CHAPTER 4

PROBLEM DESCRIPTION AND SOLUTION
METHODOLOGY

4.1 INTRODUCTION

The VRP consists of designing a set of at the most m delivery or collection routes such that (1) each route starts and ends at the depot, (2) each customer is visited exactly once by exactly one vehicle, (3) the total demand of each route does not exceed the vehicle capacity, (4) the total duration of each route (including travel and service times) does not exceed a preset limit, and (5) the total routing cost is minimized.

The VRPTW is one of the important variant of VRP and basic distribution-management problem that can be used to model many real-world problems. This is due to the fact that the VRPTW is a useful abstraction of many real-life problems dealing with distribution of goods or services. Furthermore, finding good solutions to this problem contributes to reducing the transportation and distribution costs of a company.

The following definition can be used to describe the problem:

The VRPTW is concerned with the design of minimum-cost vehicle routes, originating and ending at a central depot. The routes must service a set of customers with known demands. Each customer is to be serviced exactly once during the planning horizon and customers must be
assigned to the vehicles without exceeding vehicle capacities. Furthermore, each customer must be serviced only during allowable delivery times or time windows.

4.2 CONCEPT OF TIME WINDOWS

A time window can be described as a window of opportunity for deliveries. A time window is the period of time during which deliveries can be made to a specific customer \( i \), and has three main characteristics:

- Earliest allowed arrival time, \( e_i \), also referred to as the opening time
- Latest allowed arrival time, \( l_i \), also referred to as the closing time
- Whether the time window is considered soft or hard

Consider the example, illustrated in Figure 4.1, where customer \( i \) requests delivery between 06:30 and 17:00. To distinguish between the actual and the arrival, specified times of the variable \( a_i \) denotes the actual time of arrival at node \( i \). Should the actual arrival time at node \( i \), denoted by \( a_i \), be earlier than the earliest allowed arrival at the node \( e_i \), then the vehicle will incur a waiting time \( w_i \) which can be calculated as \( w_i = \max\{0, e_i - a_i\} \).

![Figure 4.1 Double sided hard time window](image-url)
When both earliest and latest allowed arrival times are stipulated then the time window is referred as double sided time window. If no arrivals are allowed outside the given parameters, the time window is said to be hard as shown in the Figure 4.1. When delivery is allowed outside the specified time window, the time window is said to be soft, and customer $i$ may penalize lateness at penalty cost. Customer $i$ may specify a maximum lateness $L_i^\text{max}$. The example illustrated in Figure 4.2 sees customer $i$ specifying a time window between 06:30 and 15:30. The customer will, however, allow late deliveries until 17:00.

![Figure 4.2 Soft time window](image)

A hard time window is therefore a special type of soft time window where $L_i^\text{max} = 0$. Should a vehicle arrive after the latest allowed arrival time, $l_i$, but prior to the maximum lateness, $L_i^\text{max}$, the lateness at node $i$, $L_i$ can be calculated as $L_i = \max\{0, a_i - l_i\} a_i < L_i^\text{max}$. The lateness is penalized by introducing a penalty term to the VRP objective function.

The time window for the depot, node 0, can be specified. The case illustrated in Figure 4.3 sees the depot specifying operating hours (time window) from 06:00 to 18:00, while the first customer on the route, customer 1, specifies a time window between 07:00 and 09:00, and the last customer N, requests delivery between 15:00 and 17:00.
4.3 VRPTW PROBLEM DEFINITION

The VRPTW is defined on a graph \((N, A)\). The node set \(N\) consists of the set of customers, denoted by \(C\), and the node 0, which represents the depot. There are \(n\) customers in set \(C\) and are denoted by \(1, 2, \ldots, n\). The arc set \(A\) corresponds to possible connections between the nodes. All routes start and end at 0. A distance \(d_{ij}\) and time \(t_{ij}\) are associated with each arc \((i, j)\) of the network. The set of vehicles are denoted by \(V\). Each vehicle has a given capacity \(D_k\) and maximum route duration \(T_k\). Each customer has a demand \(d_i\), \(i \in C\). For each customer, the start of the service must be within a given time interval, called a time window, \([e_i, l_i]\), \(i \in C\). Vehicles must also leave the depot within the time window \([e_0, l_0]\). A vehicle is permitted to arrive before the opening of the time window, and wait at no cost until service becomes possible, but it is not permitted to arrive after the deadline. Since waiting time is permitted at no cost, it may be assumed without loss of generality that \(e_0 = l_0 = 0\); that is, all routes start at time 0. Let \(x_{ijk}\) denote whether the vehicle \(k\) travels from \(i\) to \(j\) and \(s_{ik}\) indicates the Start Service time of vehicle \(k\) at customer \(i\).
4.4 NOTATIONS USED IN THE MODEL FOR VRPTW

\[ x_{ijk} \in (0,1) \text{ 0 if there is no arc from nodes } i \text{ to } j, \text{ and 1 otherwise.} \]

\[ V \quad \text{total number of vehicles} \]

\[ C \quad \text{total number of nodes (customers)} \]

\[ d_{ij} \quad \text{Euclidean distance between node } i \text{ and node } j \]

\[ t_{ij} \quad \text{travel time between node } i \text{ and } j \]

\[ d_i \quad \text{demand at node } i \]

\[ D_k \quad \text{capacity of vehicle } k \]

\[ e_i \quad \text{earliest arrival time at node } i \]

\[ l_i \quad \text{latest arrival time at node } i \]

\[ s_i \quad \text{service time at node } i \]

\[ T_k \quad \text{maximum route time allowed for vehicle } k \]

\[ s_{ik} \quad \text{start service time of vehicle } k \text{ at customer } i \]

\[ w_i \quad \text{wait time at node } i \]

4.5 MATHEMATICAL FORMULATION FOR VRPTW

\[ \text{Min} \quad \sum_{k=1}^{V} \sum_{i=1}^{n} \sum_{j=0}^{n} x_{ijk} d_{ij} \quad \text{(4.1)} \]

The objective function (4.1) which states that distance should be minimized.
\[ \text{Min} \sum_{j=1}^{n} \sum_{k=1}^{V} x_{ijk}, \quad i = 0 \quad (4.2) \]

The objective function (4.2) which states that total number of vehicles used should be minimised.

\[ \sum_{k \in V} \sum_{j \in N} x_{ijk} = 1, \quad \forall i \in C \quad (4.3) \]

Constraint (4.3) indicates that each customer should be assigned to exactly one vehicle.

\[ \sum_{j \in C} x_{0jk} = 1, \quad \forall k \in V \quad (4.4) \]

Constraint (4.4) states that each vehicle must begin its tour from the depot.

\[ \sum_{i \in C} x_{i0k} = 1, \quad \forall k \in V \quad (4.5) \]

\[ \sum_{i \in N} x_{ik} - \sum_{j \in N} x_{ijk} = 0, \quad \forall t \in C \quad (4.6) \]

Constraint (4.6) implies that a vehicle can start from a customer location only if it has reached that location.

\[ \sum_{i \in C} (d_i \sum_{j \in N} x_{ijk}) \leq D_k, \quad \forall k \in V \quad (4.7) \]

Constraint (4.7) indicates that the total demand to be serviced by a vehicle must be less than or equal to its capacity.
\[
\sum_{i \in C} \sum_{j \in C} x_{i,k} \left( t_{ij} + s_i + w_j \right) \leq T_k, \forall k \in V
\]  

(4.8)

Constraint (4.8) ensures maximum travel time.

\[
x_{i,k} \left( s_{i,k} + t_{ij} - s_{j,k} \right) \leq 0, \quad \forall (i, j) \in A, \forall k \in V
\]

(4.9)

Constraint (4.9) specifies that if i-j is a part of the route, then the start service time at j must be greater than the sum of the start service time at i, the service time at i and the total time taken to travel from i to j.

\[
e_i \leq s_{i,k} \leq l_i, \quad \forall i \in C, \forall k \in V
\]

(4.10)

Constraint (4.10) maintains that the start service time for any customer must fall within the time windows specified by him.

\[
x_{i,j} \in \{0,1\}, \quad \forall (i, j) \in A, \forall k \in V
\]

(4.11)

Constraint (4.11) specifies the integrality of the existence of path from i to j.

A graphical representation of the routing solution for VRPTW is given in Figure 4.4. In this example, there is one depot (D) with three routes with different number of customers serviced on each route.
Figure 4.4 Example of a routing solution for VRPTW

4.6 MDVRPTW PROBLEM DEFINITION

Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW) is a variant of the well known Vehicle Routing Problem with Time Windows (VRPTW) where instead of one depot, several depots at different locations and associated fleets are to be considered. Thus, the problem is defined on a complete graph $G = (V, A)$, where $V = \{v_1, v_2, \ldots, v_m, v_{m+1}, \ldots, v_{m+n}\}$ is the vertex set and $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ is the arc set. Vertices $v_1$ to $v_m$ correspond to the $m$ depots, while the vertices $v_{m+1}$ to $v_{m+n}$ represent the $n$ customers. Each vertex $v_i \in V$ has several nonnegative weights associated with it, namely, a demand $q_i$, a service time $s_i$, as well as an earliest $e_i$ and latest $l_i$ possible start time for the service, which defines the time window $[e_i, l_i]$. For the depots these time windows correspond to the opening hours. Further, the depot vertices $v_1$ to $v_m$ feature no demands and service times, i.e. $q_i = s_i = 0, \ \forall \ i \in \{v_1, v_2, \ldots, v_m\}$. A distance $d_{ij}$ and travel time $t_{ij}$ are associated with each arc $(i, j) \in A$ of the network. Finally, a fleet of $K$ vehicles is located at the $m$ depots. Each vehicle $k$ is associated with a
non-negative capacity $D_k$ and nonnegative maximum route duration $T_k$. Based on this graph, the MDVRPTW consists of building $K$ vehicle routes such that

a) each vehicle starts and ends at its home depot,
b) each customer is served by one and only one vehicle,
c) the total load and duration of vehicle $k$ does not exceed $D_k$ and $T_k$ respectively,
d) the service at each customer $i$ begins within the associated time window $[e_i, l_i]$ and
e) each vehicle route starts and ends within the time window of its depot.

### 4.7 NOTATIONS USED IN THE MODEL FOR MDVRPTW

to possible connections between nodes

- $d_{ij}$: euclidean distance between node $i$ and node $j$
- $t_{ij}$: travel time between node $i$ and node $j$
- $q_i$: demand at node $i$
- $D_k$: capacity of vehicle $k$
- $e_i$: earliest arrival time at node $i$
- $l_i$: latest arrival time at node $i$
- $s_i$: service time at node $i$
- $T_k$: maximum route time allowed for vehicle $k$
- $a_{ik}$: arrival time of vehicle $k$ at node $i$
- $w_i$: wait time at node $i$

### 4.8 MATHEMATICAL FORMULATION FOR MDVRPTW
\[
\text{Min } \sum_{k=1}^{K} \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ijk} d_{ij} \quad (4.12)
\]

The objective function (4.12) which states that distance should be minimised.

\[
\text{Min } \sum_{j=1}^{n} \sum_{k=1}^{K} \sum_{i=1}^{n} x_{ijk}, \quad i = 0 \quad (4.13)
\]

The objective function (4.13) which states that total number of vehicles used should be minimised.

\[
\sum_{k=1}^{K} \sum_{j=1}^{n} x_{ijk} = 1, \quad \forall i \in n \quad (4.14)
\]

Constraint (4.14) states that each customer should be assigned to exactly one vehicle.

\[
\sum_{j=1}^{n} x_{ijk} = 1, \quad \forall i \in m, \forall k \in K, j \in n \quad (4.15)
\]

Constraint (4.15) states that each vehicle must begin its tour from the depot

\[
\sum_{i=1}^{n} x_{ijk} = 1, \quad \forall j \in m, \forall k \in K, i \in n \quad (4.16)
\]

Constraint (4.16) denotes that each vehicle must end its tour at the depot.

\[
\sum_{i=1}^{n} x_{itk} - \sum_{j=1}^{n} x_{ijk} = 0, \quad \forall t \in n, \forall k \in K \quad (4.17)
\]
Constraint (4.17) implies that a vehicle can start from a customer location only if it has reached that location

\[ \sum_{j=1}^{n} x_{ijk} - \sum_{j=1}^{n} x_{jik} = 0, \quad \forall i \in m, \forall k \in K \] (4.18)

Constraint (4.18) ensures each vehicle must end at the same depot it starts from

\[ \sum_{i=1}^{n} (q_i \sum_{j=1}^{n} x_{ijk}) \leq D_k, \quad \forall k \in K \] (4.19)

Constraint (4.19) ensures that the total demand to be serviced by a vehicle must be less than or equal to its capacity

\[ \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ijk} (t_{ij} + s_i + w_j) \leq T_k, \quad \forall k \in K \] (4.20)

Constraint (4.20) ensures maximum travel time

\[ x_{ijk} (a_{ik} + s_i + t_{ij} - a_{jk}) \leq 0, \quad \forall (i, j) \in A, \forall k \in K \] (4.21)

Constraint (4.21) specifies that if i-j is a part of the route, then the start service time at j must be greater than the sum of the start service time at i, the service time at i and the total time taken to travel from i to j.

\[ e_i \leq a_{ik} \leq l_i, \quad \forall i \in n, \forall k \in K \] (4.22)

Constraint (4.22) ensures that the time windows are observed and constraint

\[ x_{ijk} \in \{0,1\}, \quad \forall (i, j) \in A, \forall k \in K \] (4.23)
Constraint (4.23) specifies the integrality of the existence of path from i to j.

A graphical representation of the routing solution for MDVRPTW is given in Figure 4.5. In this example, there are 3 depots (D1, D2 and D3) each with two routes, with a different number of customers serviced on each route.

![Diagram of routing solution](image)

Figure 4.5 Example of a routing solution for MDVRPTW

4.9 COMPUTATION COMPLEXITY OF ROUTING PROBLEMS

Routing problems are simple to state in terms of describing them in words. But they are very complex in terms of providing a suitable
mathematical formulation and a valid procedure to solve them. The solution procedure for classical TSP and VRP are very complex in terms of computational effort required to solve an instance of a problem. The computational complexity of an algorithm to solve a problem is the worst case computational effort taken over all instances of the same size of the problem and is expressed as a function of the problem size. Whenever this function is polynomial, the algorithm is considered to be good. For many combinatorial problems such as TSP and VRP, the computational complexity of all the known exact algorithms grows exponentially as a function of the size of the problem.

The TSP and VRP belong to the class of NP problems. VRPTW also has been proved to be NP-hard. It is estimated that an average VRPTW has $N^2 \times K$ decision variables, where $N$ is the number of nodes including the depot and $K$ is the number of vehicles used. An exhaustive search has to investigate $2^{N^2 \times K}$ different combinations trying to satisfy a total of $N^2 K + NK + 3K + N + 1$ constraints. With the present knowledge it is believed that the problems in NP hard are unlikely to have any good algorithm. Thus, for many combinatorial optimization problems in the class of NP hard, researchers have focused their effort on the following:

- development and implementation of an exact algorithm exploiting the modern computing power to solve large size problem as possible, within a reasonable computational effort;
- development and implementation of heuristic algorithms so as to produce near optimal solutions quickly;
- development and implementation of metaheuristic algorithms so as to produce near optimal solutions quickly.
4.10 SOLUTION METHODOLOGY

A solution to the MDVRPTW is obtained in two phases namely (a) Clustering and (b) Routing. For SDVRPTW, clustering is not required.

4.10.1 Clustering

The clustering phase deals with assigning customers to the depots in a multi-depot environment. Partitioning Around Medoids (PAM) is found to be suitable for the clustering technique for MDVRPTW. PAM works, by first selecting m out of n total objects that are the closest (according to the distance matrix) to the remaining (n-m) objects. The fitness of these medoids is calculated by placing the remaining (n-m) objects in a group according to the nearest medoid and summing up all the distances of the group members from this medoid.

These m selected objects are the initial medoids. A swapping procedure is then applied until there is no improvement in fitness. Swapping involves generating all the possible medoid and non-medoid pairs, evaluating the fitness of each pair, and then performing the swap that improves fitness, the most. The pseudo code for PAM is given below:

1. Construct m initial medoids that minimize $\sum_{i=1}^{n} w_{i,j}$
2. for i = 1 to iterations
3. for all object pairs, (i,j), where i is a medoid and j is not a medoid
4. perform the i,j swap which decreases the fitness the most
5. end for
6. end for
7. allocate each medoid to a new group
8. allocate the non-medoids to their nearest medoid

Since Time windows are also taken into account, a weighted distance ($W_{ij}$) is used to calculate the distance between $(x_i, y_i)$ and $(x_j, y_j)$ as follows

Weighted Distance $W_{ij} = \alpha \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} + \beta |t_i - t_j|$  \hspace{1cm} (4.24)

$t_i = \left( \frac{e_i + l_i}{2} \right)$  \hspace{1cm} (4.25)

where

- $x_i, y_i$ - co-ordinates of node $i$
- $t_i$ - mean time window
- $e_i$ - earliest arrival time at node $i$
- $l_i$ - latest arrival time at node $i$
- $\alpha$ - weight assigned to distance
- $\beta$ - weight assigned to time window

$0 \leq \alpha \leq 1; 0 \leq \beta \leq 1$ and $\alpha + \beta = 1$  \hspace{1cm} (4.26)

The PAM procedure adopted for MDVRPTW to form clustering of customers is explained in Figure 4.6.
4.10.2 Routing

After allocating customers to the depot, it is required to determine the best possible sequence in which customers in a given depot are covered by the vehicle. This process is performed in the routing phase.

Routing is done by the proposed Genetic algorithm (PGA), Ant colony optimization (PACO), Simulated Annealing (PSA) and Metaheuristic for Randomized Priority Search Meta RaPS (PMR) approaches. Detailed descriptions of these approaches are provided in chapters 5-8.

4.11 BENCHMARK PROBLEMS USED

The different heuristics developed for the single depot vehicle routing problem is tested on 56 Solomon’s VRPTW benchmark problems (Solomon, 1987). Each of these problems has 100 customers and the travel time between nodes is equal to the Euclidean distance. The 56 test problems
are grouped into six problem types. The details of the instances are provided in Table 4.1. Problem sets R1, C1, and RC1 have a shorter scheduling horizon and allow fewer customers per route. Problem sets R2, C2, and RC2 have a longer scheduling horizon and allow a larger number of customers per route.

Table 4.1 Details of Solomon VRPTW data set

<table>
<thead>
<tr>
<th>Type of Problem</th>
<th>No. of test cases</th>
<th>Description of data set</th>
</tr>
</thead>
</table>
| R1              | 12                | Vehicle capacity is small  
|                 |                   | Spatial distribution of the customer is uniformly distributed  
|                 |                   | Service duration is small  
|                 |                   | Width of servicing time window varies |
| R2              | 11                | Vehicle capacity is large  
|                 |                   | Spatial distribution of the customer is uniformly distributed  
|                 |                   | Service duration is small  
|                 |                   | Width of servicing time window varies |
| C1              | 9                 | Vehicle capacity is small  
|                 |                   | Spatial distribution of the customer is clustered  
|                 |                   | Service duration is large  
|                 |                   | Width of servicing time window varies |
| C2              | 8                 | Vehicle capacity is large  
|                 |                   | Spatial distribution of the customer is clustered  
|                 |                   | Service duration is large  
|                 |                   | Width of servicing time window varies |
| RC1             | 8                 | Vehicle capacity is small  
|                 |                   | Spatial distribution of the customer is clustered  
|                 |                   | Service duration is small  
|                 |                   | Width of servicing time window varies |
| RC2             | 8                 | Vehicle capacity is large  
|                 |                   | Service duration is small  
|                 |                   | Spatial distribution of the customer is clustered  
|                 |                   | Width of servicing time window varies |
In order to evaluate the different heuristics developed for MDVRPTW, the algorithms are tested on the benchmark instances for MDVRPTW provided by Cordeau et al (2001). This consists of 20 instances. Based on customer sizes as well as time window tightness the twenty instances are divided into PR1 (pr01 to pr10) and PR2 (pr11 to pr20) groups. The main characteristics like number of customers, number of depots, capacity of vehicle (number of units can handle) and other details of the test problems are summarized in Table 4.2.

Table 4.2 Details of MDVRPTW benchmark instances

<table>
<thead>
<tr>
<th>Problem. No</th>
<th>No. of customers</th>
<th>No. of depots</th>
<th>No. of available vehicles at each depot</th>
<th>Maximum route duration</th>
<th>Capacity of the vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr01</td>
<td>48</td>
<td>4</td>
<td>2</td>
<td>500</td>
<td>200</td>
</tr>
<tr>
<td>Pr02</td>
<td>96</td>
<td>4</td>
<td>3</td>
<td>480</td>
<td>195</td>
</tr>
<tr>
<td>Pr03</td>
<td>144</td>
<td>4</td>
<td>4</td>
<td>460</td>
<td>150</td>
</tr>
<tr>
<td>Pr04</td>
<td>192</td>
<td>4</td>
<td>5</td>
<td>440</td>
<td>185</td>
</tr>
<tr>
<td>Pr05</td>
<td>240</td>
<td>4</td>
<td>6</td>
<td>420</td>
<td>180</td>
</tr>
<tr>
<td>Pr06</td>
<td>288</td>
<td>4</td>
<td>7</td>
<td>400</td>
<td>175</td>
</tr>
<tr>
<td>Pr07</td>
<td>72</td>
<td>6</td>
<td>2</td>
<td>500</td>
<td>200</td>
</tr>
<tr>
<td>Pr08</td>
<td>144</td>
<td>6</td>
<td>3</td>
<td>475</td>
<td>190</td>
</tr>
<tr>
<td>Pr09</td>
<td>216</td>
<td>6</td>
<td>4</td>
<td>450</td>
<td>180</td>
</tr>
<tr>
<td>Pr10</td>
<td>288</td>
<td>6</td>
<td>5</td>
<td>425</td>
<td>170</td>
</tr>
<tr>
<td>Pr11</td>
<td>48</td>
<td>4</td>
<td>1</td>
<td>500</td>
<td>200</td>
</tr>
<tr>
<td>Pr12</td>
<td>96</td>
<td>4</td>
<td>2</td>
<td>480</td>
<td>195</td>
</tr>
<tr>
<td>Pr13</td>
<td>144</td>
<td>4</td>
<td>3</td>
<td>460</td>
<td>150</td>
</tr>
<tr>
<td>Pr14</td>
<td>192</td>
<td>4</td>
<td>4</td>
<td>440</td>
<td>185</td>
</tr>
<tr>
<td>Pr15</td>
<td>240</td>
<td>4</td>
<td>5</td>
<td>420</td>
<td>180</td>
</tr>
<tr>
<td>Pr16</td>
<td>288</td>
<td>4</td>
<td>6</td>
<td>400</td>
<td>175</td>
</tr>
<tr>
<td>Pr17</td>
<td>72</td>
<td>6</td>
<td>1</td>
<td>500</td>
<td>200</td>
</tr>
<tr>
<td>Pr18</td>
<td>144</td>
<td>6</td>
<td>2</td>
<td>475</td>
<td>190</td>
</tr>
<tr>
<td>Pr19</td>
<td>216</td>
<td>6</td>
<td>3</td>
<td>450</td>
<td>181</td>
</tr>
<tr>
<td>Pr20</td>
<td>288</td>
<td>6</td>
<td>4</td>
<td>425</td>
<td>170</td>
</tr>
</tbody>
</table>
4.12 SUMMARY

In this section the Vehicle Routing Problem with Time Windows and the Multi Depot Vehicle Routing Problem with Time Windows are described and the model for the problem is also presented. Further, the solution methodology and the details about benchmark problems is also discussed.