Assessment of fleet control algorithms for autonomous mobility on demand systems

Date of submission: 2017-08-01

Sebastian Hörl
IVT, ETH Zürich, 8093 Zürich, Switzerland
phone: +41 44 633 38 01
sebastian.hoerl@ivt.baug.ethz.ch

Claudio Ruch
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: 4726 words + 7 figures = 6476 word equivalents

ABSTRACT

The performance of four different dispatching and rebalancing algorithms for the control of an Autonomous Mobility On-Demand system is tested. The case study, which is based 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, which is more expensive than using a private car today, but significantly cheaper than a conventional taxi. The results show that this service can be offered while maintaining a higher fleet occupancy than can be observed for private cars today and that the application of intelligent rebalancing algorithms is inceasing the share of miles driven without a customer, 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 [cite nutonomy, cite uber]. They promise to increase road capacity and speeds [cite tientrakool, friedrich] and would give access to mobility to formerly inhibited user groups (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 at hand 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 [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, however almost certainly makes it necessary to relocate them without a passenger on-board, which directly translates to costs for the operator.

In the study at hand we contriubte to research around 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 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

Let's go through this again. We need:

- Literature on rebalancing in general (Claudio bike rebalancing, but there is a lot for car-sharing available), just indicate importance not go into detal on algorithms
- Simulation studies on AVs (we have that already! commented out right now), start with the conceptual ones (Spieser) and mention we take algos from there
- Literature MATSim

It goes as follows: First literature on rebalancing is introduced, then the simulation studies on AVs are presented with the remark that they do NOT consider rebalancing. We end with Kockelman and Bösch, who use MATSim but just as a preparational step and lead to full

MATSIm simulations, first Michal Joschka, then ABMTRANS, which is used here.

This way we have introduced everything we're using here.

CONTROL OF AN AMOD SYSTEM

An AMoD service only makes sense if it is attractive to any customers. More specifically, it can only be maintained if a sufficient number of customers wants to use the service such that the financial benefits for the operator exceed the costs.

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 from 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.

Based on the customer expectation, a number of options are available for the operator to become profitable. Here, we focus on two strategies:

- 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 save money. 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. the open customer requests, with the available supply (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 is possible at any time if a more viable request comes in.
- 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, because then a vehicle would always already be present when a request pops up.
- **Customer mileage** is produced with a customer on-board. 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. *(1)* 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 *(2)*).

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:

[TODO: I would do this more in detail! Why not explicitly state the LPs, at least for the last one, which is "new"? I think if somebody reads this he cannot really figure out whats going on.]

1. The single heuristic dispatcher is a strategy presented in *(3)*. 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. *(4)*. In contrast to the previous strategy, the assignments can be changed until a vehicle actually reaches its target. If arrival probabilties for future time steps is taken into account, this strategy can be considered as the optimal dispatching strategy based on Euclidean distances.
3. In *(5)* 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 used in a linear program to compute the optimal rebalancing flows α_{ij} in that time step assuming that the system is at equilibrium. To implement this strategy, we divided the city of Zurich into a set of areas. The nodes from *(5)* represent the centroids of these areas on which a complete directed

graph called virtual network is placed, see figure **??**. 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.

4. The last implemented strategy is as well derived from *(5)*. 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 which calculates 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.

Performance driven fleet sizing

In order to serve the demand with a certain strategy the fleet size must be sufficient etc etc ... [TODO Explain theory for performance driven fleet sizing]

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 (cite Treleaven).

MATSim and AMoD Simulation

MATSim (cite Horni) is an agent-based simulation framework that makes it possible to simulate large numbers of agents, which model a real population, in the same 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 by Hörl (cite ABMTRANS) 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 agents 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 *(6)* 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 *(7)*.

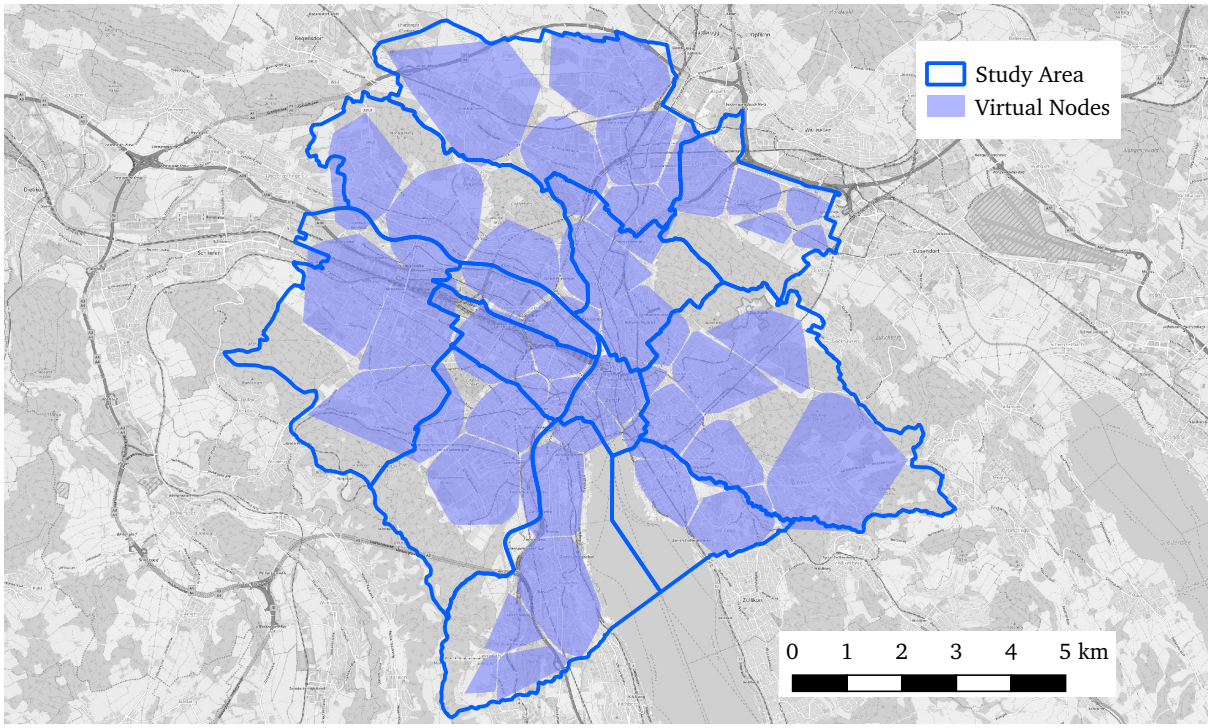

FIGURE 1 The study area covering the 12 districts of Zurich and the nodes of the virtual network for the rebalancing algorithms.

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 19,354 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 4030 trips that are to be served by AVs. In reality, this service would hence need to serve 403,000 requests by 97,000 persons.

Theoretical Fleet Sizing

[TODO Performance driven fleet sizing for Zurich! Here EMD, minimum fleet size, fleet sizing]

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 60 seconds and 20 minutes, respectively.

For Zurich, the times with peak congestion and, hence, longest travel times are from 6:30am to 9:00am and from 4:30pm to 6:30pm. In figure 2 all trips by AV with departure times in these time windows are collected and the mean waiting time is computed. As expected, the average waiting time is decreasing with larger fleet sizes and higher availabiltiy of vehicles. Almost over the whole range of fleet sizes the feedback algorithm performs best, while the load-balancing heuristic yields the longest waiting times.

Figure 3 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 4), which indicates that the surplus of empty distance for the intelligent dispatchers does not stem from inefficient movements, but rather effective movements towards

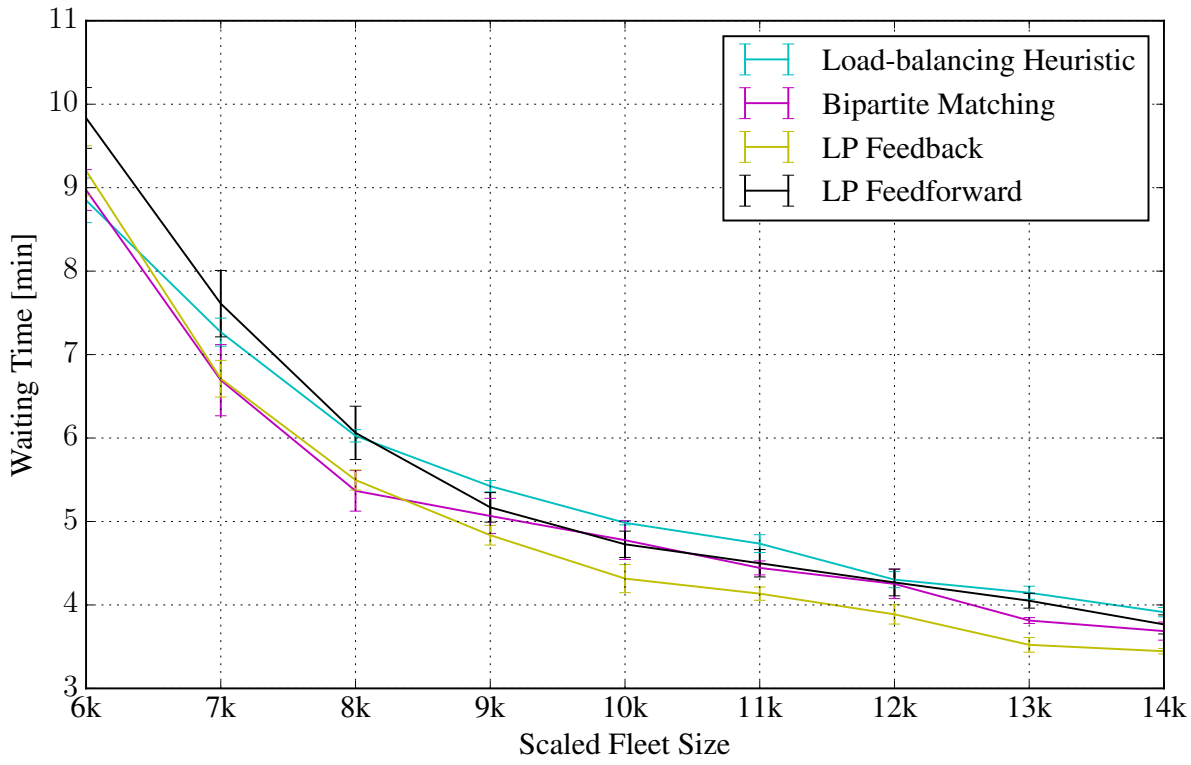

FIGURE 2 Average waiting time for an AV to arrive at peak times

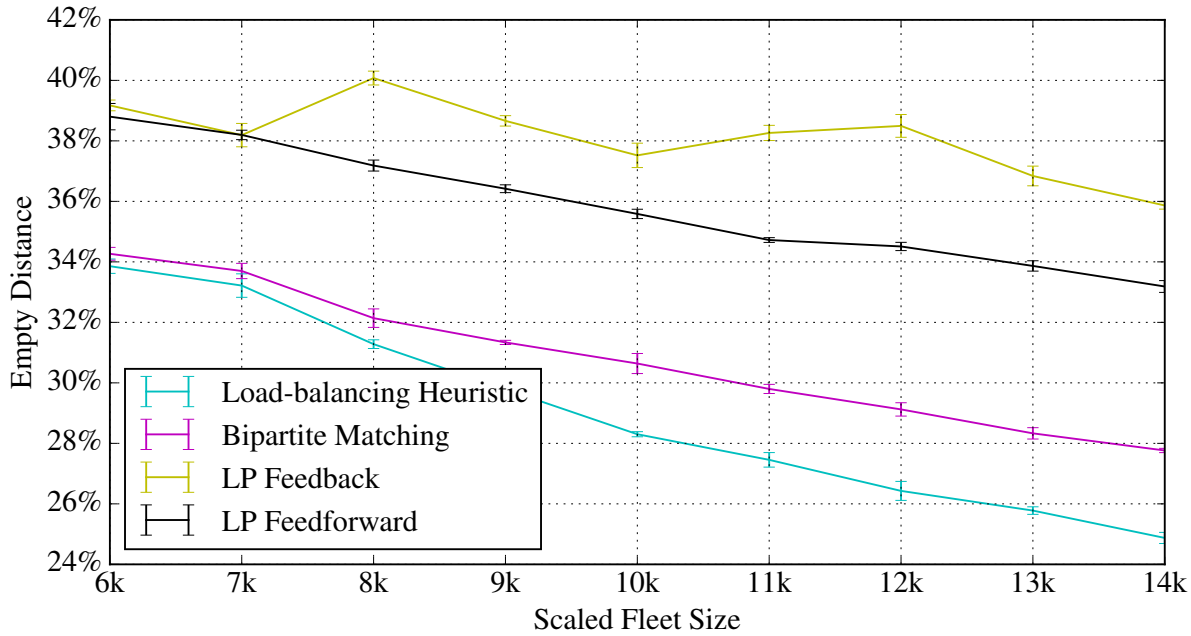

FIGURE 3 The fraction of distance that is driven by AVs without a passenger.

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, figure 5 shows the occupancy of the fleet for different fleet sizes. Since in the 30h

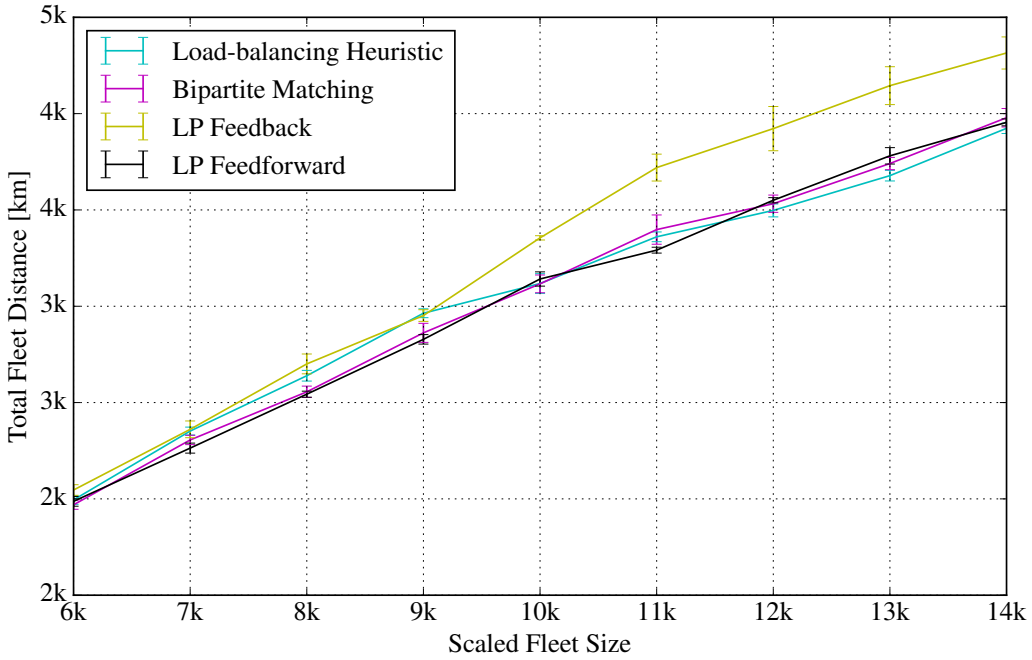

FIGURE 4 The total distance that is driven by AVs, with and without passenger onboard.

MATSim simulation no AV trips are registered in the hours around midnight, it is possible to correct the resulting 30h occupancy rate to one that is based on a 24h day. As can be seen, the occupancy of all fleet dispatchers exceeds the 8% that is common today. In general, one can say that the dispatching algorithm has only little influence on fleet occupancy. The differences lie in the range of 0.5% between the best and worst performing algorithm, which are the LP Feedback dispatcher and the load-balancing heuristic, respectively. Nevertheless, one can see that the occupancy of the latter is systematically lowest.

Cost Analysis

Based on a paper Bösch et al. *(8)* the costs of operating the simulated AV services are computed. Specifically, by providing their 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 if a profit margin of at least 3% is targeted. The calculation is based on a detailed analysis of running and fixed costs. Figure 6 shows the results from this analysis. Unsurprisingly, the price that needs to be imposed on the customer increases with larger fleet sizes. However, the increase is stronger for the load-balancing heuristic than for any other distpaching strategy. Therefore, with the same fleet being available to an operator, he would be able to offer the service for almost 0.10 CHF less per kilometer than before or save this amount of money.

Compared to the average running costs of driving a private car in Switzerland (around 0.17 CHF/km) or using public transit (around 0.25 CHF/km) [TODO CITATIONS] the computed prices still seem rather high. Compared to conventional taxi operators, however, the price is extremely low (around 6 CHF/km). Therefore, it is imaginable that the AV service would still be attractive for a large group of people, for which a conventional taxi would be too expensive on a

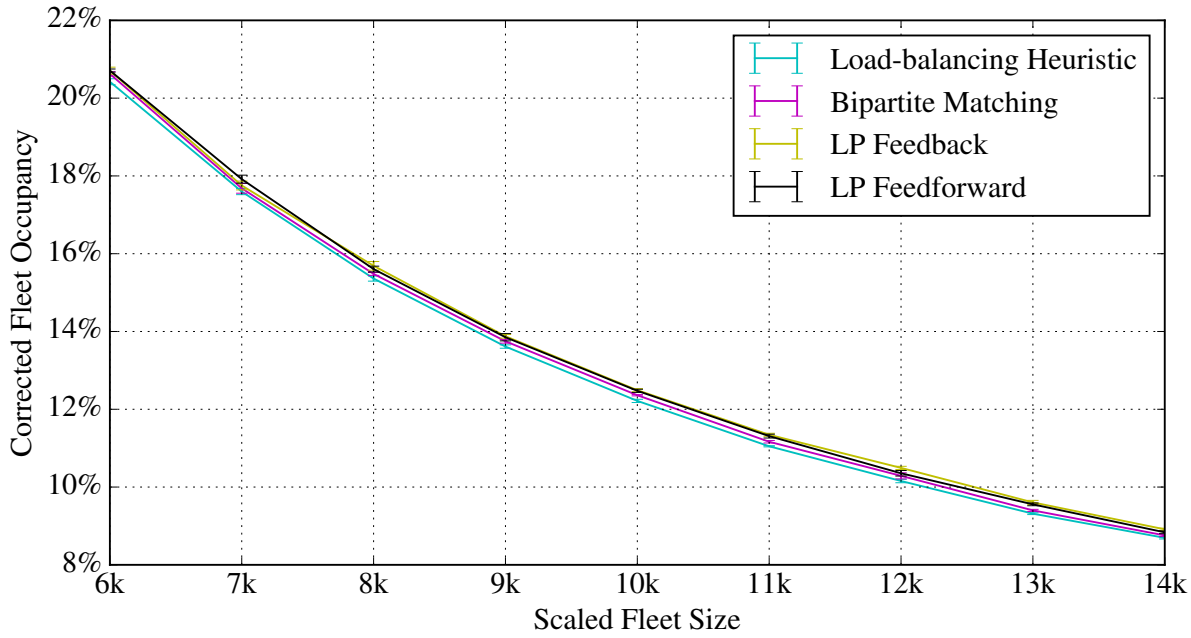

FIGURE 5 The occupancy of the AV fleet for different flet sizes.

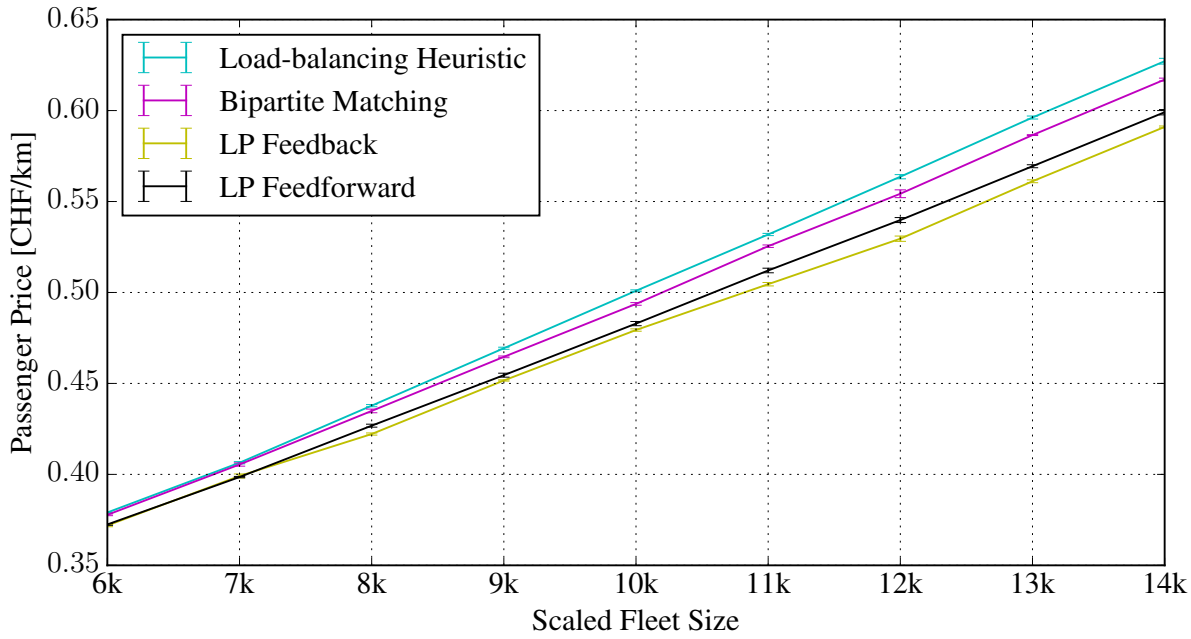

FIGURE 6 The minimum customer prices that an AV operator needs to charge the customer in order to have a win margin of at least 3%.

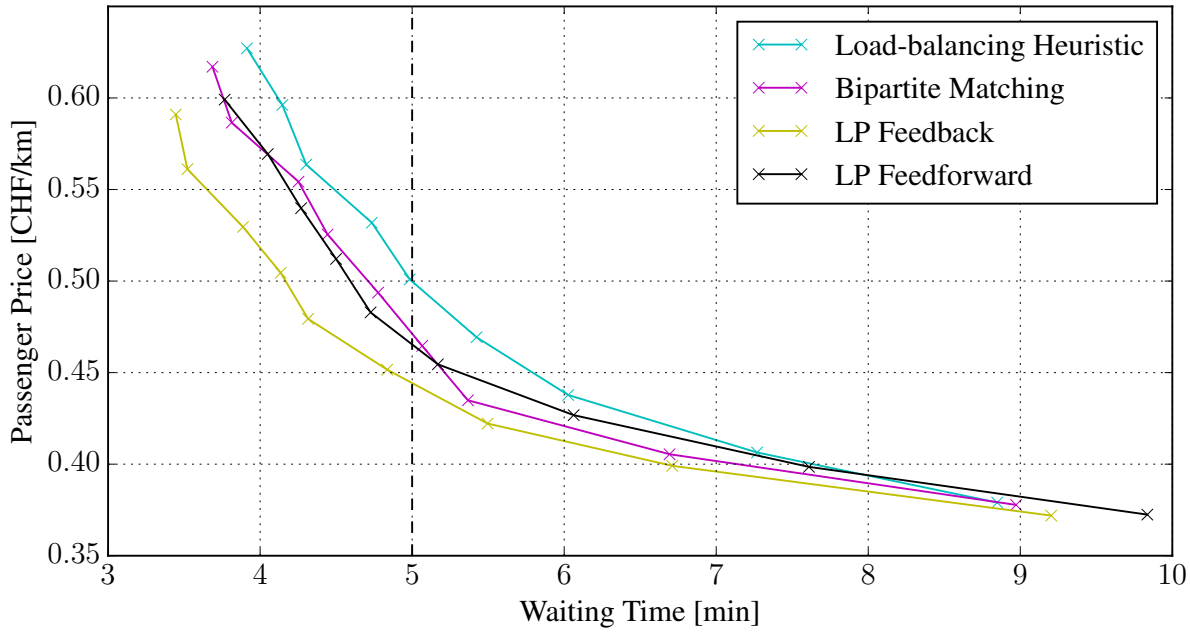

FIGURE 7 Time vs. Price

daily basis, but an AV would make such travels affordable.

However, the attractiveness of an AV service does not only depend the price itself, but also on the attitudes of the people towards the service. One key component to the acceptance of an AMoD system is the waiting times that customers need to endure. Figure 7 combines the key results from our simulations. There, the price that a specific operator configuration (fleet size and dispatcher) is displayed in comparison to the waiting time that this operator can offer. Assuming that, for instance, a waiting time of five minutes is tolerable, the operator could 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. The better the level of service of the operator is ought to be, the larger this margin becomes.

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 waiting times for the customer, but also that it bears a significant economic advantage for the operator. He is able to attract more customers through quicker pickups and lower prices than a competitor with only little investment.

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

Throughout the paper, a "100%" demand scenario has been used, in which all trips that possibly could be undertaken by AV were converted to the automated mode. The MATSim framework, however, 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 *(9)* the actual attractiveness of an AV service could 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 actually be performed by AV, but only a fraction. In such a scenario, also if maybe more remote areas would be included, completely different properties of an AV fleet control algorithm would be of interest, e.g. how well it is able to attract new customer groups in new regions by offering unproportionally low waiting times and make them stick to the service.

[TODO: OTHER LIMITATIONS]

REFERENCES

1. 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.

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

3. 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.

4. 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.

5. Pavone, M., S. L. Smith and E. F. D. Rus (2011) Load balancing for mobility-on-demand systems.

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

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

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

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