

# Discrete Optimization

Assignments: Vehicle Routing

# The Vehicle Routing Problem (VRP)

- Many variants
  - This is the Capacitated VRP - CVRP

# The Vehicle Routing Problem (VRP)

- ▶ Many variants
  - This is the Capacitated VRP - CVRP
- ▶ Like the Traveling Salesman Problem

# The Vehicle Routing Problem (VRP)

- ▶ Many variants
  - This is the Capacitated VRP - CVRP
- ▶ Like the Traveling Salesman Problem
  - on steroids...

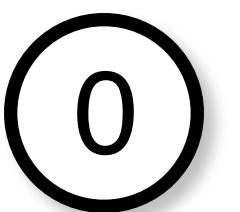
# Vehicle Routing

# Vehicle Routing

$x_2, y_2 = -10, 10$     $x_1, y_1 = 0, 10$



$x_0, y_0 = 0, 0$



$x_3, y_3 = 0, -10$     $x_4, y_4 = 10, -10$

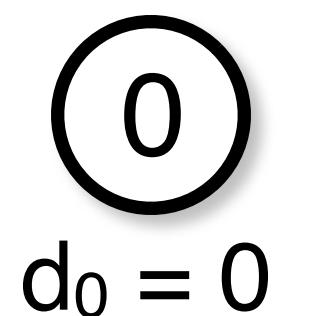


# Vehicle Routing

$x_2, y_2 = -10, 10$     $x_1, y_1 = 0, 10$



$x_0, y_0 = 0, 0$



$x_3, y_3 = 0, -10$     $x_4, y_4 = 10, -10$



# Vehicle Routing

$x_2, y_2 = -10, 10$     $x_1, y_1 = 0, 10$



vehicles = 4

$x_0, y_0 = 0, 0$

Diagram showing node 0, which is a circle with a black outline centered at the origin (0, 0).

Node 0:  $d_0 = 0$

$x_3, y_3 = 0, -10$     $x_4, y_4 = 10, -10$



# Vehicle Routing

$x_2, y_2 = -10, 10$     $x_1, y_1 = 0, 10$



$x_0, y_0 = 0, 0$

vehicles = 4

capacity = 10

Node 0:  $d_0 = 0$

$x_3, y_3 = 0, -10$     $x_4, y_4 = 10, -10$



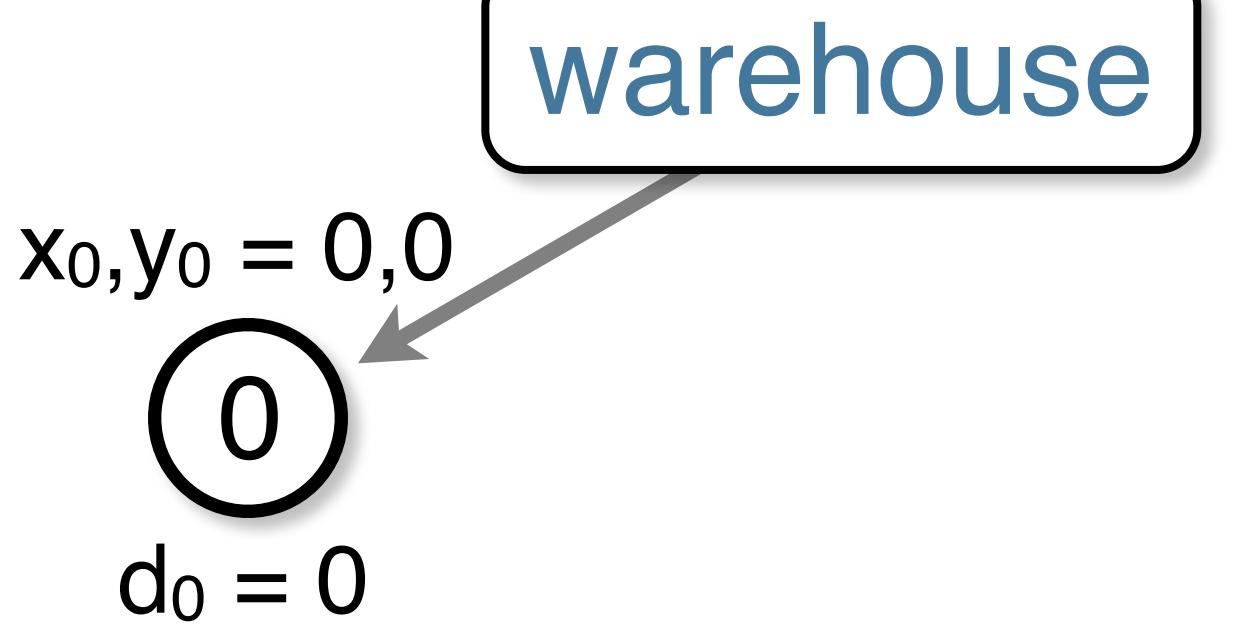
# Vehicle Routing

$$x_2, y_2 = -10, 10 \quad x_1, y_1 = 0, 10$$

(2)  
 $d_2 = 3$

(1)  
 $d_1 = 3$

vehicles = 4  
capacity = 10



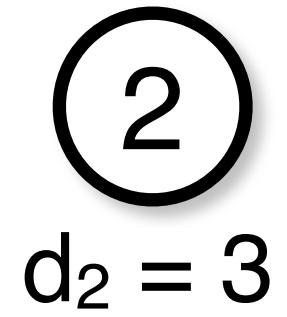
$$x_3, y_3 = 0, -10 \quad x_4, y_4 = 10, -10$$

(3)  
 $d_3 = 3$

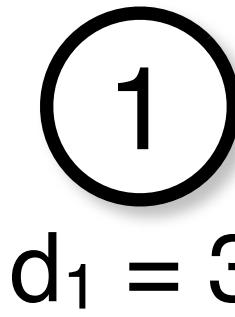
(4)  
 $d_4 = 3$

# Vehicle Routing

$x_2, y_2 = -10, 10$     $x_1, y_1 = 0, 10$



$d_2 = 3$



$d_1 = 3$

vehicles = 4  
capacity = 10

$x_3, y_3 = 0, -10$     $x_4, y_4 = 10, -10$

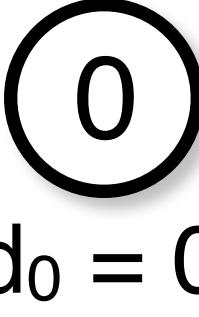


$d_3 = 3$



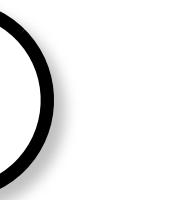
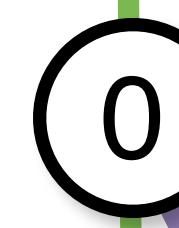
$d_4 = 3$

$x_0, y_0 = 0, 0$



warehouse

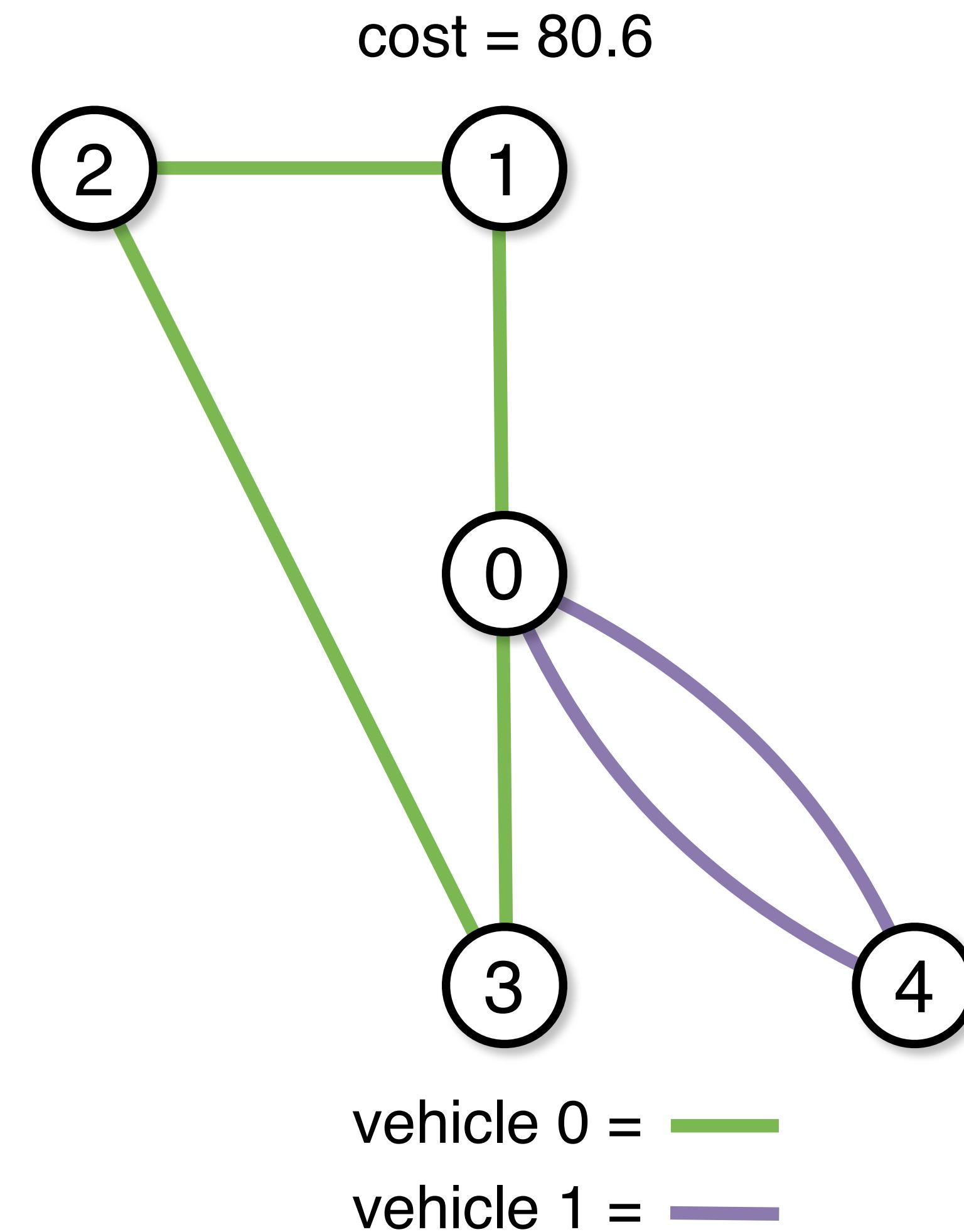
cost = 80.6



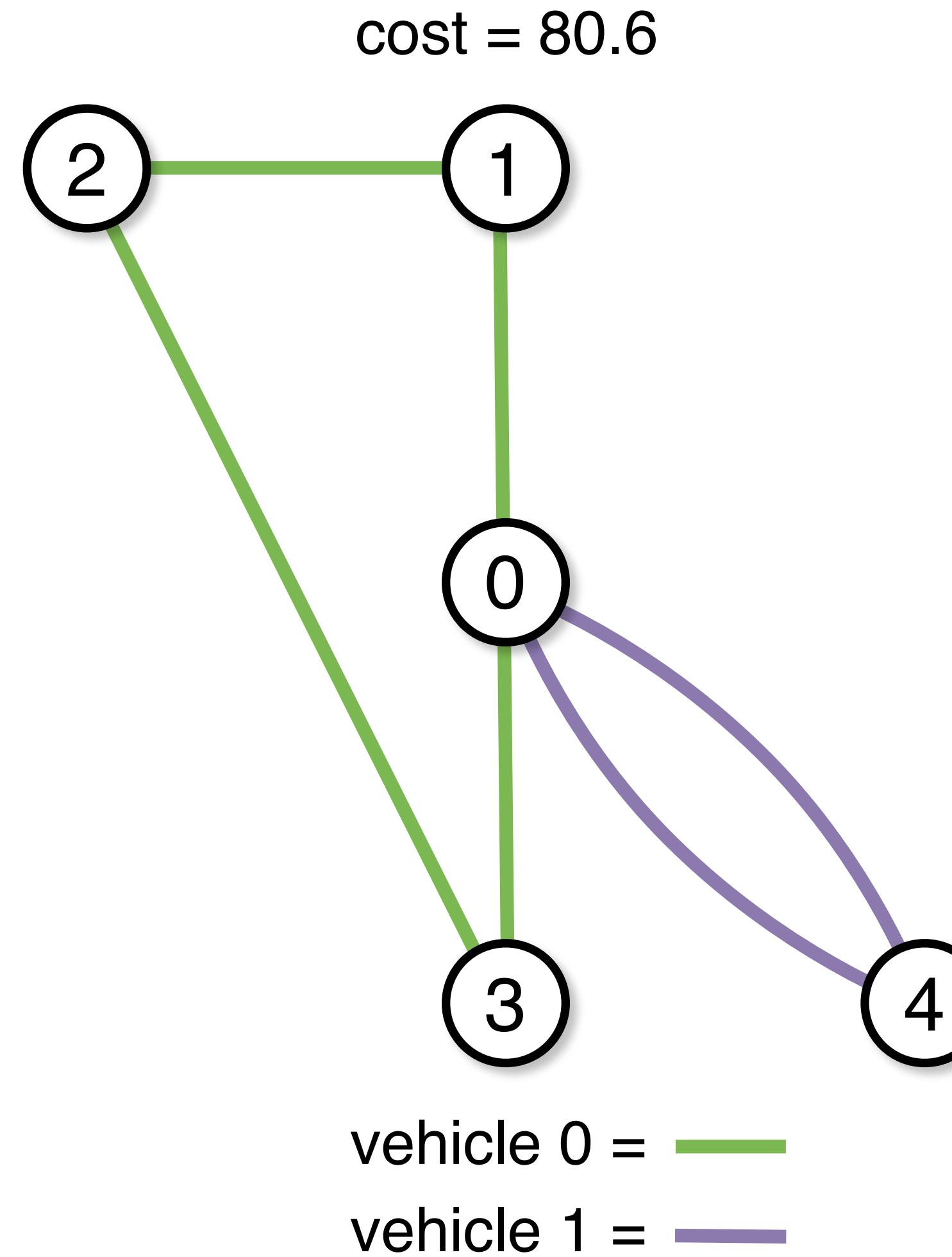
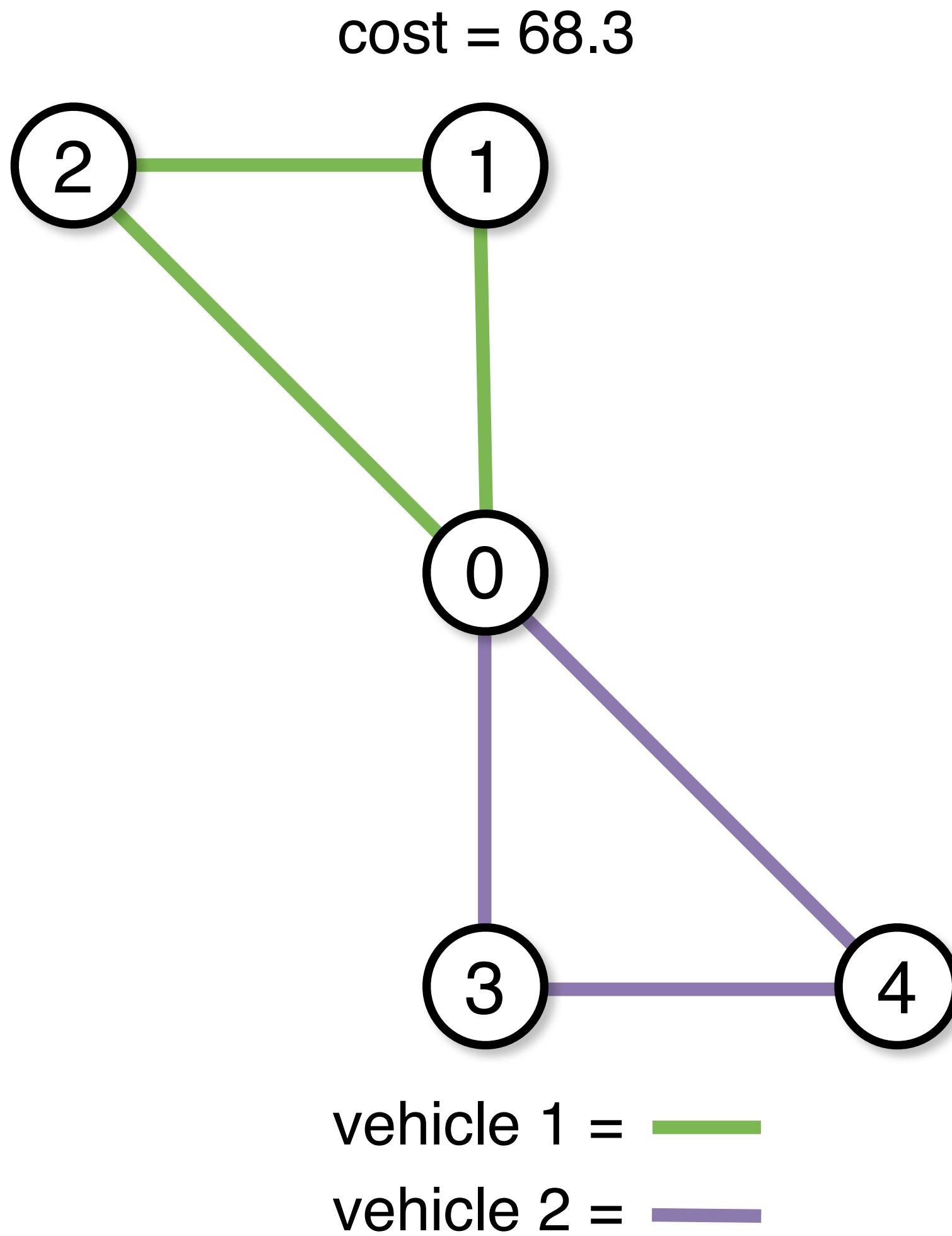
vehicle 0 =

vehicle 1 =

# Vehicle Routing



# Vehicle Routing



# Vehicle Routing

- $n$  Locations,  $v$  Vehicles
- For each location,
  - demand  $d_i$  and location  $x_i, y_i$
- The capacity of the vehicles  $c$
- The sequence of deliveries of vehicle  $i$ ,  $T_i$

minimize:  $\sum_{i \in V} \left( dist(0, T_{i,0}) + \sum_{\langle j, k \rangle \in T_i} dist(j, k) + dist(T_{i,|T_i|-1}, 0) \right)$

subject to:

$$\sum_{j \in T_i} d_j \leq c \quad (i \in V)$$

$$\sum_{i \in V} (j \in T_i) = 1 \quad (j \in N \setminus \{0\})$$

# Vehicle Routing Data Format

minimize:  $\sum_{i \in V} \left( dist(0, T_{i,0}) + \sum_{\langle j, k \rangle \in T_i} dist(j, k) + dist(T_{i,|T_i|-1}, 0) \right)$

subject to:

$$\sum_{j \in T_i} d_j \leq c \quad (i \in V)$$

$$\sum_{i \in V} (j \in T_i) = 1 \quad (j \in N \setminus \{0\})$$

Input

N	V	c
d_0	x_0	y_0
d_1	x_1	y_1
...		
d_  N -1	x_  N -1	y_  N -1

Output

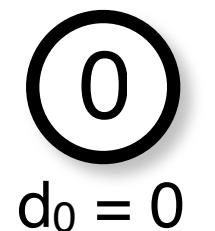
obj	opt
0	t_0_1 t_0_2 ... 0
0	t_1_1 t_1_2 ... 0
...	
0	t_ V -1_1 t_ V -1_2 ... 0

# Vehicle Routing Example

$x_2, y_2 = -10, 10$      $x_1, y_1 = 0, 10$



$x_0, y_0 = 0, 0$   
vehicles = 4  
capacity = 10



$x_3, y_3 = 0, -10$      $x_4, y_4 = 10, -10$



Input

5	4	10
0	0	0
3	0	10
3	-10	10
3	0	-10
3	10	-10

# Vehicle Routing Example

$x_2, y_2 = -10, 10$      $x_1, y_1 = 0, 10$

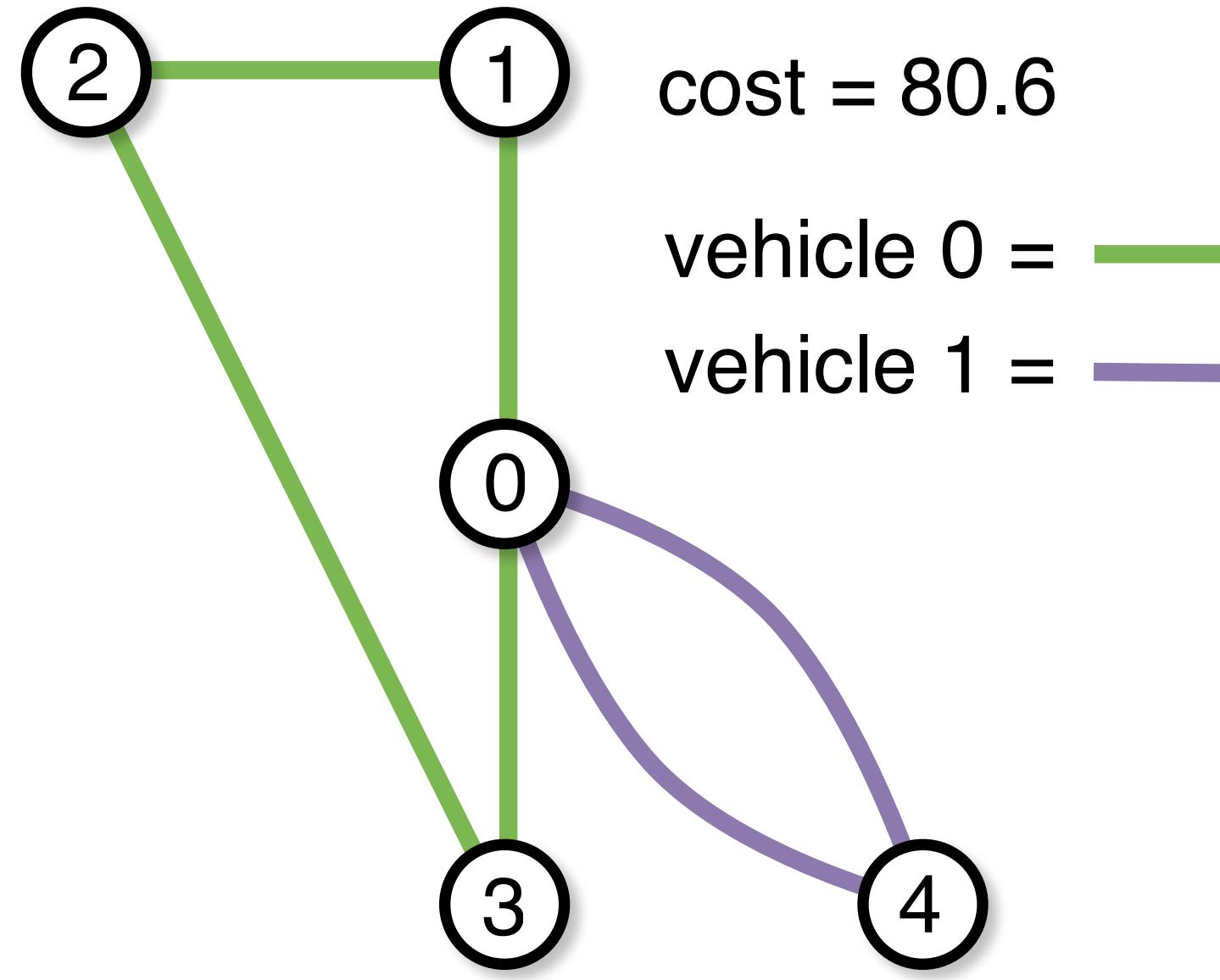
2  
 $d_2 = 3$

1  
 $d_1 = 3$

0  
 $x_0, y_0 = 0, 0$   
vehicles = 4  
capacity = 10

3  
 $x_3, y_3 = 0, -10$   
 $d_3 = 3$

4  
 $x_4, y_4 = 10, -10$   
 $d_4 = 3$



Input

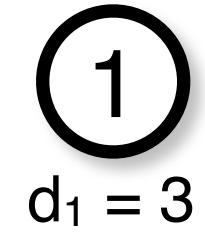
5	4	10
0	0	0
3	0	10
3	-10	10
3	0	-10
3	10	-10

Output

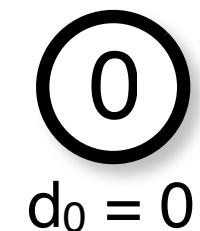
80.6	0
0	1
2	3
0	4
0	0
0	0

# Vehicle Routing Example

$x_2, y_2 = -10, 10$      $x_1, y_1 = 0, 10$



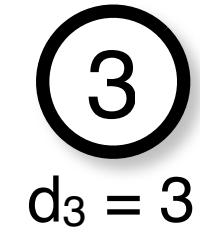
$x_0, y_0 = 0, 0$



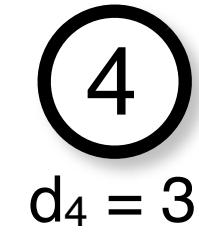
vehicles = 4

capacity = 10

$x_3, y_3 = 0, -10$

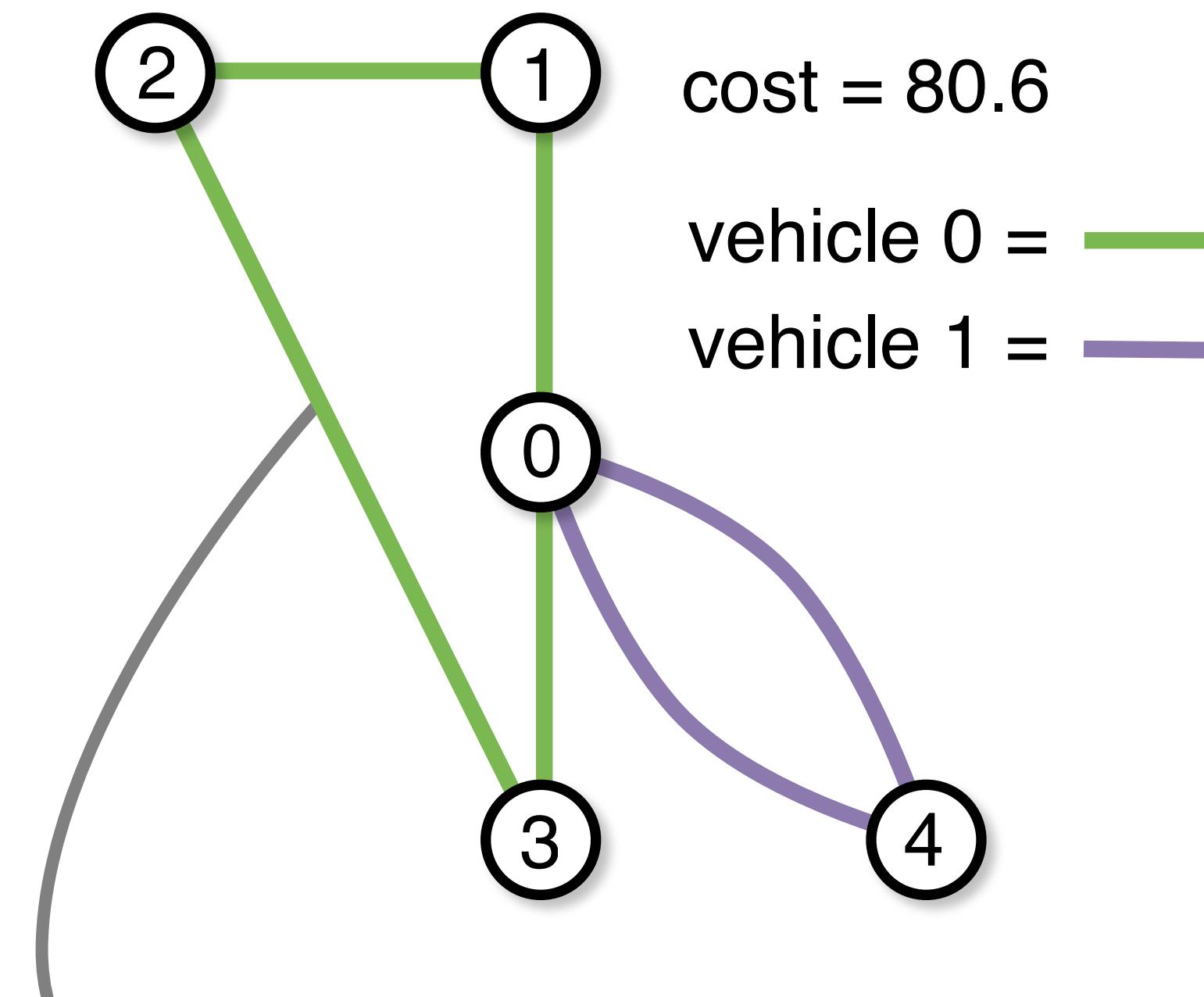


$x_4, y_4 = 10, -10$



Input

5	4	10
0	0	0
3	0	10
3	-10	10
3	0	-10
3	10	-10

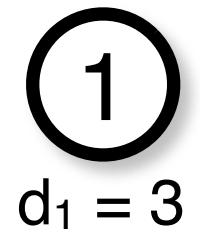


Output

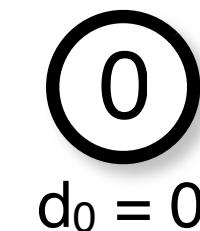
80.6	0
0	1
2	3
0	4
0	0
0	0

# Vehicle Routing Example

$x_2, y_2 = -10, 10$      $x_1, y_1 = 0, 10$



$x_0, y_0 = 0, 0$



vehicles = 4

capacity = 10

$x_3, y_3 = 0, -10$

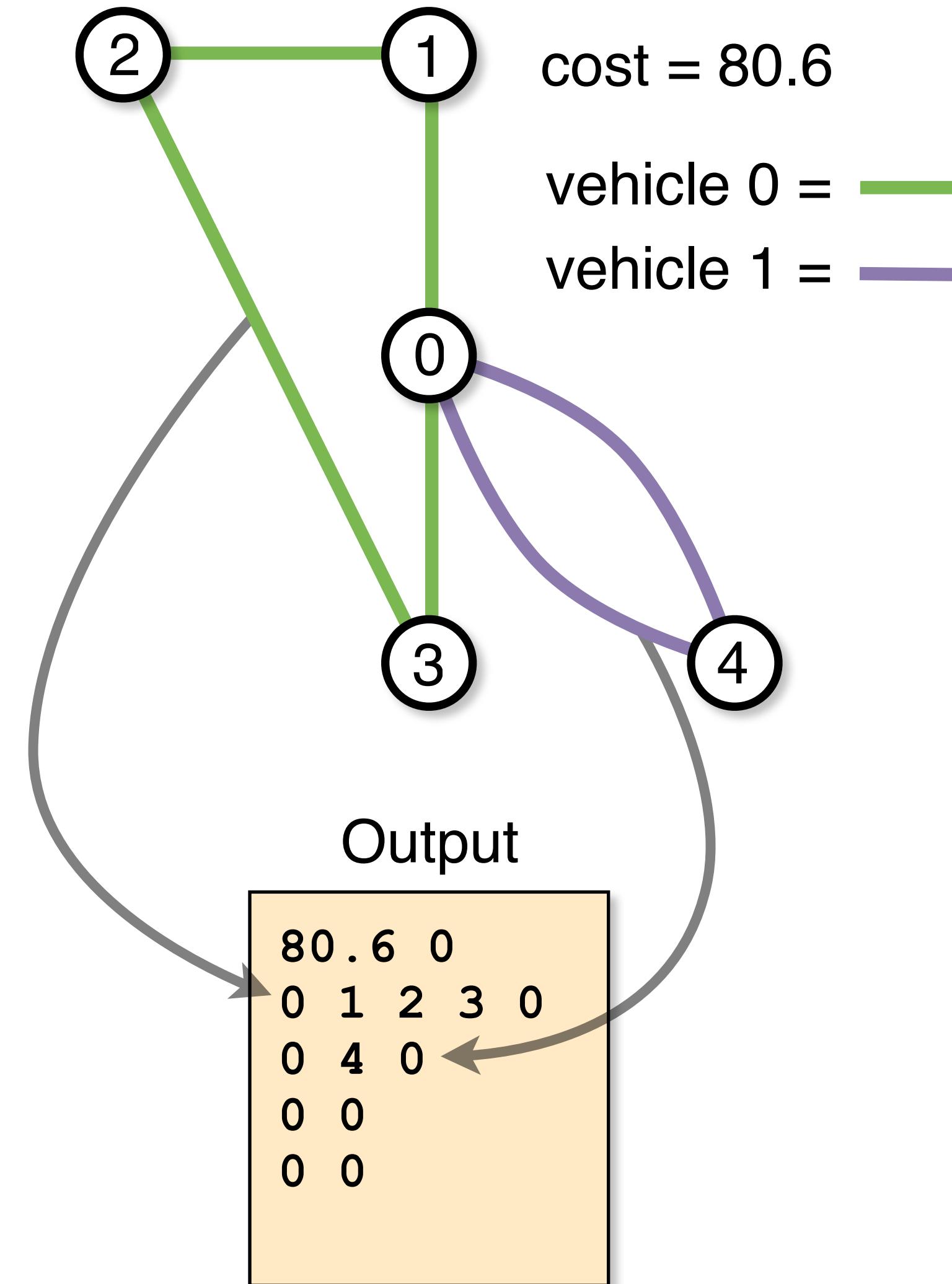


$x_4, y_4 = 10, -10$



Input

5	4	10
0	0	0
3	0	10
3	-10	10
3	0	-10
3	10	-10



Output

80.6	0
0	1
1	2
2	3
3	0
0	4
4	0
0	0
0	0

# Getting Started

- This assignment is really hard.
  - Very close to a real world application.

# Getting Started

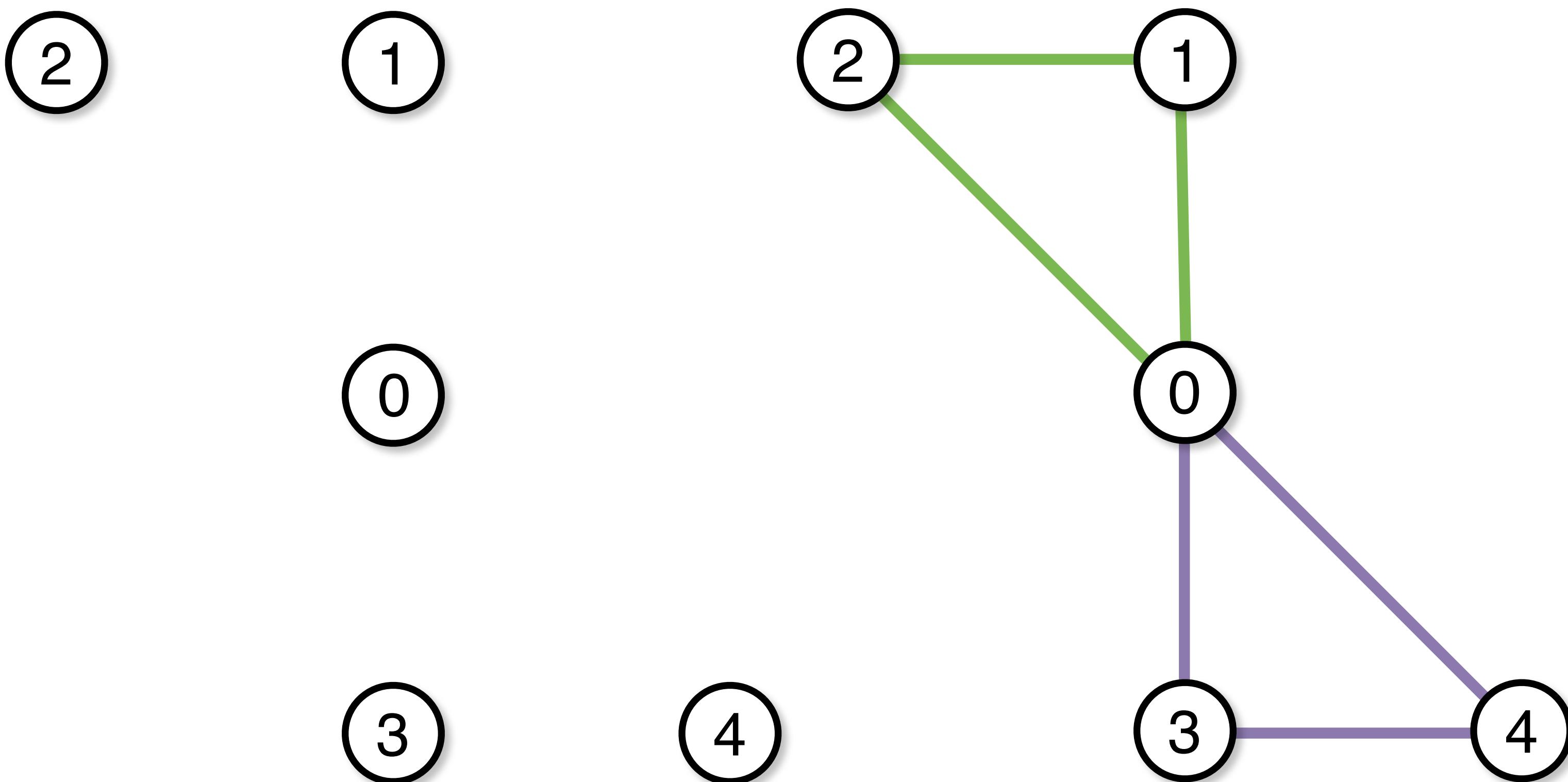
- ▶ This assignment is really hard.
  - Very close to a real world application.
- ▶ Three Models
  - CP
  - MIP
  - Local Search

# Getting Started

- ▶ This assignment is really hard.
  - Very close to a real world application.
- ▶ Three Models
  - CP
  - MIP
  - Local Search
- ▶ All connected to the TSP
  - Multi-Colored TSP

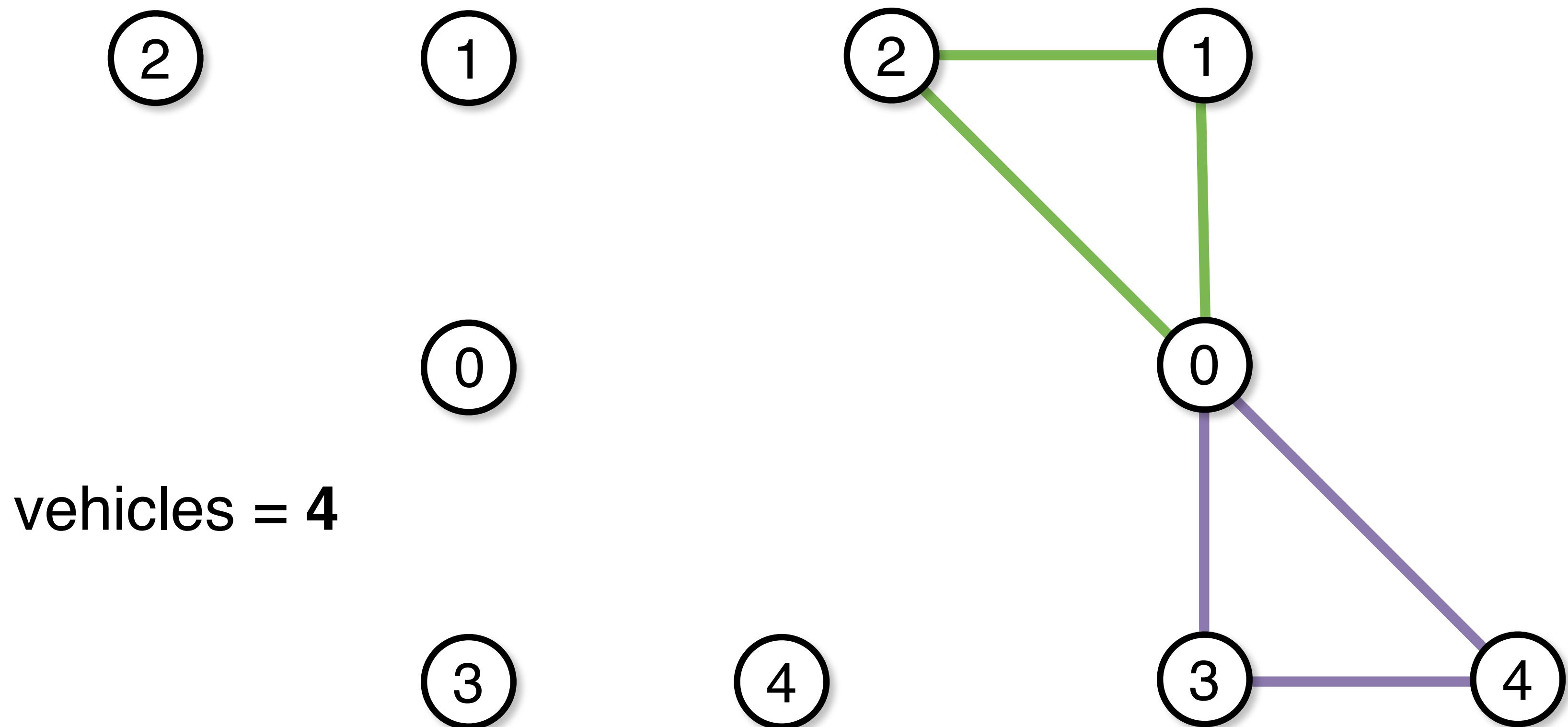
# A CP Model

- One Big ***Circuit*** Constraint



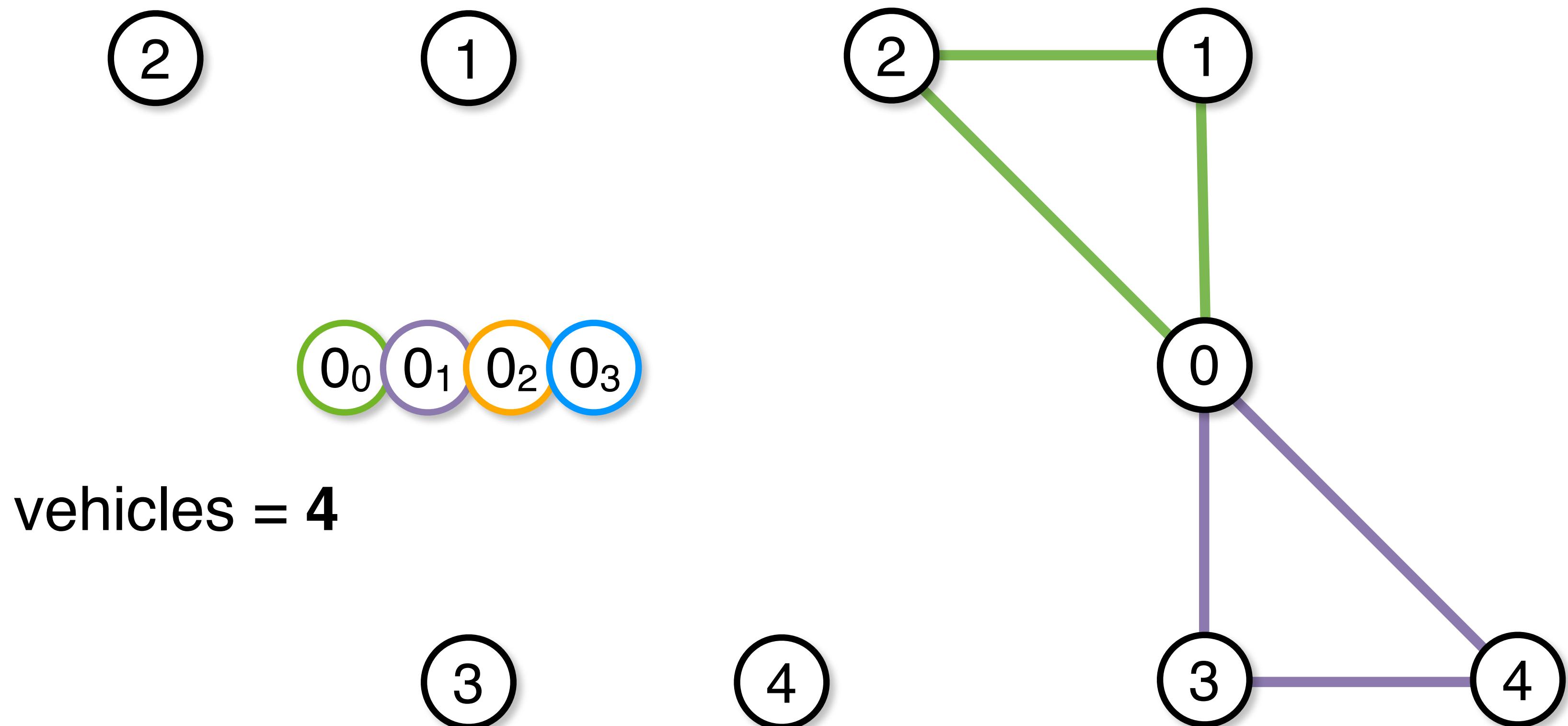
# A CP Model

- One Big ***Circuit*** Constraint



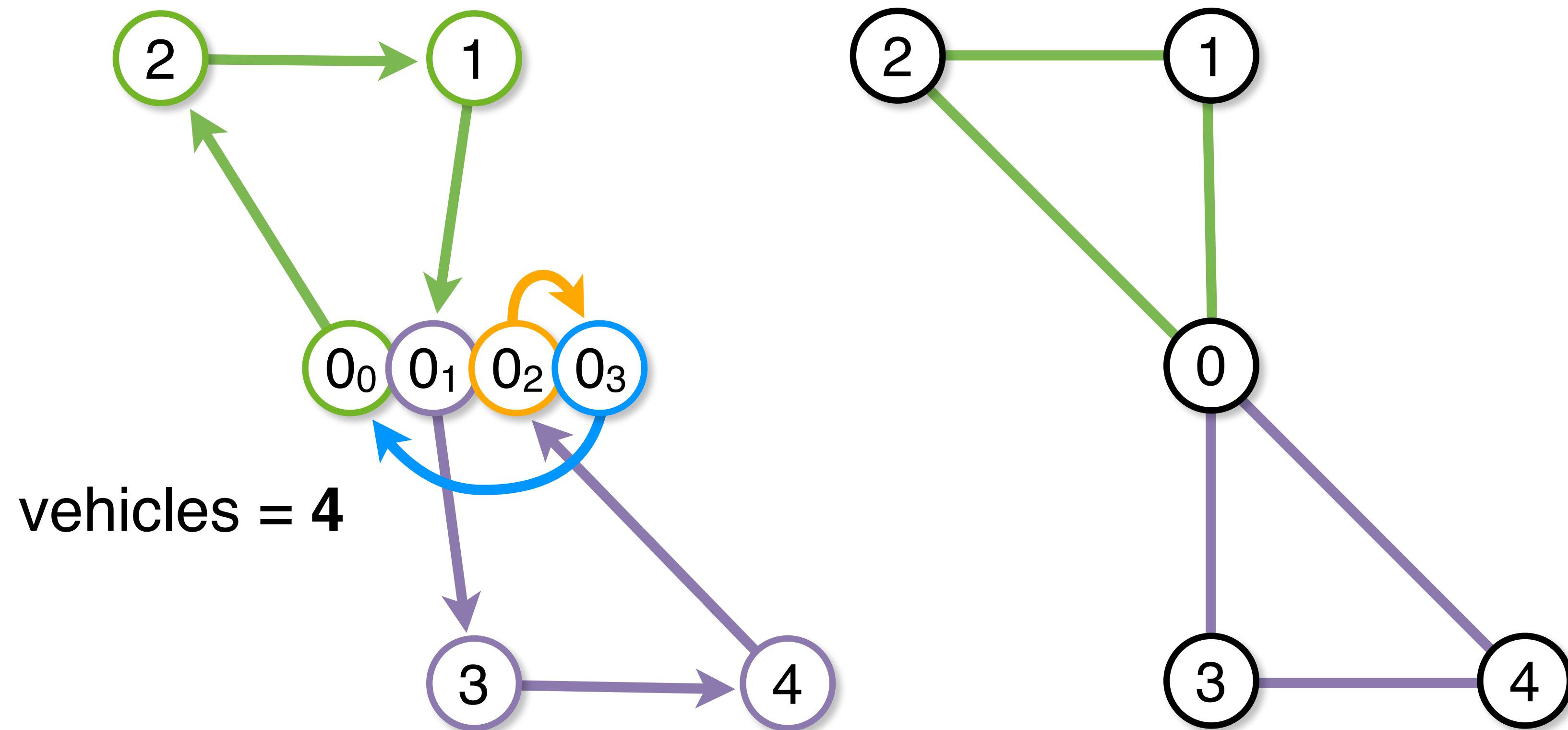
# A CP Model

- One Big ***Circuit*** Constraint



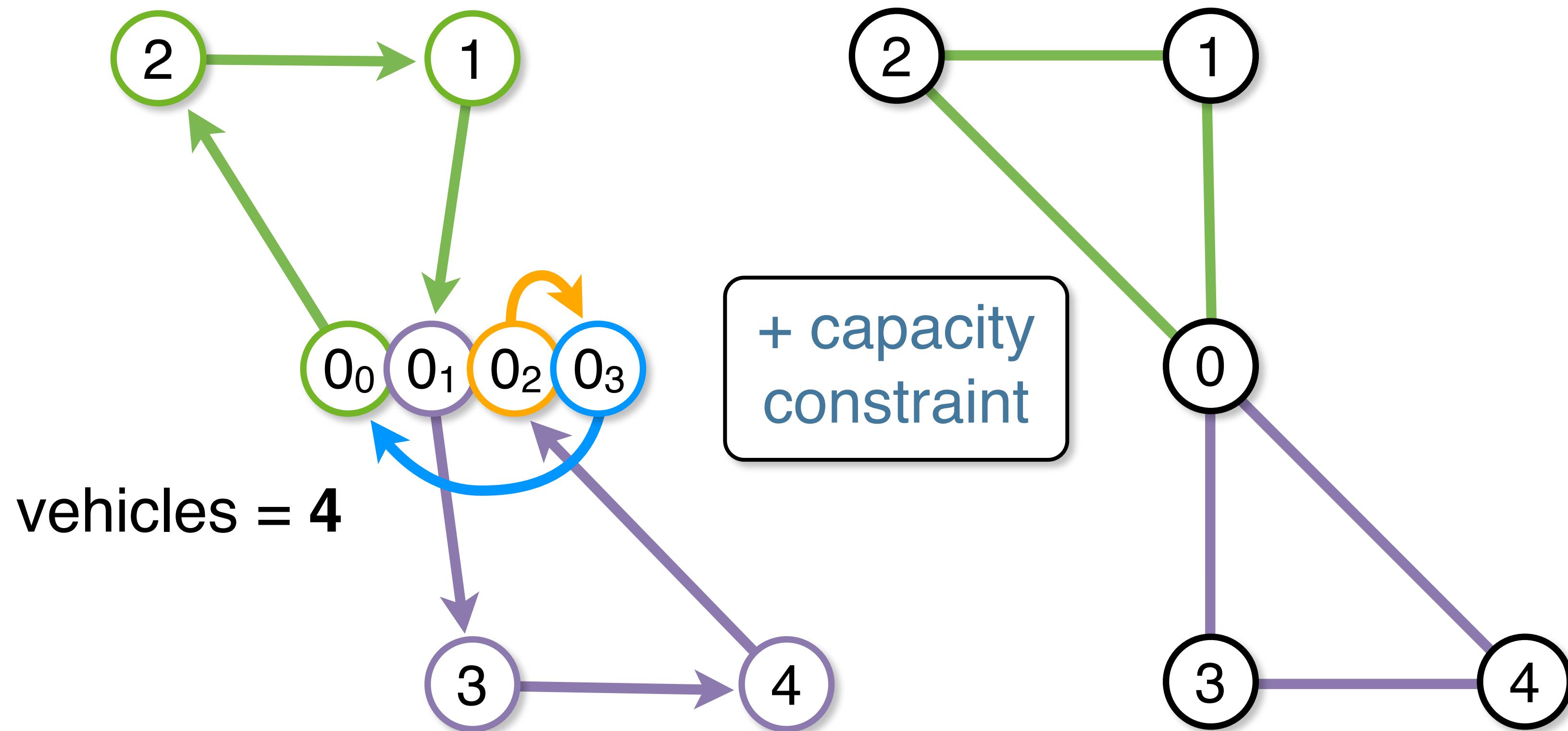
# A CP Model

- # ► One Big *Circuit* Constraint



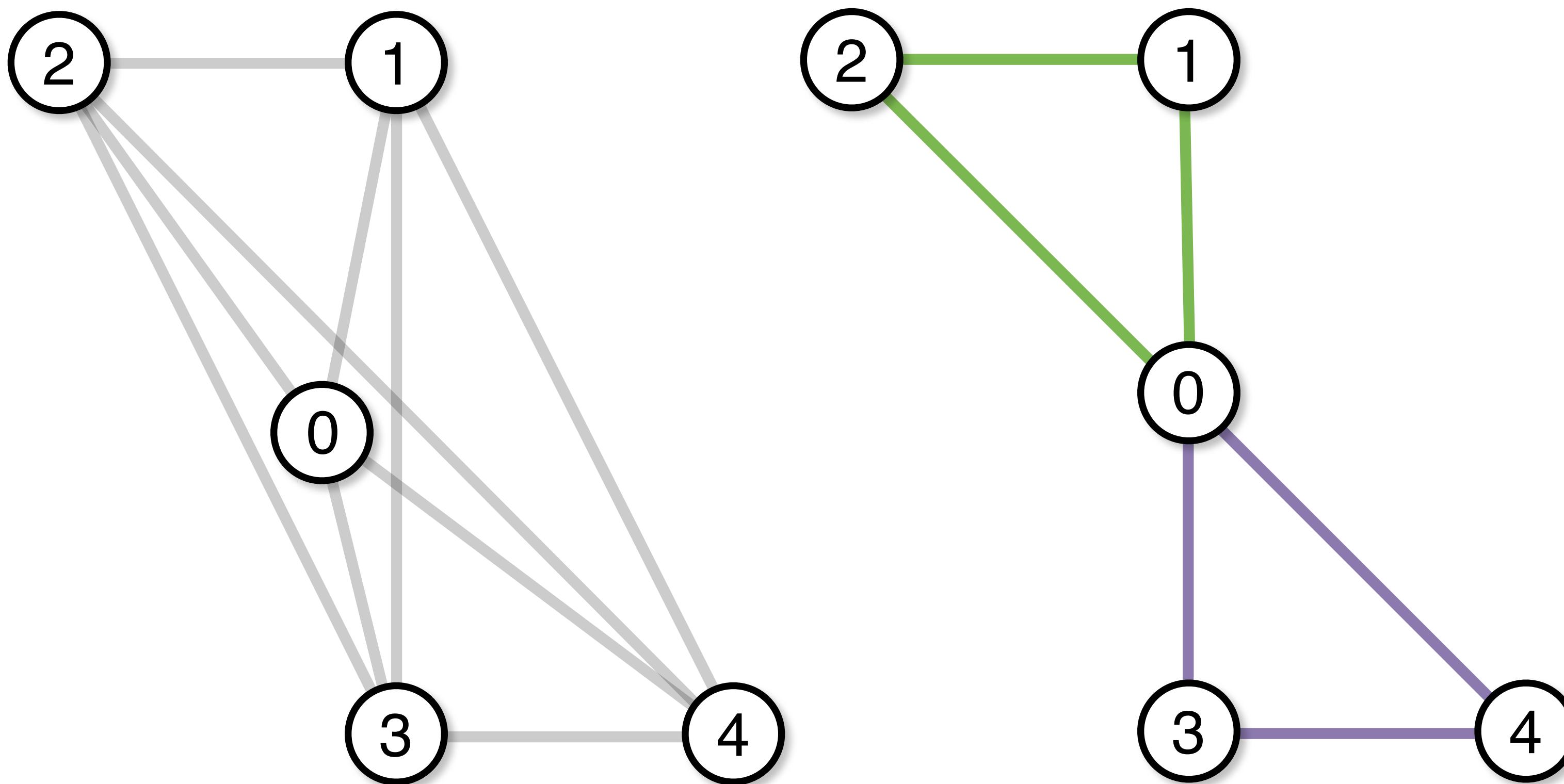
# A CP Model

- One Big ***Circuit*** Constraint



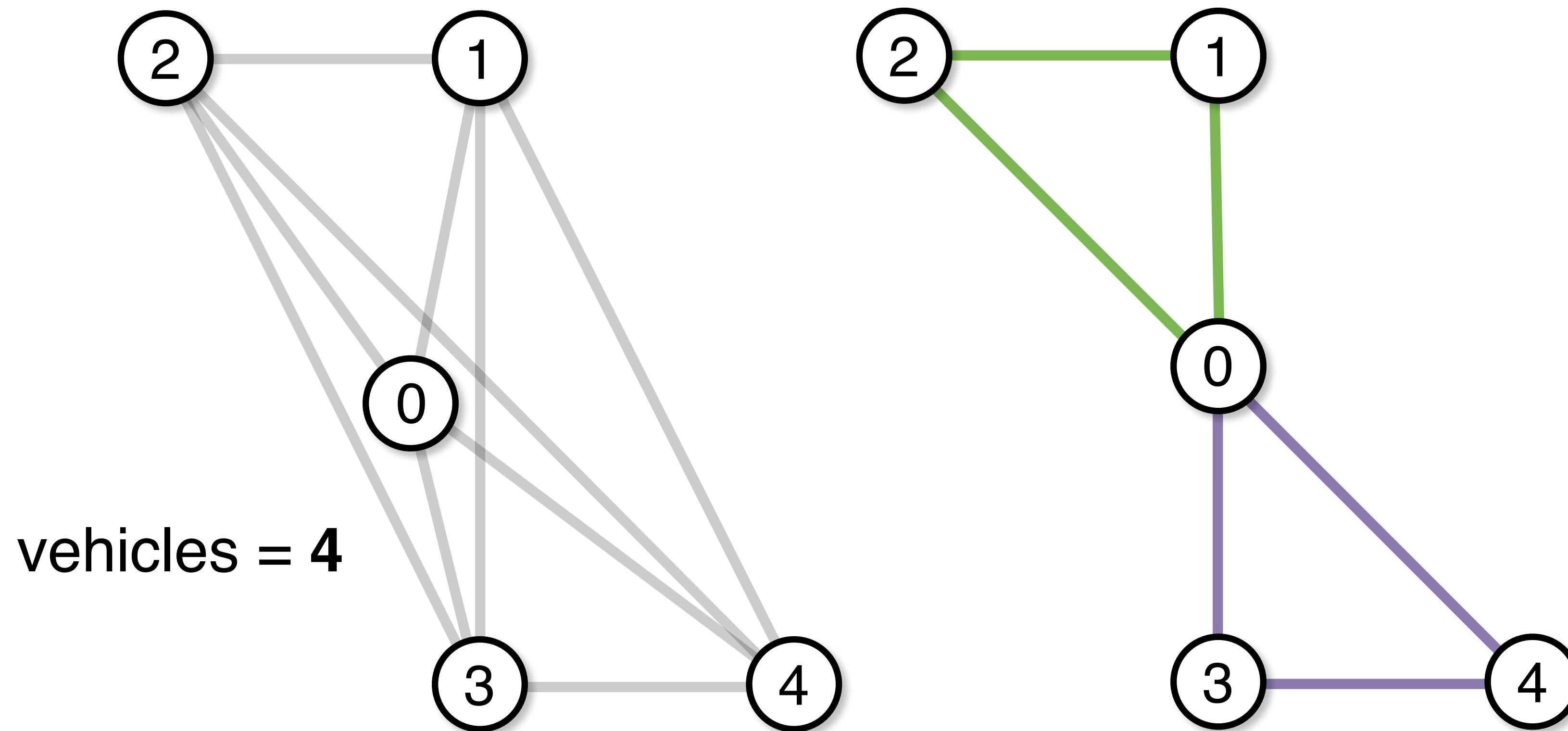
# A MIP Model

- Go with the *Flow* (recall MIP TSP Model)



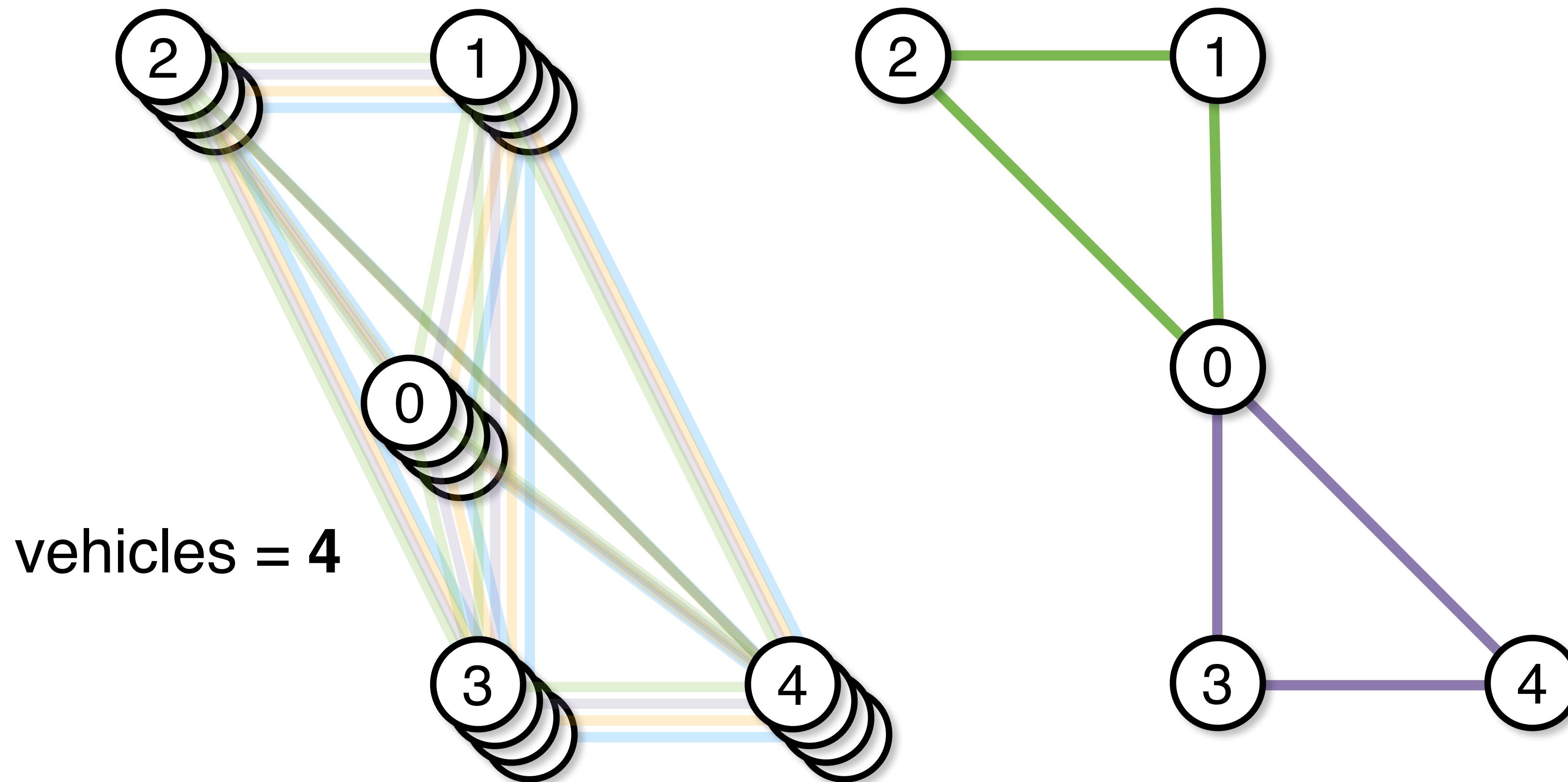
# A MIP Model

- Go with the *Flow* (recall MIP TSP Model)



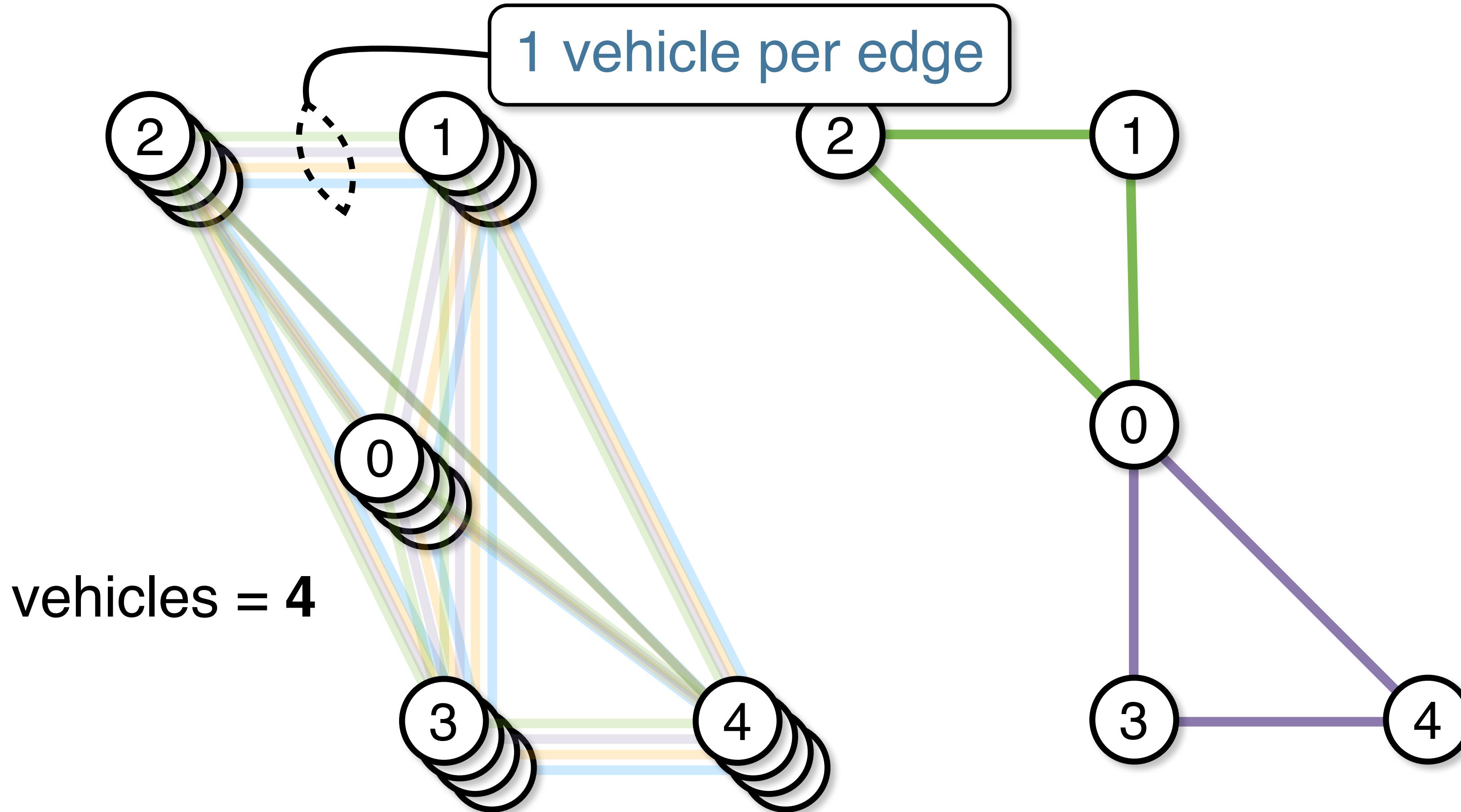
# A MIP Model

- Go with the *Flow* (recall MIP TSP Model)



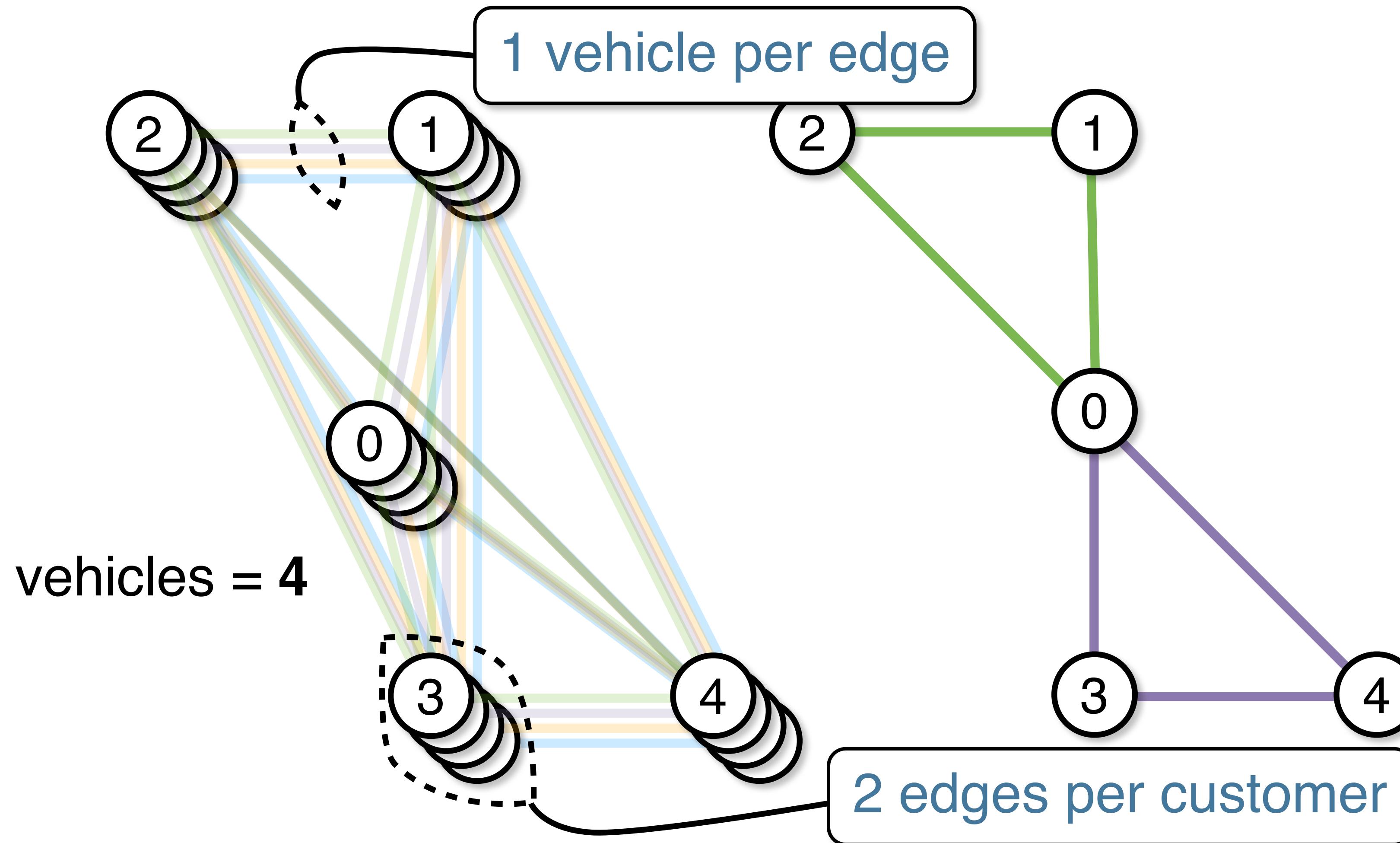
# A MIP Model

- Go with the *Flow* (recall MIP TSP Model)



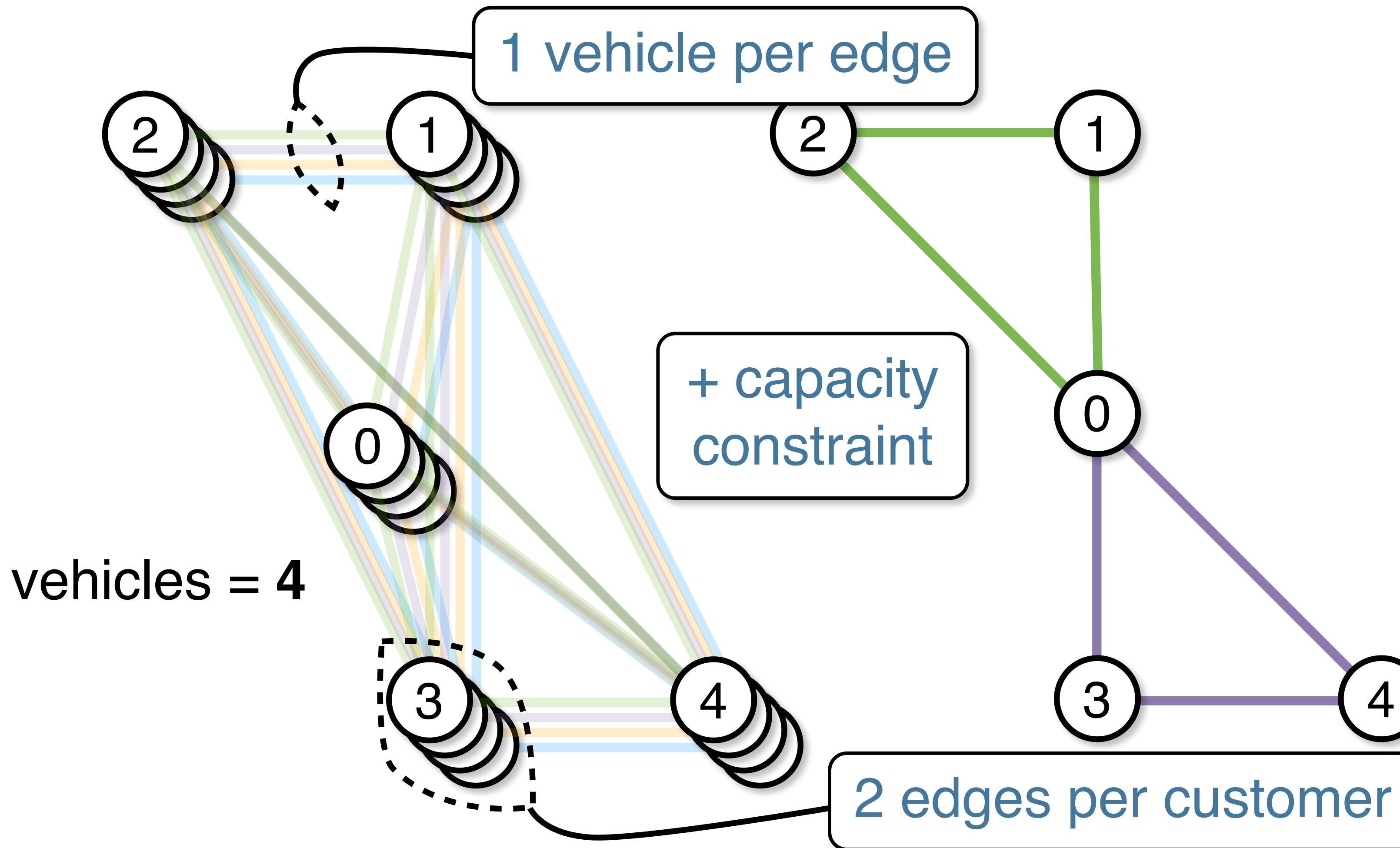
# A MIP Model

- # ► Go with the *Flow* (recall MIP TSP Model)



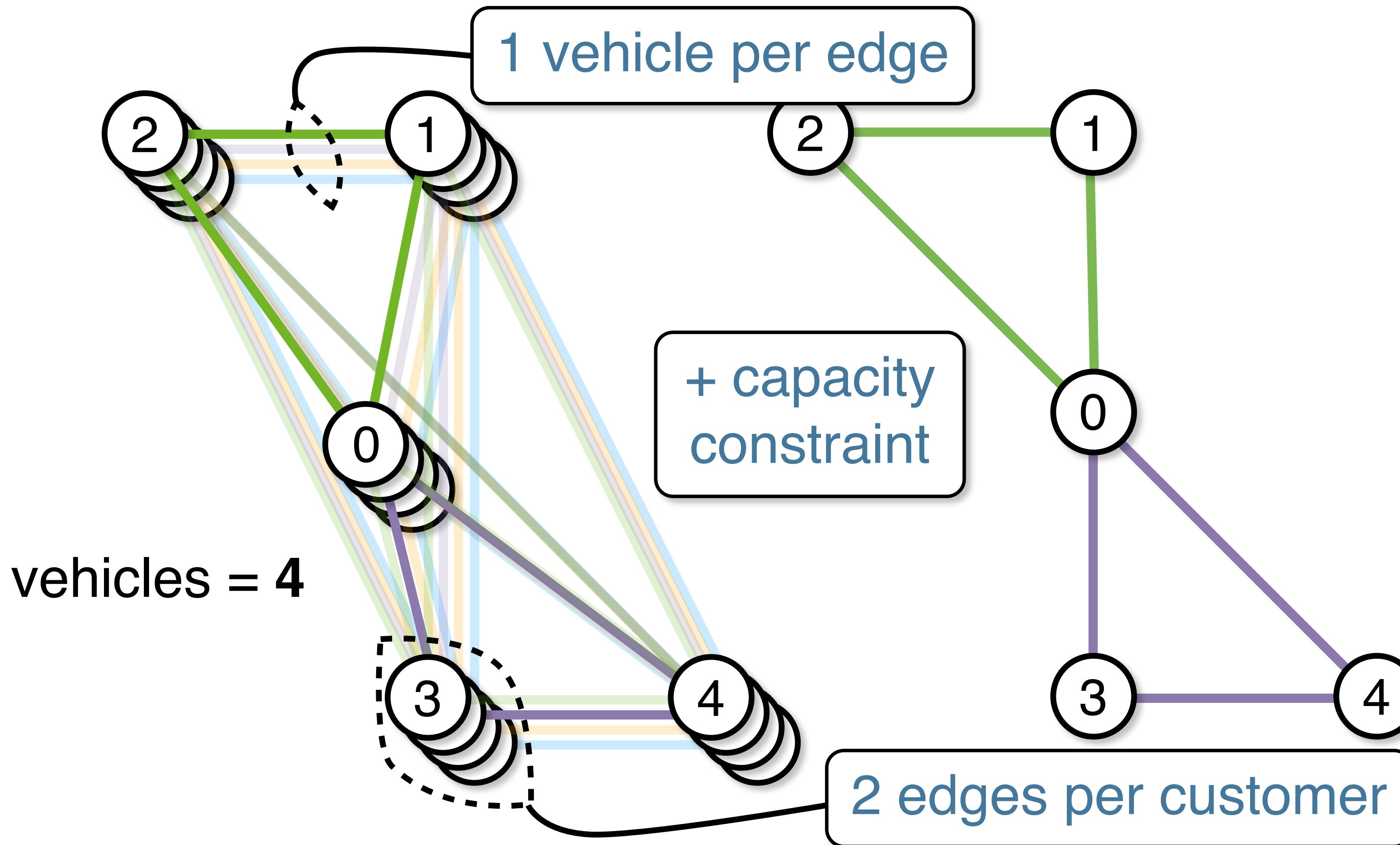
# A MIP Model

- Go with the ***Flow*** (recall MIP TSP Model)



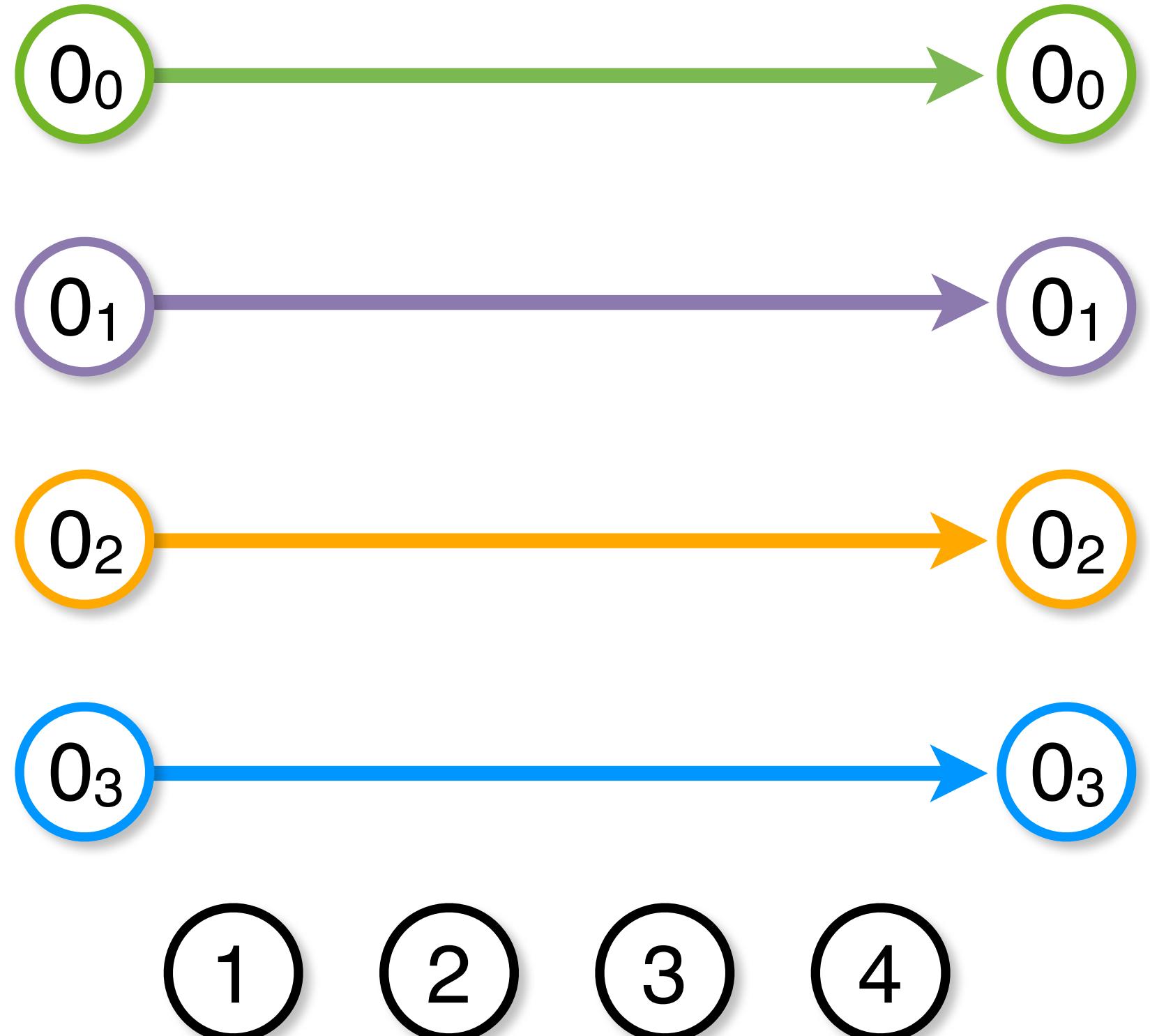
# A MIP Model

- Go with the ***Flow*** (recall MIP TSP Model)



# A Local Search Model

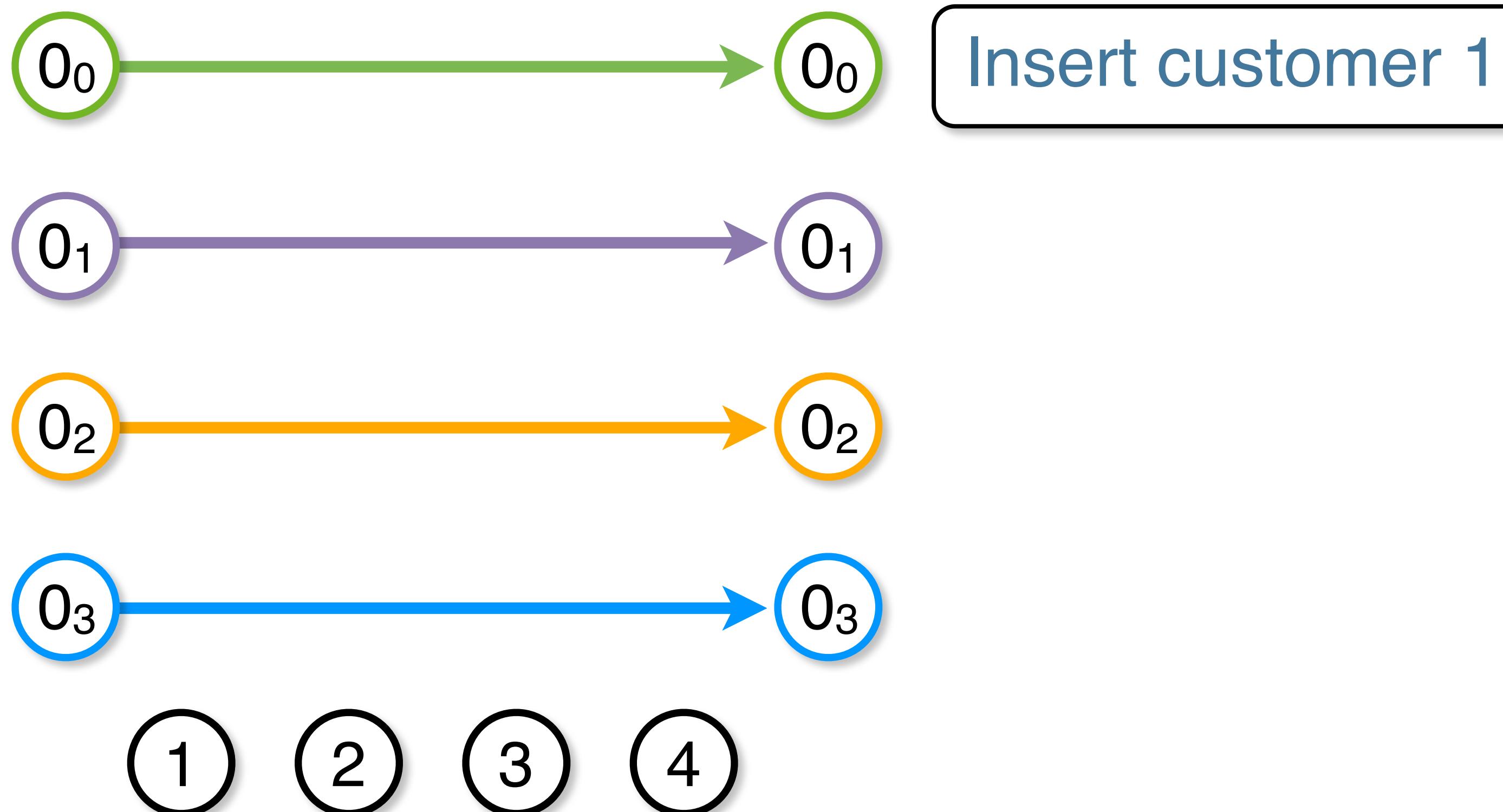
## ► Insert Customers



vehicles = 4

# A Local Search Model

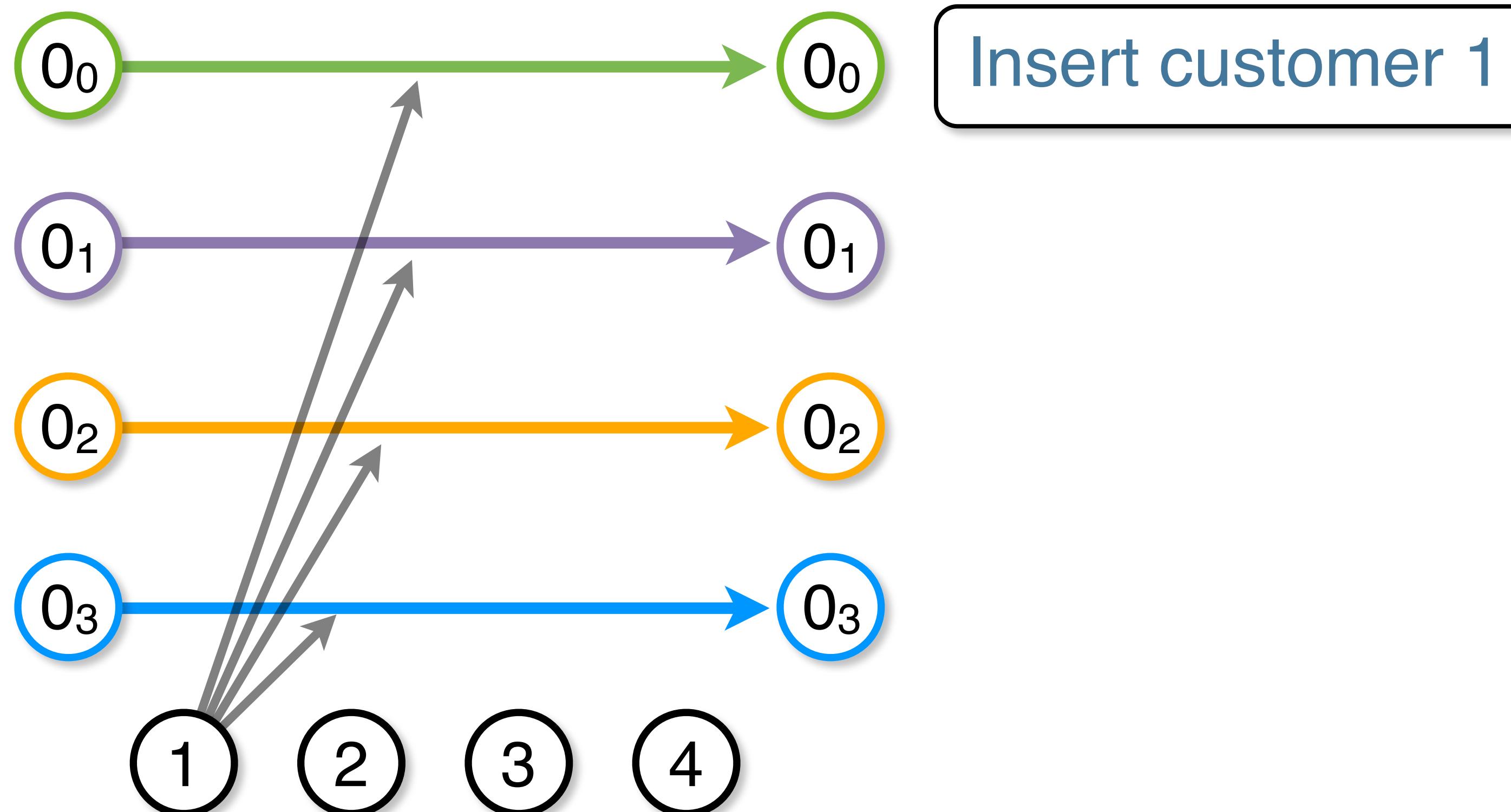
## ► Insert Customers



vehicles = 4

# A Local Search Model

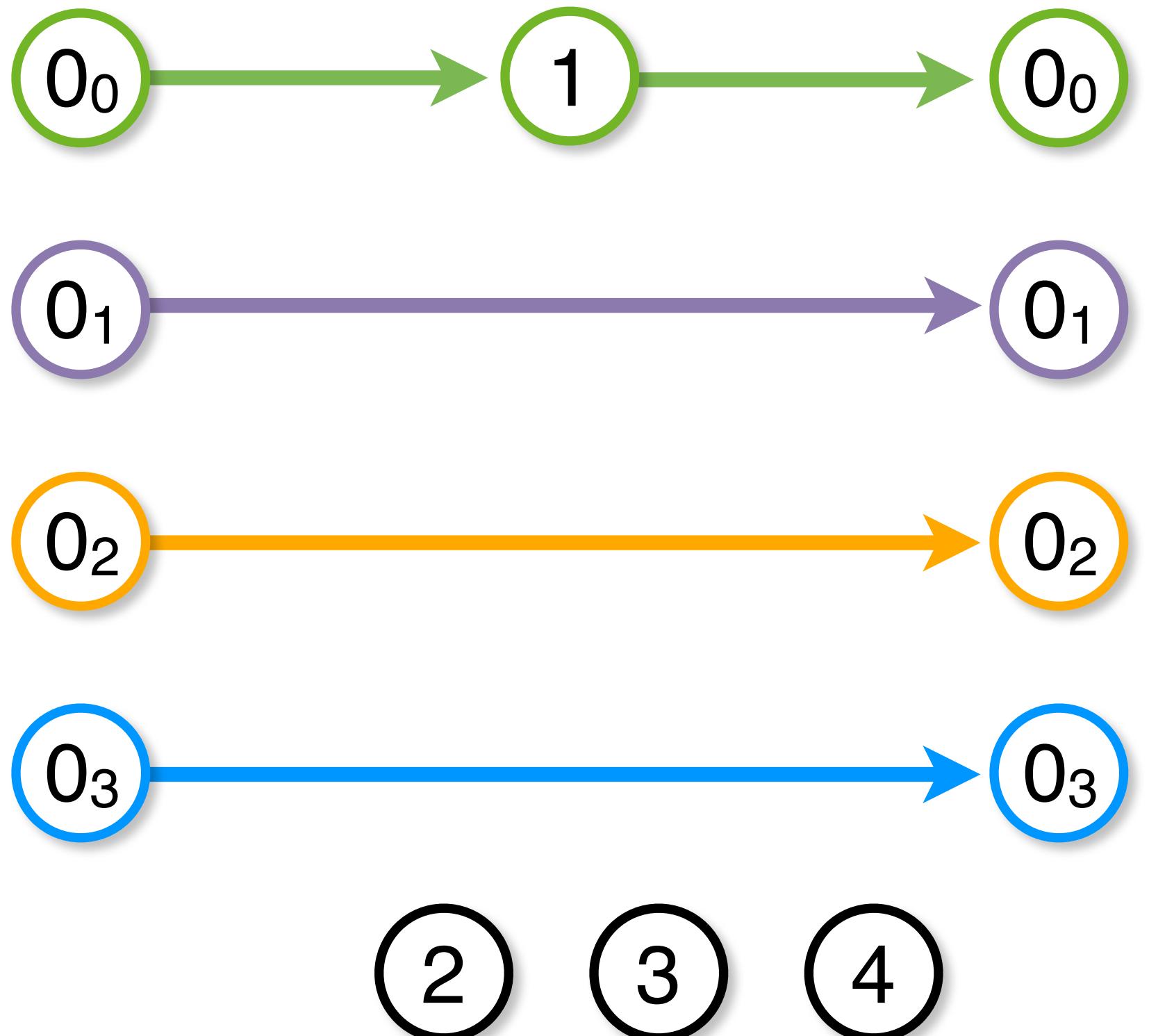
## ► Insert Customers



vehicles = 4

# A Local Search Model

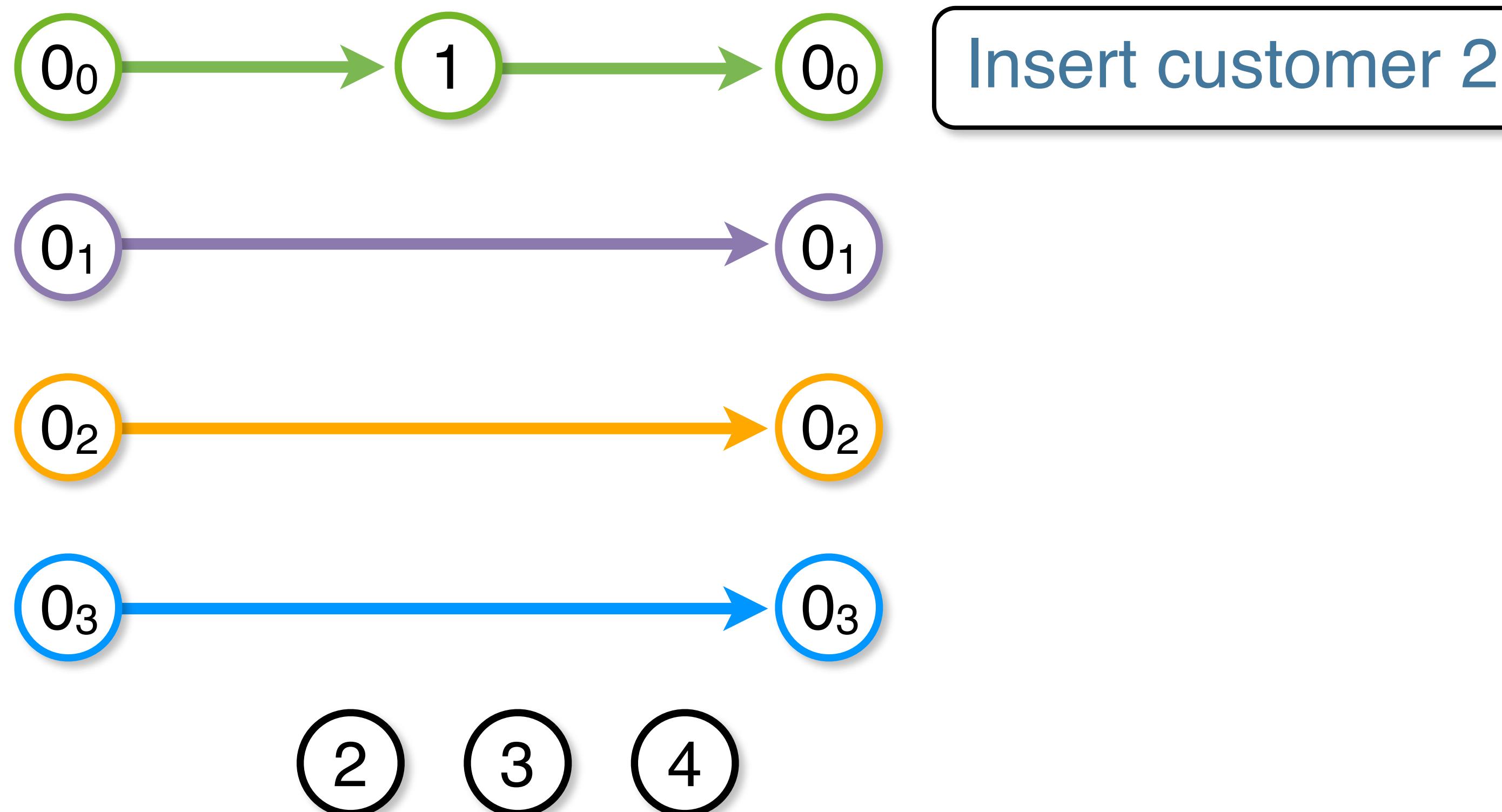
## ► Insert Customers



**vehicles = 4**

# A Local Search Model

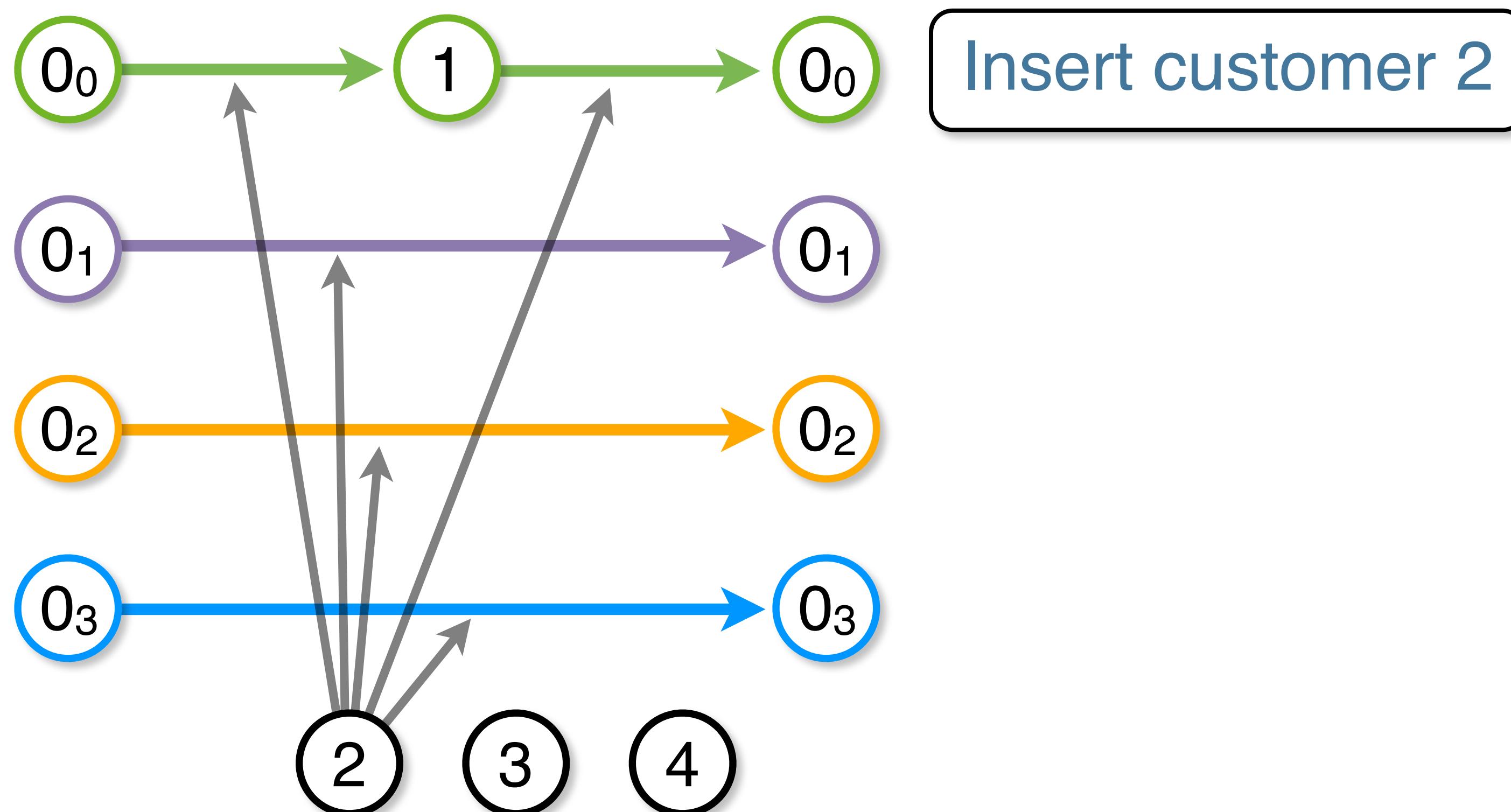
## ► Insert Customers



vehicles = 4

# A Local Search Model

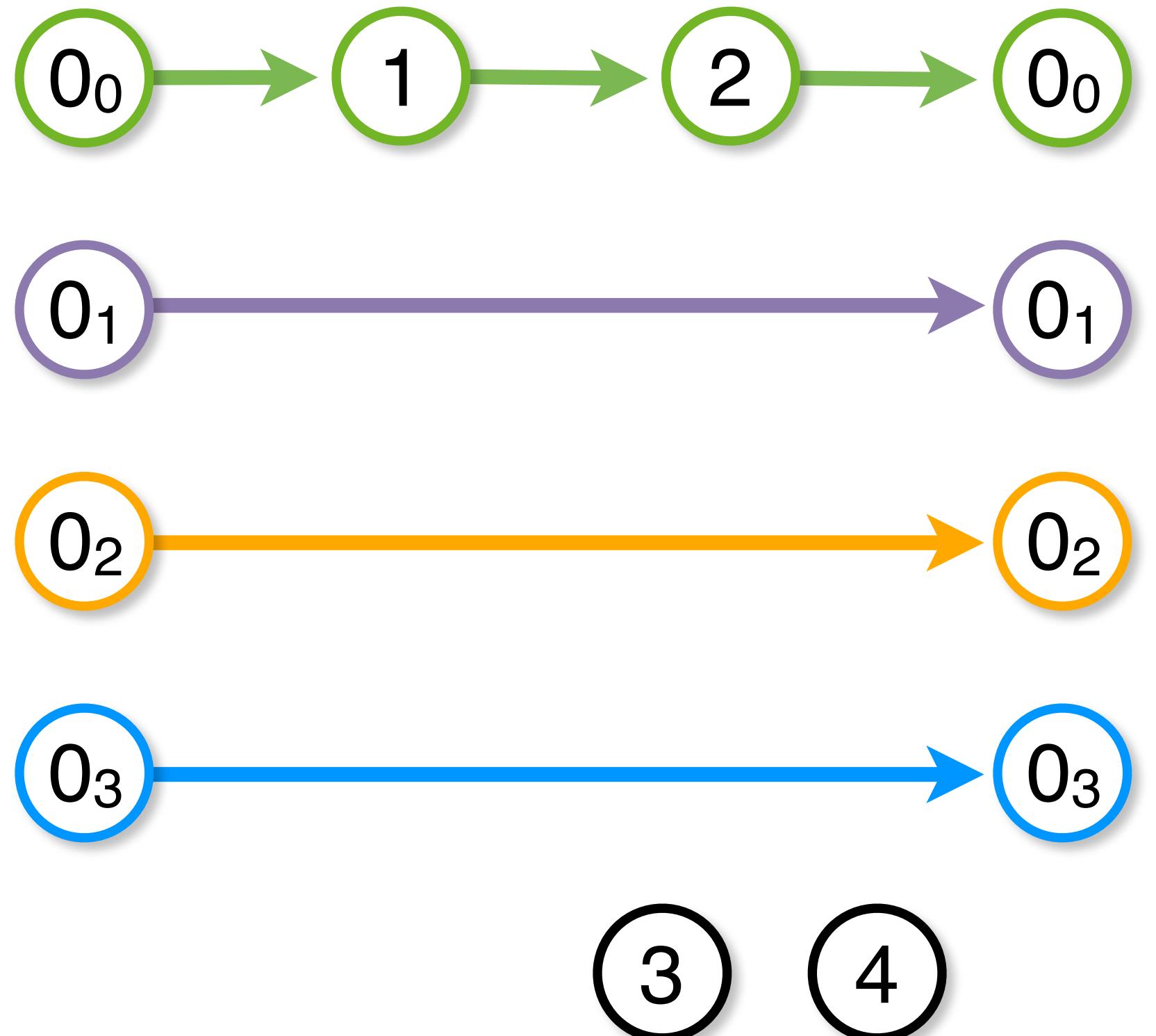
## ► Insert Customers



vehicles = 4

# A Local Search Model

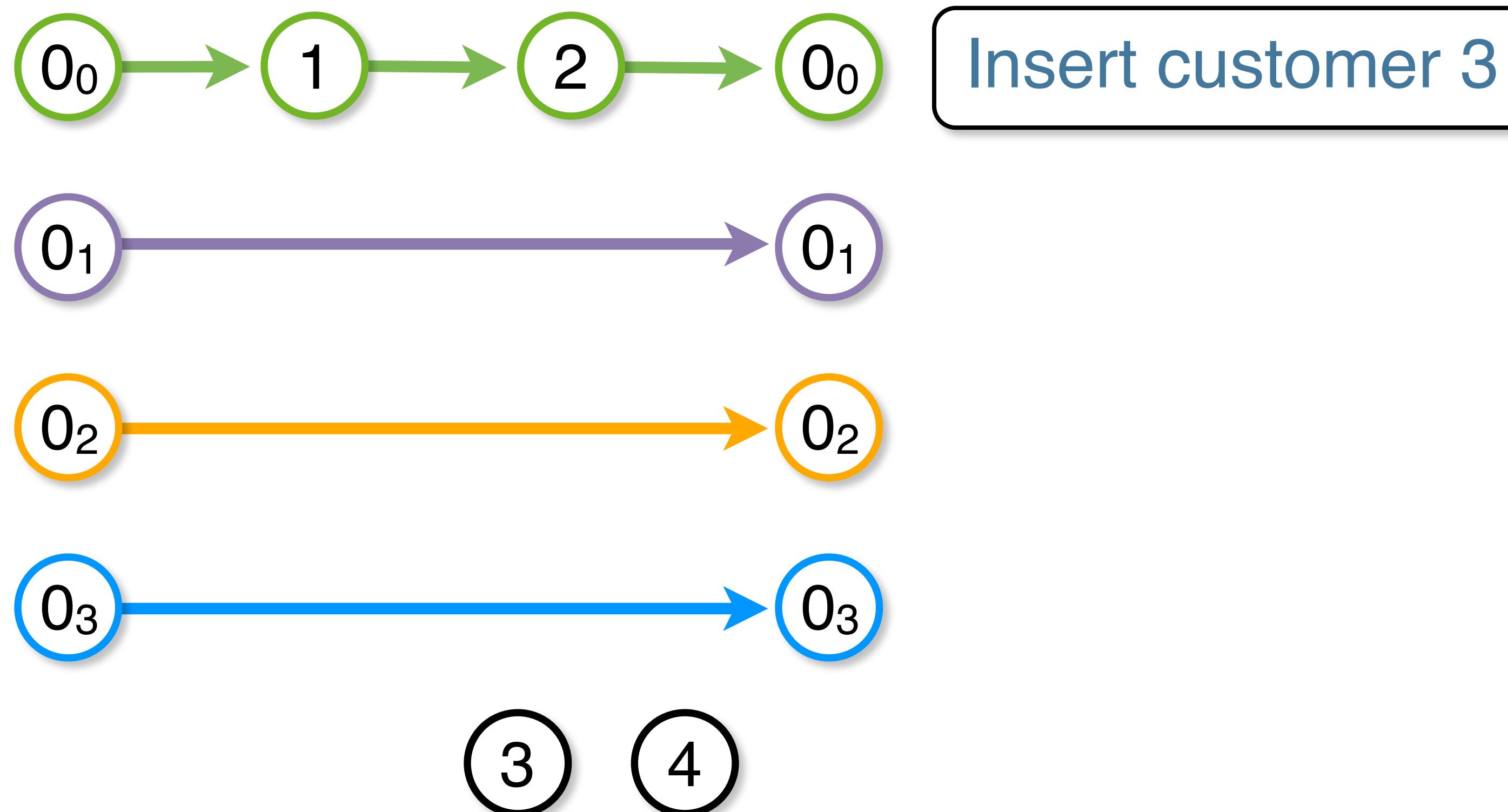
## ► Insert Customers



vehicles = 4

# A Local Search Model

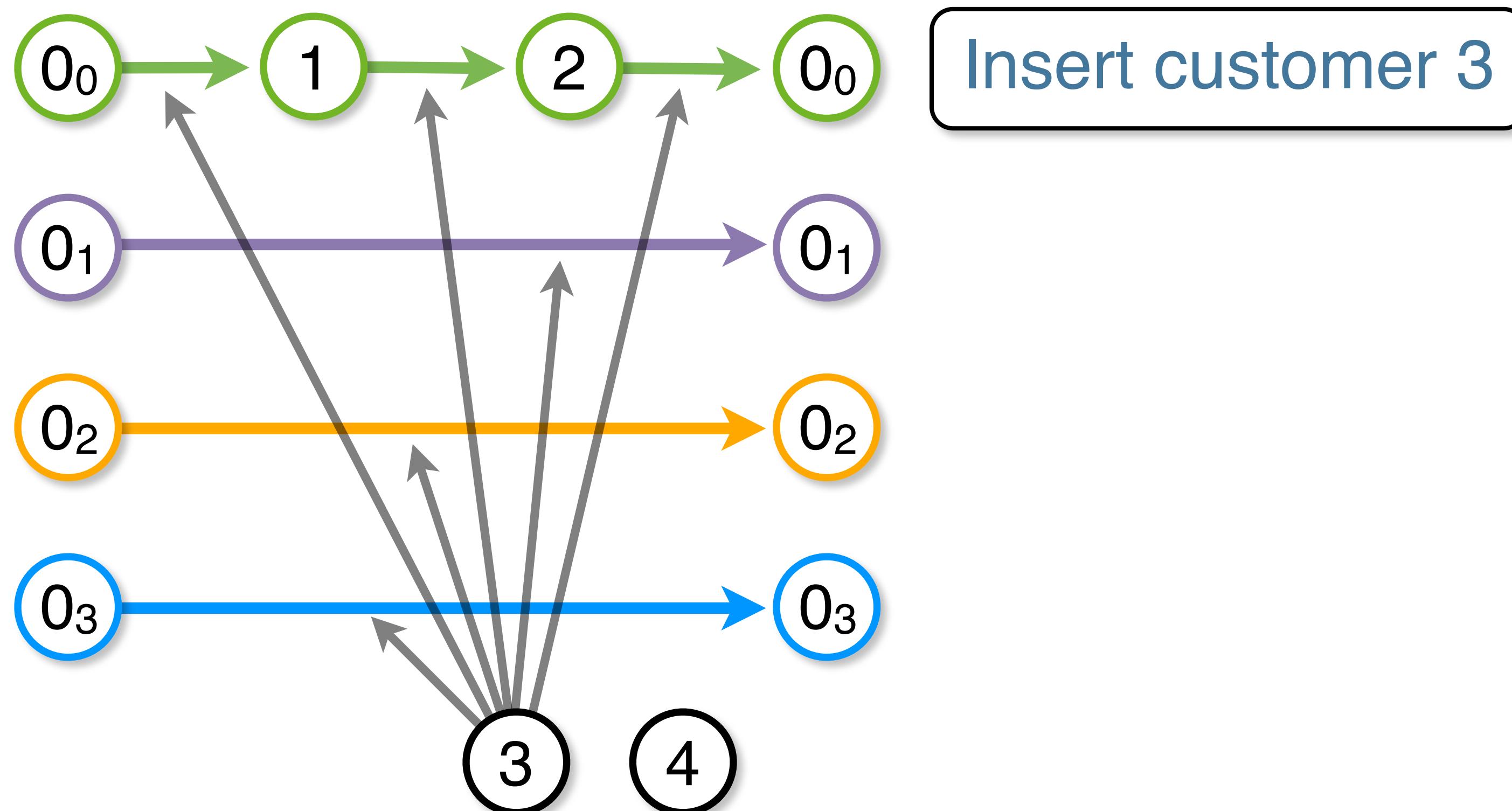
## ► Insert Customers



vehicles = 4

# A Local Search Model

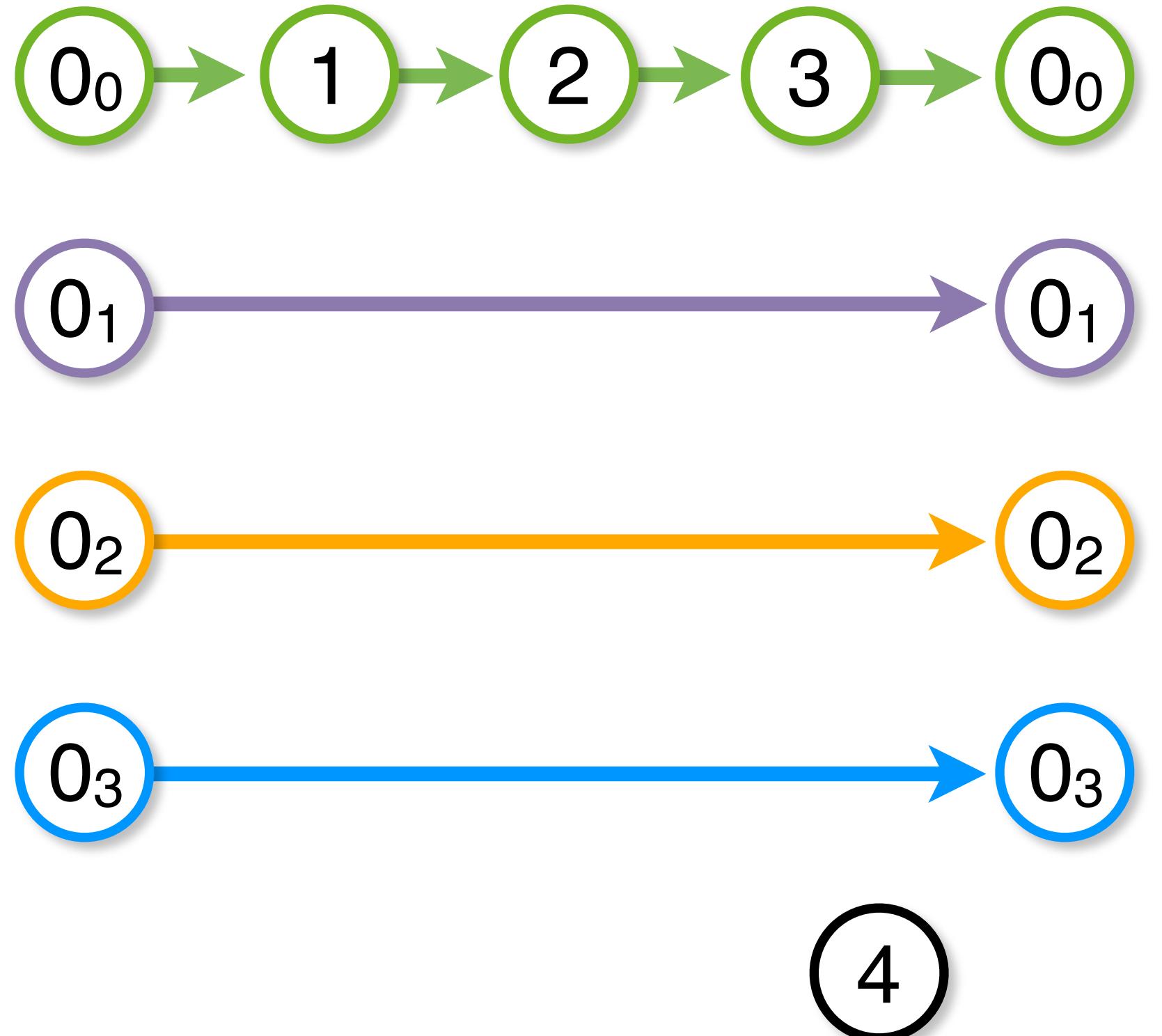
## ► Insert Customers



vehicles = 4

# A Local Search Model

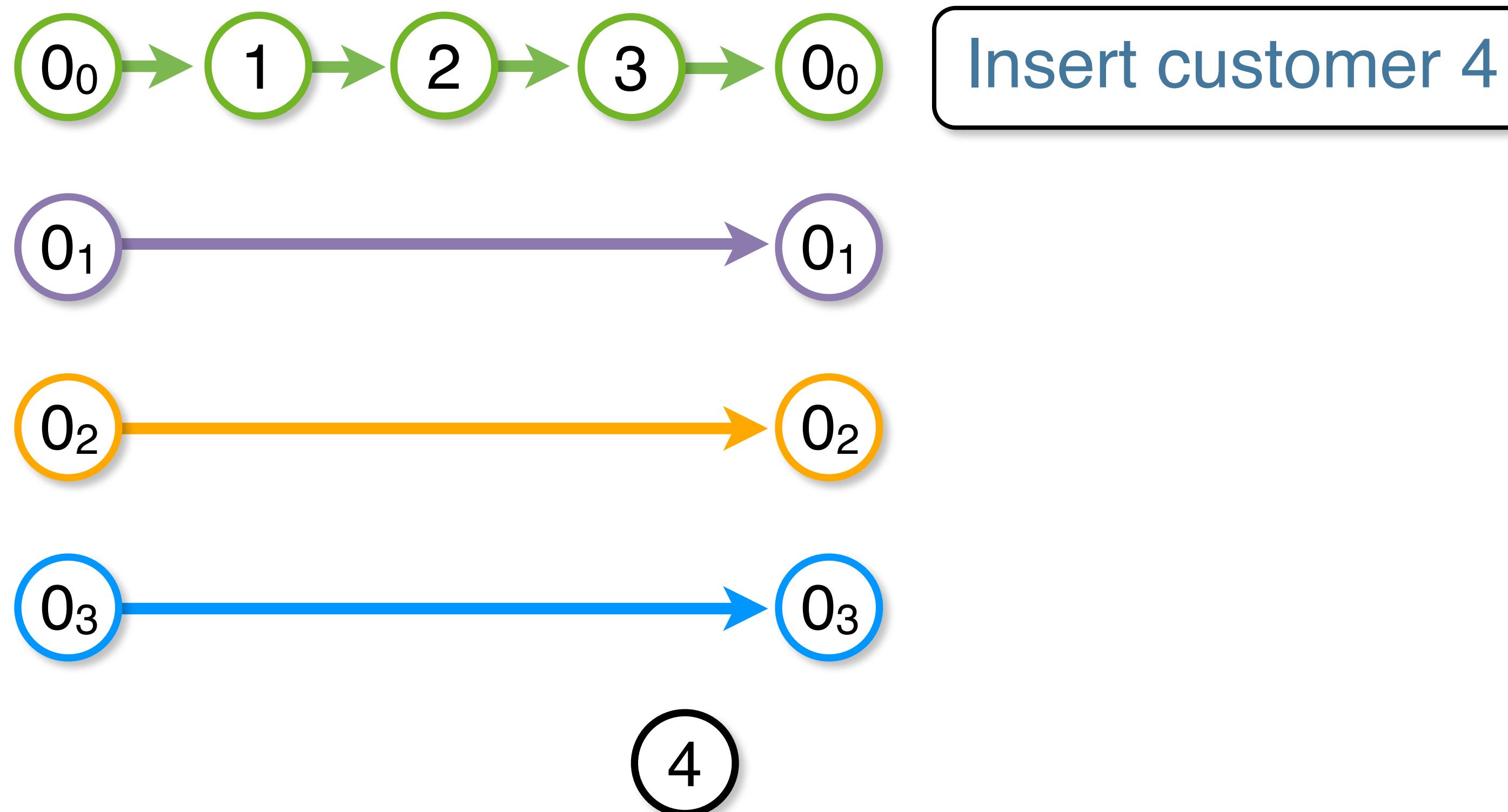
## ► Insert Customers



vehicles = 4

# A Local Search Model

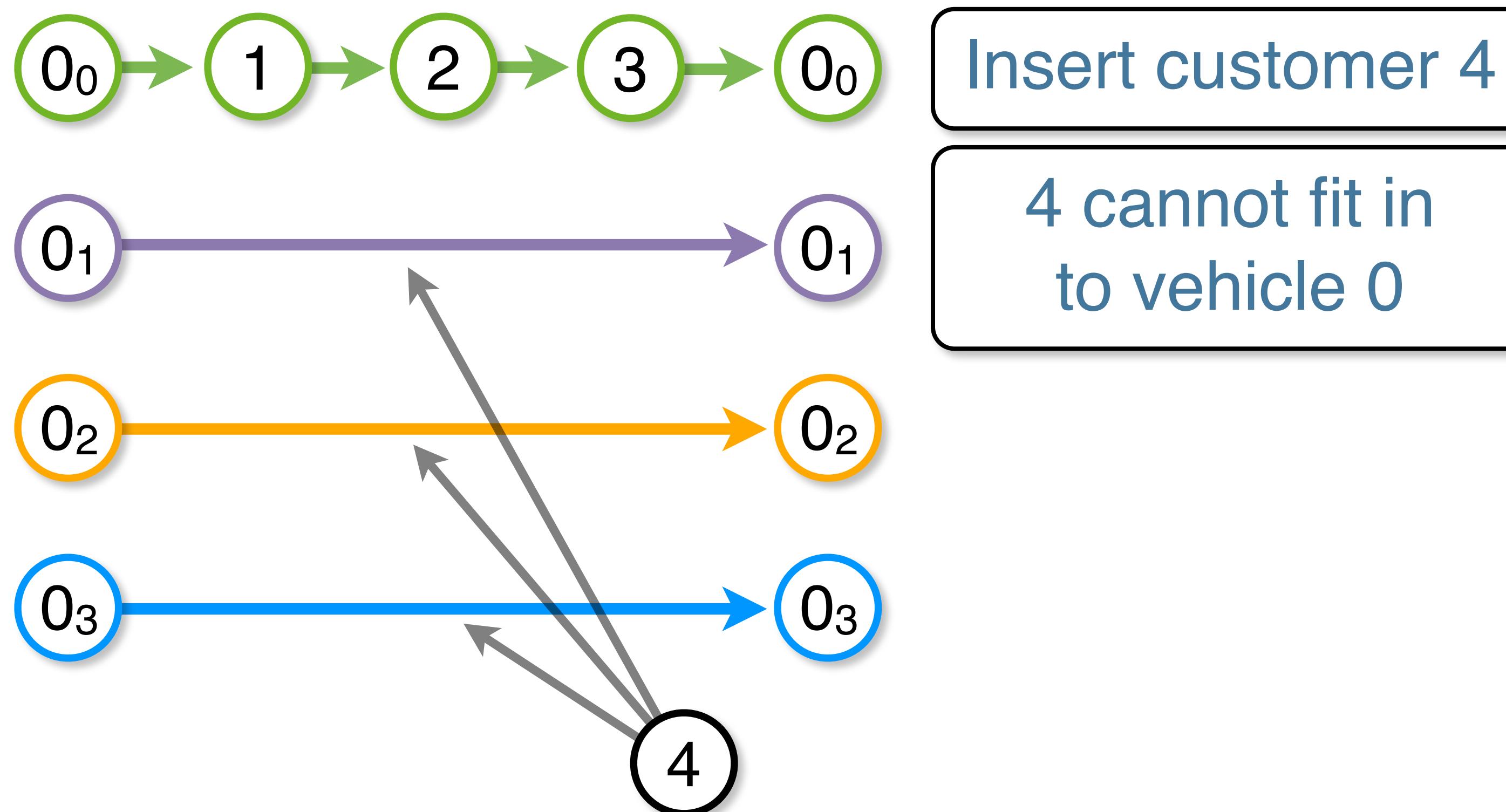
## ► Insert Customers



vehicles = 4

# A Local Search Model

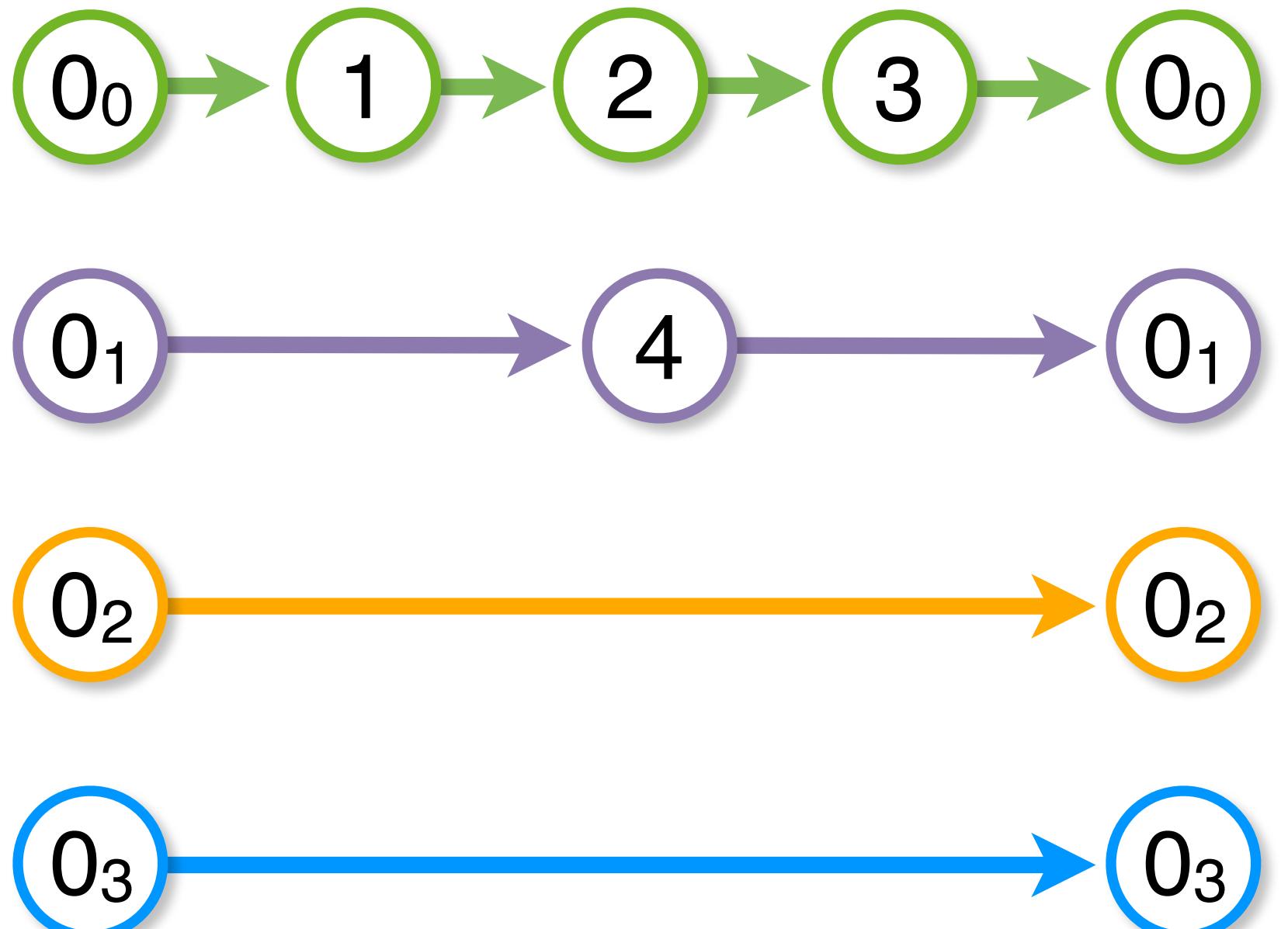
## ► Insert Customers



vehicles = 4

# A Local Search Model

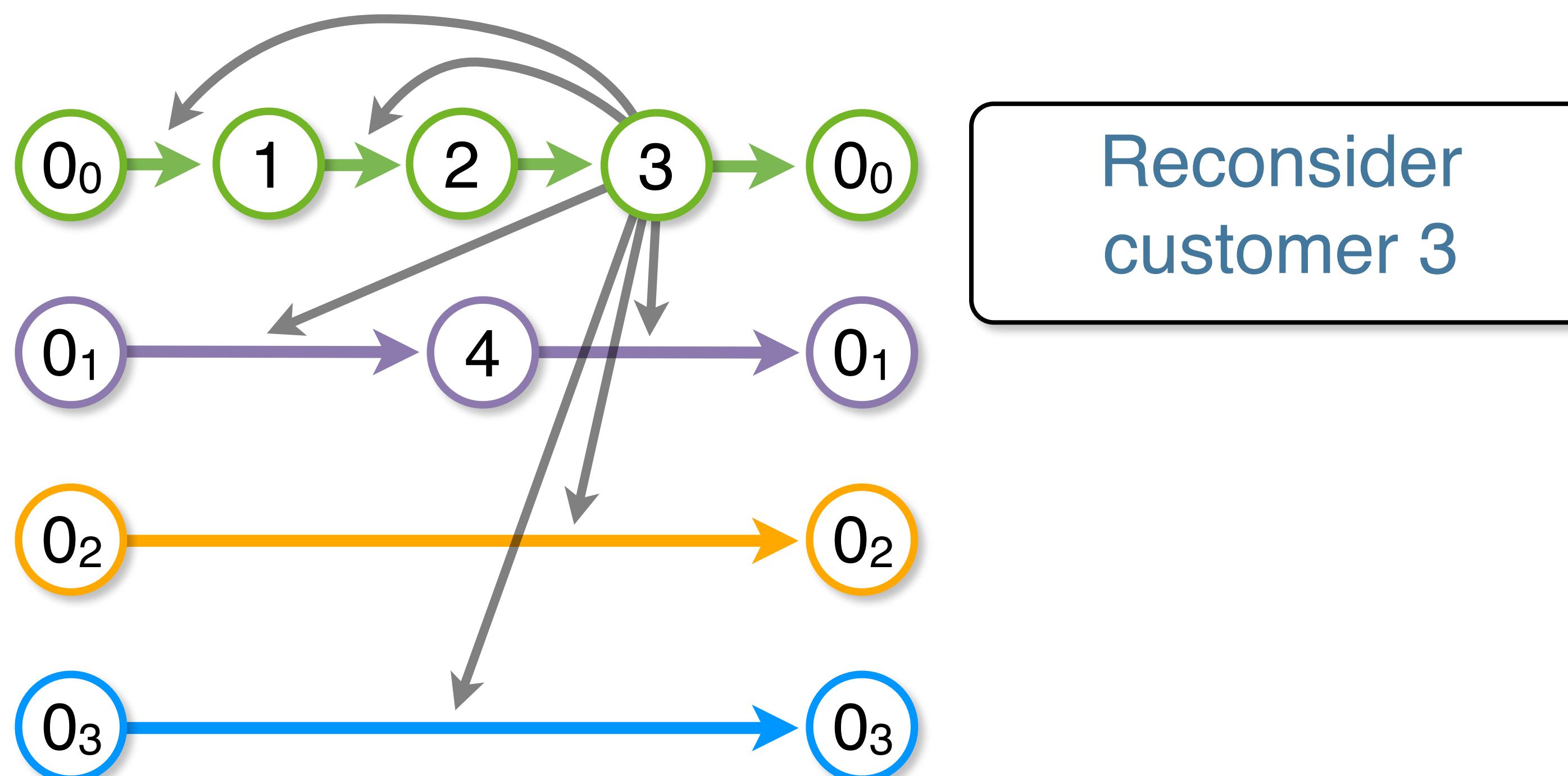
## ► Insert Customers



vehicles = 4

# A Local Search Model

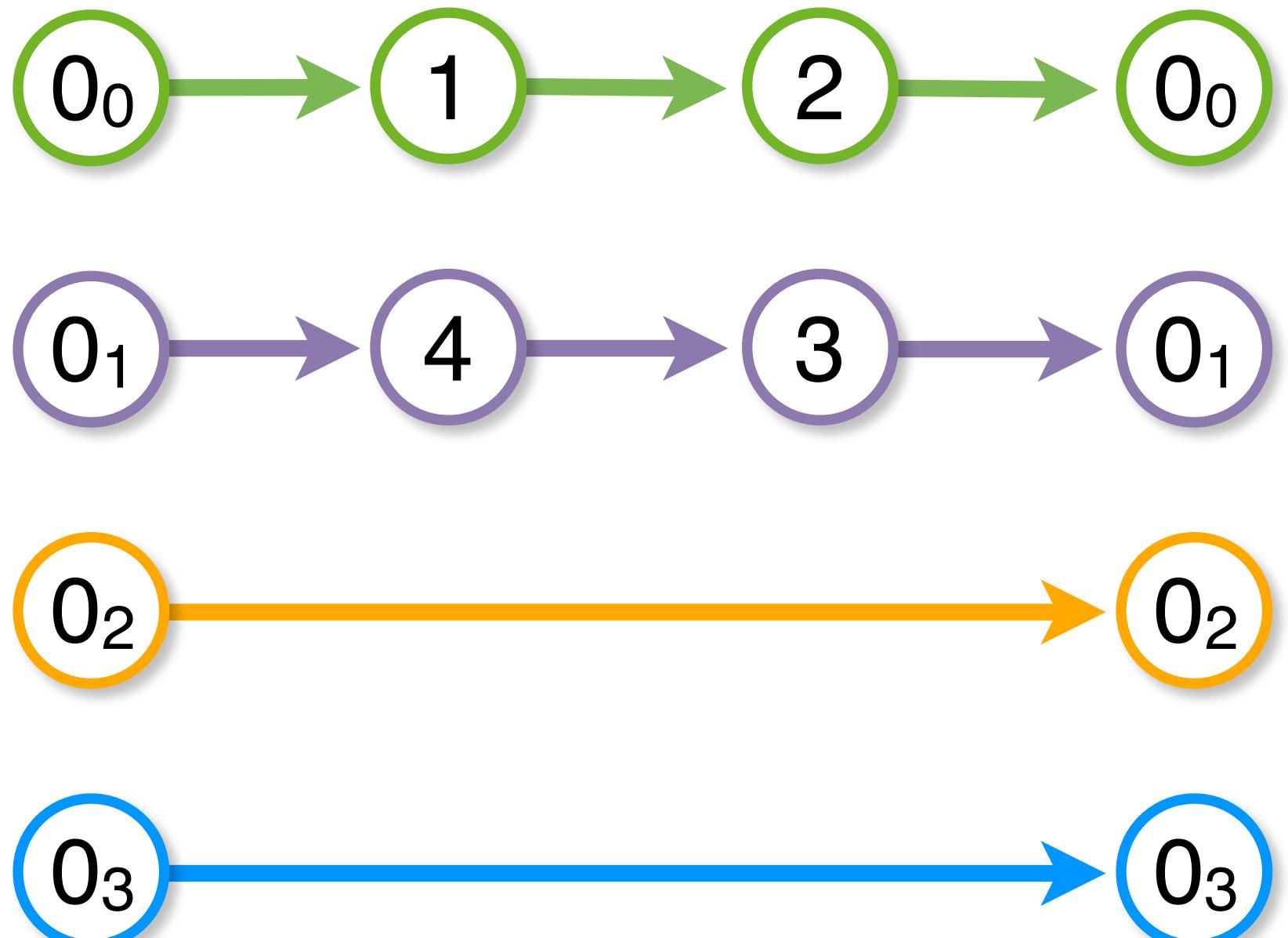
## ► Insert Customers



vehicles = 4

# A Local Search Model

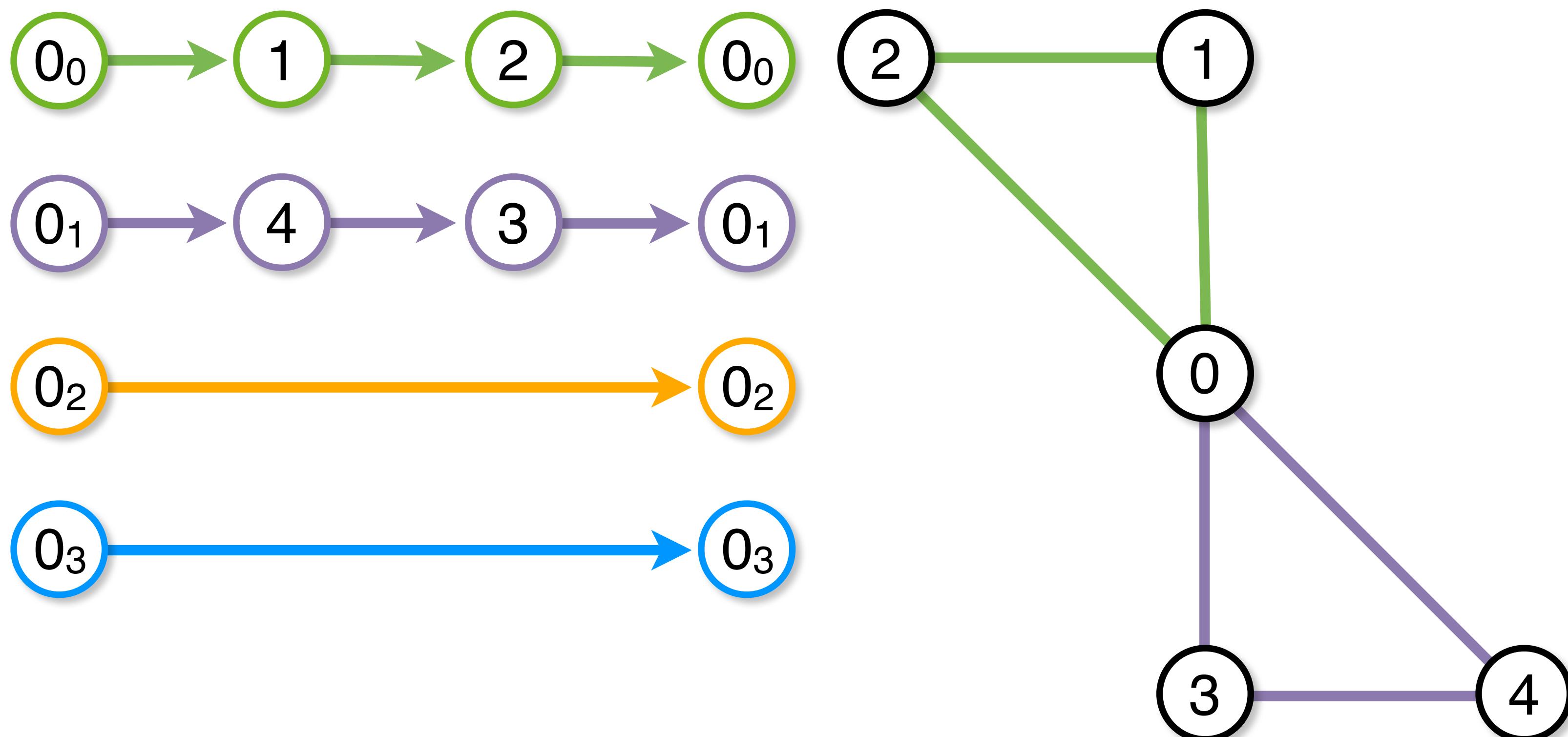
## ► Insert Customers



**vehicles = 4**

# A Local Search Model

## ► Insert Customers



vehicles = 4

# A Two-Stage Decomposition

- All of these methods consider the routing and customer assignment simultaneously
  - but we could break these into two steps

# A Two-Stage Decomposition

- All of these methods consider the routing and customer assignment simultaneously
    - but we could break these into two steps
1. Assign the customers to vehicles
    - and ensure the capacity is satisfied

# A Two-Stage Decomposition

- All of these methods consider the routing and customer assignment simultaneously
  - but we could break these into two steps
- 1. Assign the customers to vehicles
  - and ensure the capacity is satisfied
- 2. Solve a TSP for each vehicle (by any method)
  - CP
  - Local Search
  - MIP

# A Two-Stage Decomposition

- All of these methods consider the routing and customer assignment simultaneously
  - but we could break these into two steps
- 1. Assign the customers to vehicles
  - and ensure the capacity is satisfied
- 2. Solve a TSP for each vehicle (by any method)
  - CP
  - Local Search
  - MIP
- Decouples capacity constraint and routing objective

# Packing isn't Easy

- Even with all these tips, its still tricky

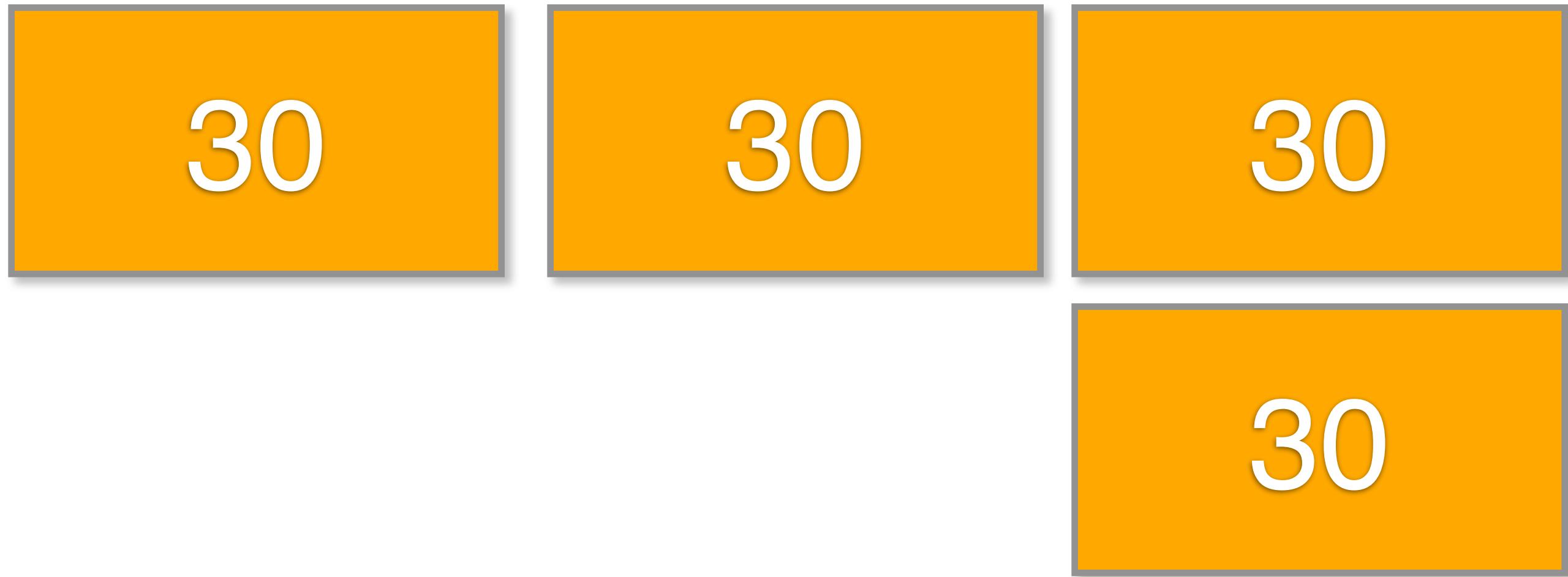
# Packing isn't Easy

- Even with all these tips, its still tricky
- Consider
  - 4 customers of size 30
  - vehicles of capacity 40

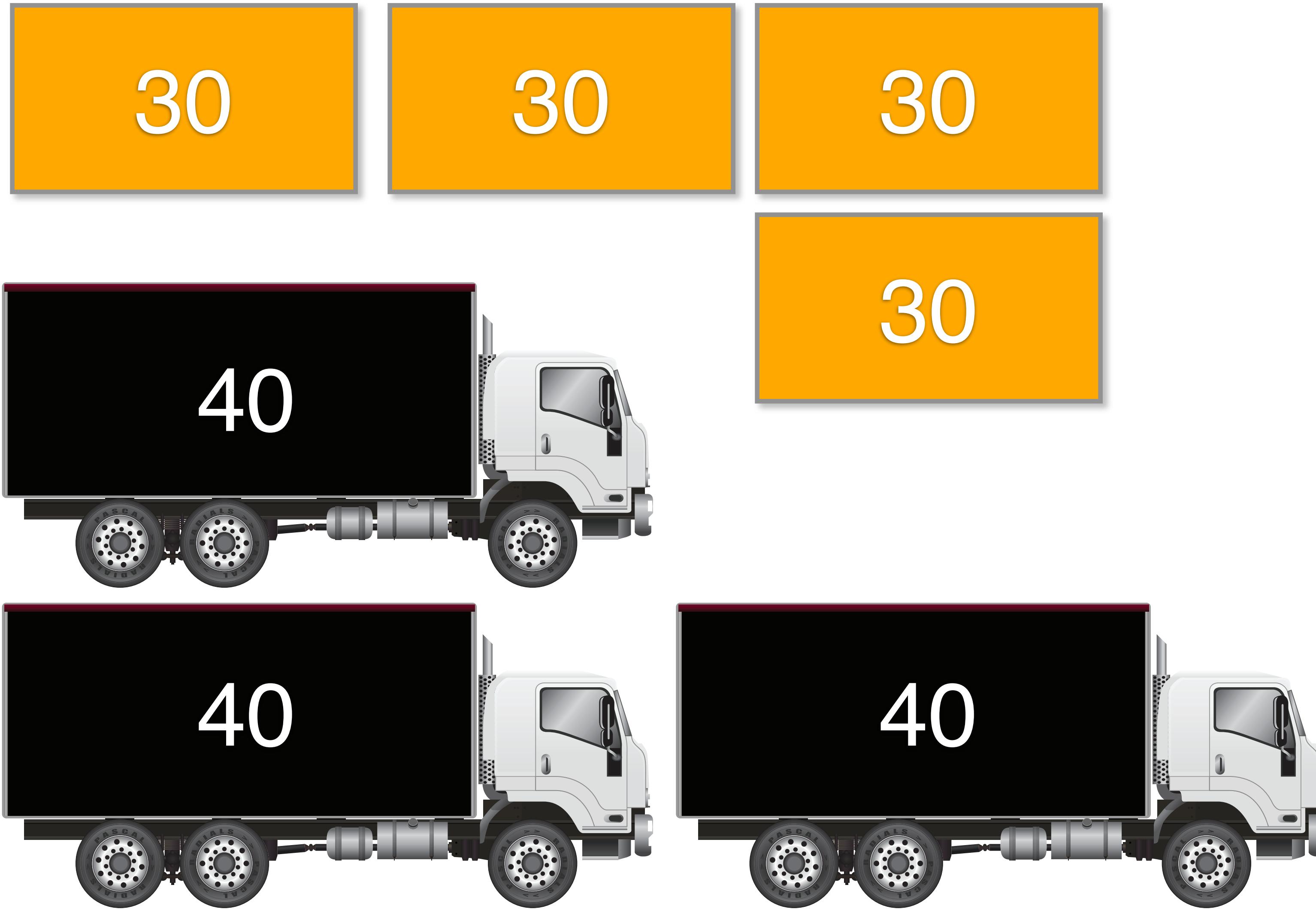
# Packing isn't Easy

- ▶ Even with all these tips, its still tricky
- ▶ Consider
  - 4 customers of size 30
  - vehicles of capacity 40
- ▶ How many vehicles do we need to server these customers?
  - $4 * 30 = 120$  is the total demand
  - total demand / vehicle capacity =  $120 / 40 = 3$
  - Looks like 3 vehicles will do!
  - Let us try it.

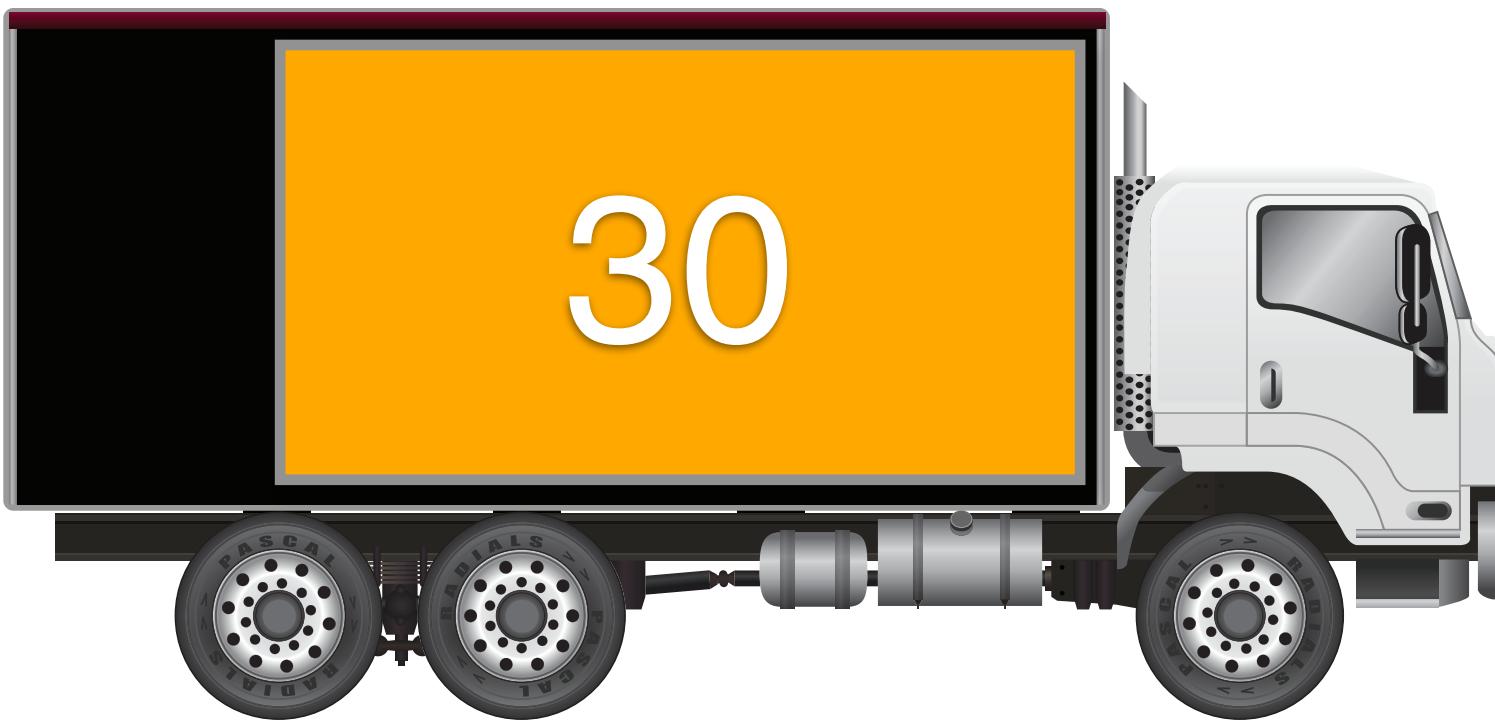
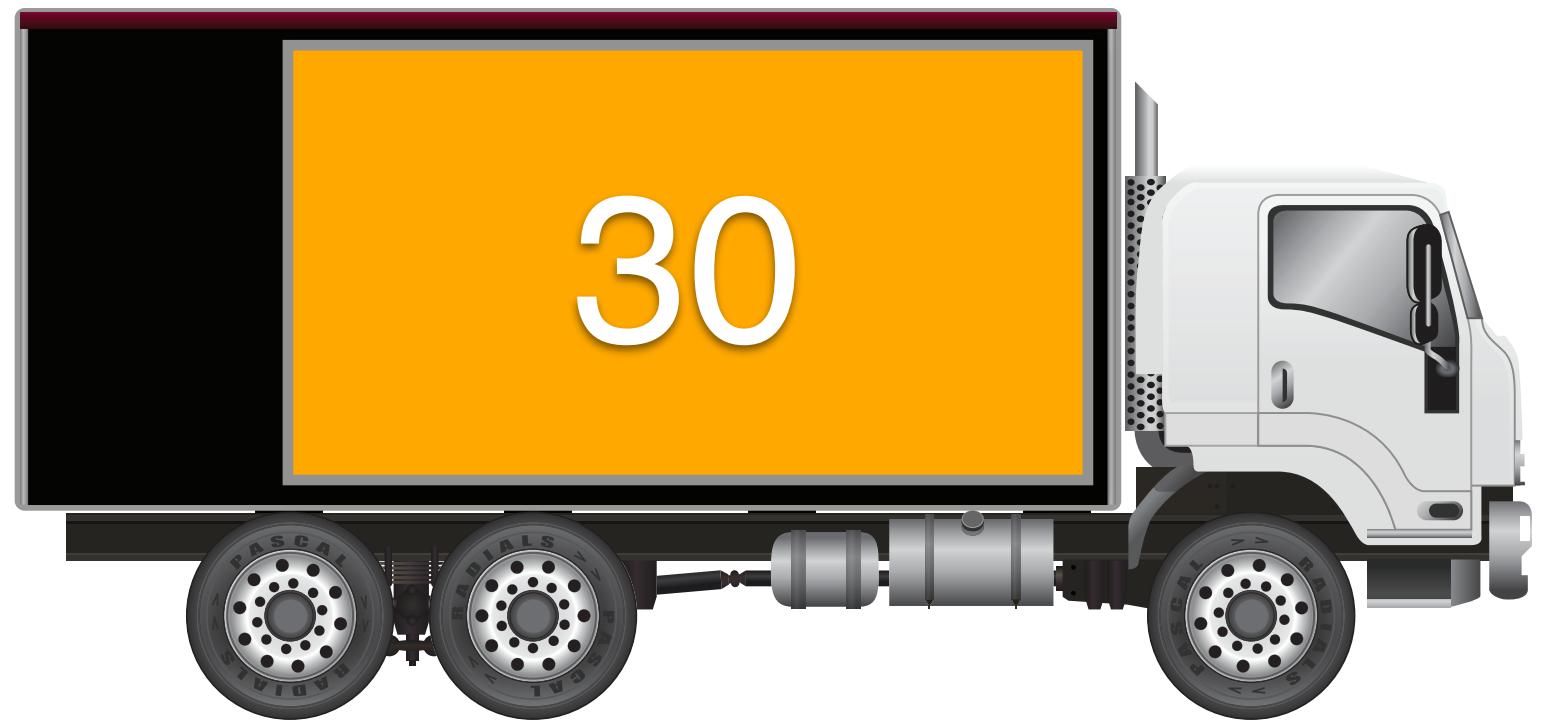
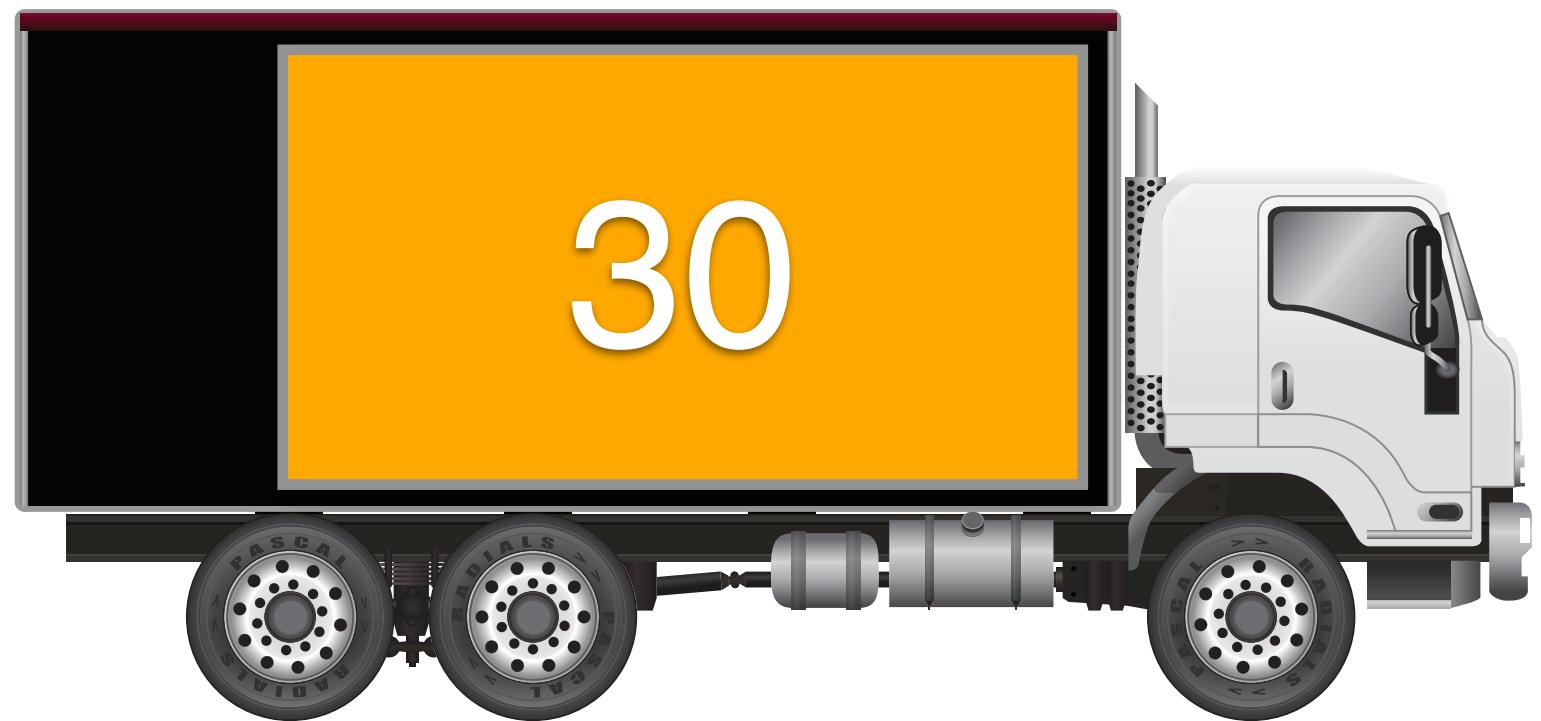
# Packing isn't Easy



# Packing isn't Easy



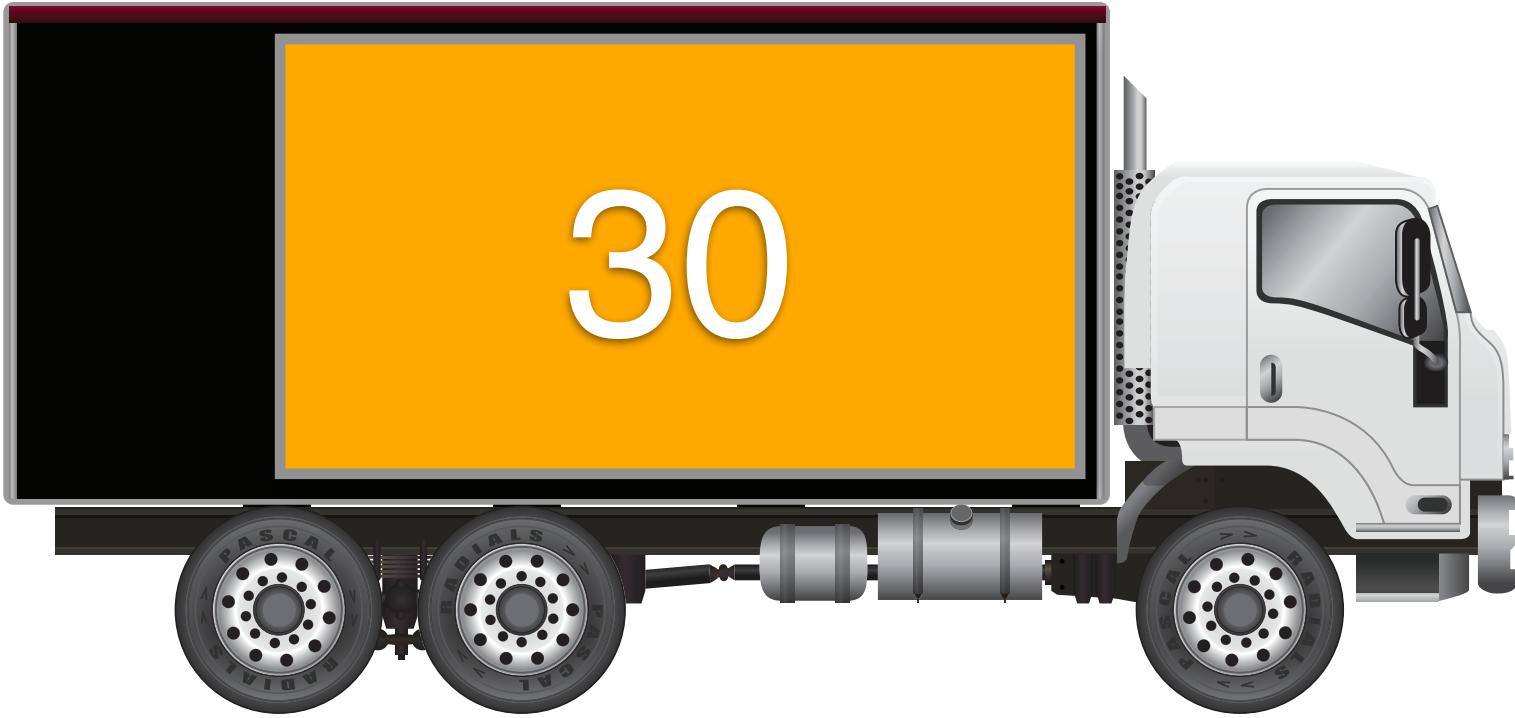
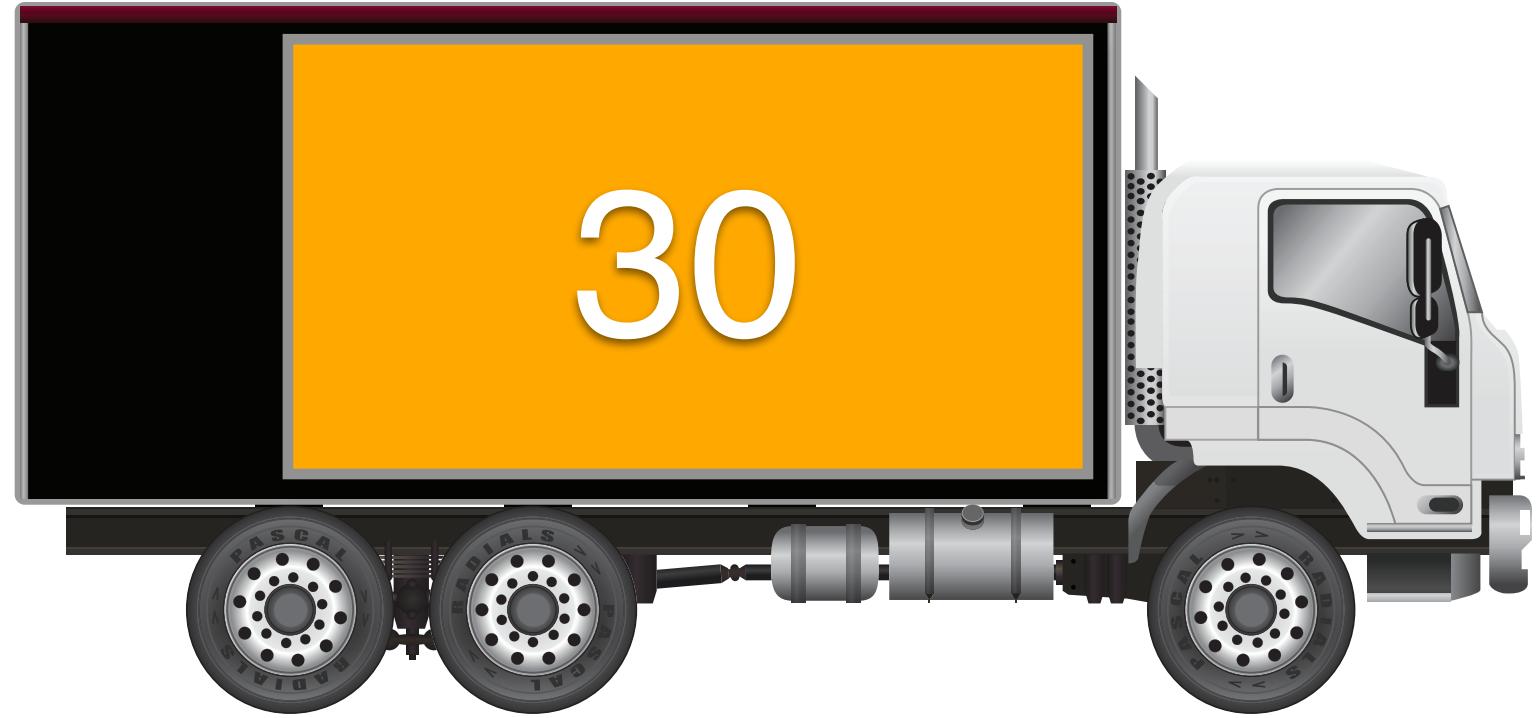
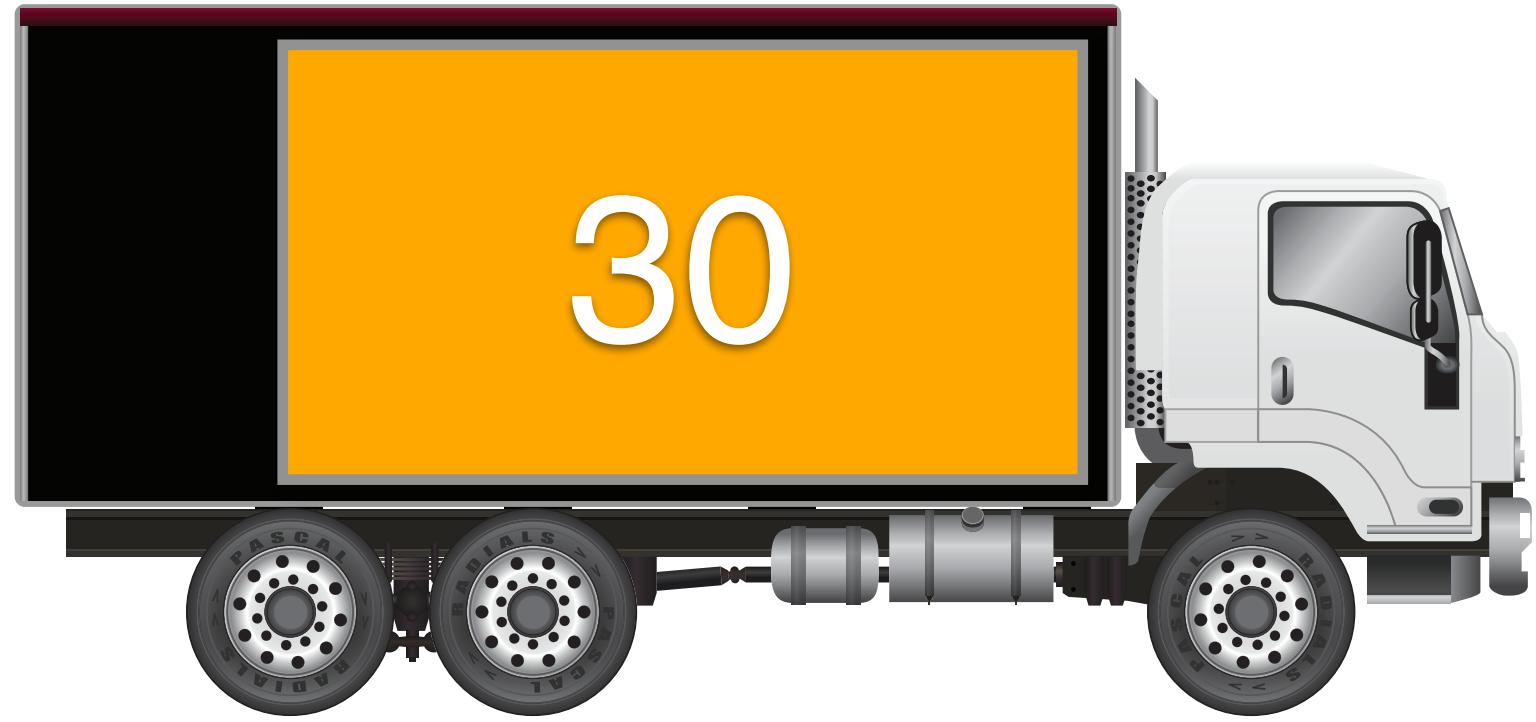
# Packing isn't Easy



# Packing isn't Easy

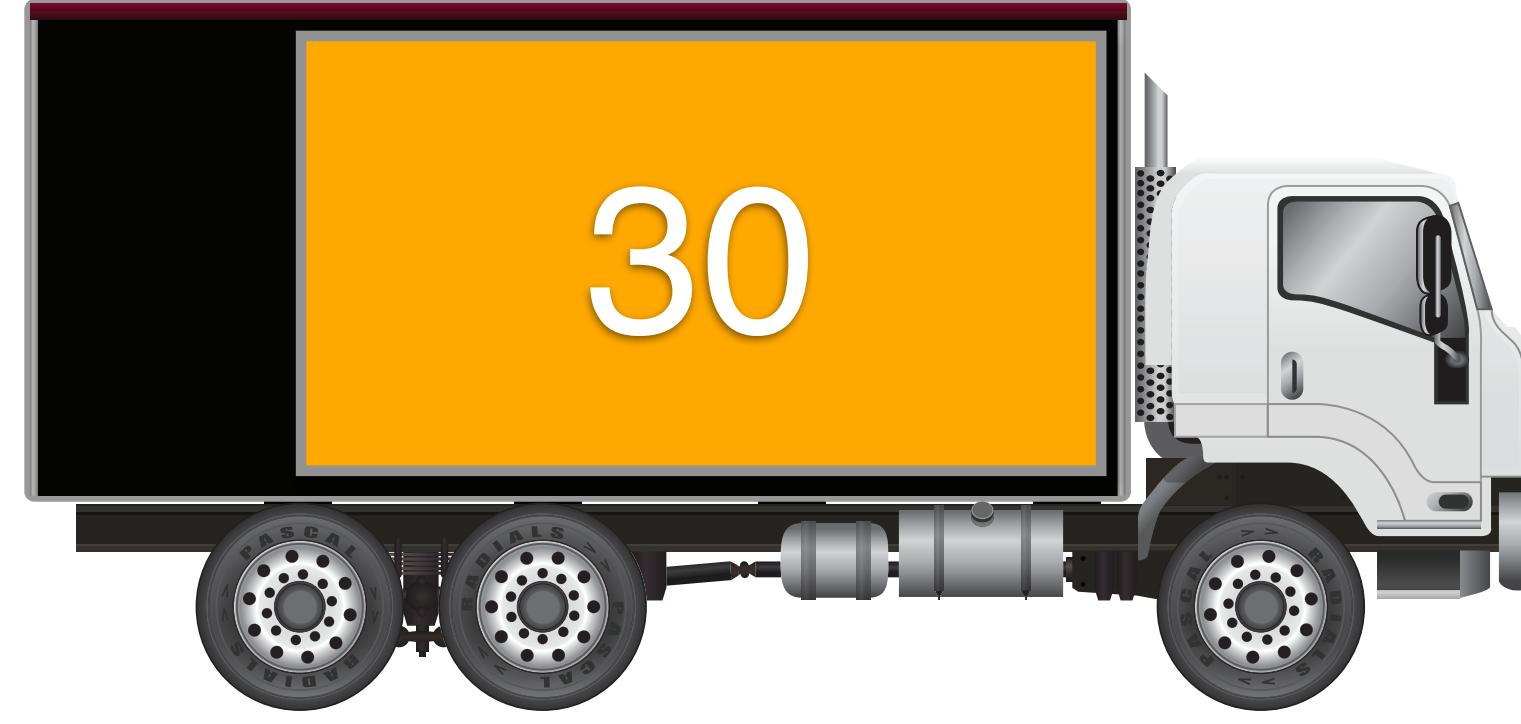
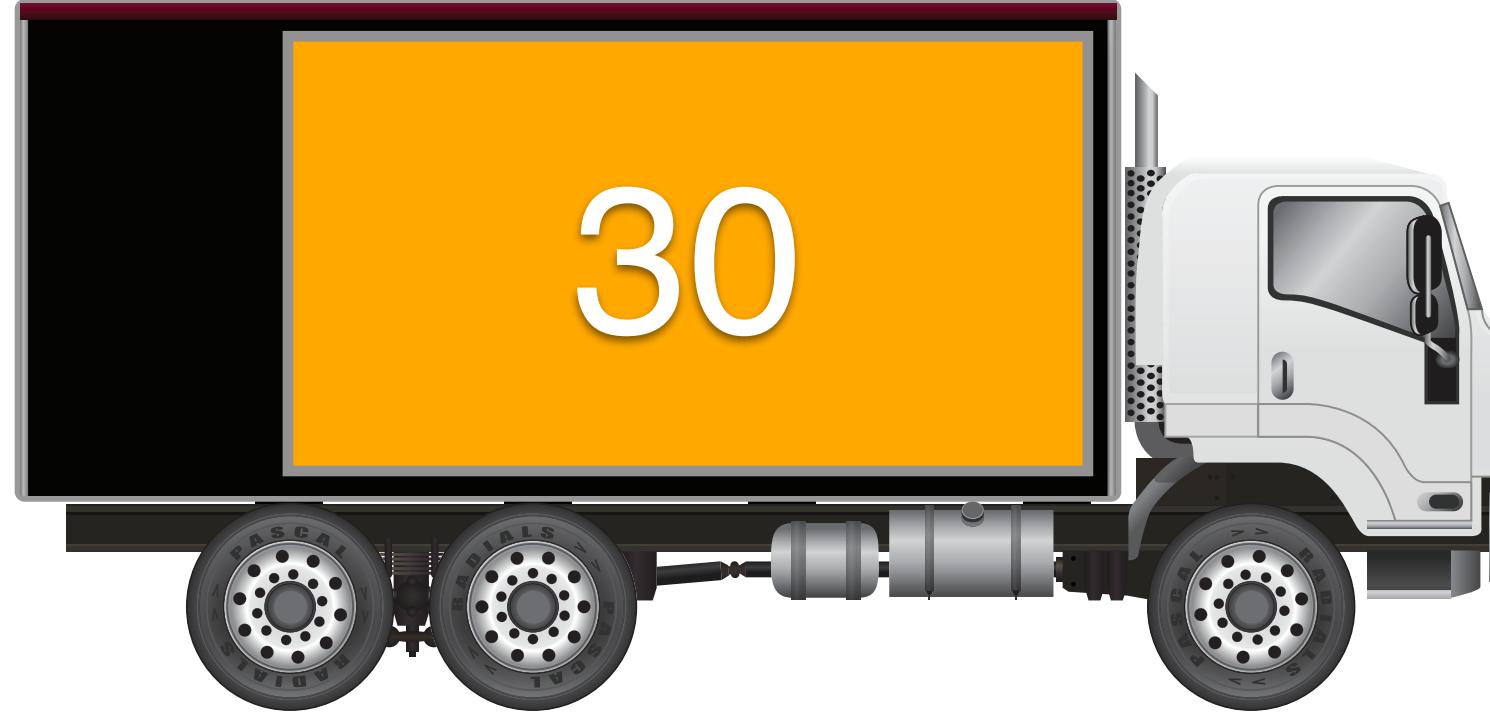
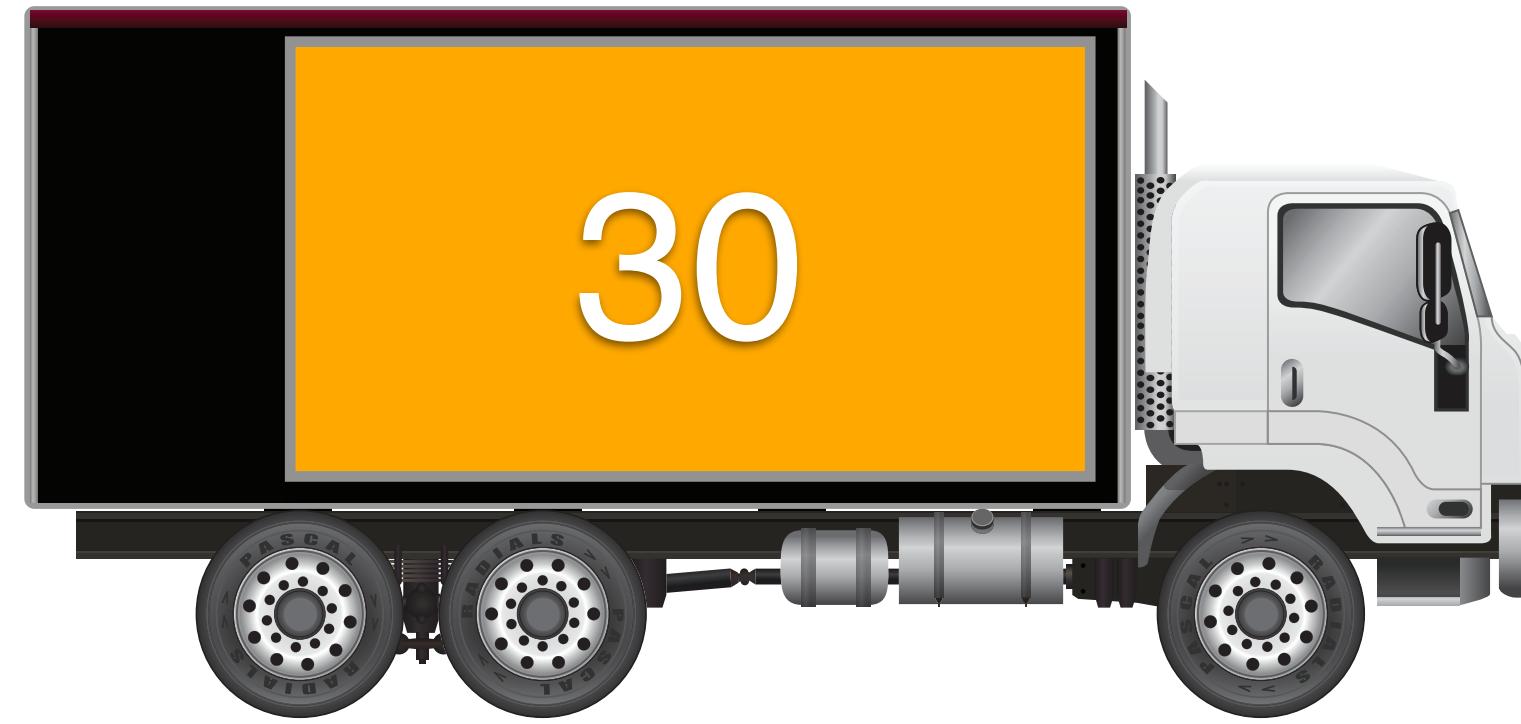
30

Houston, we have a  
problem.



# Packing isn't Easy

Houston, we have a  
problem.



# Packing isn't Easy

- Luckily for you, the number of trucks in the assignment is fixed
  - However, you still need to find out how to pack them...

# Packing isn't Easy

- ▶ Luckily for you, the number of trucks in the assignment is fixed
  - However, you still need to find out how to pack them...
- ▶ This is a well known feasibility problem
  - called multi-knapsack

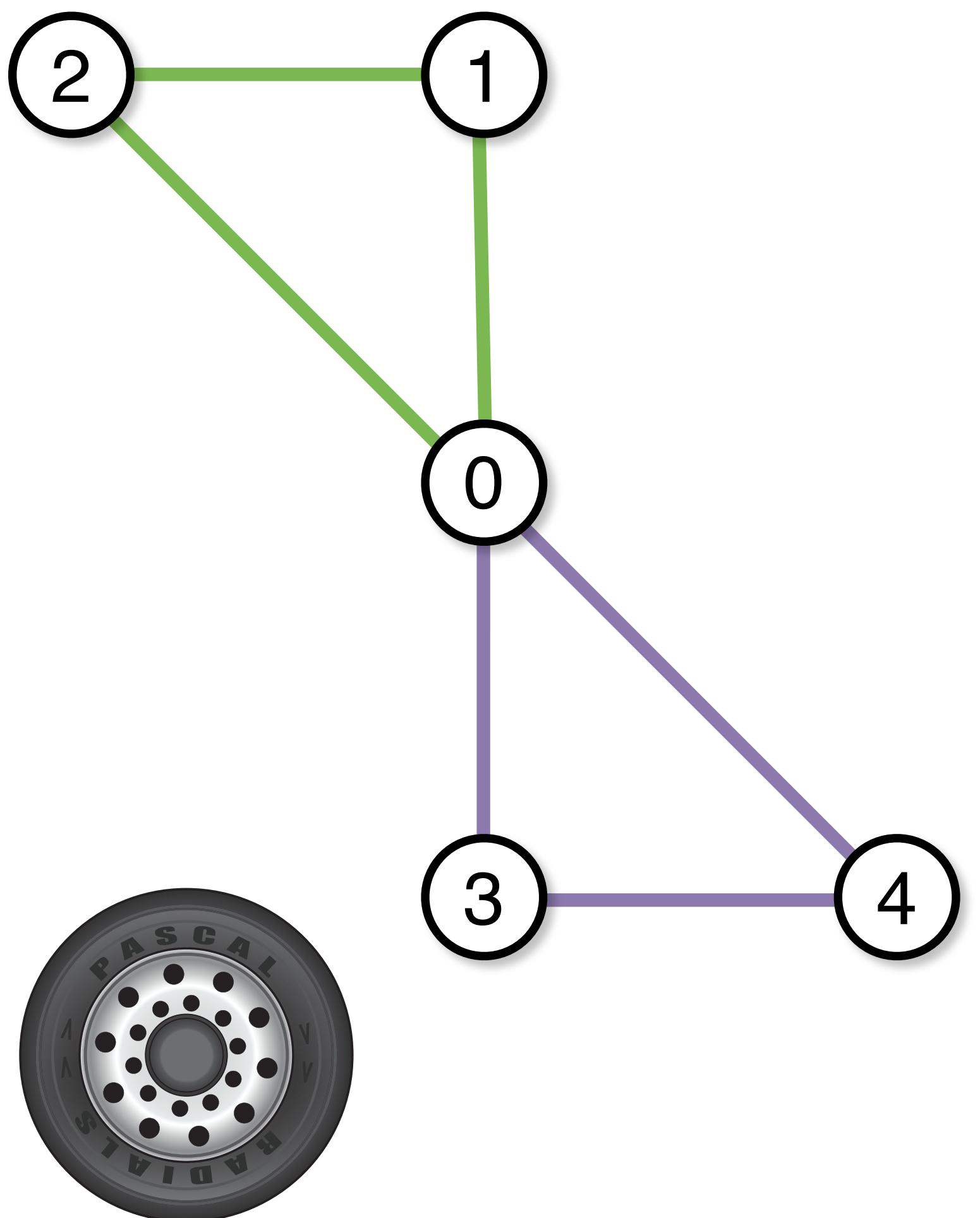
# Packing isn't Easy

- ▶ Luckily for you, the number of trucks in the assignment is fixed
  - However, you still need to find out how to pack them...
- ▶ This is a well known feasibility problem
  - called multi-knapsack
- ▶ Capacitated VRP is multi-knapsack and TSP combined

# Assignment Tips

- ▶ Many approaches can work
  - Start off with the methods you like
- ▶ Reusing your TSP solver may be helpful
- ▶ Symmetries between vehicles
- ▶ *FAST* neighborhood computation

# Have Fun!



# Have Fun!

