Simulation-based assessment of fleet control algorithms for autonomous mobility on demand systems for the Zurich case

Date of submission: 2017-08-01

Sebastian Hörl (joint first co-author)
IVT, ETH Zürich, 8093 Zürich, Switzerland
phone: +41 44 633 38 01
sebastian.hoerl@ivt.baug.ethz.ch

Claudio Ruch (join first co-author)
IDSC, ETH Zürich, 8092 Zürich, Switzerland
clruch@idsc.mavt.ethz.ch

Kay W. Axhausen
IVT, ETH Zürich, 8093 Zürich, Switzerland
phone: +41-44-633 39 43
axhausen@ivt.baug.ethz.ch

Emilio Frazzoli
IDSC, ETH Zürich, 8092 Zürich, Switzerland
phone: +41 44 632 79 28
emilio.frazzoli@idsc.mavt.ethz.ch

Words: 6776 words + 7 figures = 8526 word equivalents

ABSTRACT

The performance of four different dispatching and rebalancing algorithms for the control of an Autonomous Mobility On-Demand system is evaluated in simulation. The case study conducted on an agent-based simulation scenario of the city of Zurich shows that the right choice of control algorithm not only minimzes customer waiting times, but also offers large economical benefits to the operator. For an average waiting time at peak hours of five minutes the most performant algorithm would allow the operator to offer his service for around 0.45 CHF per km, which is more expensive than using a private car today. Yet it is significantly cheaper than a conventional taxi. The results show that autonomous mobility on demand service can be offered while maintaining a higher fleet occupancy than with private cars today. Simulation also confirms that the application of intelligent rebalancing algorithms does decrease the average wait time in the system, but does not necessarily increase the total amount of miles driven.

INTRODUCTION

The rapid techological development in recent years has led to the point where automated vehicles are tested in various pilot projects around the world, e.g by nutonomy Inc. in Singapore *(1)*. They promise to increase road capacity and speeds [TODO include this cite tientrakool, friedrich] and would give access to mobility to formerly inhibited user groups (TODO cite Victoria). On the flipside an increase of vehicle miles travelled (VMT) is expected due to empty rides (cite Litman), and the general increase of users has the potential to clog road in the urban environment even more than today (cite Becker, Meyer). Hence, the net effects on the transport system, environment and society are unclear. Simulations, such as the one presented in the work at hand can help to better understand the impact of future developments in vehicle automation.

A number of studies in recent years debated the feasibility of an autonomous mobility on demand (AMoD) system (see Related Research). With such a system travellers would not need to own their own car, but could call an automated vehicle [AV] to pick them up at any location and bring them to their desired destination. For the customer this would offer the comfortability of an individual taxi service for a fraction of today's cost. It is predicted that the costs of using the service on a daily basis heavily compete with privately owned cars and even public transit, depending on the use scenario [TODO cite cost paper].

The success of an AV operator would depend on the pricing of his service as well as the wait and travel times that he is able to offer. While high prices may restrict the user group drastically, long wait times may have the same effect if they make travelling less predictable than before. Both quantities are inherently linked by the way the fleet is operated: If wait times should be minimized, vehicles should be at all times present where the demand is expected. This makes it necessary to to relocate them without a passenger on-board, which directly translates to costs for the operator. Furthermore both quantities are also linked to the vehicle fleet size that heavily influences both cost and wait times.

In the present study we contriubte to research on AMoD system as follows: We (a) present a simulation scenario of a fleet of automated taxis for Zurich, Switzerland, based on the MATSim framework (cite Horni), we (b) test and compare four different dispatching and rebalancing algorithms from literature for different fleet sizes, (c) analyse the results in terms of customer acceptance and (d) compare our results with theoretical predictions for fleet sizing.

The remainder is structured as follows: First, an overview of related search is given, then the simulaton scenario and environment are introduced, as well as the proposed fleet control algorithms. Thereafter, simulaton results are presented and analysed, followed by a discussion of our findings.

RELARED RESEARCH

Two-way mobility on demand systems, e.g. car sharing schemes like Mobility in Switzerland *(2)* are a well-established part of the modal share of many cities. These schemes offer flexibility, competitive prices and good service levels. However, their popularity is heavily limited by the fact that the vehicle has to be dropped off at the origin of the journey. On the contrary, in one-way mobility on demand systems customers can travel with a vehicle (e.g. autonomous car or bike) from any origin to any destination in the city wich dramatically increases the flexiblity of these systems.

The price for the increased flexibilty is system inbalance. Due to the spacio-temporal and in general unbalanced characteristics of travel demand, vehicles tend to accumulate at certain locations and get depleted at others. Furthermore system inbalance is not an exception but occurs for most demand patterns. This can be seen for instance using queuing-theoretical arguments as shwon in *(3)*.

System inbalance leads to drastically decreased service levels and must be countered with the targeted repositioning of vehicles from overfull to empty areas of the city. This repositioning of vehicles represents a significant contribution to the operational cost of operators and therefore various strategies have been tried to minimize the rebalancing effort. For intance in bike-sharing schemes trucks are used to move vehicles from full to empty stations, in *(4)* algorithms have been proposed to route these trucks at minimal cost. In *(5)* price incentive controllers are proposed to encourage customers to travel to depleted stations at the end of their trip. Rebalancing was also researched for car sharing schemes, e.g. in *(6)* a scheme is proposed to reposition the rebalancing drivers for one-way car sharing schemes in an optimal way. The decisive difference of autonomous mobility on demand systems to the previous two cases is that the vehicles can reposition themselves without the use of transporting trucks or auxiliary drivers. Therefore rebalancing can be carried out more efficiently and with more degrees of freedom.

Rebalancing of autonomous mobility on demand systems was first presented as a research problem in *(7)*. Optimal rebalancing flows for the vehicles are obtained by solving a linear program. In *(3)* the relation to queuing theoretical concepts was established. In *(8)* the relation of the rebalancing effort to the underlying distributions of origins and destinations was established and it was shown that for general distributions the total minimal rebalancing distance is strictly more than zero. In *(9)* the rebalancing problem was solved with a model predictive control algorithm which performs well but does not scale to large systems.

Most of these algorithms were tested on simplified traffic simulations that capture the main characteristics but do not allow the same level of detail as agent based traffic simulations like MATSim. For such simulation platforms various results exist which are presented in the following paragraphs. Most of them do not implement and compare the algorithms mentioned above which is an important contribution that we make in this work.

Spieser et al. *(10)* present a systematic approach to the design of an autonomous mobility on demand system that is able to serve the entire travel demand of Singapore with a fleet of automated shared vehicles. Analytic results are used to compute both the minimal number of vehicles needed to stabilize the number of open requests as well as the amount of vehicles that is needed to provide an acceptable level of service. The authors conclude that a fleet size of 25% of today's vehicle fleet would be able to offer average wait times of around 15 minutes and could half the external and internal costs of mobility. The study does not compare different fleet control algorithms and does not elaborate on whether congestion effects have been taken into account.

Fagnant et al. *(11)* present a case study for Austin, Texas which focuses on the use of shared autonomous vehicles with ride-sharing capabilities, i.e. vehicles that can transport more than one customer under some circumstances. The scenario presented on vehicles with unit capacity yields that 10% of today's vehicle fleet could serve the entire demand with average wait times of 4.49 min.

(12) present a case study for New Jersey also focused on the potential for ride-sharing. The trips generated by a population of $8,791,894$ individuals in New Jersey are covered by walking and biking if the distance is less than a mile. All other trips are either served by the New Jersey train system, by autonomous taxis or both. The study concludes that the ride-sharing potential is large, especially during rush-hour and autonomous vehicles could significantly reduce congestion levels in the city. The required fleet size is not commented as well as the influence of the

rebalancing and dispatching strategy for the fleet.

In *(13)* the authors present a study on the effects of introducting autonomous taxis and autonomous shared taxis to the city of Lisbon, Portugal. The agent-based simulation includes 1.2 million trips and three scenarios: a baseline scenario showing the current situation and two scenarios where private car, taxi and bus trips are replaced by autonomous taxis and autonomous taxis and shared taxis respectively. The fleet size of autonomous (shared) taxis is set at 4.8% of the baseline vehicle fleet. In these scenarios about 50-70 % of trips are serviced by the autonomous (shared) taxis which increases the vehicle occupancy from 50 mins to 12.87 h on average per day. The authors conclude a decrease in cost by 55% per kilometer, highly increased transportation accessibility in the city and carbon emission reductions of almost 40%. The simulation does not consider the changes on traffic density parameters resulting from self-driving vehicles. Furthermore the demand choice of the agents is static and according to preset parameters. Finally the fleet control (rebalancing and dispatching) for the (shared) autonomous taxis is implemented based on heuristics and a local gradient based optimization method.

Boesch et al *(14)* investigate a scenario of the greater Zurich region in Switzerland. They use a demand pattern for private vehicles generated with MATSim: 1.3 million private vehicle users out of a total of 2.1 million agents generate 3.6 million trips. This demand profile generated with the co-evolutionary algorithm inherent to MATSim is then post-processed in a static simulation where $1 - 10\%$ of the car trips are served by $10 - 100\%$ of the total number of substituted users. The authors conclude that approximately 30 % of the substituted fleet can serve almost 100% of the substituted requests within less than 10 mins wait time. If this threshold wait time is surpassed, then the request is dropped. The limitations of the results are that no rebalancing or dispatching is taking place, furthermore network routing is not considered, travel times are based on Euclidean distance and a scaling factor. The demand profile is static and does not vary depending on service times, congestion rates and performance of the modes.

In contrast to the study for Zurich presented above, a case study for Berlin presented in *(15)* takes into account dynamic demand. It considers a city-wide replacement of private vehicles with autonomous taxis. The dispatching of the car works according to a policy that distinguishes between oversupply (more available vehicles than open requests) and undersupply and matches the closest vehicle to an appearing request, the next available vehicle to the closest request respectively. Using this strategy called single heuristic dispatcher in our work, the authors are able to serve 4.7 million requests generated by 1.1 million car users with a fleet of $100,000$ autonomous vehicles. The recorded average wait time for this case is about 2.5% minutes and the 95% quantile approximately 8.5 minutes. The resulting sharing factor is approximately 10 to 12. The study is one of the first large-scale dynamic simulations of a shared autonomous taxi system, however it does not consider different rebalancing and dispatching strategies and it does not rigorously evaluate the performance metrics of the autonomous vehicle fleet as we do in [XYZ].

CONTROL OF AN AMOD SYSTEM

An AMoD service is can only be realized if it is attractive to customers. More specifically, it can only be maintained if a sufficient number of customers wants to use the service and the service is profitable for the operator.

While a multitude of factors influence the attractiveness of the service (perhaps multimedia offers in the vehicle, the quality of Wifi, ...) the authors assume two key properties: The time that passes between a customer making a request and a vehicle arriving (i.e. the wait time) and the price that is charged to the customer. All else being equal, an operator that can offer the shortest wait times at the lowest price will attract more customers than his competitors. For now, it remains unknown how those two factors would be valued against each other by potential customers.

We focus on two main ways for operators to influence the service level of their system:

- The **fleet size** can be increased. In general, this should lead to a decrease of wait time, because the availability of vehicles improves. However, having a larger number of vehicles imposes more fixed costs that would need to be balanced by higher demand. In general, adding more vehicles to the fleet can be regarded as a long-term investment that cannot be altered on a daily basis.
- The **fleet control** can be optimized. Since in an AMoD system it is asssumed that any vehicle can be tracked and controlled online, intelligent fleet control algorithms can be used to minimize the wait times, but also minimize the driven distance in order to reduce operational cost. Applying the proper algorithm is a much less costly intervention than increasing the fleet size with assumably smaller effects, but may bring a competitive advantage on the market.

In the presented experiments both components are investigated by comparing a number of control algorithms for fleets of varying sizes.

Problem Statement

For the algorithmic improvement of the fleet management the authors distinguish between two stages:

- The **dispatching strategy** decides how to serve the demand, i.e. how to match the open customer requests, with the available vehicles. At any time the dispatcher can send tasks to pickup a specific customer to any vehicle that is not currently having a customer onboard (since we do not consider ride-sharing with multiple customers). Also a reassignment of a previously assigned vehicle to another request is possible.
- The **rebalancing strategy** decides where to send vehicles when they are not in use and the demand allows for supplementary movements of the vehicles. The task of the rebalancer is to anticipate future requests and position vehicles such that they are able to optimally react to the upcoming demand.

Hence, vehicles will produce three kinds of mileage:

- **Empty pickup mileage** is produced when an AV is dispatched to a request and is driving to the pick-up location. It is the mileage that needs to be covered in order to serve the customer in any way and may be minimized by an intelligent dispatching algorithm.
- **Empty rebalancing mileage** is produced when an AV is sent to a different location where demand is expected. An ideal operator would exchange all the pickup mileage in the system against rebalancing mileage, i.e. the operator would always send empty vehicles before an actual request turns up.
- **Customer mileage** is produced with a customer on-board. This mileage does only depend on the routing of the cars. In any combination of fleet size and control algorithm, this mileage stays constant, because it is defined by the origin-destinaton relations of all customer trips.

Assuming a common pricing scheme that defines a price per distance, the customer mileage is the only component that produces a benefit for the operator. All other mileage can directly be translated into costs and should therefore be minimized. For general demand patterns, however, it cannot be driven to zero. Treleaven et al. *(8)* show that it is bounded below by the earth mover's distance, which is a measure of how different the distributions of trip origins and destinations are (see *(16)*).

The objectives for a fleet management algorithm can therefore be defined as:

1. Minimize the total pickup distance given the non-optimal locations of the vehicles (dispatcher)
2. Exchange as much pickup distance as possible for rebalancing distance (rebalancer)

Selected Algorithms

In this work we analyze four different operating strategies from literature, which are briefly outlined below:

1. The single heuristic dispatcher is a strategy presented in *(15)*. In every dispatching time step δt_D If there are more available vehicles than requests, it iterates on the list of requests and assigns to each request the closest vehicle. If there are more open requests than available vehicles, the controller iterates on the available vehicles and assigns the closest open request to each vehicle. The assignments are binding, i.e. they are not reopened once concluded.
2. The global Euclidean bipartite matching dispatcher determines an optimal bipartite matching between all open requests and available vehicles in every dispatching time step δt_D. The used distance function is the Euclidean distance which allows to use fast algorithms, e.g. *(17)*. In contrast to the previous strategy, the assignments can be changed until a vehicle actually reaches its target. For a given set of open requests and available vehicles, this strategy can be considered as the optimal dispatching strategy based on Euclidean distances.
3. In *(7)* a feedforward strategy is presented on how to rebalance vehicles between different vertices in a directed graph $G = (V, E)$. For each vertex i and time step δ_t, the arrival rates λ_i and transition probabilities p_{ij} for any nodes $v_i, v_j \in V$ are computed from historical data. The linear program in equation 1 computes the optimal rebalancing flows α_{ij} for an equilibrium point of the underlying fludidic model with travel times $T_{i,j} \forall v_i, v_j \in V$.

$$\begin{aligned}
&\text{minimize} && \sum_{i,j} T_{i,j}\alpha_{ij} && \\
&\text{subject to} && \sum_{i\neq j} \alpha_{ij} - \alpha_{ji} = -\lambda_i + \sum_{i\neq j} \lambda_j p_{ji} && \forall v_{i\in V} \\
& && \alpha_{ij} \geq 0 && \forall v_{i,v_j\in V}
\end{aligned} \tag{1}$$

   To implement this strategy, we divided the city of Zurich into a set of areas. The nodes from *(7)* represent the centroids of these areas on which a complete directed graph called virtual network is placed, see figure 1. Available cars are continuously rebalanced between the vertices of the virtual network according to the static rebalancing rates α_{ij}. As the work does not detail the proposed dispatching algorithm for this strategy, we match cares using global Euclidean bipartite matching. Rebalancing vehicles cannot be dispatched until they reach their destination virtual node.

4. The last implemented strategy is as well derived from *(7)*. Instead of a pure feedforward solution, here in every rebalancing timestep δt_R for every area of the virtual network the avaialble cars and open requests are counted and fed into an integer linear program derived from equation 1 calculating the number of cars reb_{ij} to be sent from virtual vertex i to virtual vertex j. As in the feedforward strategy, the matching of the cars is done via global Euclidean bipartite matching.

SIMULATION SETUP

In order to assess the performance of the different fleet sizes and control algorithms a novel scenario for the city of Zurich, Switzerland is set up for the MATSim transport simulation framework. The section is structured as follows: First, we give an overview about the used simulation components, second, we specify the scenario and finally, we provide fleet sizing results from the theoretical methodology presented in *(10)*.

MATSim and AMoD Simulation

MATSim (cite Horni) is an agent-based simulation framework that makes it possible to simulate large numbers of agents representing a real population in a traffic environment. Similar to reality, each agent has a daily plan with activities that he wants to perform for a certain duration and finish at a specific time of the day. Since these activities take place at different locations in the scenario, agents need to move from activity to activity. By default, MATSim allows the simulation of car traffic, public transit and slow modes such as going by bike or walking. Network-based modes, such as private cars are simulated in a time-step based manner in a network of queues with all participants at the same time. This way it is possible that congestion emerges and agents arrive late at their activity locations. While MATSim provides more functionality, e.g. the replanning of agents plans to adapt to the traffic conditions that they perceive, only the network simulation is used in this research.

An extension developed by the authors of this paper is used to add automated taxis to the set of available travel modes. A virtual dispatcher, for which different algorithms are used in this study, is constantly giving them instructions where to go and what to do. The “lifecycle” of a request is always the same: First, whenever an agent wants to depart from his current activity location by AV, a request is issued to the dispatcher and saved. Then, an AV needs to be sent to the customer. The choice which vehicle to send and when is completely defined by the dispatching algorithm. Once the vehicle arrives at the customer’s location he is picked up, the AV drives to the destiation and finally drops him off. Then, the vehicle is available for dispatching again. Alternatively, vehicles can be rebalanced, which simply means that the dispatcher gives an AV the instruction to drive to a different location. All of this is performed in the MATSim traffic simulation such that AVs suffer from congestion as any other vehicle.

It should be noted that AVs drive directly to the locations where agents finish and start their activities. So far no mechanism is implemented that would allow them to meet at optimized locations (e.g. a high-capacity avenue instead of a small alley).

Scenario Definition

For Switzerland the Microcensus on mobility and transport *(18)* is available, which features the daily travel patterns of 60,000 Swiss residents. It is the basis for a readily available agent population of Switzerland, which reproduces the demographic attributes and travel patterns in the country to great detail *(19)*.

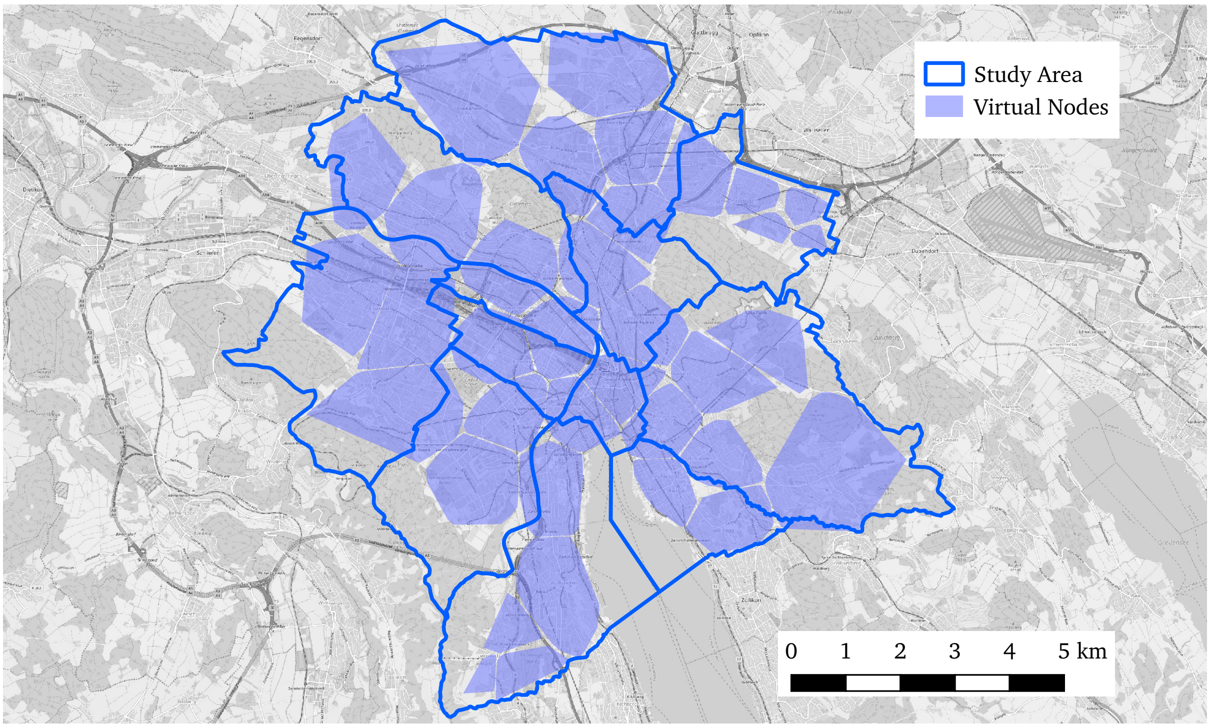

FIGURE 1 The study area covering the 12 districts of Zurich and the nodes of the virtual network for the rebalancing algorithms.

Additional modifications are applied to this population of around 8 million agents to make it suitable for the study at hand. First, a best-response routing of the travels of all agents is performed to find all agents that interfere with the study area, which has been defined to the 12 districts of Zurich (Figure 1). All agents which do not interact with that region (performing an activity within the area or crossing the area) are deleted from the population as they do not contribute to the state of the traffic system in that area. Finally, a 1% sample of the remaining agents is created, which is the basis for our simulations. The rather extensive downscaling becomes necessary for the computationally demanding algorithms, given that they need to be performed hundreds of times fater than reality to allow for multiple runs and iterations.

In order to define the travel demand for the fleet of automated vehicles, agents are tagged as whether they are viable for using an automated vehicle or not. While pedestrians and cyclists are not simulated at all here (since they do not contribute to congestion in the current version of the framework), agents that travel by car or public transit at least once during their daily plan are handled differently.

An agent that travels at least once by private car during the simulation is tagged as an AV user *only* if all of the legs in the agent's plan take place within the study area. This constraint makes sure that no unrealistic travel plans are generated, where an agent performs his first leg by AV although his private car is at home and then wants to depart at the next location with that car. Finally, the "car" legs of all viable agents are converted to the "av" mode. All other legs are kept as before, i.e. short legs that are assigned the "walk" mode initially are still performed in this mode.

For agents that use public transit, the procedure is different. Here, any leg that is performed by the "pt" mode in the original population is converted to "av" if it lies within the study area. As for car users, connecting non-motorized legs are kept fixed.

This way a demand for Zurich is generated where each leg that possibly *can* be performed using an AV *is* performed by AV. In that sense we simulate a scenario where 100% of the AV travel demand must be served by the dispatchers.

To summarize, the 8,230,971 agents in the population are decimated to 1,935,400 agents, which interfere with the study area. From this set of agents a 1% sample has is drawn, leading to 13,141 agents that mainly constitute background traffic for congestion. Among those are 970 agents that are viable for the AV service. The plans of these agents contain 2,096 trips that are to be served by AVs. In reality, this service would hence need to serve 209,000 requests by 97,000 persons.

Theoretical Fleet Sizing

Both the capital cost of an AMoD system and the service rates are highly dependent on the fleet size which makes fleet siziing an important aspect of AMoD system design. If the fleet size is chosen too small, then the service levels will be inacceptable, if the fleet size is chosen too large, the cost of the system becomes unbearable due to low utilization rates.

Fleet sizes can be estimated using simulations, as for instance done in *(15)*. Despite of the accuracy of these simulation results, they do not provide insights into the fundamental properties influencing the relationship between fleet size and performance metrics.

For this reason we have implemented theoretical results from *(10)* for the case of Zurich. The authors present two methods for fleet size evaluation. The first method estimates the theoretical minimum fleet size to stabilize the system, i.e. ensure that the number of open requests stays bounded at all times. To do so, for every vertex i and timestep δ_t the added unserved mileage per timestep is calculated as $\lambda_i \cdot (\bar{d}_{OD,i} + \bar{d}_{EMD,i})$ where $\bar{d}_{OD,i}$ is the average distance per trip and $\bar{d}_{EMD,i}$ the earth mover's distance per vehicle in the timeslice. $\bar{d}_{OD,i} + \bar{d}_{EMD,i}$ represents the average distance that has to be driven per request. A total of m vehicles at an average speed of v are collectively able to reduce this added mileage at a rate of $m \cdot v$, this quantity has to be larger than the added unserved mileage per timestep. For the scenario of this work the minium fleet size compute with this measure are 4271 vehicles, which can be seen as a non-tight lower bound.

While the knoweldge of the minimum fleet size is useful, it does not reveal the relation between service level and fleet size, especially to what number the fleet size has to be augmented before further addition of vehicles will not result in a significant increase in service level. In *(3)* a method is presented of how an AMoD system can be cast in a Jackson network. For such networks, queuing theoretical results allow the computation of performance measures such as vehicle wait times, queue lengths or availabilities at vertices. The quantity of interets is the availability of a vehicle at a vertex, which is the probability that 1 or more idle vehicles are at that vertex. Computation of the mean availability of all timesteps and vertices as a function of the fleet size for Zurich results in the curve shown in 2. Note that these results are purely theoretical and can be derived solely from input data without performing simulations. Therefore they can serve as a measure of accuracy for the simulation results.

For the peak case, the increase of the number of vehicles up to a fleet size of $15,000$ results in a availability rising from 0 to 0.85. The addition of another $15'000$ vehicles does only increase the availability by 0.05 to 0.90. This corresponds well to the wait times observed in simulation shown in figure 3. The simulated wait times decrease significantly up to a fleet size of $15,000$

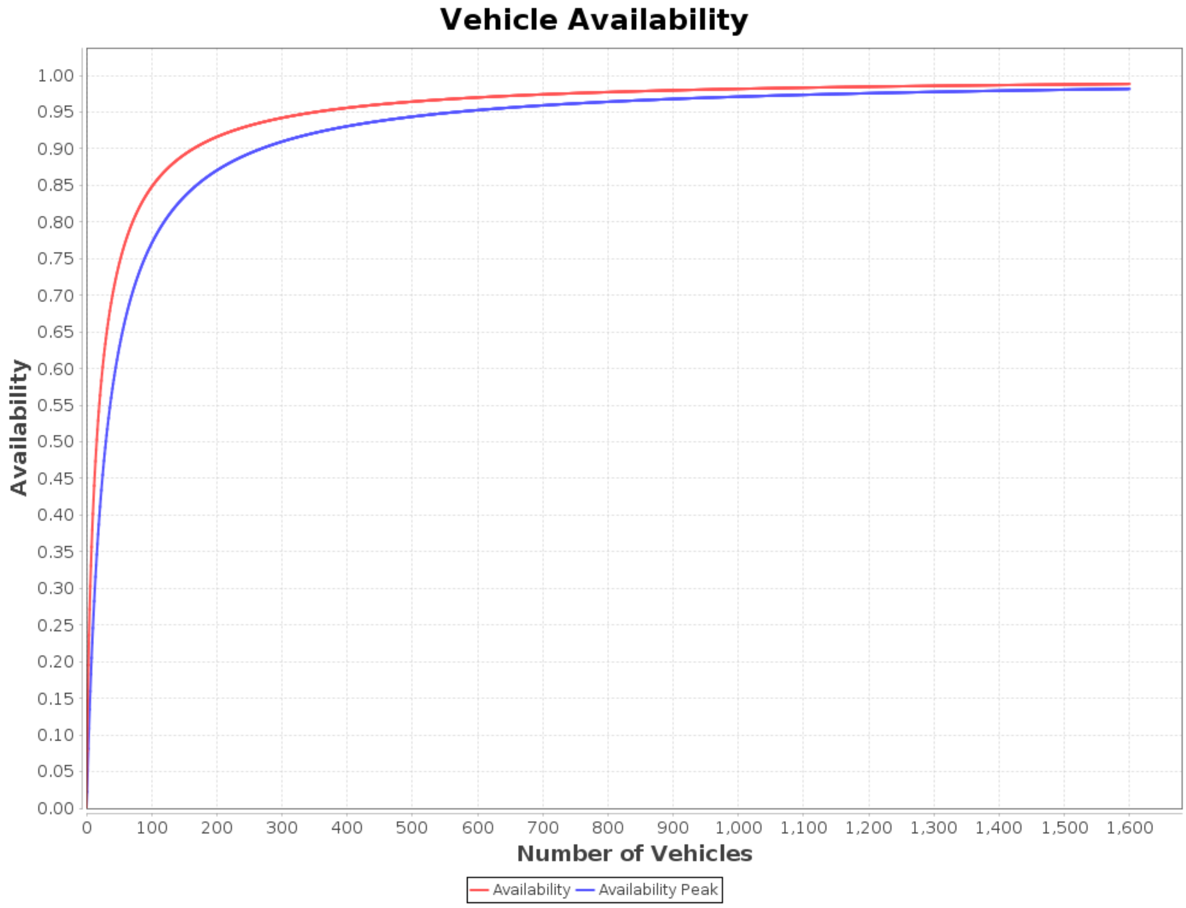

FIGURE 2 Mean availability of vehicles over all timesteps and virtual vertices.

before saturating.

RESULTS

We test the four proposed dispatching strategies in the Zurich scenario with ten runs per fleet size and strategy. Since the dispatchers rely on freeflow speeds in the network for their routing when the simulation starts, we let each run perform 20 iterations in which the dispatcher step by step senses the traffic conditions, e.g. how to avoid traffic jams at peak hours.

The dispatching stages of all algorithms are called once every 60 seconds in simulated time, while the rebalancing periods for the feedforward and feedback dispatcher are 10 minutes and 20 minutes, respectively. Those values have been obtained from prior simulation runs.

For Zurich, the times with peak congestion and, hence, longest travel times are from 6:30pm to 9:00am and from 4:30pm to 6:30pm. Figure 3 shows the average customer wait time over the whole day and just for peak hours. While the simple heuristic approach consistently yields the longest wait times for any fleet size, the feedback dispatcher performs best. The bipartite matching performs in between, since it is based on an optimal request assignment, but does not do any rebalancing. Surprisingly, the feedforward dispatcher performs similar to the bipartite matching algorithm. This means, that the expected demand that has been generated from the scenario data does not optimally predict the actual travel patterns in the simulation.

Assuming that 5 minutes at peak times are an acceptable wait time, that delay is achieved

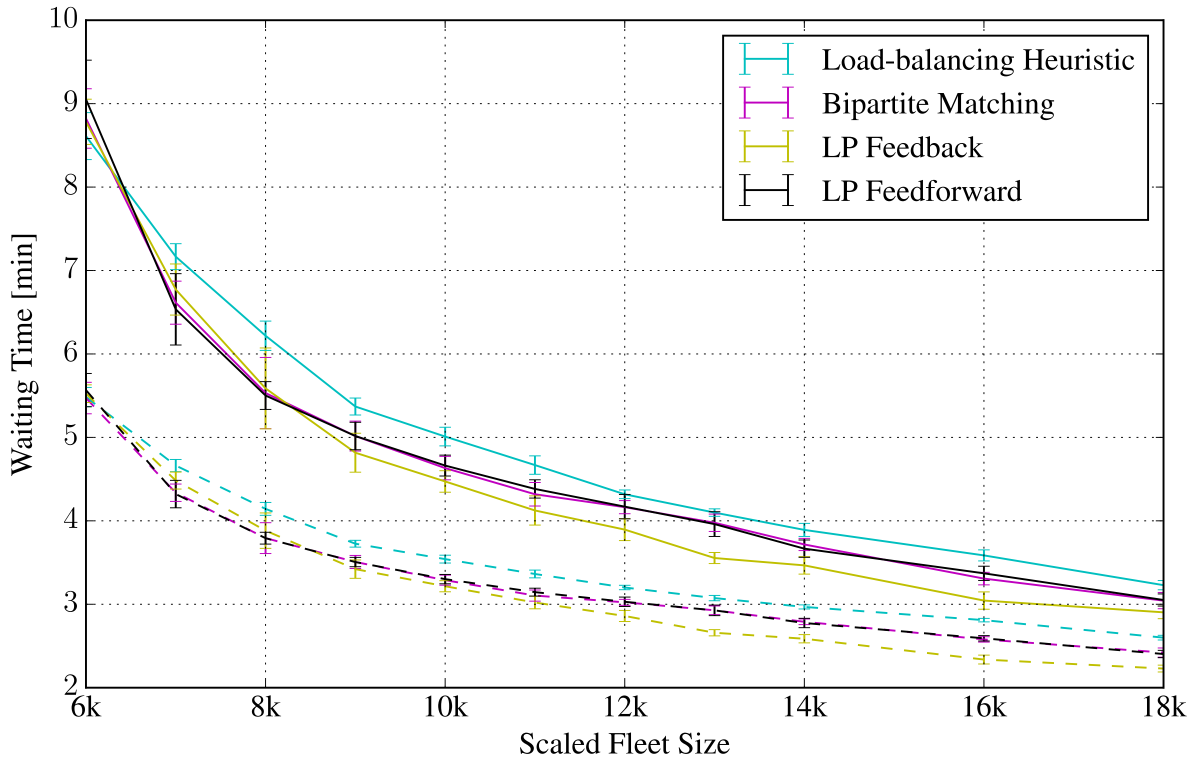

FIGURE 3 Average waiting time for an AV to arrive at peak times (solid) and over the entire day (dashed)

with a fleet of 10,000 vehicles for the heuristic, but with only 8,700 for the feedback dispatcher.

[TODO: Updat empty rides!!!]

Figure 4 shows the percentage of fleet mileage that is driven without a customer, either for pickup or rebalancing purposes. Clearly, the LP algorithms, which both use rebalancing, have a higher share of empty mileage that the non-rebalancing approaches. The heuristic approach manages to keep the share lowest, since it mainly operates in a best-response state, where only the shortest pickup trips are chosen. Remarkably, the total driven distance for all dispatchers is very similar (Figure 5), which indicates that the surplus of empty distance for the intelligent dispatchers does not stem from inefficient movements, but rather effective movements towards the expected customer demand for shorter waiting times.

[TODO: Do we need two plots here? Also a plot Total Distance <-> Relative Distance would be possible, where one can traverse the fleet size along the graph]

Finally, the occupancy of the fleet is measured. For a fleet size of 6,000 vehicles, they are busy carrying a passenger for around 4.8h per day, while this value drops to 2.16h for the maximum fleet size of 140,000. In both cases, those numbers exceed the average 1.92h [TODO Cite, do we have this for switzerland???] of today's vehicle fleet. Comparing the heuristic and feedback dispatchers at a fleet size for five minute wait time, the vehicles in the latter one are occupied almost one hour more per day.

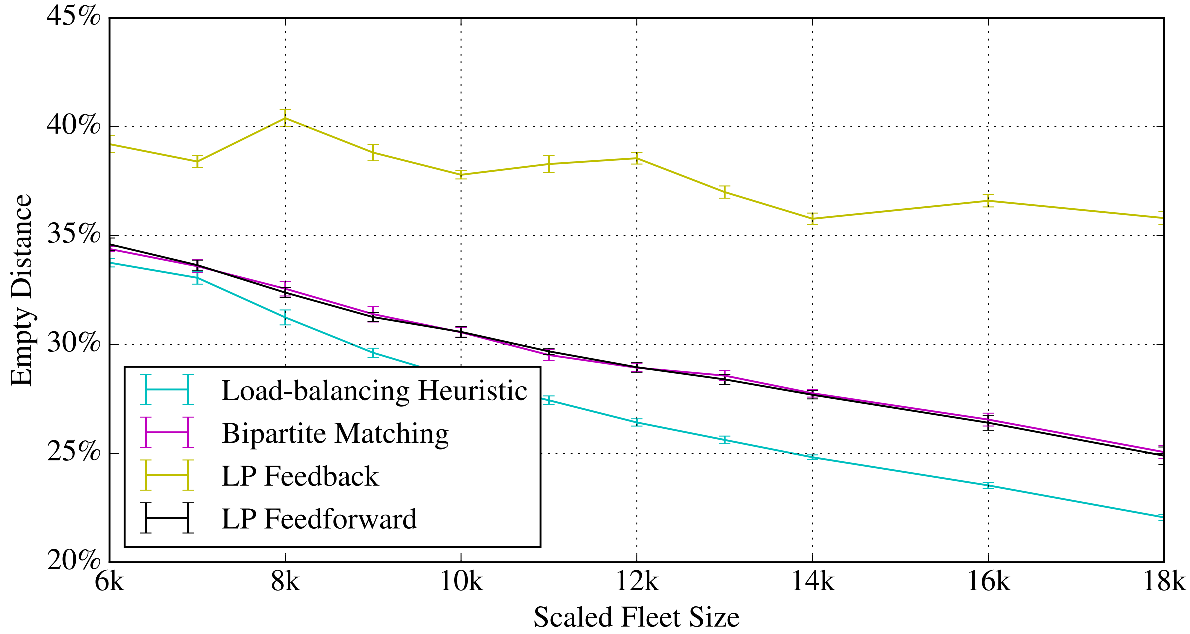

FIGURE 4 The fraction of distance that is driven by AVs without a passenger.

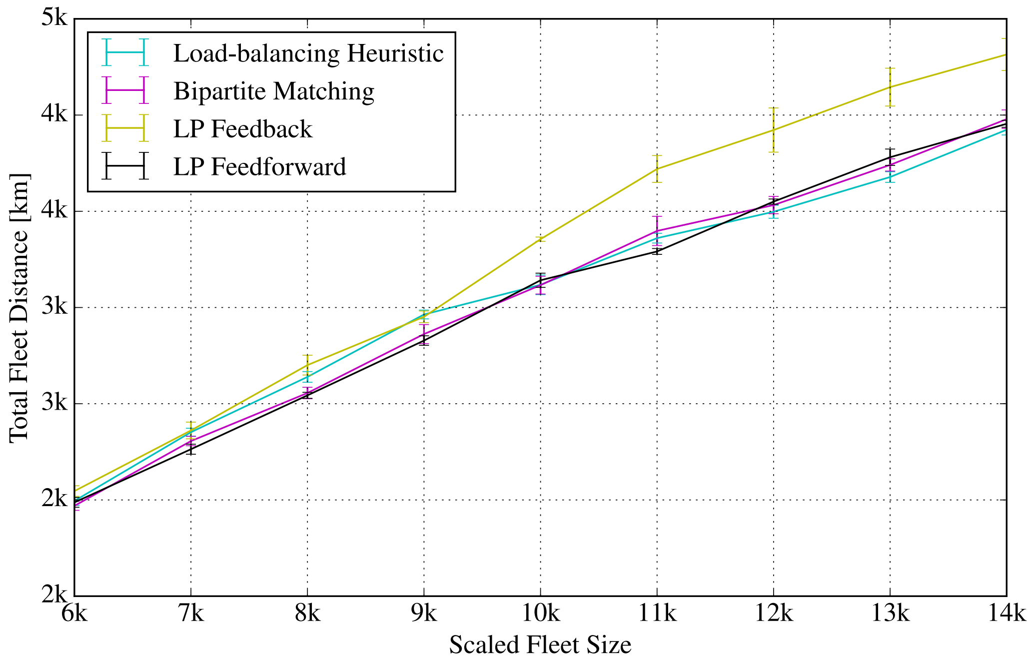

FIGURE 5 The total distance that is driven by AVs, with and without passenger on-board.

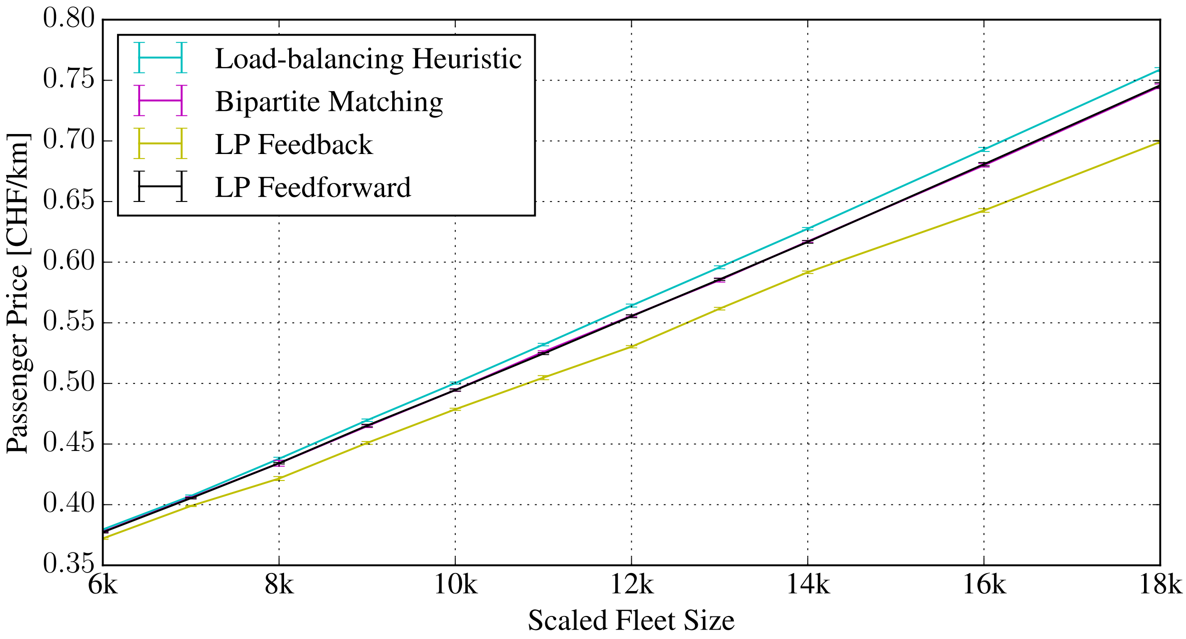

FIGURE 6 The minimum customer prices that an AV operator needs to charge the customer in order to have a profit margin of at least 3%.

Cost Analysis

Based on the cost calculator for fleets of automated vehicles by Bösch et al. *(20)* the costs of operating the simulated AV services are computed. Specifically, by providing the calculator with key figures of the operator (among them the occupancy, the share of empty rides, the average travel distance) the price that the operator would at least need to ask a customer per kilometer is computed. The calculation is based on a detailed analysis of running and fixed costs and includes a profit margin of 3% for the operator. Figure 6 shows the results from this analysis. Unsurprisingly, the price that needs to be charged from the customer increases with larger fleet sizes, while a clear difference between the algorithms can be observed. Clearly, the simple heuristic is the most costly operating scheme, while the feedback dispatcher can be operated with the lowest passenger prices.

Compared to the average price of a taxi operator in Zurich (6 CHF/km, [TODO Cite]) the computed prices are extremely low. Hence, an automated service would clearly push conventional taxi operators out of the market. While in the short term a private vehicle today is cheaper than the analysed AV services (0.17 CHF/km, [CITE]) except for very large fleet sizes the service is cheaper in the long run (compared to 0.72 CHF/km overall costs of owning a private car). However, compared to (subsidized) prices for public transport (0.25 CHF/km), the services are more expensive.

Therefore, the proposed AV services are highly attractive to car users, but may not be able to compete with subsidized public transport. On the other hand, AVs allow for more direct trips and thus for savings in travel time. Further studies may analyse how these affect the attractiveness of the AMoD services.

Looking at different choice alternatives for customers, two components are important that are weighed against each other: The expected wait time and the price. Figure 7 combines the key results from our simulations. There, the price that a specific operator configuration (fleet size

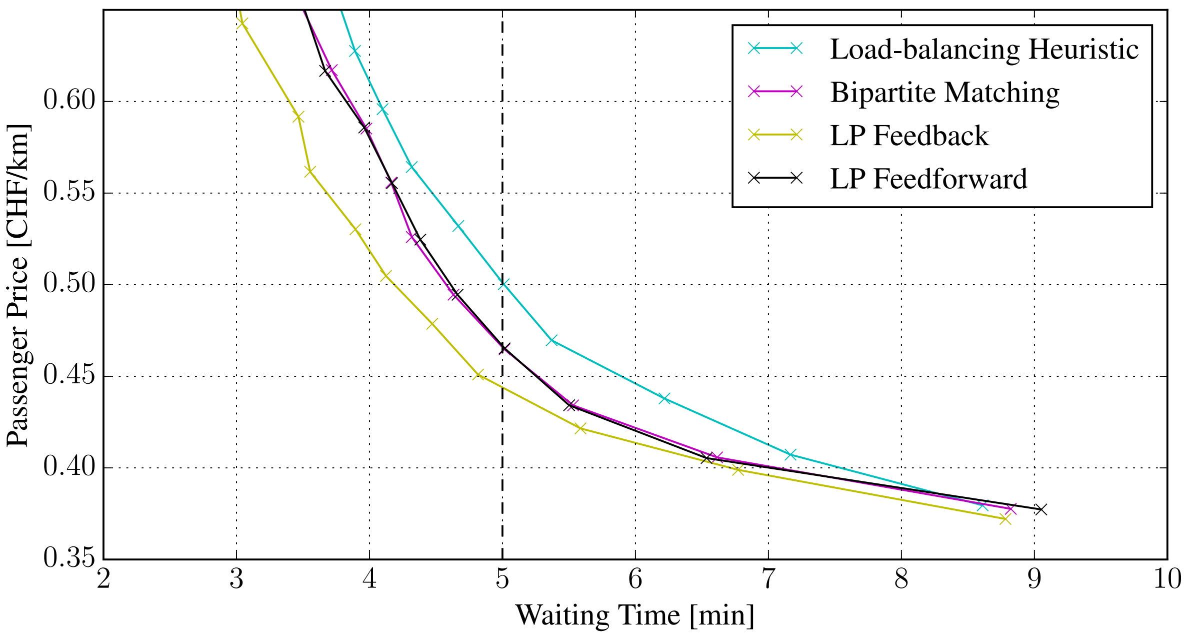

FIGURE 7 Time vs. Price

and dispatcher) needs to charge is displayed in comparison to the wait time that this operator is offering. At a wait time of five minutes an operator would be able to offer a satisfactory service for around 0.45 CHF with the feedback dispatcher, while he would need to charge 0.50 CHF with the simple load-balancing heuristic.

DISCUSSION & CONCLUSION

The study shows that the right choice of dispatching algorithm for an AMoD system does not only have strong impact on the performance in terms of wait time for the customer, but also that it bears a significant economic advantage for the operator. Operators with intelligent redispatching and rebalancing algorithms are able to attract more customers through quicker pickups and lower prices than a competitor at small additional cost.

In order to assess the significance for real fleets of (not neccessarily automated) taxis it needs to be noted that all of the presented algorithms are able to process dispatching and rebalancing tasks for fleets of thousands of vehicles within minutes. It is perfectly feasible to control 100k vehicles in five minute updates using a standard laptop for the computational tasks.

For the presented simulations, this still poses a burden, because there a speedup compared to reality of around one thousand times is desired to be able to run large numbers of simulations. Hence, the algorithms are only tested on a subsample of 1% of the agent population that is available. In future studies, effort will be put into overcoming this restriction, either by finding approximate formulations for the presented algorithms or pursuing research on completely new algorithms.

Throughout the paper, a “100%” demand scenario is used, in which all trips that possibly could be undertaken by AV are converted to the automated mode. MATSim offers the possibility to explicitly simulate attitudes toward new elements in the traffic system by defining utilities for using specific modes with distinct valuation of travel costs, travel times and distances. This way, by integrating the presented algorithms into the full MATSim loop, as shown in *(21)*, the actual attractiveness of an AV service can be analyszed including the tradeoff that people make between paying for the service, spending time in the vehicle and having to wait for it. Naturally, not 100% of possible trips would be performed by AV then, but only a fraction. Future work will take these considerations into account.

REFERENCES

1. Ackerman, E. (2017) Hail, robo-taxi![top tech 2017], *IEEE Spectrum*, **54** (1) 26–29.
2. Katzev, R. (2003) Car sharing: A new approach to urban transportation problems, *Analyses of Social Issues and Public Policy*, **3** (1) 65–86.
3. Zhang, R. and M. Pavone (2016) Control of robotic mobility-on-demand systems: a queueing-theoretical perspective, *The International Journal of Robotics Research*, **35** (1-3) 186–203.
4. Pfrommer, J., J. Warrington, G. Schildbach and M. Morari (2014) Dynamic vehicle redistribution and online price incentives in shared mobility systems, *IEEE Transactions on Intelligent Transportation Systems*, **15** (4) 1567–1578.
5. Ruch, C., J. Warrington and M. Morari (2014) Rule-based price control for bike sharing systems, paper presented at the *Control Conference (ECC), 2014 European*, 708–713.
6. Smith, S. L., M. Pavone, M. Schwager, E. Frazzoli and D. Rus (2013) Rebalancing the rebalancers: Optimally routing vehicles and drivers in mobility-on-demand systems, paper presented at the *American Control Conference (ACC), 2013*, 2362–2367.
7. Pavone, M., S. L. Smith and E. F. D. Rus (2011) Load balancing for mobility-on-demand systems.
8. Treleaven, K., M. Pavone and E. Frazzoli (2011) An asymptotically optimal algorithm for pickup and delivery problems, paper presented at the *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, 584–590.
9. Zhang, R., F. Rossi and M. Pavone (2016) Model predictive control of autonomous mobility-on-demand systems, paper presented at the *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, 1382–1389.
10. Spieser, K., K. Treleaven, R. Zhang, E. Frazzoli, D. Morton and M. Pavone (2014) Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in singapore, in *Road Vehicle Automation*, 229–245, Springer.
11. Fagnant, D. J. and K. M. Kockelman (2015) Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in austin, texas, *Transportation, Under review*.
12. Zachariah, J., J. Gao, A. Kornhauser and T. Mufti (2014) Uncongested mobility for all: A proposal for an area wide autonomous taxi system in new jersey, paper presented at the *Transportation Research Board 93rd Annual Meeting*, no. 14-2373.
13. Martinez, L. M. and J. M. Viegas (2017) Assessing the impacts of deploying a shared self-driving urban mobility system: an agent-based model applied to the city of lisbon, portugal, *International Journal of Transportation Science and Technology*.

14. Boesch, P. M., F. Ciari and K. W. Axhausen (2016) Autonomous vehicle fleet sizes required to serve different levels of demand, *Transportation Research Record: Journal of the Transportation Research Board*, (2542) 111–119.

15. Bischoff, J. and M. Maciejewski (2016) Simulation of city-wide replacement of private cars with autonomous taxis in berlin, *Procedia computer science*, **83**, 237–244.

16. Rüschendorf, L. (1985) The wasserstein distance and approximation theorems, *Probability Theory and Related Fields*, **70** (1) 117–129.

17. Agarwal, P. and K. Varadarajan (2004) A near-linear constant-factor approximation for euclidean bipartite matching?, paper presented at the *Proceedings of the twentieth annual symposium on Computational geometry*, 247–252.

18. Federal Statistical Office (2012) Microcensus on mobility and transport 2010.

19. Bösch, P. M., K. Müller and F. Ciari (2016) The ivt 2015 baseline scenario, *16th Swiss Transport Research Conference*.

20. Bösch, P. M., F. Becker and H. Becker (2017) Cost-based analysis of autonomous mobility services, *Arbeitsberichte Verkehrs- und Raumplanung*, **1225**.

21. Hörl, S. (2017) Agent-based simulation of autonomous taxi services with dynamic demand responses, *Procedia Computer Science*, **109**, 899–904.