Simulating on-demand person and food transport with intra-day vehicle reconfiguration in Paris

Tarek Chouaki*[1], Sebastian Hörl[1], Oliver Ludwig[2], Steffen Axer[2], and Hannes Rewald[2]

[1]Institut de Recherche Technologique SystemX, Palaiseau, France
[2]Volkswagen AG, Wolfsburg, Germany

*Corresponding author

Short summary

Mobility-on-demand (MoD) services for passenger trips are extensively studied in the literature. Quantitative methodologies relying on agent-based mobility simulations, such as MATSim, allow assessing such systems in realistic settings and aid their design and sizing. Similarly, MoD services for parcel deliveries and logistics are gaining interest in the research community around agent-based simulations. The paper at hand presents a work investigating the potential of mixed MoD services for passengers and goods where vehicles must perform physical reconfigurations to switch from each demand type. A demand-dependant reconfiguration planning algorithm is proposed and the mixed service compared against using two separate fleets to assess the potential advantage. This work is highly reproducible and presents a technical contribution to the MATSim tool to allow such simulations to be performed.

Keywords: mobility-on-demand, heterogeneous demand, reconfigurable fleets, simulation

1 Introduction

Mobility-on-Demand (MoD) services for passengers are increasingly studied in the literature and evaluated for their potential to encourage a modal shift away from private cars and reduce congestion through ride-pooling and system-wide fleet optimization (Narayanan et al., 2020). Agent-based mobility simulations represent an important tool that allows to simulate and evaluate such services (Jing et al., 2020). The use of realistic demand models and the ability to feature multiple modes enables to study on-demand services in close to real-life settings while considering the interaction with other modes such as public transport.

In addition, on-demand parcel delivery and other freight transportation services are studied using agent-based simulations. Although less data are openly available to build realistic delivery demand models, several contributions have been made in this area recently (Sakai et al., 2022; Hörl & Puchinger, 2023).

MATSim (Horni et al., 2016) is an agent-based simulation framework that is increasingly used in research and in the industry to study, design, and evaluate MoD systems thanks to the DRT and DVRP modules included within the tool (Maciejewski et al., 2017). The modules handle the service's logic: processing requests, performing vehicle assignment, and relocation decisions. While the modules were primarily designed for passenger trips, parcel delivery systems have been studied using MATSim by representing parcels as travelling agents (Meinhardt et al., 2022).

Today, fleet operators are investigating the feasibility of combining passenger mobility and goods transportation within a single on-demand system (one fleet) (Fehn et al., 2023). This is motivated by the potential advantage that can be achieved by operating one single fleet for both passenger and goods, instead of two separate (but perhaps smaller) fleets. The potential advantage is even more increased with Business-To-Customer deliveries where the customer is required to be present on delivery time, which implies that passenger trips and deliveries occur at different times. However, in such systems, vehicles might need to undergo reconfiguration operations to switch between person trips and goods transportation.

In this work, a novel type of MoD services for a mixed demand of passengers and food deliveries is studied. To our knowledge, this study is the first to consider a mixed demand for a MoD system in an agent-based simulation with fleet reconfiguration. The next section presents an overview of the current state of the literature. The simulation methodology, the service operation that is implemented, and the adaptive reconfiguration logic are then detailed, and the experimental protocol is outlined. Afterwards, the simulation results are presented and discussed before concluding with the main takeaways and perspectives for future research.

2 BACKGROUND

Mobility-on-Demand

Passenger mobility-on-Demand (MoD) systems are typically accessed via a mobile application that allows to send travel requests. The assigned vehicle's driver is then able to travel to pick-up the traveller and then ride to the drop-off location. Such services are implemented by various operators throughout the world (Uber, DiDi, Lift, Heetch... (Schaller, 2018)). The evolution of autonomous driving technology is expected to further increase the interest of MoD. Mainly, an AMoD (Autonomous MoD) system would be able to ensure a consistent offer throughout the day. Such systems are increasingly studied in the literature. The research in this area has matured enough to reach the design of these services on the operational level while taking into account close to real-life settings(Golpayegani et al., 2022).

The design of MoD systems, whether they rely on autonomous vehicles or not, involves different decisions on various levels. The fleet size, vehicle characteristics, pricing, service area, operational strategy, all have an impact on the performance and attractiveness of these systems towards users (Vosooghi et al., 2019). Operational strategies or policies refer to how a MoD system performs the day-to-day trip-level decisions. The most commonly studied ones are: (i) Empty vehicle rebalancing: where to send empty idle vehicles and (ii) Vehicle assignment: to match available vehicles with pending travel requests.

Agent-based mobility simulation tools, and MATSim in particular, are currently used to assess and evaluate MoD services. In the following, we detail the current state of the MoD-related functionalities present in the tool.

MoD Fleet operation in MATSim

The DVRP and DRT modules within MATSim simulate MoD fleets by maintaining and processing vehicle schedules. Each vehicle's schedule is represented as a sequence of tasks that can be of three types: (i) *drive* tasks represent the vehicle moving from one location to another following a certain route, (ii) *stop* tasks represent the vehicle being immobilized at a certain time period to pick up and/or drop off passengers, and (iii) *stay* tasks represent the vehicle being idle with no planned pick-up or drop-off. At the beginning of the day, all fleet vehicles start with a schedule containing only a stay task spanning the vehicle's service period. Then, as other tasks are added, *stay* tasks are used to fill the gaps and ensure that the schedule covers the target period.

The vehicle assignment algorithm featured in the DRT module processes travel requests in sequence and employs an insertion heuristic to determine to which vehicle the request will be assigned. To do so, possible insertion points of the request in the vehicle's schedule are evaluated for their impact on the traveller behind the request, the travellers behind requests already assigned to the vehicle, and the vehicle itself (added driving distance). Only insertions that do not violate any hard constraint are considered. The most fundamental constraint relates to the vehicle's capacity, which cannot be exceeded anywhere alongside the vehicle's schedule. When evaluating a request insertion, only the resulting occupancy between the insertion's related pick-up and drop-off is compared to the vehicle's capacity, which is assumed to be *constant* during the day. Other constraints relate to users whose waiting and arrival times cannot exceed certain thresholds. Among admissible insertions of all vehicles, the best one is selected and the related vehicle's schedule updated accordingly before processing the next request. As a result, already planned *stop* tasks can be shifted to accommodate the new ones as long the constraints are not violated. *Stay* tasks on the other hand are shifted, removed and created freely.

Another part of MoD operation that is natively supported within the DRT module is empty vehicle relocation, also called rebalancing. This consists in deciding where idle, unoccupied MoD vehicles should be relocated in order to better serve future trip requests. In general, a rebalancing algorithm attempts to predict where future demand will arise and proactively redistribute empty vehicles to achieve better wait times and maximize demand satisfaction. Various rebalancing strategies are included in the DRT module (Ruch et al., 2020; Bischoff & Maciejewski, 2020).

3 Methodology

As stated in Section 1, we consider a MoD service with a fleet of vehicles that can serve both passenger requests and food deliveries. The only constraint is that a vehicle can have only one compatibility (passengers or food) at a time and must undergo a reconfiguration in a dedicated facility to change its compatibility setting between passenger-setting and food-setting. Furthermore, each setting is associated with a maximum capacity that typically represents the maximum number of simultaneous occupants in a passenger setting and the total number, volume or weight of items in a food setting.
The reconfiguration constraint adds an extra layer to fleet operational strategies that need to include a process for deciding when and where a vehicle should perform reconfigurations.
In the following, we detail the logic implemented within the DRT module to enable the simulation of such a service, the reconfiguration algorithm proposed in this work, and the case study under which the developed features and approaches are experimented.

Configurable vehicle capacities

To allow MoD fleet vehicles to change capacities during the day within MATSim simulations, a *reconfiguration* task type is introduced to vehicle schedules. The vehicle assignment algorithm is then adapted to detect *reconfiguration* tasks while going over schedules and evaluating insertion points, and keep track of the capacity resulting from each *reconfiguration*. In the current implementation, the main assumption behind these tasks is that they are planned at the beginning of the day and, unlike previously existing *stop* tasks, they cannot be shifted by insertion of new ones by the vehicle assignment algorithm.
As a result, the algorithm will not consider inserting any pickup or drop-ff that would delay arrival to a scheduled *reconfiguration*, thus potentially causing requests to be rejected. This can cause the formation of an unavailability window of the vehicle before *reconfiguration*, with an extent that would increase as the vehicle travels farther away from the *reconfiguration* area. This represents a point of attention in the design of capacity reconfiguration algorithms. The implementation structure is modular and generic, allowing the community to implement and evaluate such algorithm in a flexible manner.
In the following, a first demand-dependant capacity reconfiguration algorithm is proposed and used to illustrate the functionality implemented in this work, and provide first insights on the interest of a mixed MoD service.

Reconfiguration algorithm

The reconfiguration algorithm proposed in this work considers the demand observed in the last iteration and attempts to achieve the same temporal distribution of vehicle capacities as the distribution of request types.
Notation: Let C be the set of possible configurations a fleet vehicle can have. Each vehicle $v \in V$ has a starting configuration $c_v \in C$. The fleet size is noted $n_v = |V|$. The goal of the algorithm is to determine the set R of planned vehicle reconfiguration tasks. A reconfiguration $r = (v, t, l, c) \in R$ is characterized by a vehicle $v \in V$, a time t, a location l, and the configuration c in which the vehicle is after the task.
Configuration: the behaviour of the algorithm is configured by two parameters: the reconfiguration interval t_r, which specifies the minimum time between two consecutive vehicle reconfigurations. The day is consequently divided into $n_s = 24h/t_r$ slots of length t_r, noted as $s_0, s_1, \ldots, s_{n_s}$. The other configuration element is the strategy with which the reconfiguration locations are selected. The latter is noted below as the function $\phi(v, t, c)$.
Input: The algorithm takes as input the sets $Req_{i,c}$ of requests emitted in the slot s_i that require a vehicle with configuration c. We assume $\forall c, c' \in C, i \in \{0, \ldots, n_s\}\ c \neq c' \implies Req_{i,c} \cap Req_{i,c'} = \emptyset$.
Process: The algorithm keeps track of vehicle configurations noted as $cap(v, i)$ throughout time slots, starting by setting $cap(v, 0) = c_v$. Then, time slots $s_i \in \{s_0, \ldots, s_{n_s}\}$ are looped over progressively while performing the following:

Request type	Prebooking slack	Max wait time	Max delay on arrival	Pickup time	Drop-off time
Passenger	$300s$	$300s$	$600s$	$60s$	$60s$
Food	$900s$	$3600s$	$300s$	$30s$	$240s$

Table 1: Operational differences between passenger and food requests

- For every $c \in C$, the number $curr(c)$ of vehicles planned to be in configuration c is calculated as $curr(c) \leftarrow |\{v \in V \,|\, cap(v,i) = c\}|$. Similarly, the number $target(c)$ of vehicles that are desired in configuration c is determined proportionally as:

$$target(c) \leftarrow \frac{n_v.|Req_{i,c}|}{\sum\limits_{c \in C} |Req_{i,c}|}$$

  the difference between the current and target values is then calculated as $\Delta(c) = curr(c) - target(c)$

- Looping over capacities with fewer vehicles than necessary ($\{c \in C \,|\, \Delta(c) < 0\}$), that gap is filled by planning reconfigurations one after another determining:
  - The vehicle $v*$ from a configuration with a surplus that has the least number of reconfigurations already planned is selected ($v* = \arg\min\limits_{v \in V} card(\{(v,t,l,c') \in R \,|\, \Delta(c) > 0\})$.
  - A reconfiguration $r = (v*, i.t_r, l, c)$ is planned after determining $l = \phi(v*, i.t_r, c)$
  - Δ is updated for c and for the previous configuration of $v*$

Simulation use case

A simple synthetic use-case is constructed to illustrate a MoD service aiming at satisfying two types of requests. The geographical area of the city of Paris is considered and the two demand types are passengers and food deliveries. The demand is randomly sampled, the origins and destinations follow a uniform distribution whereas the type and departure time are determined by a gaussian mixture. The latter produces 10000 passenger requests and 5891 food ones such as passenger trips produce two equivalent peak time in the morning and in the evening (at 8am and 5pm) and food ones produce a minor peak time at 12pm and a major one concentrating 80% of this particular demand at 8pm.
Furthermore, passenger and food requests are assumed to differ in their operational mechanisms and constraints they have on the MoD system. First, passenger requests are submitted five minutes before the desired departure, while food requests are submitted 15 minutes before. Second, the maximum wait time is set to 5 minutes for passenger requests, while foods can theoretically wait up to one hour for pickup. However, a maximum delay on arrival (in comparison to performing a direct trip) is ensured and is limited to 10 minutes for passengers and to 5 minutes for foods. The last operational differences are durations for pickups and drop-offs. Whereas both take one minute for passengers, the former operation takes 30 seconds and the latter 4 minutes for foods. Note that pickup and drop-off times are not included in maximum wait time and maximum arrival delay. These differences are summarized in table 1
On the service side, two main settings are evaluated and compared. A first setting, labelled *two fleets*, representing a baseline situation involves using two different and independent fleets to serve passengers and perform food deliveries, and sending each request to the appropriate service according to its type. In this setting, vehicles of each fleet are readily adapted to the target requests, hence no reconfiguration is performed.
In the second setting, labelled *mixed fleet*, one fleet serves both request types is simulated using the implemented features detailed above. In this work, the duration of reconfigurations is assumed to be 20 minutes. To investigate the behaviour of the reconfiguration algorithm proposed in above, a sensitivity analysis is performed on the reconfiguration interval t_r with values of 30, 60, 90, and 120 minutes. Moreover, two strategies for determining reconfiguration locations are evaluated: the first randomly selects a location from the network (following a uniform distribution) whereas the second selects the vehicle's initial position on the network, which is also determined randomly in the beginning of the simulation. The idea behind these two strategies is to broadly represent ones that tend to restrict vehicles to specific parts of the network against ones that tend to cause the

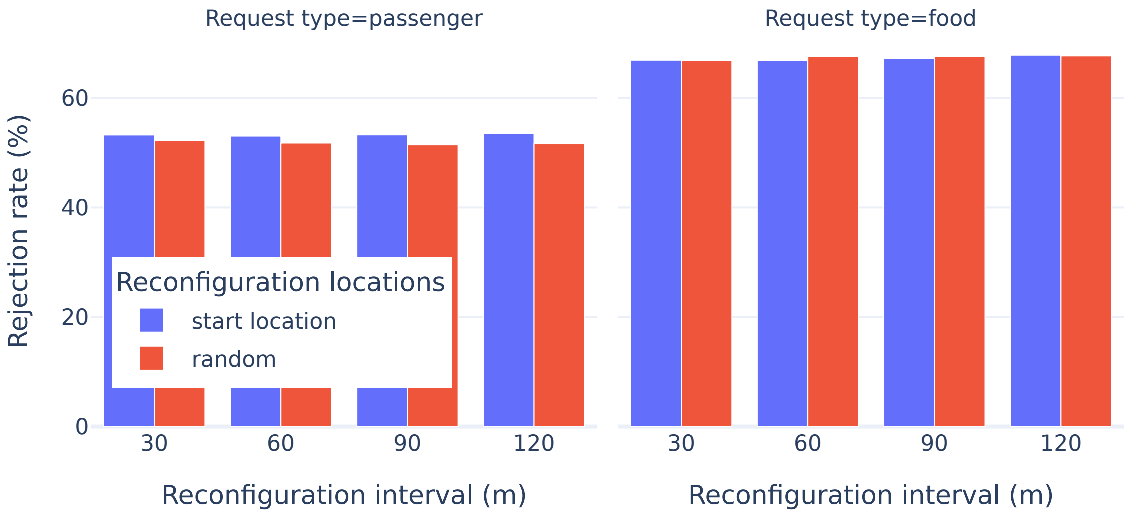

Figure 1: Rejection rates achieved by mixed MoD services with 100 vehicles, varying the reconfiguration interval $t_r \in \{30m, 60m, 90m, 120m\}$ and the reconfiguration selections between vehicle's start location and random.

vehicles to travel extensively. This sensitivity analysis is performed with a rather constrained 100 vehicles fleet to uncover the impact of evaluated parameters.

In order to compare between the two settings, the fleet size is varied between 50 and 700 vehicles with steps of 50 to identify the number of vehicles required to achieve less than 5% of rejections for both request types and assess the potential advantage of using a single reconfigurable fleet. Finally, in all simulations, the service is using the rebalancing strategy depicted in Ruch et al. (2020).

4 Results and discussion

In this section are presented the results of simulations performed in this work. First, the focus is put on the behaviour of the reconfiguration algorithm under the *mixed fleet* described above and a sensitivity analyses of the parameters involved by describing the results and discussing elements specific to this topic. Then, the *mixed fleet* setting is evaluated against the *two fleets* one to assess its advantage, and the comparison results discussed. Finally, general points of improvements upon the work at hand are outlined.

Behaviour of the reconfiguration algorithm

Figure 1 shows the rejection rates achieved by the mixed MoD services, all comprising fleets of 100 vehicles. The first observation is that passenger requests are less rejected than food ones. This can stem from two sources, the first being that the algorithm may not be able to react quickly enough to increases in demand for food deliveries. The second lies in the difference between the operational constraints of food and passenger requests shown in 1. While food deliveries are essentially not subject to a maximum wait time, their maximum allowed delay on arrival is half the allowed for passengers.

Comparing the rejection rates for each of passenger and food requests across reconfiguration intervals and reconfiguration location selection strategies, no substantial differences are observed within the intervals and strategies tested. Moreover, the same minor increases and decreases of rejection rates are not recurrent with other, still constrained, fleet sizes.

Regarding the reconfiguration interval, on the one hand, the potential negative drawback of low values is mitigated by the algorithm's design that attempts to minimize the number of performed reconfigurations. On the other hand, the potential lack of responsiveness of vehicle configurations due to high values is not identified under this demand, where the distribution of requests does not change drastically within 120 minutes.

Regarding the impact of the reconfiguration location strategies, their potential different impacts on global service performance is diminished by the rather small operating area of the city of Paris. Figure 2 shows the distribution of request types during the day in bars, and the distribution of

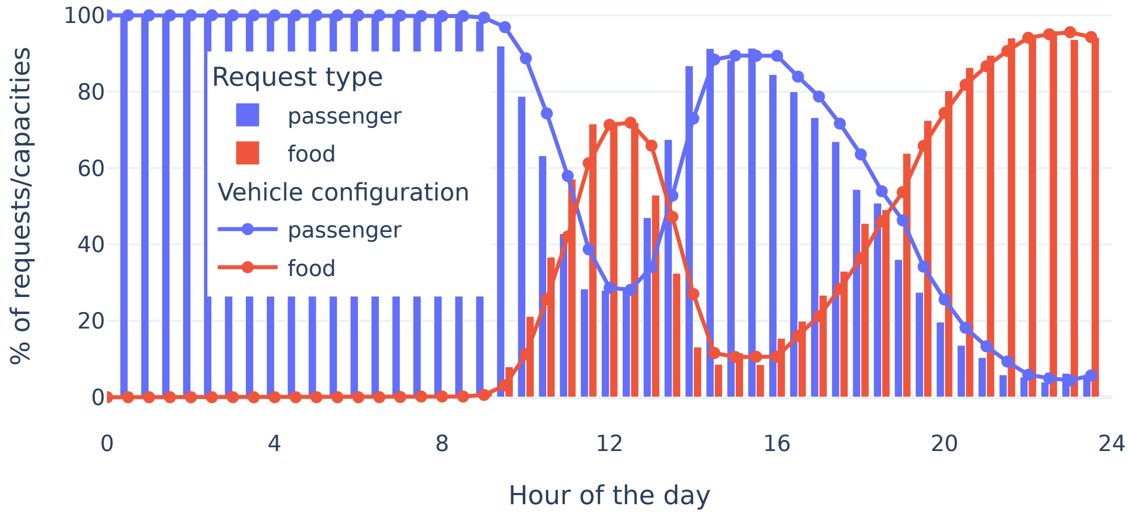

Figure 2: Temporal distribution of request types and vehicle configurations achieved by a mixed fleet of 550 vehicles.

vehicle configuration achieved by the reconfiguration algorithm in lines, under a simulation of 550 vehicles with a reconfiguration interval of 30 minutes and with the random reconfiguration location selection. The results show that the algorithm is able to closely match the temporal vehicle configuration distribution to requests.

Comparing against two separate fleets

Figure 3 shows the fleet sizing results between the two fleet settings with the rejection rates for passengers and for food achieved by each setting. The fleet setting is designated by line style, straight lines refer to *two fleets* whereas dashed ones refer to *mixed fleet*. Request types are designated by colour similarly to Figure 2 with blue lines referring to passengers and red ones to foods. In the *two fleets* setting, the number of vehicles is equally shared between the passenger fleet and the food one. Thus the points at Fleet size = 100 on the straight lines reads as 50 vehicles for the passenger fleet and 50 for the food one.
With a target rejection rate of below 5% in mind, under the *two fleets* setting 300 vehicles are required to meet passenger demand (corresponding to total Fleet size = 600), whereas the food demand requires 350 vehicles (total of Fleet size = 700). In total, serving both demands with two separate fleets requires 650 vehicles. Under the *mixed fleet* setting, 600 vehicles are required to serve both passenger and food demands with less than 5% of rejections.
The results presented above show that the use of a mixed reconfigurable fleet, equipped with the reconfiguration algorithm proposed in this paper, can allow savings in fleet size. Although the conducted experiment shows a reduction of less than 10% in fleet size, this work present a first step in simulations of novel MoD services and further sensitivity analyses and experiments are required to further examine the behaviour of reconfigurable fleets.
Among future research pathways on mixed reconfigurable MoD services, the consideration of different distributions of demand by varying the relative shares of each demand type as well as their relative peak times is of particular interest. Moreover, while synthetic populations are well-developed in the literature for providing, realistic passenger demands, food delivery demand data, and more generally parcels, are less common.

General discussion

In this work, the main indicator used to evaluate the *mixed fleet* setting is the rejection rate. While this indicator allows to encapsulate the quality of the service from the user perspective, more indicators should be considered to also include the operator's perspective such as: total vehicle driven distances, empty vehicle driven distances, occupation rates, operation cost...etc. The latter can be affected if a monetary cost is associated with each vehicle reconfiguration.
Moreover, mixed reconfigurable MoD services allow exploring new areas of operational strategies related to reconfiguration algorithms with a first approach proposed in this paper. An extension

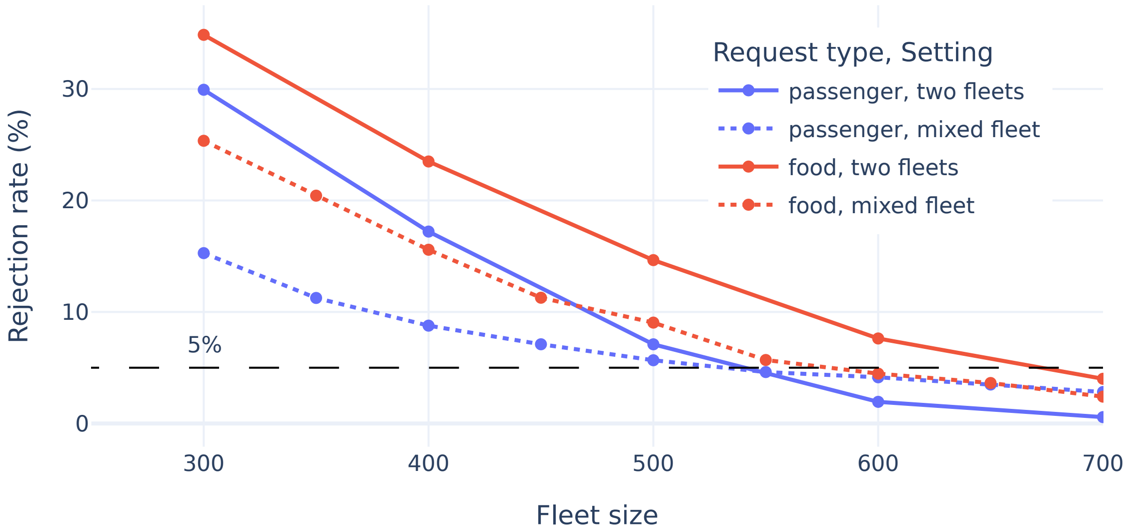

Figure 3: Rejection rates observed for passenger and food requests across various fleet sizes under each of the two settings.

point of the current approach is to consider the distribution of request types per zone and try to locally achieve the corresponding distribution of vehicle configurations, instead of over the whole operating area. Moreover, future contributions will consider the interplay between this particular operational aspect and other ones, particularly rebalancing.

5 Conclusions

To our knowledge, this work is the first to consider mixed reconfigurable MoD fleets with a high level of detail and use a well-established agent-based simulation tool. In such services, a heterogeneous demand (illustrated here with passengers and foods) can be served with a single fleet with the constraint that a vehicle is compatible with only one type of demand at a time and must perform a reconfiguration before serving the other type. An algorithm to plan reconfigurations depending on the demand is proposed and a well-established agent-based mobility simulation tool is extended to allow the simulation of such services and used to evaluate the potential of such a system on a synthetic use case.

The results presented in this work show the interest these services have in reducing overall fleet sizes. Presented results call for more detailed sensitivity analyses of the algorithm parameters, which will be included in future extended versions of the paper at hand. Moreover, this work opens a new area of research on reconfiguration algorithms.

This research is highly reproducible and the technical work performed in this research resents a significant technical contribution to the open-source simulation framework MATSim that allows other researchers to simulate such services and propose new reconfiguration algorithms[1].

References

Bischoff, J., & Maciejewski, M. (2020). Proactive empty vehicle rebalancing for demand responsive transport services. *Procedia Computer Science*, *170*, 739–744.

Fehn, F., Engelhardt, R., Dandl, F., Bogenberger, K., & Busch, F. (2023, March). Integrating parcel deliveries into a ride-pooling service—An agent-based simulation study. *Transportation Research Part A: Policy and Practice*, *169*, 103580. Retrieved 2025-01-17, from `https://www.sciencedirect.com/science/article/pii/S0965856422003317` doi: 10.1016/j.tra.2022.103580

[1] https://github.com/matsim-org/matsim-libs/pull/3627

Golpayegani, F., Gueriau, M., Laharotte, P.-A., Ghanadbashi, S., Guo, J., Geraghty, J., & Wang, S. (2022). Intelligent shared mobility systems: A survey on whole system design requirements, challenges and future direction. *IEEE Access*, *10*, 35302–35320.

Horni, A., Nagel, K., & Axhausen, K. W. (Eds.). (2016). *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press. doi: 10.5334/baw

Hörl, S., & Puchinger, J. (2023, January). From synthetic population to parcel demand: A modeling pipeline and case study for last-mile deliveries in Lyon. *Transportation Research Procedia*, *72*, 1707–1714. Retrieved 2024-12-12, from `https://www.sciencedirect.com/science/article/pii/S2352146523009420` doi: 10.1016/j.trpro.2023.11.644

Jing, P., Hu, H., Zhan, F., Chen, Y., & Shi, Y. (2020). Agent-Based Simulation of Autonomous Vehicles: A Systematic Literature Review. *IEEE Access*, *8*, 79089–79103.

Maciejewski, M., Bischoff, J., Hörl, S., & Nagel, K. (2017). Towards a Testbed for Dynamic Vehicle Routing Algorithms. In J. Bajo et al. (Eds.), *Highlights of Practical Applications of Cyber-Physical Multi-Agent Systems* (Vol. 722, pp. 69–79). Cham: Springer International Publishing. Retrieved 2020-06-10, from `http://link.springer.com/10.1007/978-3-319-60285-1_6` (Series Title: Communications in Computer and Information Science) doi: 10.1007/978-3-319-60285-1_6

Meinhardt, S., Schlenther, T., Martins-Turner, K., & Maciejewski, M. (2022, January). Simulation of On-Demand Vehicles that Serve both Person and Freight Transport. *Procedia Computer Science*, *201*, 398–405. Retrieved 2025-01-17, from `https://www.sciencedirect.com/science/article/pii/S1877050922004665` doi: 10.1016/j.procs.2022.03.053

Narayanan, S., Chaniotakis, E., & Antoniou, C. (2020, February). Shared autonomous vehicle services: A comprehensive review. *Transportation Research Part C: Emerging Technologies*, *111*, 255–293.

Ruch, C., Gächter, J., Hakenberg, J., & Frazzoli, E. (2020). The+ 1 method: model-free adaptive repositioning policies for robotic multi-agent systems. *IEEE Transactions on Network Science and Engineering*, *7*(4), 3171–3184.

Sakai, T., Hara, Y., Seshadri, R., Alho, A. R., Hasnine, M. S., Jing, P., ... Ben-Akiva, M. (2022, February). Household-based E-commerce demand modeling for an agent-based urban transportation simulation platform. *Transportation Planning and Technology*, *45*(2), 179–201. Retrieved 2025-01-17, from `https://doi.org/10.1080/03081060.2022.2084397` (Publisher: Routledge _eprint: https://doi.org/10.1080/03081060.2022.2084397) doi: 10.1080/03081060.2022.2084397

Schaller, B. (2018). The new automobility: Lyft, uber and the future of american cities.

Vosooghi, R., Puchinger, J., Jankovic, M., & Vouillon, A. (2019). Shared autonomous vehicle simulation and service design. *Transportation Research Part C: Emerging Technologies*, *107*, 15–33.