A real-life Multi Depot Multi Period Vehicle Routing Problem with a Heterogeneous Fleet: Formulation and Adaptive Large Neighborhood Search based Metaheuristic

Simona Mancini

(Transportation Research Part C, 2015)

2016.01.07

Joo Ga-Hyun
1. Introduction

- **Vehicle Routing Problem (VRP)**
  - Design of the optimal routes used by a fleet of vehicles, based at one or more depots, to serve a set of customers with known demand

- **The rich vehicle routing problems**
  - Routing problems are of concern in real life whenever things needs to be transported from one place to another.
  - These real life routing problem usually include complications that are not considered by the basic CVRP.
  - Most complications are related to the following aspects
    - Planning horizon
    - Customer
    - Depot
    - Vehicle
    - Driver
    - Objective
    - Uncertainty
1. Introduction

- Multi Period Vehicle Routing Problem (PVRP)
  - Service occurs in several days of a time horizon
  - Customers may be served according to different day combinations (day{1,3} or {2,4})

- Multi Depot Vehicle Routing Problem (MDVRP)
  - Having several depots
  - Determining the routes for several vehicles from multiple depots to a set of customers and then return to the same depots

- The Heterogeneous Fleet Vehicle Routing Problem (HVRP)
  - Different type of fleet
  - Fleet size (+ fixed cost)

North of Italy
- Truin (TU)
- Milan (MI)
- Venice (VE)
  - First day delivery D1
  - Second day delivery D2
  - Third day delivery D3
  - Vehicle types V1, V2, V3, V4
1. Introduction

- In this paper
  - Objective with minimizing the total delivery cost.
  - Considering a real life context with the Multi Depot Multi Period Vehicle Routing Problem with Heterogeneous Fleet
    - Multi Period problem, each customer must be served in one of the time slots.
      - Customers need to receive the goods within a given deadline.
    - Multi Depot problem, it considered routes that could end at different depot.
      - More convenient to end a route in a different depot from the starting one, avoiding a long trip back to the depot
      - A better exploitation of the available fleet would be obtained and get much better solution
      - All the period must be considered together at the same time, the final depot for each vehicle must be chosen taking into account the activity to be exploited in the following day.
    - Heterogeneous fleet problem,
      - Average speed is known, easy to transform driving cost.

North of Italy
- Truin (TU)
- Milan (MI)
- Venice (VE)
  - First day delivery D1
  - Second day delivery D2
  - Third day delivery D3
  - Vehicle types V1, V2

Fig. 2. Solution of the Multi-Period Multi-Depot without the constraint of a vehicle having to return to the starting depot.
2. Problem Description

- Objective function
  - minimizing the total delivery cost

- Decision variable
  - Assignment of vehicle route

- Solution approach
  - The adaptive large neighborhood search based meta heuristic
2. Problem Description

- MDMPVRPHF definitions
  - The serving set of customers $I=\{0, 1, 2, \ldots, I_{\text{max}}\}$
    - Each customer $i$ requires a quantity of goods $q_i$
    - Customer $i$ must receive his order before day $s$ or according to the availability of the goods
  - Several depots $d$, the set of depots $D$
    - Depot $d$ where the vehicle is located are known for each vehicle
    - If $d$ is the arrival depot of the routed followed by $v$ on day $s-1$
      - A vehicle $v$ is supposed to be located at depot $d$ on day $s$
    - If $v$ has not been used on day $s-1$
      - Supposed to be located at the arrival depot of its last followed route
    - If it has not been used yet at the depot
      - It was located at the beginning of the time horizon.
  - Delivery may be carried out by vehicles that are compatible with the customer’s request during a time-slots(day) in which the customer is available.
  - Different capacity and characteristic for each vehicle
    - Large standard, small standard, refrigerated, freezer, etc.
    - A vehicle-customer compatibility matrix is computed in which the customer’s need and vehicle characteristics are matched
2. Problem Description

Notation

\begin{align*}
\text{Input data:} \\
I &= \{1 \ldots I_{\text{max}}\} & \text{Set of customers} \\
D &= \{I_{\text{max}} + 1 \ldots D_{\text{max}}\} & \text{Set of depots} \\
N &= \{1 \ldots I_{\text{max}} + D_{\text{max}}\} & \text{Set of nodes } (N = I \cup D) \\
K &= \{1 \ldots K_{\text{max}}\} & \text{Set of routes} \\
V &= \{1 \ldots V_{\text{max}}\} & \text{Set of vehicles} \\
S &= \{1 \ldots S_{\text{max}}\} & \text{Set of days} \\
M & & \text{Very large constant} \\
\epsilon & & \text{Very small constant} \\
\alpha & & \text{Maximum route duration} \\
v & & \text{Average speed (expressed in km/h)} \\
q_i & & \text{Customer's demand } i \\
r_{ij} & & \text{Distance between node } i \text{ and node } j \\
v_k & & \text{Vehicle associated to route } k \\
t_k & & \text{Day on which route } k \text{ is scheduled} \\
C_k & & \text{Capacity of route } k \text{ (i.e., capacity of vehicle } v \text{ performing route } k) \\
F_{id} & & \text{Equal to 1 if customer } i \text{ can be served by depot } d \\
E_{iv} & & \text{Equal to 1 if customer } i \text{ can be served by vehicle } v \\
G_{is} & & \text{Equal to 1 if customer } i \text{ can be served on day } s \\
I_{kd} & & \text{Equal to 1 if vehicle } v \text{ associated to route } k \text{ is initially located at depot } d \\
\mu_v & & \text{cost per minute of usage for vehicle } v \\
\text{Variables:} \\
X_{ijk} & & \text{Boolean variable equal to 1 if arc } ij \text{ is used by route } k \\
Y_{ijk} & & \text{Boolean variable equal to 1 if customer } i \text{ is served by route } k \\
Z_k & & \text{Boolean variable equal to 1 if route } k \text{ belongs to the solution} \\
T_i & & \text{Time at which node } i \text{ is visited} \\
L_{kd} & & \text{Boolean variable equal to 1 if route } k \text{ can start from depot } d \\
W_k & & \text{Duration of route } k \\
\lambda & & \text{Objective function}
\end{align*}
2. Problem Description

- A Mixed Integer Programming model for MDMPVRPHF

\[
\min \Lambda = \sum_{i \in N} \sum_{j \in N} \sum_{k \in N} \frac{v}{60} T_{ij} \mu_{ik} X_{ijk} \\
\text{s.t.} \\
\sum_{j \in N \setminus i} X_{jk} = Y_{ik} \quad \forall i \in I, \forall k \in K \\
\sum_{j \in N \setminus i} X_{jk} = Y_{ik} \quad \forall i \in I, \forall k \in K \\
\sum_{k \in K} Y_{ik} = 1 \quad \forall i \in I \\
\sum_{j \in N \setminus i} X_{dj} \leq L_{kd} \quad \forall k \in K, \forall d \in D \\
\sum_{j \in N \setminus i} \sum_{d \in D} X_{dj} = Z_k \quad \forall k \in K \\
\sum_{j \in N \setminus i} \sum_{d \in D} X_{jk} = Z_k \quad \forall k \in K \\
\sum_{i \in I} Y_{ik} \leq M \cdot Z_k \quad \forall k \in K \\
X_{ik} = 0 \quad \forall i \in I, \forall k \in K \\
T_i \geq T_j + \frac{v}{60} r_g X_{gjk} - M \cdot (1 - X_{jk}) \quad \forall i \in I, \forall j \in N, \forall k \in K \\
T_d = 0 \quad \forall d \in D \\
\sum_{i \in I} q_i Y_{ik} \leq C_k \quad \forall k \in K
\]
2. Problem Description

- A Mixed Integer Programming model for MDMPVRPHF

\[
W_k \leq \alpha \quad \forall k \in K
\]

\[
W_k = \sum_{i \in N} \sum_{j \in N} \frac{v}{50} r_{ij} X_{ijk} \quad \forall k \in K
\]

\[
L_{kd} = l_{kd} \quad \forall k \in K \cap t_k = 1 \quad \forall d \in D
\]

\[
\sum_{d \in D} L_{kd} \leq 1 \quad \forall k \in K
\]

\[
L_{kd} = l_{kd}(1 - Z_{k-V_{\max}}) + \sum_{j \in N} \sum_{w \in K} \frac{X_{jdw}}{p_{w-k} p_{w-k}} \quad \forall k \in K \cap t(k) \geq 1, \forall d \in D
\]

\[
L_{kd} \leq \sum_{j \in N} \sum_{w \in K} X_{jdw} + l_{kd} \quad \forall k \in K \cap t(k) \geq 1, \forall d \in D
\]

\[
L_{kd} \geq \sum_{j \in N} \sum_{w \in K} X_{jdw} \quad \forall k \in K \cap t(k) \geq 1, \forall d \in D
\]

\[
Y_{ik} \leq \sum_{d \in D} F_{id} L_{kd} \quad \forall i \in I, \forall k \in K
\]

\[
Y_{ik} \leq E_{ip_k} \quad \forall i \in I, \forall k \in K
\]

\[
Y_{ik} \leq G_{ik} \quad \forall i \in I, \forall k \in K
\]

\[
X_{ijk} \in \{0, 1\} \quad \forall i \in N, \forall j \in N, \forall k \in K
\]

\[
Y_{ik} \in \{0, 1\} \quad \forall i \in I, \forall k \in K
\]

\[
Z_k \in \{0, 1\} \quad \forall k \in K
\]

\[
L_{kd} \in \{0, 1\} \quad \forall k \in K, \forall d \in D
\]
3. The Adaptive Large Neighborhood Search

- The Large Neighborhood Search
  - The large neighborhood search (LNS) metaheuristic was proposed by Shaw (1998).
  - Defined implicitly by a destroy and a repair method
    - A destroy operator
      - Destructs part of the current solution
      - Contains an element of stochasticity such that different parts of the solution are destroyed.
      - A destroy operator consists in breaking down k routes or in removing a fixed percentage of the arcs in the current solution.
    - A repair operator
      - Rebuilds the destroyed solution by inserting removed customers, starting from partially destroyed one, using greedy heuristic.

```
Algorithm 1: Large neighborhood search
1: input: a feasible solution $x$
2: $x^b = x$
3: repeat
4: $x' = r(d(x))$
5: if accept($x', x$) then
6:   $x = x'$
7: end if
8: if $c(x') < c(x^b)$ then
9:   $x^b = x'$
10: end if
11: until stop criterion is met
12: return $x^b$
```
The Adaptive Large Neighborhood Search (ALNS)

- Extend the LNS heuristic by allowing multiple destroy and repair operators to be used within the same search.
- A destroy operator is chosen at each iteration, on the basis of probabilities.
- The choice of neighborhood to use is controlled dynamically using recorded performance (=probabilities) of the neighborhoods.

Algorithm 1. An ALNS for the MDMPVRPHF.

- run the model with a time limit equal to TIMELIMIT and take the best found solution \( S^0 \) as the initial solution
- set the current solution \( S \) equal to \( S^0 \)
- repeat
  - select a destroy operator \( \omega_k \)
  - select a subset \( P \) of the customers using \( \omega_k \)
  - for all \( i \in I - P \) do
    - assign customer \( i \) to route \( k \) to which it is assigned in \( S \) adding constraints \( (Y_{ik} = Y_{ik}^S) \)
  - end for
  - run the model again until the optimal solution is reached
  - if the newly obtained solution \( S' \) is better than \( S \) then
    - set the current solution \( S \) equal to \( S' \)
  - else
    - update probabilities
  - end if
- until maximum number of iterations, \( MAXITER \), is reached or no improvement has been obtained for \( MAXNOIMPROVE \) iterations
3. The Adaptive Large Neighborhood Search

- **Algorithm description**
  - **Initial solution**
    - Taking the best feasible solution obtained by the model (MIP) within a given short time
      - TIMELIMIT as the initial solution
    - Result obtained running the model on XPRESS 7.3 (FICO, opt package)
  - **Proper value** $p$
    - The following additional constraints are imposed
      \[ Y_{ik} = Y_{ik}^* \quad \forall i \in I - P \]
      $Y_{ik}$: the customers-to-route assignment variables
      $P$: the set of selected customers (selected given by operators)
    - $p$ customers are selected at each iteration of the algorithm
      - First, All the other $|I| - P$ customers are assigned to the same route to which they are assigned in the current solution
      - Second, the selected $p$ ones are left free to be assigned to any route.
    - Small value of $p \rightarrow$ the risk of remaining trapped in local minima
    - Large value of $p \rightarrow$ cannot be easily explored in a short computational time
3. The Adaptive Large Neighborhood Search

- Algorithm description
  - Neighborhood generation method
    - Combining different neighborhood strategies
    - Four destroy operators are described used to select the customers
  - Destroy operators descriptor
    - Random Removal (RR)
      - Simply selecting \( p \) customers at random
    - Cluster Removal (CR)
      - Selecting a cluster of customers
      - Randomly selecting a customers \( p^* \) and removes him/her and the nearest \( p - 1 \)
    - Pair Removal (PR)
      - Selecting pairs of very close customers
      - Selecting customers \( \frac{p}{2} \) and for each of customers, \( p^* \), the nearest customers \( p'' \) being chosen
    - Route Destruction (RD)
      - Completely destroying routes
      - A route \( k \) is randomly selected and customers belonging to it are selected until the selected number of customer is equal to \( p \)
  - Initial probabilities are uniformly set as
    \[
    \Pi_{\omega_1} = \Pi_{\omega_2} = \Pi_{\omega_3} = \Pi_{\omega_4} = 0.25
    \]
3. The Adaptive Large Neighborhood Search

- **Algorithm description**
  - **Update probabilities**
    - A destroy operator is chosen at each iteration, given probabilities.
    - During the search process, important for the information on the operator’s previous performance
    - Each time a destroy operator \( w_* \), fail to improve the current best solution
      - Its probability of being chosen, \( \Pi_{w_*} \)
        \[
        \Pi_{w_*} = \Pi_{w_*} - \delta
        \]
      - \( \Pi_\omega = \Pi_\omega + \frac{\delta}{N_\Omega} \) \( \forall a \in \Omega - w_* \)
    - \( \Omega \): the set of destroy operators
    - \( \delta \): the penalty inflicted on the non-successful operator, the large the value
    - The probability updating parameter, \( \delta \) has been set equal to 0.01
  - **Termination condition**
    - MAXITER : a maximum number of iterations
    - MAXNOIMPROVE : a maximum number of iterations without improvement
4. Computational results

- Parameters used in the instance
  - Number of customers : 30
  - Number of depots : 3
  - Number of vehicles : 6
  - Number of days : 5
  - Number of potential routes : number of vehicles * number of days = 30
  - Maximum route duration : 660
  - Average speed : 80

- Parameters of algorithm
  - Perturbation size $p$ : 10
  - Initial solution time limit $TIMLIMIT$ : 10s
  - Maximum number of iterations $MAXITER$ : 100
  - Maximum number of iterations without improving $MAXNOIMPROVE$ : 10
4. Computational results

- The difference levels of customers-vehicles compatibility and of customer availability
  - The customers-vehicle compatibility
    - High compatibility: around 90% of customer-vehicle assignments are feasible.
    - Medium compatibility: around 80% of customer-vehicle assignments are feasible.
    - Low compatibility: around 60% of customer-vehicle assignments are feasible.
  - The customers availability levels are:
    - High availability: around 95% of customer-day assignments are feasible.
    - Medium availability: around 70% of customer-day assignments are feasible.
    - Low availability: around 30% of customer-day assignments are feasible.
  
  Example)
  Supermarkets food delivery : low compatibility, high availability
  Electronic product : high compatibility, low availability

- These levels represent realistic cases arise in practical applications
- To test the model on instances having different characteristics, in order to recreate realistic situation
4. Computational results

Table 1

Show the best solution (UB) and the lower bound (LB) obtained by the model within time limit of 3600s
- Result obtained running the model on XPRESS 7.3 (FICO, opt package)
- The gap obtained by the model are very large
  - Improving heuristic is required

<table>
<thead>
<tr>
<th>Istanza</th>
<th>V-COMP</th>
<th>DAY-AV</th>
<th>Model-LB</th>
<th>Model-UB</th>
<th>Gap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>High</td>
<td>High</td>
<td>1292.79</td>
<td>1902.85</td>
<td>47.19</td>
</tr>
<tr>
<td>12</td>
<td>High</td>
<td>Medium</td>
<td>1346.74</td>
<td>1848.79</td>
<td>37.28</td>
</tr>
<tr>
<td>13</td>
<td>High</td>
<td>Low</td>
<td>1359</td>
<td>1715</td>
<td>26.20</td>
</tr>
<tr>
<td>14</td>
<td>Medium</td>
<td>High</td>
<td>1341.11</td>
<td>1913.14</td>
<td>42.65</td>
</tr>
<tr>
<td>15</td>
<td>Medium</td>
<td>Medium</td>
<td>1364.49</td>
<td>1779.92</td>
<td>30.45</td>
</tr>
<tr>
<td>16</td>
<td>Medium</td>
<td>Low</td>
<td>1384.01</td>
<td>1673.03</td>
<td>20.88</td>
</tr>
<tr>
<td>17</td>
<td>Low</td>
<td>High</td>
<td>1422.71</td>
<td>1743.33</td>
<td>22.54</td>
</tr>
<tr>
<td>18</td>
<td>Low</td>
<td>Medium</td>
<td>1650.32</td>
<td>1811.37</td>
<td>9.75</td>
</tr>
<tr>
<td>19</td>
<td>Low</td>
<td>Low</td>
<td>1700.01</td>
<td>1796.14</td>
<td>5.65</td>
</tr>
</tbody>
</table>
4. Computational results

- Testing result
  - Table 2.
    - Each operator has been tested separately, the probability of choosing other operators is kept equal to zero => possible to evaluate the behavior of the single operator
    - The algorithm presents a random component, each method has been used 10 times on each instance
    - LNS, in this version operators choosing probabilities, assigned following a uniform distribution, are kept constant during the search process
    - Since algorithm presents a random component, each method has been run 10 times on each instance

<table>
<thead>
<tr>
<th>Model</th>
<th>IS</th>
<th>RR</th>
<th>CR</th>
<th>PR</th>
<th>RD</th>
<th>LNS</th>
<th>ALNS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
<td>Avg</td>
<td>Best</td>
<td>Avg</td>
<td>Best</td>
<td>Avg</td>
<td>Avg</td>
</tr>
<tr>
<td>I1</td>
<td>1902.85</td>
<td>2290.65</td>
<td>1626.89</td>
<td>1673.05</td>
<td>1655.22</td>
<td>1706.59</td>
<td>1622.31</td>
</tr>
<tr>
<td>I2</td>
<td>1848.79</td>
<td>1981.54</td>
<td>1544.50</td>
<td>1577.43</td>
<td>1544.50</td>
<td>1554.64</td>
<td>1575.22</td>
</tr>
<tr>
<td>I3</td>
<td>1715.00</td>
<td>2466.64</td>
<td>1649.92</td>
<td>1713.15</td>
<td>1602.29</td>
<td>1645.21</td>
<td>1649.92</td>
</tr>
<tr>
<td>I4</td>
<td>1913.14</td>
<td>1975.44</td>
<td>1574.42</td>
<td>1594.07</td>
<td>1574.42</td>
<td>1614.33</td>
<td>1564.47</td>
</tr>
<tr>
<td>I5</td>
<td>1779.92</td>
<td>2392.48</td>
<td>1720.08</td>
<td>1753.41</td>
<td>1667.61</td>
<td>1737.07</td>
<td>1701.99</td>
</tr>
<tr>
<td>I6</td>
<td>1673.03</td>
<td>2377.26</td>
<td>1582.56</td>
<td>1612.40</td>
<td>1582.56</td>
<td>1601.92</td>
<td>1582.56</td>
</tr>
<tr>
<td>I7</td>
<td>1743.33</td>
<td>2408.86</td>
<td>1564.54</td>
<td>1661.39</td>
<td>1638.60</td>
<td>1659.79</td>
<td>1655.77</td>
</tr>
<tr>
<td>I8</td>
<td>1811.37</td>
<td>2257.23</td>
<td>1809.75</td>
<td>1863.60</td>
<td>1842.67</td>
<td>1851.51</td>
<td>1791.66</td>
</tr>
<tr>
<td>I9</td>
<td>1796.14</td>
<td>2236.91</td>
<td>1791.71</td>
<td>1848.05</td>
<td>1784.62</td>
<td>1826.98</td>
<td>1872.46</td>
</tr>
<tr>
<td>Avg</td>
<td>1798.17</td>
<td>2280.44</td>
<td>1660.71</td>
<td>1699.62</td>
<td>1654.72</td>
<td>1688.67</td>
<td>1668.48</td>
</tr>
<tr>
<td>Gap model</td>
<td>7.64%</td>
<td>5.48%</td>
<td>7.98%</td>
<td>6.09%</td>
<td>7.21%</td>
<td>4.04%</td>
<td>9.31%</td>
</tr>
</tbody>
</table>
4. Computational results

- **Parameters tuning**
  - **Table 3**
    - Carried out to tune algorithm parameters like the `TIMLIMIT` for the initial solution.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Initial solution</th>
<th>ALNS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$t = 10$</td>
<td>$t = 30$</td>
</tr>
<tr>
<td></td>
<td>$t = 10$</td>
<td>$t = 30$</td>
</tr>
<tr>
<td>11</td>
<td>2290.65</td>
<td>2290.65</td>
</tr>
<tr>
<td>12</td>
<td>1981.54</td>
<td>1981.54</td>
</tr>
<tr>
<td>13</td>
<td>2466.64</td>
<td>2466.64</td>
</tr>
<tr>
<td>14</td>
<td>1915.44</td>
<td>1837.27</td>
</tr>
<tr>
<td>15</td>
<td>2529.48</td>
<td>1861.88</td>
</tr>
<tr>
<td>16</td>
<td>2377.26</td>
<td>1863.4</td>
</tr>
<tr>
<td>17</td>
<td>2488.86</td>
<td>2072.14</td>
</tr>
<tr>
<td>18</td>
<td>2257.23</td>
<td>2062.63</td>
</tr>
<tr>
<td>19</td>
<td>2236.9</td>
<td>2001.82</td>
</tr>
<tr>
<td>Avg</td>
<td>2280.444</td>
<td>2048.663</td>
</tr>
</tbody>
</table>

- **Table 4**
  - The number of customers to be removed from the solution at each iteration, $p$

<table>
<thead>
<tr>
<th>Instance</th>
<th>$p = 5$</th>
<th>$p = 10$</th>
<th>$p = 15$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
<td>Avg</td>
<td>Best</td>
</tr>
<tr>
<td>11</td>
<td>1545.95</td>
<td>1622.42</td>
<td>1544.50</td>
</tr>
<tr>
<td>12</td>
<td>1544.5</td>
<td>1552.63</td>
<td>1544.50</td>
</tr>
<tr>
<td>13</td>
<td>1551.59</td>
<td>1566.69</td>
<td>1544.50</td>
</tr>
<tr>
<td>14</td>
<td>1557.79</td>
<td>1599.13</td>
<td>1557.79</td>
</tr>
<tr>
<td>15</td>
<td>1618.47</td>
<td>1685.13</td>
<td>1618.47</td>
</tr>
<tr>
<td>16</td>
<td>1582.56</td>
<td>1587.80</td>
<td>1582.56</td>
</tr>
<tr>
<td>17</td>
<td>1638.77</td>
<td>1681.50</td>
<td>1627.52</td>
</tr>
<tr>
<td>18</td>
<td>1796.94</td>
<td>1830.40</td>
<td>1778.85</td>
</tr>
<tr>
<td>19</td>
<td>1798.93</td>
<td>1892.78</td>
<td>1778.85</td>
</tr>
<tr>
<td>Avg</td>
<td>1626.167</td>
<td>1668.719</td>
<td>1619.727</td>
</tr>
</tbody>
</table>
4. Computational results

- **Large size instance**
  - **Table 5**
    - Instance with this characteristics are the most difficult to be solved by the model
      - Most challenging for the ALNS
    - Lower and upper bound obtained by the model within a time limit of 3600s

<table>
<thead>
<tr>
<th>Customers</th>
<th>V-Comp</th>
<th>Day-AV</th>
<th>Model LB</th>
<th>Model UB</th>
<th>Gap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>High</td>
<td>High</td>
<td>619.46</td>
<td>897.15</td>
<td>44.83</td>
</tr>
<tr>
<td>50</td>
<td>High</td>
<td>Medium</td>
<td>664.68</td>
<td>876.38</td>
<td>31.85</td>
</tr>
<tr>
<td>50</td>
<td>High</td>
<td>Low</td>
<td>1667.84</td>
<td>1667.84</td>
<td>0.00</td>
</tr>
<tr>
<td>75</td>
<td>High</td>
<td>High</td>
<td>825.42</td>
<td>2584.52</td>
<td>213.12</td>
</tr>
<tr>
<td>75</td>
<td>High</td>
<td>Medium</td>
<td>867.2</td>
<td>2584.52</td>
<td>198.03</td>
</tr>
<tr>
<td>75</td>
<td>High</td>
<td>Low</td>
<td>1929.56</td>
<td>2337.66</td>
<td>21.15</td>
</tr>
</tbody>
</table>

- **Table 6**
  - ALNS show the high quality performance of ALNS(within an average computational time of 400s)

<table>
<thead>
<tr>
<th>Customers</th>
<th>V-Comp</th>
<th>Day-AV</th>
<th>Model</th>
<th>Init sol</th>
<th>ALNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>High</td>
<td>High</td>
<td>897.15</td>
<td>4066.3</td>
<td>858.13</td>
</tr>
<tr>
<td>50</td>
<td>High</td>
<td>Medium</td>
<td>876.38</td>
<td>3078.19</td>
<td>867.1</td>
</tr>
<tr>
<td>50</td>
<td>High</td>
<td>Low</td>
<td>1667.84</td>
<td>2784.41</td>
<td>1667.84</td>
</tr>
<tr>
<td>75</td>
<td>High</td>
<td>High</td>
<td>2584.52</td>
<td>3048.03</td>
<td>1396.01</td>
</tr>
<tr>
<td>75</td>
<td>High</td>
<td>Medium</td>
<td>2584.52</td>
<td>3048.03</td>
<td>1396.01</td>
</tr>
<tr>
<td>75</td>
<td>High</td>
<td>Low</td>
<td>2337.66</td>
<td>3048.03</td>
<td>2290.63</td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
<td>1824.68</td>
<td>3178.83</td>
<td>1414.29</td>
</tr>
</tbody>
</table>
5. Conclusion

- conclusion
  - The Multi Depot Multi Period Vehicle Routing Problem with Heterogeneous Fleet (MDMPVRPHF) has been introduced and formalized in this paper.
  - A new Vehicle Routing Problem arise in real-life contexts
    - The goal is to carry out delivery operations at the minimum cost
    - Considering respecting constraints due to driver scheduling, customer/vehicle compatibilities and customer availability

- 장점

- 단점
Thank You!

www.themegallery.com