Special Topic: Vehicle Routing Problem

Chapter 1

The Traveling Salesman Problem

1.1 Formulations

In this section, we are going to compare between different formulations of TSP. Generally speaking, let $G = (V, A)$ be a graph where V is a set of n vertices, and A is a set of arcs (or edges). Let $C = c_{ij}$ be a cost (distance) matrix associated with A. The TSP consists of determining a minimum cost (distance) Hamiltonian circle (or cycle) that visits each vertex once and only once. If for all $i, j \in V, c_{ij} = c_{ji}$, then the TSP is symmetrical, otherwise is asymmetrical.
Define the decision variable x_{ij} as the following

$$x_{ij} = \begin{cases} 1, & \text{if goes from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases}, \quad (i,j) \in A \tag{1.1}$$

The objective function will be

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{1.2}$$

1.1.1 Dantzig-Fulkerson-Johnson (DFJ) Formulation

The first famous formulations for TSP is the **Dantzig-Fulkerson-Johnson (DFJ) formulation**:

$$\sum_{j\in V,(i,j)\in A} x_{ij} = 1, \quad \forall i \in V \tag{1.3}$$

$$\sum_{i\in V,(i,j)\in A} x_{ij} = 1, \quad \forall j \in V \tag{1.4}$$

$$\sum_{j\notin S, i\in S,(i,j)\in A} x_{ij} \geq 1, \quad \forall S \subset V, 2 \leq |S| \leq n-1 \tag{1.5}$$

In the formulation, constraints (1.3) and constraints (1.4) are degree constraints, which specify that every vertex is entered exactly once. Constraints (1.5) is the sub-tour constraints, they prohibit the formation of sub-tours. S is a non-empty subset of V, and has at least 2 vertices. 1.5 can be replaced by

$$\sum_{i,j\in S,(i,j)\in A} x_{ij} \leq |S| - 1, \quad \forall S \subset V, 2 \leq |S| \leq n-1 \tag{1.6}$$

If we list all sub-tour constraints in DFJ, there will be $O(2^n)$ constraints and $O(n^2)$ binary variables. The exponential number of constraints makes it impractical to solve directly. Instead, lazy constraints are usually implemented for the sub-tour elimination constraints (1.5 or 1.6).

1.1.2 Miller-Tucker-Zemlin (MTZ) Formulation

We can also formulate TSP using sequential formulations, namely, **Miller-Tucker-Zemlin (MTZ) formulation**. In the MTZ formulation, the degree constraints (1.3 and 1.4) are the same as in DFJ formulation.

Define a new set of integer decision variables u_i, u_i defined as the sequence in which node i is visited, $u_1 = 1$. The sub-tour constraints (1.5 or 1.6) are replaced by the following:

$$u_i - u_j + (n-1)x_{ij} \leq n-2, \quad i,j = 2,\cdots,n \in V, (i,j) \in A \tag{1.7}$$
$$1 \leq u_i \leq n-1, \quad i \in 2,\cdots,n \in V \tag{1.8}$$

In MTZ formulation, there are $O(n^2)$ constraints, $O(n^2)$ binary variables, and $O(n)$ continuous variables.

1.1.3 Quadratic Formulation (QAP)

In this section, we are going to go over a TSP formulation are super bad. However, it still has some value for further study.
The idea is to transform TSP into an assignment problem. Assuming we have n boxes, which represents n steps in the path. Define x_{ij} as

$$x_{ij} = \begin{cases} 1, & \text{Vertex } i \text{ is assigned to box } j \\ 0, & \text{Otherwise} \end{cases} \tag{1.9}$$

The constraints are simple as an assignment problem as following

$$\sum_{j=1}^{n} x_{ij} = 1, \quad \forall i \in V \tag{1.10}$$
$$\sum_{i \in V}^{n} x_{ij} = 1, \quad j = 1,\ldots,n \tag{1.11}$$

However, the tricky part is in the objective function

$$\min \quad \sum_{i \in V} \sum_{j \in V \setminus \{i\}} \sum_{k=1}^{n-1} c_{ij} x_{ik} x_{j,k+1} + \sum_{i \in V} \sum_{j \in V \setminus \{i\}} c_{ij} x_{in} x_{j1} \tag{1.12}$$

Notice that the objective function is not linear function, with the multiplications of decision variables. Now we are going to linearize them. The linearized version is as following

$$\min \quad \sum_{i \in V} \sum_{j \in V \setminus \{i\}} \sum_{k=1}^{n-1} c_{ij} w_{ij}^k + \sum_{i \in V} \sum_{j \in V \setminus \{i\}} c_{ij} w_{ij}^n \tag{1.13}$$
$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} = 1, \quad \forall i \in V \tag{1.14}$$
$$\sum_{i \in V}^{n} x_{ij} = 1, \quad j = 1,\ldots,n \tag{1.15}$$
$$w_{ij}^k \geq x_{ik} + x_{j,k+1} - 1, \quad i \in V, j \in V \setminus \{i\}, k = 1,\cdots,n-1 \tag{1.16}$$
$$w_{ij}^k \geq x_{ik} + x_{j1} - 1, \quad i \in V, j \in V \setminus \{i\}, k = n \tag{1.17}$$
$$w_{ij}^k \in \{0,1\}, \quad i \in V, j \in V \setminus \{i\}, k = 1,\ldots,n \tag{1.18}$$
$$x_{ij} \in \{0,1\}, \quad i \in V, j \in V \setminus \{i\} \tag{1.19}$$

We can prove that this is very very bad.

1.1.4 Flow Based Formulations

In this section, flow based formulations are discussed, which includes **Single Commodity Flow**, **Two Commodity Flow** and **Multi-Commodity Flow**. In these formulations, continuous variables are introduced to represent the flow on the arcs.
In Single Commodity Flow formulation, define y_{ij} as the flow in an arc $(i,j) \in A$.

Degree constraints 1.3 and 1.4 are retained. The following constraints are introduced:

$$y_{ij} \leq (n-1)x_{ij}, \quad \forall i,j \in V, (i,j) \in A \tag{1.20}$$

$$\sum_{j \in V, (1,j) \in A} y_{1j} = n-1 \tag{1.21}$$

$$\sum_{i \in V, (i,j) \in A} y_{ij} - \sum_{k \in V, (j,k) \in A} y_{jk} = 1, \quad \forall j \in V \setminus \{1\} \tag{1.22}$$

$$\tag{1.23}$$

Constraints (1.20) can be tighten by the following:

$$y_{ij} \leq (n-1)x_{ij}, \quad i = 1, j \in V \setminus \{1\}, (i,j) \in A \tag{1.24}$$

$$y_{ij} \leq (n-2)x_{ij}, \quad i,j \in V \setminus \{1\}, (i,j) \in A \tag{1.25}$$

$$\tag{1.26}$$

In SCM formulation, there are $O(n^2)$ constraints, $O(n^2)$ binary variables and $O(n^2)$ continuous variables.
In Two Commodity Flow formulation, define y_{ij} as the flow in an arc $(i,j) \in A$, for commodity type 1, and define z_{ij} as the flow in an arc $(i,j) \in A$, for commodity type 2.
Besides degree constraints, the other constraints are as following

$$y_{ij} + z_{ij} = (n-1)x_{ij}, \quad \forall i,j \in V, (i,j) \in A \tag{1.27}$$

$$\sum_{j \in V \setminus \{1\}} (y_{1j} - y_{j1}) = n-1, \quad (1,j) \in A \tag{1.28}$$

$$\sum_{j \in V} (y_{ij} - y_{ji}) = 1, \quad \forall i \in V \setminus \{1\}, (i,j) \in A \tag{1.29}$$

$$\sum_{j \in V \setminus \{1\}} (z_{1j} - z_{j1}) = 1-n, \quad (1,j) \in A \tag{1.30}$$

$$\sum_{j \in V} (z_{ij} - z_{ji}) = -1, \quad \forall i \in V \setminus \{1\}, (i,j) \in A \tag{1.31}$$

$$\sum_{j \in V} (y_{ij} + z_{ij}) = n-1, \quad \forall i \in V \tag{1.32}$$

$$\tag{1.33}$$

In TCM formulation, constraints (1.27) only allow flow in an arc if present. Constraints (1.28) and (1.29) forces $(n-1)$ units of commodity type 1 to flow in at node 1 and 1 unit to flow out at every other nodes. Constraints (1.30) and (1.31) are similar, those forces $(n-1)$ units of commodity type 2 to flow out at node 1 and 1 unit to flow in at every other nodes. Constraints (1.32) forces exactly $(n-1)$ units of combined commodity in each arc.
In TCM formulation, there are $O(n^2)$ constraints, $O(n^2)$ binary variables and $O(n^2)$ continuous variables.
The SCM and the TCM can be generalized into **Multi-Commodity Flow formulation**. As usual, degree constraints are retained. The following continuous variables are introduced. Define y_{ij}^k as the flow of commodity type k in arc $(i,j) \in A$.
The other constraints are

$$y_{ij}^k \leq x_{ij}, \quad \forall i,j,k \in N, k \neq 1 \tag{1.34}$$

$$\sum_{i \in V} y_{1i}^k = 1, \quad \forall k \in V \setminus \{1\} \tag{1.35}$$

$$\sum_{i \in V} y_{i1}^k = 0, \quad \forall k \in V \setminus \{1\} \tag{1.36}$$

$$\sum_{i \in V} y_{ik}^k = 1, \quad \forall k \in V \setminus \{1\} \tag{1.37}$$

$$\sum_{j \in V} y_{kj}^k = 0, \quad \forall k \in V \setminus \{1\} \tag{1.38}$$

$$\sum_{i \in V} y_{ij}^k - \sum_{i \in V} y_{ji}^k = 0, \quad \forall j,k \in V \setminus \{1\}, j \neq k \tag{1.39}$$

Constraints (1.34) only allow flow in an arc which is present. Constraints (1.35) forces exactly one unit of each type of commodity to flow in at node 1. Constraints (1.36) prevent any commodity flow out at node 1.Constraints (1.37), and Constraints (1.38), forces exactly one unit of type k commodity to flow out, and in, at every node except node 1. Constraints (1.39) forces balance of all types of commodities at every node except node 1.

This formulation has $O(n^3)$ constraints, $O(n^2)$ binary variables, and $O(n^3)$ continuous variables.

1.1.5 Shortest Path Formulation

A graph for timed staged shortest path

In this section, we are going to introduce another form of formulation with different definition of decision variable and objective function.

Assuming for a completed graph $G = (V, A)$. Define x_{ij}^t as the following

$$x_{ij}^t = \begin{cases} 1, & \text{If path crosses arc } (i,t) \text{ and } (j,t+1) \\ 0, & \text{Otherwise} \end{cases} \quad , \quad i \in V, j \in V \setminus \{i\}, t = 1, \cdots, n \tag{1.40}$$

The objective function will be

$$\min \quad \sum_{i \in V} \sum_{j \in V \setminus \{i\}} c_{ij} \sum_{t=1}^{n} x_{ij}^t \tag{1.41}$$

The constraints are as following

$$\sum_{j \in V \setminus \{1\}} x_{1j}^1 = 1 \tag{1.42}$$

$$\sum_{j \in V \setminus \{1,i\}} x_{ij}^2 - x_{1i}^1 = 0, \quad \forall i \in V \setminus \{1\} \tag{1.43}$$

$$\sum_{j \in V \setminus \{1,i\}} x_{ij}^t - \sum_{j \in V \setminus \{1,i\}} x_{ji}^{t-1} = 0, \quad \forall i \in V \setminus \{1\}, t \in \{2, \ldots, n-1\} \tag{1.44}$$

$$x_{i1}^n - \sum_{j \in V \setminus \{1,i\}} x_{ji}^{n-1} = 0, \quad \forall i \in V \setminus \{1\} \tag{1.45}$$

$$\sum_{i \in V \setminus \{1\}} x_{i1}^n = 1 \tag{1.46}$$

$$\sum_{t=2}^{n-1} \sum_{j \in V \setminus \{1,i\}} x_{ij}^t + x_{i1}^n \leq 1, \quad \forall i \in V \setminus \{1\} \tag{1.47}$$

Notice that constraint (1.47) can be replaced by

$$x_{1i}^1 + \sum_{t=2}^{n-1} \sum_{j \in V \setminus \{1,i\}} x_{ji}^t \leq 1, \quad \forall i \in V \setminus \{1\} \tag{1.48}$$

1.2 NP Completeness of TSP

1.2.1 Proof of TSP $\in$ NPC

1.2.2 Polynomially Solvable Special Cases of TSP

1.3 Lower Bounds of TSP

1.3.1 The Assignment Lower Bound

1.3.2 The Minimum Spanning Tree (Arborescence) Bound

1.3.3 The 2-match Problem

1.3.4 Held & Karp Bound (Lagrangian Relaxation)

In this section, we will solve the Dantzig-Fulkerson-Johnson formulation using Lagrangian Relaxation. The bound found by this method is also known as Held & Karp Bound. In the Held & Karp relaxation, the degree constraints are relaxed, as a result, we require our solution to be connected and to contain n edges, but it might not have exactly two edges incident to each vertex.
The feasible solution to the Lagrangian relaxation is called **1-tree**, which

- Have a spanning tree on nodes $\{2, 3, \cdots, n\}$
- Two edges incident to node 1

1.4 Constructive Heuristic

1.4.1 Nearest Neighborhood Algorithm

1.4.2 Insertion Algorithm

1.4.3 Sweep Algorithm

1.4.4 Christofides Algorithm

1.5 Local Search Heuristic

1.5.1 Lin-Kernighan Algorithm

The general idea of Lin-Kernighan Algorithm is based on a substantial generalization of the interchange transformation.

1.6 Metaheuristic

1.6.1 Simulated Annealing

1.6.2 Genetic Algorithm

Chapter 2

The Vehicle Routing Problem

Chapter 3

The Capacitate Vehicle Routing Problem

Chapter 4

The Vehicle Routing Problem with Time Windows

Chapter 5

Pickup-and-Delivery Problem

5.1 Problem Formulation

Let N be the set of transportation requests. For each transportation request $i \in N$, a load of size $\bar{q}_i \in \mathbb{N}$ has to be transported from a set of origins N_i^+ to a set of destinations N_i^-. Each load is subdivided as follows

$$\bar{q}_i = \sum_{j \in N_i^+} q_j = - \sum_{j \in N_i^-} q_j \tag{5.1}$$

Define $N^+ = \cup_{i \in N} N_i^+$ as the set of all origins and $N^- = \cup_{i \in N} N_i^-$ as the set of all destinations. Let $V = N^+ \cup N^-$. Furthermore, the M be the set of vehicles. Each vehicle $k \in M$ has a capacity $Q_k \in \mathbb{N}$, a start location k^+, and an end location k^-. Define $M^+ = \{k^+ | k \in M\}$ as the set of start locations and $M^- = \{k^- | k \in M\}$ as the set of end locations. Let $W = M^+ \cup M^-$.
For all $i, j \in V \cup W$ let d_{ij} denote the travel distance, t_{ij} denote the travel time, and c_{ij} denote the travel cost. Note that the dwell time at origins and destinations can be easily incorporated in the travel time and therefore will not be considered explicitly.

Definition 5.1.1. A pickup and delivery route R_k for vehicle k is a directed route through a subset $V_k \in V$ such that:

- R_k starts in k^+
- $\forall i \in N, (N_i^+ \cup N_i^-) \cap V_k = \emptyset$ or $N_i^+ \cup N_i^-$
- If $N_i^+ \cup N_i^- \subseteq V_k$, then all locations in N_i^+ are visited before locations in N_i^-
- Vehicle k visits each location in V_k exactly once
- The vehicle load never exceeds Q_k
- R_k ends in k^-

Definition 5.1.2. A pickup and delivery plan is a set of routes $\mathcal{R} = \{R_k | k \in M\}$ such that:

- R_k is the pickup and delivery route for vehicle k, for each $k \in M$
- $\{V_k | k \in M\}$ is a partition of V,

Here are the special cases of the General Pickup and Delivery Problem

Example. The pickup and delivery problem, where $|W| = 1$, $|N_i^+| = |N_i^-| = 1, \forall i \in N$. In this case we define i^+ as the unique element of $|N_i^+|$ and i^- as the unique element of $|N_i^-|$.

Example. The dial-a-ride problem, where $|W| = 1$ and $|N_i^+| = |N_i^-| = 1, \bar{q}_i = 1, \forall i \in N$

Example. The vehicle routing problem, where $|W| = 1$, $|N_i^+| = |N_i^-| = 1 \forall i \in N$ and $N^+ = W$ or $N^- = W$.

Notice: Generally speaking, we usually have problem with $|N_i^+| = |N_i^-| = 1$. In the cases where $|N_i^+| > 1$ or $|N_i^-| > 1$, the transportation requests can be decomposed into several independent requests with $|N_i^+| = |N_i^-| = 1$, unless it has to be served by the same vehicle.

Notice: We are not aware of any real-life applications where both $|N_i^+| > 1$ and $|N_i^-| > 1$ at the same time yet.

The following is the formulation for General Pickup and Delivery Problem.

Table 5.1: Decision variable notation

z_i^k	For $i \in N, k \in M$ Equals to 1 if transportation request i is assigned to vehicle k and 0 otherwise.
x_{ij}^k	For $(i,j) \in (V \times V) \cup \{(k^+, j) \mid j \in V\} \cup \{(j, k^-) \mid j \in V\}, k \in$ Equals to 1 if vehicle k travels from location i to location j and 0 otherwise.
D_i	For $i \in V \cup W$, specifying the departure time at vertex i.
y_i	For $i \in V \cup W$, specifying the load of vehicle arriving at vertex i. Define $q_{k^+} = 0, \forall k \in M$.

$$
\begin{aligned}
\min\quad & f(x) && && (5.2)\\
\text{s.t.}\quad & \sum_{k \in M} z_i^k = 1 && \forall i \in N && (5.3)\\
& \sum_{j \in V \cup W} x_{lj}^k = z_i^k && \forall i \in N, l \in N_i^+ \cup N_i^-, k \in M && (5.4)\\
& \sum_{j \in V \cup W} x_{jl}^k = z_i^k && \forall i \in N, l \in N_i^+ \cup N_i^-, k \in M && (5.5)\\
& \sum_{j \in V \cup \{k^-\}} x_{k^+ j}^k = 1 && \forall k \in M && (5.6)\\
& \sum_{j \in V \cup \{k^+\}} x_{ik^-}^k = 1 && \forall k \in M && (5.7)\\
& D_{k^+} = 0 && \forall k \in M && (5.8)\\
& D_p \leq D_q && \forall i \in N, p \in N_i^+, q \in N_i^- && (5.9)\\
& D_i + t_{ij} \leq D_j + M(1 - x_{ij}^k) && \forall i, j \in V \cup W, k \in M && (5.10)\\
& y_{k^+} = 0 && \forall k \in M && (5.11)\\
& y_l \leq \sum_{k \in M} Q_k z_i^k && \forall i \in N, l \in N_i^+ \cup N_i^- && (5.12)\\
& y_j \geq y_l + q_i x_{lj}^k - M(1 - x_{lj}^x) && \forall i \in N, l \in N_i^+, j \in V \cup \{k^+\}, l \neq j && (5.13)\\
& y_j \leq y_l + q_i x_{lj}^k + M(1 - x_{lj}^x) && \forall i \in N, l \in N_i^+, j \in V \cup \{k^+\}, l \neq j && (5.14)\\
& y_j \geq y_l - q_i x_{lj}^k - M(1 - x_{lj}^x) && \forall i \in N, l \in N_i^-, j \in V \cup \{k^-\}, l \neq j && (5.15)\\
& y_j \leq y_l - q_i x_{lj}^k + M(1 - x_{lj}^x) && \forall i \in N, l \in N_i^-, j \in V \cup \{k^-\}, l \neq j && (5.16)\\
& x_{ij}^k \in \{0, 1\} && \forall i, j \in V \cup W, k \in M && (5.17)\\
& z_i^k \in \{0, 1\} && \forall i \in N, k \in M && (5.18)\\
& D_i \geq 0 && \forall i \in V \cup W && (5.19)\\
& y_i \geq 0 && \forall i \in V \cup W && (5.20)
\end{aligned}
$$

Explanation of constraints:

- Constraint (5.3) - Each transportation request is assigned to exactly one vehicle.

- Constraints (5.4), (5.5) - A vehicle only enters or leaves a location l if it is an origin or a destination of a transportation request assigned to that vehicle.
- Constraints (5.6), (5.7) - Make sure the vehicle is leaving from /arriving at the correct place.
- Constraint (5.8) - Initial starting time for all vehicles is 0.
- Constraint (5.9) - For each item to be delivered, it should be picked up before delivery
- Constraint (5.10) - Traveling distance from location i to j
- Constraint (5.11) - All vehicles leaving the initial location with no loading.
- Constraint (5.12) - Load capacity limit for vehicles.
- Constraints (5.13), (5.14), (5.15), (5.16) - Load / unload item when arriving pickup / delivery location
- Constraints (5.17), (5.18), (5.19), (5.20) - Binary variable definition and non-negativity constraints.

5.2 Heuristic Method

Chapter 6

Stochastic Vehicle Routing Problem

Chapter 7

Dynamic Vehicle Routing Problem