Multidimensional vehicle loads and capacities for Demand Responsive Transport in MATSim

Tarek Chouaki*[a], Sebastian Hörl[a], Volker Grajewski[b], Oliver Ludwig[b], Hannes Rewald[b], and Steffen Axer[b]

[a]Institut de Recherche Technologique SystemX, Palaiseau, France
[b]Volkswagen AG, Wolfsburg, Germany

Submitted for presentation at the MATSim User Meeting 2025 (MUM 2025)
12-13 June 2025, Munich, Germany

Introduction

On-demand mobility systems have been studied extensively for passenger transport. In general, these are systems in which a fleet of shared vehicles (human-driven or automatic) is controlled by a central dispatcher. The dispatcher has knowledge about the currently pending requests with their individual origins, destinations and temporal constraints such as earliest and latest departure and arrival times. Such systems can be simulated using the DVRP and DRT components of MATSim. While the former manages the bare simulation of shared vehicles, the latter focuses on defining the dispatching logic, i.e., which request to match with which vehicle and in which order.

Currently, the main use case for DRT in MATSim is to simulate passenger services in which individual agents are transported by a shared mobility service. Each request, hence, represents one passenger. Likewise, fleet vehicles have individual passenger capacities that are described as an integer value. Recently, features have been added to MATSim that allow the generation of group requests, in which certain agents can travel along with others that send the request to the dispatcher.

Yet, the assumption of one-dimensional loads (passengers) and one-dimensional capacities (seats) poses a limit to the services that can be simulated using MATSim. In particular, two examples shall be cited:

- For the simulation of systems with diverse user profiles, it can be useful to define capacities for those groups. For instance, one may define a seat capacity as well as a wheelchair capacity for each vehicle. This way, some vehicles may be equipped with a dedicated space for wheelchair users. In turn, their seat capacity may be reduced. Wheelchair users would then consume one "wheelchair capacity" of a vehicle, while others consume the "seat capacity".

- Recently, hybrid on-demand transport systems for passengers and goods have gained interested in literature. As certain goods are less time-critial than passengers, they may be transported outside peak times to increase the operational efficiency of the service. In that case, use cases become interesting in which vehicles have a passenger capacity and carrying capacity for goods, or maybe, even various different capacity slots for different goods.

The simulation of such use cases requires the MATSim components to work with (1) multidimensional capacities for vehicles, (2) multidimensional loads for requests, which need to be taken into account in dispatching. The present abstract describes how this functionality has been implemented into MATSim and how it can be used.

*Corresponding Author: tarek.chouaki@irt-systemx.fr

Implementation

Capacity and load representation: Initially, all capacities and loads in DVRP were represented as integers. To represent multidimensional entities, the new `DvrpLoad` interface has been introduced:

public interface DvrpLoad DvrpLoad add(DvrpLoad other); DvrpLoad subtract(DvrpLoad other); boolean fitsIn(DvrpLoad other); boolean isEmpty(); Number getElement(int i);

Internally, the simplest default implementation of the interface is the `IntegerLoad` which, in all its simplicity, encapsulates a single integer value.

Insertion algorithm: The major connecting point to DRT's dispatching request insertion algorithm is the `fitsIn` method. The insertion algorithm traverses all potential points along a vehicle's task schedule to determine whether a new pickup or dropoff for an incoming request could be inserted there. Before anything else, it is checked whether the capacity of the vehicle is not exceeded. In the unidimensional case, the new request would have load $requestLoad \in \mathbb{N}$, the vehicle would already have load $vehicleLoad \in \mathbb{N}$ on board and using the capacity of the vehicle $vehicleCapacity \in \mathbb{R}$ the algorithm would check whether $vehicleLoad + requestLoad \leq vehicleCapacity$. In the multidimensional case, all involved quantities become arbitrarily complex implementations of the `DvrpLoad` interface. The insertion condition is then translated into

```
vehicleLoad.add(requestLoad).fitsIn(vehicleCapacity)
```

The other default implementation of `DvrpLoad` that is provided is `IntegersLoad`, which defines a fixed-size list of integer capacity slots. Note that the implementation assumes *independent* capacity dimensions, i.e., the sum of two loads contains the sum of its individual dimensions and a load fits into a capacity if the values in all dimensions are less or equal to the capacity values.

This does not hinder the framework user of defining loads and capacity with a more complex logic. For instance, one could define a two-dimensional load (L_p, L_l) of persons and luggage. One could then define that two luggage items can go into the trunk, but that each additiona luggage item consumes one seat. The `fitsIn` conditions with capacity C would then become

$$L_p + \max(0, L_l - C_l) \leq C_p \quad \text{and} \quad \min(L_l, C_l) \leq C_l \tag{1}$$

While the modification of the request insertion logic is the largest change to the code base of MATSim, various other changes were required such as including a `DvrpLoad` for each `Request` and each `DvrpVehicle`.

Capacity changes: However, even more advanced functionality has been added that can be used optionally. During development, DVRP has been extended such that the capacity of a vehicle can change throughout the day. Those changes are encoded by dedicated `DrtCapacityChangeTask`s that can be integrated into the vehicle schedules and which, upon being start, modify the capacity of the vehicle. This way it is possible to simulate vehicles that return to the depot, change their configuration (by adding a container for cold goods, for instance) and continue their operations after. Examples on how to use this functionality are given in the accompanying hEART paper and in the MATSim repository[1].

Usage

Multidimensional loads and capacities can now be used in MATSim out-of-the-box. If independent dimensison are sufficient for a specific use case, all relevant information can be set up in the MATSim configuration per DRT mode, the population file (defining the loads) and the fleet/vehicles file used by DRT (defining the capacities).

[1] `https://github.com/matsim-org/matsim-libs/tree/main/contribs/drt-extensions/src/main/java/org/matsim/contrib/drt/extension/reconfiguration`

The available slots are defined per DRT mode in configuration (`slots`):

```
<parameterset name="load">
  <param name="slots" value="passengers,luggage">
  <param name="defaultRequestSlot" value="passengers">
  <param name="mapFleetCapacity" value="passengers">
  <param name="mapVehicleTypeSeats" value="passengers">
  ...
</parameterset>
```

If no specific loads are defined for the persons and trips in the MATSim population, DVRP falls back to sending a request with unit capacity for the slot defines in `defaultRequestSlot`. Likewise, if vehicles are read from a fleet file (`fleet.xml`), the given capacity of a vehicle is interpreted as the slot defined in `mapFleetCapacity`. Alternatively, vehicles can be provided to DRT through a standard MATSim vehicles file (`vehicles.xml`) which already defines different capacity dimensions (previously unused in DRT). In the example above, the 'seats' attribute of a 'VehicleType' is mapped to the 'passengers'.

However, DRT now provides additional functionality to define loads individually in the population file. Using specific attributes, one can define loads per trip (as an attribute of the preceding activity) or person person (in the person attributes):

```
<attribute name="dvrp:load:passengers" class="java.lang.Integer" value="1">
<attribute name="dvrp:load:luggage" class="java.lang.Integer" value="2">
```

Furthermore, vehicle capacities can be defined using the vehicles file per-vehicle. The respective attribute names and inheritance rules are documented in the MATSim repository [2].

[2] `https://github.com/matsim-org/matsim-libs/tree/main/contribs/dvrp/src/main/java/org/matsim/contrib/dvrp/load`