Non Hamiltonian 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
The **Single Vehicle Routing Problem with Deliveries and Selective Pickups** (SVDSP) is one of the so-called *one-to-many-to-one single vehicle pickup and delivery problems*. [3]

- In the SVDSP
  - Deliveries are mandatory
  - Pickups are optional
    - Each pickup performed generates a certain revenue

- In case a customer has combined demand
  - Two different visits
  - One visit (Perform only the delivery or both demands simultaneously)

- Objective is to minimize the total routing cost
  - Travel cost – Revenue generated
The SVDSP is a generalization of some others *one-to-many-to-one single vehicle pickup and delivery problems.*

For instance

- Linehauls and Backhauls
- Mixed load

(a) backhaul  
(b) mixed load  
(c) selective pickups
Motivation

- Arises in the so-called *reverse logistics* domain
- Common in several practical contexts
  - Beverage distributors
  - 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
**Exact approaches (MILP formulations)**

- Sural and Bookbinder [Networks-2003][6]
- Gutiérrez-Jarpa, Marianov and Obreque [IIE Transactions - 2009][5]

**In terms of heuristic to our knowledge the best ones are**

- General Variable Neighborhood Search by Coelho *et al* [Electronic Notes in Discrete Mathematics - 2012][2]
- Evolutionary Algorithm by Bruck, Santos and Arroyo [CEC - 2012][1]
All exact approaches from the literature work on what we call *Network of Duplicates* to deal with combined demand customers.

- Each customer having combined demand is split into two different customers (One with each demand)
- Travel cost between the two equal to zero

Using the *Network of Duplicates* can be very costly.

A non-hamiltonian formulation

- Can work on the original network
- Has a totally different structure
- Special cases arise
To explore this new possible path of research in the SVDSP we developed a Relaxed Non-Hamiltonian Formulation that works on the original network.

In order to describe the formulation let:

- \( V \rightarrow \) the set of customers + depot
- \( A \rightarrow \) the set of arcs
- \( D \rightarrow \) set of delivery customers
- \( P \rightarrow \) set of pickup customers
- \( PD \rightarrow \) set of combined demand customers
- \( Q \rightarrow \) vehicle capacity
Relaxed Non-Hamiltonian Formulation

And consider the following variables:

- $x_{ij} = \begin{cases} 1 & \text{in case the arc } (i, j) \text{ is traversed} \\ 0 & \text{otherwise} \end{cases} \quad \forall (i, j) \in A$

- $y_{ij} = \begin{cases} 1 & \text{if the pickup of customer } i \text{ is performed} \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in P \cup PD$

- $f^d_{ij} = \text{delivery load in arc } (i, j) \quad \forall (i, j) \in A$

- $f^p_{ij} = \text{pickup load in arc } (i, j) \quad \forall (i, j) \in A$

Then we define the following \textit{two-commodity non-hamiltonian formulation} (TCHN)
(TCNH) \[ \min \ z_{TCNH} = \sum_{(i,j) \in A} c_{ij} x_{ij} + \sum_{j \in P \cup PD} r_j (1 - y_j) \]  

subject to:

\[ \sum_{i \in V} x_{ij} = 1 \quad \forall j \in D \cup \{0\}, \]  

\[ \sum_{i \in V} x_{ij} = y_j \quad \forall j \in P, \]  

\[ \sum_{i \in V} x_{ij} \geq 1 \quad \forall j \in PD, \]  

\[ \sum_{i \in V} x_{ij} \leq y_j + 1 \quad \forall j \in PD, \]  

\[ \sum_{i \in V} (x_{ij} - x_{ji}) = 0 \quad \forall j \in V, \]  

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

\[ \sum_{i \in V} (f_{ij}^d - f_{ji}^d) = d_j \quad \forall j \in V \setminus \{0\}, \]  

\[ \sum_{i \in V} (f_{ji}^p - f_{ij}^p) = p_j y_j \quad \forall j \in P \cup PD, \]  

\[ \sum_{i \in V} (f_{ji}^p - f_{ij}^p) = 0 \quad \forall j \in D, \]  

\[ x_{ij} \in \{0, 1\} \quad \forall (i,j) \in A \setminus A(PD), \]  

\[ x_{ij} \in \{0, 1, 2\} \quad \forall (i,j) \in A(PD), \]  

\[ y_j \in \{0, 1\} \quad \forall j \in P \cup PD. \]
Since TCHN allows second visits to customers having combined demand two interesting cases should be noted, which can result in solutions that are not feasible for the SVDSP

- 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

(a) pickup $p_1$ is split in two visits
(b) dropoff of 2 units of load in vertex 1
It is possible to prove that solutions having split deliveries or pickups can be converted into a feasible solution for the SVDSP having the same cost and in which not demands are split.

The prove is based on the fact that it is never suboptimal to perform the delivery first.

(a) pickup $p_1$ is split in two visits

(b) no split of $p_1$
However for the case of dropoff there is not such a nice property and is the reason why TCNH is a relaxation for the SVDSP

(a) dropoff of 2 units of load in vertex 1

(b) no dropoff but capacity exceeded on (2,3)
Another interesting property of this formulation is that for the single demand case the TCNH is exact since it is impossible to perform dropoffs given that no second visits are allowed.

- Variables $f^d$ and $f^p$ are an inconvenient of TCHN and may slow down the solution process for large instances.

- To obtain an equivalent but faster formulation we use the classical decomposition of Benders to project out the $f^d$ and $f^p$ variables.

- By doing so the constraints involving these variables are removed.
The following constraint is added corresponding to the *bender’s feasibility cuts*

\[
\sum_{(i,j) \in A} (Qt_{ij}^r)x_{ij} + \sum_{j \in V \setminus \{0\}} d_j v_j^r + \sum_{j \in P \cup PD} (p_j w_j^r)y_j \leq 0 \quad \forall r \in R
\]  

Where \(t^r\), \(v^r\) and \(w^r\) are extreme ray values associated with the dual subproblem and \(R\) is the set of all extreme rays.

This new formulation is called Benders-Based Non-Hamiltonian (BBNH) and is equivalent to TCNH
Two-index Non-Hamiltonian Formulation

Benders feasibility cuts are known to be weak in practice, and preliminary computational tests confirmed this behavior for BBNH even by separating them inside a Branch&Cut (B&C) framework.

In order to try improving the efficiency of BBNH we use a few families of valid inequalities valid for the SVDSP:

- Subtour elimination constraints (SECs)
- Capacity constraints
- Extended cover inequalities (ECIs)
Two-index Non-Hamiltonian Formulation

One of the most important and most interesting families is the following set of capacity constraints

\[
x(\bar{S} : S) \geq \frac{d(V) - d(S) + \sum_{j \in S} p_j y_j}{Q} - 1 \quad \forall S \subseteq V \setminus \{0\} : d(V) - d(S) + p(S) > Q
\] (16)

Where

\[
x(\bar{S} : S) = \sum_{i \in \bar{S}} \sum_{j \in S} x_{ij} \text{ for any set } S \subseteq V \setminus \{0\} \text{ and } \bar{S} = V \setminus S \setminus \{0\}
\]

\[
d(S) \text{ and } p(S) \text{ be the sum of the delivery and pickup demands, respectively, for all customers in } S
\]
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 constraint will specify that

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

$$x(\bar{S} : S) \geq \frac{30 - 19 + 23}{30} - 1$$

$$x(\bar{S} : S) \geq 0.13$$
By taking out the $f^d$ and $f^p$ variables from TCNH and incorporating the SECs, ECIs and the capacity constraints we developed a new formulation called Two-index Non-Hamiltonian (TINH).

Since the SECs and capacity constraints are exponential we separate them in a B&C framework as done for the BBNH.

During the B&C we are faced with two different types of nodes:
- Fractional nodes
- Integer nodes
Two-index Non-Hamiltonian Formulation

- In order to take advantage of this we use different separation procedures for each type of node.

- In case of fractional nodes we use the max flow algorithm under a supporting graph to find violations.

- For integer nodes we tailored some algorithms that take advantage of the fact that all values are integer to separate in faster way than the max flow.

- Notice that as for the TCNH formulation the TINH is also exact for the single demand case.

- Therefore from now on let us call this B&C algorithm as `singleDemandB&C`.
The \textit{singleDemandB&C} is susceptible to dropoffs in the optimal solution when having combined demand customers.

In order to address this problem we developed the following algorithms that make use of the \textit{singleDemandB&C} and are able to deliver the optimal solution for the SVDSP:

- Throw Away B&C (TA-B&C)
- 2 Step B&C (2S-B&C)
- Minimal Network of the Duplicates B&C (MinND-B&C)
Throw Away B&C (TA-B&C)

- This is the simplest approach
- During the \textit{singleDemandB&C} when an incumbent solution is found a feasibility check is done and if it has at least one dropoff it is rejected and the process continues
- At the end the solution found will be the optimal for the SVDSP and will not contain dropoffs
During preliminary tests we found out that the \textit{singleDemandB&C} is considerably faster than TA-B&C and finds the optimal solution for several instances.

We take advantage of these observations to develop our second approach:

2 Step B&C (2S-B&C)

- At first we solve the normal \textit{singleDemandB&C} keeping track of the best feasible solution found (\textit{bestFeasible}) and all cuts added.

- In case the solution found has dropoffs we solve the TA-B&C giving \textit{bestFeasible} as an initial solution and installing all the cuts from the first step.
Minimal Network of the Duplicates B&C (MinND-B&C)

- 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 such as the optimal solution will be the optimal for the SVDSP.
Exact algorithms for combined demand

Bruno P. Bruck, Manuel Iori (UNIMORE)  Non Hamiltonian Formulations for the SVDSP
In order to test our approaches we use three sets available in the literature for the SVDSP

- The first set is composed by 63 single demand instances with number of customers equal to 10, 20 and 30. This set was proposed by Sural and Bookbinder and is referred from now on as setSingle1.

- The second one is composed by 74 single demand instances with sizes ranging from 25 to 90 customers. It was proposed by Gutiérrez-Jarpa et al. We refer to this set as setSingle2.

- The third set has 68 combined demand instances and was proposed by Laporte et al. The number of customers range from 15 to 110 and we refer to it setCombined.
The implementations were done in C++ and the models were solved using Cplex 12.6 on a computer running 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 Branch&Cut implemented by Guttierrez et al. However they consider only the symmetric case and only tested their approach with the sets *setSingle1* and *setSingle2*.

Since both of these sets contain only single demand instances there is no need to use any of our exact algorithm.

Therefore we compare their results with our most efficiency formulation, which is TINH.
### Computational experiments

**Table**: Average time in seconds (time) and number of optimal solutions (# opt) for the *setSingle1*

<table>
<thead>
<tr>
<th>n (# instances)</th>
<th>SB</th>
<th>GMO</th>
<th>TINH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time</td>
<td># opt</td>
<td>time</td>
</tr>
<tr>
<td>10 customers (24)</td>
<td>5.47</td>
<td>24</td>
<td>0.04</td>
</tr>
<tr>
<td>20 customers (21)</td>
<td>766.61</td>
<td>17</td>
<td>0.17</td>
</tr>
<tr>
<td>30 customers (18)</td>
<td>1093.54</td>
<td>13</td>
<td>0.56</td>
</tr>
<tr>
<td>all (63)</td>
<td>570.06</td>
<td>54</td>
<td>0.23</td>
</tr>
</tbody>
</table>

- SB = Sural and Bookbinder formulation
- GMO = Branch&Cut of Guttierrez et al
### Computational experiments

**Table:** Average time in seconds (time) and number of optimal solutions (# opt) for the `setSingle2`

<table>
<thead>
<tr>
<th>size</th>
<th>SB</th>
<th>GMO</th>
<th>TINH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time</td>
<td># opt</td>
<td>time</td>
</tr>
<tr>
<td>small (18)</td>
<td>553.44</td>
<td>18</td>
<td>2.22</td>
</tr>
<tr>
<td>medium (39)</td>
<td>13301.41</td>
<td>23</td>
<td>47.03</td>
</tr>
<tr>
<td>large (17)</td>
<td>16528.53</td>
<td>5</td>
<td>3510.76</td>
</tr>
<tr>
<td>all (74)</td>
<td>10941.92</td>
<td>46</td>
<td>831.85</td>
</tr>
</tbody>
</table>

- Small $\rightarrow$ $n$ in $[25, 30]$
- Medium $\rightarrow$ $n$ in $[40, 57]$
- Large $\rightarrow$ $n$ in $[68, 90]$
Table: Average time in seconds (time) and number of optimal solutions (# opt) and average gap (gap) for the *setSingle1*

<table>
<thead>
<tr>
<th>size</th>
<th>TA-B&amp;C</th>
<th>2S-B&amp;C</th>
<th>MinND-B&amp;C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time</td>
<td># opt</td>
<td>gap</td>
</tr>
<tr>
<td>small (28)</td>
<td>0.41</td>
<td>28</td>
<td>0.00</td>
</tr>
<tr>
<td>medium (24)</td>
<td>837.34</td>
<td>21</td>
<td>0.01</td>
</tr>
<tr>
<td>large (16)</td>
<td>1412.08</td>
<td>10</td>
<td>0.27</td>
</tr>
<tr>
<td>all (68)</td>
<td>627.95</td>
<td>59</td>
<td>0.07</td>
</tr>
</tbody>
</table>

- The gap was calculated based on the Upper and Lower bound values found after finishing solving.

- Notice that only 1 instance was not solved to optimality by the MinND-B&C algorithm in the time limit of 1 hour. Currently we are able to solve this instance in about 4.7 hours.
Conclusions

For all single demand instances (sets setSingle1 and setSingle2) our formulation TINH outmatched the others approaches and found the optimal solution for every instance.

- It is worth noticing that we were able to find in less than 3 minutes the optimal solution for the open instance in the setSingle2 while the best approach in the literature failed in 5.8 hours.

For the combined demand set (setCombined) our best algorithm proved to be the MinND-B&C is able to solve to optimality all instances.

- All but one with an average time of 105 seconds for each instance.
- The other one with around 4.7 hours.
Conclusions

- We have created a new set of 48 combined demand instances using the procedure described in the literature.
- This set includes bigger instances (maximum of 200 customers) and also asymmetric ones.
- Preliminary results confirm the efficiency of our MinND-B&C by solving to optimality 40 of these instances.
B.P. Bruck, A.G. dos Santos, and J.E.C. Arroyo.
Hybrid metaheuristic for the single vehicle routing problem with deliveries and selective pickups.

A hybrid heuristic based on general variable neighborhood search for the single vehicle routing problem with deliveries and selective pickups.

I. Gribkovskaia and G. Laporte.
One-to-many-to-one single vehicle pickup and delivery problems.
I. Gribkovskaia, G. Laporte, and A. Shyshou.  
The single vehicle routing problem with deliveries and selective pickups.  

A single vehicle routing problem with fixed delivery and optional collections.  

H. Süral and J.H. Bookbinder.  
The single-vehicle routing problem with unrestricted backhauls.  