Lecture – 4

VEHICLE ROUTING AND SCHEDULING

Learning Objective

• To discuss methods of solving vehicle routing and scheduling problems
1. Introduction

Most of our daily requirements are made by the service provider coming to our premises. These services are home delivery of Pizza within 30 minutes, transportation service of office picking all employees or school children, Milkman delivering milk door-to-door and postal/courier services. In such services, service delivery and timely service are very important. These issues mainly require scheduling and routing of service vehicles.

The vehicle routing problem (VRP) is a combinatorial optimization and integer programming problem seeking to service a number of customers with a fleet of vehicles. Often the context is that of delivering goods located at a central depot to customers who have placed orders for such goods. Objective of such problems is to minimize the time and distance traveled. Many methods have been developed for searching for good solutions to the problem, but for all but the smallest problems, finding global minimum for the cost function is computationally complex. Hence many good heuristics have been developed for these types of problems which yield good solutions if not optimal solutions.

Example

Consider a milk van delivering milk to four distribution centers (DC) every day morning as shown in the Figure 11.14 below.
Figure 11.14: A depot supplying milk to four distribution centers

The objective of depot is to minimize the total cost of providing the services. This includes the vehicle capital cost, mileage and personnel costs. There can be other service types addressing different objectives as shown in Table 11.19.

Table 11.19: Scheduling or routing objectives of different types of services

<table>
<thead>
<tr>
<th>Service type</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>School bus</td>
<td>Minimize student-minutes on the bus</td>
</tr>
<tr>
<td>Logistics</td>
<td>Minimize the distance travelled to pick up and/or to deliver parcels following optimum routes</td>
</tr>
<tr>
<td>Emergency (ambulance, police, fire)</td>
<td>Minimize response time</td>
</tr>
</tbody>
</table>
The subjective cost associated with failing to provide adequate service to the customer should be considered.

1.1 Characteristics of Routing and Scheduling

We can see Figure 1, which is comprised of nodes and arcs. Various such characteristics of routing problems are discussed below.

Nodes

It consists of five circles called nodes. Node 1 is the depot node from which the vehicle starts and ends. Nodes 2, 3, 4 and 5 represent four distribution centers.

Arcs

The line segments connecting the nodes are called as arcs. Arcs may describe the time, cost or distance required to travel from one node to another. In Figure 1, arcs describe the distance in miles between the DCs.

Arcs may be directed (arrows) or undirected (simple line segments). Arrows represent the direction of travel in the case of routing problems (e.g. one-way streets) or precedence relationships in case of scheduling problems.

Tour

Tour is the route for the vehicle. In Figure 1, travelling nodes in the order as given below

1→2→3→4→5→1

1→5→4→3→2→1

are called tours. The total distance traveled is 53 miles in either case.
**Feasibility**

Minimum-cost solution or any other criterion like time or distance traveled is subject to the tour being feasible. Feasibility implies that

(i) A tour must include all nodes
(ii) A node must be visited only once
(iii) A tour must begin and end at a depot.

**Route:** Sequence in which the nodes (or) arcs are to be visited

**Schedule:** Specifies when each node has to be visited

### 1.2 Classification of Routing and Scheduling Problems

**Traveling Salesman Problem (TSP)** is the simplest case where the nodes have no precedence relationship, travel costs between two nodes are the same regardless of the direction traveled; no delivery time restrictions. Vehicle capacity is not considered. The objective is to find the shortest possible route that visits each city exactly once and returns to the origin city. TSP has several applications even in planning, logistics, and the manufacture of microchips.

**Multiple Traveling Salesman Problem (MTSP)** is an extension of the TSP used when a fleet of vehicles have to be routed from a single depot. A set of routes are generated, one for each vehicle in the fleet. A node is assigned to only one vehicle; A vehicle will have more than one node assigned to it.

**Vehicle Routing Problem (VRP)** is a MTSP with capacity restriction of the multiple vehicles coupled with varying demands at each node. VRP was proposed by Dantzig and Ramser in 1959. It is an important problem in the fields of transportation, distribution and logistics. Several variations and specializations of the vehicle routing problem exist like VRP problems having time windows within which the deliveries must be made and VRP with limited carrying capacity of vehicles.
Chinese Postman Problem (CPP) is a special case where the demand for the service occurs on the arcs rather than at the nodes. Examples include street sweeping, snow removal, refuse collection, postal delivery and paper delivery.

1. **Solution Approach to Routing and Scheduling Problems**

Consider the delivery of milk cans to DCs as described in Fig. 1. Suppose there are 10 DCs, we can have $2^{10}$ or 1024 possible routings. Realistic problems may be of greater size making the solution to become expensive to solve optimally. Hence heuristic solution techniques have been developed to yield good solutions if not optimal solutions to these problems.

Two commonly used heuristics for the traveling salesman problem are the nearest neighbor procedure and the Clark and Wright savings heuristic.

2.1 **Nearest Neighbor Procedure (NNP)** builds a tour based on the cost or distance of traveling from the last-visited node to the closest node in the network. The steps in NNP are:

1) Start with a node at the beginning of the tour (say depot node)
2) Find the node closest to the last node and add to the tour. If the closest node is already in the tour or already there in the path then select next closest node.
3) Go to step 2 until all nodes have been added
4) Connect the first and the last node to form a complete tour

**Example**

Gather the data of distance or cost of traveling for milk cans delivery example from every node in the network to every other node in the network with undirected arcs and present it in a distance matrix as shown below. The distance from node, i, to node, j, will be the same as the distance from j to i, provided i ≠ j. Such a network is said to be symmetrical.
**Distance matrix**

<table>
<thead>
<tr>
<th>From Node</th>
<th>To Node (distances in miles)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>6</td>
<td>3</td>
<td>12</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>-</td>
<td>5</td>
<td>10</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>5</td>
<td>-</td>
<td>8</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>10</td>
<td>8</td>
<td>-</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>9</td>
<td>3</td>
<td>10</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

**Step 1:** Start with depot node (node 1). Examine the distances between node 1 and every other node. Closest node is node. So fix the partial tour or path as 1 → 3

**Step 2:** Find the closest node to the last node added (node 3) that is not currently in the path. This is node 2. Connect it to the path to yield 1 → 3 → 2

**Step 3:** The node closest to node 2 is node 5. Connect it to yield 1 → 3 → 2 → 5

**Step 4:** Connect the last node i.e. node 4 to the path and complete the tour by connecting node 4 to the depot. The complete tour formed is 1 → 3 → 2 → 5 → 4 → 1 as shown in Figure. The length of the tour is 33 miles
This final tour determined by nearest neighbor procedure may not be the best-possible route. As the alternative path, 1→2→5→4→3→1 has the total distance of 6+4+10+8+3 = 31 miles. This shows the limitation of the heuristics for not resulting in optimality. Enumeration is possible for this small network. But for large problems with 100 or 200 nodes, enumeration becomes very difficult.

**Clark and Wright Savings Heuristic (C-W)**

Clark and Wright (C-W) algorithm was developed by Enter G Clarke and J. W. Wright. The basis for C-W algorithm is savings concept where these savings are realized by linking pairs of delivery points served by a single depot in the network. First step in C & W heuristic is to select a node as depot node and label as node 1. To understand the savings concept, assume n-1 vehicles are available where n is the number of nodes. Each vehicle travels from the depot directly to the node and return to the depot. As we can see in the network below for milk delivery example, one vehicle goes from depot to node 2
and come back and other vehicle goes from depot to node 3 and comes back to depot (node 1).

The total distance traveled by two vehicles is 34 miles (2*10 + 2*7 =34). This is not a feasible solution if all nodes in TSP should be visited by one vehicle.

We know that the distance from node 2 to node 3 is 5 miles. If we select the tour starting from depot (node 1) to be 1\rightarrow 2\rightarrow 3\rightarrow 1 that is linking node 2 and 3 before returning to node 1 we can achieve savings of 12 mile.

These savings can be computed as follows. Let, $D_{ij}$, presented the distance between node i and node j. Suppose a vehicle travels from depot (node 1) to DC node 2 and comes back and again make tour to other DC node 3 and finally return to node 1. The total cost of such tour will be as given below.

$$\text{Total cost 1} = 2D_{12} + 2D_{13}$$

The cost for following 1\rightarrow 2\rightarrow 3\rightarrow 1 tour will be

$$\text{Total cost 2} = D_{12} + D_{23} + D_{31}$$
The difference between total cost 2 and total cost 1 will give us savings, $S_{23}$, by pairing up nodes 2 and 3 as given below.

\[ S_{23} = 2D_{12} + 2D_{13} - (D_{12} + D_{23} + D_{31}) \]

Or \[ S_{23} = D_{12} + D_{13} - D_{23} \]

Hence savings is computed as a measure of how much the tour length or cost can be reduced by ‘hooking up’ a pair of nodes (nodes 2 and 3).

Steps in the C & W savings heuristic for networks with n nodes is as follows.

**STEPS IN C-W ALGORITHM**

**Step 1**: Construct a shortest distance half matrix comprised of shortest distance or least time between each pair of nodes including starting node.

**Step 2**: Develop an initial allocation of one round-trip from starting node to each destination.

**Step 3**: Calculate the net savings for each pair of nodes (excluding starting node) and construct net savings half-matrix. Net savings are the savings achieved by pairing nodes relative to the cost of making round trip to each paired node from depot or node 1.

**Step 4**: Introduce a special indicator, I, into appropriate cells of the net savings half-matrix. This indicator will tell if the two nodes in question are directly linked. The link can be from either node 1 to any other node j or it can be between any pair of nodes, i and j when (i≠1 and i≠j). This trip indicator, I may have one of three values.

**Step 4.1**: when a vehicle travels from point of origin (node 1) to node j (other than node 1) and then returns to point of origin then, I=2. That is, a round trip will have I=2. In the matrix we can write the value of indicator that is $I_{ij} = 2$ (where j≠1). This value will appear only in the first row of the net-savings half matrix.
**Step 4.2:** When a vehicle travels one way directly between two nodes, the \( I_{ij} = 1 \).

**Step 4.3:** The value of trip indicator \( I_{ij} = 0 \), if a vehicle does not travel directly between two particular nodes that is no trip between a pair of nodes.

**Step 5:** Select the \((i,j)\) cell in the net savings half matrix having maximum net savings and link \( i \) and \( j \). But before linking the pair of nodes, following conditions should be fulfilled to link \( i \) and \( j \).

**Step 5.1:** \( I_{ii} \) and \( I_{ij} \) must be greater than 0.

**Step 5.2:** Nodes \( i \) and \( j \) are not already on the same route or loop.

**Step 5.3:** There is no violation of constraint in linking \( i \) and \( j \). (There can be some constraints like one way route is only permissible between two streets, there is not a proper road between two nodes, the vehicle has limited capacity etc.)

If the cell meets all the above conditions they assign \( I_{ij} = 1 \) otherwise assign zero and select the cell with next highest net savings and check for the conditions stated in 5.1, 5.2 and 5.3.

**Step 6:** When all nodes are linked on a single route and no other cell meets the conditions in step 5, stop the algorithm. Otherwise go to step 5.

**Example**

We will solve the milk delivery problem from depot to four distribution centers in figure below using C-W algorithm. The arcs between all the nodes present the distance.
**Step 1:** Construct shortest distance half matrix as given below.

```
<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>12</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>13</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>
```

**Step 2:** Develop an initial trip allocation
**Step 3:** Calculate net savings for each pair of nodes as given below.

<table>
<thead>
<tr>
<th>Paired Nodes</th>
<th>Savings $s_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,3</td>
<td>$S_{23} = 10 + 12 - 14 = 8$</td>
</tr>
<tr>
<td>2,4</td>
<td>$S_{24} = 10 + 9 - 13 = 6$</td>
</tr>
<tr>
<td>2,5</td>
<td>$S_{25} = 10 + 8 - 8 = 10$</td>
</tr>
<tr>
<td>3,4</td>
<td>$S_{34} = 12 + 9 - 6 = 15$</td>
</tr>
<tr>
<td>3,5</td>
<td>$S_{35} = 12 + 8 - 7 = 13$</td>
</tr>
<tr>
<td>4,5</td>
<td>$S_{45} = 9 + 8 - 9 = 8$</td>
</tr>
</tbody>
</table>

Construct net savings half matrix

Step 4: Introduce indicator $I$ for initial round trip allocation.
Step 5: Select the cell with maximum net savings

We can see that cell (3,4) has maximum savings so we will link (3,4) after checking conditions 5.1, 5.2 and 5.3 given in step 5 of C-W algorithm.

- $I_{13}$ and $I_{14}$ are greater than 0 and both have value equal to 2.
- Node 3 and node 4 are not already on the same node.
- No constraints are mentioned for this problem.

Hence, we will link nodes 3 and 4 which will change the indicator values in the net savings matrix as given below.
The value of indicator in the cell (3,4) will become 1. The values for $I_{13}$ and $I_{14}$ will also change from 2 to 1.

Now look for next highest savings in the matrix, which is cell (3, 5) that is with the value of 13. Check the conditions of step 5.

- $I_{13}$ and $I_{15}$ are greater than 0
- Node 3 and node 5 are not already on the route

So, we will link nodes 3 and 5 with following route and update in net savings matrix.
The indicator $I_{15}$ will take value of 1 and $I_{35}=1$ will be introduced in cell (3,5). Look now $I_{13}=0$, hence we cannot link node 3 with any other node except 4 and 5.
Look for next highest savings, which are in cell (2,5).

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[I_{12} = 2]</td>
<td>[I_{13} = 0]</td>
<td>[I_{14} = 1]</td>
<td>[I_{15} = 1]</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>6</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>[I_{34} = 1]</td>
<td>[I_{35} = 1]</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>15</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Check for the conditions in step 5.

- \[I_{12}\] and \[I_{15}\] are greater than 0

- Node 2 and node 5 are not on the route.

So, pair up node 2 and node 5 and update the network and the matrix as given below.
Selected path or route is

1 → 4 → 3 → 5 → 2 → 1

or

1 → 2 → 5 → 3 → 4 → 1
With distance of $10 + 8 + 7 + 6 + 9 = 40$Km

For the multiple vehicles routing problem, there can be some constraints regarding the limited capacity of vehicle. In such case introduce this constraint in the condition number 5.3 of step 5 in C-W algorithm. One should have data on the demand at each node. Then apply C-W algorithm and keep on linking nodes until the demand at linked nodes reach capacity of vehicle. Stop further linking of nodes and introduce second vehicle.

**Scheduling Service Vehicles**

Scheduling problems have delivery-time restrictions with specified starting and ending times for a service in advance. Subway schedules fall into this category. A service scheduling problem is called two-sided window if the time limits are specified such as a delivery has to be made between 11 am and 2 pm. A service scheduling problem is called one-sided window if a service specifies that it should precede a given time, for example the case of newspaper, delivery should complete before 7 am.

These problems consists of a (i) set of tasks, each with starting time and ending times (ii) set of directed arcs with starting and ending locations. The set of vehicles may be housed at one or more depots.

Consider the network shown below with a depot serving to five locations (nodes) with specified starting time, $S$ and Ending time, $E$ in AM.
An arc may join node i to node j if the start time of task j is greater than the end time of task i.

**Deadhead time**

Deadhead time is a user-specified period of time such that start time of task j must be longer than the end time of task i. It is the non-productive time required for the vehicle to travel from one task location to another or return to the depot empty. The deadhead time for the above example is 45 minutes.

**The concurrent Scheduler Approach**

This heuristic is used to solve the above type of scheduling problem. The procedure is as follows:

(i) Order all tasks by starting times
(ii) Assign first task to vehicle 1
(iii) For the remaining number of tasks, assign the next task to vehicle that has the minimum deadhead time to that task. Otherwise create a new vehicle and assign the task to it.

By doing so, the schedule obtained for the example given above is as follows:

<table>
<thead>
<tr>
<th>Vehicle</th>
<th>Task</th>
<th>Start time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle 1</td>
<td>1</td>
<td>8:00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>9:30</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>10:45</td>
</tr>
<tr>
<td>Vehicle 2</td>
<td>2</td>
<td>8:30</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>10:15</td>
</tr>
</tbody>
</table>
Questions

1. Apply C & W savings heuristic to the network given below. The distance between nodes 4 and 2 is 4 miles; nodes 2 and 3 is 5 miles; nodes 4 and 3 is 7 miles.

2. For the network given below find tours that will minimize the total distance traveled by the two vehicles starting and ending at the depot (node 1). The demand at each node is given in parentheses and arcs represent the distance (kms) between nodes. Capacity of vehicle 1 is 40 tons and for vehicle 2 capacity limit is 50 tons.
3. Solve the problem given below using concurrent scheduler approach. Deadhead time is 20 minutes. Find the minimum number of vehicles required to meet the service requirement.

<table>
<thead>
<tr>
<th>Task</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9:00 AM</td>
<td>9:45 AM</td>
</tr>
<tr>
<td>2</td>
<td>9:15 AM</td>
<td>10:00 AM</td>
</tr>
<tr>
<td>3</td>
<td>10:10 AM</td>
<td>11:15 AM</td>
</tr>
<tr>
<td>4</td>
<td>10:20 AM</td>
<td>11:45 AM</td>
</tr>
<tr>
<td>5</td>
<td>10:45 AM</td>
<td>12:00 NOON</td>
</tr>
<tr>
<td>6</td>
<td>1:00 PM</td>
<td>1:45 PM</td>
</tr>
<tr>
<td>7</td>
<td>1:20 PM</td>
<td>2:30 PM</td>
</tr>
<tr>
<td>8</td>
<td>1:45 PM</td>
<td>2:45 PM</td>
</tr>
<tr>
<td>9</td>
<td>2:15 PM</td>
<td>3:00 PM</td>
</tr>
<tr>
<td>10</td>
<td>3:30 PM</td>
<td>4:15 PM</td>
</tr>
</tbody>
</table>