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

Bruno P. Bruck   Manuel Iori

DISMI, University of Modena and Reggio Emilia, Italy

ROUTE 2014

Outline

One-to-Many-to-One Transportation

We study the area of *one-to-many-to-one single vehicle pickup and delivery problems* (1-M-1 SVPDPs):

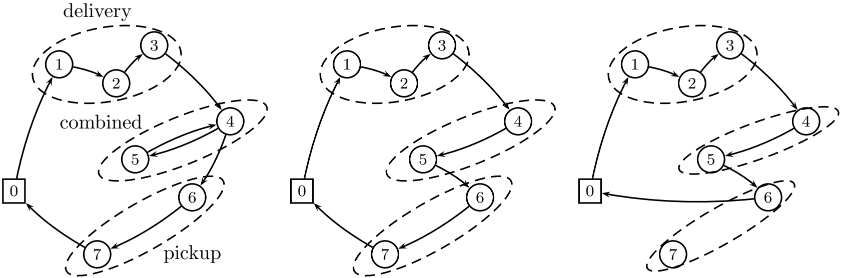
- A single vehicle laves the depot and delivers a first commodity to customers, it also collects a second commodity and brings it back to the depot

- Classical example: the vehicle delivers full drink bottles to customers and retrieves empty bottles to the depot
- 1-M-1 SVPDPs model many other real-world situations (distribution of electrical and electronic devices, courier service transportation, *reverse logistics*, ...)

General 1-M-1 SVPDPs Variants

Several variants are studied to better model the different real-world issues:

- Here we focus on the *general 1-M-1 SVPDPs* (Gribkovskaia and Laporte, 2008), where customers are not restrained to be visited once
- The most common general variants are the following ones



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

Selective pickups case can lead to lowest costs and can generalize the other cases (Golden&Assad, *Oper. Res.*, 1986)

Single Vehicle with Deliveries and Selective Pickups

We study the *Single VRP with Deliveries and Selective Pickups* (SVDSP):

- we are given a graph $G = (V, A)$, where $V = \{0\} \cup P \cup D \cup PD$
- $P = \{\text{pickup cust.}\}$, $D = \{\text{delivery cust.}\}$, $PD = \{\text{combined cust.}\}$
- each $j \in D$ asks for a delivery of weight d_j
- each $j \in P$ offers a pickup of weight p_j and revenue r_j
- each $j \in PD$ has both delivery and pickup
- a vehicle of capacity Q
- a traveling cost c_{ij} is associated with each arc $(i, j) \in A$

The SVDSP is to find a route that performs all deliveries and possibly some pickups, and minimizes the total cost

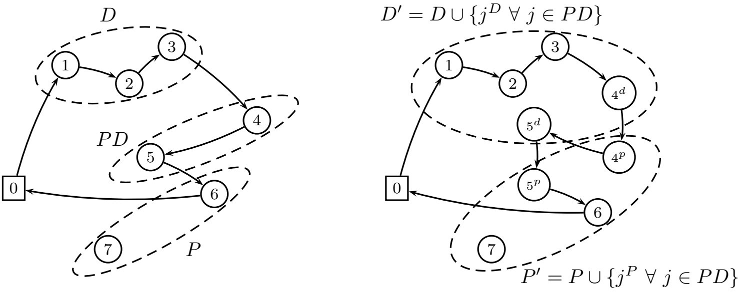
Single-demand vs Combined-demand

Two SVDSP cases are known in the literature:

- *single-demand* (SD) case contains only customers in P or in D
- *combined-demand* (CD) case 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



(a) selective pickups (b) selective pickups – *network of the duplicates*

Prior Work

Probably all SVDSP algorithms in the literature refer to the network of the duplicates and solve the SD case. In terms of exact algorithms:

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

Prior Work

Probably all SVDSP algorithms in the literature refer to the network of the duplicates and solve the SD case. In terms of exact algorithms:

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

Pros and cons of this idea:

- Advantage: optimal solutions are Hamiltonian, so many classical results from the literature can be reused
- Disadvantage: network might be doubled in size

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

Two-commodity Hamiltonian (TCH) formulation

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

$$\begin{aligned}
&\sum_{i\in V'} x_{ij} = 1 && \forall\, j \in D' \cup \{0\},\\
&\sum_{i\in V'} x_{ij} = y_j && \forall\, j \in P',\\
&\sum_{i\in V'} (x_{ij} - x_{ji}) = 0 && \forall\, j \in V',\\
&f_{ij}^d + f_{ij}^p \le Q x_{ij} && \forall\, (i,j) \in A',\\
&\sum_{i\in V'} \left(f_{ij}^d - f_{ji}^d\right) = d_j && \forall\, j \in V' \setminus \{0\},\\
&\sum_{i\in V'} \left(f_{ji}^p - f_{ij}^p\right) = p_j y_j && \forall\, j \in P',\\
&\sum_{i\in V'} \left(f_{ji}^p - f_{ij}^p\right) = 0 && \forall\, j \in D',\\
&x_{ij} \in \{0,1\} && \forall\, (i,j) \in A',\\
&y_j \in \{0,1\} && \forall\, j \in P',\\
&f_{ij}^d, f_{ij}^p \ge 0 && \forall\, (i,j) \in A'.
\end{aligned}$$

- $V' = \{0\} \cup P' \cup D'$
- $x_{ij} = 1$ 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 non-Hamiltonian (TCNH) formulation

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

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

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

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

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

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

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

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

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

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

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

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

$$y_j \in \{0,1\} \qquad \forall\, j \in P \cup PD,$$

$$f_{ij}^d, f_{ij}^p \geq 0 \qquad \forall\, (i,j) \in A.$$

- $V = \{0\} \cup P \cup D \cup PD$
- 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

Some Properties

Property 1

For the SD case, TCNH is equivalent to TCH and thus provides the optimal SVDSP solution value

Trivially checked by setting $PD = \emptyset$

Some Properties

Property 1

For the SD case, TCNH is equivalent to TCH and thus provides the optimal SVDSP solution value

Trivially checked by setting $PD = \emptyset$

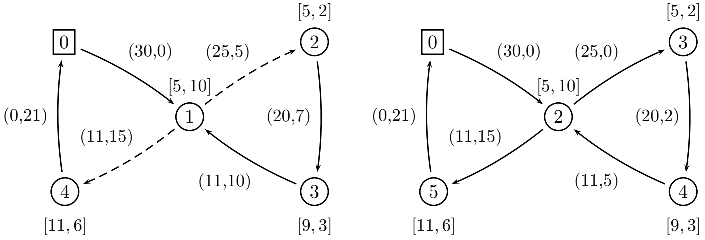
Property 2

For the CD case, $z_{TCNH} \leq z_{TCH}$ and thus TCNH provides a relaxation of the SVDSP

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

Solutions with Split Deliveries

TCNH accepts solutions where pickups and/or deliveries are *split* in the two visits to combined customers

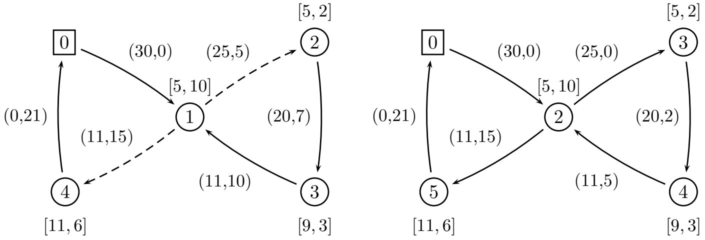


(a) pickup p_1 is split in two visits

(b) no split of p_1

Solutions with Split Deliveries

TCNH accepts solutions where pickups and/or deliveries are *split* in the two visits to combined customers



(a) pickup p_1 is split in two visits (b) no split of p_1

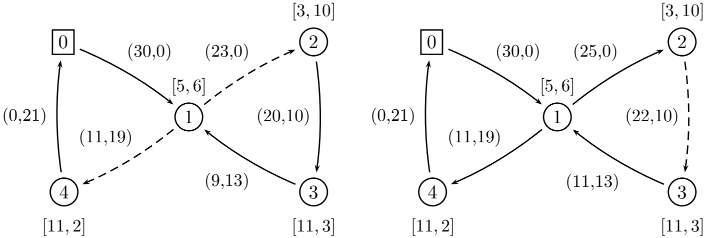
Property 3

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

Done with a path-search algorithm (that might take exponential time!)

Solutions with Temporary Dropoffs

TCNH also accepts solutions where some part of the load is temporary dropped off in the first visit to a customer, and then recollected during the second visit



(a) dropoff of 2 units of load in vertex 1 (b) no dropoff but capacity exceeded on (2,3)

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

Benders' Decomposition

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

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

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 the following *Benders' feasibility cut*:

$$
\sum_{(i,j) \in A} (Q t_{ij}^r) x_{ij} + \sum_{j \in V \setminus \{0\}} d_j v_j^r + \sum_{j \in P \cup PD} (p_j w_j^r) y_j \leq 0 \qquad \forall r \in R,
$$

where R denotes the set of the extreme rays of the DSP

Cutting Planes Generation

Benders' cuts are weak, so we tried to improve them (but failed). However, we found additional families of valid inequalities, and provided for them polynomial-time separation procedures (max-flow). We solved the resulting *Two-Index Non-Hamiltonian* (TINH) formulation by B&Cut

Cutting Planes Generation

Benders' cuts are weak, so we tried to improve them (but failed). However, we found additional families of valid inequalities, and provided for them polynomial-time separation procedures (max-flow). We solved the resulting *Two-Index Non-Hamiltonian* (TINH) formulation by B&Cut

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 TINH:

- separate Benders' cuts only when new incumbent is found
- separate all other cuts at integer nodes

An example of a Valid Cut

Property 4

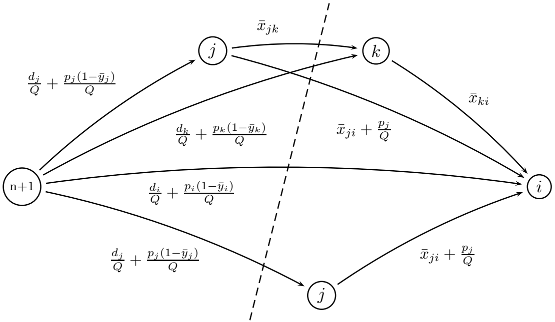
The following capacity constraint is valid for the SDSP:

$$Qx(\bar{S}:S) - \sum_{j \in S} p_j y_j \geq d(V) - d(S) - Q \quad \forall S \subseteq V \setminus \{0\} : p(S) - d(S) > Q - d(V)$$

where $\bar{S} = V \setminus S \setminus \{0\}$, $d(S) = \sum_{j \in S} d_j$, $p(S) = \sum_{j \in S} p_j$

Property 5

Capacity constraints can be separated in polynomial time



B&Cut algorithms for the CD case

The B&Cut for TINH 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:

1. *Throw-Away* (TA): inside B&Cut for TINH, when a new incumbent is found, check if it is feasible (i.e., it has no dropoffs). If not, simply disregard solution
2. *2-Steps* (2S): first invoke B&Cut for TINH. If final solution is not feasible, invoke TA (but initialize it with all cuts already generated)
3. *Minimal Network of the Duplicates* (MND): invoke B&Cut for TINH but disregard Benders' cuts. Check if capacity or dropoff are violated in final solution. If so, duplicate the CD customer originating the violation and re-iterate (inserting all generated cuts)

Single-Demand Case

Algorithms coded in C++, all tests run with Cplex 12.6, i7 3.4 GHz, 3600 sec t.lim.

		literature				new					
		SB		GMO*		TCNH		BBNH		TINH	
SB benchmark	#	opt	sec	opt	sec	opt	sec	opt	sec	opt	sec
($\|V\|$=10)	24	**24**	1.8	**24**	0.0	**24**	0.1	**24**	0.0	**24**	0.0
($\|V\|$=20)	21	18	461.2	**21**	0.2	**21**	0.5	**21**	0.5	**21**	0.0
($\|V\|$=30)	18	14	823.2	**18**	0.6	**18**	2.3	**18**	3.1	**18**	0.1
SB All	63	56	389.6	**63**	0.2	**63**	0.9	**63**	1.1	**63**	0.0
GMO benchmark	#	opt	sec	opt	sec	opt	sec	opt	sec	opt	sec
(25≤ $\|V\|$ ≤30)	18	**18**	32.3	**18**	2.2	**18**	1.5	**18**	4.8	**18**	0.1
(40≤ $\|V\|$ ≤60)	39	21	2441.8	**39**	47.0	**39**	44.1	36	466.4	**39**	3.6
(68≤ $\|V\|$ ≤90)	17	3	3898.4	16	3510.8	**17**	314.6	6	2513.0	**17**	37.3
GMO All	74	42	2190.4	73	831.9	**74**	95.9	60	824.3	**74**	10.5

- SB = Süral and Bookbinder (2003)
- GMO = Gutiérrez-Jarpa, Marianov and Obreque (2009), run with with Cplex 10, Dual Core AMD 2.7 GHz, 21000 sec t.lim.

Combined-Demand Case

Algorithms coded in C++, all tests run with Cplex 12.6, i7 3.4 GHz, 3600 sec t.lim.

		Hamiltonian						non-Hamiltonian					
		GLS		SB		TCH		TA		2S		MND	
GLS benchmark	#	opt	sec	opt	sec	opt	sec	opt	sec	opt	sec	opt	sec
$(15 \leq \lvert V \rvert \leq 30)$	28	1	3487	5	2961	23	915	**28**	0.4	**28**	0.4	**28**	0.2
$(32 \leq \lvert V \rvert \leq 50)$	24	0	3600	0	3600	1	3592	21	837.3	19	833.2	**24**	5.2
$(71 \leq \lvert V \rvert \leq 100)$	16	0	3600	0	3600	0	3600	10	1412.1	10	1369.0	$^{(*)}$**15**	375.5
All GLS	68	1	3553	5	3336	24	2492	59	628.0	57	616.4	**67**	90.3

- GLS = Gribkovskaia, Laporte and Shyshou (2008)
- SB = Süral and Bookbinder (2003)
- * = remaining instance solved to proven optimality in about 4.7 CPU hours

Conclusions

We developed new formulations that exploit the original non-Hamiltonian structure of the problem and provided a few theoretical properties

By developing several B&Cut algorithms we found good computational results, especially for the more general combined-demand case

Solutions are frequently non-Hamiltonian. Temporary dropoffs, if allowed, are frequent

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 very much for your attention

Comments and questions are welcome