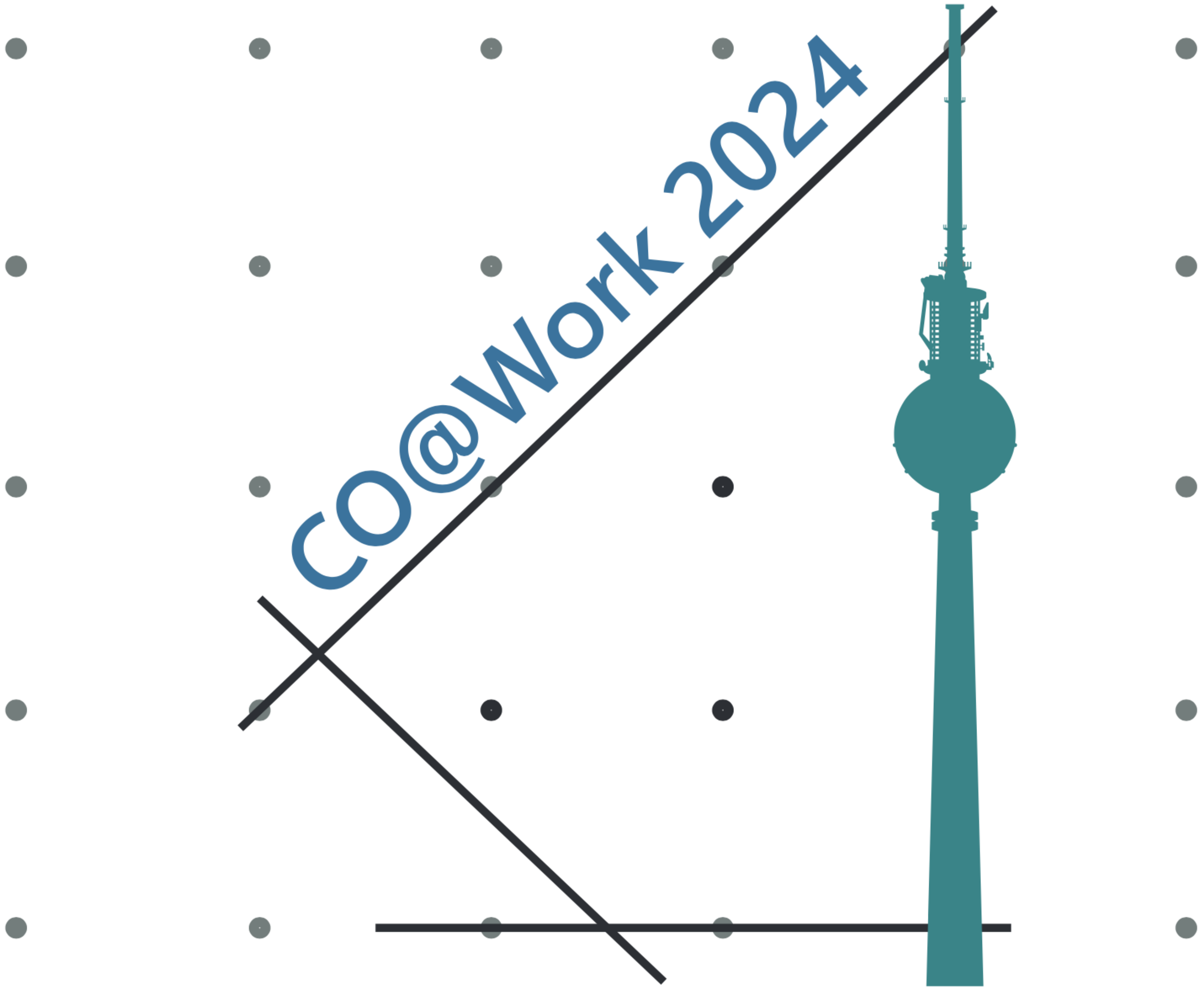
Welcome to Delivery Hero’s Computational Challenge



Kai Hoppmann Baum
Delivery Hero

Milena Petkovic
IKZ Berlin, ZIB

Computational Challange:
Optimize the Future of Delivery – Delivery Hero’s VRPPD Challenge

An unique opportunity to tackle a Vehicle Routing Problem with Pickup and Delivery (VRPPD) arising in the real world.

The Vehicle Rotting Problem? 🍕 🤢 Well, let's hope not!
We’re solving the VRPPD—figuring out how to deliver items efficiently without the food going bad!
Think you can conquer the chaos and keep the pizza fresh?
Let’s see what you’ve got! 🚚 💨

The schedule

Day 1, Saturday 21.09.2024.

- Introduction
- Team draw
- Teams getting together
- Initial setting and data check

Day 2 and Day 3, Monday 23.09.2024/ Wednesday 25.09.2024

- Model development

Day 4, Thursday 26.09.2024.

- Additional test instaces
- Testing and submission preparation

Day 5 , Friday 27.09.2024.

- Winner anouncment
- Winner presentation

Computational Challenge: Submission

Submission deadline: Thursday, 26.09.2024. 18:00h
Email: coaw-data@zib.de
Subject: Computational Challenge Submission TeamNumber

1. Solution Description Document (PDF):

- Filename: teamnumber_teamname_solution_description.pdf
- Length: 1-2 pages
- Content:
  - Final Approach: Describe final solution, model, methodology.
  - Alternative Attempts: Summary of other approaches.
  - Challenges and Solutions: Explain difficulties and solutions.

2. Solution Files (ZIP):

- Filename: teamnumber_teamname_solutions.zip
- Contents:
  - CSV file per test instance, named as instance name.
  - Code files (scripts, notebooks, etc.)
  - Configuration files (if applicable).
  - Any other resources required for evaluation.

Computational Challenge: Scoring

Feasibility:

- One point for each instance with a feasible solution.

+

Ranking:

- Top 5 objective function values get points: 10, 8, 6, 4, and 2.
- Ties: Points awarded to all teams with the same score.

Problem Description: Crutial Components

Vehicles (Riders):

- A fleet consisting of delivery drivers (riders) on bikes, motorcycles, or cars.
- All riders travel at the same speed.
- Each rider starts from a designated location and is responsible for multiple deliveries.
- ^Vehicle capacity is limited

Pickup and Delivery:

- Pickup locations are restaurants
- Riders can handle multiple deliveries, referred to as stacking.
- Riders must always pick up before delivery, maintaining the pairing constraint.

Time Windows:

- Restaurants have specific time windows for food preparation and pickup.
- Arriving too early might result in waiting time for riders.

Capacity Constraints:

- Riders cannot exceed the maximum load capacity during deliveries.

Objective:

- Minimize the sum of all delivery times to ensure food freshness.

Problem Description: A Graph Model of the VRPPD

Vertices

- Set of n vehicles: $K := \{1, \ldots, n\}$
- Each vehicle k has capacity $Q_k \in \mathbb{R}_{\geq 0}$
- Maximum capacity: $Q_{\max} := \max\{Q_1, \ldots, Q_n\}$
- Set of depots: $S := \{s_1, \ldots, s_n\}$
- Set of orders: $O := \{1, \ldots, m\}$, each with pickup and delivery vertices
- Pickup vertices: $\mathcal{P} := \{p_1, \ldots, p_m\}$
- Delivery vertices: $\mathcal{D} := \{d_1, \ldots, d_m\}$
- Total set of vertices: $\mathcal{V} := S \cup \mathcal{P} \cup \mathcal{D}$

Load and Time Windows

- Each order o has a load Q_o
- Pickup: $Q_{p_o} := Q_o$, Delivery: $Q_{d_o} := -Q_o$
- Depot load: $Q_{s_k} := Q_{\max} - Q_k$
- Time windows for pickups: $[\ell_{p_i}, u_{p_i}]$
- No time windows for depots or deliveries: $[0, \infty]$

Problem Description: Crutial Components

Arcs

- $\mathcal{A}_1$: From depots to pickup vertices
- $\mathcal{A}_2$: Between pickup and delivery vertices
- $\mathcal{A}_3$: From delivery vertices to depots
- $\mathcal{A}_4$: Loops at depots
- Each arc (v, w) has travel time T_{vw}
- Total travel time: $T := \sum_{a \in \mathcal{A}} T_a$

Solution

- Vertex-disjoint cycle cover where each cycle contains exactly one depot
- Pickup before delivery in each cycle
- Capacity constraints: load in cycle $\leq$ vehicle capacity
- Feasibility: All conditions (timing, load, pickup/delivery sequence) must be satisfied

Problem Description: Solution

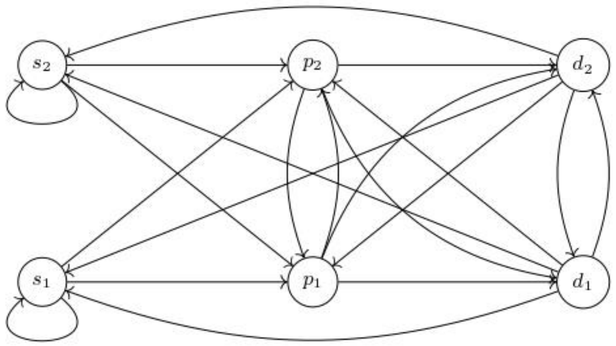


Figure 1: Example graph for two riders and two deliveries.

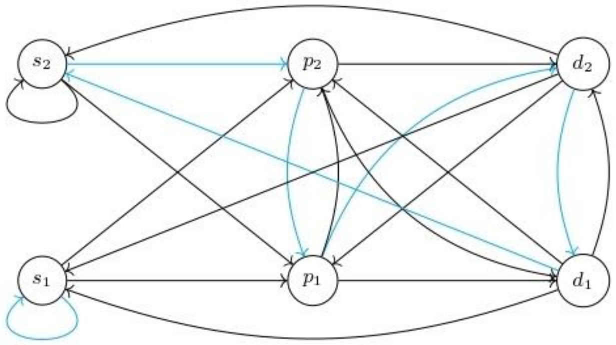


Figure 2: A possible solution to the VRPPD. Here, rider 1 is idle and not assigned any delivery, indicated by the blue loop above its depot. Rider 2 on the other hand picks up delivery 2 first and then delivery 1, thereby creating a so-called stack. Afterwards, they drop off delivery 2 and then delivery 1 before "returning" to its depot.

Problem Description: Example model

$$
\begin{aligned}
\min \; & \sum_{v \in \mathcal{D}} t_v && && (1)\\
\text{s.t.} \; & \sum_{k \in K} \sum_{(u,v) \in \delta^-(v)} x_{uvk} = 1 && \forall v \in \mathcal{V} && (2)\\
& \sum_{(v,u) \in \delta^+(v)} x_{vuk} = \sum_{(u,v) \in \delta^-(v)} x_{uvk} && \forall v \in \mathcal{V}, \forall k \in K && (3)\\
& \sum_{(u,p_o) \in \delta^-(p_o)} x_{up_ok} = \sum_{(u,d_o) \in \delta^-(d_o)} x_{ud_ok} && \forall o \in O, \forall k \in K && (4)\\
& \sum_{(s_k,v) \in \delta^+(s_k)} x_{s_kvk} = 1 && \forall k \in K && (5)\\
& t_u + (T_{uv} + T)\, x_{uvk} \le t_v + T && \forall (u,v) \in \mathcal{A}_1 \cup \mathcal{A}_2, \forall k \in K && (6)\\
& t_{d_o} - t_{p_o} \ge T_{p_o d_o} && \forall o \in O && (7)\\
& q_u + (Q_v + Q_{max})\, x_{uvk} \le q_v + Q_{max} && \forall (u,v) \in \mathcal{A}_1 \cup \mathcal{A}_2, \forall k \in K && (8)\\
& q_{s_k} = Q_{s_k} && \forall k \in K && (9)\\
& \ell_v \le t_v \le u_v && \forall v \in \mathcal{V} && (10)\\
& 0 \le q_v \le Q_{max} && \forall v \in \mathcal{V} && (11)\\
& x_{uvk} \in \{0,1\} && \forall (u,v) \in \mathcal{A}, \forall k \in K && (12)
\end{aligned}
$$

Problem Description: Input data

Couriers.csv

ID	Location	Capacity
1	1	100
2	2	100
3	3	100
4	4	100

Deliveries.csv

ID	Capacity	Pickup Loc	Time Window St	Pickup Stacking_	Dropoff Loc
5	17	5	10	108056	6
6	24	5	11	108056	7
7	3	8	10	151391	9
8	7	10	13	147289	11
9	4	12	23	132201	13
10	15	10	16	147289	14
11	14	15	8	98018	16
12	12	15	7	98018	17
13	15	12	28	132201	18

Traveltimes.csv

Locations	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	12	10	10	15	11	14	10	15	7	1	10	8	4	6	6	10	13
2	12	0	3	11	2	3	6	3	7	6	13	12	6	10	7	7	3	5
3	10	3	0	9	4	0	7	1	8	5	10	9	5	10	6	7	2	6
4	10	11	9	0	13	9	9	8	9	8	11	0	7	11	8	9	8	8
5	15	2	4	13	0	4	4	4	6	8	15	12	7	12	8	9	5	5
6	11	3	0	9	4	0	7	0	8	6	11	8	5	10	6	6	2	6
7	14	6	7	9	4	7	0	7	2	12	15	8	11	16	13	13	8	1
8	10	3	1	8	4	0	7	0	8	6	10	9	5	9	6	7	2	6
9	15	7	8	9	6	8	2	8	0	14	16	9	13	16	13	14	10	2
10	7	6	5	8	8	6	12	6	14	0	7	8	1	4	1	2	4	11
11	1	13	10	11	15	11	15	10	16	7	0	10	8	4	7	7	10	13
12	10	12	9	0	12	8	8	9	9	8	10	0	7	12	8	9	8	8
13	8	6	5	7	7	5	11	5	13	1	8	7	0	5	2	2	3	10
14	4	10	10	11	12	10	16	9	16	4	4	12	5	0	4	3	7	16
15	6	7	6	8	8	6	13	6	13	1	7	8	2	4	0	1	4	12
16	6	7	7	9	9	6	13	7	14	2	7	9	2	3	1	0	5	12
17	10	3	2	8	5	2	8	2	10	4	10	8	3	7	4	5	0	8
18	13	5	6	8	5	6	1	6	2	11	13	8	10	16	12	12	8	0

Problem Description: Output data

bef9b6bc-5be0-4f70-a58a-42461d24e744.csv

bf51ea50-65c7-4064-8267-95d59f98a85c.csv

bf69d1de-260c-4af9-bc57-294b8bc8ee5c.csv

c6fa2c03-39b4-4e04-bdfc-635fbf4d4358.csv

c34a7fb0-dc22-4335-a7c6-dfbc74671e55.csv

c53c456f-0193-4f70-ae7e-b5fdff5bc94f.csv

c079b76e-2c39-4d8b-a77b-52bdee37c9c5.csv

c0201f80-2e86-4743-b658-396017fd6907.csv

c369f40a-d875-4bff-b605-2f7e815799df.csv

c458d4da-f92f-40f7-8ac0-4faf81e34d98.csv

c777e557-4eaa-48b3-ba6d-2f5d660bd683.csv

InstanceName.csv

ID								
1	13	7	13	7				
2	5	6	5	9	9	6		
3								
4	10	8	10	11	12	11	12	8

Computational Challange:
Optimize the Future of Delivery – Delivery Hero's VRPPD Challenge

An unique opportunity to tackle a Vehicle Routing Problem with Pickup and Delivery (VRPPD) arising in the real world.

Thank You and Good Luck!

We encourage you to:
Have fun throughout the challenge
Be creative in exploring different approaches
Enjoy the process of finding innovative solutions