Dynamic ride-sharing for MATSim – presenting the DRS module

Eyad Nassar[a], Markus Straub[b,*], Johannes Müller[b]

[a] *Technical University Munich (TUM), Munich, Germany*

[b] *Austrian Institute of Technology (AIT), Vienna, Austria*

Keywords: MATSim module, dynamic ride-sharing, carpooling, drs

The MATSim simulation framework allows agents to move along the network using vehicles, thereby accurately modeling traffic demand. However, the framework lacks the ability to properly simulate occupancy rates since agents in vehicles cannot pick up other agents. In this work, we present a MATSim module that addresses this limitation by enabling car drivers to pick up other agents, share a ride with them in the same vehicle, and drop them off before continuing their own trip.

Often passengers in cars have been incorporated as `ride` agents teleported from one activity location to another using car travel times and without being matched with other agents (e.g., Ziemke et al. (2019)). While this approach suffices for private ride-sharing, it fails to represent the potential for general dynamic ride-sharing (DRS) which refers to the practice of sharing a vehicle where non-acquaintances share rides to reach similar destinations. Wang et al. (2017) conducted previous endeavors to model DRS in MATSim, employing the `dvrp` module (Maciejewski and Nagel, 2013). Their approach involved introducing a dynamic agent alongside the standard population agent during simulation iterations, specifically chosen when the leg's mode is ride share driver.

We present a novel MATSim module to simulate DRS: Agents get an a priori attribute that determines if they are potential DRS driver, DRS rider, both or no DRS user. A maximum of one rider is allowed per DRS trip. Pickup and dropoff of the rider takes place directly at the rider's activity locations. In contrast to the work by Wang et al. (2017) this approach reduces complexity and avoids the waiting time for the rider between making the request and getting picked up. Our module assigns DRS driver and rider in the `replanning` phase of the MATSim loop before `mobsim`, i.e. from the users' perspective requests must be made the day before.

In more detail, the presented module integrates DRS into the MATSim loop as follows:

- **Replanning:** First the slightly adjusted innovation strategy `SubtourModeChoiceForDrs` assigns the new modes `drsDriver` and `drsRider` to agents' subtours. This assignment can be restricted with the optional person attribute `drsAffinity`. Then requests are collected and matched based on origin, destination, departure time and detour time. The riders' acceptance of deviations to their desired departure time can be controlled with the DRS config parameter

*Corresponding author: markus.straub@ait.ac.at

`riderDepartureTimeAdjustmentSeconds`. The agents' plans are adjusted accordingly. This entails adding pickup and dropoff activities at the riders' origins and destinations to the plans of matched drivers. For matched riders the departure time is adjusted if necessary.

- **Mobsim:** Each matched driver proceeds to the specified pickup point to collect the assigned rider. The DRS config parameter `pickupWaitingSeconds` determines how long a driver waits for a delayed rider before proceeding. Subsequently, the driver transports the rider to the designated dropoff point. Concurrently, matched riders await the arrival of their driver for pickup and subsequently for dropoff at the predetermined locations. Riders that could not be matched still get a "mobility guarantee" and are teleported to their next activity based on `teleportedModeParameters` in the `planscalcroute` config.

  Note, that to successfully simulate DRS it is necessary to (1) kickstart the pool of potential drivers and (2) to assign costs to the mobility guarantee. Before the first iteration all agents that can potentially act as a driver are assigned a copy of their original plan with as many driver legs as possible. This avoids the problem of riders not finding a match. On the other hand the mobility guarantee must be assigned a cost with the DRS config parameter `riderMobilityGuaranteeMonetaryConstant`. Otherwise unmatched ride trips can make up a relevant part of the modal split when ideally they should only occur in negligible amounts.

- **Before next iteration:** Adjustments to the agents' plans are reverted.

The application of the DRS module in a case study conducted in Upper Austria yielded successful results (Nassar, 2023). Various parameters, such as the proportion of the population willing to participate as DRS drivers or riders, driver incentives, and matching cell size, were implemented as scenarios. The study revealed that approximately 60-80% of all DRS riders were successfully matched with DRS drivers, while the proportion of matched DRS drivers was considerably lower, reaching around 5%. For scenarios with 324k agents and 181k potential DRS users runtime per iteration increased but is still feasible (`iterationStartsListener`: +40%, `mobsim`: +20%).

There are multiple ways to further develop this module. It would be advantageous to allow DRS drivers and riders to meet at specific designated points instead of the riders' activity locations. Additionally, providing alternative modes of transportation for unmatched riders instead of enabling the mobility guarantee via teleportation would enhance the module's functionality and flexibility.

This work is part of the scientific project DOMINO supported by the Austrian Research Promotion Agency under grant 3300226. The DRS MATSim extension is available as open source software on https://github.com/ait-energy/matsim-drs.

References

Maciejewski, M., Nagel, K., 2013. Simulation and dynamic optimization of taxi services in matsim. Transportation Science .

Nassar, E., 2023. Integrating Dynamic Ride-Sharing in MATSim. Master's thesis. Technical University Munich.

Wang, B., Liang, H., Hörl, S., Ciari, F., 2017. Dynamic ride sharing implementation and analysis in matsim. hEART 2017: List of Abstracts .

Ziemke, D., Kaddoura, I., Nagel, K., 2019. The matsim open berlin scenario: A multimodal agent-based transport simulation scenario based on synthetic demand modeling and open data. Procedia computer science 151, 870–877.