The Transformer Network for the Dial-a-Ride Problem

– Semester Project (12 Credits) –

Student: Aoyu Gong
Section: Communication Science
Supervisor: Prof. Alexandre Alahi

École Polytechnique Fédérale de Lausanne

Background

1. Dial-a-Ride Problem (DARP)

- The DARP consists of designing vehicle routes and schedules for n users who specify pick-up and drop-off requests between origins and destinations.
- In the standard version, transport is supplied by a fleet of K identical vehicles based at the same depot.
- The aim is to plan a set of K minimum cost vehicle routes capable of accommodating as many users as possible, under a set of constraints.

2. Application

- The most common application arises in healthcare services, e.g., door-to-door transportation services for elderly or disabled people.

Formulation (Sec. 3, Cordeau 2006)

- Let n denote the number of users (or requests) to be served.
- Let K denote the number of vehicles providing a shared serve.
- The DARP may be defined on a complete directed graph $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N} = \mathcal{P} \cup \mathcal{D} \cup \{0, 2n+1\}$.
  - Subsets $\mathcal{P} = \{1, 2, \ldots, n\}$ and $\mathcal{D} = \{n+1, n+2, \ldots, 2n\}$ contain pick-up and drop-off nodes, respectively.
  - Nodes 0 and $2n+1$ represent the origin and destination depots.
- Each user i has a maximum ride time L and associates with an origin node i and a destination node $n+i$.
- Each vehicle $k \in \mathcal{K}$ has a maximum vehicle capacity Q_k and a maximum route duration T_k.
  - Set $\mathcal{K} = \{1, 2, \ldots, K\}$ is the set of vehicles.

Problem Statement (cont'd)

Formulation (Sec. 3, Cordeau 2006)

- A load q_i and a non-negative service duration d_i are associated with each node $i \in \mathcal{N}$.
    - For $i \in \{0, 2n+1\}$, $q_i = 0$. For $i \in \mathcal{P}$, $q_i = -q_{n+i}$.
    - For $i \in \{0, 2n+1\}$, $d_i = 0$. For $i \in \mathcal{P} \cup \mathcal{D}$, $d_i \geqslant 0$.
- A time window $[e_i, l_i]$ is specified either for the origin node or for the destination node of a user i, but not both.
    - e_i and l_i represent the earliest and latest time, respectively, at which service may begin at the origin node or the destination node.
- A routing cost c_{ij} and a travel time t_{ij} are associated with each arc $(i, j) \in \mathcal{A}$.
    - c_{ij} and t_{ij} are calculated based on the Euclidean distance.
- The DARP can be formulated as a mixed-integer programming.
    - The objective is to minimize the total routing cost.

Motivation

1. Problem Domain

- The vehicle routing problem (VRP) is a popular and widely studied combinatorial problem (Golden *et al.* 2008, Toth *et al.* 2014).
- The DARP is a difficult problem because it generalizes the VRP by incorporating time windows and maximum ride-time constraints.
  - Finding a feasible solution for the DARP is NP-hard.
  - Approaches for exact solutions can only be designed to solve small to medium-size instances.

2. Technical Domain

- In the last decade, deep learning (DL) has significantly improved computer vision, natural language processing, and speech recognition by replacing hand-crafted features by features learned from data.
- For combinatorial problems, the main question is whether DL can learn better heuristics from data than hand-crafted heuristics?

Generate and Solve Instances

1. Generate Instances

- We generate instances using a method proposed in 2006[1].
  - We consider 2 types of instances: type-a and type-b instances.
  - For each type of instances, we consider 8 kinds of instances.
    - a2-16, a2-20, a2-24, a3-24, a3-36, a4-32, a4-40, a4-48
    - b2-16, b2-20, b2-24, b3-24, b3-36, b4-32, b4-40, b4-48

2. Solve Instances

- We solve instances using an algorithm proposed in 2021[2].

3. Prepare for Supervised Learning

- For each kind of instances, we generate and solve
  - 10,000 instances for creating training and validation sets,
  - and 100 instances for evaluating models.

[1] J.-F. Cordeau, "A branch-and-cut algorithm for the dial-a-ride problem," Oper. Res., vol. 54, no. 3, pp. 573–586, 2006.

[2] Y. Rist and M. A. Forbes, "A new formulation for the dial-a-ride problem," Transp. Sci., vol. 55, no. 5, pp. 1113–1135, 2021.

Inputs and Outputs (Instance b2–16)

1. **Inputs**

By using the method (Cordeau 2006), we have the following inputs:

- $n = 16$, $K = 2$, $Q_k = 6$ and $T_k = 480$ for each vehicle $k \in \mathcal{K}$, $L = 45$
- x_i, y_i, q_i, d_i, $[e_i, l_i]$ for each node $i \in \mathcal{N}$

2. **Outputs**

By using the algorithm (Rist 2021), we have the following outputs:

- The total routing cost: 242.0676
- The **route** and schedule of Vehicle 1:
  - **0** | 59.14 → **11** | 63.00 → **27** | 67.12 → **14** | 93.00 → **30** | 107.14 → ⋯
- The **route** and schedule of Vehicle 2:
  - **0** | 52.76 → **3** | 59.00 → **5** | 71.12 → **21** | 94.00 → **19** | 105.00 → ⋯

Working Procedure (Instance b2–16)

- The **route** and **schedule** of Vehicle 1:
  - **0** | 59.14 → **11** | 63.00 → **27** | 67.12 → **14** | 93.00 → **30** | 107.14 → ⋯
- The **route** and **schedule** of Vehicle 2:
  - **0** | 52.76 → **3** | 59.00 → **5** | 71.12 → **21** | 94.00 → **19** | 105.00 → ⋯

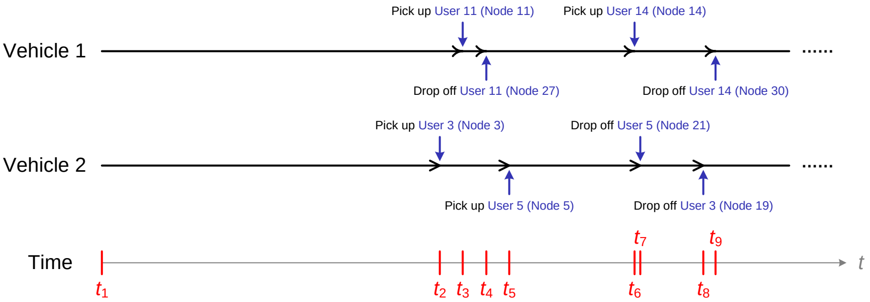

Figure: The working procedure of the two vehicles.

Simulator

Prepare for Supervised Learning

- Given the inputs and outputs, we can simulate the working procedure.
- The simulator will be used to create training and validation sets.

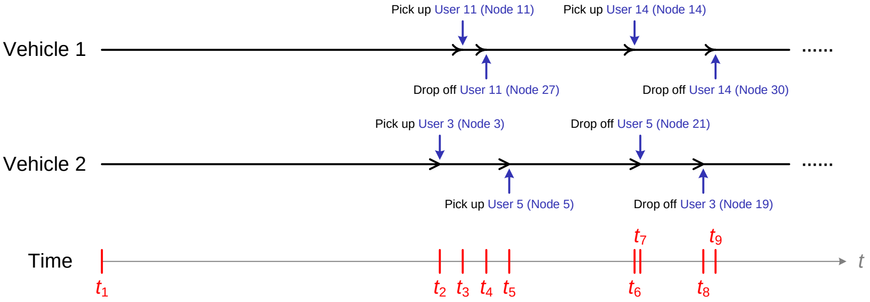

Figure: The working procedure of the two vehicles.

Proposed Formulation

1. States

- At time step t, the system occupies a state, represented by $\mathbf{s}_t$.
- A state consists of $K + 11$ features for each user.
- At time step t, User i have the following features.
  - Travel time from its pick-up or drop-off nodes to K vehicles
  - Load q_i
  - Serve duration d_i
  - Shifted pick-up time window $[e_i - t, l_i - t]$
  - Shifted drop-off time window $[e_{n+i} - t, l_{n+i} - t]$
  - Ride time L_i
  - ID of the vehicle which is serving this user $a_t \in \mathcal{K} \cup \{\text{None}\}$
  - Status $\alpha_t \in \{0, 1, 2\}$
    - α_t: 0 - waiting, 1 - being served by Vehicle a_t, 2 - done
  - ID of the vehicle which will perform an action $b_t \in \mathcal{K}$
  - Status $\beta_t \in \{0, 1, 2\}$
    - β_t: 0 - waiting, 1 - being served by Vehicle b_t, 2 - unable to be served

Proposed Formulation (cont'd)

2. Actions

- At time step t, Vehicle b_t observes the system in state $\mathbf{s}_t$ and then performs an action.
- An action is defined to be the ID of the next user to be served by Vehicle b_t, represented by $i_t \in \{1, 2, \ldots, n\} \cup \{2n+1\}$.
  - $2n + 1$ represents the ID of the destination depot.

3. Environment

- The environment is shown as follows.

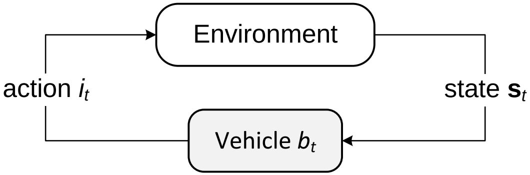

Figure: The environment.

Create Dataset

Observation

- A state-action pair can be generated whenever at least one vehicle becomes free, e.g., Vehicle 1 at t_1, Vehicle 2 at t_1, Vehicle 2 at t_2, Vehicle 1 at t_3, Vehicle 1 at t_4, Vehicle 2 at t_5, and so on.

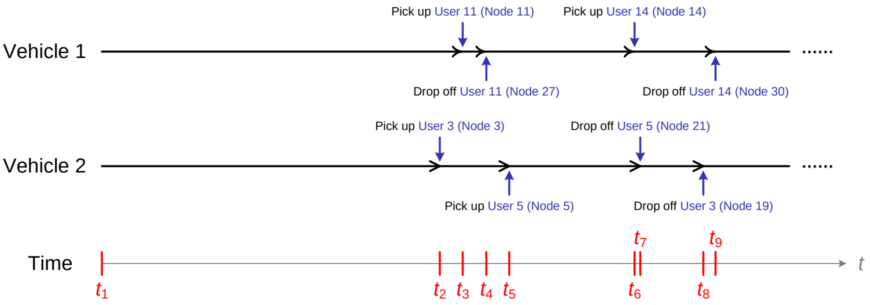

Figure: The simulator of the b2-16 instance.

Create Dataset (cont'd)

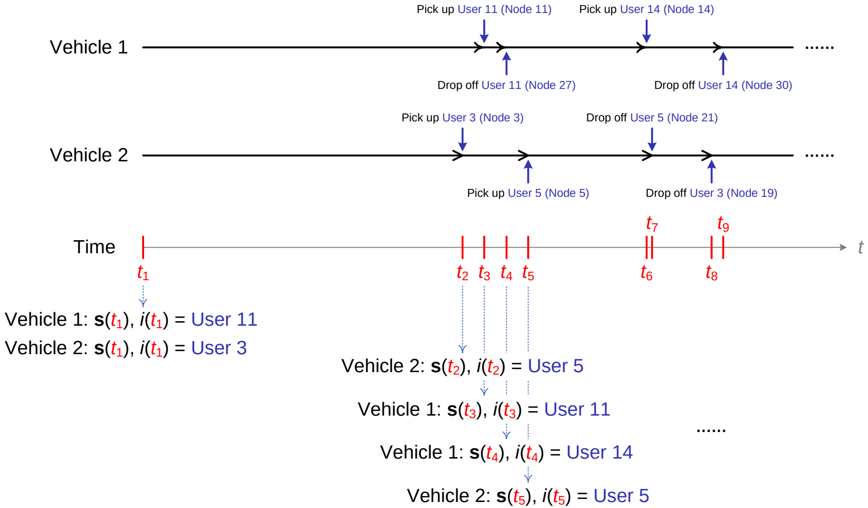

Figure: The method of creating state-action (i.e., feature-label) pairs.

Create Dataset (cont'd)

1. Generate and Solve Instances (Recall)

- We generate instances using the method proposed by Cordeau *et al.*
  - We consider 16 kinds of instances.
    - a2-16, a2-20, a2-24, a3-24, a3-36, a4-32, a4-40, a4-48
    - b2-16, b2-20, b2-24, b3-24, b3-36, b4-32, b4-40, b4-48
- We solve instances using the algorithm proposed by Rist *et al.*
- For each kind of instances, we generate and solve
  - 10,000 instances for creating training and validation sets.

2. Create Dataset

- A dataset can be created by the simulator on 10,000 instances.
  - For b2-16 instances, a dataset consists of 340,000 state-action pairs.
- A dataset is split into training and validation sets with a 98-2 split.

Proposed Model

1. Model Architecture

- The proposed model architecture consists of a input block, an encoder, and an output block.
  - The input block converts a state to n vectors of dimension d_{model}.
  - The encoder is a standard transformer encoder with an input sequence length of n.
  - The output block takes the encoder's output as inputs and predicts user probabilities.
- The cross entropy loss is used as the loss function.

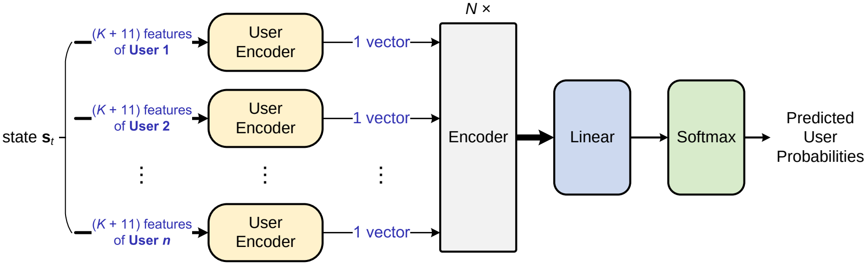

Figure: The proposed model architecture.

Proposed Model (cont'd)

2. User-Encoder Architecture

- The user-encoder architecture consists of a embedding layer, an encoder, and a linear layer.
  - The embedding layer uses 9 lookup tables to store embeddings for $K + 11$ features. It converts them to $K + 11$ vectors of dimension d_{model}.
  - The encoder is a transformer encoder with an input sequence length of $K + 11$.
  - The linear layer applies a linear transformation to the encoder's output. It outputs 1 vector of dimension d_{model}.

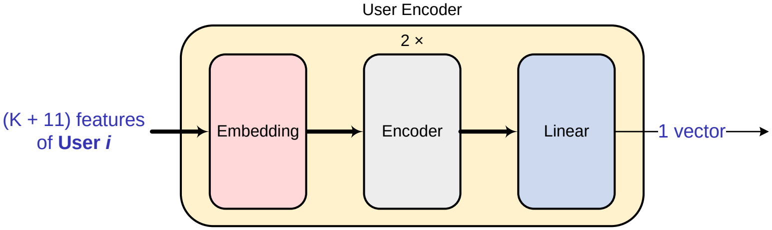

Figure: The user-encoder architecture.

Evaluation Metrics

1. Standard Instances

- For each kind of instances, we have 1 standard instance.
- We propose three metrics to evaluate models.
  - Δ: $\dfrac{\text{Cost (Predicted)} - \text{Cost (Rist 2021)}}{\text{Cost (Rist 2021)}} \times 100\%$
  - N_{TW}: the number of users whose time windows are not satisfied
  - N_{RT}: the number of users whose ride time exceeds the maximum ride time

2. Random Instances

- For each kind of instances, we have 100 random instances.
- We compute the average of the metrics on the 100 instances.
  - $\bar{\Delta}$: $\dfrac{\overline{\text{Cost}}(\text{Predicted}) - \overline{\text{Cost}}(\text{Rist 2021})}{\overline{\text{Cost}}(\text{Rist 2021})} \times 100\%$
  - $\bar{N}_{\text{TW}}$: the average of N_{TW} on the 100 instances
  - $\bar{N}_{\text{RT}}$: the average of N_{RT} on the 100 instances

Results

1. Standard Instances

- For each kind of instances, we have 1 standard instance.

	Cost (Rist)	Cost (Pred.)	Δ	N_{TW}	N_{RT}
b2-16	309.41	302.41	$+2.26$	2	1
b2-20	332.64	347.25	-4.39	1	0
b2-24	444.71	458.53	-3.11	2	1

2. Random Instances

- For each kind of instances, we have 100 random instances.

	$\overline{\text{Cost}}$ (Rist)	$\overline{\text{Cost}}$ (Pred.)	$\bar{\Delta}$	$\bar{N}_{\mathrm{TW}}$	$\bar{N}_{\mathrm{RT}}$
b2-16	281.14	299.65	-6.59	1.55	1.27
b2-20	349.83	370.72	-5.97	1.75	1.51
b2-24	413.06	443.12	-7.28	1.80	1.66

Results

b2-20 (Standard Instance)

- $\Delta = -4.39\%$
  - Cost (Rist 2021): 332.64
  - Cost (predicted): 347.25
- $N_{\mathrm{TW}} = 1$
  - The drop-off time window of User 9 is broken: $118.71 \notin [102.0, 117.0]$
- $N_{\mathrm{RT}} = 0$
- The routes of two vehicles (Rist 2021)
  - 9, 7, 7, 15, 9, 15, 13, 13, 14, 11, 11, 14, 2, 2, 8, 8, 19, 19, 10, 10, 16, 16, 1, 1
  - 5, 5, 20, 20, 3, 3, 17, 17, 12, 12, 6, 6, 4, 4, 18, 18
- The routes of two vehicles (predicted)
  - 7, 7, 15, 9, 9, 15, 13, 13, 14, 11, 11, 14, 2, 2, 8, 8, 19, 19, 16, 16
  - 5, 5, 20, 20, 3, 3, 17, 17, 12, 12, 6, 6, 10, 10, 4, 4, 18, 18, 1, 1

Future Work

1. Formulation

- Change the formulation and create new datasets.
  - Change states to represent the environment more efficiently.
  - Change actions to allow users to wait after the beginning of time windows.

2. Model Architecture

- Change the model architecture and test new models.

3. Machine Learning Paradigm

- Combine supervised and reinforcement learning.
  - First, we train a model with supervised learning and save it.
  - Next, we load the model and train it with reinforcement learning.