{i\in V} (x_{ij} - x_{ji}) = 0 \qquad \forall\, j \in V \quad (4)$$

$$f_{ij}^d + f_{ij}^p \le Qx_{ij} \qquad \forall\, (i,j) \in A \quad (5)$$

$$\sum_{i\in V} \left(f_{ij}^d - f_{ji}^d \right) = d_j \qquad \forall\, j \in V \setminus \{0\} \quad (6)$$

$$\sum_{i\in V} \left(f_{ji}^p - f_{ij}^p \right) = p_j y_j \qquad \forall\, j \in P \quad (7)$$

$$\sum_{i\in V} \left(f_{ji}^p - f_{ij}^p \right) = 0 \qquad \forall\, j \in D \quad (8)$$

$$(11)$$

- x_{ij} = 1 if vehicle travels along arc (i,j), 0 otherwise
- y_j = 1 if pickup of j is performed, 0 otherwise
- f_{ij}^d = flow of delivery commodity
- f_{ij}^p = flow of pickup commodity

Two-commodity Elementary (TCE) formulation

$$\min \; z_{TCE} = \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{j\in P} r_j(1-y_j) \qquad (1)$$

$$\sum_{i\in V} x_{ij} = 1 \qquad \forall\, j \in D \cup \{0\} \quad (2)$$

$$\sum_{i\in V} x_{ij} = y_j \qquad \forall\, j \in P \quad (3)$$

$$\sum_{i\in V} (x_{ij} - x_{ji}) = 0 \qquad \forall\, j \in V \quad (4)$$

$$f^d_{ij} + f^p_{ij} \le Qx_{ij} \qquad \forall\, (i,j) \in A \quad (5)$$

$$\sum_{i\in V} \left(f^d_{ij} - f^d_{ji}\right) = d_j \qquad \forall\, j \in V \setminus \{0\} \quad (6)$$

$$\sum_{i\in V} \left(f^p_{ji} - f^p_{ij}\right) = p_j y_j \qquad \forall\, j \in P \quad (7)$$

$$\sum_{i\in V} \left(f^p_{ji} - f^p_{ij}\right) = 0 \qquad \forall\, j \in D \quad (8)$$

$$x_{ij} \in \{0,1\} \qquad \forall\, (i,j) \in A \quad (9)$$

$$y_j \in \{0,1\} \qquad \forall\, j \in P (10)$$

$$f^d_{ij}, f^p_{ij} \ge 0 \qquad \forall\, (i,j) \in A (11)$$

- x_{ij} = 1 if vehicle travels along arc (i,j), 0 otherwise
- y_j = 1 if pickup of j is performed, 0 otherwise
- f^d_{ij} = flow of delivery commodity
- f^p_{ij} = flow of pickup commodity

Two-commodity non-Elementary (TCNE) formulation

- x_{ij} = number of times that vehicle travels along arc (i, j)
- y_j = 1 if pickup of j is performed, 0 otherwise
- f_{ij}^d = flow of delivery commodity
- f_{ij}^p = flow of pickup commodity

(25)

Two-commodity non-Elementary (TCNE) formulation

$$\min \ z_{TCNE} = \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{j\in P\cup PD} r_j(1-y_j) \quad (12)$$

- x_{ij} = number of times that vehicle travels along arc (i,j)
- y_j = 1 if pickup of j is performed, 0 otherwise
- f^d_{ij} = flow of delivery commodity
- f^p_{ij} = flow of pickup commodity

(25)

Two-commodity non-Elementary (TCNE) formulation

$$\min \; z_{TCNE} = \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{j\in P\cup PD} r_j(1-y_j) \quad (12)$$

$$\sum_{i\in V} x_{ij} = 1 \qquad \forall\, j \in D \cup \{0\}(13)$$

$$\sum_{i\in V} x_{ij} = y_j \qquad \forall\, j \in P(14)$$

- x_{ij} = number of times that vehicle travels along arc (i,j)
- y_j = 1 if pickup of j is performed, 0 otherwise
- f_{ij}^d = flow of delivery commodity
- f_{ij}^p = flow of pickup commodity

(25)

Two-commodity non-Elementary (TCNE) formulation

$$\min \; z_{TCNE} = \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{j\in P\cup PD} r_j(1-y_j) \quad (12)$$

$$\sum_{i\in V} x_{ij} = 1 \qquad \forall\, j \in D \cup \{0\} (13)$$

$$\sum_{i\in V} x_{ij} = y_j \qquad \forall\, j \in P (14)$$

$$\sum_{i\in V} x_{ij} \geq 1 \qquad \forall\, j \in PD (15)$$

- x_{ij} = number of times that vehicle travels along arc (i,j)
- y_j = 1 if pickup of j is performed, 0 otherwise
- f_{ij}^{d} = flow of delivery commodity
- f_{ij}^{p} = flow of pickup commodity

(25)

Two-commodity non-Elementary (TCNE) formulation

$$\min \; z_{TCNE} = \sum\nolimits_{(i,j)\in A} c_{ij}x_{ij} + \sum\nolimits_{j\in P\cup PD} r_j(1-y_j) \quad (12)$$

$$\sum\nolimits_{i\in V} x_{ij} = 1 \qquad \forall\, j \in D \cup \{0\}(13)$$

$$\sum\nolimits_{i\in V} x_{ij} = y_j \qquad \forall\, j \in P(14)$$

$$\sum\nolimits_{i\in V} x_{ij} \geq 1 \qquad \forall\, j \in PD(15)$$

$$\sum\nolimits_{i\in V} x_{ij} \leq y_j + 1 \qquad \forall\, j \in PD(16)$$

- x_{ij} = number of times that vehicle travels along arc (i,j)
- y_j = 1 if pickup of j is performed, 0 otherwise
- f_{ij}^d = flow of delivery commodity
- f_{ij}^p = flow of pickup commodity

(25)

Two-commodity non-Elementary (TCNE) formulation

$$\min \ z_{TCNE} = \sum\nolimits_{(i,j)\in A} c_{ij} x_{ij} + \sum\nolimits_{j \in P \cup PD} r_j (1 - y_j) \quad (12)$$

$$\sum\nolimits_{i \in V} x_{ij} = 1 \qquad \forall\, j \in D \cup \{0\} \quad (13)$$

$$\sum\nolimits_{i \in V} x_{ij} = y_j \qquad \forall\, j \in P \quad (14)$$

$$\sum\nolimits_{i \in V} x_{ij} \geq 1 \qquad \forall\, j \in PD \quad (15)$$

$$\sum\nolimits_{i \in V} x_{ij} \leq y_j + 1 \qquad \forall\, j \in PD \quad (16)$$

$$\sum\nolimits_{i \in V} (x_{ij} - x_{ji}) = 0 \qquad \forall\, j \in V \quad (17)$$

(25)

- x_{ij} = number of times that vehicle travels along arc (i, j)
- y_j = 1 if pickup of j is performed, 0 otherwise
- f_{ij}^{d} = flow of delivery commodity
- f_{ij}^{p} = flow of pickup commodity

Two-commodity non-Elementary (TCNE) formulation

$$\min \ z_{TCNE} = \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{j\in P\cup PD} r_j(1-y_j) \quad (12)$$

$$\sum_{i\in V} x_{ij} = 1 \qquad \forall\, j \in D \cup \{0\}(13)$$

$$\sum_{i\in V} x_{ij} = y_j \qquad \forall\, j \in P(14)$$

$$\sum_{i\in V} x_{ij} \geq 1 \qquad \forall\, j \in PD(15)$$

$$\sum_{i\in V} x_{ij} \leq y_j + 1 \qquad \forall\, j \in PD(16)$$

$$\sum_{i\in V} (x_{ij} - x_{ji}) = 0 \qquad \forall\, j \in V(17)$$

$$f^d_{ij} + f^p_{ij} \leq Qx_{ij} \qquad \forall\, (i,j) \in A(18)$$

- x_{ij} = number of times that vehicle travels along arc (i,j)
- y_j = 1 if pickup of j is performed, 0 otherwise
- f^d_{ij} = flow of delivery commodity
- f^p_{ij} = flow of pickup commodity

(25)

Two-commodity non-Elementary (TCNE) formulation

$$\min \ z_{TCNE} = \sum\nolimits_{(i,j)\in A} c_{ij}x_{ij} + \sum\nolimits_{j\in P\cup PD} r_j(1-y_j) \quad (12)$$

$$\sum\nolimits_{i\in V} x_{ij} = 1 \qquad \forall\, j \in D\cup\{0\}(13)$$

$$\sum\nolimits_{i\in V} x_{ij} = y_j \qquad \forall\, j \in P(14)$$

$$\sum\nolimits_{i\in V} x_{ij} \geq 1 \qquad \forall\, j \in PD(15)$$

$$\sum\nolimits_{i\in V} x_{ij} \leq y_j + 1 \qquad \forall\, j \in PD(16)$$

$$\sum\nolimits_{i\in V} (x_{ij} - x_{ji}) = 0 \qquad \forall\, j \in V(17)$$

$$f^d_{ij} + f^p_{ij} \leq Qx_{ij} \qquad \forall\, (i,j) \in A(18)$$

$$\sum\nolimits_{i\in V} \left(f^d_{ij} - f^d_{ji}\right) = d_j \qquad \forall\, j \in V\setminus\{0\}(19)$$

$$\sum\nolimits_{i\in V} \left(f^p_{ji} - f^p_{ij}\right) = p_j y_j \qquad \forall\, j \in P\cup PD(20)$$

$$\sum\nolimits_{i\in V} \left(f^p_{ji} - f^p_{ij}\right) = 0 \qquad \forall\, j \in D(21)$$

(25)

- x_{ij} = number of times that vehicle travels along arc (i,j)
- y_j = 1 if pickup of j is performed, 0 otherwise
- f^d_{ij} = flow of delivery commodity
- f^p_{ij} = flow of pickup commodity

Two-commodity non-Elementary (TCNE) formulation

$$\min \; z_{TCNE} = \sum\nolimits_{(i,j)\in A} c_{ij}x_{ij} + \sum\nolimits_{j\in P\cup PD} r_j(1-y_j) \qquad (12)$$

$$\sum\nolimits_{i\in V} x_{ij} = 1 \qquad \forall\, j \in D \cup \{0\} \; (13)$$

$$\sum\nolimits_{i\in V} x_{ij} = y_j \qquad \forall\, j \in P \; (14)$$

$$\sum\nolimits_{i\in V} x_{ij} \geq 1 \qquad \forall\, j \in PD \; (15)$$

$$\sum\nolimits_{i\in V} x_{ij} \leq y_j + 1 \qquad \forall\, j \in PD \; (16)$$

$$\sum\nolimits_{i\in V} (x_{ij} - x_{ji}) = 0 \qquad \forall\, j \in V \; (17)$$

$$f^d_{ij} + f^p_{ij} \leq Qx_{ij} \qquad \forall\, (i,j) \in A \; (18)$$

$$\sum\nolimits_{i\in V} \left(f^d_{ij} - f^d_{ji}\right) = d_j \qquad \forall\, j \in V \setminus \{0\} \; (19)$$

$$\sum\nolimits_{i\in V} \left(f^p_{ji} - f^p_{ij}\right) = p_j y_j \qquad \forall\, j \in P \cup PD \; (20)$$

$$\sum\nolimits_{i\in V} \left(f^p_{ji} - f^p_{ij}\right) = 0 \qquad \forall\, j \in D \; (21)$$

$$x_{ij} \in \{0,1\} \qquad \forall\, (i,j) \in A \setminus A(PD) \; (22)$$

(25)

- x_{ij} = number of times that vehicle travels along arc (i,j)
- y_j = 1 if pickup of j is performed, 0 otherwise
- f^d_{ij} = flow of delivery commodity
- f^p_{ij} = flow of pickup commodity

Two-commodity non-Elementary (TCNE) formulation

$$\min\ z_{TCNE} = \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{j\in P\cup PD} r_j(1-y_j) \qquad (12)$$

$$\sum_{i\in V} x_{ij} = 1 \qquad \forall\ j \in D \cup \{0\} (13)$$

$$\sum_{i\in V} x_{ij} = y_j \qquad \forall\ j \in P (14)$$

$$\sum_{i\in V} x_{ij} \geq 1 \qquad \forall\ j \in PD (15)$$

$$\sum_{i\in V} x_{ij} \leq y_j + 1 \qquad \forall\ j \in PD (16)$$

$$\sum_{i\in V} (x_{ij} - x_{ji}) = 0 \qquad \forall\ j \in V (17)$$

$$f_{ij}^d + f_{ij}^p \leq Qx_{ij} \qquad \forall\ (i,j) \in A (18)$$

$$\sum_{i\in V} \left(f_{ij}^d - f_{ji}^d\right) = d_j \qquad \forall\ j \in V \setminus \{0\} (19)$$

$$\sum_{i\in V} \left(f_{ji}^p - f_{ij}^p\right) = p_j y_j \qquad \forall\ j \in P \cup PD (20)$$

$$\sum_{i\in V} \left(f_{ji}^p - f_{ij}^p\right) = 0 \qquad \forall\ j \in D (21)$$

$$x_{ij} \in \{0,1\} \qquad \forall\ (i,j) \in A \setminus A(PD) (22)$$

$$x_{ij} \in \{0,1,2\} \qquad \forall\ (i,j) \in A(PD) (23)$$

$$(25)$$

- x_{ij} = number of times that vehicle travels along arc (i,j)
- y_j = 1 if pickup of j is performed, 0 otherwise
- f_{ij}^d = flow of delivery commodity
- f_{ij}^p = flow of pickup commodity

Two-commodity non-Elementary (TCNE) formulation

$$
\begin{aligned}
&\min \; z_{TCNE} = \sum\nolimits_{(i,j)\in A} c_{ij}x_{ij} + \sum\nolimits_{j\in P\cup PD} r_j(1-y_j) && (12)\\
&\sum\nolimits_{i\in V} x_{ij} = 1 && \forall\, j \in D \cup \{0\} \; (13)\\
&\sum\nolimits_{i\in V} x_{ij} = y_j && \forall\, j \in P \; (14)\\
&\sum\nolimits_{i\in V} x_{ij} \geq 1 && \forall\, j \in PD \; (15)\\
&\sum\nolimits_{i\in V} x_{ij} \leq y_j + 1 && \forall\, j \in PD \; (16)\\
&\sum\nolimits_{i\in V} (x_{ij} - x_{ji}) = 0 && \forall\, j \in V \; (17)\\
&f^d_{ij} + f^p_{ij} \leq Qx_{ij} && \forall\, (i,j) \in A \; (18)\\
&\sum\nolimits_{i\in V} \left(f^d_{ij} - f^d_{ji}\right) = d_j && \forall\, j \in V \setminus \{0\} \; (19)\\
&\sum\nolimits_{i\in V} \left(f^p_{ji} - f^p_{ij}\right) = p_j y_j && \forall\, j \in P \cup PD \; (20)\\
&\sum\nolimits_{i\in V} \left(f^p_{ji} - f^p_{ij}\right) = 0 && \forall\, j \in D \; (21)\\
&x_{ij} \in \{0,1\} && \forall\, (i,j) \in A \setminus A(PD) \; (22)\\
&x_{ij} \in \{0,1,2\} && \forall\, (i,j) \in A(PD) \; (23)\\
&y_j \in \{0,1\} && \forall\, j \in P \cup PD \; (24)\\
&f^d_{ij}, f^p_{ij} \geq 0 && \forall\, (i,j) \in A \; (25)
\end{aligned}
$$

- x_{ij} = number of times that vehicle travels along arc (i,j)
- y_j = 1 if pickup of j is performed, 0 otherwise
- f^d_{ij} = flow of delivery commodity
- f^p_{ij} = flow of pickup commodity

Some Properties

Property 1

For the SD case, TCNE is equivalent to TCE and thus provides the optimal SVRPDSP solution value

Trivially checked by setting $PD = \emptyset$

Some Properties

Property 1

For the SD case, TCNE is equivalent to TCE and thus provides the optimal SVRPDSP solution value

Trivially checked by setting $PD = \emptyset$

Property 2

For the CD case, $z_{TCNE} \leq z_{TCE}$ and thus TCNE provides a relaxation of the SVRPDSP

Proven by showing that any TCE solution is mapped into a TCNE one, but the opposite does not hold

Relaxed Non-Elementary Formulation

- Why TCNE is a relaxed formulation for the SVRPDSP?

Relaxed Non-Elementary Formulation

- Why TCNE is a relaxed formulation for the SVRPDSP?
- Since TCNE allows second visits to CD customers

Relaxed Non-Elementary Formulation

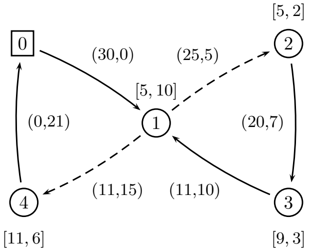
- Why TCNE is a relaxed formulation for the SVRPDSP?
- Since TCNE allows second visits to CD customers
  - Split deliveries or pickups (perform one part in the first visit and the other part in the second visit)

Relaxed Non-Elementary Formulation

- Why TCNE is a relaxed formulation for the SVRPDSP?
- Since TCNE allows second visits to CD customers
    - Split deliveries or pickups (perform one part in the first visit and the other part in the second visit)
    - Temporary dropoff of part of the load in a certain customer

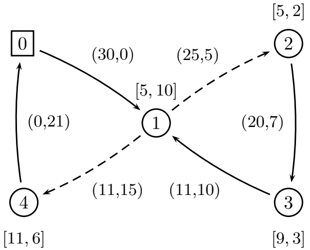
Relaxed Non-Elementary Formulation

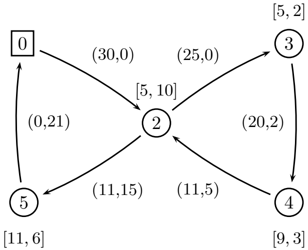
Example of split deliveries or pickups (Q = 30)



Relaxed Non-Elementary Formulation

Example of split deliveries or pickups (Q = 30)



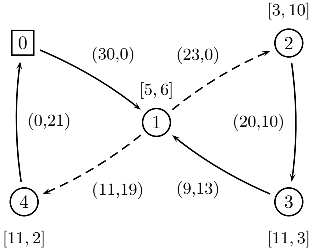


Property 3

A TCNH solution with split deliveries or pickups can be transformed into a solution having the same cost and for which no delivery or pickup is split

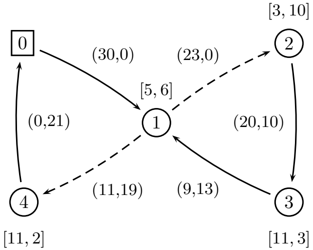
Relaxed Non-Elementary Formulation

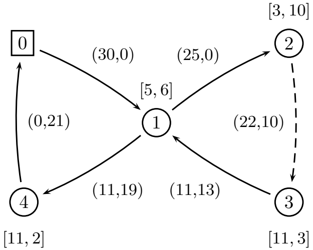
Example of temporary dropoff of part of the load in a certain customer (Q = 30)



Relaxed Non-Elementary Formulation

Example of temporary dropoff of part of the load in a certain customer (Q = 30)





No nice property can help us here, so we deal with dropoffs using tailored *branch-and-cut* (B&Cut) algorithms

Benders' Decomposition

We first obtain a *Benders' Based non-Elementary* (BBNE) formulation, by projecting out from TCNE the two-commodity flow variables:

$$\begin{aligned}
& f_{ij}^d + f_{ij}^p \leq Qx_{ij} && \forall\, (i,j) \in A, \\
& \sum_{i \in V} \left(f_{ij}^d - f_{ji}^d \right) = d_j && \forall\, j \in V \setminus \{0\}, \\
& \sum_{i \in V} \left(f_{ji}^p - f_{ij}^p \right) = p_j y_j && \forall\, j \in P \cup PD, \\
& \sum_{i \in V} \left(f_{ji}^p - f_{ij}^p \right) = 0 && \forall\, j \in D, \\
& f_{ij}^d, f_{ij}^p \geq 0 && \forall\, (i,j) \in A.
\end{aligned}$$

Benders' Decomposition

We first obtain a *Benders' Based non-Elementary* (BBNE) formulation, by projecting out from TCNE the two-commodity flow variables:

$$
\begin{aligned}
& f_{ij}^d + f_{ij}^p \leq Qx_{ij} && \forall\, (i,j) \in A,\\
& \sum_{i \in V} \left(f_{ij}^d - f_{ji}^d \right) = d_j && \forall\, j \in V \setminus \{0\},\\
& \sum_{i \in V} \left(f_{ji}^p - f_{ij}^p \right) = p_j y_j && \forall\, j \in P \cup PD,\\
& \sum_{i \in V} \left(f_{ji}^p - f_{ij}^p \right) = 0 && \forall\, j \in D,\\
& f_{ij}^d, f_{ij}^p \geq 0 && \forall\, (i,j) \in A.
\end{aligned}
$$

When a solution is found for the "difficult" (x, y) variables, we solve the dual of the above linear subproblem (DSP). If unbounded, we obtain a *Benders' feasibility cut* by using an extreme ray

Cutting Planes Generation

- Benders’ cuts are weak

Cutting Planes Generation

- Benders' cuts are weak
- Developed several additional families of valid inequalities
    - Classical inequalities
    - Tailored for the SVRPDSP

Cutting Planes Generation

- Benders' cuts are weak
- Developed several additional families of valid inequalities
  - Classical inequalities
  - Tailored for the SVRPDSP
- We provided for them polynomial-time separation procedures (max-flow)
  - In integer nodes we have faster separation procedures

Two-index Non-Elementary Formulation (TINE)

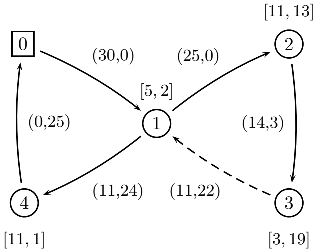
- Our capacity constraints are the following

$$x(\bar{S}:S) \geq \frac{d(V) - d(S) + p(S)}{Q} - 1 \qquad \begin{array}{l} \forall S \subseteq V \setminus \{0\}, \bar{S} = V \setminus S \setminus \{0\} : \\ d(V) - d(S) + p(S) > Q \end{array} \tag{26}$$

- Where
  - $d(S) = \sum_{i \in S} d_i$
  - $p(S) = \sum_{i \in S} p_i y_i$

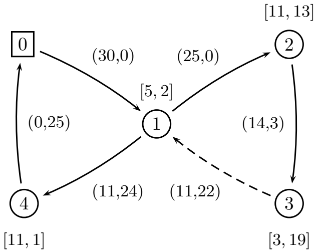
- $d(V) - d(S) + p(S)$ is the residual load of the vehicle after performing the demands in S

Two-index Non-Elementary Formulation (TINE)



- In this example $Q = 30$, $S = \{1, 2, 3\}$ and $\bar{S} = \{4\}$
- $d(V) - d(S) + p(S) = 30 - 19 + 23 = 34 > Q$
- Therefore the cut will specify that

Two-index Non-Elementary Formulation (TINE)



- In this example $Q = 30$, $S = \{1, 2, 3\}$ and $\bar{S} = \{4\}$
- $d(V) - d(S) + p(S) = 30 - 19 + 23 = 34 > Q$
- Therefore the cut will specify that

$$x(\bar{S} : S) \geq \frac{d(V) - d(S) + \sum_{j \in S} p_j y_j}{Q} - 1$$

$$x(\bar{S} : S) \geq \frac{30 - 19 + 24}{30} - 1$$

$$x(\bar{S} : S) \geq 0.16$$

Two-index Non-Elementary Formulation (TINE)

There are several ways to implement a “modern” B&Cut:

Two-index Non-Elementary Formulation (TINE)

There are several ways to implement a “modern” B&Cut:

- separate at fractional nodes (classical)
- separate at integer nodes (Subramanian et al., *OR Letters*, 2011)
- separate outside the MILP (Pferschy and Staněk, tech. rep., 2014)

Two-index Non-Elementary Formulation (TINE)

There are several ways to implement a “modern” B&Cut:

- separate at fractional nodes (classical)
- separate at integer nodes (Subramanian et al., *OR Letters*, 2011)
- separate outside the MILP (Pferschy and Staněk, tech. rep., 2014)

Our best B&Cut for TINE:

- Separate cuts only at integer nodes
- Separate Benders’ cuts just when strictly necessary

B&Cut algorithms for the CD case

The B&Cut for TINE solves to optimality the SD case

- but only provides a relaxation for the CD case

B&Cut algorithms for the CD case

The B&Cut for TINE solves to optimality the SD case

- but only provides a relaxation for the CD case

We used it as a basis for CD exact algorithms:

- *Throw-Away* (TA)
- *2-Steps* (2S)
- *Minimal Extended Network* (MEN)

Exact algorithms for combined demand

- **Throw Away (TA)**
  - This is the simplest approach
  - When an incumbent solution is found a feasibility check is done, if it has at least one dropoff the solution is rejected and the process continues
  - At the end the solution found will be the optimal for the SVRPDSP and will not contain dropoffs

Exact algorithms for combined demand

- During preliminary tests we found out that the *B&Cut* for TINE is considerably faster than TA and finds the optimal solution for several instances
- We take advantage of these observations to develop our second approach

Exact algorithms for combined demand

- During preliminary tests we found out that the *B&Cut* for TINE is considerably faster than TA and finds the optimal solution for several instances
- We take advantage of these observations to develop our second approach
- **2 Step (2S)**
  - At first we solve the normal *B&Cut* for the TINE keeping track of all cuts added
  - In case the solution found has dropoffs we solve the TA installing all the cuts found in the first step

Exact algorithms for combined demand

- **Minimal Extended Network (MEN)**
  - This algorithm works on the basis that solving the network of the duplicates always results in a solution with no dropoffs
  - The idea is to duplicate as few customers as possible to arrive at the optimal for the SVRPDSP
  - At each iteration
    - invoke B&Cut for TINE
    - check if final solution is feasible
    - if not, duplicate the CD customer originating the violation and re-iterate (inserting all generated cuts)

Computational experiments

In order to test our approaches we first use three sets available in the literature for the SVRPDSP

- SB set
  - 63 single demand instances
  - Number of customers equal to 10, 20 and 30
  - Proposed by Sural and Bookbinder
- GMO set
  - 74 single demand instances
  - Sizes ranging from 25 to 90 customers
  - Proposed by Gutiérrez-Jarpa, Marianov and Obreque

Computational experiments

In order to test our approaches we first use three sets available in the literature for the SVRPDSP

- GLS set
  - 68 combined demand instances
  - Sizes ranging from 15 to 100 customers
  - Proposed by Gribkovskaia, Laporte and Shyshou

Computational experiments

The implementations

- Done in C++
- Models were solved using Cplex 12.6
- 1 hour of time limit
- Ubuntu 13.04 with processor Intel(R) Core(TM) i7-3770 CPU 3.40GHz and 8Gb of RAM

The best exact approach in the literature so far was the B&Cut implemented by Guttierrez *et al*.

- However they consider only the symmetric case and only tested their approach with the sets SB and GMO

Computational experiments

Notice that since

- Dropoffs can only occur when there is a second visit to a customer
- Sets SB and GMO are single demand only

Our relaxed formulation TINE is enough to solve these instances

Computational results on the Single-Demand case.

			literature				new algorithms					
			SB		GMO*		TCNE		BBNE		TINE	
set	size	#	opt	sec	opt	sec	opt	sec	opt	sec	opt	sec
SB	*n*=10	24	**24**	0.5	**24**	0.0	**24**	0.1	**24**	0.0	**24**	0.0
	n=20	21	18	516.3	**21**	0.2	**21**	0.5	**21**	0.5	**21**	0.0
	n=30	18	17	621.2	**18**	0.6	**18**	2.3	**18**	3.1	**18**	0.1
	avg/sum	63	59	349.8	**63**	0.2	**63**	0.9	**63**	1.1	**63**	0.0
GMO	$25 \leq n \leq 30$	18	**18**	13.8	**18**	2.2	**18**	1.5	**18**	4.8	**18**	0.1
	$40 \leq n \leq 60$	39	26	1691.9	**39**	47.0	**39**	44.1	36	466.4	**39**	3.6
	$68 \leq n \leq 90$	17	2	3375.9	16	3510.8	**17**	314.6	6	2511.6	**17**	37.3
	avg/sum	74	46	1670.6	73	831.9	**74**	95.9	60	824.0	**74**	10.5

(∗ = run with with Cplex 10 on a Dual Core AMD 2.7 GHz, with 21000 sec time limit)

- SB = Süral and Bookbinder (2003)
- GMO = Gutiérrez-Jarpa, Marianov and Obreque (2009)

Computational results on the Combined-Demand case.

		Elementary						non-Elementary					
		GLS		SB		TCE		TA		2S		MEN	
GLS set	#	opt	sec	opt	sec	opt	sec	opt	sec	opt	sec	opt	sec
$15 \leq n \leq 30$	28	1	3486.5	5	3056.8	23	914.9	**28**	1.7	**28**	0.6	**28**	0.2
$32 \leq n \leq 50$	24	0	t.lim.	0	t.lim.	1	3591.9	18	916.2	18	912.0	**24**	6.5
$71 \leq n \leq 100$	16	0	2903.7	0	t.lim.	0	t.lim.	8	2023.7	10	1386.1	***15**	416.6
avg/sum	68	1	3389.5	5	3376.3	24	2491.5	54	800.2	56	648.3	**67**	100.4

($\ast$ = remaining instance solved to proven optimality by MEN in 10729 seconds)

- GLS = Gribkovskaia, Laporte and Shyshou (2008)
- SB = Süral and Bookbinder (2003)

What is the impact of working on the original network?

What is the impact of working on the original network?

Computational results on the Combined-Demand case.

		TCE			TCNE		
GLS set	#	opt	gap	sec	opt	gap	sec
$15 \leq n \leq 30$	28	23	0.68	914.95	28	0.00	40.53
$32 \leq n \leq 50$	24	1	3.79	3591.95	13	1.97	1976.31
$71 \leq n \leq 100$	16	0	22.25	t.lim.	0	17.34	t.lim.
avg/sum	68	24	7.47	2491.55	41	5.19	1561.27

- GLS = Gribkovskaia, Laporte and Shyshou (2008)
- SB = Süral and Bookbinder (2003)

Evaluation of the dropoff relaxation on the Combined-Demand case

		TINE (for the SVRPDSP-D)							MEN (for the SVRPDSP)						
GLS set	#	opt	sec	gap$_r$	gap$_c$	nodes	feas	gap$_d$	opt	sec	gap$_r$	gap$_c$	nodes	dupl	iter
$15 \leq n \leq 30$	28	28	0.3	4.74	0.59	236	26	0.00	28	0.2	4.74	0.59	206	0.1	1.1
$32 \leq n \leq 50$	24	24	10.4	6.27	0.88	6785	24	0.00	24	6.5	6.27	0.98	5209	0.4	1.4
$71 \leq n \leq 100$	16	16	352.6	10.66	0.83	40085	10	0.05	15	416.6	10.70	0.88	32174	1.2	2.2
avg/sum	68	68	86.78	6.67	0.75	11924	60	0.01	67	100.4	6.68	0.79	9493	0.5	1.5

We adapted our algorithms to solve the variant in which pickups are mandatory (Mixed load)

We adapted our algorithms to solve the variant in which pickups are mandatory (Mixed load)

Computational results on the SVRPPD (i.e., mandatory pickups case).

		literature			new algorithms								
		GHLV			TA-MP			2S-MP			MEN-MP		
GHLV set	#	opt	sec	gap	opt	sec	gap	opt	sec	gap	opt	sec	gap
15$\leq n \leq$30	14	1	3419.2	10.8	**14**	0.4	0.0	**14**	0.5	0.0	**14**	0.4	0.0
32$\leq n \leq$50	12	0	t.lim.	16.2	11	559.7	0.1	11	471.3	0.2	**12**	3.6	0.0
71$\leq n \leq$100	8	0	t.lim.	22.0	7	745.4	0.1	**8**	198.9	0.0	**8**	88.2	0.0
avg/sum	42	1	3525.5	19.9	32	373.1	0.1	33	213.4	0.1	**34**	22.2	0.0

- GHLV = Gribkovskaia, Halskau, Laporte and Vlcek (2007)

Since we were able to solve all instances from the literature to optimality we propose 2 new sets of large size instances

- BI set: 16 CD symmetric instances for the SVRPDSP (selective pickups)
- BI-MP set: 8 CD symmetric instances for the SVRPPD (mandatory pickups)
- Sizes ranging from 120 to 200 customers

Since we were able to solve all instances from the literature to optimality we propose 2 new sets of large size instances

- BI set: 16 CD symmetric instances for the SVRPDSP (selective pickups)
- BI-MP set: 8 CD symmetric instances for the SVRPPD (mandatory pickups)
- Sizes ranging from 120 to 200 customers

We attempted instances with asymmetric cost matrices

- but did not find relevant differences

Computational results on large-size SVRPDSP and SVRPPD instances.

Selective pickups (SVRPDSP)						Mandatory pickups (SVRPPD)					
		MEN						MEN-MP			
BI set	#	opt	sec	gap	nodes	BI-MP set	#	opt	sec	gap	nodes
n=120	4	1	3208.6	1.6	208570	*n*=120	2	2	960.3	0.0	42510
n=135	4	0	t.lim.	2.2	91743	*n*=135	2	2	2257.6	0.0	12980
n=151	4	4	669.6	0.0	35754	*n*=151	2	2	341.5	0.0	1922
n=200	4	2	2654.0	0.7	49967	*n*=200	2	0	t.lim.	1.1	46100
avg/sum	16	7	2533.0	1.1	96508	avg/sum	8	6	1789.9	0.3	25878

Conclusions

We developed new formulations that exploit the original non-Elementary structure of the problem

We found good computational results, especially for the more general combined-demand case

- For all SB and GMO instances, TINE found the optimal in just a few seconds
- MEN found the optimal for all GLS instances
    - All but one within the time limit of 1 hour

Conclusions

Although TA and 2S were not a match for MEN they still perform better than the approaches in the literature

We were able to see that it is worth it investing on approaches that deal with the original network

- TCNE in average about twice as efficient as TCE

Conclusions

- We have created 2 new sets of large size instances
  - BI set: 16 CD symmetric instances for the SVRPDSP (selective pickups)
  - BI-MP set: 8 CD symmetric instances for the SVRPPD (mandatory pickups)
- Results attest to the efficiency of our algorithm MEN, solving
  - BI set: 7 out of 16
  - BI-MP set: 6 out of 8

Conclusions

Solutions are frequently non-Elementary.

Ideas can be extended to several other general PD problems:

- Mixed deliveries
- Backhaul deliveries
- Single vehicle problems with dropoff
- Multiple vehicle problems (with and without transhipment)

Thank you, questions?