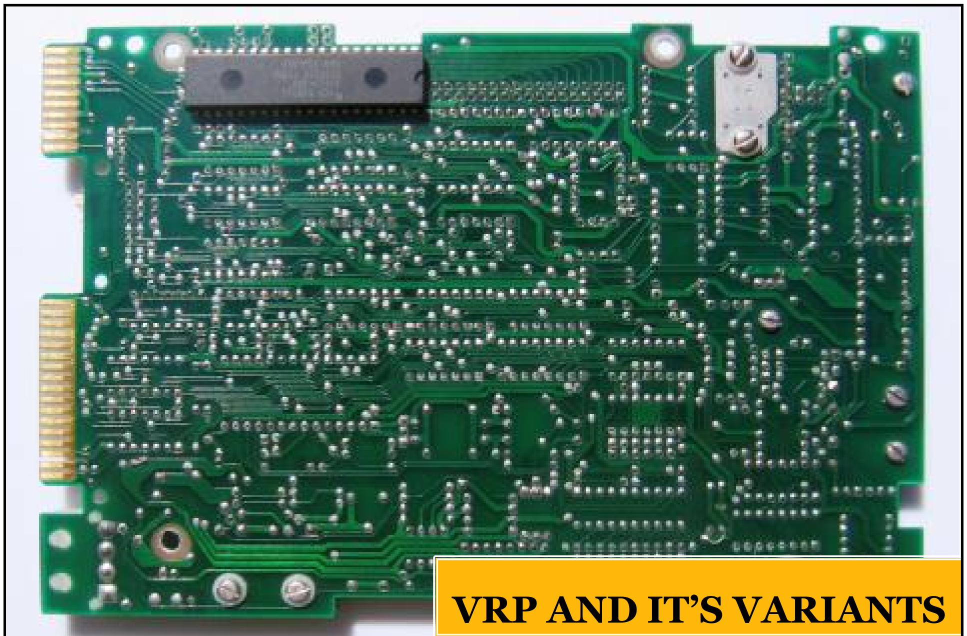


# VRP AND IT'S VARIANTS

Course: Transportation Network Design

## Vehicle Routing Problem

- **Setup and Model.**
- **Problem Variety.**
  - **Pure Pickup or Delivery Problems.**
  - **Mixed Pickups and Deliveries.**
  - **Pickup-Delivery Problems.**
  - **Backhauls.**
  - **Complications.**
- **Simplest Model: TSP**
  - **Heuristics.**
  - **Optimal methods.**

## Vehicle Routing Problem

- **Find best vehicle route(s) to serve a set of orders from customers.**
- **Best (Optimal) route may be**
  - **minimum cost,**
  - **minimum distance, or**
  - **minimum travel time,**
  - **Maximum coverage**
  - **...**

Objective Functions!

- **Orders may be**
  - **Delivery from depot to customers.**
  - **Pickup at customer and return to depot.**
  - **Pickup at one place and deliver to another place.**
  - **...**

Constraints

## VRP applications

- Goods distribution systems
- Solid waste collection
- Street cleaning
- School bus routing
- Dial-a-ride systems
- Transportation of handicapped persons
- routing of salespeople and of maintenance unites.
- ...

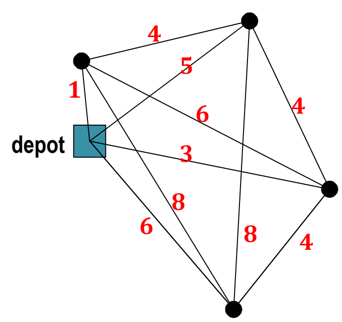
## General Setup

- Assign customer orders to vehicle routes (designing routes).

- Assign vehicles to routes.
  - Assigned vehicle must be compatible with customers and orders on a route.
- Assign drivers to vehicles.
  - Assigned driver must be compatible with vehicle.
- Assign tractors to trailers.
  - Tractors must be compatible with trailers.

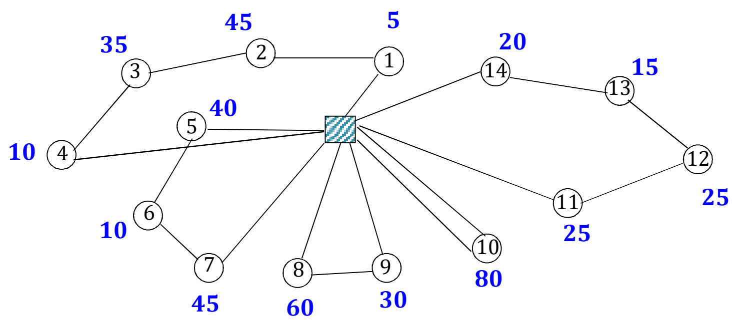
## Model (Graphical!)

- Nodes: physical locations
  - Depot.
  - Customers.
- Arcs or Links
  - Transportation links.
- Number on each arc represents cost, distance, or travel time.




Why the graph is complete graph?!!!
How can we build a complete graph from a non-complete graph in VRP applications?

# Vehicle Routing Problem (base case)



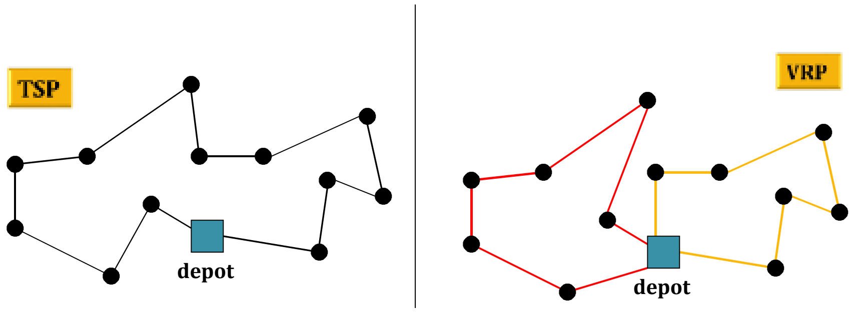

- Each customer must be visited by a vehicle
- The demand assigned to vehicle *k* must not exceed its capacity
- The tour for vehicle *k* begins at the depot, visits all its customers and returns to the depot.

## Simplest Case: TSP

- **Given a depot and a set of *n* customers, find a route (or "tour") starting and ending at the depot, that visits each customer once and is of minimum length.**
- **One vehicle.**
- **No capacities.**
- **Minimize distance.**
- **No time windows.**
- **No compatibility constraints.**

## TSP vs. VRP

- TSP: Travelling Salesman Problem (The simplest relaxed case of VRP!)
  - One route can serve all orders.
- VRP: Vehicle Routing Problem
  - May be more than one route (vehicle) is required to serve all orders.




## Symmetric and Asymmetric

Let $c_{ij}$ be the cost (distance or time) to travel from $i$ to $j$.

If $c_{ij} = c_{ji}$ for all customers, then the problem is *symmetric* (STSP or simply TSP).

- Direction does not affect cost.

If $c_{ij} \neq c_{ji}$ for some pair of customers, then the problem is *asymmetric* (ATSP).

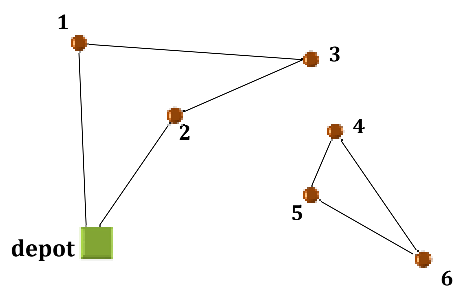
- Direction does affect cost.

## TSP - Integer Programming

- **Variables:** $y_{ij} = 1$ **if arc** $i, j$ **is on the route;**
  **= 0 otherwise.**
- **Objective: Minimize cost of a route**
  **Minimize** $\sum C_{ij} y_{ij}$
- **Constraints**
  - **Every node (customer) has one arc out.**
  - **Every node (customer) has one arc in.**
  - **No subtours.**

## TSP - Integer Programming

- **No subtour constraints prevent this:**



## TSP: Integer Programming Formulation

$$\min \sum_{i} \sum_{j} c_{ij} y_{ij}$$

$$\text{s.t} \quad \sum_{i<k} y_{ik} + \sum_{j>k} y_{kj} = 2, \quad k \in V$$

$$\sum_{i} \sum_{j} y_{ij} \leq |S| - 1 \quad S \subset V, 3 \leq |S| \leq n-3$$

$$y_{ij} \in \{0,1\} \quad \forall i,j \in E$$

## ATSP: Integer Programming Formulation

$$\min \sum_{i} \sum_{j} c_{ij} y_{ij}$$

$$\text{s.t} \quad \sum_{j} y_{ij} = 1, \quad i = 0, 1, \ldots, n-1$$

$$\sum_{i} y_{ij} = 1, \quad j = 0, 1, \ldots, n-1$$

$$\sum_{i} \sum_{j} y_{ij} \leq |S| - 1 \quad S \subset V, 2 \leq |S| \leq n-2$$

$$y_{ij} \in \{0,1\} \quad \forall i,j \in E$$

## Assignments

(1)

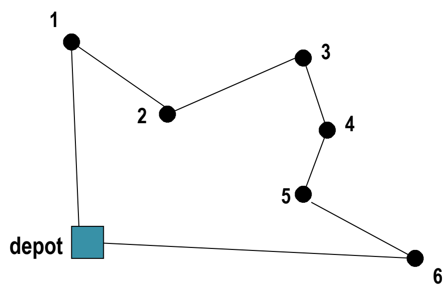
- Search for Miller–Tucker–Zemlin (MTZ) formulation for TSP
- Write both models in GAMS
- Solve two instances (small and large)
- Compare its efficiency with the previous formulation

(2)

- Are there any other formulations for TSP? Search for them and write about them? Prepare the GAMS model of one of these models and compare it with two previous formulation by solving two instances.
  - Deadline: next week. Tuesday,, 19 April

## TSP - Optimal Solutions

- **Route is as short as possible.**
- **Every customer (node) is visited once, including the depot.**
  - **Each node has one arc in and one arc out.**



## TSP Solution Methods

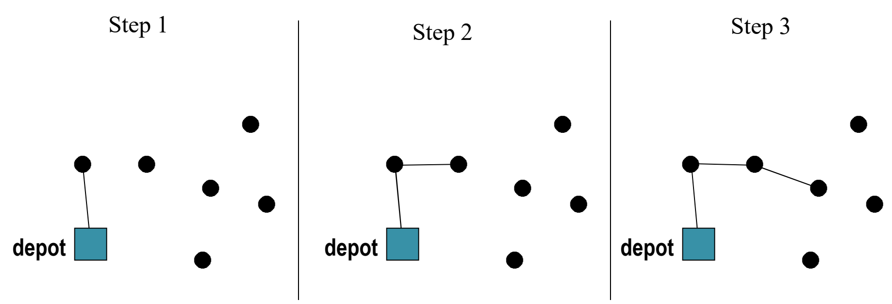
- Heuristics
  - **Construction: build a feasible route.**
  - **Improvement: improve a feasible route.**
    - Not necessarily optimal, but fast.
    - Performance depends on problem.
    - Worst case performance may be very poor.
- Exact algorithms
  - **Integer programming.**
  - **Branch and bound.**
    - Optimal, but usually slow.
    - Difficult to include complications.

## TSP Construction Heuristics

- Nearest neighbor.
  - **Add nearest customer to end of the route.**
- Nearest insertion.
  - **Go to nearest customer and return.**
  - **Insert customer closest to the route in the best sequence.**
- Savings method.
  - **Add customer that saves the most to the route.**
- Other Methods!

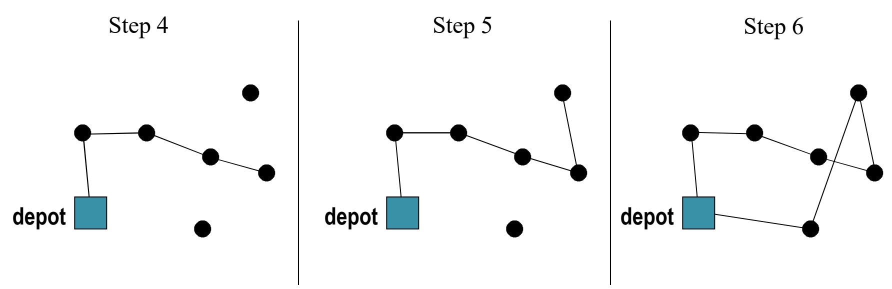
## Nearest Neighbor

Add nearest customer to end of the route.



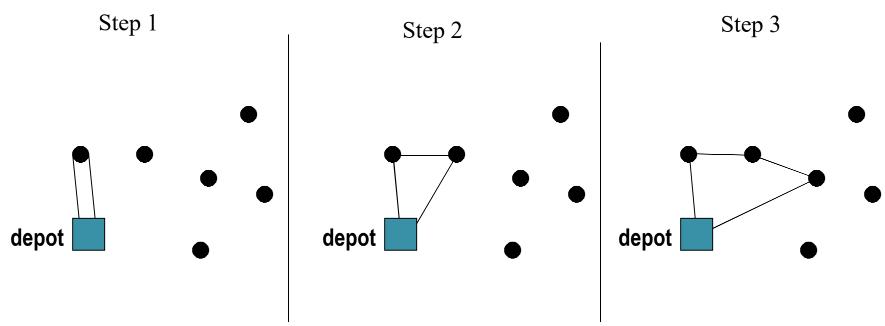

## Nearest Neighbor (Cont.)

Add nearest customer to end of the route.



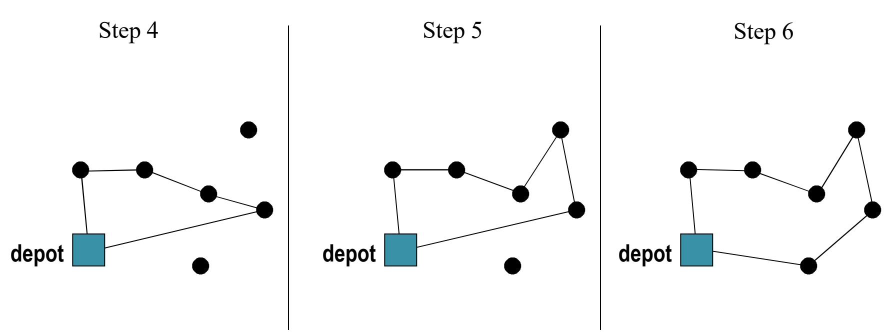
## Nearest Insertion

Insert customer closest to the route in the best sequence.




## Nearest Insertion

Insert customer closest to the route in the best sequence.



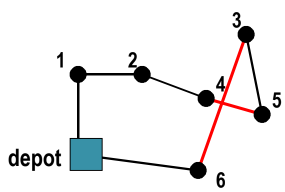
## Route Improvement Heuristics

- **Start with a feasible route.**
- **Make changes to improve route.**
  - **Exchange heuristics.**
    - Switch position of one customer in the route.
    - Switch 2 arcs in a route.
    - Switch 3 arcs in a route.
  - **Local search methods.**
    - Simulated Annealing.
    - Tabu Search.
    - Genetic Algorithms.

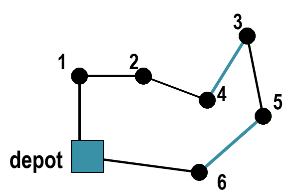
## K-opt Exchange

- **Replace k arcs in a given TSP tour by k new arcs, so the result is still a TSP tour.**
- **2-opt: Replace 4-5 and 3-6 by 4-3 and 5-6.**

Original TSP tour



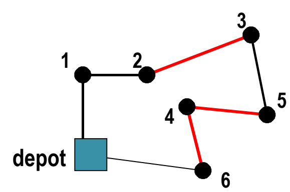

Improved TSP tour



## 3-opt Exchange

- **3-opt: Replace 2-3, 5-4 and 4-6 by 2-4, 4-3 and 5-6.**

Original TSP tour




Improved TSP tour