

Chapter 5

Vehicle Routing & Scheduling

**Zemenawit M. (MSc.)
2012 E.C**

Transportation systems

The primary components of any transportation system are facilities, equipment and people.

- **Facilities** include terminals, tracks, bridges, tunnels, signals, roadways, waterways and docks.
- **Equipment** includes containers, cars, tractors, trailers, locomotives, aircraft and vessels.
- **People** resources include loading and unloading crew, maintenance workforce, operating crew and other administrative and support staff.



www.Lachen-ist-gesund.de

Long-haul versus short-haul transportation

- In *long-haul freight transportation*, goods are moved over relatively long distances, between terminals or other facilities. Commodities may be transported by truck, rail, ship or any combination of modes.
- In *short-haul freight transportation*, goods are transported, usually by truck, between pick-up and delivery points situated in the same area. Such tasks are of short duration, shorter than a work shift

Transportation modes

Mode	fixed/variable costs	Applications
Road	Low/medium	Light industries, retail distribution
Railway	High/low	Extractors, Agricultural products
Water	Medium/low	Minerals, chemicals, some agricultural commodities
Air	Medium/high	Emergencies, perishable products

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 warehouse or depot to customers who have placed orders for such goods. Objective of such problems is to minimize the time and distance traveled.

Vehicle Routing & Scheduling: Part 1

- 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

- Find best vehicle route(s) to serve a set of orders from customers.
- Best route may be
 - **minimum cost,**
 - **minimum distance,** or
 - **minimum travel time.**
- Orders may be
 - Delivery from depot to customer.
 - Pickup at customer and return to depot.
 - Pickup at one place and deliver to another place.

APPLICATIONS OF VRP

The VRP is applicable in many practical situations directly related to

➤ the **physical delivery of goods** such as

- ◇ distribution of petroleum products,
- ◇ distribution of industrial gases,
- ◇ newspaper deliveries,
- ◇ delivery of goods to retail store,
- ◇ garbage collection and disposal,
- ◇ package pick-up and delivery,
- ◇ milk pick-up and delivery, etc.

➤ the **non-movement of goods** such as

- ◇ picking up of students by school buses,
- ◇ routing of salesmen,
- ◇ reading of electric meters,
- ◇ preventive maintenance inspection tours,
- ◇ employee pick-up and drop-off , etc.

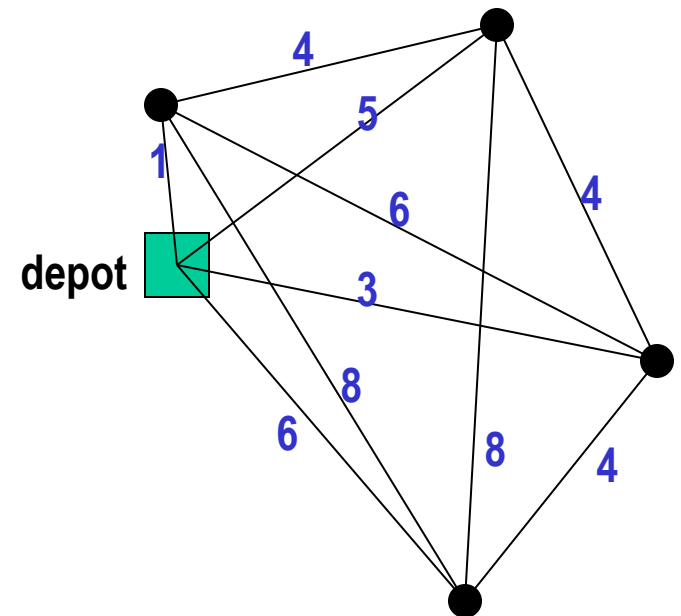


General Setup

- 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.

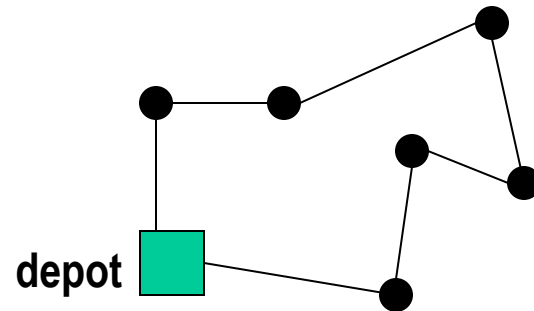
Model

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

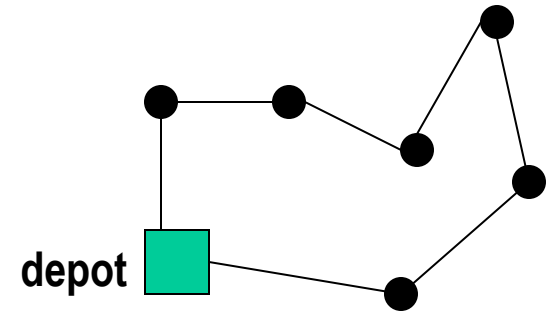
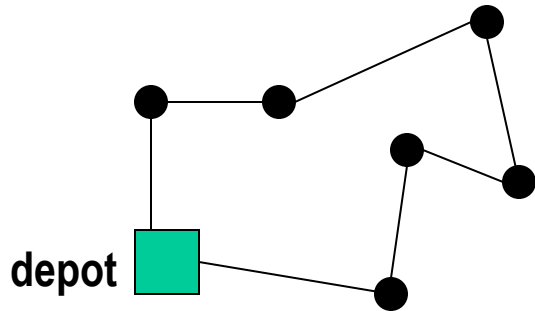


Pure Pickup or Delivery

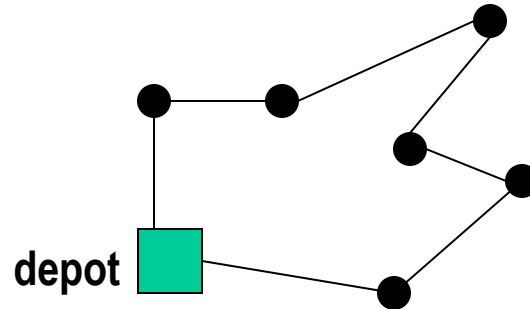
- **Delivery:** Load vehicle at depot. Design route to deliver to many customers (destinations).
 - **Pickup:** Design route to pickup orders from many customers and deliver to depot.
-
- Examples:
 - UPS, FedEx, etc.
 - Manufacturers & carriers.
 - Carpools, school buses, etc.



Pure Pickup or Delivery

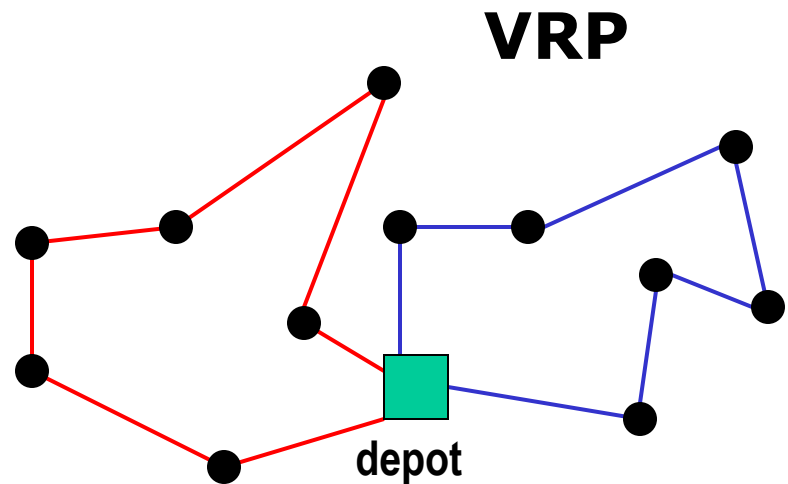
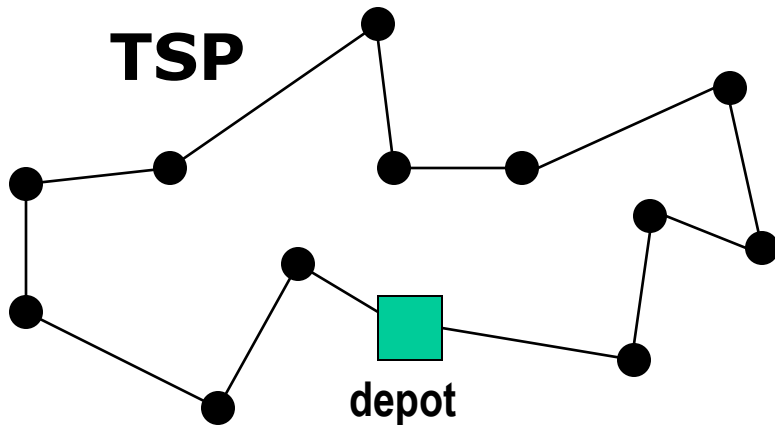


Which route is best????



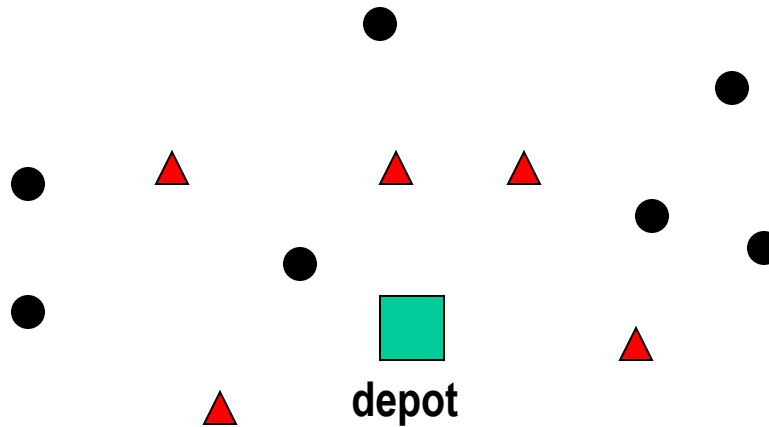
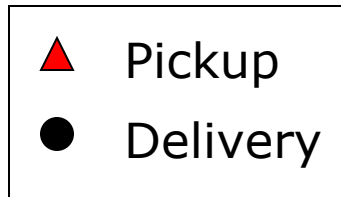
TSP & VRP

- **TSP: Travelling Salesman Problem**
 - One route can serve all orders.
- **VRP: Vehicle Routing Problem**
 - More than one route is required to serve all orders.





Mixed Pickup & Delivery

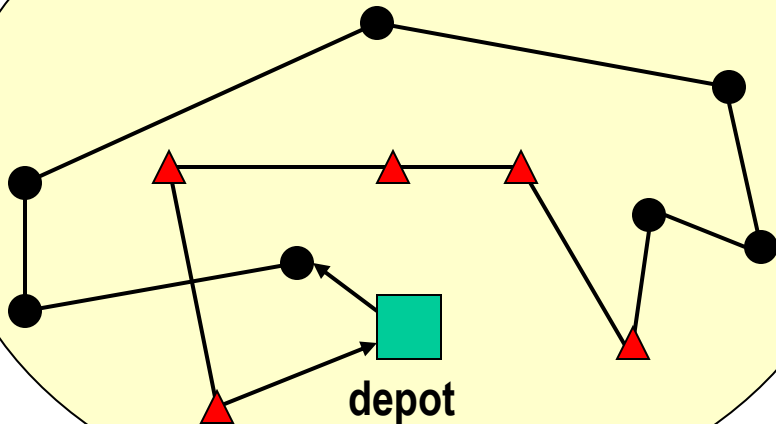


- Can pickups and deliveries be made on same trip?
- Can they be interspersed?

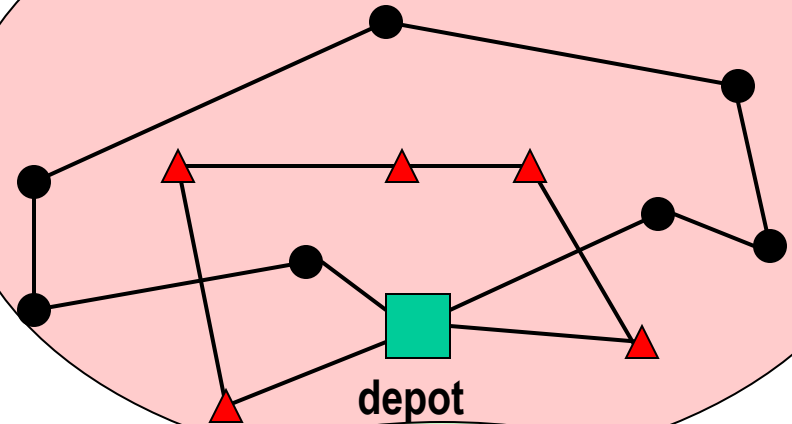
Mixed Pickup & Delivery

- ▲ Pickup
- Delivery

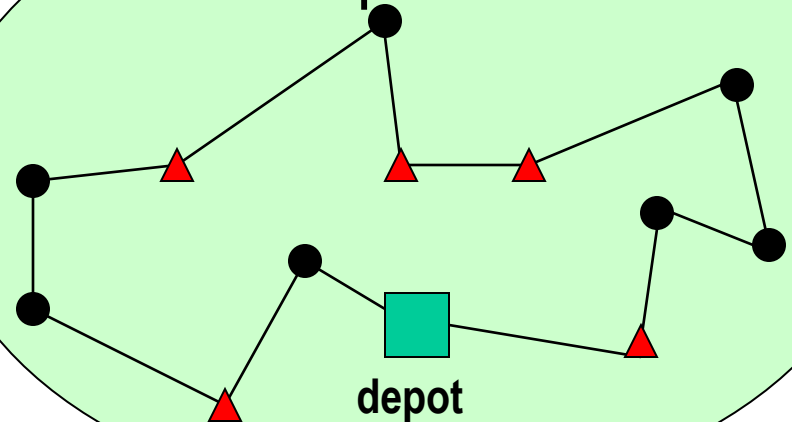
One Route
Not Interspersed



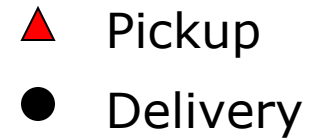
Separate routes



Interspersed



Interspersed Routes



- For clockwise trip:

- Load at depot

ACDFIJK

- Stop 1: Deliver A

CDFIJK

- Stop 2: Pickup B

BCDFIJK

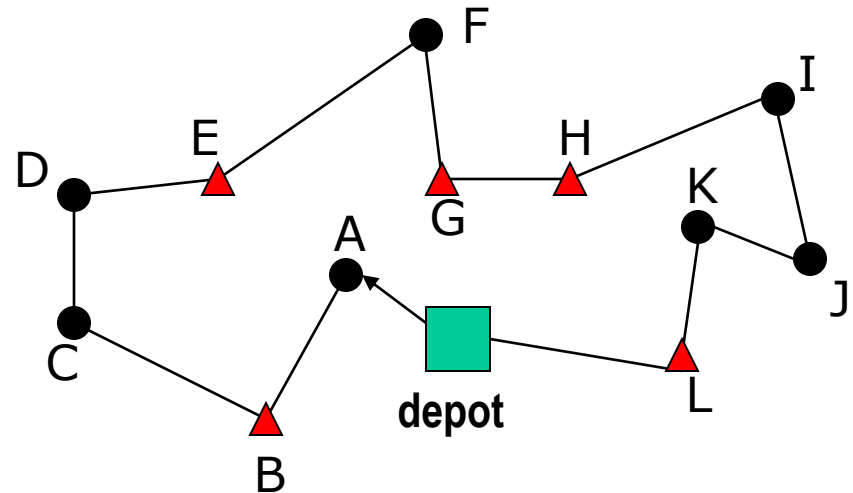
- Stop 3: Deliver C

BDFIJK

- Stop 4: Deliver D

BFIJK

- etc.

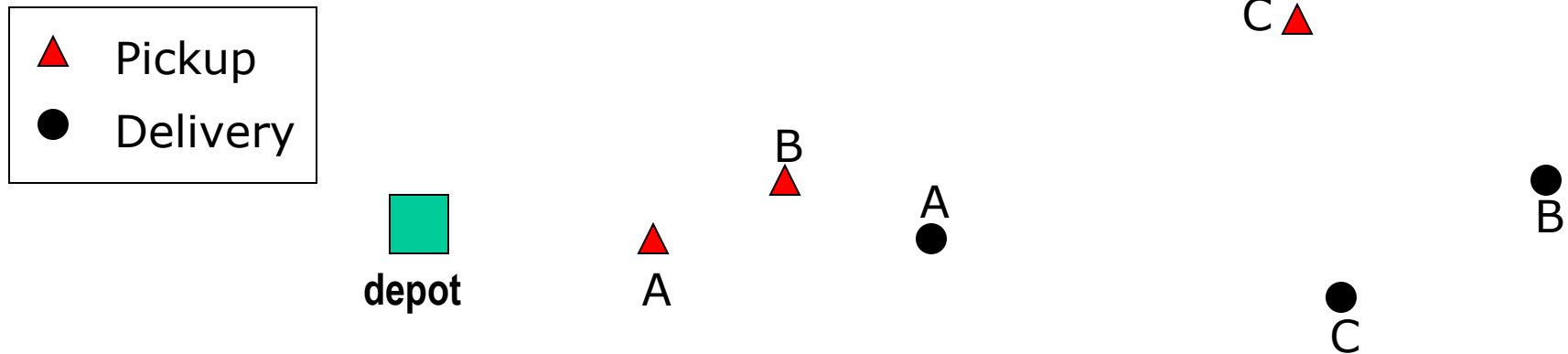


Delivering C requires moving B

Delivering D requires moving B

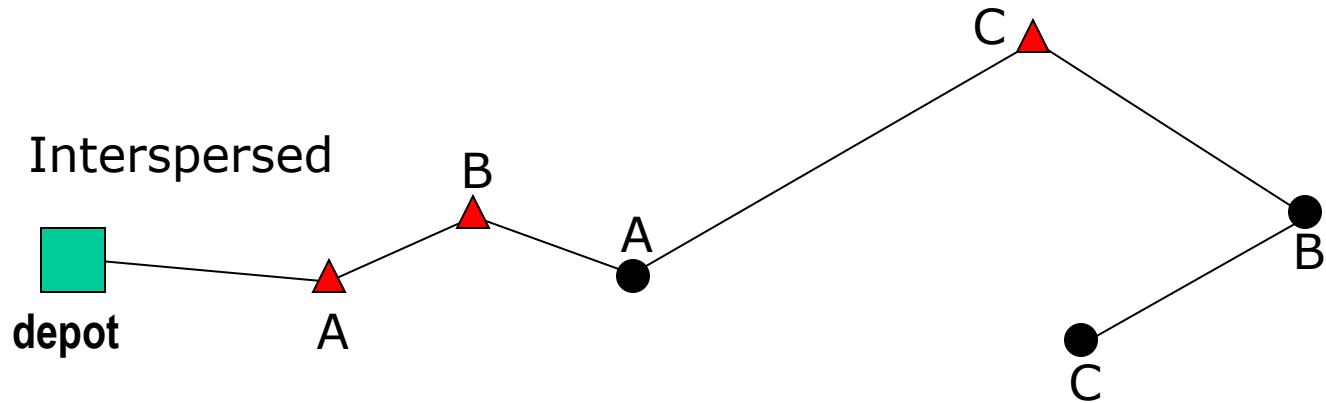
Pickup-Delivery Problems

- Pickup at one or more origin and delivery to one or more destinations.
- Often long haul trips.

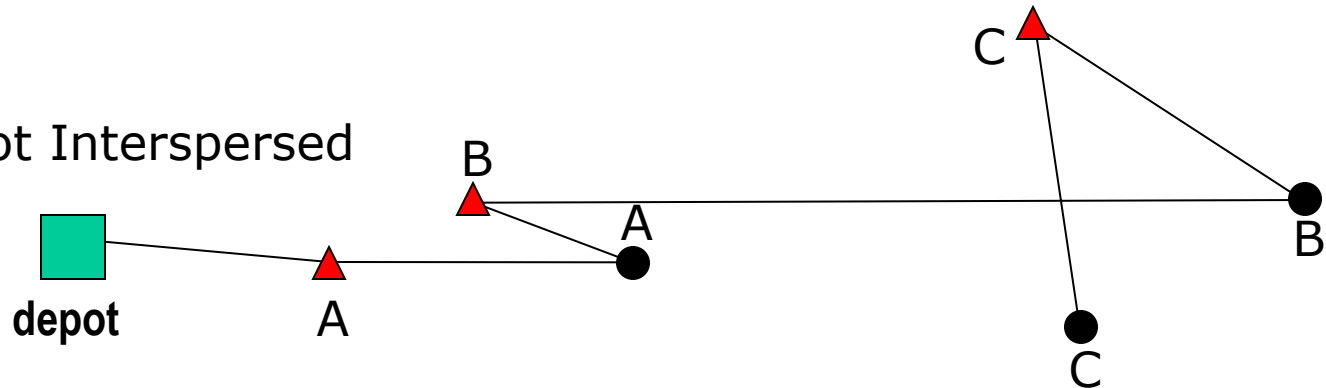


Intersperse Pickups and Deliveries?

- Can pickups and deliveries be interspersed?

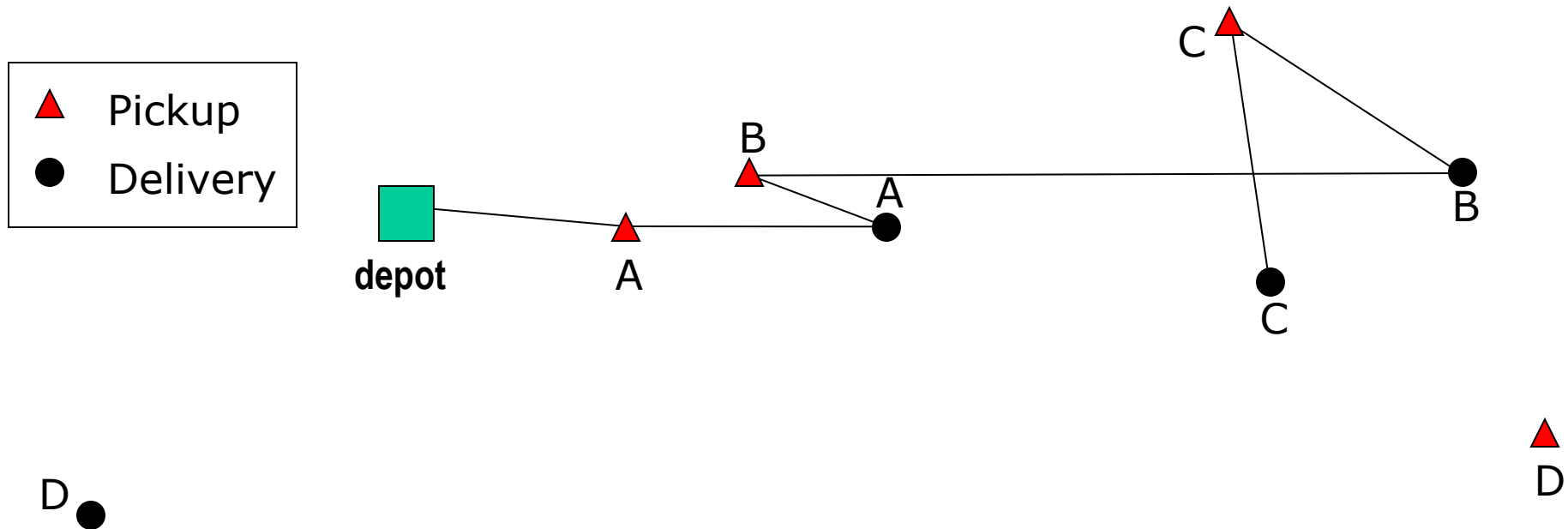


Not Interspersed



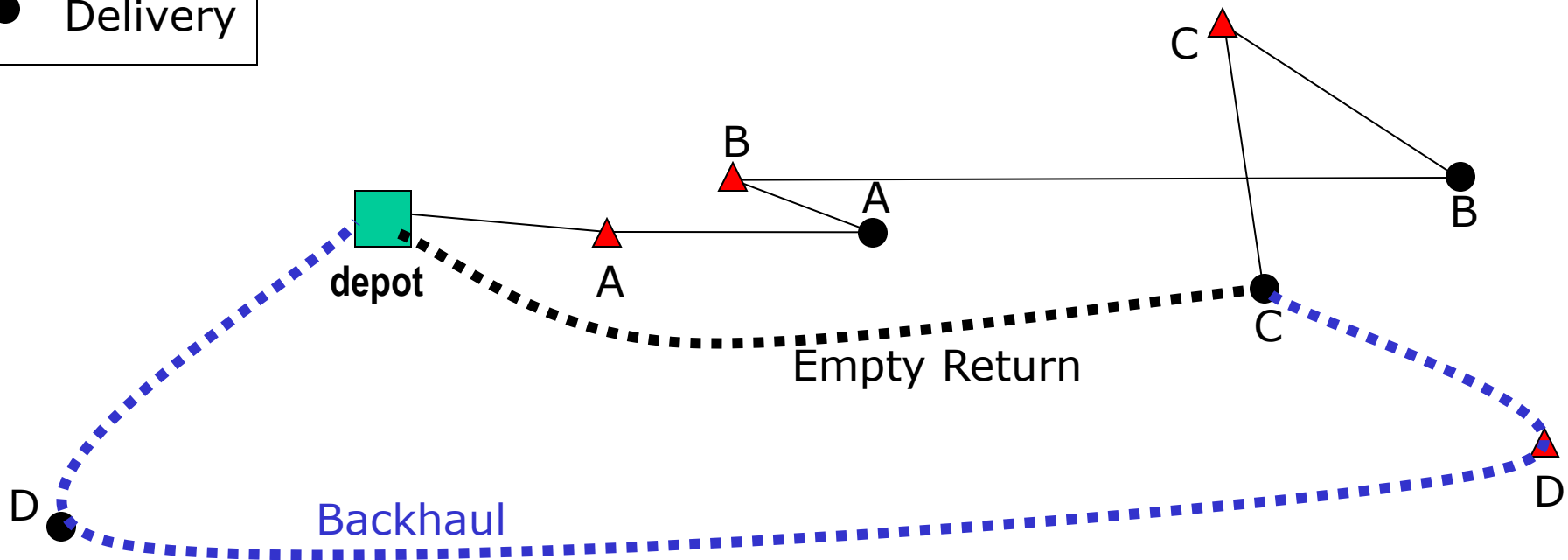
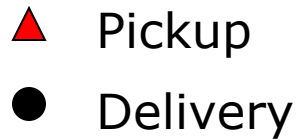
Backhauls

- If vehicle does not end at depot, should it return empty (deadhead) or find a backhaul?
 - How far out of the way should it look for a backhaul?



Backhauls

- Compare profit from deadheading and carrying backhaul.



Complications

- Multiple vehicle types.
- Multiple vehicle capacities.
 - Weight, Cubic feet, Floor space, Value.
- Many Costs:
 - Fixed charge.
 - Variable costs per loaded mile & per empty mile.
 - Waiting time; Layover time.
 - Cost per stop (handling).
 - Loading and unloading cost.
- Priorities for customers or orders.

More Complications

- Time windows for pickup and delivery.
 - Hard vs. soft
- Compatibility
 - Vehicles and customers.
 - Vehicles and orders.
 - Order types.
 - Drivers and vehicles.
- Driver rules
 - Max drive duration = 10 hrs. before 8 hr. break.
 - Max work duration = 15 hrs. before 8 hr break.
 - Max trip duration = 144 hrs.

Simple Models

- Homogeneous vehicles.
- One capacity (weight or volume).
- Minimize distance.
- No time windows or one time window per customer.
- No compatibility constraints.
- No driver rules.

Simplest Model: 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.
- No Driver rules.

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*.

- Direction does not affect cost.

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

- Direction does affect cost.

Applications of TSP

- Many more applications of TSP come from logistics, transportation
 - Scheduling of buses to pick up school children
 - Pickup and Ride
 -
 - Air force Logistics
- Yet TSP & VRP provide a perfect test ground for the development of novel ideas
 - Simulated Annealing, GA, Tabu Search
 - Lagrangean relaxation, Branch and Cut, Column Generation.



TSP Solutions

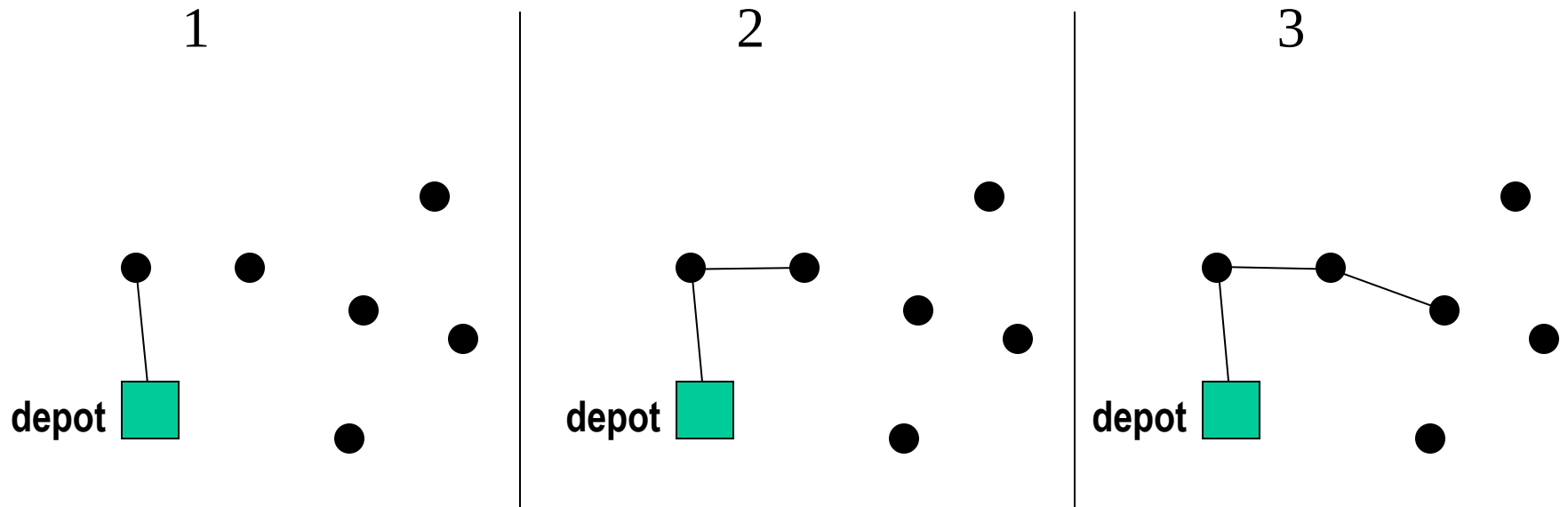
- Heuristics
 - **Construction:** build a feasible route.
 - **Improvement:** improve a feasible route.
 - Not necessarily optimal, but fast.
 - Performance depends on problem.
 - Worst case performance may be very poor.
- Exact algorithms
 - Integer programming.
 - Branch and bound.
 - Optimal, but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

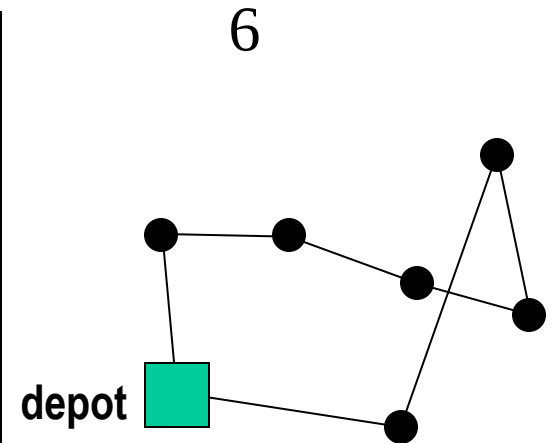
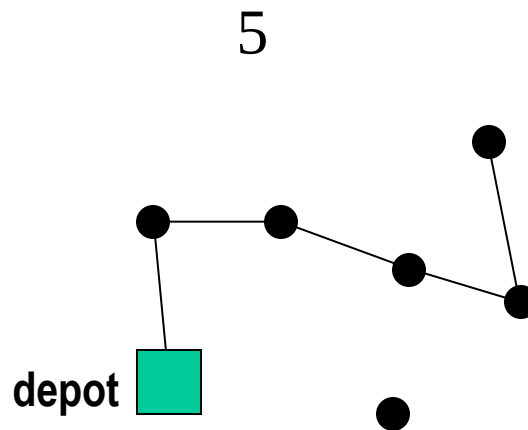
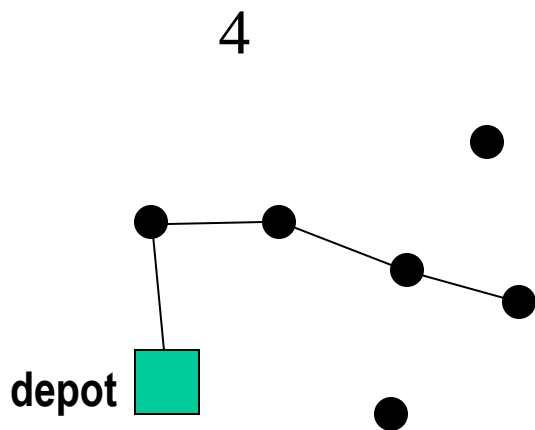
Nearest Neighbor

- Add nearest customer to end of the route.



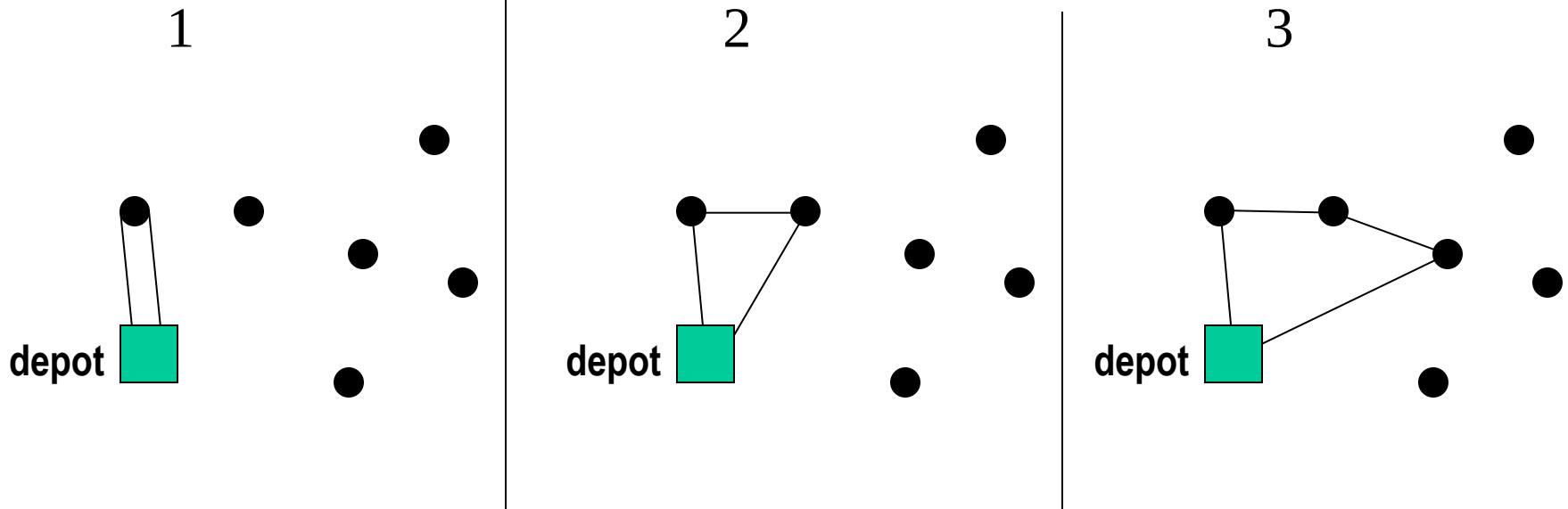
Nearest Neighbor

- Add nearest customer to end of the route.



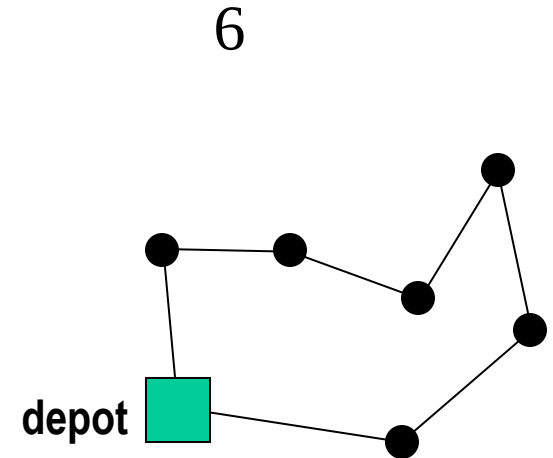
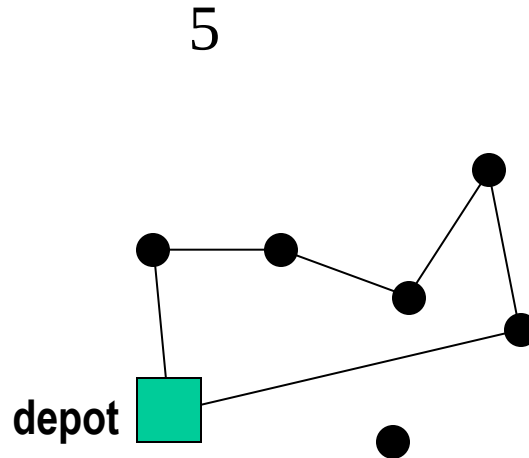
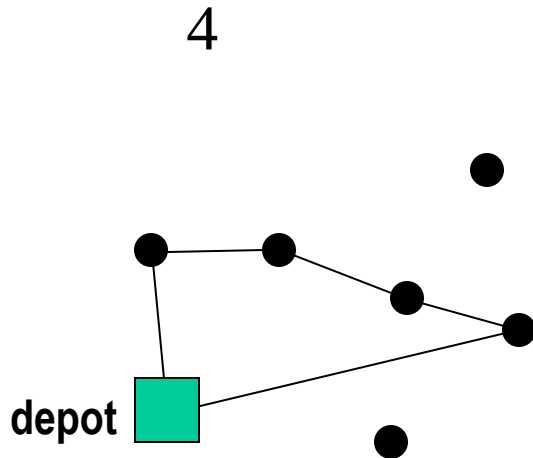
Nearest Insertion

- Insert customer closest to the route in the best sequence.



Nearest Insertion

- Insert customer closest to the route in the best sequence.



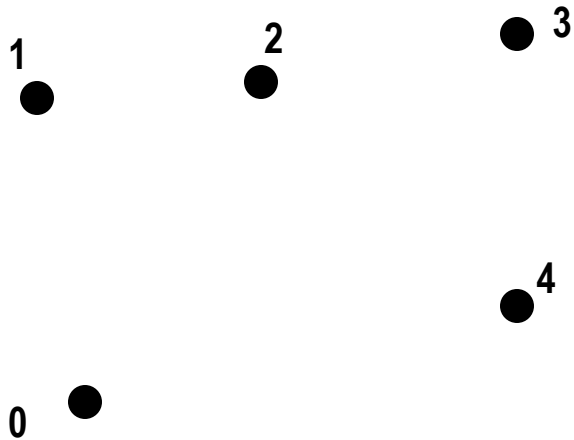


Savings Method

1. Select any city as the “depot” and call it city “0”.
 - Start with separate one stop routes from depot to each customer.
2. Calculate all *savings* for joining two customers and eliminating a trip back to the depot.
$$S_{ij} = C_{i0} + C_{0j} - C_{ij}$$
3. Order savings from largest to smallest.
4. Form route by linking customers according to savings.
 - Do not break any links formed earlier.
 - Stop when all customers are on the route.

Savings Method Example

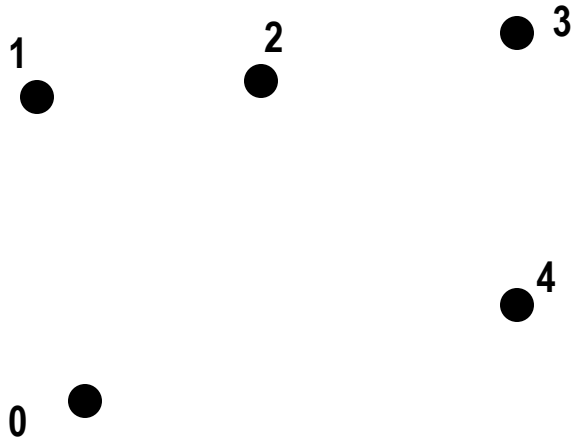
Given 5 customers and the costs (distances) between them.



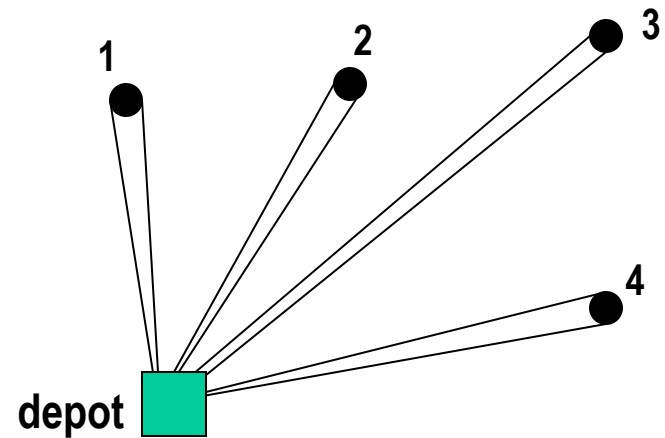
		j				
		0	1	2	3	4
i	0	-	8	9	13	10
	1	8	-	4	11	13
	2	9	4	-	5	8
	3	13	11	5	-	7
	4	10	13	8	7	-

Savings Method Example

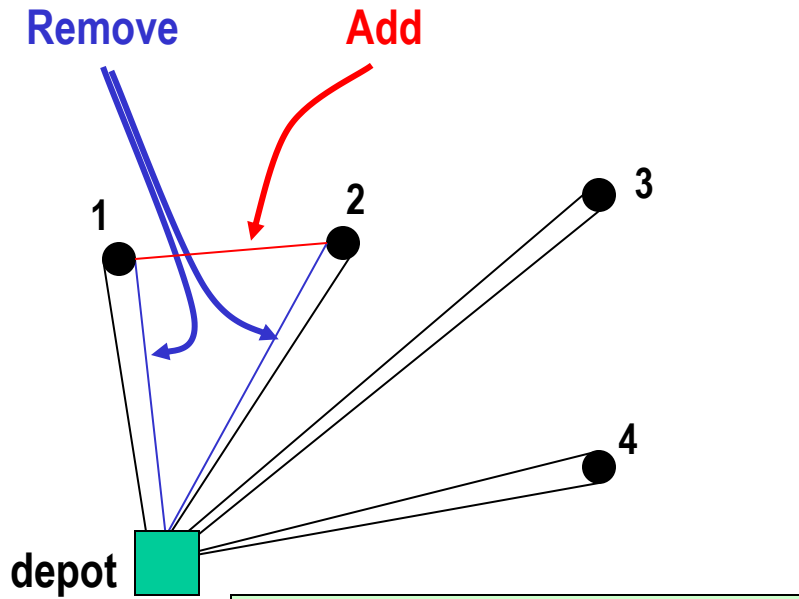
Given 5 customers, select the lower left as the depot.



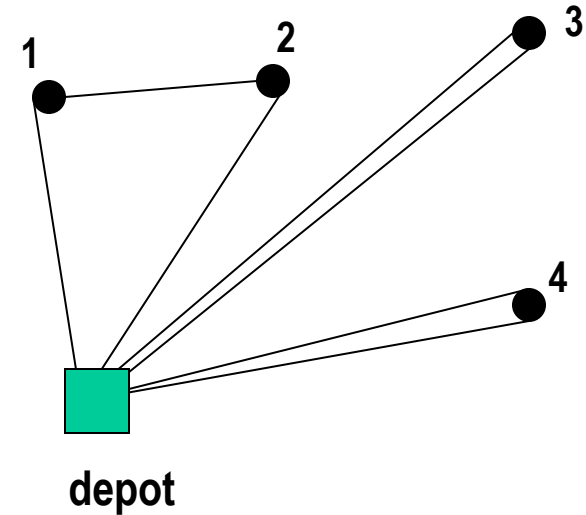
Conceptually form routes from the depot to each customer.



Savings Method: S_{12}



$$\text{Savings} = S_{12} = C_{10} + C_{02} - C_{12} \\ = 8 + 9 - 4 = 13$$



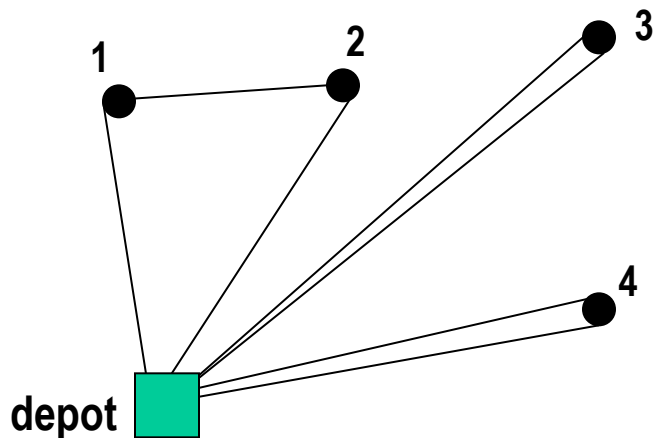
		j				
i	C_{ij}	0	1	2	3	4
	0	-	8	9	13	10
1	8		-	4	11	13
2	9		4	-	5	8
3	13		11	5	-	7
4	10		13	8	7	-

Savings Method

$$S_{12} = C_{10} + C_{02} - C_{12}$$

Note: $S_{21} = C_{20} + C_{01} - C_{21}$

so $S_{12} = S_{21}$

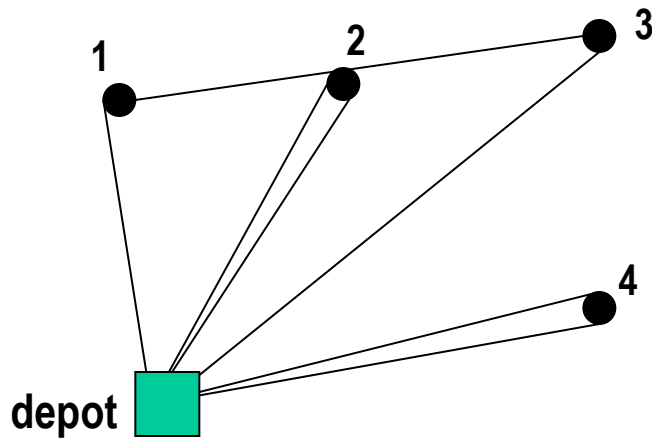


If problem is symmetric, then $S_{ij} = S_{ji}$, so $S_{21} = S_{12}$, $S_{32} = S_{23}$, etc. There are $(n-1)(n-2)/2$ savings to calculate.

If problem is asymmetric, then all s_{ij} 's must be calculated. There are $(n-1)(n-2)$ savings to calculate.

Savings Method: S_{13}

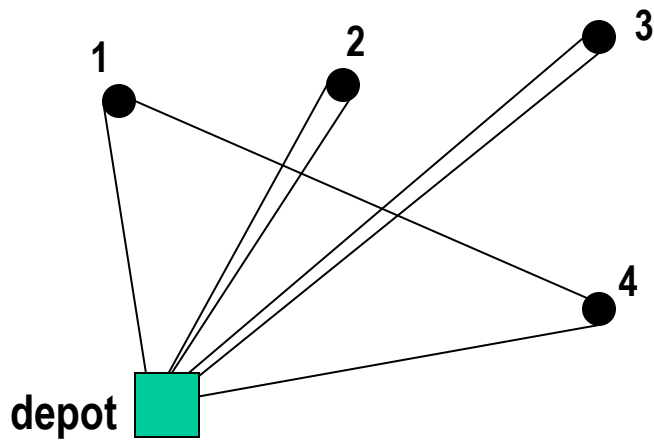
$$\begin{aligned} S_{13} &= C_{10} + C_{03} - C_{13} \\ &= 8 + 13 - 11 = 10 \end{aligned}$$



		j				
C _{ij}		0	1	2	3	4
i	0	-	8	9	13	10
	1	8	-	4	11	13
	2	9	4	-	5	8
	3	13	11	5	-	7
	4	10	13	8	7	-

Savings Method: S_{14}

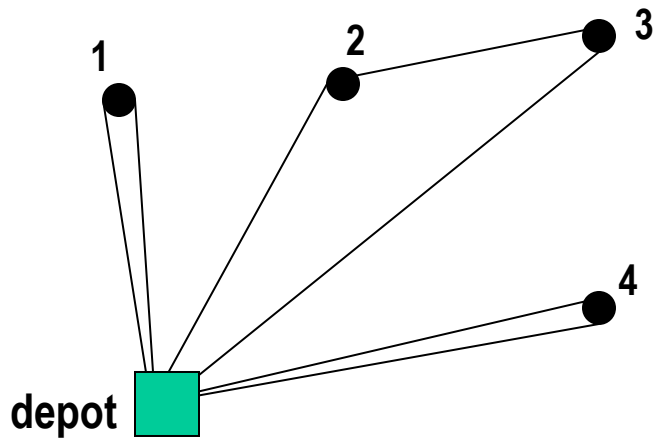
$$\begin{aligned} S_{14} &= C_{10} + C_{04} - C_{14} \\ &= 8 + 10 - 13 = 5 \end{aligned}$$



		j				
i	C_{ij}	0	1	2	3	4
	0	-	8	9	13	10
1	1	8	-	4	11	13
2	2	9	4	-	5	8
3	3	13	11	5	-	7
4	4	10	13	8	7	-

Savings Method: S_{23}

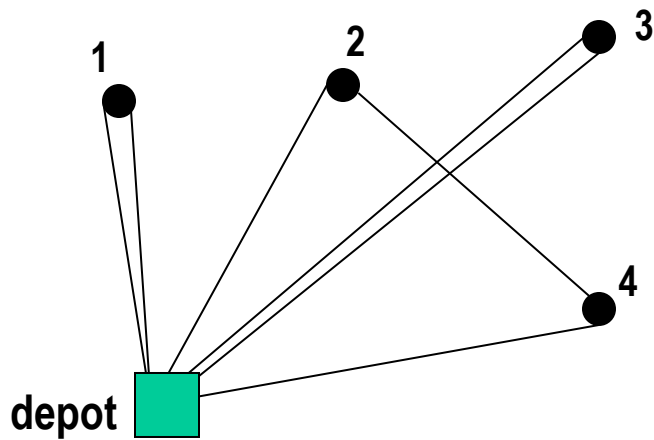
$$\begin{aligned} S_{23} &= C_{20} + C_{03} - C_{23} \\ &= 9 + 13 - 5 = 17 \end{aligned}$$



		j				
C _{ij}		0	1	2	3	4
i	0	-	8	9	13	10
	1	8	-	4	11	13
	2	9	4	-	5	8
	3	13	11	5	-	7
	4	10	13	8	7	-

Savings Method: S_{24}

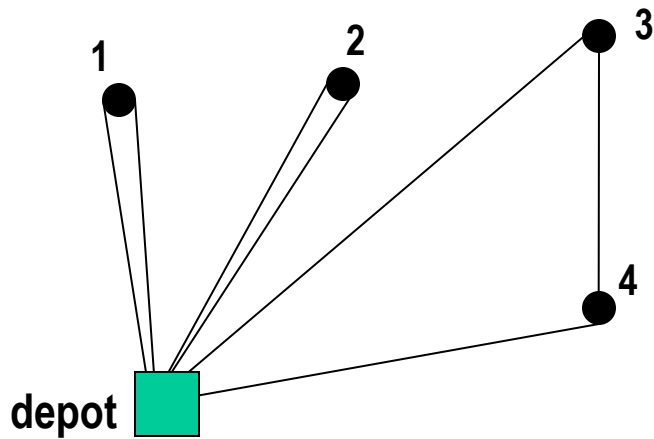
$$\begin{aligned} S_{24} &= C_{20} + C_{04} - C_{24} \\ &= 9 + 10 - 8 = 11 \end{aligned}$$



		j				
i	C_{ij}	0	1	2	3	4
	0	-	8	9	13	10
1	0	8	-	4	11	13
2	0	9	4	-	5	8
3	0	13	11	5	-	7
4	0	10	13	8	7	-

Savings Method: S_{34}

$$\begin{aligned} S_{34} &= C_{30} + C_{04} - C_{34} \\ &= 13 + 10 - 7 = 16 \end{aligned}$$



		j				
i	C_{ij}	0	1	2	3	4
	0	-	8	9	13	10
1	1	8	-	4	11	13
2	2	9	4	-	5	8
3	3	13	11	5	-	7
4	4	10	13	8	7	-

Savings Method

- Order savings from largest to smallest.

$$S_{23} (= S_{23}) = 17$$

$$S_{34} (= S_{43}) = 16$$

$$S_{12} (= S_{21}) = 13$$

$$S_{24} (= S_{42}) = 11$$

$$S_{13} (= S_{31}) = 10$$

$$S_{14} (= S_{41}) = 5$$

Savings Method

- Form route by linking customers according to savings.

S_{23}



Link 2 and 3.

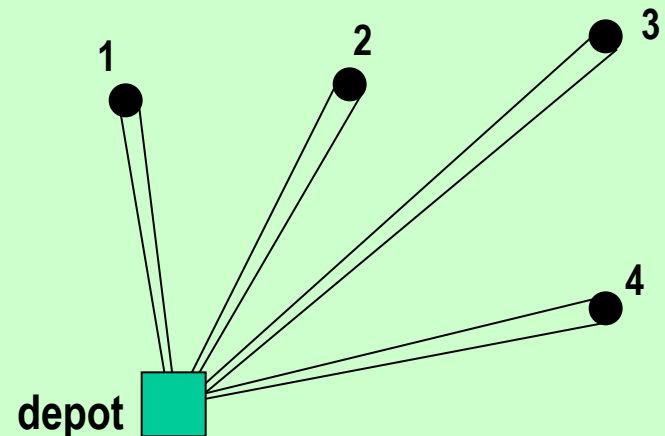
S_{34}

S_{12}

S_{24}

S_{13}

S_{14}



Savings Method

- Form route by linking customers according to savings.

S_{23} 0-2-3-0

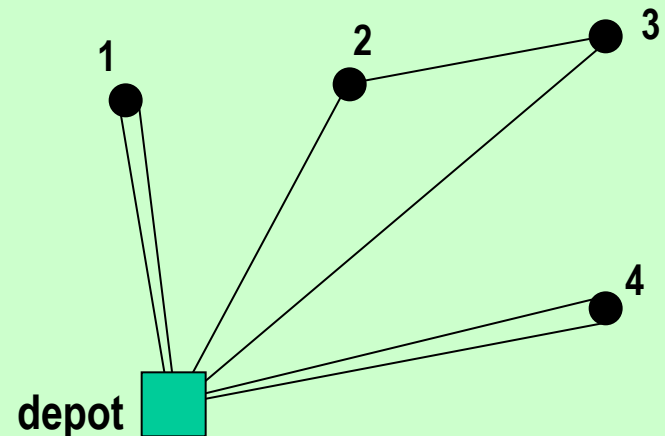
S_{34}

S_{12}

S_{24}

S_{13}

S_{14}



Savings Method

- Form route by linking customers according to savings.

S_{23} 0-2-3-0

S_{34} 

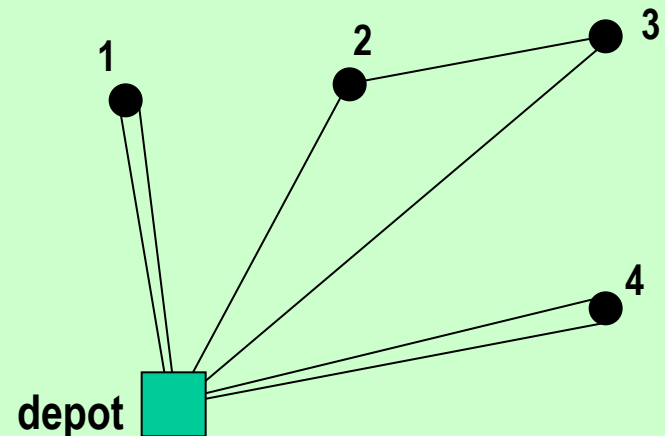
S_{12}

S_{24}

S_{13}

S_{14}

Link 3 and 4.
Do not break earlier links.



Savings Method

- Form route by linking customers according to savings.

S_{23} 0-2-3-0

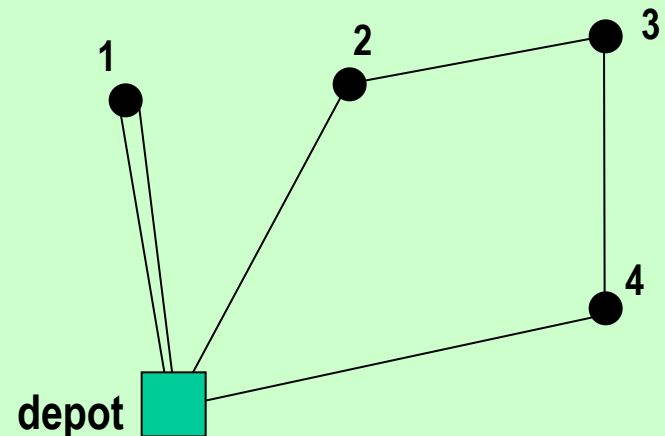
S_{34} 0-2-3-4-0

S_{12}

S_{24}

S_{13}

S_{14}



Savings Method

- Form route by linking customers according to savings.

S_{23} 0-2-3-0

S_{34} 0-2-3-4-0

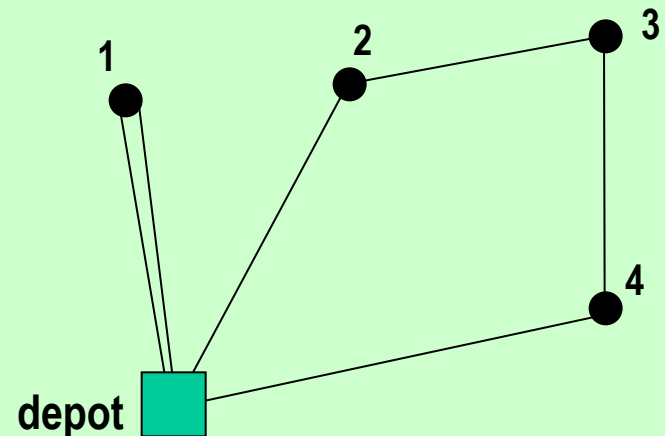
S_{12} →

S_{24}

S_{13}

S_{14}

Link 1 and 2.
Do not break earlier links.

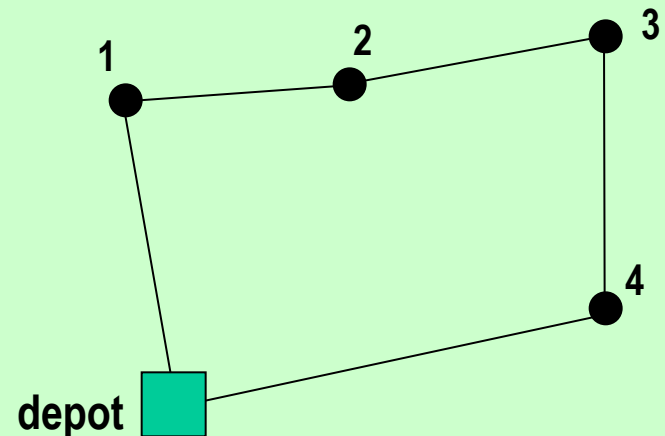


Savings Method

- Form route by linking customers according to savings.

S_{23}	0-2-3-0
S_{34}	0-2-3-4-0
S_{12}	0-1-2-3-4-0
S_{24}	
S_{13}	
S_{14}	

Done!



Larger Problem

- Find the best route using the following savings, in decreasing order, for a symmetric vehicle routing problem:

S_{35}

S_{34}

S_{45}

S_{36}

S_{56}

S_{23}

S_{46}

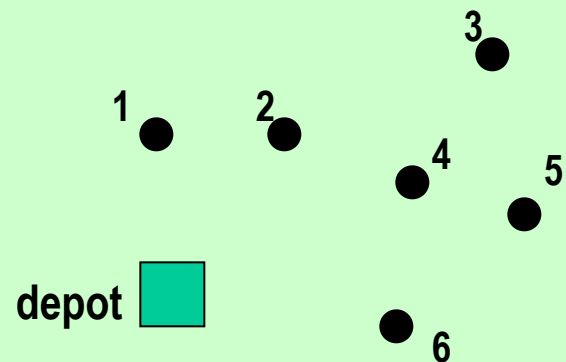
S_{24}

S_{25}

S_{12}

S_{26}

etc.



Larger Problem

- Form route by linking customers according to savings.

S_{35}



Link 3 and 5.

S_{34}

S_{45}

S_{36}

S_{56}

S_{23}

S_{46}

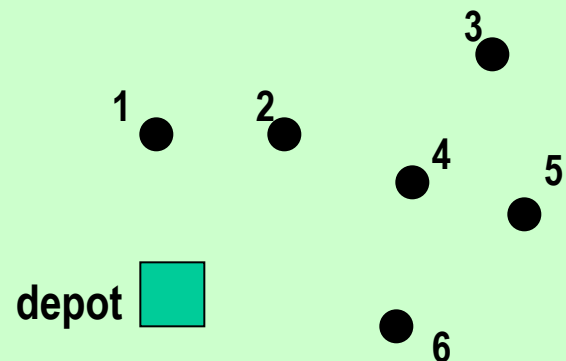
S_{24}

S_{25}

S_{12}

S_{26}

etc.



Larger Problem

- Form route by linking customers according to savings.

S_{35} 0-3-5-0

S_{34}

S_{45}

S_{36}

S_{56}

S_{23}

S_{46}

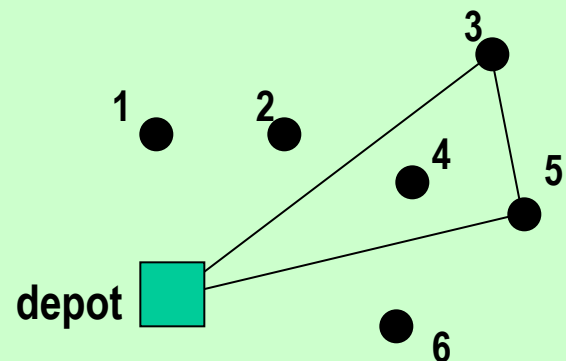
S_{24}

S_{25}

S_{12}

S_{26}

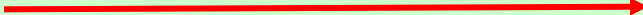
etc.



Larger Problem

- Form route by linking customers according to savings.

S_{35} 0-3-5-0

S_{34} 

S_{45}

S_{36}

S_{56}

S_{23}

S_{46}

S_{24}

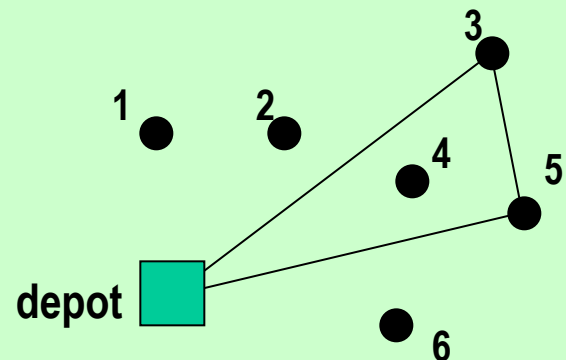
S_{25}

S_{12}

S_{26}

etc.

Link 3 and 4.
Do not break earlier links.



Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35} 0-3-5-0

S_{34} 0-4-3-5-0

S_{45}

S_{36}

S_{56}

S_{23}

S_{46}

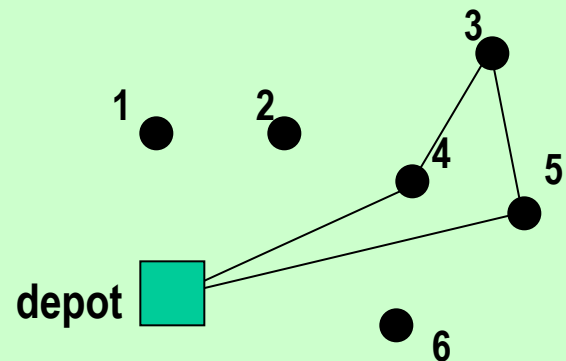
S_{24}

S_{25}

S_{12}

S_{26}

etc.

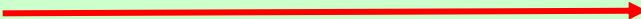


Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35} 0-3-5-0

S_{34} 0-4-3-5-0

S_{45} 

S_{36}

S_{56}

S_{23}

S_{46}

S_{24}

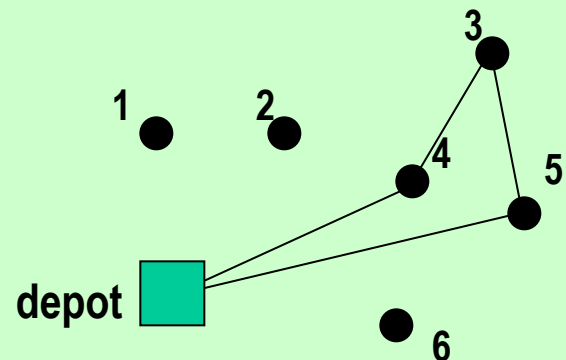
S_{25}

S_{12}

S_{26}

etc.

Link 4 and 5.
Do not break earlier links.



Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35} 0-3-5-0

S_{34} 0-4-3-5-0

S_{45} skip

S_{36}

S_{56}

S_{23}

S_{46}

S_{24}

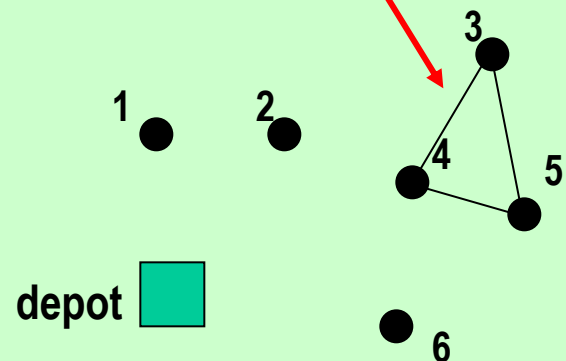
S_{25}

S_{12}

S_{26}

etc.

Not feasible!



Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35} 0-3-5-0

S_{34} 0-4-3-5-0

S_{45} skip

S_{36} →

Link 3 and 6.
Do not break earlier links.

S_{56}

S_{23}

S_{46}

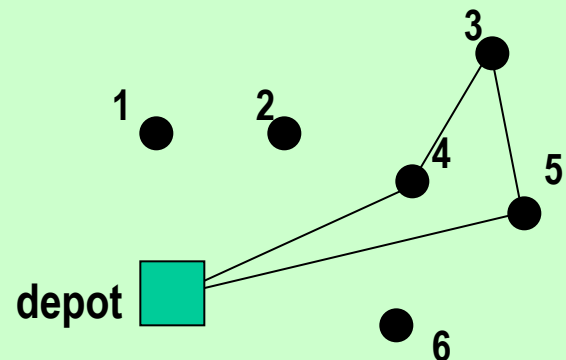
S_{24}

S_{25}

S_{12}

S_{26}

etc.



Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35} 0-3-5-0

S_{34} 0-4-3-5-0

S_{45} skip

S_{36} skip

S_{56}

S_{23}

S_{46}

S_{24}

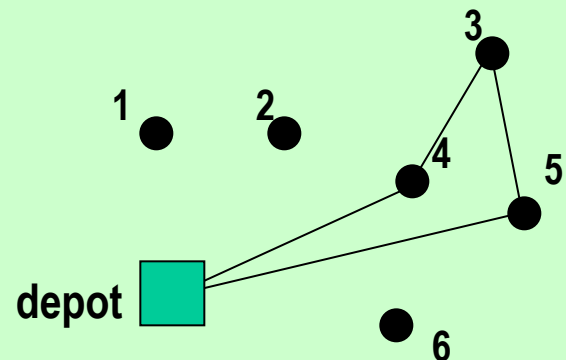
S_{25}

S_{12}

S_{26}

etc.

Cannot link 3 & 6 without breaking 4-3 or 3-5.

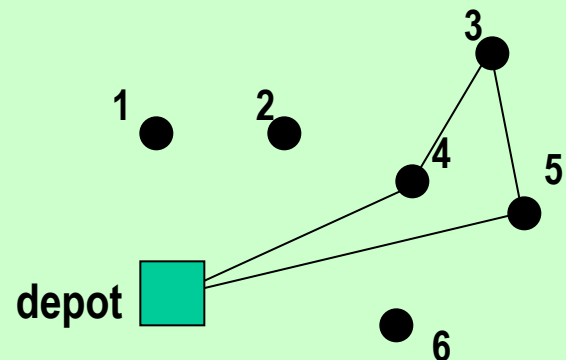


Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35} 0-3-5-0
 S_{34} 0-4-3-5-0
 S_{45} skip
 S_{36} skip
 S_{56} ←
 S_{23}
 S_{46}
 S_{24}
 S_{25}
 S_{12}
 S_{26}
etc.

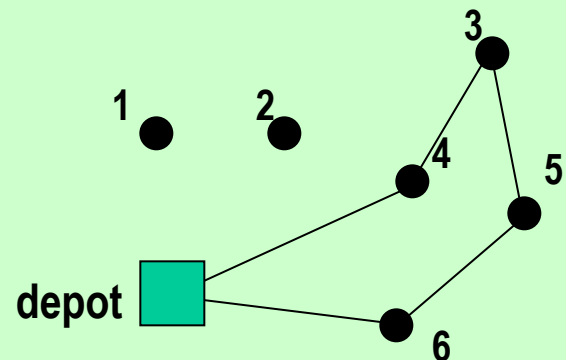
Link 5 and 6.
Do not break earlier links.



Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35}	0-3-5-0
S_{34}	0-4-3-5-0
S_{45}	skip
S_{36}	skip
S_{56}	0-4-3-5-6-0
S_{23}	
S_{46}	
S_{24}	
S_{25}	
S_{12}	
S_{26}	
etc.	

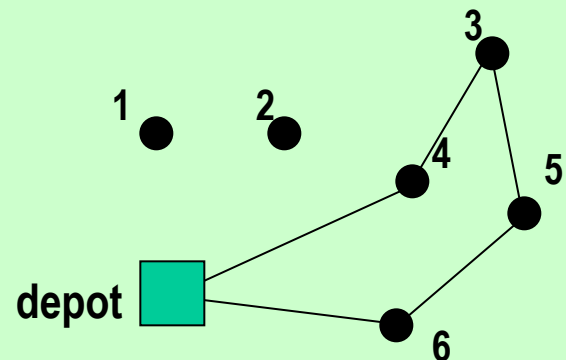


Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35}	0-3-5-0
S_{34}	0-4-3-5-0
S_{45}	skip
S_{36}	skip
S_{56}	0-4-3-5-6-0
S_{23}	
S_{46}	
S_{24}	
S_{25}	
S_{12}	
S_{26}	
etc.	

Link 2 and 3.
Do not break earlier links.

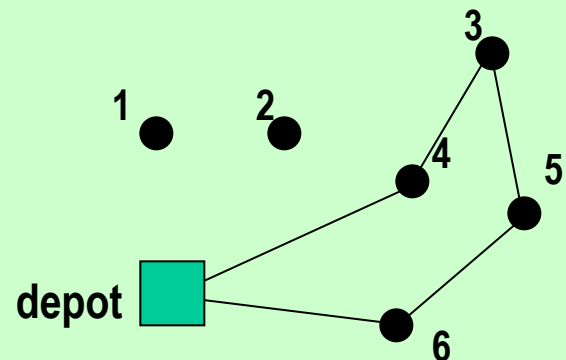


Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35}	0-3-5-0
S_{34}	0-4-3-5-0
S_{45}	skip
S_{36}	skip
S_{56}	0-4-3-5-6-0
S_{23}	skip
S_{46}	
S_{24}	
S_{25}	
S_{12}	
S_{26}	
etc.	

Cannot link 2 & 3 without breaking 4-3 or 3-5.

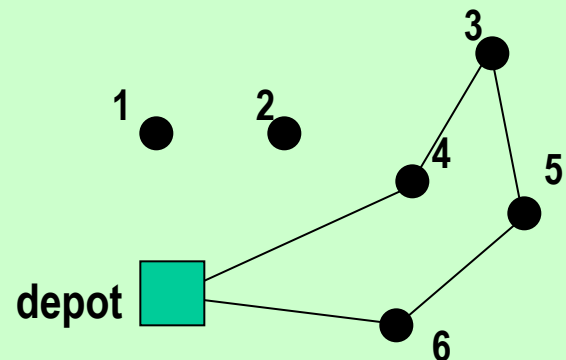


Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35}	0-3-5-0
S_{34}	0-4-3-5-0
S_{45}	skip
S_{36}	skip
S_{56}	0-4-3-5-6-0
S_{23}	skip
S_{46}	
S_{24}	
S_{25}	
S_{12}	
S_{26}	
etc.	

Link 4 and 6.
Do not break earlier links.

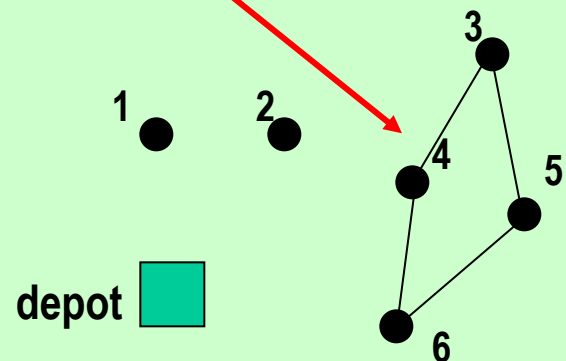


Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35}	0-3-5-0
S_{34}	0-4-3-5-0
S_{45}	skip
S_{36}	skip
S_{56}	0-4-3-5-6-0
S_{23}	skip
S_{46}	skip
S_{24}	
S_{25}	
S_{12}	
S_{26}	
etc.	

Cannot link 4 & 6 and stay feasible.

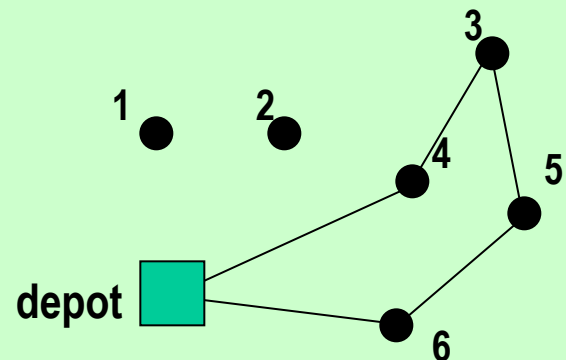


Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35}	0-3-5-0
S_{34}	0-4-3-5-0
S_{45}	skip
S_{36}	skip
S_{56}	0-4-3-5-6-0
S_{23}	skip
S_{46}	skip
S_{24}	
S_{25}	
S_{12}	
S_{26}	
etc.	

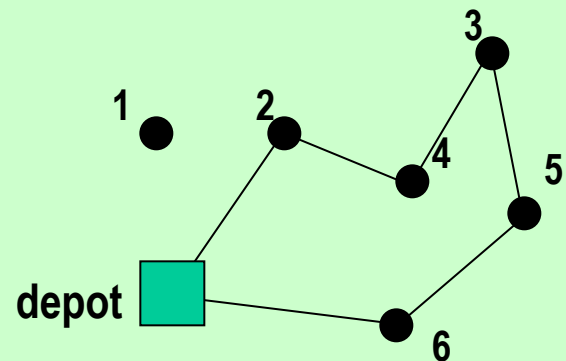
Link 2 and 4.
Do not break earlier links.



Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35}	0-3-5-0
S_{34}	0-4-3-5-0
S_{45}	skip
S_{36}	skip
S_{56}	0-4-3-5-6-0
S_{23}	skip
S_{46}	skip
S_{24}	0-2-4-3-5-6-0
S_{25}	
S_{12}	
S_{26}	
etc.	

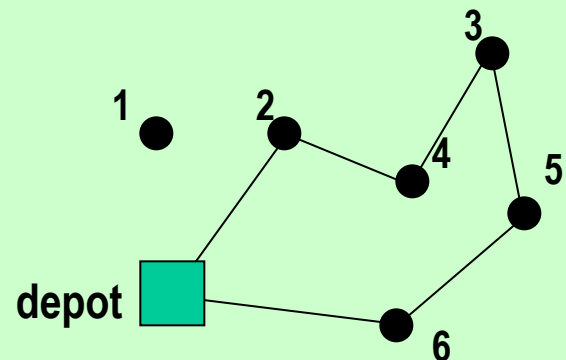


Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35}	0-3-5-0
S_{34}	0-4-3-5-0
S_{45}	skip
S_{36}	skip
S_{56}	0-4-3-5-6-0
S_{23}	skip
S_{46}	skip
S_{24}	0-2-4-3-5-6-0
S_{25}	
S_{12}	
S_{26}	
etc.	

Link 2 and 5.
Do not break earlier links.

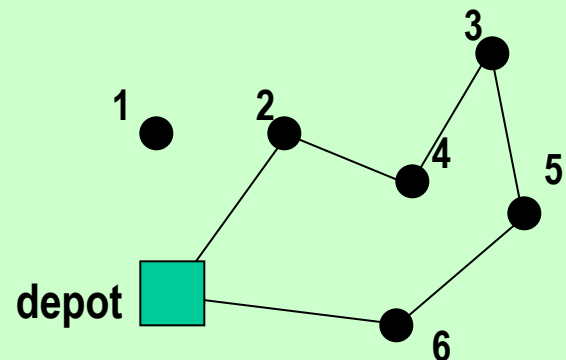


Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35}	0-3-5-0
S_{34}	0-4-3-5-0
S_{45}	skip
S_{36}	skip
S_{56}	0-4-3-5-6-0
S_{23}	skip
S_{46}	skip
S_{24}	0-2-4-3-5-6-0
S_{25}	skip
S_{12}	
S_{26}	
etc.	

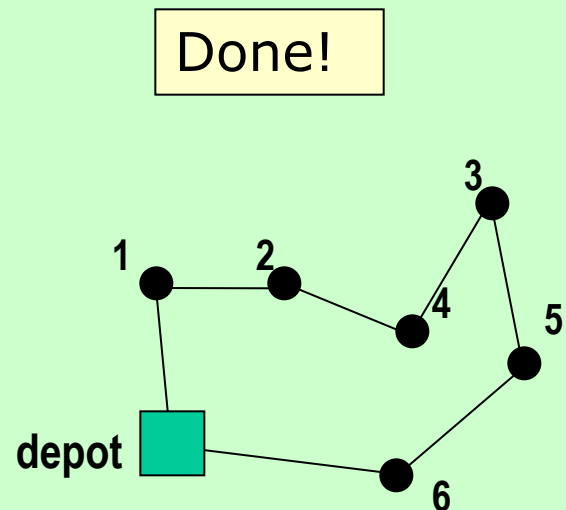
Link 1 and 2.
Do not break earlier links.



Larger Problem - Solution

- Form route by linking customers according to savings.

S_{35}	0-3-5-0
S_{34}	0-4-3-5-0
S_{45}	skip
S_{36}	skip
S_{56}	0-4-3-5-6-0
S_{23}	skip
S_{46}	skip
S_{24}	0-2-4-3-5-6-0
S_{25}	skip
S_{12}	0-1-2-4-3-5-6-0
S_{26}	
etc.	



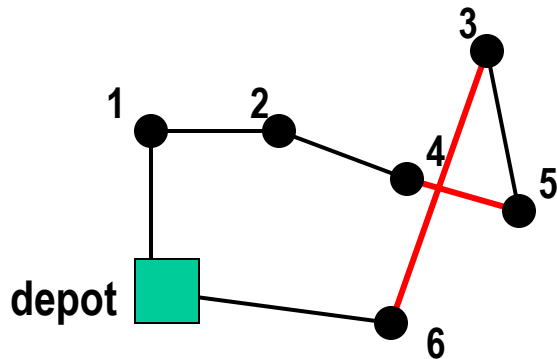
Route Improvement Heuristics

- 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.

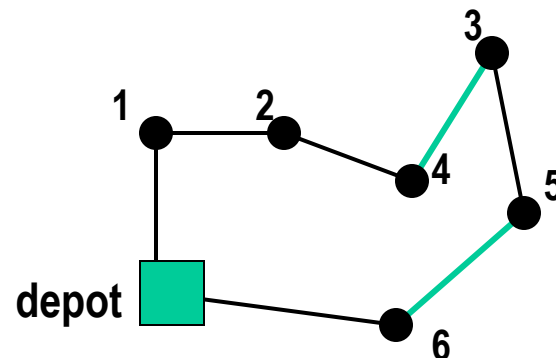
K-opt Exchange

- 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



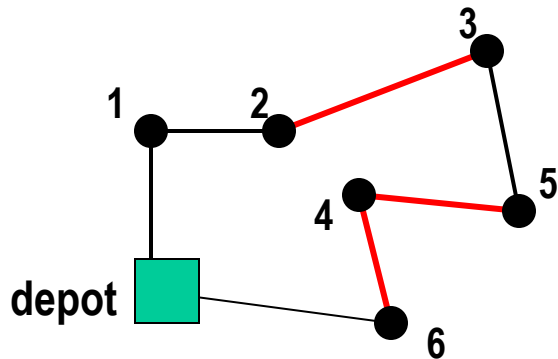
Improved TSP tour



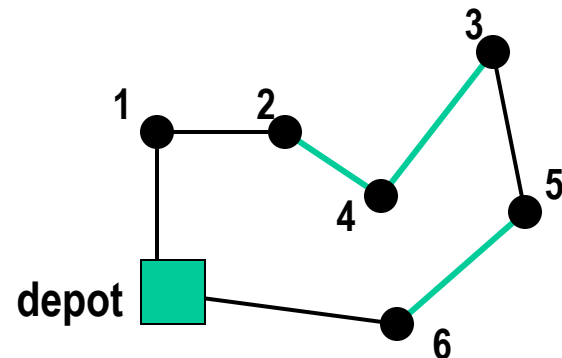
3-opt Exchange

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

Original TSP tour

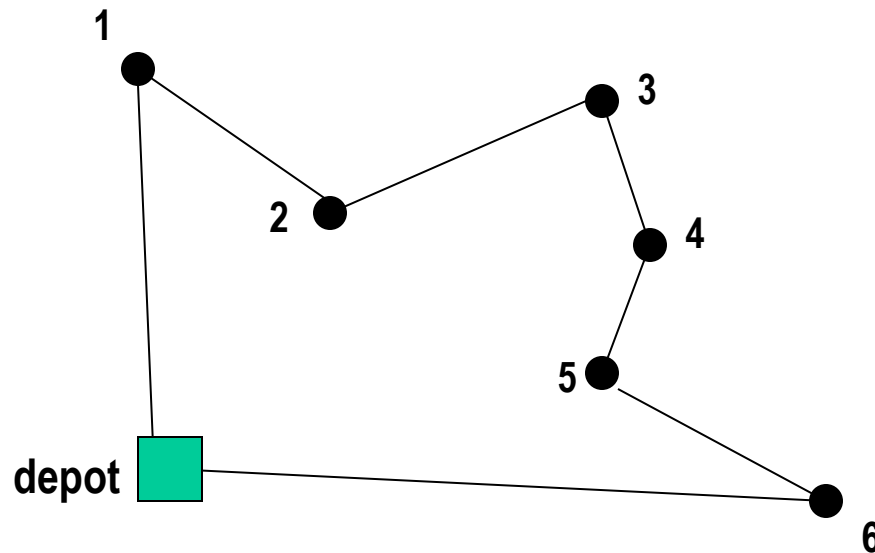


Improved TSP tour



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.

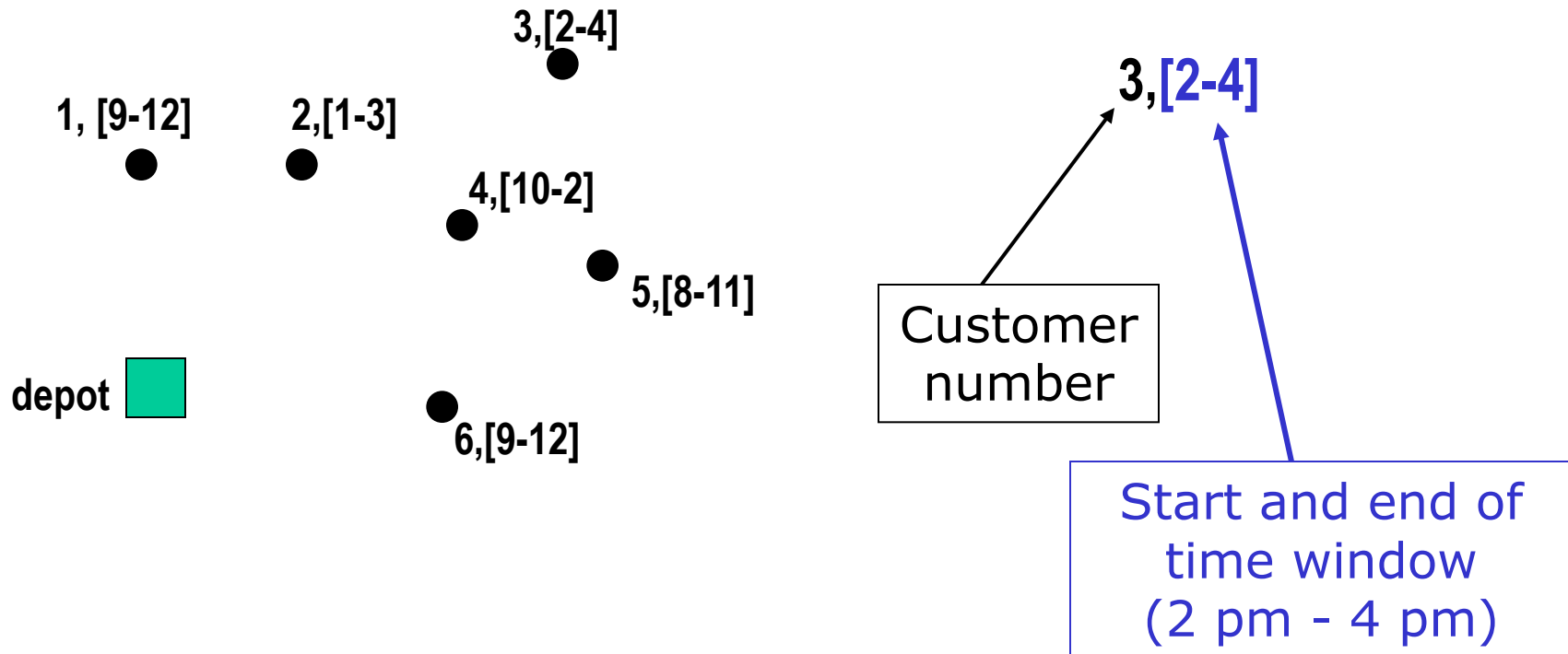


TSP - Integer Programming

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

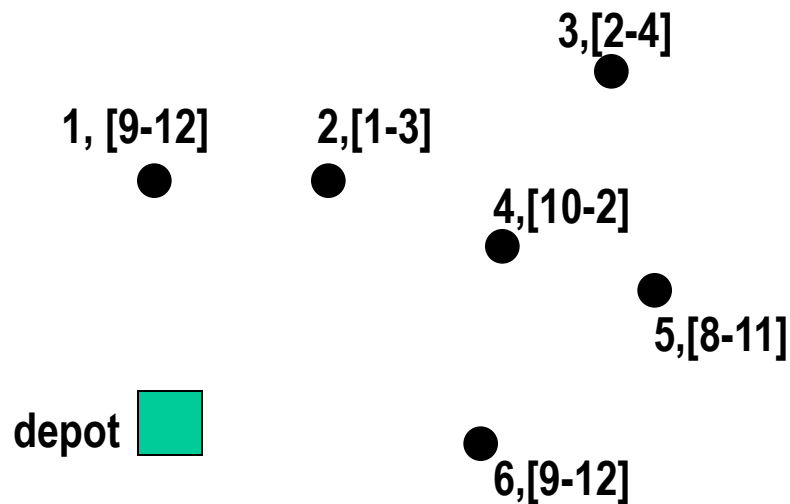
Time Windows

- Problems with time windows involve routing and scheduling.



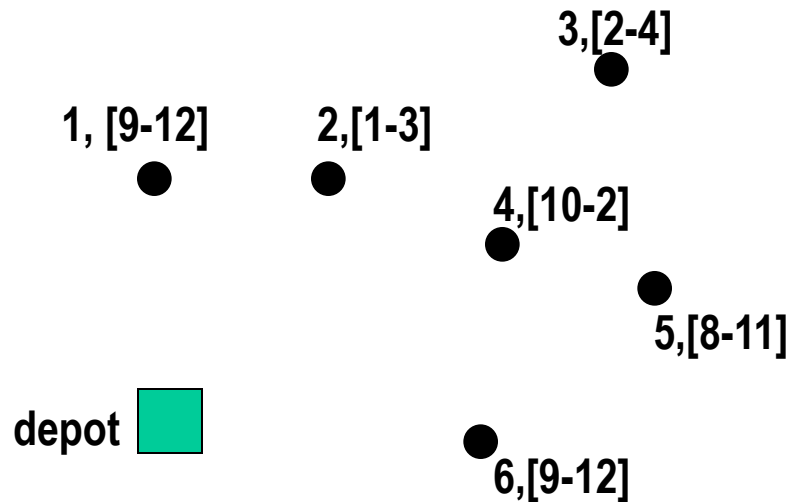
Routing with Time Windows

Route must be built with both
time windows &
geography in
mind!



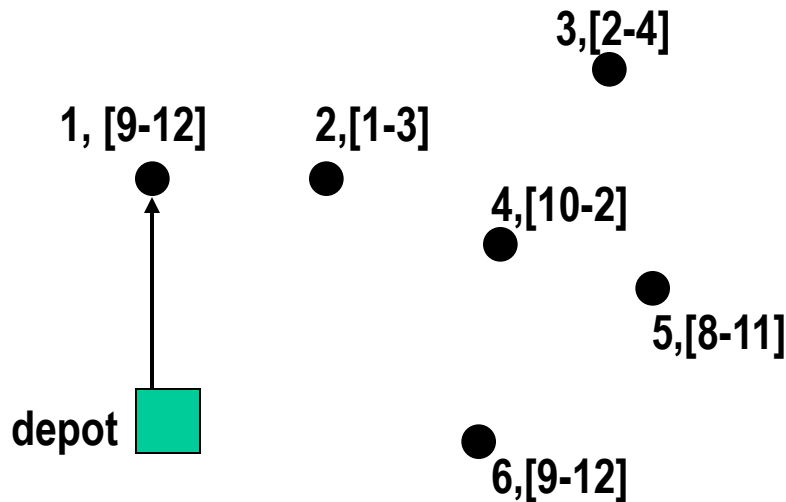
Routing Example with Time Windows

One hour travel time between any two customers.
Half hour delivery time at each customer.



Nearest Insertion with Time Windows

One hour travel time between any two customers.
Half hour delivery time at each customer.

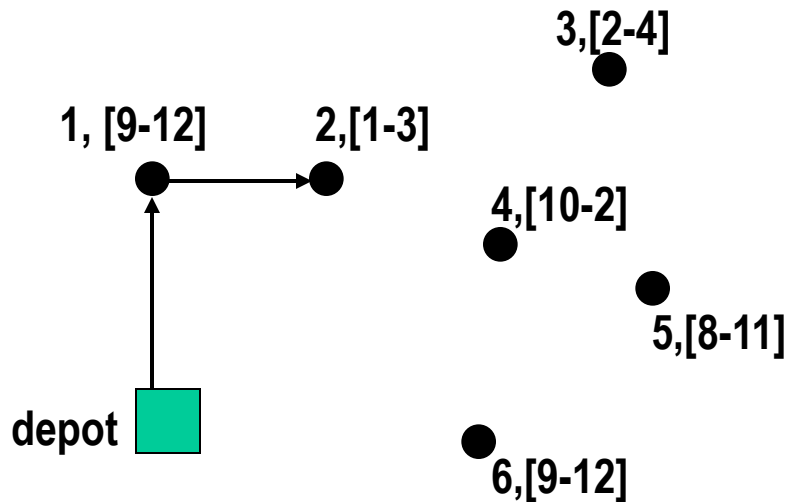


Leave depot: 8:00 - 11:00
Arrive at 1: 9:00 - 12:00
Leave 1: 9:30 - 12:30

depot-1

Nearest Insertion with Time Windows

One hour travel time between any two customers.
Half hour delivery time at each customer.

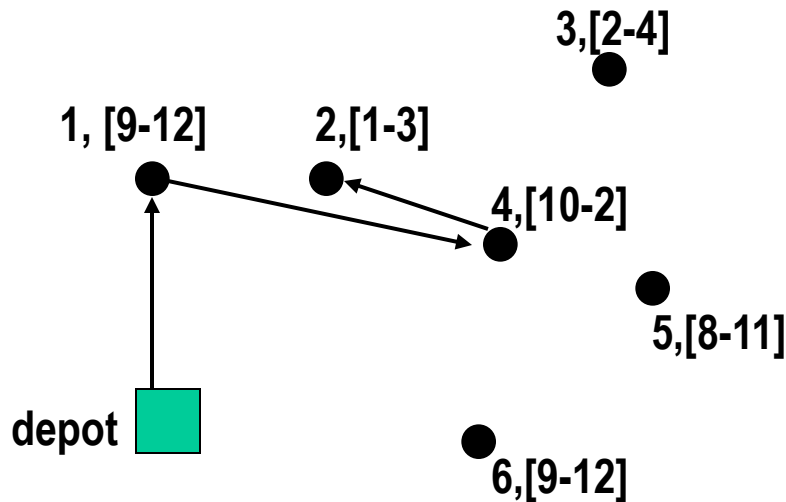


Leave depot:	10:30
Arrive at 1:	11:30
Leave 1:	12:00
Arrive at 2:	1:00
Leave 2:	1:30

depot-1-2

Nearest Insertion with Time Windows

One hour travel time between any two customers.
Half hour delivery time at each customer.

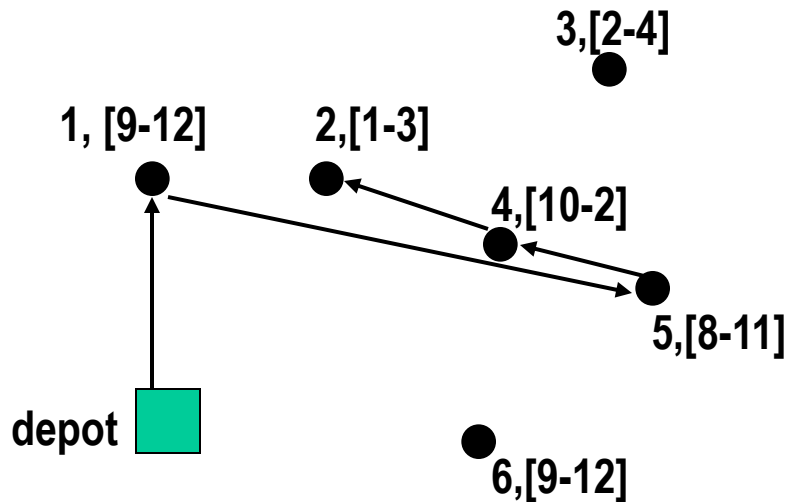


Leave depot:	9:00
Arrive at 1:	10:00
Leave 1:	10:30
Arrive at 4:	11:30
Leave 4:	12:00
Arrive at 2:	1:00
Leave 2:	1:30

depot-1-4-2

Nearest Insertion with Time Windows - one option

One hour travel time between any two customers.
Half hour delivery time at each customer.

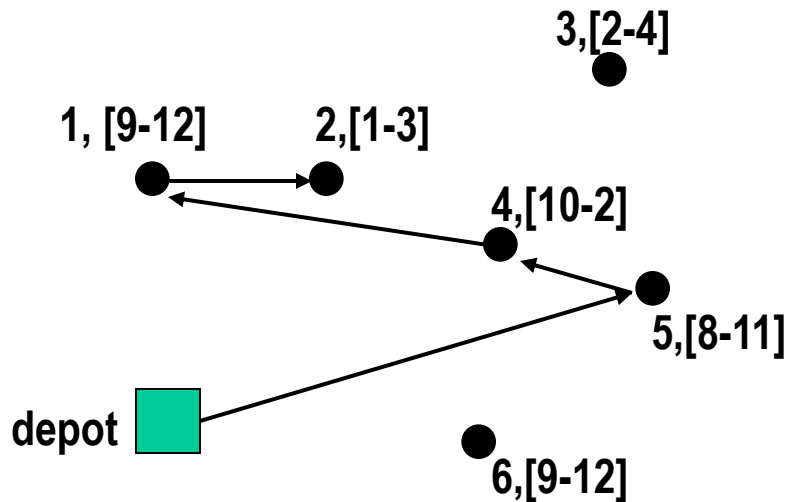


Leave depot:	8:00
Arrive at 1:	9:00
Leave 1:	9:30
Arrive at 5:	10:30
Leave 5:	11:00
Arrive at 4:	12:00
Leave 4:	12:30
Arrive at 2:	1:30
Leave 2:	2:00

depot-1-5-4-2

Nearest Insertion with Time Windows - a better option

One hour travel time between any two customers.
Half hour delivery time at each customer.

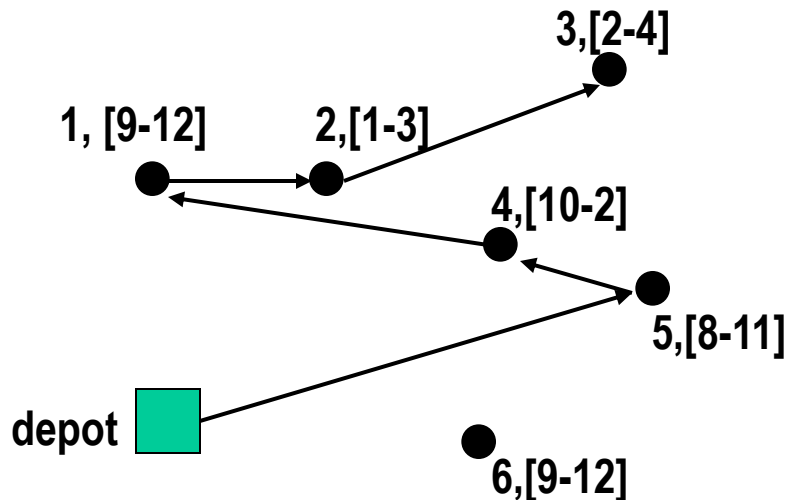


Leave depot:	7:30
Arrive at 5:	8:30
Leave 5:	9:00
Arrive at 4:	10:00
Leave 4:	10:30
Arrive at 1:	11:30
Leave 1:	12:00
Arrive at 2:	1:00
Leave 2:	1:30

depot-5-4-1-2

Nearest Insertion with Time Windows

One hour travel time between any two customers.
Half hour delivery time at each customer.

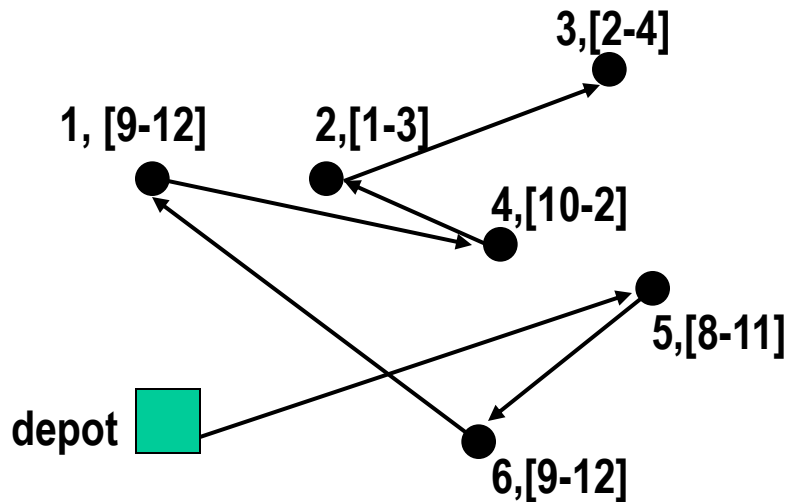


Leave depot:	7:30
Arrive at 5:	8:30
Leave 5:	9:00
Arrive at 4:	10:00
Leave 4:	10:30
Arrive at 1:	11:30
Leave 1:	12:00
Arrive at 2:	1:00
Leave 2:	1:30
Arrive at 3:	2:30
Leave 3:	3:30

depot-5-4-1-2-3

Nearest Insertion with Time Windows

One hour travel time between any two customers.
Half hour delivery time at each customer.



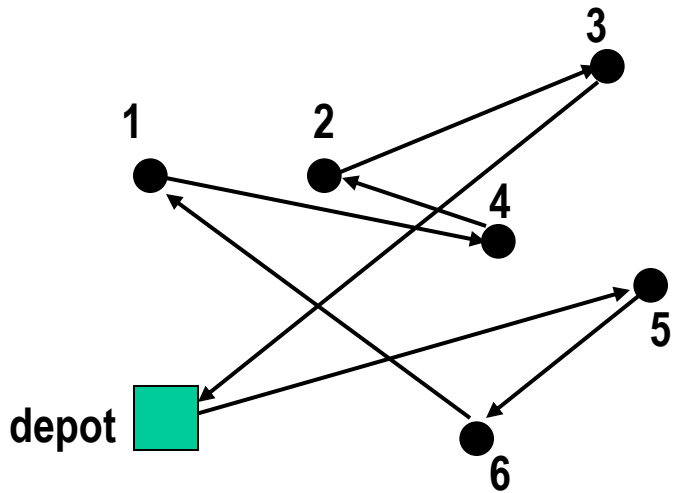
Leave depot:	7:00
Arrive at 5:	8:00
Leave 5:	8:30
Arrive at 6:	9:30
Leave 6:	10:00
Arrive at 1:	11:00
Leave 1:	11:30
Arrive at 4:	12:30
Leave 4:	1:00
Arrive at 2:	2:00
Leave 2:	2:30
Arrive at 3:	3:30

depot-5-6-1-4-2-3

Nearest Insertion

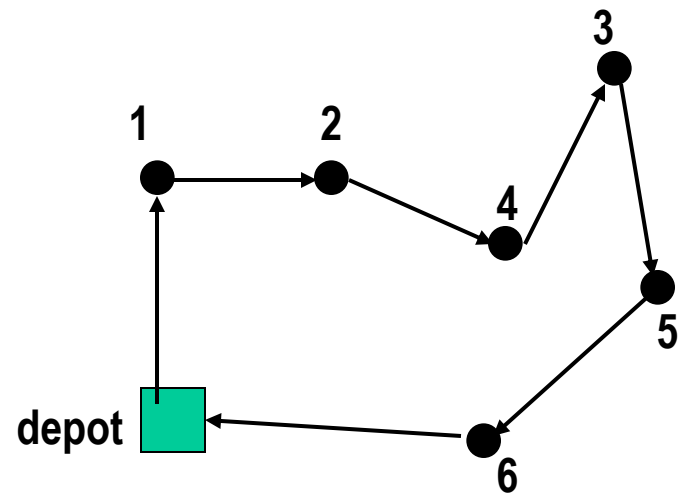
Time Windows

depot-5-6-1-4-2-3



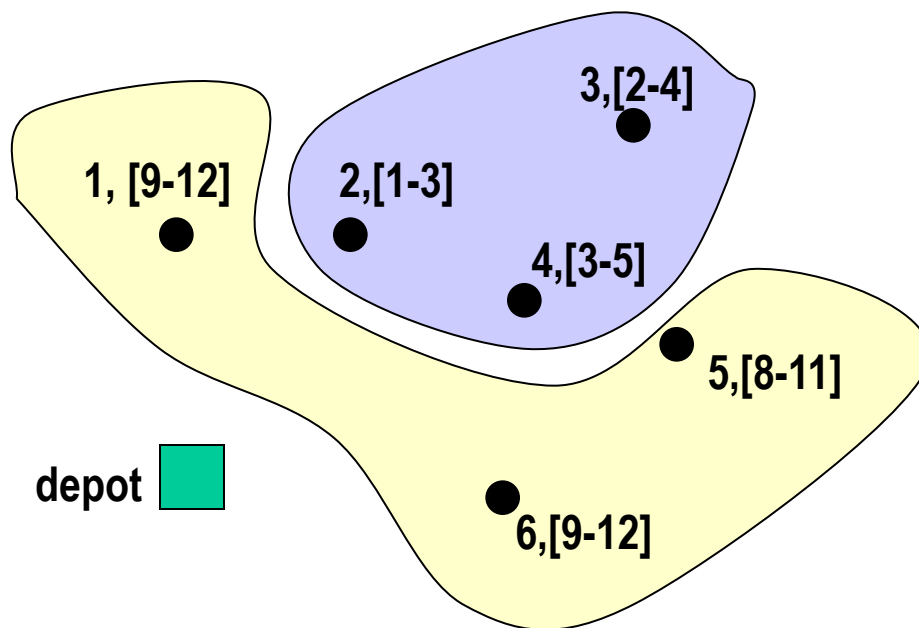
No Time Windows

depot-1-2-4-3-5-6



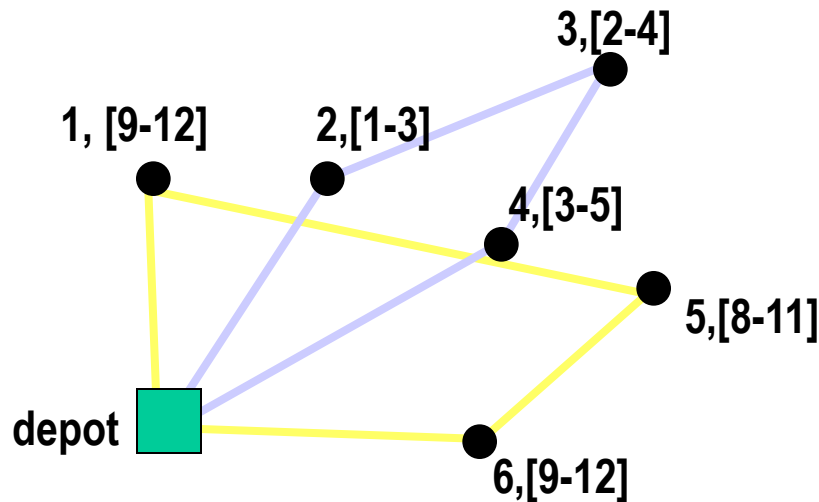
Clustering and Time Windows

- Cluster customers based on location **and time window**.
- Design routes for each cluster.



Clustering and Time Windows

- Cluster customers based on location **and time window**.
- Design routes for each cluster.



The end

Thank you