

26th Euro Working Group on Transportation Meeting (EWGT 2024)

Impacts of On-Demand Shuttle System Parameters on its Metrics: A Microsimulation Study

Oytun Arslan^{a,*}, Silja Hoffmann^a

^a*Chair of Intelligent, Multimodal Transport Systems, University of the Bundeswehr Munich, Neubiberg 85577, Germany*

Abstract

The essence of an on-demand shuttle system is an intelligent matching algorithm, where vehicles and passenger requests are matched to each other in an optimized way. Many studies have touched upon this issue by using macroscopic simulation or agent-based simulation. Modelling on-demand shuttle systems in microsimulation can be very advantageous since it is in general a good means to analyze traffic management measures and new mobility concepts. The aim of this paper is to model an on-demand shuttle system in microsimulation environment and analyze the impact of its system parameters which are fleet size, vehicle capacity, maximum waiting time and maximum detour ratio on some metrics such as served requests ratio, shared rides ratio, average waiting time and average detour time.

© 2024 The Authors. Published by ELSEVIER B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>)

Peer-review under responsibility of the scientific committee of the 26th Euro Working Group on Transportation Meeting

Keywords: on-demand shuttle system; microsimulation; matching algorithms; dispatching

1. Introduction

On-demand mobility has gained a lot of popularity in the recent years. Both in urban and rural areas, there is a high tendency to offer this new mobility concept to the passengers. Assmann (2020) investigated the effects of on-demand mobility on sustainable urban mobility based on three indicators: traffic flow, emissions and usage of spaces. Harz and Sommer (2021) tried to answer the question how to enhance the mobility options in the rural areas and analysed different on-demand offers.

Modelling on-demand systems, especially ride-pooling where requests can be bundled together if more efficiency is possible, occurred up to now in different platforms. Thomsen (2022) implemented a ride-sharing algorithm in a

* Corresponding author. Tel.: +49-89-6004-2525.

E-mail address: oytun.arslan@unibw.de

macroscopic model. Bischoff et al. (2017), Engelhardt and Bogenberger (2021) and Engelhardt et al. (2019, 2020 and 2022) used agent-based simulation to model ride-pooling. Apart from these, some studies such as Kümmel (2016) and Dandl et al. (2017) worked on ride-hailing, where exclusive taxi systems were analyzed elaborately. Kümmel (2016) modelled his system in MATLAB and Dandl et al. (2017) in microscopic simulation tool Aimsun. Tang and Armellini (2021) and Armellini (2021) used SUMO microsimulation to simulate demand responsive systems, however in a static manner where all requests are known in advance. Arslan and Hoffmann (2023) aimed to close a research gap that a microsimulation tool for dynamic ride-pooling has not been utilized so far and modelled such a system in microscopic simulation tool Vissim.

Bischoff et al. (2017) utilized an insertion-based algorithm to the taxi data set of Berlin in an agent-based simulation (MATSim) and found out that overall vehicle kilometers traveled could be reduced by 15-20 %. Engelhardt et al. (2019) introduced an autonomous on-demand ride-pooling system by using batch assignment and allowing reassignments, also in an agent-based simulation. According to their findings, ride-pooling has a negative effect on traffic, increasing vehicle kilometers in case of low demand. On the other hand, with higher demands vehicle kilometers could be reduced by up to 18%, predominantly on the major roads. Engelhardt et al. (2020) extended their previous work and added a heuristic to improve the computation time while sacrificing some efficiency. This heuristic could speed up the algorithm by a factor of 8 while keeping the number of served requests almost constant. Liebchen et al. (2020) focused on the measurability of a good ride-pooling system. They considered only the operator point of view and defined a KPI called system efficiency which is equal to the direct trip distances of passengers (benefit) divided by the total mileage (cost). Bilali et al. (2019a) studied the impacts of detour time, waiting time and boarding time on the shareability of requests. In a following research, they introduced an advanced booking system as well and analysed the impact of reservation time (Bilali et al. 2019b). They showed that the shareability increases with increasing detour time, waiting time and reservation time.

The objective of this paper is to analyse the impacts of on-demand shuttle system parameters such as fleet size, vehicle capacity, maximum waiting time and maximum detour ratio on system metrics such as served requests ratio, average occupancy and average waiting time. The microsimulation tool Vissim (PTV, 2024) was used for the analysis. This paper is based on the study of Arslan and Hoffmann (2023), which we refer to for further details. The application of microsimulation will bring the following advantages to the research community:

- facilitate a testbed to investigate various vehicle-request matching algorithms,
- provide a simulation platform for other on-demand modes as well such as micro-mobility and taxis
- enable interaction between on-demand vehicles and other transport modes,
- allow a holistic microscopic traffic simulation considering all transport modes.

2. Methodology

In an on-demand shuttle service, a passenger sends a request to the system, stating that he wants to travel from A to B at a certain point of time. This request can either be for the time being (an immediate trip) or for some time in the future (advanced booking). For the advanced booking case, the requests are known by the system in advance, and the vehicle tours between the pick-up and drop-off points of requests can be calculated in a static manner. This phenomenon is known as static dial-a-ride-problem (DARP) (Psaraftis, 1980). Some studies developed tools to calculate best routes by solving a static DARP (Armellini 2021, Tang and Armellini 2021). In the immediate trip case, on the other hand, requests are revealed one-by-one within the time course, which necessitates a dynamic decision mechanism, since the requests are not known in advance. Therefore, the system has to react “on-line”. This is known as dynamic DARP, and several studies in the literature developed methodologies to solve such problems (Pillac et al. 2013, Alonso-Mora et al. 2017, Engelhardt et al. 2019, 2020 and 2022).

In dynamic DARP, there are also different approaches about how to deal with the incoming requests. One approach is using the spontaneous assignment, which means that the requests are processed one by one as soon as they appear in the system. The matching decision is done immediately, and this matching will not be changed later. Spontaneous assignment has the advantage of replying the customer immediately and having a less computational complexity. The disadvantage, however, is that the incoming requests are not optimized globally and a potential combination with other upcoming requests can be missed. The other approach, known as batch assignment, processes the requests in batches

within certain time periods. Requests arriving at the beginning of this period can experience longer response times, yet the optimization potential to combine different requests is then higher.

This paper deals with the dynamic DARP in a spontaneous assignment context. A rejection of the request is also possible if exceeding the limits for request waiting time or detour ratio is inevitable. At this stage, the rejected request leaves the system immediately, a further consideration is not the case.

2.1. Creating valid permutations

Every time a new request arrives in the system, it will be checked which vehicle can take it over. Furthermore, for a single vehicle, there can be different pick-up and drop-off point permutations for the already assigned requests and the new request, depending on the current status of the vehicle. If for instance 2 requests (A and B) are already in the vehicle, 1 request (C) is already assigned but not yet picked-up, and a new request (X) arrives in the system at that moment, all possibilities of a permutation of 6 points (these are $A_D, B_D, C_P, C_D, X_P, X_D$ where the indices are P: pick-up, D: drop-off) have to be considered. For each permutation, a route legitimacy check needs to be carried out, i.e. checking whether the drop-off point of a request happens not before its pick-up point.

Assuming that i is the number of in-vehicle passengers in a vehicle and t is the number of requests to be picked up by a vehicle, a general formula for the number of valid permutations to transport the in-vehicle passengers, to-be-picked-up passengers as well as the new request can be stated as:

$$\text{Total number of permutations} = \frac{(i+2t+2)!}{2^{t+1}} \quad (1)$$

According to this formula, the above example ($i=2$ and $t=1$) leads to 180 permutations in total. Then another check regarding the vehicle capacity must be conducted in order to avoid an excess of capacity in any permutation. Here, if the number of requests inside the vehicle is more than its capacity at any point of time, the concerned permutation will be omitted.

Following examples illustrate the generation of valid permutations for the vehicle mentioned above, assuming a vehicle capacity of 3:

- Permutation $A_D \rightarrow X_D \rightarrow C_P \rightarrow C_D \rightarrow X_P \rightarrow B_D$ is not valid since drop-off of X (X_d) happens before its pick-up (X_p).
- Permutation $C_P \rightarrow X_P \rightarrow B_D \rightarrow X_D \rightarrow C_D \rightarrow A_D$ is not valid since the number of requests inside the vehicle goes up to 4 after the pick-up of X (X_p), hence exceeds the vehicle capacity.
- Permutation $B_D \rightarrow X_P \rightarrow C_P \rightarrow C_D \rightarrow X_D \rightarrow A_D$ is valid since route legitimacy is guaranteed and vehicle capacity is never exceeded.

2.2. Filtering out permutations

Among valid permutations, another filtering mechanism is used in terms of waiting times and detour ratios of passengers. Permutations resulting in high waiting times or detour ratios are cancelled out since they are not acceptable for passengers at all, even if the permutation globally turns out to have a good overall point. Here maximum waiting time is the maximum acceptable time for a passenger to be picked-up after sending a request. The detour ratio is the ratio of the actual travel time (increased due to pooling) to the direct travel time. These two values are used as parameters in the following case study, to see their impact on different metrics.

2.3. Calculation of overall points

The procedure mentioned above is applied to all vehicles in the system in order to create a pool of valid vehicle-permutation combinations. The aim is then to find the best solution from this pool. For this, both the passenger's point of view and the operator's point of view are analyzed. Passengers have mainly two discouragements in ride-pooling: i) waiting time for the vehicle, ii) detour time due to pooling. While the in-vehicle passenger suffers only from the latter, the new request and the to-be-picked-up passenger suffer from both, depending on the situation. Generally, an "acceptability ratio" for each corresponding passenger and the new request is calculated via the following formula:

$$AR = \frac{\text{direct travel time}}{\text{waiting time} + \text{in_vehicle travel time}} \quad (2)$$

AR can have a maximum value of 1 if the passenger experiences no waiting time (i.e. the vehicle is just at his pick-up point) as well as no detour (hence direct travel time is equal to in_vehicle travel time). The average of the acceptability ratios of regarded passengers returns a value about the passenger's point of view (AR_{avg}).

From the operator's point of view though, the aim is to bundle the requests as much as possible, leading to a calculation of a pooling ratio given in the following formula:

$$PR = \frac{\text{total passenger minutes}}{\text{total vehicle_seat minutes}} \quad (3)$$

In this formula, total passenger minutes is the sum of the time that all passengers in a tour spend in the vehicle. Total vehicle_seat minutes is the time a vehicle spends to complete the whole tour, multiplied by the vehicle capacity. PR therefore can have a maximum value of 1, if a vehicle makes the whole tour with its full capacity.

The overall point of a permutation is calculated by the addition of two ratios, which leads to $OP = AR_{avg} + PR$. The higher the OP value, the better the chances are for the permutation. We refer to (Arslan, Hoffmann 2023) for a more elaborate description of the algorithm and an exemplary illustration.

2.4. Application in microsimulation

This on-demand system was modelled in a microsimulation environment, in the simulation software Vissim from the company PTV AG (2024). Vissim does not offer a standard solution for on-demand approach, nevertheless, it offers an external programming via an interface called COM. With the COM interface, codes such as in Python can be applied to the model, enabling an external control of the system.

In order to mimic the demand, a request pool was created which consisted of single requests that involved the information of an origin stop, a destination stop, and the time of the request. Instead of generating the requests randomly both for the request occurrence times and origin-destination relations, as was the case in Arslan and Hoffmann (2023), the trip base of the micromobility sharing system of the campus was utilized. This platform, known as MORE Sharing, is operated on a free-floating system and enables users, who are solely the university members, to book e-scooters, e-bikes, e-cargo bikes, city bikes and e-mopeds for their trips (Pobudzei, 2023). The sharing data was analysed from the system launch (06.03.2023) until 05.11.2023, for an 8-month period. For the shuttle system demand, the day with the highest demand was taken as base, which led to 385 requests from 7 a.m. until midnight. The simulation model was then fed with these requests, again via the COM interface. The demand profile, showing how the demand varies over hours of the selected day is shown in the following figure.

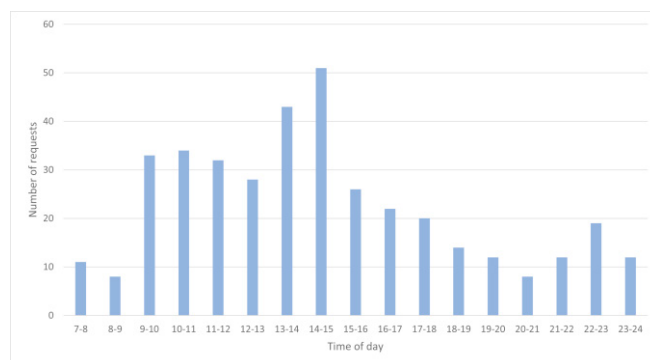


Fig. 1. Demand profile of the selected day

The microsimulation then runs with these inputs, vehicles match with the requests one-by-one (or requests are rejected), vehicles are sent to different locations according to their assignments and finally, the simulation delivers

outputs for the analysis.

3. Case Study

The algorithm was applied in a microscopic simulation of University of the Bundeswehr Munich campus. The campus has an area of 140 hectares and consists of many facilities fulfilling different purposes such as various research buildings, administrative buildings, a university library, dormitories, a post office, sports facilities, sanitary facilities, a kindergarten, and a canteen. 16 shuttle stops were decided based on the campus entry/exit points, surveys conducted among university members and the most important campus locations (Fig. 2). Moreover, a depot was implemented for the shuttles where they all start the simulation at the beginning. During the simulation, in case shuttles did not have any new task, they waited at the last stop in the idle mode. Only when a stop was occupied by a shuttle while another one wanted to approach there i.e. due to drop-off of a request, the former shuttle was sent to the depot to make room for the latter. Since the demand profile showed an even distribution within the campus and the area of campus is not too large, no rebalancing or relocation strategies were applied in this study.

Constant parameters used in the study are as follows:

- Boarding/alighting time: 30 seconds
- Shuttle maximum speed: 15 km/h (due to autonomous operation)
- Simulation period: 17 hours (7:00 – 24:00)
- Request pool: 385 requests based on More Sharing
- Simulation time step: 1 second

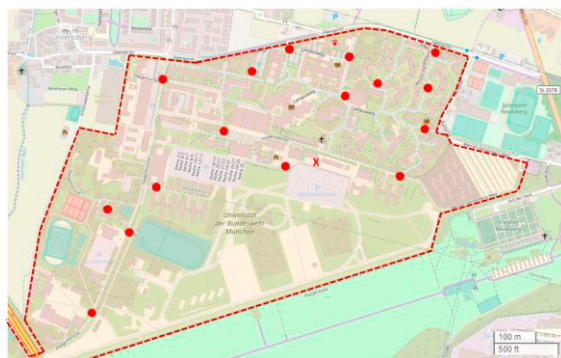


Fig. 2. Campus of the University of the Bundeswehr Munich with 16 shuttle stops shown in red circles and X as depot (Source: OpenStreetMap).

Apart from constant parameters, some parameters were assigned as variables in order to observe their impacts on the metrics of the on-demand shuttle system, which is the main objective of this paper. The variable parameters used in the study and their values in different scenarios are as follows:

Table 1. Variable system parameters used in the study.

Variable parameters	Parameter values
Fleet size	1, 2, 3, 4
Vehicle capacity	1, 2, 3, 4
Maximum waiting time (min)	3, 5, 10
Maximum detour ratio	1.4, 1.7, 2

4. Results

Different simulation runs were performed by changing the variable system parameters. Metrics investigated are the ratio of served requests, ratio of shared rides, average waiting time and average detour time.

As mentioned in the previous sections, a request can be rejected by the system if the maximum waiting time or maximum detour ratio is exceeded. This can sometimes be inevitable since there is no suitable shuttle available, yet it is an undesired situation and needs to be avoided by the operator. It can be observed from the Fig. 3 that bigger fleet size leads to higher ratio of served requests in each case, whereas the jump from the fleet size of 1 to 2 has been the highest in every case. Also with the increase in vehicle capacity, ratio of served requests increase, too. However, an increase of capacity from 3 to 4 mostly does not cause a difference, this can be due to the fact that 4 passengers are very rarely in a vehicle at the same time. Having a higher maximum waiting time also has a positive impact on served requests, which is expected since more vehicles are then available to the requests. On the other hand, higher maximum detour ratio hardly has an effect, the ratio stays almost constant along the maximum detour ratio parameter.

The highest service rate (ca. 99%) is reached with a fleet size of 4, vehicle capacity of 4, maximum waiting time of 10 min and maximum detour ratio of 2. It can be concluded that with these constellation, almost all requests will be served in the on-demand shuttle system.

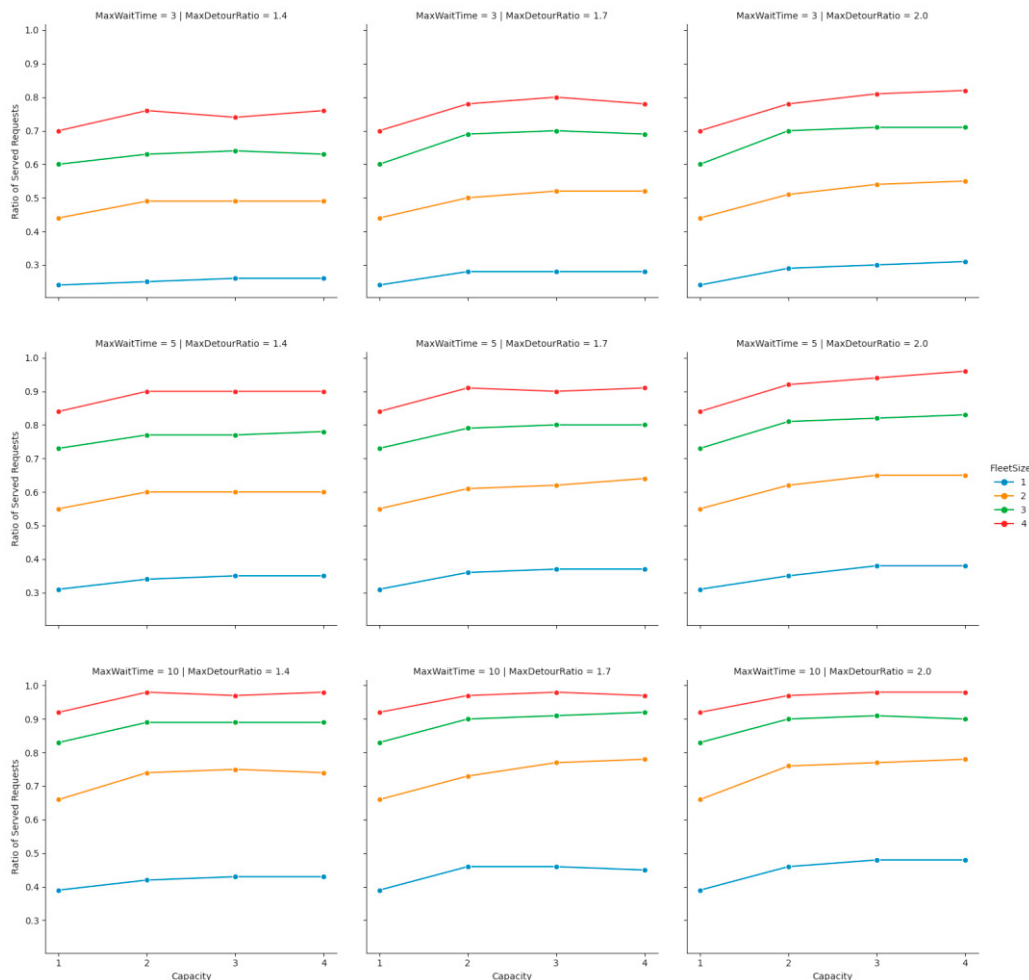


Fig. 3. Impact of the variable system parameters on the metric “Ratio of served requests”.

Fig. 4 shows the relation of the metric “ratio of shared rides” with the system parameters. Shared rides are defined as the number of requests who share a ride, divided by the total number of served requests. Since there will be no

pooling, hence no shared rides with a capacity of 1, the average detour times are 0 and were not displayed in the figure. Parallel to the findings of Bilali et al. (2019b), shareability increases with increasing maximum waiting time and maximum detour time. Generally, fleet size has a negative impact on the shareability, since more requests tend to be matched with “free” shuttles instead of being pooled. At the right bottom plot, with a fleet size of 1, ratio of shared rides increases almost linearly with capacity. Yet, sometimes it is observed that the capacity has a negative effect on this metric. This can be due to the lower number of served requests for less capacities. The highest ratio (0.72) is achieved with a fleet size of 1, vehicle capacity of 4, maximum waiting time of 10 min and maximum detour ratio of 2.

In Fig. 5, the impact of capacity and fleet size on average waiting and detour time can be observed. For constant maximum waiting time (5 min) and maximum detour ratio (2), we can see that higher capacity and bigger fleet size mostly reduces the average waiting time, which is equal to total waiting time divided by the number of served requests. A somewhat increase with capacity can only be observed for fleet size 1, this can be due to the requests that are now served instead of being rejected, and therefore increasing the total waiting time proportionally. Over all the cases, average waiting time is approximately 2.5 min, corresponding to the half of the maximum waiting time of 5 min.

Detour time takes place when an in-vehicle passenger does not travel directly to his destination due to pooling in the system. It is calculated as the realized trip time minus the direct trip time. As in Fig. 4, with a capacity of 1, the average detour times are 0 and were not displayed in Fig 5b. For average detour time, fleet size again has a decreasing effect, on the other hand capacity has an increasing effect. With bigger fleet size, less pooling is observed and detour times are lower, yet with higher capacity, pooling increases and also the detour times.

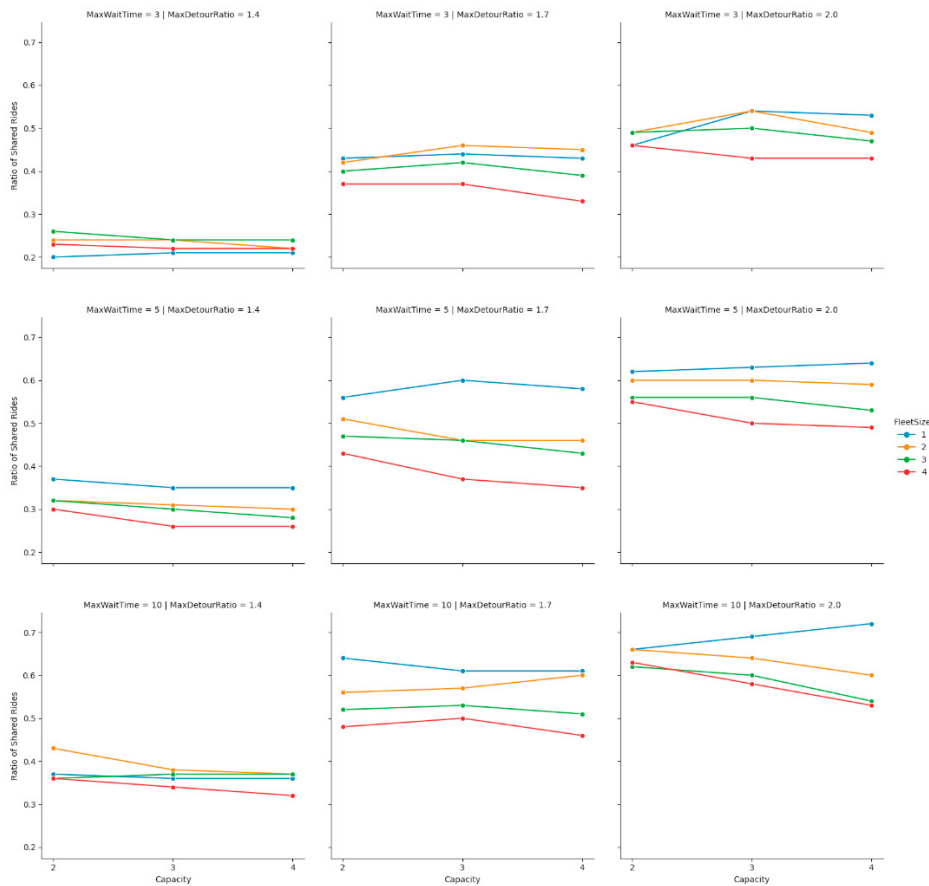


Fig. 4. Impact of the variable system parameters on the metric “Ratio of shared rides”.

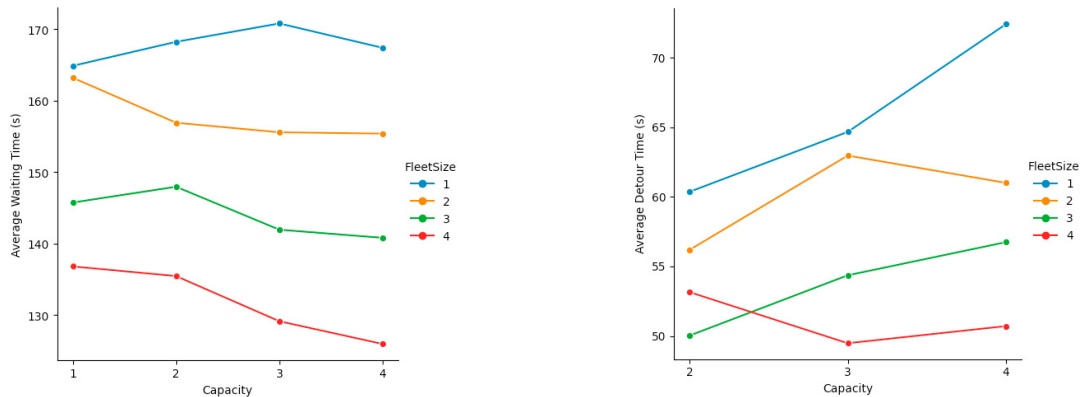


Fig. 5. Impact of capacity and fleet size on the metrics a) “Average waiting time” and b) “Average detour time”, for maximum waiting time of 5 min and maximum detour ratio of 2.

5. Conclusions and Future Work

This study aims to model an on-demand shuttle system in microsimulation environment and analyze the impact of its system parameters which are fleet size, vehicle capacity, maximum waiting time and maximum detour ratio on some metrics such as served requests ratio, shared rides ratio, average waiting time and average detour time. 385 requests during a whole day were matched with vehicles using a spontaneous assignment algorithm.

It was observed that with a constellation of a fleet size of 4, vehicle capacity of 4, maximum waiting time of 10 min and maximum detour ratio of 2, almost all the requests during the day could be served. Additionally, ratio of shared rides increased with higher maximum waiting time and maximum detour time. Furthermore, higher capacity and bigger fleet size mostly reduced the average waiting time; yet for average detour time, fleet size had a decreasing effect, on the other hand capacity had an increasing effect.

In a future research work, a different assignment algorithm will be applied, namely batch assignment, where the requests are assigned to vehicles in certain batches (time periods), rather than immediately. This can lead to more efficient matches since the search space is then greater.

In the current algorithm, the shortest paths are calculated based on the distance between nodes. However, an ideal approach would be considering the traffic situation as well. As an advantage of the microsimulation tool, the current traffic situation can be extracted, analyzed and the weightings of the edges can be modified accordingly, leading for instance to a more expensive edge in case of congestion. In this way, the Dijkstra algorithm (used for shortest path calculation) can be forced to find less congested connections instead of the shortest ones, which can again be a topic for further research.

Acknowledgements

This research is funded by dtcc.bw – Digitalization and Technology Research Center of the Bundeswehr. The dtcc.bw is funded by the European Union – NextGenerationEU.

References

- Alonso-Mora J., Samaranayake S., Wallar A., Frazzoli E., Rus D., 2017. On-Demand High-Capacity Ride-Sharing via Dynamic Trip-Vehicle Assignment. *Proceedings of the National Academy of Sciences*, vol. 114, no. 3, p. 462–467.
- Armellini M.-G., 2021. A Tool for Simulating Demand Responsive Transport System in SUMO. In *7th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pp. 425-429.

- Arslan O., Hoffmann S., 2023. Implementation of a Spontaneous Matching Algorithm for On-Demand Shuttle Systems in Microsimulation. In 9th IEEE International Smart Cities Conference, Bucharest.
- Assmann C., 2020. Systemanalyse der Wirkungen von On-Demand-Mobilitätsdienstleistungen im Kontext nachhaltiger urbaner Mobilität. Dissertation in Technische Universität München.
- Bilali A., Dandl F., Fastenrath U., Bogenberger K., 2019a. Impact of Service Quality Factors on Ride Sharing in Urban Areas. In 6th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS).
- Bilali A., Dandl F., Fastenrath U., Bogenberger K., 2019b. An Analytical Model for On-Demand Ride Sharing to Evaluate the Impact of Reservation, Detour and Maximum Waiting Time. In 22nd IEEE Intelligent Transportation Systems Conference (ITSC).
- Bischoff J., Maciejewski M., Nagel K., 2017. City-wide Shared Taxis: A Simulation Study in Berlin. In 20th IEEE Intelligent Transportation Systems Conference (ITSC).
- Dandl F., Bracher B., Bogenberger K., 2017. Microsimulation of an Autonomous Taxi-System in Munich. In 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), pp. 833–838.
- Engelhardt R., Dandl F., Bilali A., Bogenberger K., 2019. Quantifying the Benefits of Autonomous On-Demand Ride-Pooling: A Simulation Study for Munich, Germany. In 22nd IEEE Intelligent Transportation Systems Conference (ITSC).
- Engelhardt R., Dandl F., Bogenberger K., 2020. Speed-up Heuristic for an On-Demand Ride-Pooling Algorithm. Available at <https://arxiv.org/pdf/2007.14877v1.pdf>.
- Engelhardt R., Dandl F., Bogenberger K., 2021. Benefits of Flexible Boarding Locations in On-Demand Ride-Pooling Systems. In 7th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), pp. 336-341.
- Engelhardt R., Dandl F., Bogenberger K., 2022. FleetPy: A Modular Open-Source Simulation Tool For Mobility On-Demand Services. Available at <https://arxiv.org/pdf/2207.14246.pdf>.
- Harz J., Sommer C., 2021. Erfolgsfaktoren und Hemmnisse des integrierten Ridesharing in ländlichen Räumen. In HEUREKA Konferenz – Optimierung im Verkehr und Transport.
- Kümmel M., 2016. Taxis, Passengers and Stable Marriage – Stable Simultaneous Assignment of Taxis to Passenger Booking Requests. Dissertation in Technische Universität München.
- Liebchen C., Lehnert M., Mehlert C., Schiefelbusch M., 2020. Ridepooling-Effizienz messbar machen. In *Nahverkehr* 09/2020, pp 18-21.
- Pobudzei M., Kemmerzehl R., Ulusoy C., Hoffmann S., 2023. Beyond the City: Testing Shared Micromobility with MORE Sharing. 25th Euro Working Group Transportation Conference (EWGT), Santander.
- Pillac V., Gendreau M., Gueret C., Medaglia A.L., 2013. A Review of Dynamic Vehicle Routing Problems. *European Journal of Operational Research* 225(1): 1-11.
- Psaraftis H., 1980. A Dynamic Programming Solution to the Single Vehicle Many to Many Immediate Request Dial-a-ride Problem. *Transportation Science*, Vol. 14, No. 2.
- PTV AG, 2024. VISSIM Version 2024. Karlsruhe: PTV AG. Available at <https://www.ptvgroup.com/de/loesungen/produkte/ptv-vissim/>.
- Tang Q., Armellini M.-G., 2021. An Ant Colony Algorithm with Penalties for the Dial-a-ride Problem with Time Windows and Capacity Restriction. In 7th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), pp. 425-429.
- Thomsen N., 2022. Implementing a Ride-Sharing Algorithm in the German National Transport Model DEMO. In Transportation Research Board (TRB) 101st Annual Meeting, Washington DC.