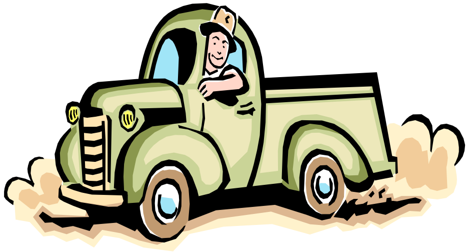
Vehicle Routing and Scheduling



Martin Savelsbergh
The Logistics Institute
Georgia Institute of Technology

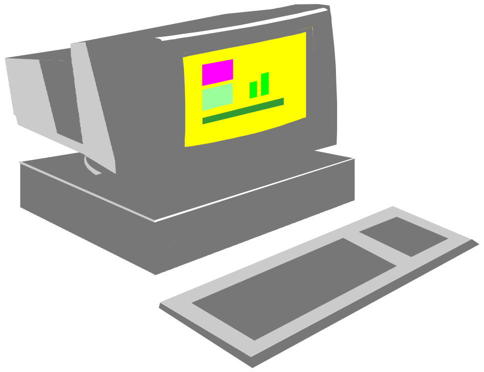
Vehicle Routing and Scheduling

Part I:

Basic Models and Algorithms

Introduction

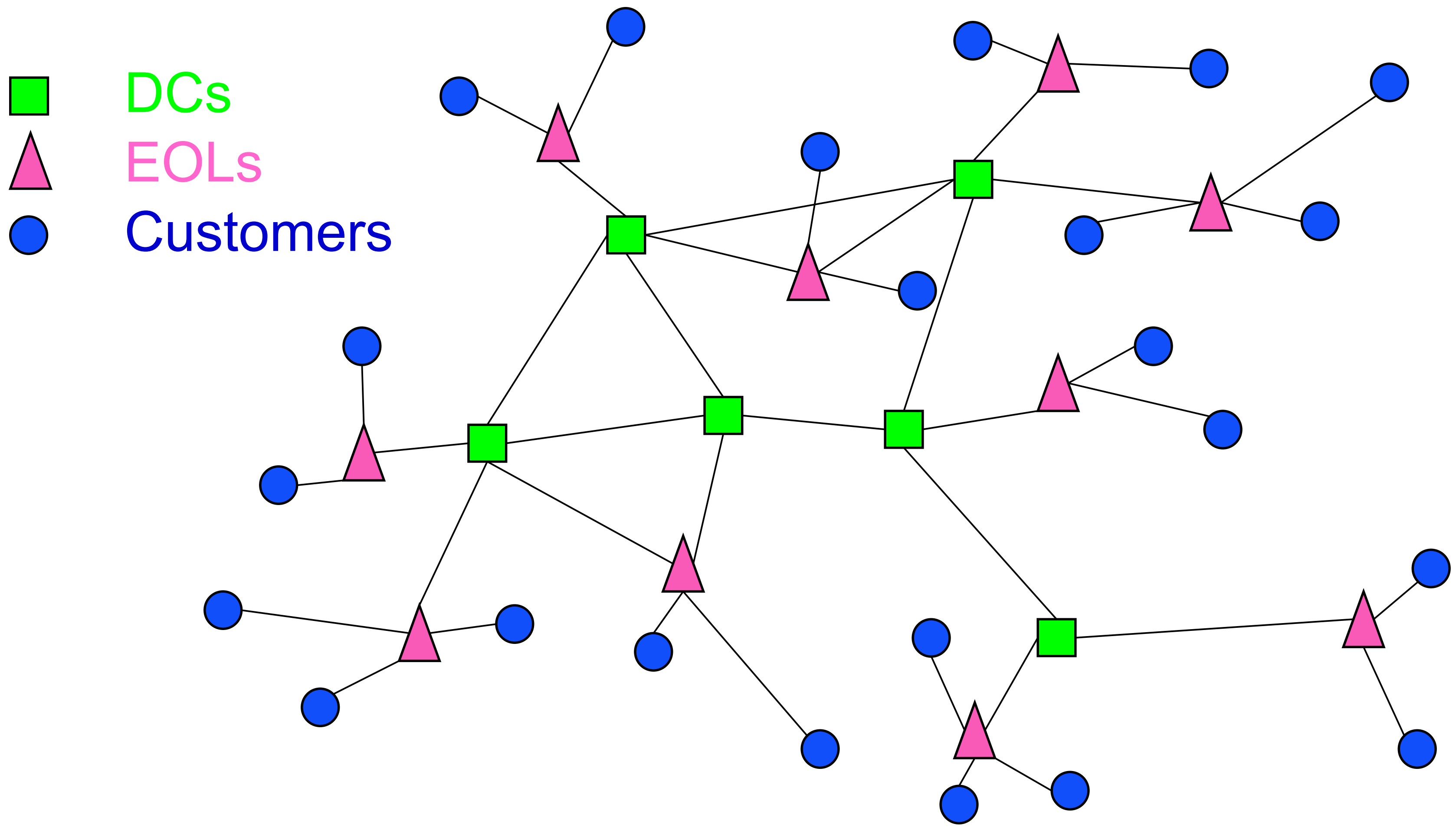
- Freight routing
  - routing of shipments
- Service routing
  - dispatching of repair technicians
- Passenger routing
  - transportation of elderly



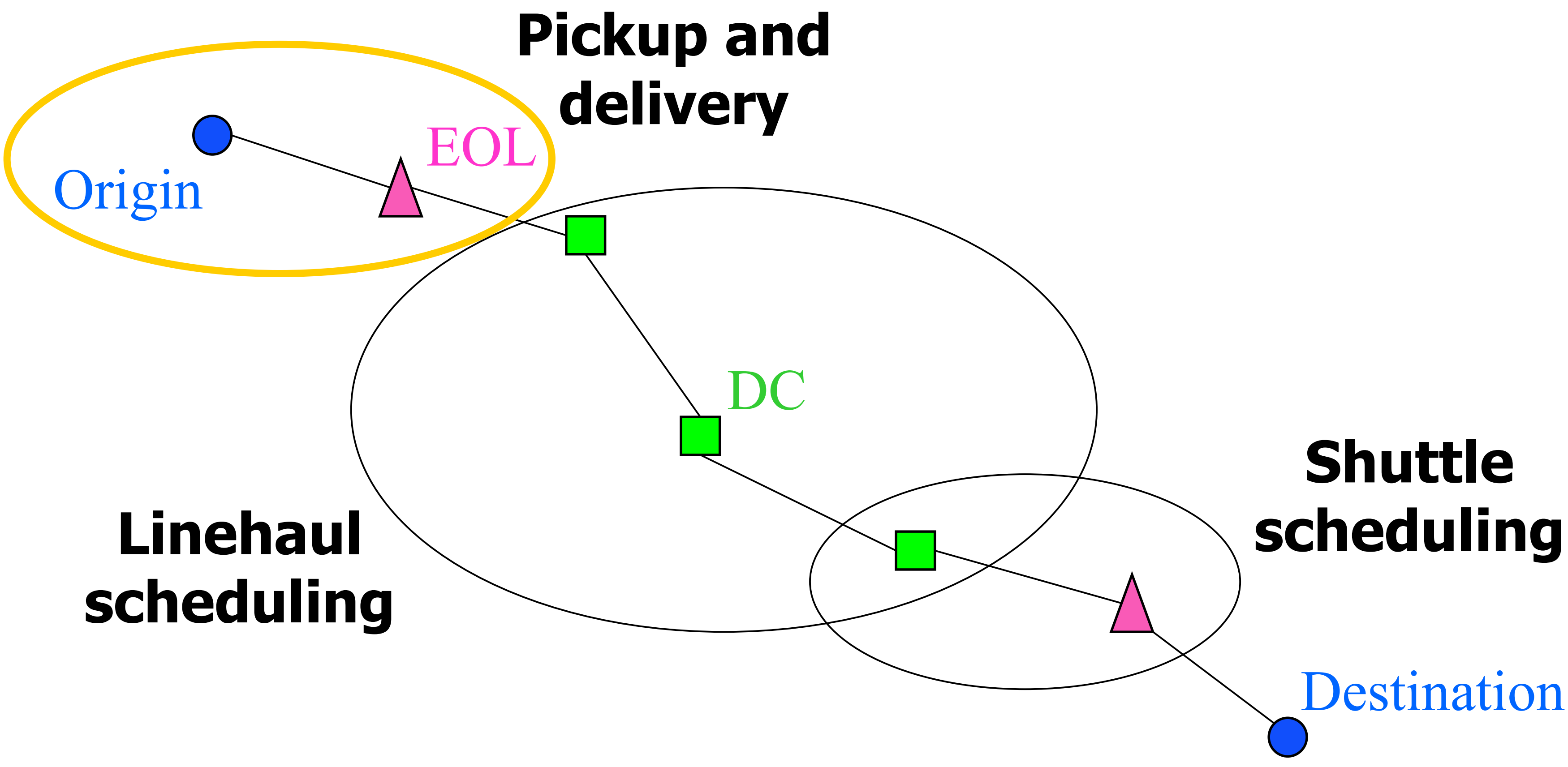
Freight Routing (LTL)

- pickup and delivery to and from end-of-line terminal
- shuttle from end-of-line terminal to regional distribution center
- transportation between distribution centers
  - rail
  - sleeper teams
  - single driver

LTL Linehaul Network



Origin-Destination Route



Routing and Scheduling

- Routing and scheduling does not follow a single "one-size-fits-all" formula. Routing and scheduling software must usually be customized to reflect the operating environment and the customer needs and characteristics

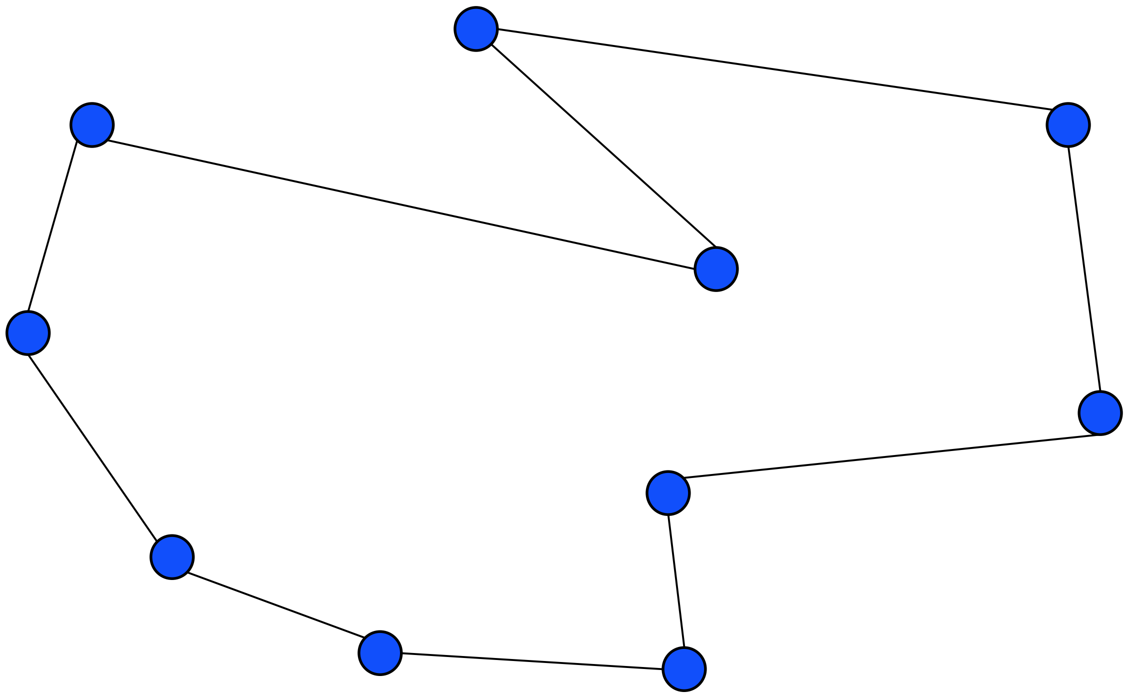
Models

- Traveling Salesman Problem (TSP)
- Vehicle Routing Problem (VRP)
- Vehicle Routing Problem with Time Windows (VRPTW)
- Pickup and Delivery Problem with Time Windows (PDPTW)

Traveling Salesman Problem

- In the TSP the objective is to find the shortest tour through a set of cities, visiting each city exactly once and returning to the starting city.
- Type of decisions:
  - routing

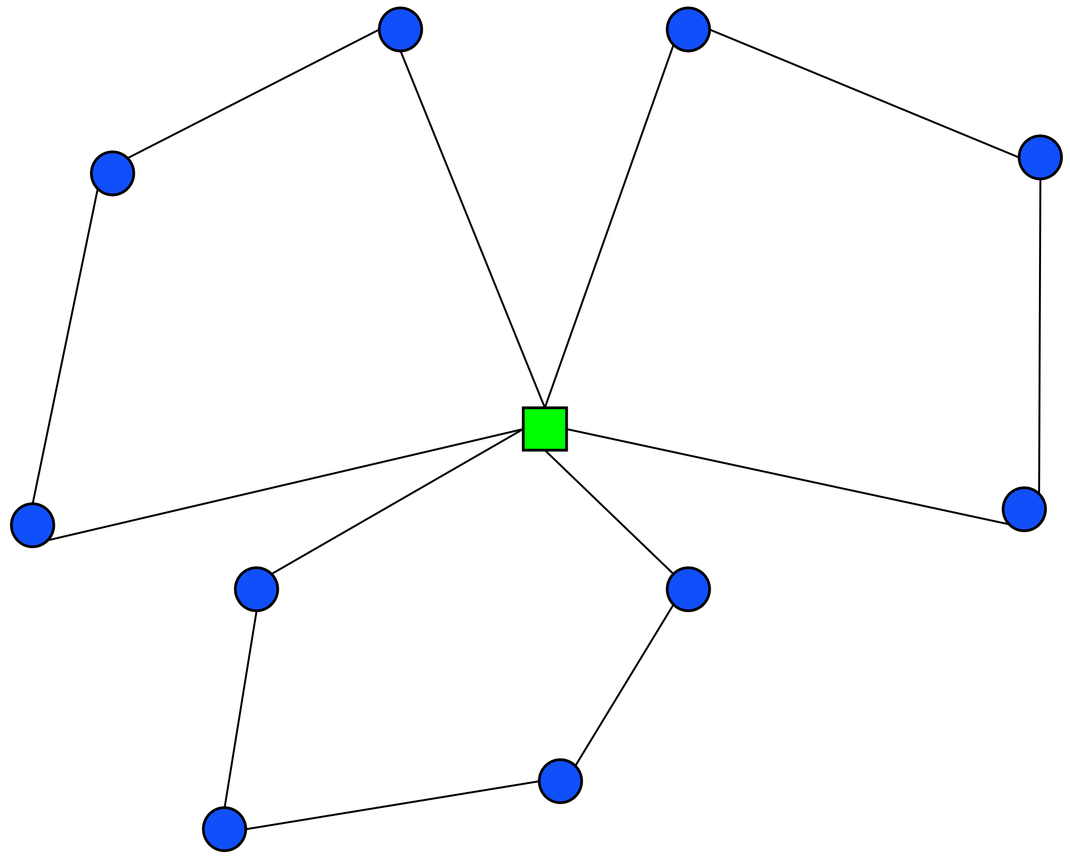
Traveling Salesman Problem



Vehicle Routing Problem

- In the VRP a number of vehicles located at a central depot has to serve a set of geographically dispersed customers. Each vehicle has a given capacity and each customer has a given demand. The objective is to minimize the total distance traveled.
- Type of decisions:
  - assigning
  - routing

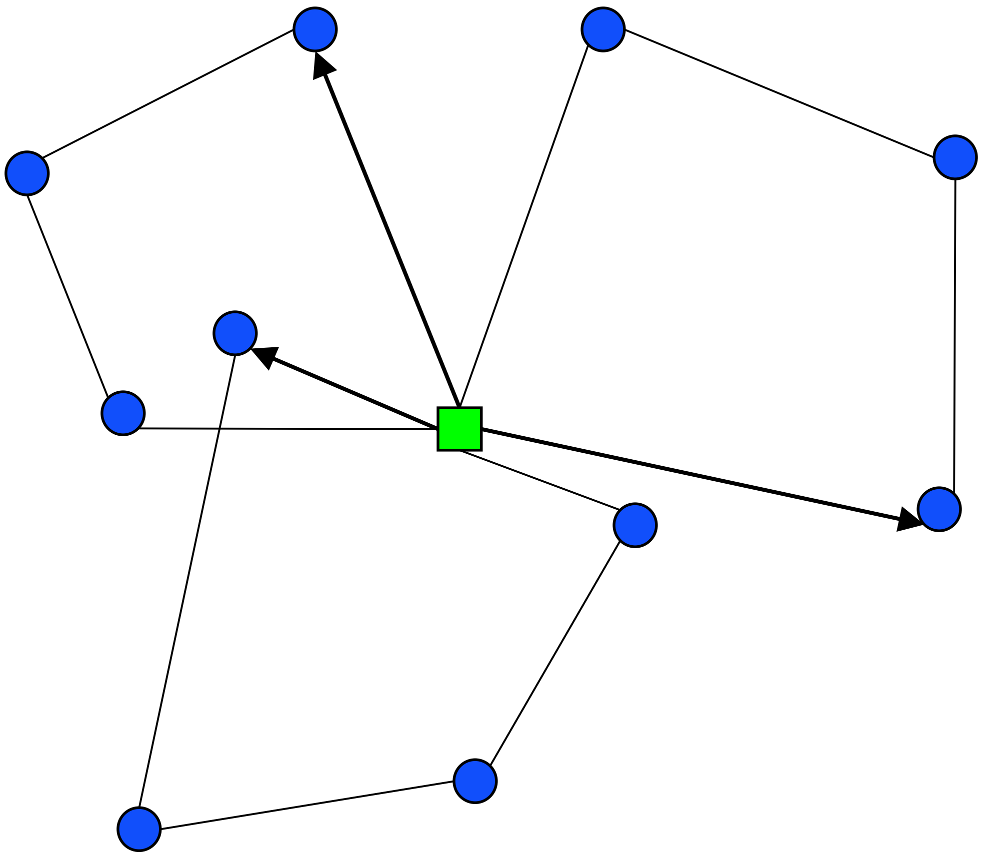
Vehicle Routing Problem



Vehicle Routing Problem with Time Windows

- In the VRPTW a number of vehicles is located at a central depot and has to serve a set of geographically dispersed customers. Each vehicle has a given capacity. Each customer has a given demand and has to be served within a given time window.
- Type of decisions:
  - assigning
  - routing
  - scheduling

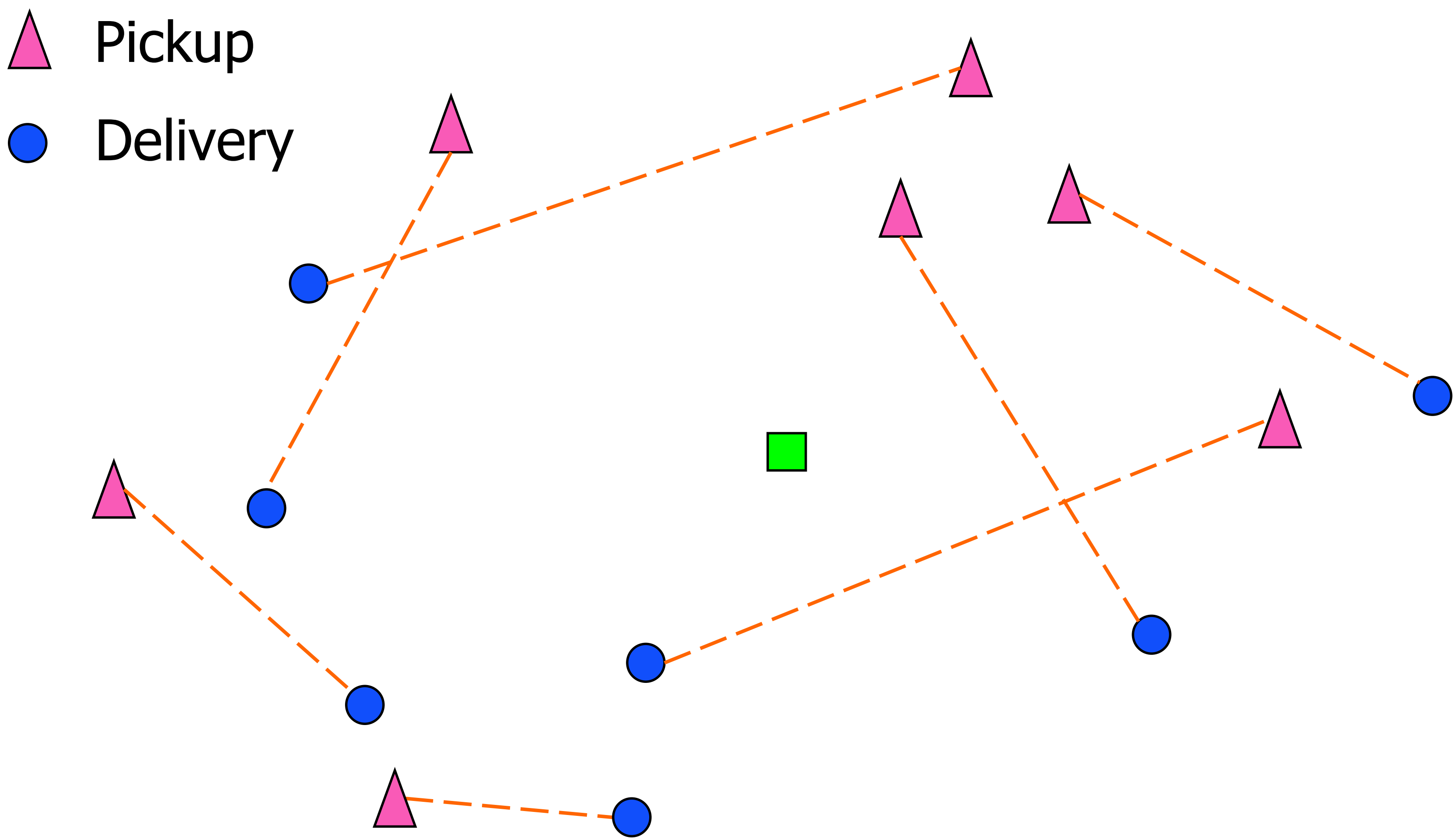
Vehicle Routing Problem with Time Windows



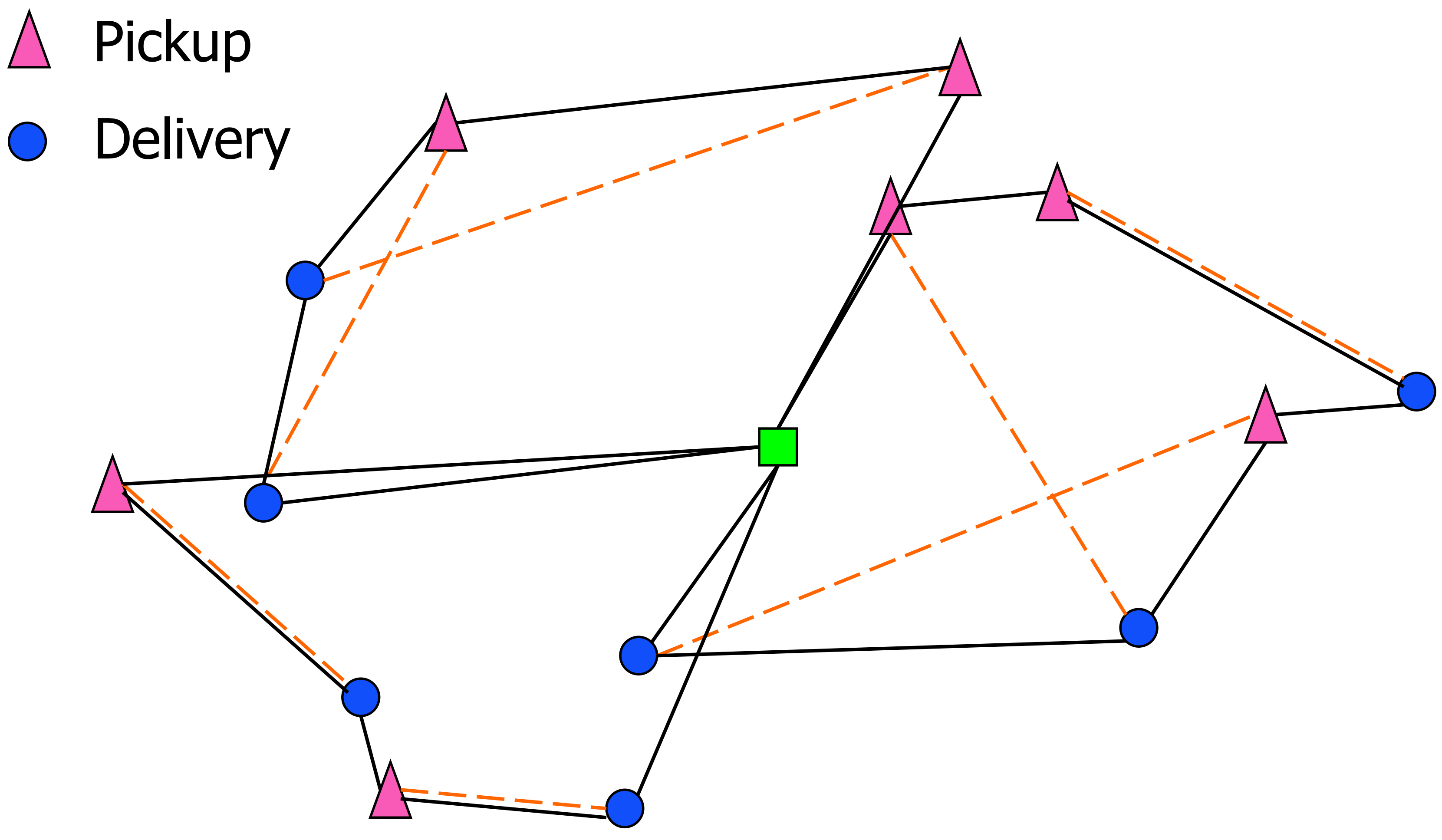
Pickup and Delivery Problem with Time Windows

- In the PDPTW a number of vehicles has to serve a number of transportation requests. Each vehicle has a given capacity. Each transportation request specifies the size of the load to be transported, the location where it is to be picked up plus a pickup time window, and the location where it is to be delivered plus a delivery time window.
- Type of decisions:
  - assigning
  - routing
  - scheduling

Pickup and Delivery Problem with Time Windows

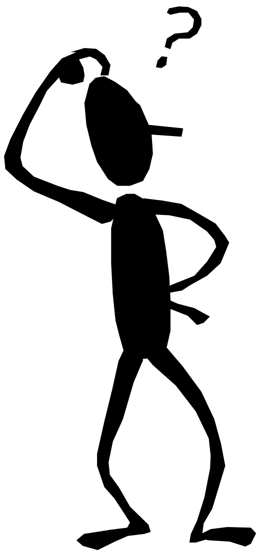


Pickup and Delivery Problem with Time Windows



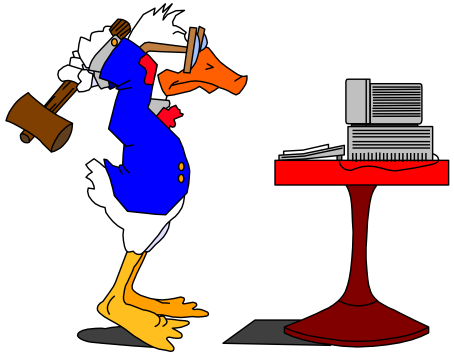
Routing and Scheduling

- Objectives
  - minimize vehicles
  - minimize miles
  - minimize labor
  - satisfy service requirements
  - maximize orders
  - maximize volume delivered per mile



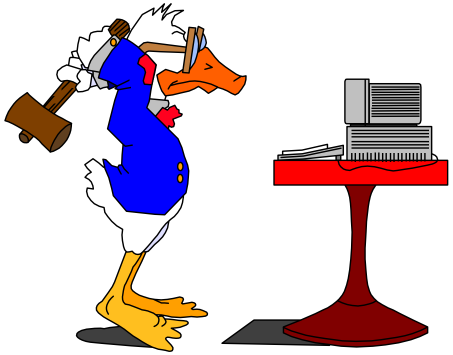
Routing and Scheduling

- Practical considerations
  - Single vs. multiple depots
  - Vehicle capacity
    - homogenous vs. hetrogenous
    - volume vs. weight
  - Driver availability
    - Fixed vs. variable start times
    - DoT regulations (10/1, 15/1, 70/8)



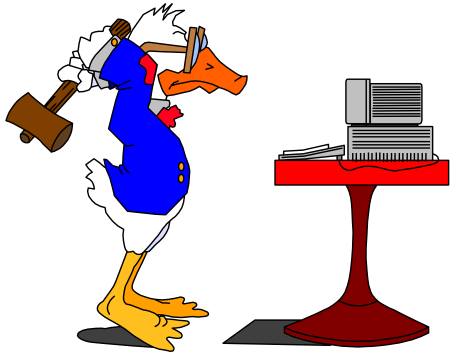
Routing and Scheduling

- Practical considerations (cont.)
  - Delivery windows
    - hard vs. soft
    - single vs. multiple
    - periodic schedules
  - Service requirements
    - Maximum ride time
    - Maximum wait time

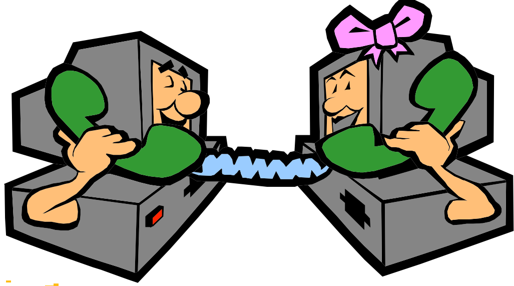


Routing and Scheduling

- Practical considerations (cont.)
  - Fixed and variable delivery times
  - Fixed vs. variable regions/route



Recent Variants



- Dynamic routing and scheduling problems
  - More and more important due to availability of GPS and wireless communication
  - Information available to design a set of routes and schedules is revealed dynamically to the decision maker
    - Order information (e.g., pickups)
    - Vehicle status information (e.g., delays)
    - Exception handling (e.g., vehicle breakdown)

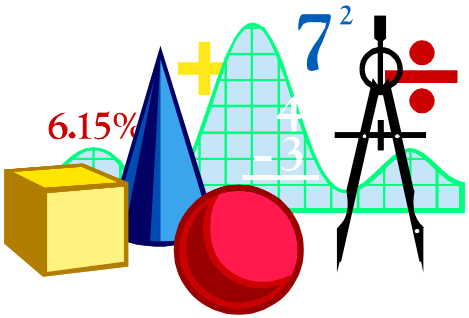
Recent Variants

- Stochastic routing and scheduling problems
  - Size of demand
  - Travel times

Algorithms

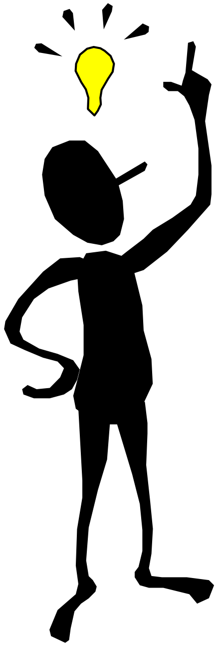
- Construction algorithms
- Improvement algorithms

- Set covering based algorithms

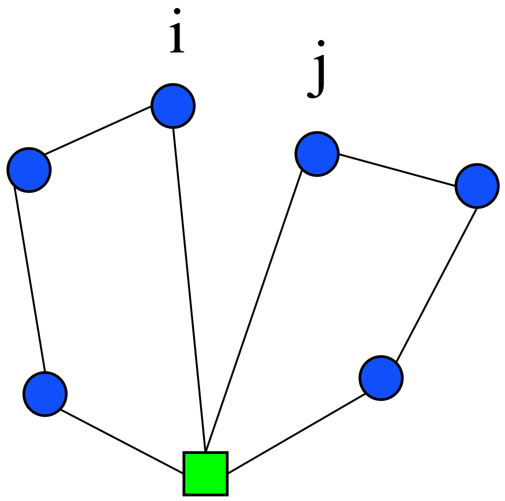


Construction Heuristics

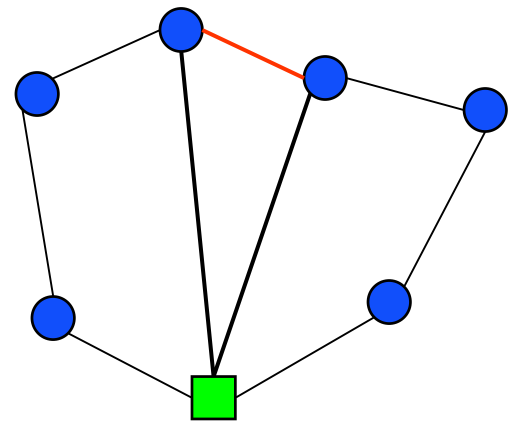
- Savings heuristic
- Insertion heuristics



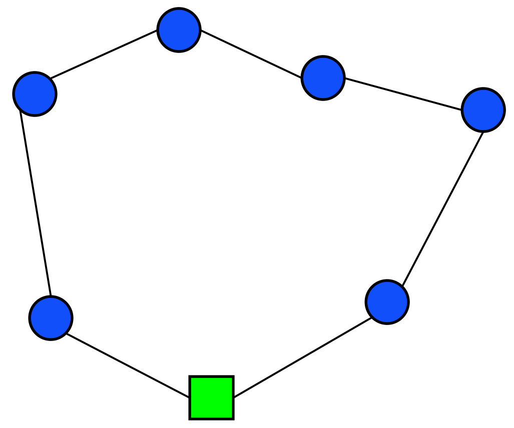
Savings



Initial



Intermediate



Final

Savings $s(i,j) = c(i,0) + c(0,j) - c(i,j)$

Savings Heuristic (Parallel)

- Step 1. Compute savings s(i,j) for all pairs of customers. ***Sort savings.*** Create out-and-back routes for all customers.
- Step 2. Starting from the top of the savings list, determine whether there exist two routes, one containing (i,0) and the other containing (0,j). If so, merge the routes if the combined demand is less than the vehicle capacity.

Savings Heuristics (Sequential)

- Step 1. Compute savings s(i,j) for all pairs of customers. Sort savings. Create out-and-back routes for all customers.
- Step 2. Consider route (0,i,…,j,0) and determine the best savings s(k,i) and s(j,l) with routes containing (k,0) and (0,l). Implement the best of the two. If no more savings exist for this route move to the next.

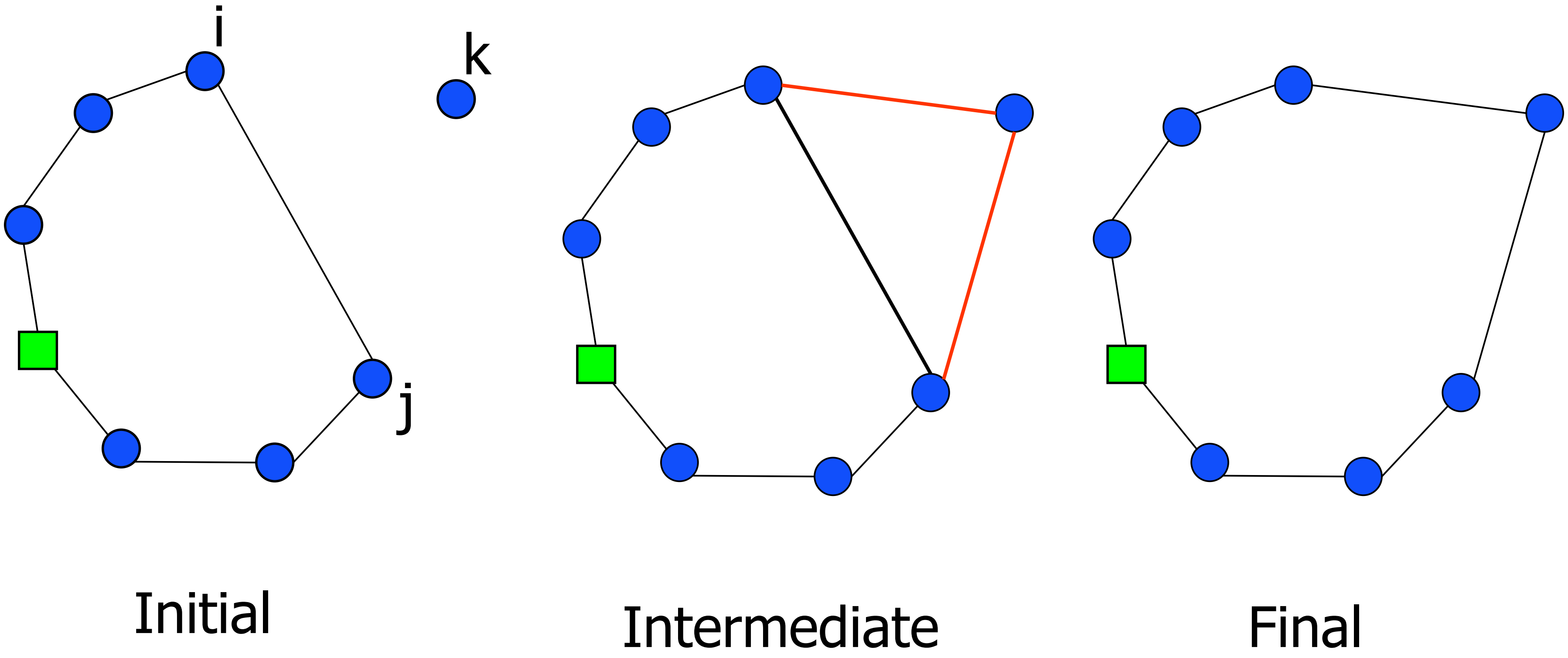
Savings Heuristic - Enhancement

- Route shape parameter

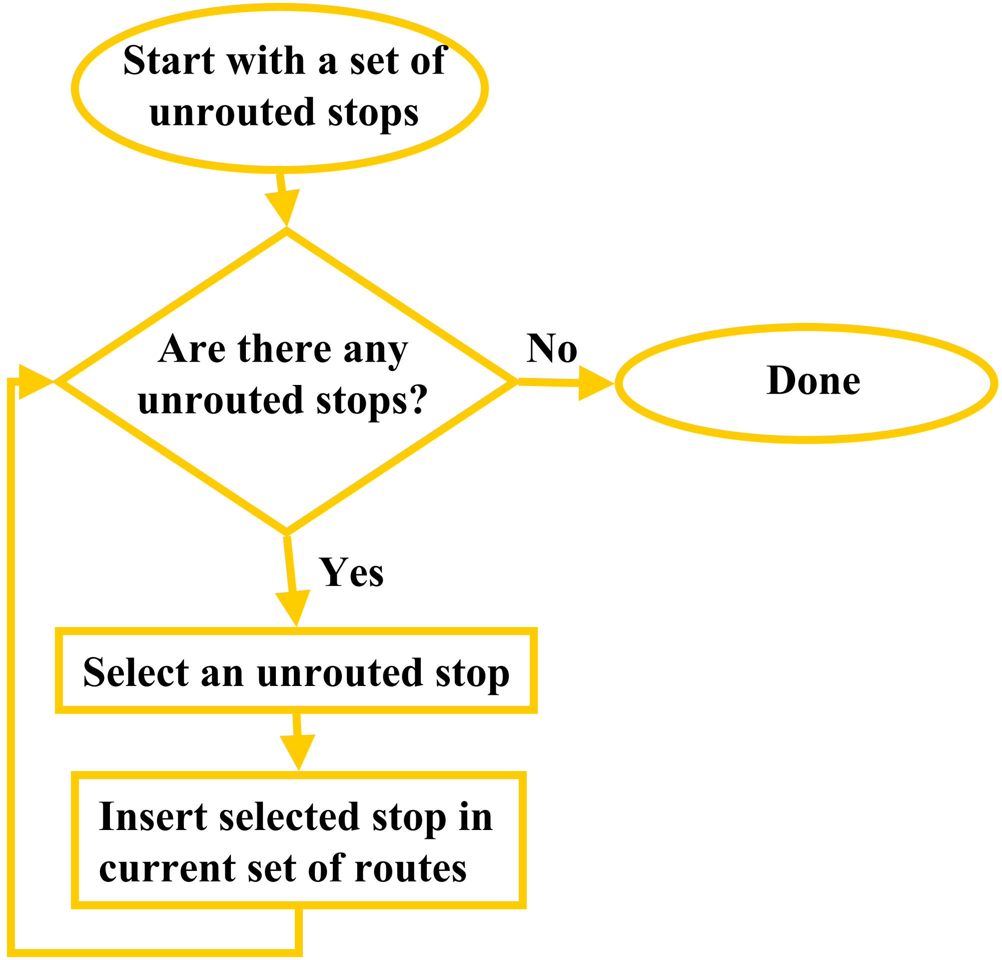
  $$s(i,j) = c(i,0) + c(0,j) - \lambda\, c(i,j)$$

- The larger λ, the more emphasis is placed on the distance between customers being connected

Insertion



Insertion Heuristics



Nearest addition

- Selection:
  - If partial tour T does not include all cities, find cities k and j, j on the tour and k not, for which c(j,k) is minimized.
- Insertion:
  - Let {i,j} be either one of the two edges involving j in T, and replace it by {i,k} and {k,j} to obtain a new tour including k.

Nearest Insertion

- Selection:
  - If partial tour T does not include all cities, find cities k and j, j on the tour and k not, for which c(j,k) is minimized.
- Insertion:
  - Let {i,j} be the edge of T which minimizes c(i,k) + c(k,j) - c(i,j), and replace it by {i,k} and {k,j} to obtain a new tour including k.

Farthest Insertion

- Selection:
  - If partial tour T does not include all cities, find cities k and j, j on the tour and k not, for which c(j,k) is **maximized**.
- Insertion:
  - Let {i,j} be the edge of T which minimizes c(i,k) + c(k,j) - c(i,j), and replace it by {i,k} and {k,j} to obtain a new tour including k.

Cheapest Insertion

- Selection:
  - If partial tour T does not include all cities, find for each k not on T the edge {i,j} of T which minimizes c(T,k) = c(i,k) + c(k,j) - c(i,j). Select city k for which c(T,k) is minimized.
- Insertion:
  - Let {i,j} be the edge of T for which c(T,k) is minimized, and replace it by {i,k} and {k,j} to obtain a new tour including k.

Worst case results

- Nearest addition: 2
- Nearest insertion: 2
- Cheapest insertion: 2

- Farthest insertion:
  - > 2.43 (Euclidean)
  - > 6.5 (Triangle inequality)

Implementation

- Priority Queue
  - insert(value, key)
  - getTop(value, key)
  - setTop(value, key)
- k-d Tree
  - deletePt(point)
  - nearest(point)

Implementation

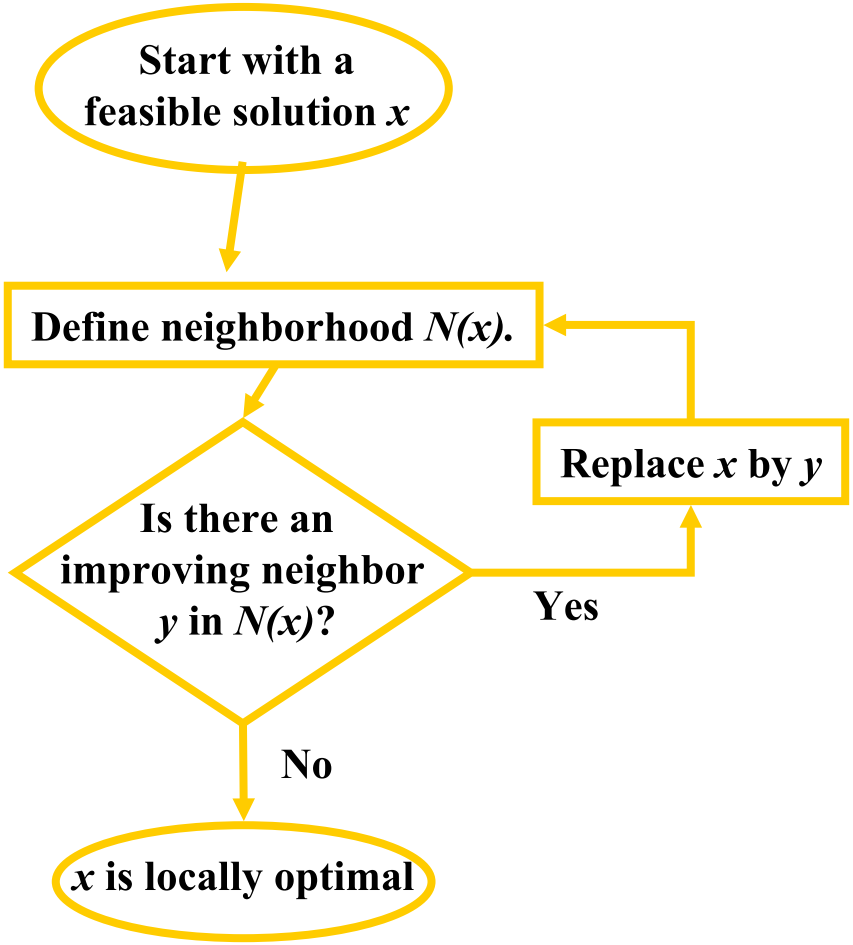
```
Tree->deletePt(StartPt)
NNOut[StartPt] := Tree->nearest(StartPt)
PQ->insert(Dist(StartPt, NNOut(StartPt)), StartPt)
loop n-1 time
    loop
        PQ->getTop(ThisDist, x)
        y := NNOut[x]
        If y not in tour, then break
        NNOut[x] = Tree->nearest(x)
        PQ->setTop(Dist(x, NNOut[x]), x)
    Add point y to tour; x is nearest neighbor in tour
    Tree->deletePt(y)
    NNOut[y] = Tree->nearest(y)
    PQ->insert(Dist(y, NNOut[y]), y)
```

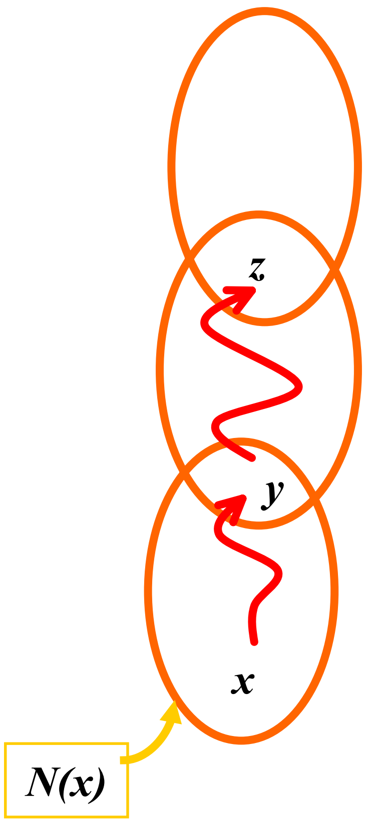
Find nearest point

Delayed update

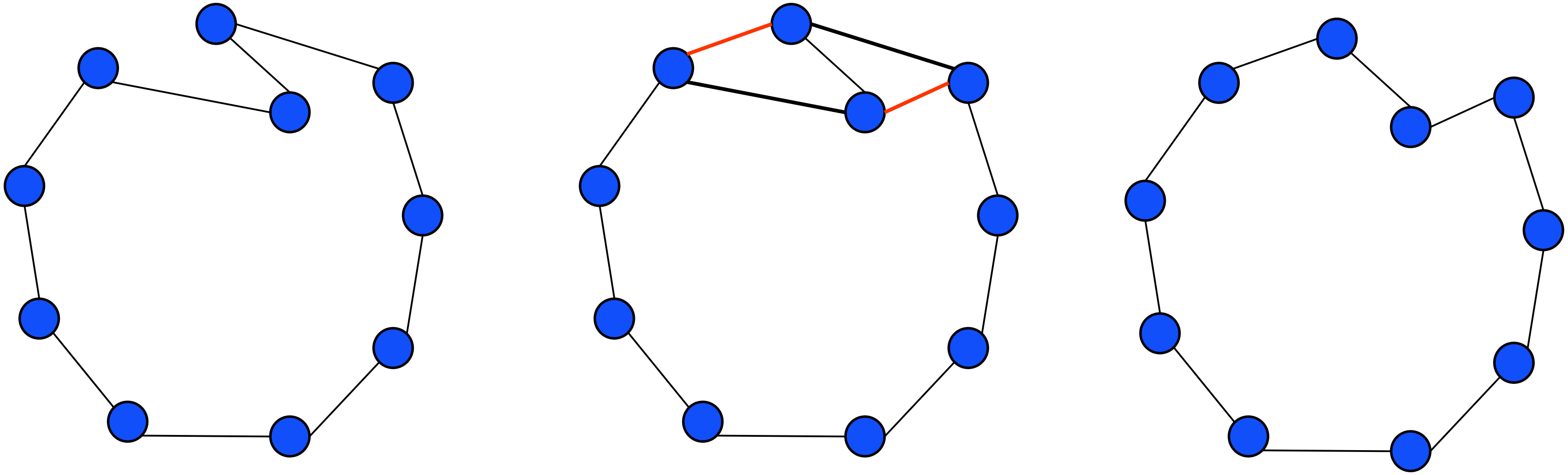
Update

Improvement Algorithms



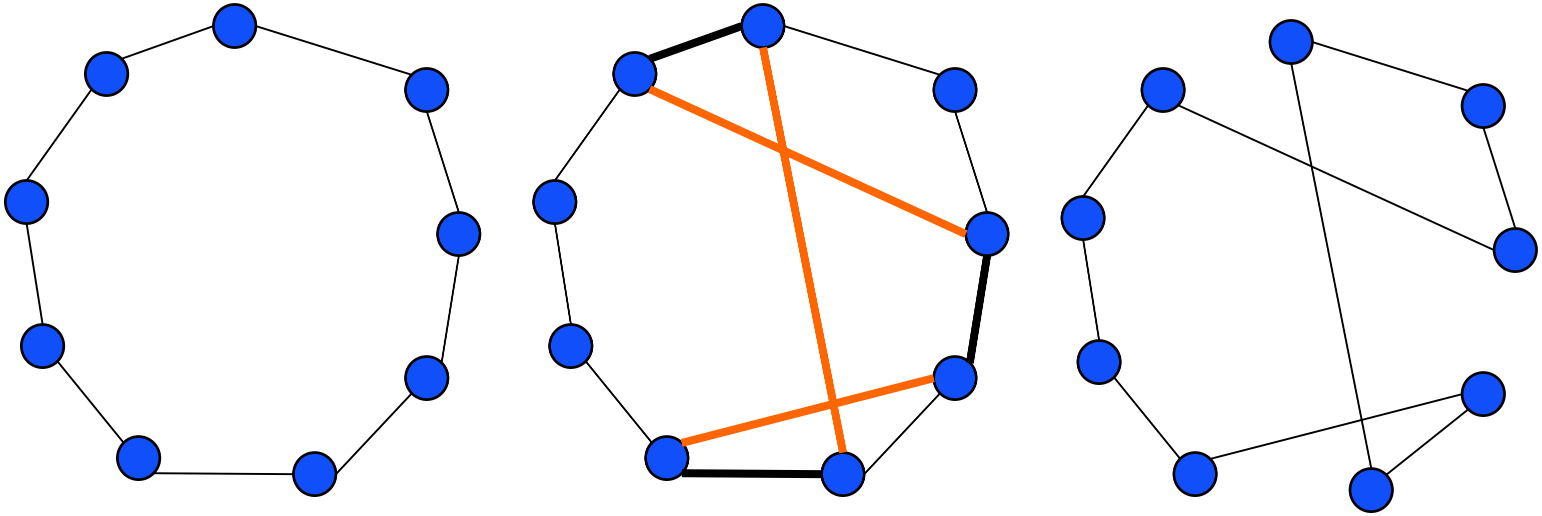


2-change



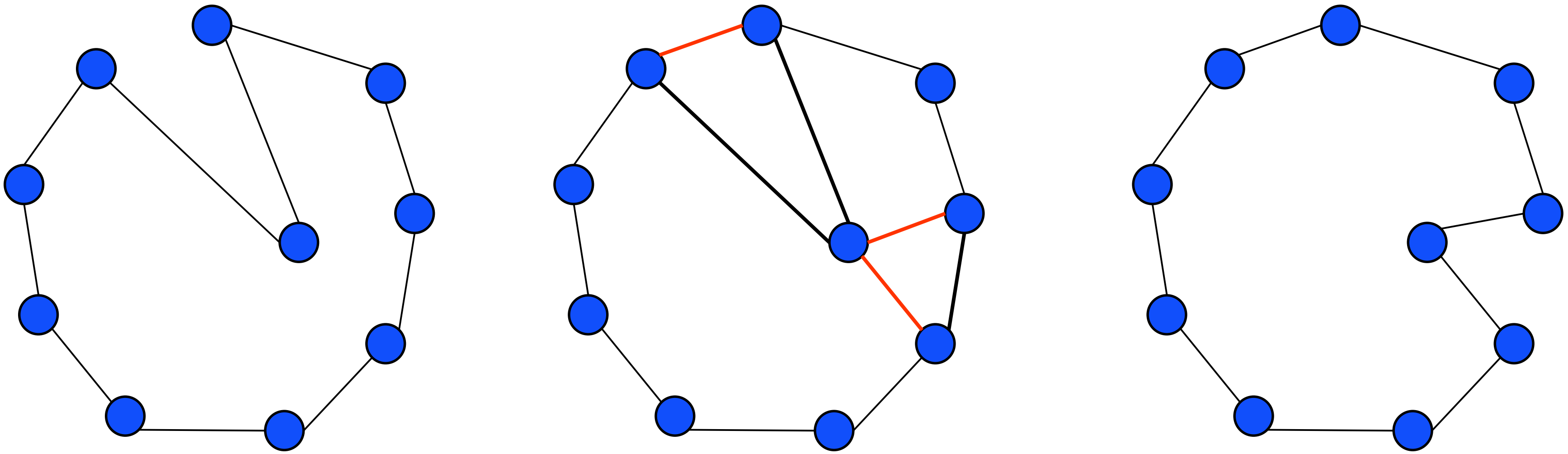
$O(n^2)$ possibilities

3-change



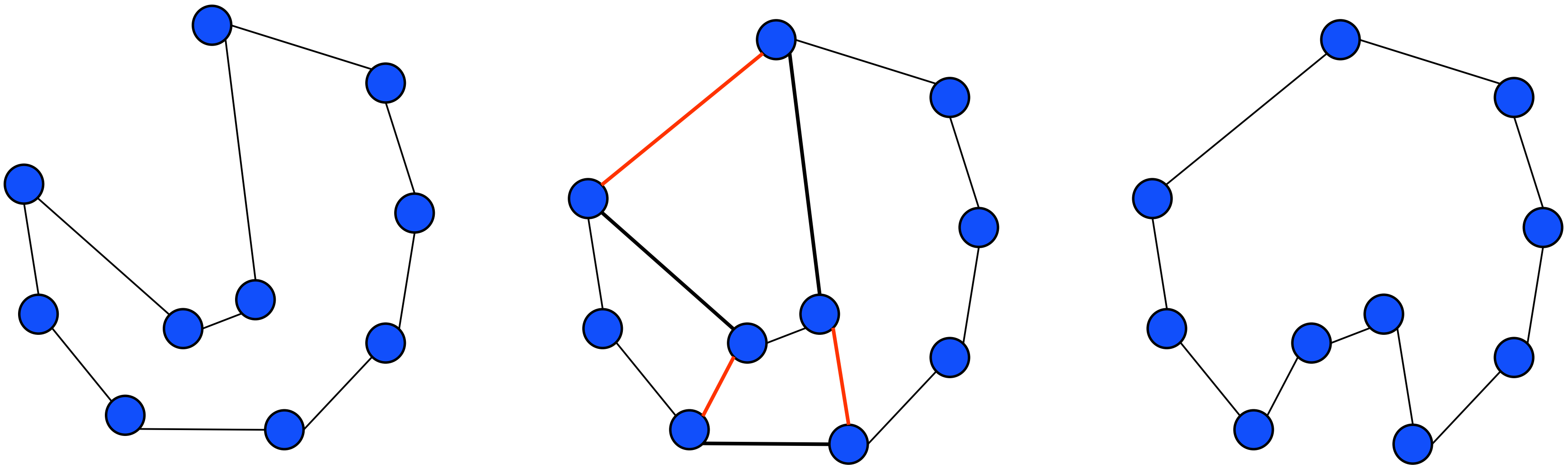
$O(n^3)$ possibilities

1-Relocate



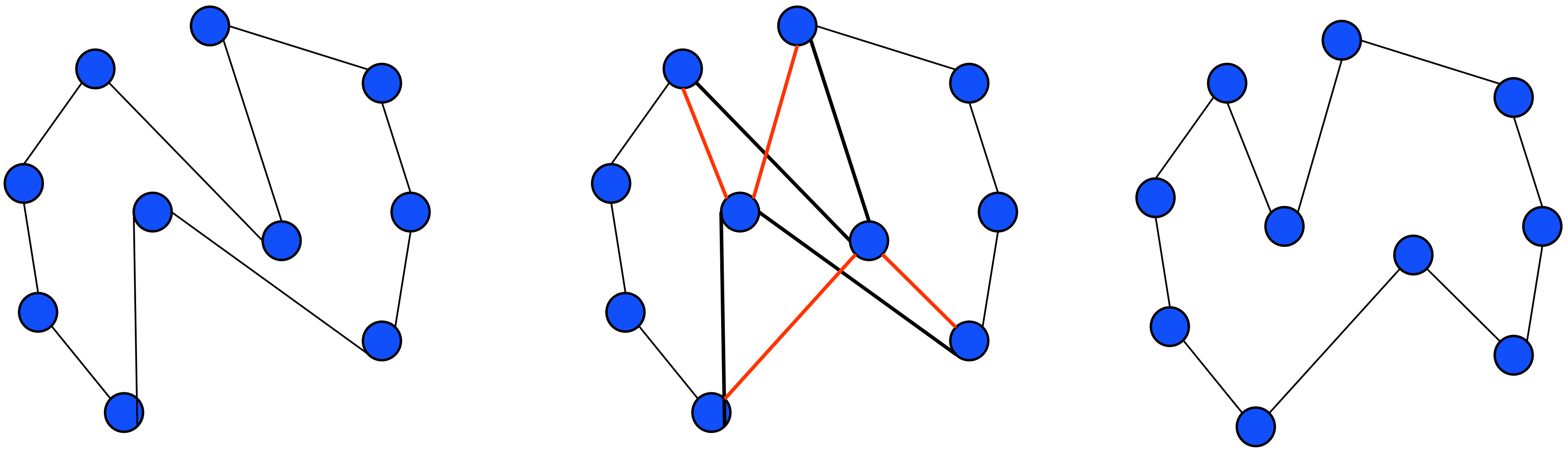
O(n^2) possibilities

2-Relocate



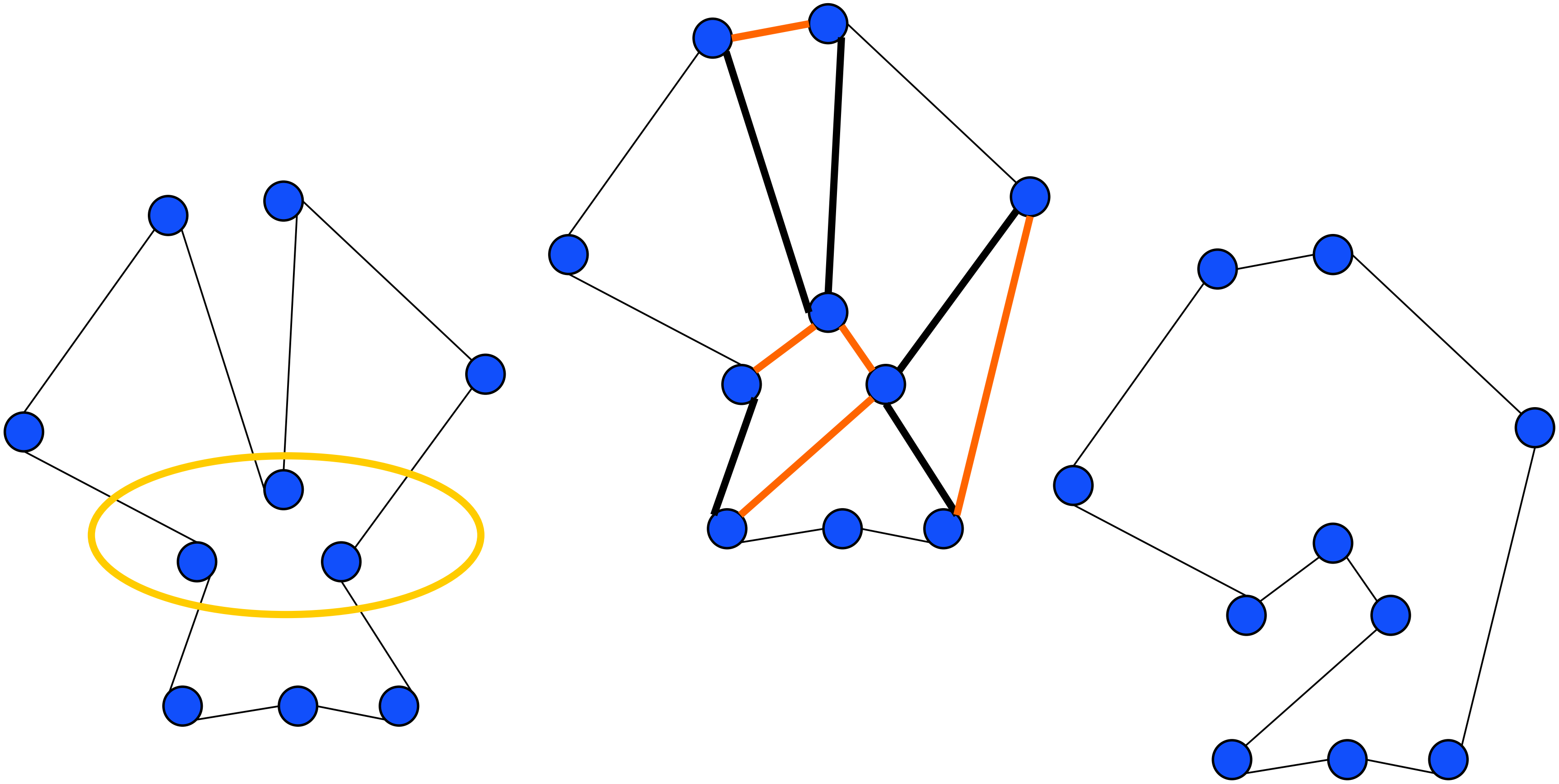
$O(n^2)$ possibilities

Swap

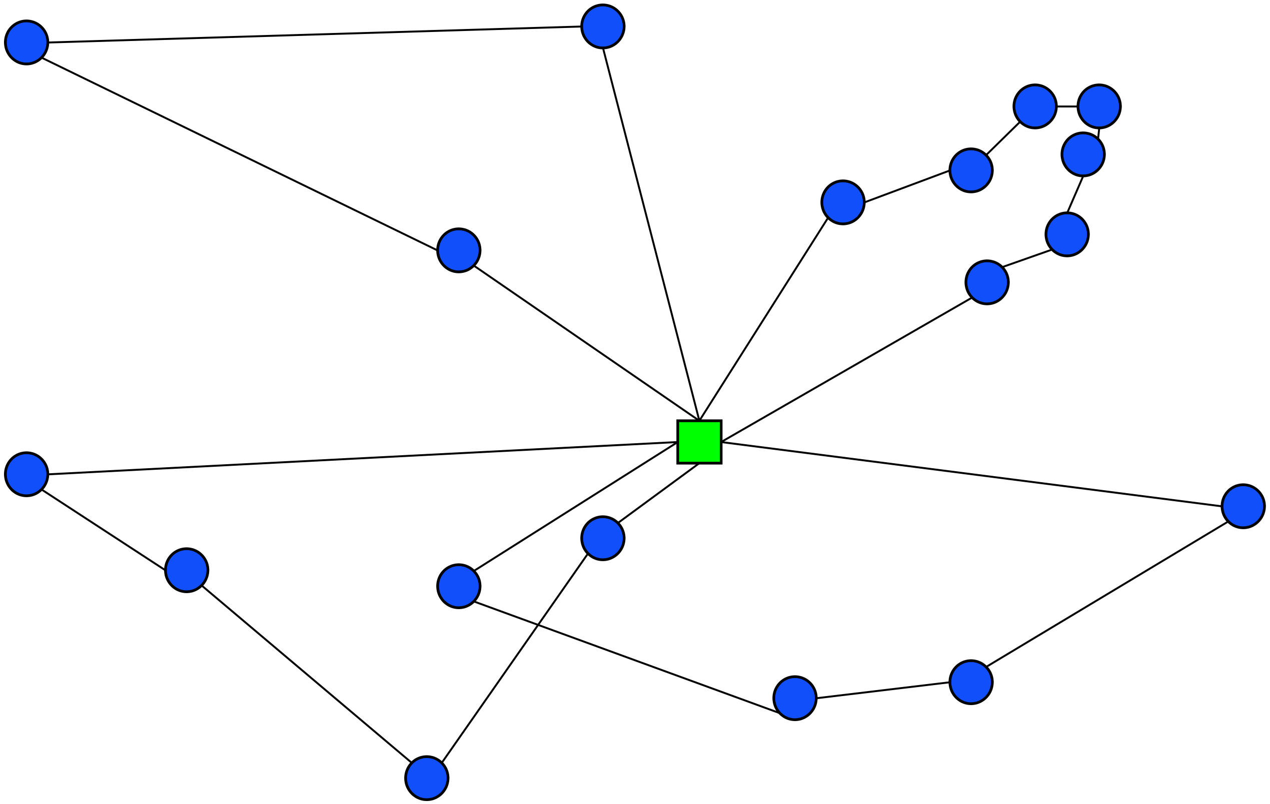


O(n^2) possibilities

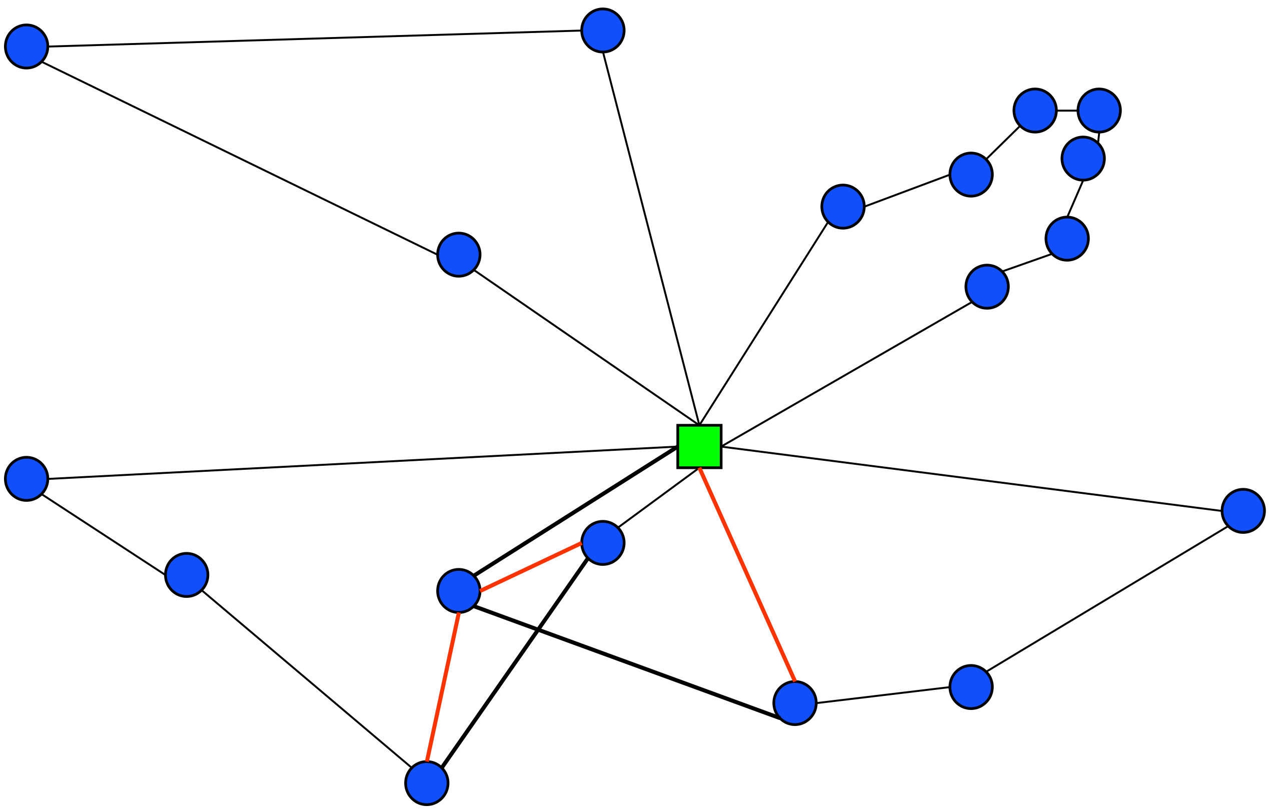
GENI



Vehicle Routing and Scheduling

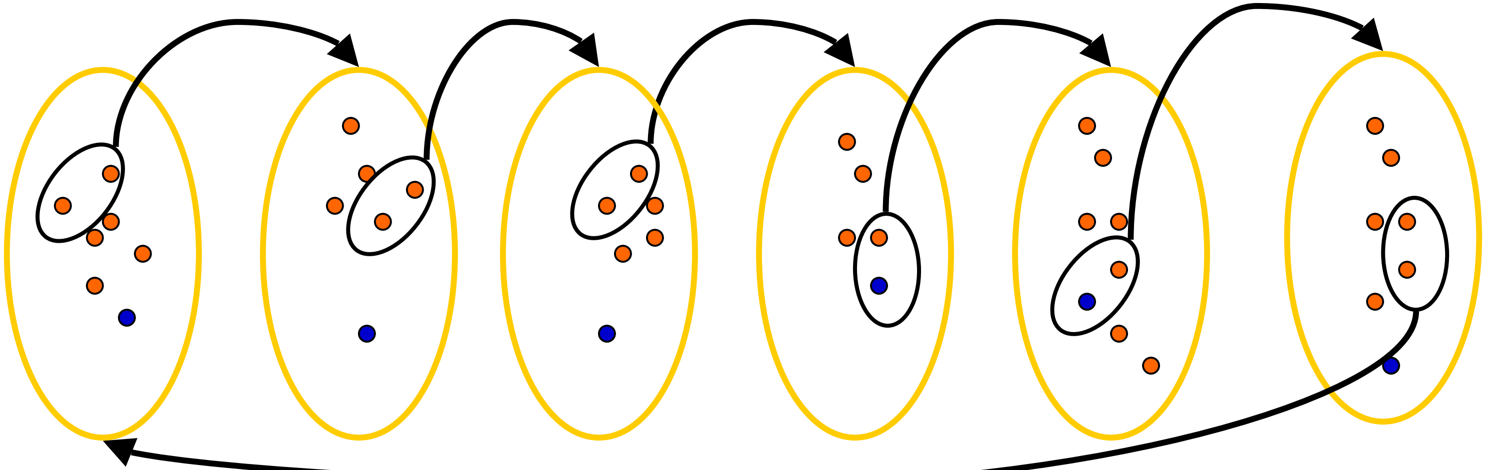


Vehicle Routing and Scheduling



1-relocate

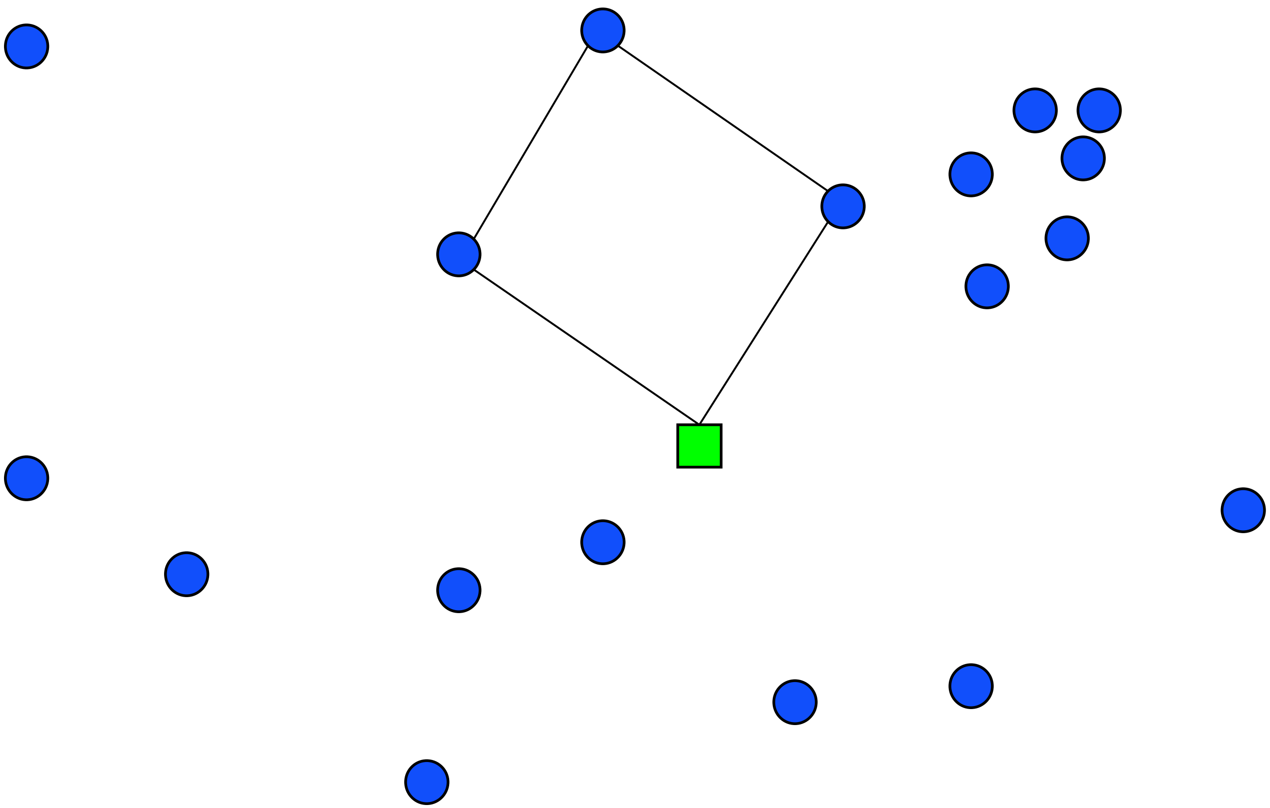
B-cyclic k-transfer



Set covering based algorithms

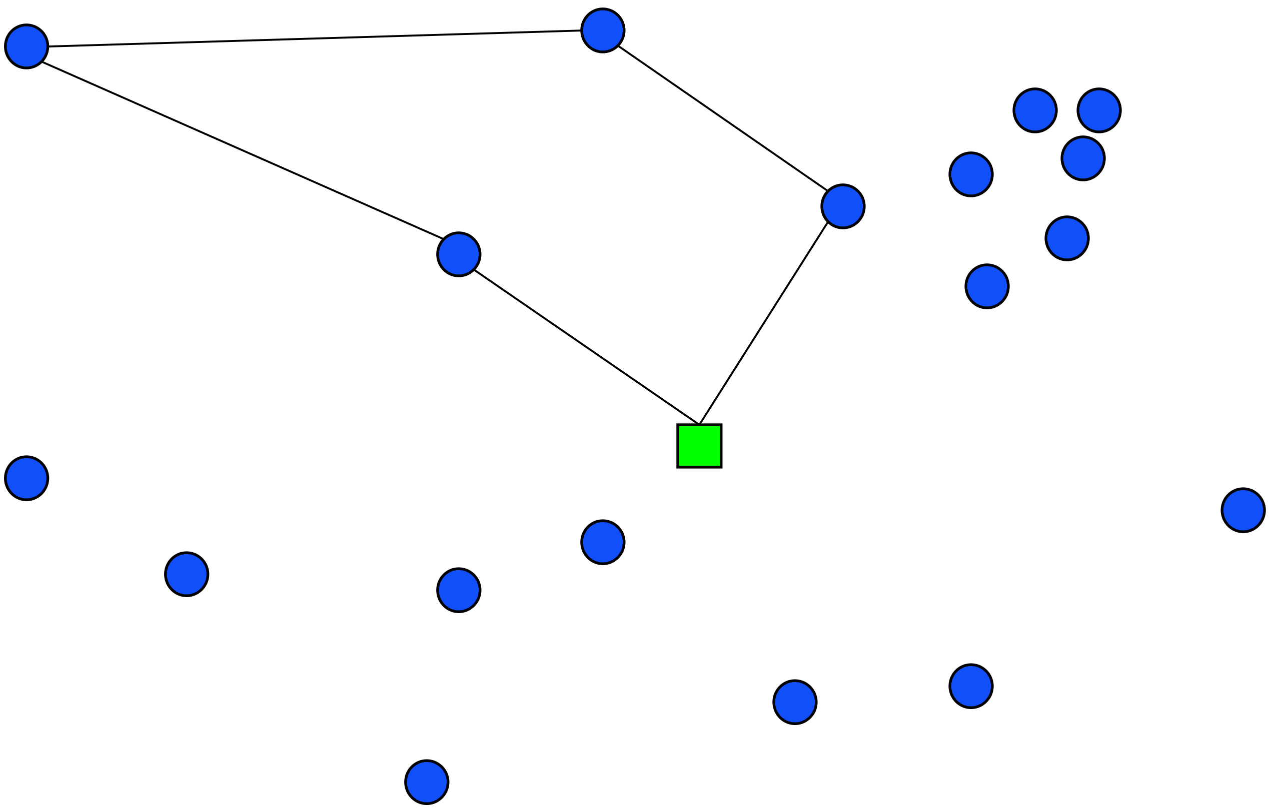
- Assignment decisions are often the most important
- Assignment decisions are often the most difficult

Vehicle routing and scheduling



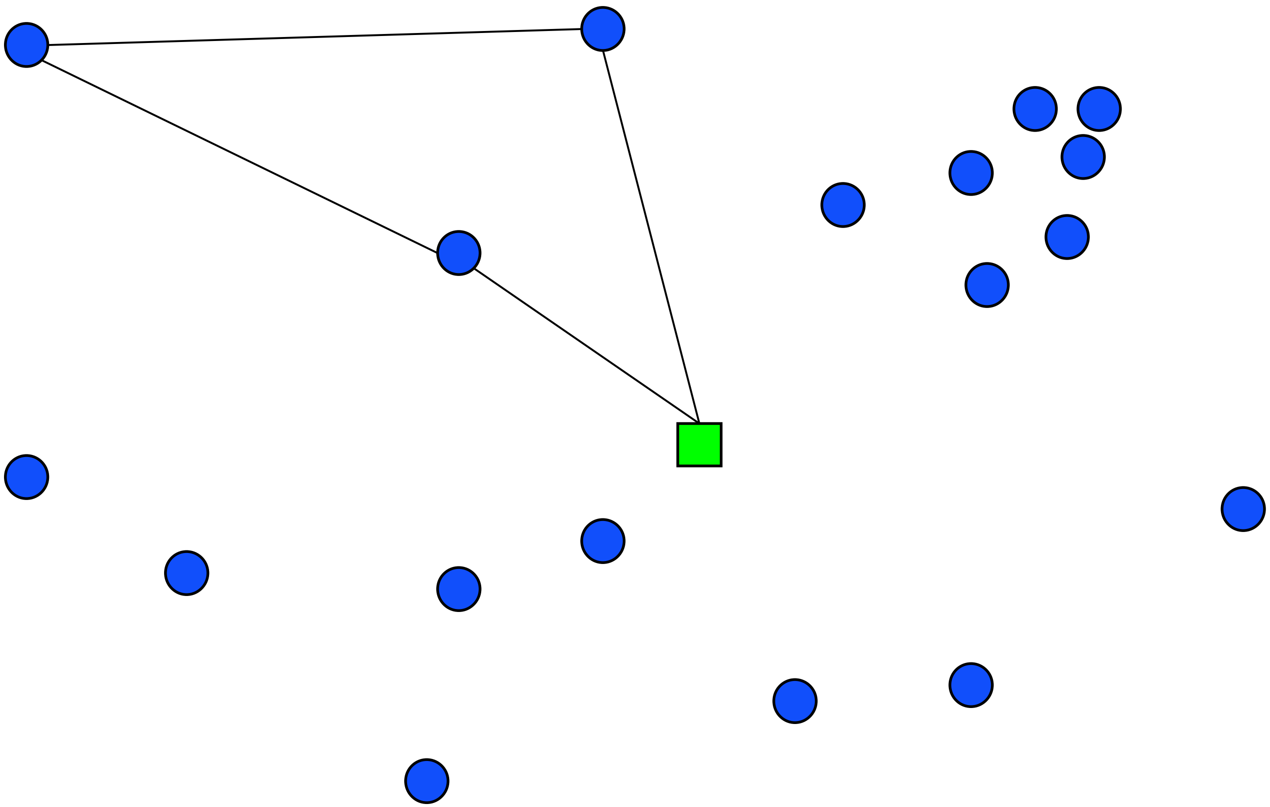
Possible route

Vehicle routing and scheduling



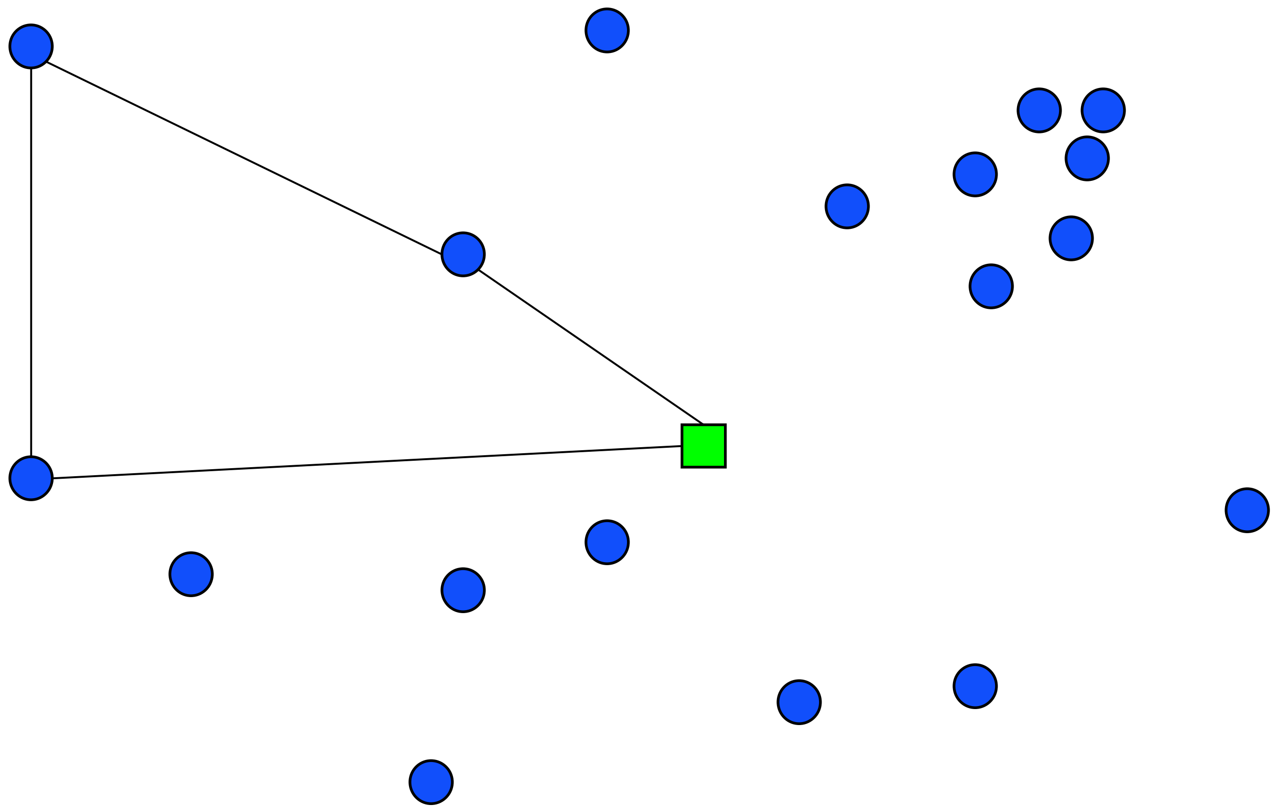
Possible route

Vehicle routing and scheduling



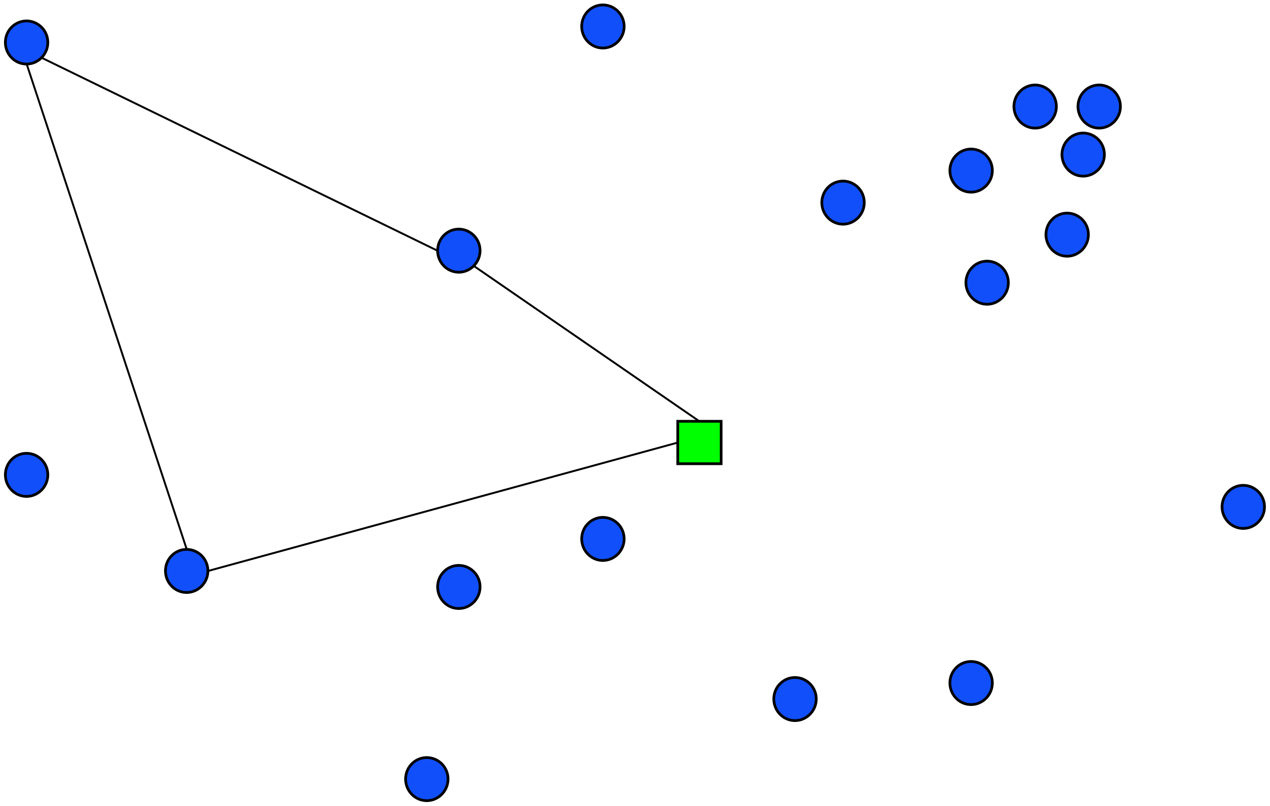
Possible route

Vehicle routing and scheduling



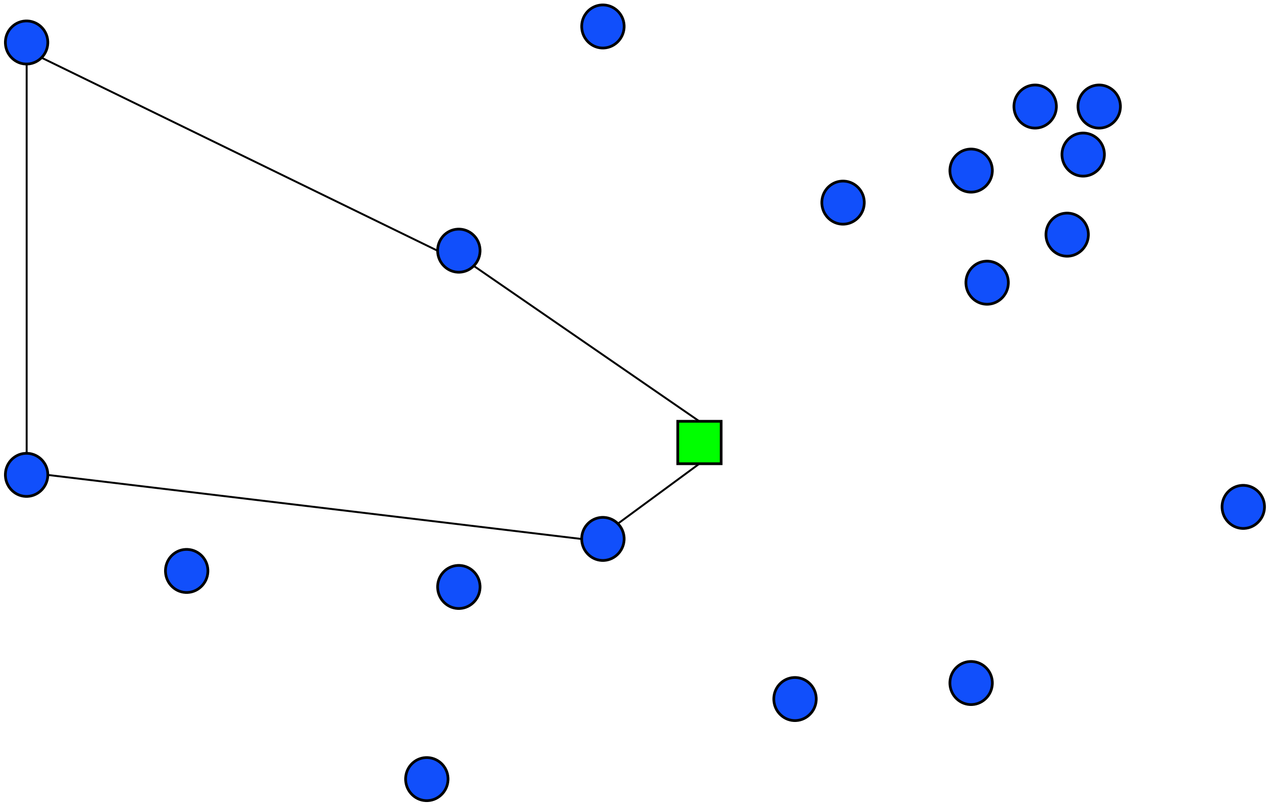
Possible route

Vehicle routing and scheduling



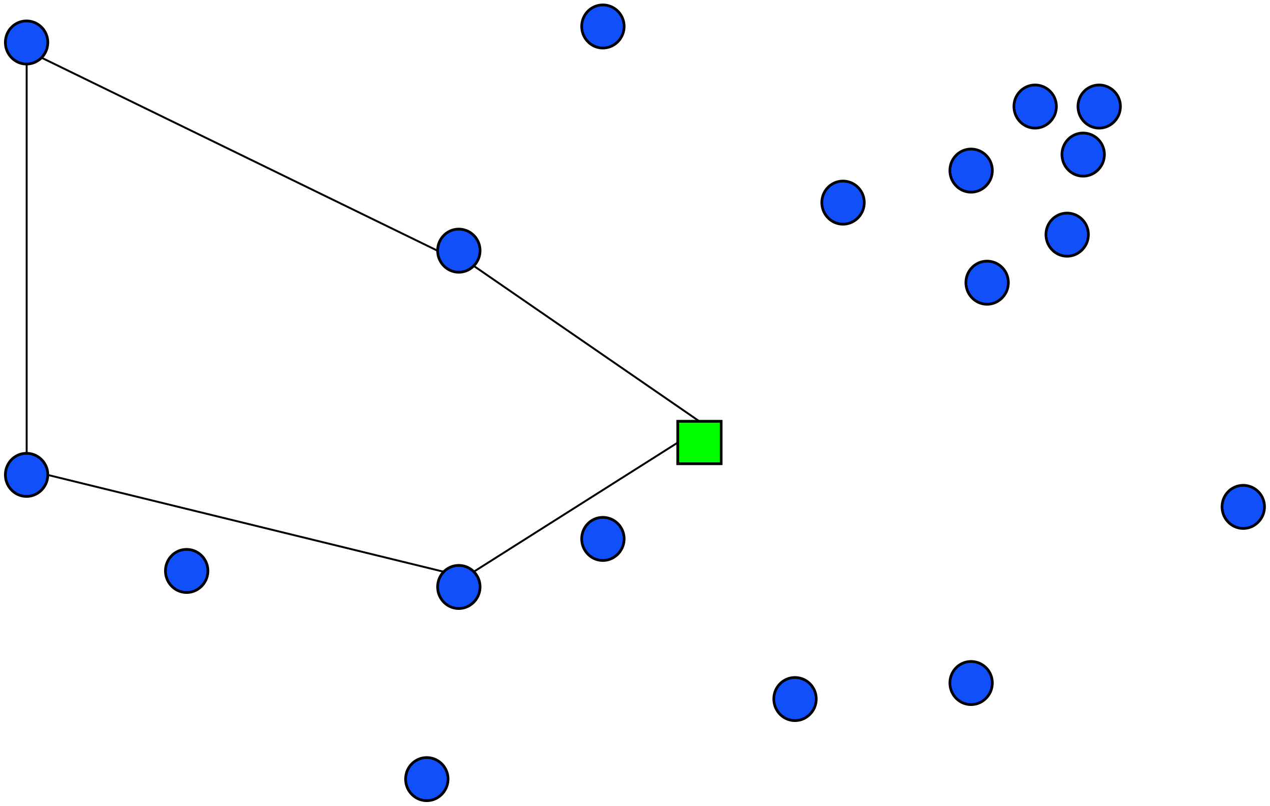
Possible route

Vehicle routing and scheduling



Possible route

Vehicle routing and scheduling



Possible route

Set partitioning formulation

	c_1	c_2	c_3	c_4	…		
Cust 1	1	1	0	1	…	=	1
Cust 2	1	0	1	0	…	=	1
Cust 3	0	1	1	0	…	=	1
…							
Cust n	0	1	0	0	…	=	1
	y/n	y/n	y/n	y/n	…		

Set covering based algorithms

- Advantage
  - very flexible
    - heuristics for route generation
    - complicating constraints in route generation
- Disadvantage
  - small to medium size instances