Vehicles Routing Task

Problem summary by Rike (last edited: 09-11-21)

What do we want?

Here, we are mainly focusing on describing the routing task related to the new arena in Lüneburg. However, main points might be transferred to the school bus task.

For the former, the input variable `destination` would be the same (arena) for all participants, while for the school task, this might differ (different schools). For the latter we also wouldn't have home addresses but instead directly get pick-up points.[1].

Here is an overview of input and output variables related to our problem, together with some ideas for objectives and constraints.

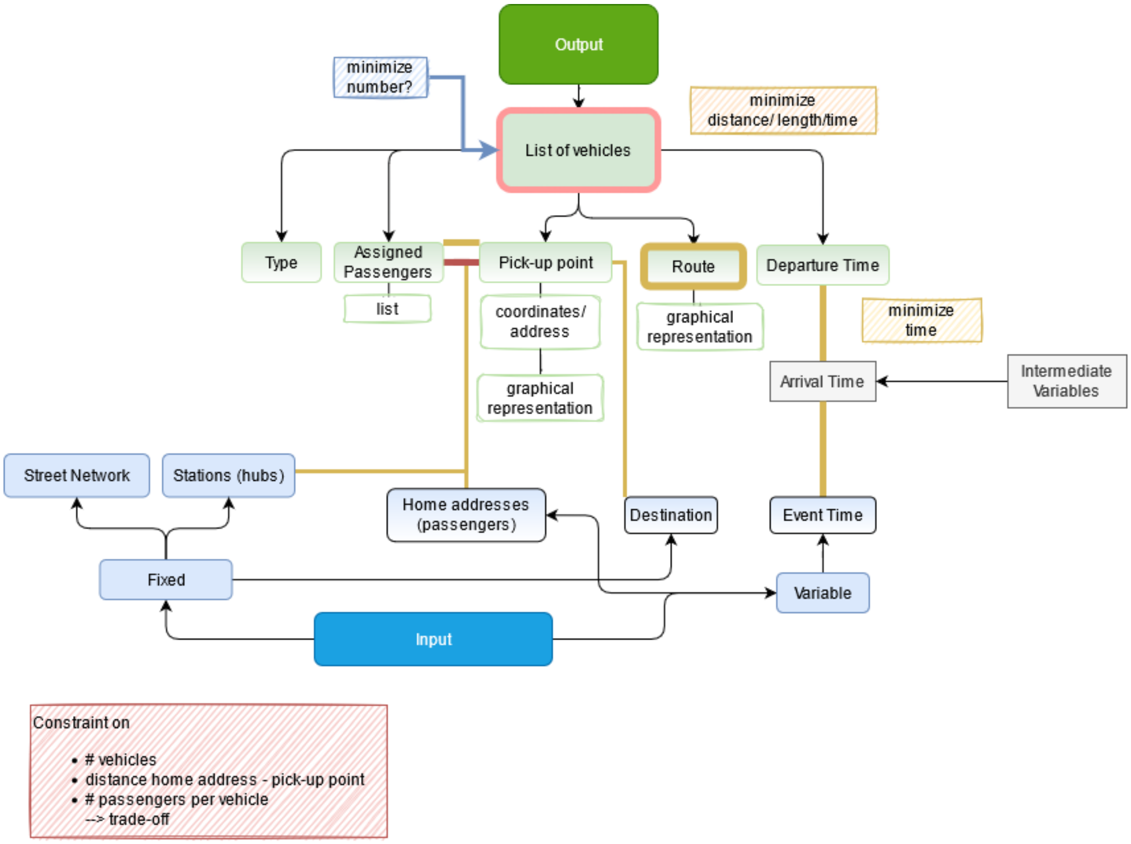

Figure 1: Overview of input and output variables for the arena task

[1] to be clarified

We want to give **passengers** a **pick-up point** and **time** where they could get in order to be picked up by a vehicle and be transported to the arena. These should of course also be given to the **vehicle drivers**.

The output should look something like this:

person_ID	home_address	vehicle_ID	pick_up	departure
1	Steinweg 8, 21335 Luneburg	21	Im Dorf 34, 21335 Lüneburg	10:00
2	Schlesienstraße 6, 21391 Reppenstedt	4	Pommernstraße 14, 21391 Reppenstedt	10:12
3	Vor dem Bardowicker Tore 39, 21339 Luneburg	4	Pommernstraße 14, 21391 Reppenstedt	10:12

Alternatively, GPS coordinates could be provided instead of addresses for pick-up points.

Additionally, the vehicle drivers could receive a map/ route for a navigation system showing the route to drive. Or if we assume that they go straight from the pick-up point to the arena, they could just use navigation systems like GoogleMaps to find the ideal route taking into account current traffic.

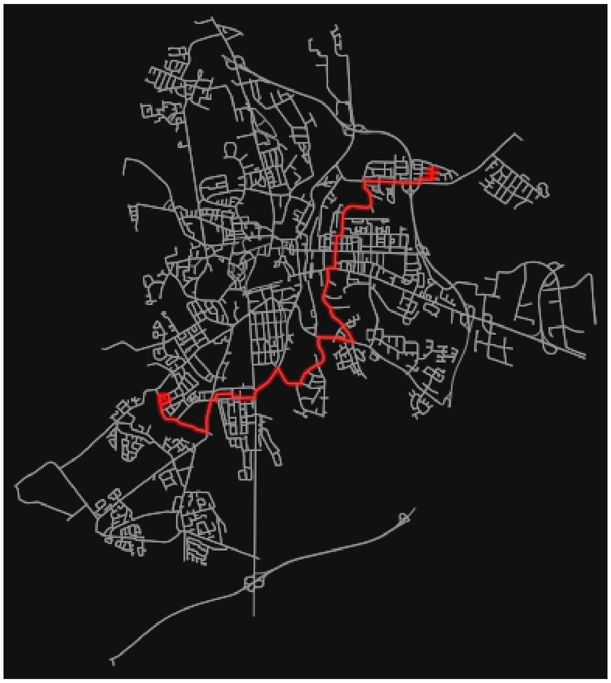

Figure 2: Example for a map of Lüneburg showing an optimal route between two points (in terms of travel time)

For the second part of the task there would also be the variable `event_time` stating when the event starts (for the school task this would be starting times of the respective schools).

Variables

- **Passengers** (*input task 1*)

Each passenger has a `home_address` and is assigned a `person_ID`.

person_ID	home_address
1	Steinweg 8, 21335 Luneburg
2	Schlesienstraße 6, 21391 Reppenstedt
3	Vor dem Bardowicker Tore 39, 21339 Luneburg

- **Event Time** (*input task 1*)
- **Arrival Time** (*intermediate variable*)
- **Pick-up points** (*output task 1*, *input task 2*)

A number of M points (addresses/coordinates) chosen according to our objectives (see task description).

- **Vehicles** (types: *input task 1/2*, routed (list): *output task 2*, scheduled (list): *output task 3*)

In our solution there would be a number of N vehicles $\kappa_1 + ... + \kappa_N$ chosen from a number of V types (multiple vehicles per type possible) that may have different features (e.g. maximum number of passengers). These types should be realistic, i.e. based on insides resulting from our research.

vehicle_ID	model	max_no_passengers
1	big_bus	30
2	mini_bus	7
3	mini_bus	7

- **Departure time** (*output task 3*)

... for each vehicle κ_n.

Tasks

The task can be broken down into smaller sub-tasks:

(1) Find pick-up points

- **Objectives**
  - minimize distance between home address & pick-up point
- **Constraints** ⟶ *How to deal with outliers?* (trade-off)
  - minimum number of passengers per vehicle
  - maximum distance between home address & pick-up point
- **Assumptions**
  - every passenger walks to nearest pick-up point
- ***To consider/ Open questions***
  - location of bus hubs/ starting point of vehicles
  - Is there practically a maximum number of vehicles?

(2) Plan routes

- **Objectives**
  - minimize total distance
  - minimize total travel time
- **Assumptions**
  - people don't change vehicle
  - people walk to nearest pick-up point

(3) Schedule routes

- **Objectives**
  - minimize time between arrival of each vehicle & start of the event
- **Constraints**
  - maximum number of vehicles being at the arena at the same time window (time during which people get off)
  - (b) maximum time between arrival of each vehicle & start of the event
- **Assumptions**
  - fixed time window per passenger number to get off (+ time to park) = time for each vehicle to "unload"