SUBPART TWO

Mobsim

CHAPTER 11

QSim

Marcel Rieser, Kai Nagel and Andreas Horni

11.1 Vehicle Types and Vehicles

For a variety of reasons—e.g., vehicle-specific emissions calculations (Chapter 36), or vehicle-specific maximum speeds (see below)—it may become necessary to assign different vehicle types to different persons, modes, or trips. The (arguably) most "honest" approach to vehicles, in terms of micro-simulation, is to generate a synthetic vehicle fleet. Each leg would then have to know which vehicle it wants to use. This is indeed possible with the planned vehicle ID that MATSim route objects can store. This functionality is switched on by first loading an additional vehicles file (see Section 6.6) and then configuring the QSim as

```
<module name="qsim">
   <param name="usePersonIdForMissingVehicleId" value="false" />
   <param name="vehiclesSource" value="fromVehiclesData" />
   ...
</module>
```

(available with release 0.8.x). This states that every time the QSim needs a vehicle with a specific ID, it will search for it in the vehicles data container, throwing an exception if it is not found there.

A Fallback for Routes that do not Contain Vehicle IDs In the above approach, vehicular routes need to provide vehicle IDs, otherwise the QSim will throw an exception. Since algorithms to compute and maintain vehicle IDs during replanning are currently not well developed within MATSim, an alternative is to assume that persons use a vehicle with the same ID as the person. This is switched on by

How to cite this book chapter:
Rieser, M, Nagel, K and Horni, A. 2016. QSim. In: Horni, A, Nagel, K and Axhausen, K W. (eds.) *The Multi-Agent Transport Simulation MATSim*, Pp. 77–80. London: Ubiquity Press. DOI: http://dx.doi.org/10.5334/baw.11. License: CC-BY 4.0

```
<module name="qsim">
   <param name="usePersonIdForMissingVehicleId" value="true" />
   ...
</module>
```

which is also the current default. With this configuration, it will still search for the vehicle ID in the route, but if this is unavailable, it will instead use the person ID as vehicle ID. Without additional configuration, it will then still search for the vehicle under that ID in the vehicles file.

Alternative Vehicles Sources A default vehicles source is defined by

```
<module name="qsim">
   <param name="vehiclesSource" value="defaultVehicle" />
   ...
</module>
```

This generates a default vehicle (typically a medium-sized 4-seater) every time a vehicle is needed and is currently the default configuration.

At the moment, alternative approaches to vehicle generation need to be programmed as script-in-Java. See, e.g., `RunMobsimWithMultipleModeVehiclesExample` under `http://matsim.org/javadoc` → main distribution for a reference to a script that generates mode-specific typical vehicles for each mode. Simulation experiments using this feature have been performed for the Patna scenario as reported in Chapter 77.

Vehicle Behavior Vehicles need to be available where they are needed. It is, for example, impossible to perform a trip by car, then another (non-circular) trip by public transit and then make another trip with the same car as before, since the car will not be available at that location. The QSim is able to enforce such behavior, with the setting

```
<module name="qsim">
   <param name="vehicleBehavior" value="exception" />
   ...
</module>
```

This means that if a necessary vehicle is not available at the location where it is needed by the traveler, the QSim throws an exception and aborts. The idea here is that such synthetic travelers should have within-day replanning strategies (see Chapter 30) to cope with unexpectedly unavailable vehicles; any attempt to use an unavailable vehicle points to an error in the driver's behavioral logic.

In many standard situations, the above behavior will be too strict. For example, a vehicle may be shared between family members, but one member will be late in returning a vehicle. For such situations,

```
<module name="qsim">
   <param name="vehicleBehavior" value="wait" />
   ...
</module>
```

may be an option. Here, a driver will wait if a vehicle is not available. Errors in the coordination logic, i.e., very long waits, will be punished via the MATSim scoring logic, thus leading to more robust coordinations.

A final alternative is

```
<module name="qsim">
   <param name="vehicleBehavior" value="teleport" />
   ...
</module>
```

With this setting, vehicles will be teleported to locations where they are needed.

Initial Vehicle Placement For vehicle behavior of type `exception` and type `wait`, vehicles need to be at the correct location when the QSim starts. Here, the default simulation currently places all vehicles at the home location—for other variants, some additional code needs to be written or used, such as the car sharing extension (Chapter 22).

PassingQ Once various vehicles have different maximum speeds, the standard QSim, even with multiple main modes, is no longer sufficient, since it uses FIFO (First In, First Out) as the queuing discipline, meaning that fast vehicles cannot pass slower vehicles. Here, the so-called `Passing(Vehicle)Q` can be used instead. It replaces the FIFO sorting criterion—where vehicles are sorted by the sequence in which they arrive on the link—by a sorting employing the so-called earliest link exit time, computed from link enter time and freespeed travel time. Now, using the minimum of vehicle and link maximum speeds, the freespeed travel time can be differentiated between vehicles, allowing fast vehicles to obtain an earlier link exit time, even if they enter later than slow vehicles. Details and resulting fundamental diagrams are given by Agarwal et al. (2015b).

This option can be enabled by using

```
<module name="qsim">
   <param name="linkDynamics" value="passingQ" />
   ...
</module>
```

in the `qsim` section of the config file.

11.2 Other

The simulation is able to handle time-variant networks (Lämmel et al., 2010), within-day replanning (Dobler, 2009, see Chapter 30) and traffic lights (Neumann, 2008; Grether et al., 2011b, 2012, see Chapter 12). An earlier multimodal approach, targeted at overcoming the teleportation estimates of non-motorized modes, and particularly focused on pedestrians, is presented in Chapter 21.