

Lecture VII - Library Routing Optimization

Applied Optimization with Julia

Dr. Tobias Vlček
University of Hamburg

Introduction

Learning Objectives

By the end of this lecture, you will be able to:

- **Formulate** a vehicle routing problem as a mathematical model
- **Explain** how subtours arise and how to prevent them
- **Assess** why large instances are hard to solve exactly
- **Choose** between exact methods and heuristics in practice

Central Libraries

Question: Does anyone have an idea what a central library is?

Central Libraries

- **Book Delivery to Libraries** in Germany
- They supply **all local libraries** within the same state
- Complex, as **number of libraries** per state can be large
- Books and media in the libraries **change often**
- Customers can **request books** from other libraries

Structure of the Deliveries

- For delivery, the central library has **several employees and cars**
- Local libraries differ in size, some receive **more items**
- Items are **collected as well** during the tours¹
- They are **transported back** to the central library

Potential Decisions

Question: What decisions can the library make for tours?

- Subdivide set of libraries into **several ordered tours**
- Decide in **which order to visit** the libraries
- Evaluate **which car to use** for each of the tours
- Decide **which driver** to assign to each of the tours

Impact of the Decisions

Question: What is the impact of the decisions?

- Longer driving **increases the footprint** of the deliveries
- Suboptimal tours can lead to **unnecessary costs**
- Fuel, personnel, and repairs **are increased**
- Unhappy customers due to **waiting times** on books

Have you heard of
this problem before?

Problem Structure

Objective

Question: What could be the objective for central libraries?

- **Lowering costs** through improved tours
- Improvement of their **footprint** through shorter tours
- **Faster fulfillment** of the deliveries

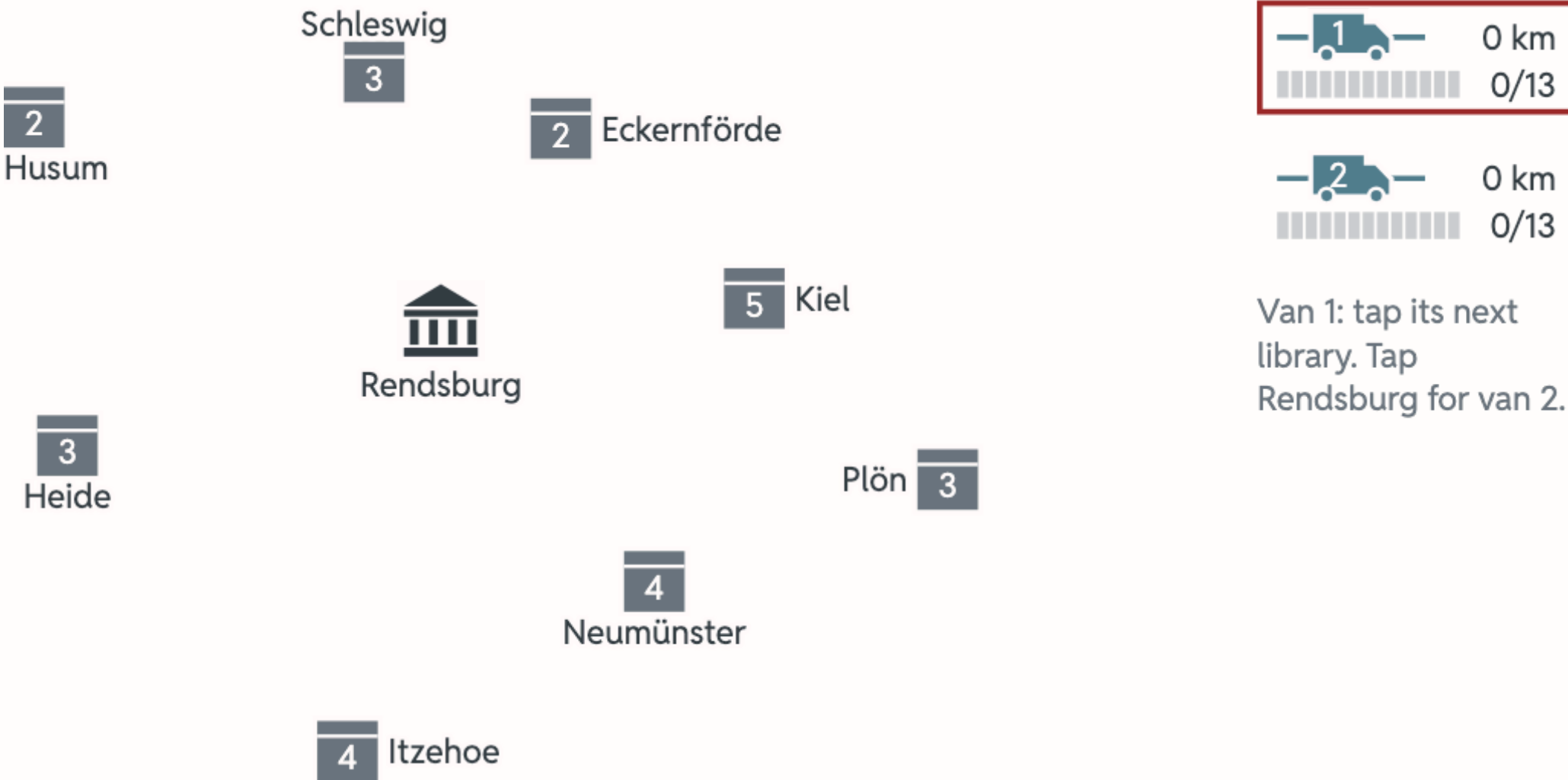
Modelling

Question: What could we try to model?

Minimization of the travel time while **supplying all libraries** in the state and respecting **the vehicle capacities and driving time restrictions**.

Your Turn

The central library in Rendsburg delivers book crates to 8 libraries with two vans of 13 crates each. Tap the libraries in the order a van visits them, tap Rendsburg to switch vans, and tap a library again to take it out. Drive as few km as possible.



You 0 km

8 libraries still wait for books

Optimize

Reset

New

Capacitated Vehicle Routing (CVRP)

Vehicle Routing Problem

- CVRP is a **subproblem**
- Main problem is **Vehicle Routing Problem (VRP)**
- Problem class about **designing routes for vehicle fleets**

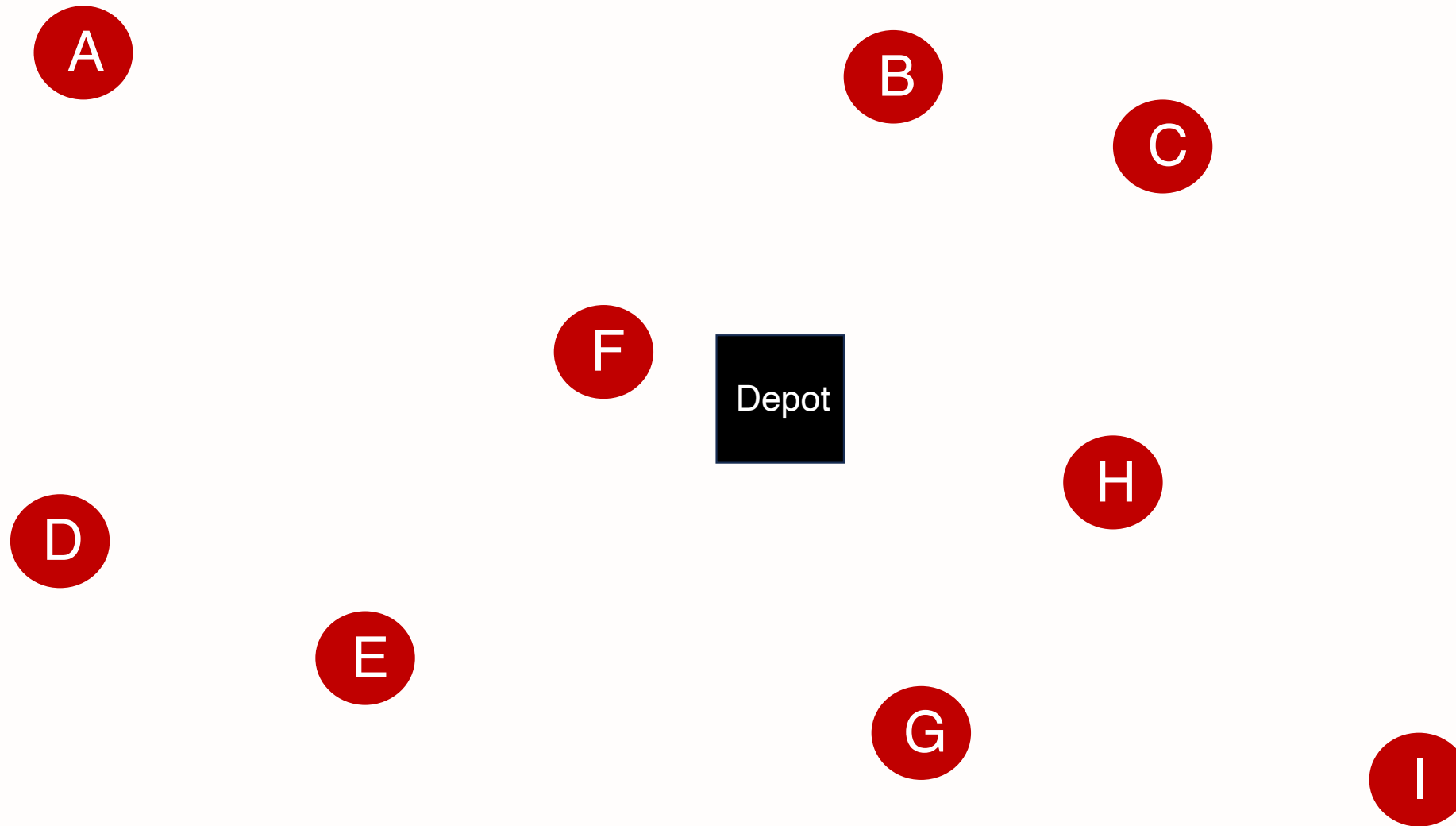
 There are many variants of the VRP! E.g., with time windows, periodic deliveries, allowing for pickups or deliveries, **and much more!**

Let's visualize
the problem!

Problem Visualization

Basic Problem Setting

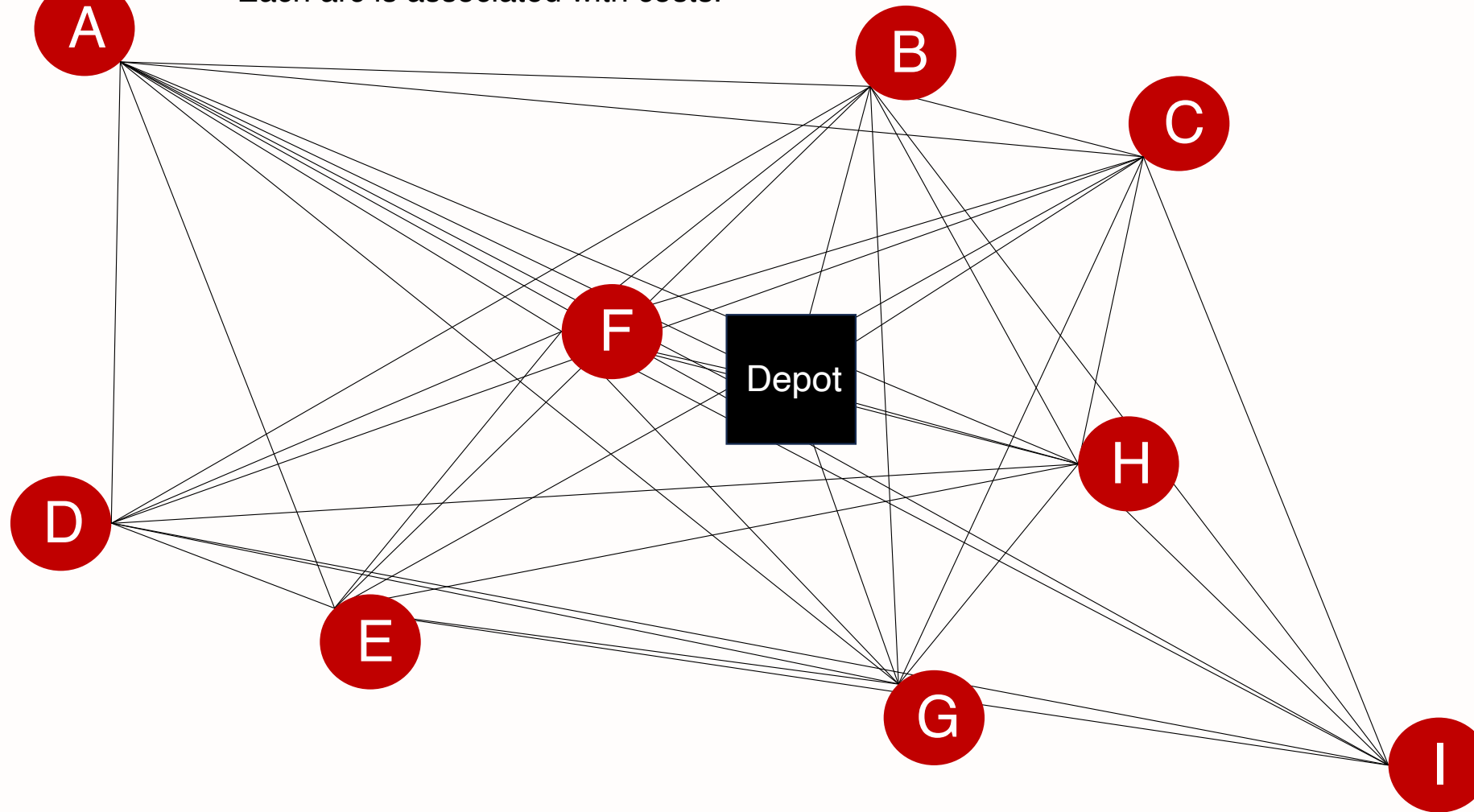
Each customer has a certain demand.



Basic Setting with Arcs

Each customer has a certain demand.

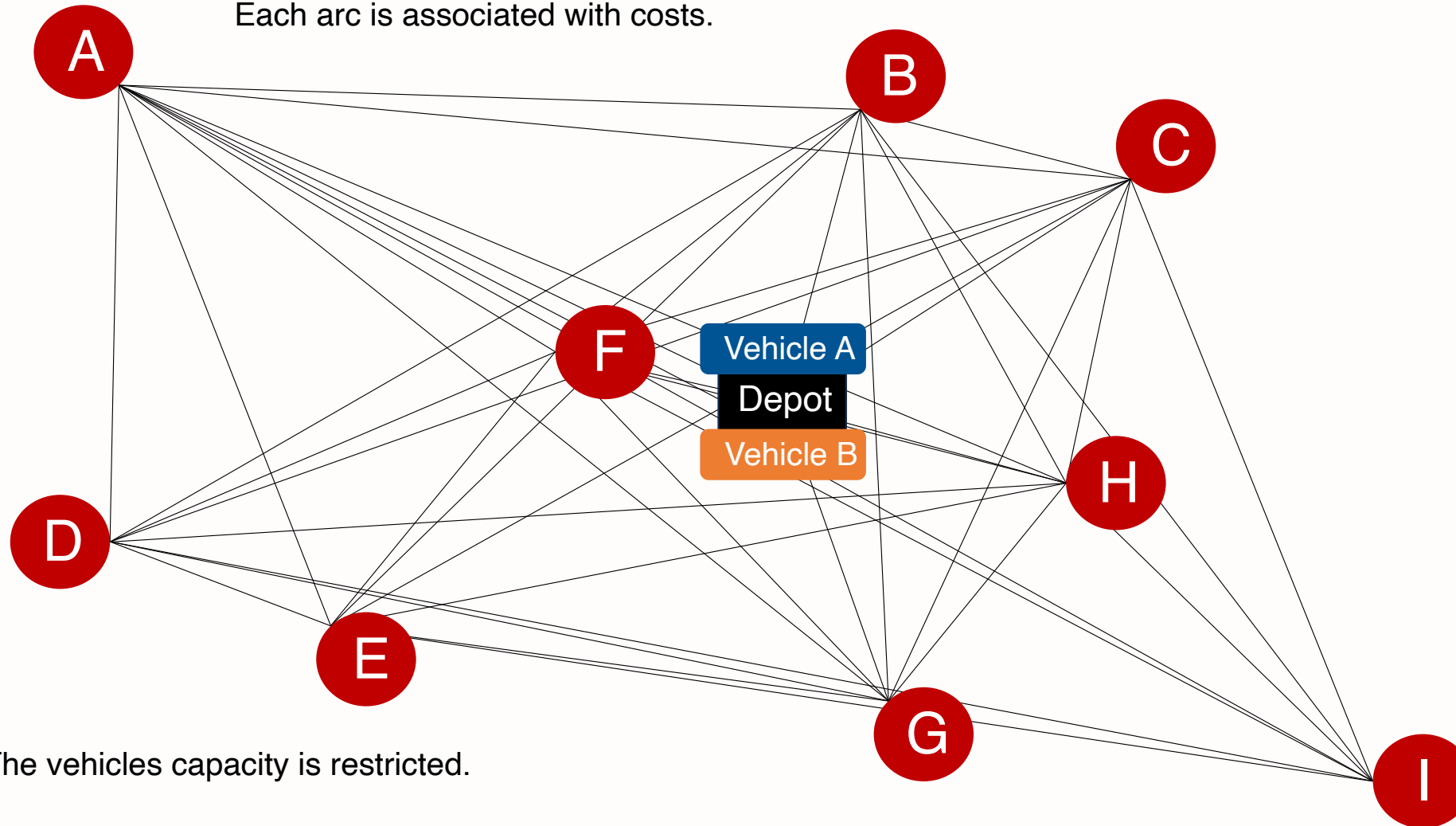
Each arc is associated with costs.



Setting with Vehicles

Each customer has a certain demand.

Each arc is associated with costs.

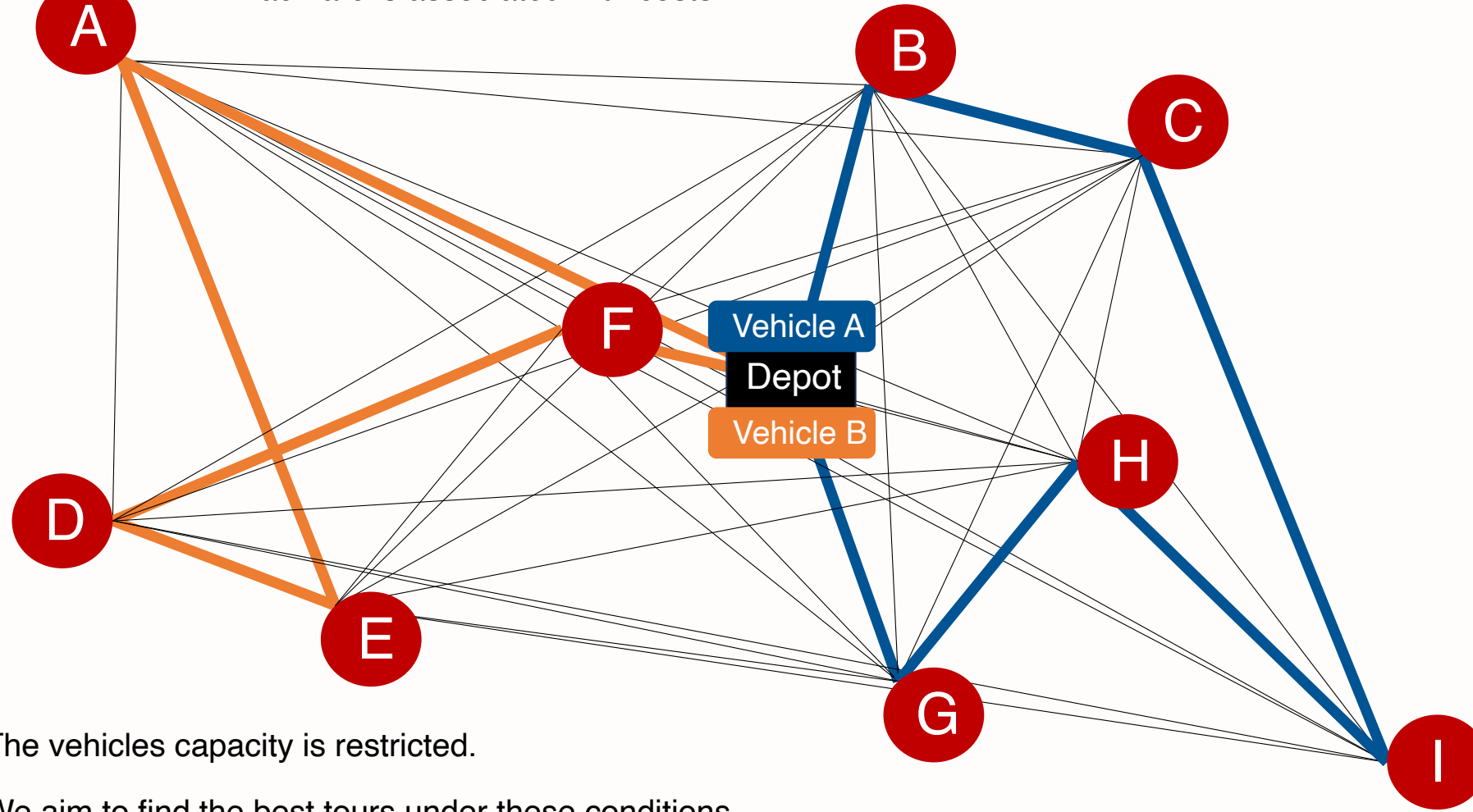


The vehicles capacity is restricted.

Basic Setting with Tours

Each customer has a certain demand.

Each arc is associated with costs.



The vehicles capacity is restricted.

We aim to find the best tours under these conditions.

Problem Structure

Available Sets

Question: What could be the sets here?

- $\mathcal{V}$ - Set of all nodes, index $i \in \{0, 1, 2, \dots, n\}$
- $\mathcal{A}$ - Set of all arcs between the nodes, index $(i, j) \in \mathcal{A}$
- $\mathcal{K}$ - Set of vehicles with identical capacity, index $k \in \mathcal{K}$
- $0 \in \mathcal{V}$ - Depot where the vehicles start

Available Parameters

Question: What are possible parameters?

- b - Capacity per vehicle
- t - Maximal duration of each tour
- d_i - Demand at node i
- $c_{i,j}$ - Travel time on an arc from i to j



t is the maximal duration of each tour, not the travel time on an arc or an index!

Decision Variable(s)?

 We have the following sets:

- All nodes, including the depot, $i \in \mathcal{V}$
- All arcs between the nodes, $(i, j) \in \mathcal{A}$
- The available vehicles, $k \in \mathcal{K}$

 Our objective is to:

Minimize the **total travel time** while supplying **all customers** and adhering to the **vehicle capacities and duration restrictions**.

Question: What could be our decision variable/s?

Decision Variables

- $X_{i,j,k}$ - 1, if k passes between i and j on its tour, 0 otherwise

i Variable Domain

$X_{i,j,k}$ is a **binary variable**, as it can only take values 0 or 1. But most likely, you will already have spotted that!

Question: Why might this make the problem difficult?

Decision Variable/s (again)?

i We have the following sets:

- All nodes, including the depot, $i \in \mathcal{V}$
- All arcs between the nodes, $(i, j) \in \mathcal{A}$
- The available vehicles, $k \in \mathcal{K}$

! Our objective is to:

Minimization of the travel time (or driving distance), while supplying all customers and adhering to the vehicle capacities and duration restrictions. **Hint:** Even with many vehicles, **each arc can maximally be passed once!**

Decision Variables (again)

Question: What could be our decision variable/s?

- $X_{i,j}$ - 1, if the arc between i and j is part of a tour, else 0

⚠ Only possible under certain conditions!

Only possible, if time and capacity constraints are equal for all vehicles!

Model Formulation

Objective Function?

! Our objective is to:

Minimization of the travel time (or driving distance), while supplying all customers and adhering to the vehicle capacities and duration restrictions.

Question: What could be our objective function?

i We need the following variable:

- $X_{i,j}$ - 1, if the arc between i and j is part of a tour, else 0

Objective Function

i We need the following parameters:

- $c_{i,j}$ - travel time on an arc from i to j

$$\text{Minimize} \quad \sum_{(i,j) \in \mathcal{A}} c_{i,j} \times X_{i,j}$$

Question: What does $(i, j) \in \mathcal{A}$ under the sum mean?

From Math to Code

The objective function in **JuMP** looks very similar:

```
1 using JuMP, HiGHS
2
3 # Simple example with 4 nodes
4 nodes = ["depot", "A", "B", "C"]
5 arcs = [(i,j) for i in nodes, j in nodes if i != j]
6 c = Dict{("depot","A") => 10, ("depot","B") => 15, ("depot","C") => 20,
7       ("A","depot") => 10, ("A","B") => 12, ("A","C") => 8,
8       ("B","depot") => 15, ("B","A") => 12, ("B","C") => 5,
9       ("C","depot") => 20, ("C","A") => 8, ("C","B") => 5}
10
11 model = Model(HiGHS.Optimizer)
12 @variable(model, X[arcs], Bin)
13 @objective(model, Min, sum(c[(i,j)] * X[(i,j)] for (i,j) in arcs))
```

i The JuMP syntax `sum(c[(i,j)] * X[(i,j)] for (i,j) in arcs)` directly mirrors our mathematical notation $\sum_{(i,j) \in \mathcal{A}} c_{i,j} \times X_{i,j}!$

Solving the Small Example

Let's add constraints and solve the example with **one vehicle**:

```
1 # Each node has to be entered and left exactly once
2 for n in nodes
3     @constraint(model, sum(X[(i,j)] for (i,j) in arcs if j == n) == 1)
4     @constraint(model, sum(X[(i,j)] for (i,j) in arcs if i == n) == 1)
5 end
6 set_silent(model)
7 optimize!(model)
8 println("Arcs: ", [a for a in arcs if value(X[a]) > 0.5])
9 println("Cost: ", objective_value(model))
```

Arcs: [("A", "depot"), ("depot", "A"), ("C", "B"), ("B", "C")]
Cost: 30.0

A Valid Solution?

The solver chose the arcs depot $\leftrightarrow$ A and B $\leftrightarrow$ C with a cost of 30.

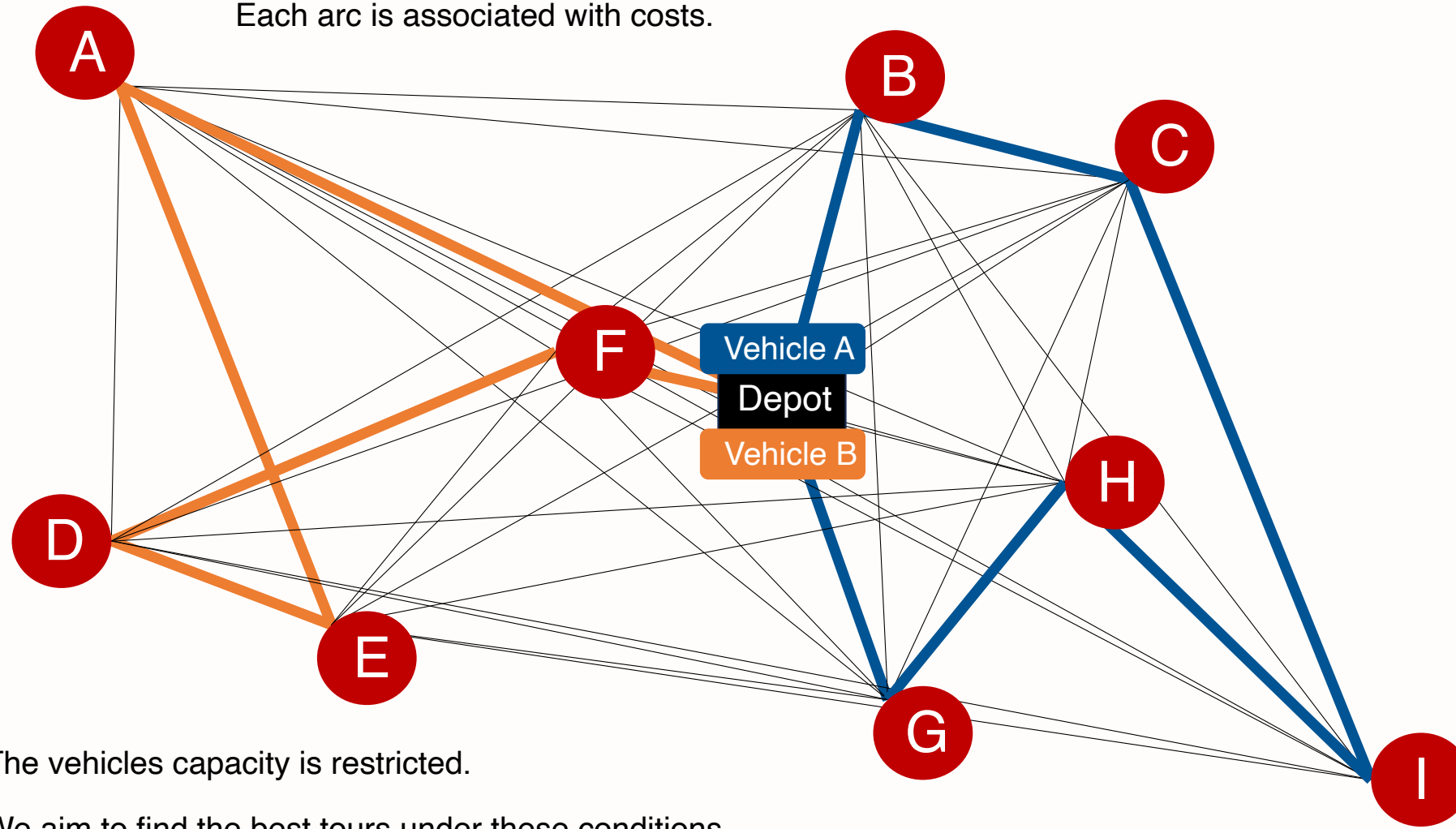
Question: Is this a valid tour for one vehicle?

- No! The loop B $\leftrightarrow$ C **never visits the depot**
- Our model is still **missing constraints**
- We will fix this in the following sections!

Problem Constraints

Each customer has a certain demand.

Each arc is associated with costs.



The vehicles capacity is restricted.

We aim to find the best tours under these conditions.

Constraints?

Question: What constraints do we need?

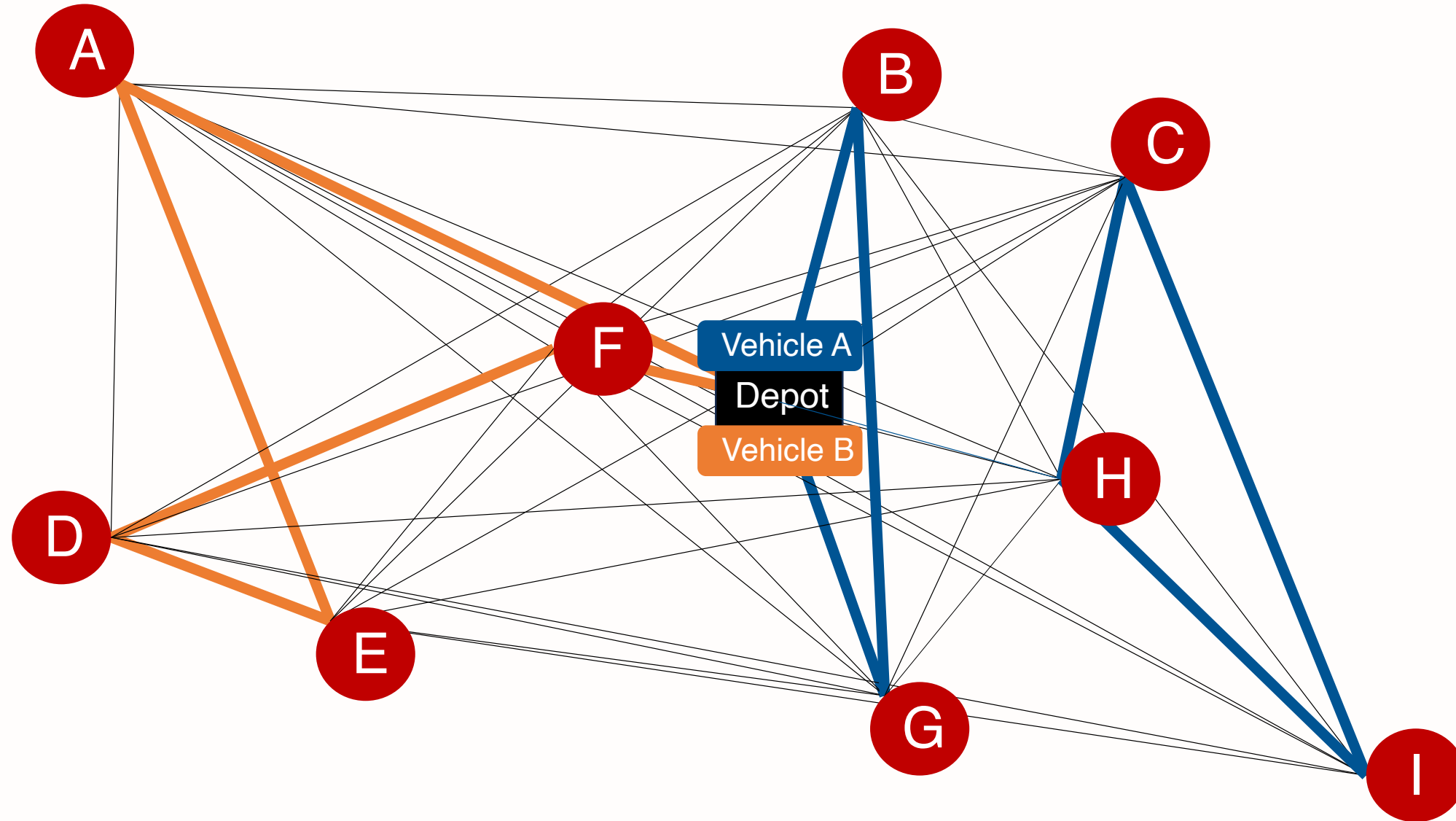
- Each customer has to be **visited once**
- The depot has to be **entered and left** $|\mathcal{K}|$ times
- We have to enforce the **capacity of our vehicles**
- We have to ensure the **maximal duration of each tour**

Subtours

In addition, we have to prevent **subtours**!

What is a
subtour?

Subtours



Constraints

Visit Each Customer Once?

! The goal of these constraints is to:

Ensure that each customer is visited exactly once. Essentially, we could also say that each node **has to be entered and left exactly once**.

i We need the following sets and variables:

- $\mathcal{V}$ - Set of all nodes, index $i \in \{0, 1, 2, \dots, n\}$
- $X_{i,j}$ - 1, if the arc between i and j is part of a tour, 0 otherwise

Visit Each Customer Once

Question: What could the constraint look like?

$$\sum_{i \in \mathcal{V}, i \neq j} X_{i,j} = 1 \quad \forall j \in \mathcal{V} \setminus \{0\}$$

$$\sum_{j \in \mathcal{V}, j \neq i} X_{i,j} = 1 \quad \forall i \in \mathcal{V} \setminus \{0\}$$

Question: Why for all nodes except the depot?

The depot is the **only node** that is **visited multiple times!**

Depot Entry/Exit Constraints?

! The goal of these constraints is to:

Ensure that each vehicle enters and leaves the depot exactly $|\mathcal{K}|$ times, as we have $|\mathcal{K}|$ vehicles and each vehicle has to return to the depot.

i We need the following sets and variables:

- $\mathcal{V}$ - Set of all nodes, index $i \in \{0, 1, 2, \dots, n\}$
- $|\mathcal{K}|$ - Number of vehicles
- $X_{i,j}$ - 1, if the arc between i and j is part of a tour, 0 otherwise

Question: What could the constraint look like?

Depot Entry/Exit Constraints

$$\sum_{i \in \mathcal{V} \setminus \{0\}} X_{i,0} = |\mathcal{K}|$$

$$\sum_{j \in \mathcal{V} \setminus \{0\}} X_{0,j} = |\mathcal{K}|$$

i Are all constraints necessary?

No, theoretically we could also say that we only have to leave **or** enter the depot exactly $|\mathcal{K}|$ times, as the other constraint is already enforced by the “visit each customer once constraint”.

Exactly $|\mathcal{K}|$ Tours?

Question: What if 3 vehicles would suffice, but we own 5?

- The constraints **force** all $|\mathcal{K}|$ vehicles to be used
- With generous capacity, this can lead to **more tours than needed** and thus longer total travel times
- Replacing $=$ with $\leq$ would allow the solver to **leave vehicles unused**

Capacity & Subtours

The next ones are
a little bit tricky.

MTZ Formulation

- Miller-Tucker-Zemlin (MTZ) Constraints from 1960
- Here, in the CVRP form of Kara et al. ([2004](#))
- Prevent subtours and track routes and capacity utilization
- First, we need **an additional variable!**

The MTZ Variable

- U_i - Capacity utilization at i of vehicle on its tour with $i \in \mathcal{V} \setminus \{0\}$
- Note that U_i is a **continuous** variable, not a binary one!

 You don't need to guess these constraints, as they are quite tricky!

MTZ Constraints

i We need the following sets, parameters, and variables:

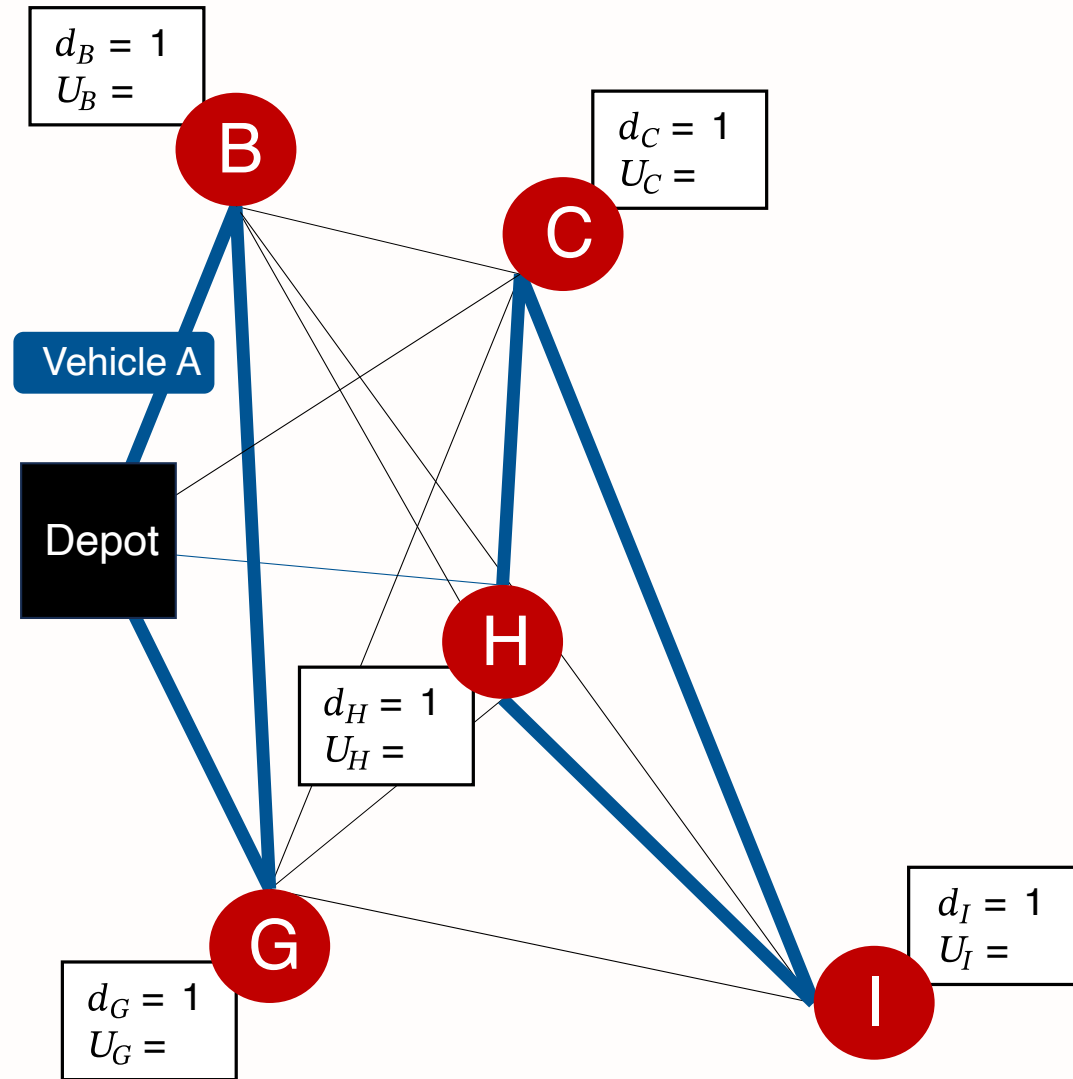
- $\mathcal{V}$ - Set of all nodes, index $i \in \{0, 1, 2, \dots, n\}$
- $X_{i,j}$ - 1, if the arc between i and j is part of a tour, 0 otherwise
- U_i - Capacity utilization at i of vehicle on its tour with $i \in \mathcal{V} \setminus \{0\}$
- b - Capacity per vehicle (all are identical!)
- d_i - Demand at node i

$$U_i - U_j + b \times X_{i,j} \leq b - d_j \quad \forall i, j \in \mathcal{V} \setminus \{0\}, i \neq j$$

$$d_i \leq U_i \leq b \quad \forall i \in \mathcal{V} \setminus \{0\}$$

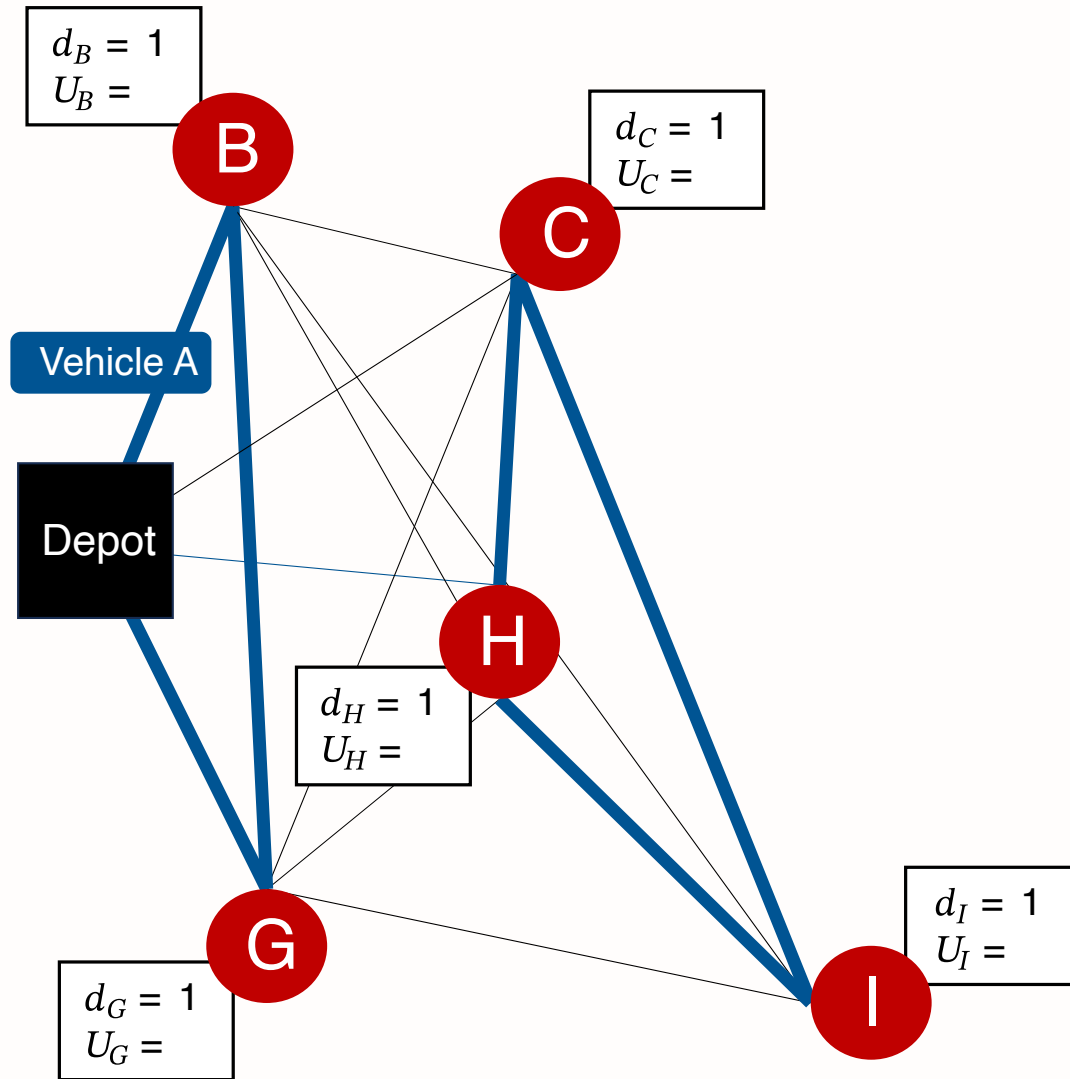
Too
complicated?

Don't worry!



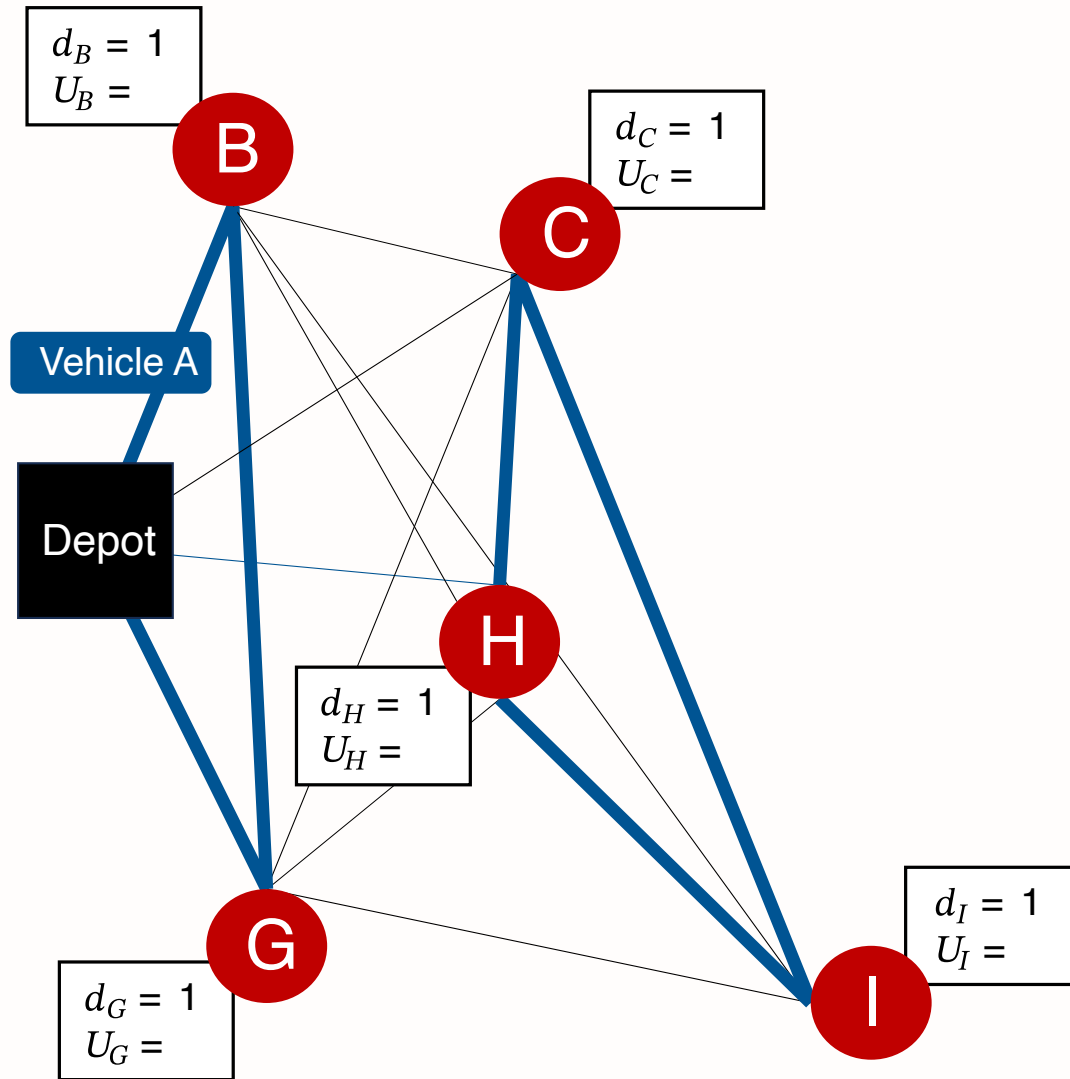
- Let's **break it down!**
- d_i for **all customers** is 1
- Capacity b **per vehicle** is 5
- U_i is the current **capacity utilization** at node $i \in \mathcal{V} \setminus \{0\}$

No connection between nodes



- In case $X_{i,j} = 0$:
 - $U_i - U_j \leq b - d_j$
 - **Non-binding** for relation between two nodes
- Following is **perfectly fine**:
 - $U_i \leq b$ and $U_j \geq d_j$

Connection between two nodes



- In case $X_{i,j} = 1$:
 - $U_i - U_j + b \leq b - d_j$
 - **Binding** for relation between two nodes
- **Can be summarized to:**
 - $U_j \geq d_j + U_i$

Connection in more detail

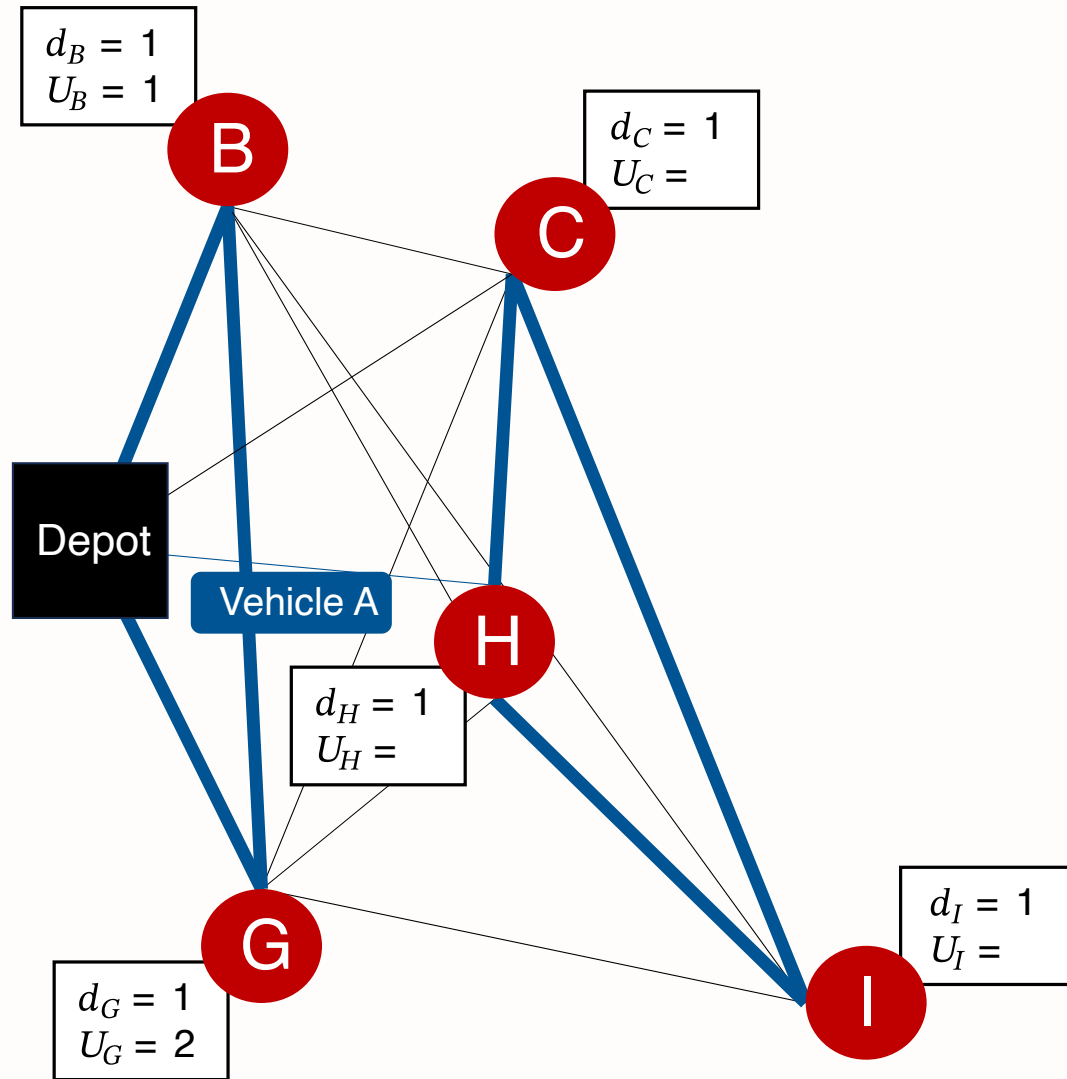
Question: Why is it binding?

- Binding as U_j has to be at least as large as $d_j + U_i$
- Hence, fulfilled if the demand of j is **added to the vehicle**

Question: Do you get the idea?

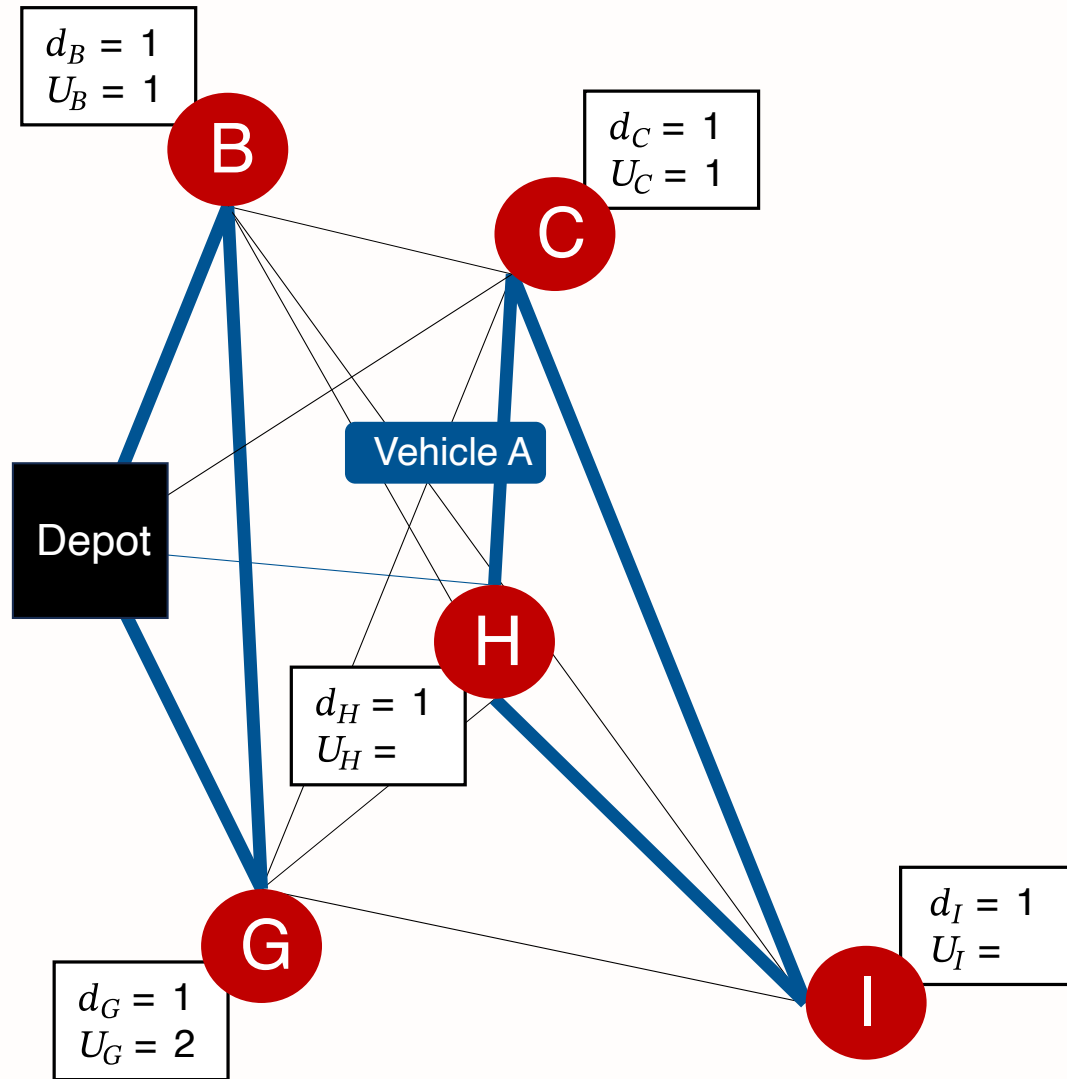
- If $X_{i,j} = 1$, then U_j has to be at least as large as $d_j + U_i$
- If $X_{i,j} = 0$, then $U_i \leq b$ and $U_j \geq d_j$

Tour from the Depot



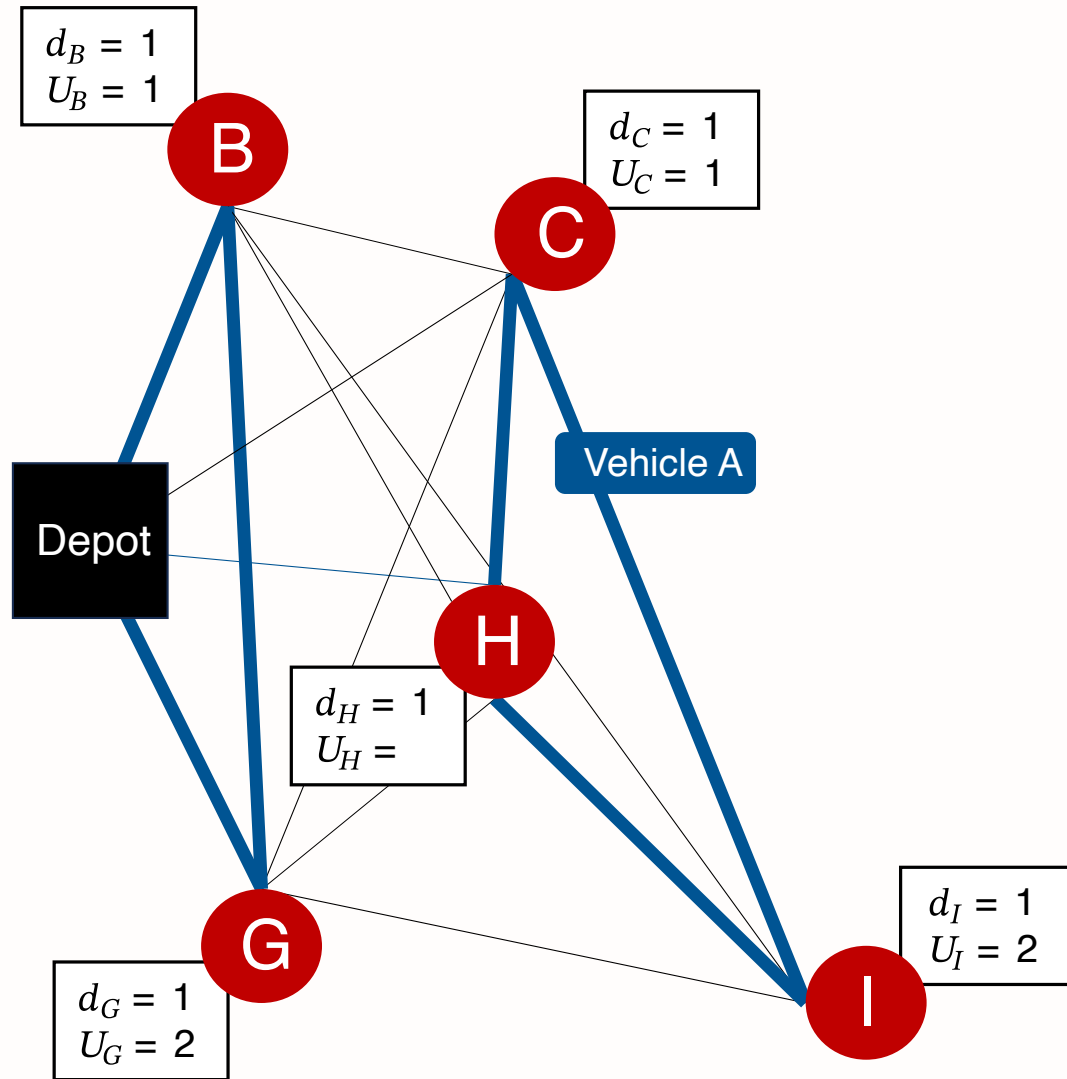
- Tour of vehicle A ok
- Depot is the only node **visited multiple times**
- But the constraints are **not applied here!**

Tour on its Own



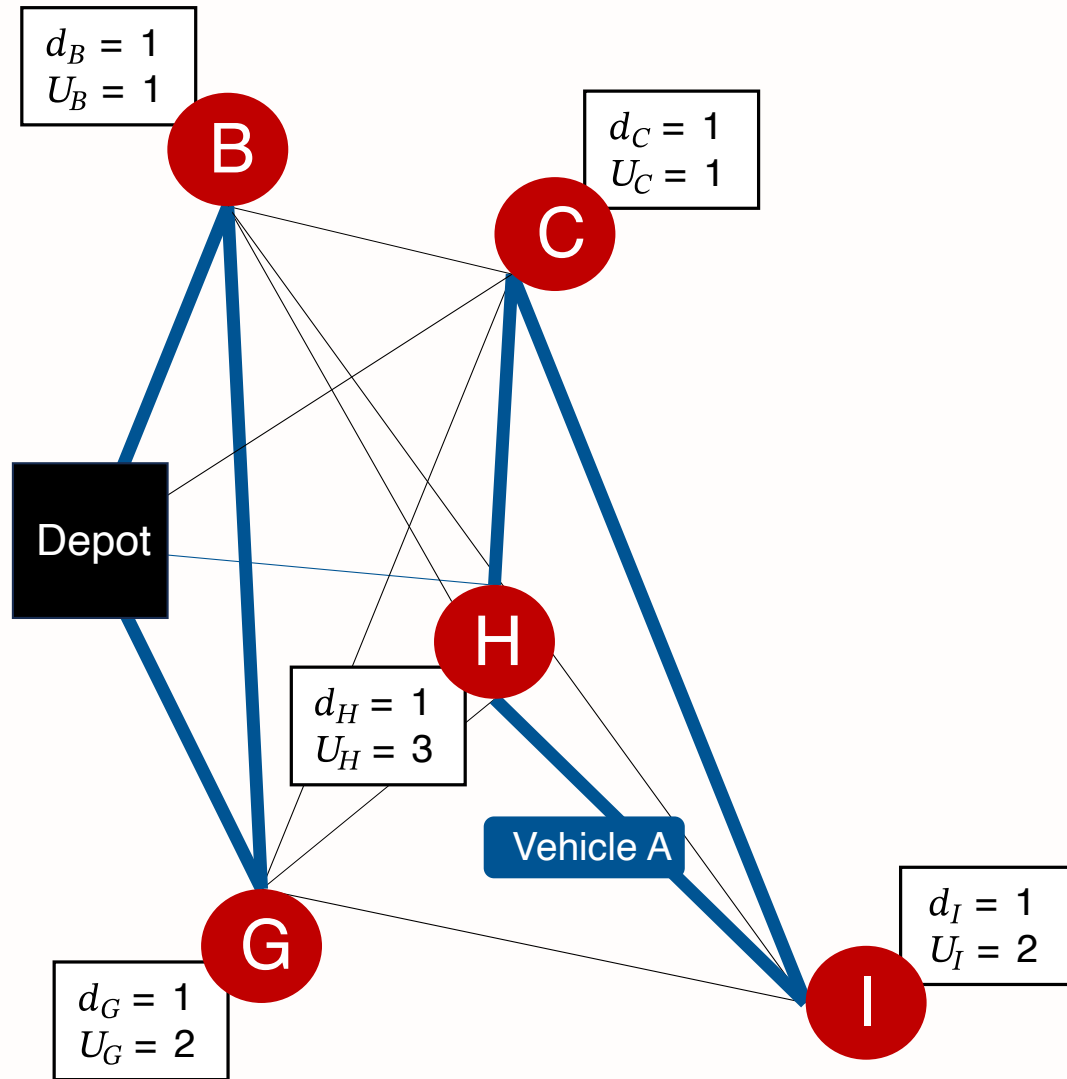
- Assume a subtour
 $H \rightarrow C \rightarrow I \rightarrow H$
- Each arc on it has $X_{i,j} = 1$
- From H to C :
- $U_C \geq U_H + 1$

Tour on its Own II



- From C to I :
- $U_I \geq U_C + 1$
- U has to **increase** along every arc of the subtour

Tour on its Own III



- From I back to H :
- $U_H \geq U_I + 1$
- Now, add up all three constraints
- All U cancel out: $0 \geq 3$
- **Impossible!** No feasible values for U exist!

Subtour Elimination

- The same argument holds for **any** subtour without the depot
- **Only depot as “reset”**, as constraints are not applied here

Question: What about the capacity?

- Remember variable domain of U_i ?
- $d_i \leq U_i \leq b \rightarrow$ **Overall capacity limit enforced!**

Last Constraint

Ensure time limit?

Question: Does anyone have an idea?

- Constraints basically **follow the same idea!**
- First, we again need an additional variable
- T_i - Time spent on tour at node i of a vehicle with $i \in \mathcal{V} \setminus \{0\}$
- T_i is a **continuous** variable, just like U_i

Ensure time limit

i We need the following sets and variables:

- $\mathcal{V}$ - Set of all nodes, index $i \in \{0, 1, 2, \dots, n\}$
- $X_{i,j}$ - 1, if the arc between i and j is part of a tour, 0 otherwise
- T_i - Time spent on tour at the node i of a vehicle with $i \in \mathcal{V} \setminus \{0\}$
- t - Maximal duration of a tour
- $c_{i,j}$ - Travel time on an arc from i to j

$$T_i - T_j + t \times X_{i,j} \leq t - c_{i,j} \quad \forall i, j \in \mathcal{V} \setminus \{0\}, i \neq j$$

$$c_{0,i} \leq T_i \leq t - c_{i,0} \quad \forall i \in \mathcal{V} \setminus \{0\}$$

Time limit and the depot

Question: Why do we need the tightened domain of T_i ?

- $T_i \geq c_{0,i}$: the drive **from the depot** to the first customer counts
- $T_i \leq t - c_{i,0}$: time to **return to the depot** is reserved
- With just $0 \leq T_i \leq t$, a tour could **exceed the limit** by the two depot legs!

Any questions?

Asymmetric CVRP

Objective

$$\text{Minimize } \sum_{(i,j) \in \mathcal{A}} c_{i,j} \times X_{i,j}$$

! The goal of the objective function is to:
Minimize the total travel distance.

Each customer is visited once

$$\sum_{i \in \mathcal{V}, i \neq j} X_{i,j} = 1 \quad \forall j \in \mathcal{V} \setminus \{0\}$$

$$\sum_{j \in \mathcal{V}, j \neq i} X_{i,j} = 1 \quad \forall i \in \mathcal{V} \setminus \{0\}$$

! Our constraints ensure:

Each customer is visited exactly once.

Depot entry and exit

$$\sum_{i \in \mathcal{V} \setminus \{0\}} X_{i,0} = |\mathcal{K}|$$

$$\sum_{j \in \mathcal{V} \setminus \{0\}} X_{0,j} = |\mathcal{K}|$$

! Our constraints ensure:

The depot is visited by exactly $|\mathcal{K}|$ vehicles. Note, that we could remove one of the constraints and the solution would still be optimal.

Capacity/subtour elimination

$$U_i - U_j + b \times X_{i,j} \leq b - d_j \quad \forall i, j \in \mathcal{V} \setminus \{0\}, i \neq j$$

$$d_i \leq U_i \leq b \quad \forall i \in \mathcal{V} \setminus \{0\}$$

! Our constraints ensure:

The capacity limit is respected and subtours are eliminated.

Time constraints

$$T_i - T_j + t \times X_{i,j} \leq t - c_{i,j} \quad \forall i, j \in \mathcal{V} \setminus \{0\}, i \neq j$$

$$c_{0,i} \leq T_i \leq t - c_{i,0} \quad \forall i \in \mathcal{V} \setminus \{0\}$$

! Our constraints ensure:

The time limit is respected (and subtours are eliminated). The domain of T_i accounts for the travel from and back to the depot.

Variables

$$X_{i,j} \in \{0, 1\} \quad \forall (i, j) \in \mathcal{A}$$

$$d_i \leq U_i \leq b \quad \forall i \in \mathcal{V} \setminus \{0\}$$

$$c_{0,i} \leq T_i \leq t - c_{i,0} \quad \forall i \in \mathcal{V} \setminus \{0\}$$

! The variable domains make sure that:

The binary setup variable is either 0 or 1 and the continuous variables U_i and T_i stay within the capacity and time limits.

Model Characteristics

Characteristics

Questions: On model characteristics

- Is the model formulation linear/ non-linear?
- What kind of variable domains do we have?
- What do you think, can the model be solved quickly?

Model Assumptions

Questions: On model assumptions

- What assumptions have we made?
- What are issues that can arise if the model is applied?
- Have we considered service times?

Model Limitations

Our formulation makes **several simplifying assumptions**:

- **No service time** at customer locations
- **Homogeneous fleet** (all vehicles are identical)
- **Deterministic demand** (known exactly in advance)
- **Single depot** (all vehicles start and end at same location)

Real-World Complications

Operational Challenges

- Traffic variability
- Driver breaks and regulations
- Vehicle breakdowns
- Weather conditions

Planning Challenges

- Dynamic customer requests
- Uncertain demand
- Last-minute cancellations
- Multi-day horizons



These complications often require **extensions** of the basic CVRP model or **robust optimization** approaches!

Extensions of the CVRP

Questions: What extensions do you know?

- time windows (TW)
- soft time windows (STW)
- multiple depots (MD)
- heterogeneous fleet (HF)
- backhauls (B)
- pickup and delivery (PD)

In Practice

Case Study in Schleswig-Holstein

- 165 libraries
- 119 visited biweekly
- Up to 9 different tours
- Time-Limit of 8 hours for each tour

**Can our formulation
solve the problem
on a real-world instance?**

No, although we can find solutions within one hour, the gap is still very large with 40%-45%.

Problem is NP-hard

- We have already encountered NP-hard problems in this course
- Likely, that there are no polynomial-time algorithms
- **Doesn't mean that it can't be solved!**

 The CVRP is a generalization of the Traveling Salesman Problem (TSP), which is itself NP-hard!

Why Does This Matter?

The solution space grows **explosively** with problem size:

Nodes	Possible Tour Orders	Approx. Computation
10	3.6 million	Seconds
15	1.3 trillion	Minutes
20	2.4×10^{18}	Hours to days
50	3×10^{64}	Infeasible

Question: Why is the 40-45% gap after 1 hour expected?

Understanding Optimality Gap

- The **optimality gap** measures:

$$\frac{\text{Best Solution} - \text{Lower Bound}}{\text{Lower Bound}} \times 100\%$$

- A 40% gap means our solution **could be** up to 40% worse than optimal¹

Consequences of the Gap

- For 165 libraries, even **state-of-the-art solvers** struggle
- **This is why we need heuristics!**

Can we do
anything to solve
the model?

Heuristics

- We can still solve the problem **with a heuristic**
- **Likely not the optimal solution**, but a lot of research goes into efficient algorithms to solve these problems



For problems with 100+ locations, heuristics are often the only practical choice.

Heuristics for VRP

Construction

Build a solution from scratch

- Nearest neighbor
- Savings algorithm
- Sweep method

Improvement

Refine existing solutions

- 2-opt (swap edges)
- Or-opt (relocate)
- Exchange moves

Metaheuristics

Intelligent search strategies

- Genetic algorithms
- Simulated annealing
- Tabu search



Interested in more details? Check the lecture [Management Science](#)

HGS-CVRP: State-of-the-Art

In our case study we applied **Hybrid Genetic Search** for the CVRP (HGS-CVRP) by Vidal ([2022](#))

- Maintains **diverse population** of solutions
- Applies **intensive local search** to improve offspring
- Achieves near-optimal solutions in **seconds to minutes**



For real-world applications, also consider OR-Tools by Google, it is open-source and production-ready!

Case Study Results

In our project, using **HGS-CVRP** instead of exact optimization:

- Solution found in **< 5 minutes** (vs. hours with MIP)
- Total driving distance reduced by **~20%** compared to manual planning
- CO₂ emissions reduced by approximately **12-15 tonnes/year**

⚠ The heuristic solution is **better** than what the MIP solver found in 1 hour!

Applications Beyond Libraries

Logistics & Delivery

- Package delivery
- Food delivery services
- Grocery delivery
- Mail distribution

Service Industries

- Maintenance crews
- Home healthcare visits
- Waste collection
- School bus routing

Question: Can you think of other applications?

Conclusion

- Standard problem that **occurs in many different places**
- Solving the problem with a mathematical model **is difficult**
- Nowadays, there are **many good heuristics**
- **Many companies** are working on the problem

 And that's it for today's lecture!

We now have covered the Capacitated Vehicle Routing Problem and are ready to start solving some tasks in the upcoming tutorial.

Questions?

Literature

Literature I

For more interesting literature to learn more about Julia, take a look at the [literature list](#) of this course.

Kara, Imdat, Gilbert Laporte, and Tolga Bektas. 2004. “A Note on the Lifted Miller–Tucker–Zemlin Subtour Elimination Constraints for the Capacitated Vehicle Routing Problem.” *European Journal of Operational Research* 158 (3): 793–95.
[https://doi.org/https://doi.org/10.1016/S0377-2217\(03\)00377-1](https://doi.org/https://doi.org/10.1016/S0377-2217(03)00377-1).

Vidal, Thibaut. 2022. “Hybrid Genetic Search for the CVRP: Open-Source Implementation and SWAP* Neighborhood.” *Computers & Operations Research* 140 (April): 105643.
<https://doi.org/10.1016/j.cor.2021.105643>.