

Shared Autonomous Mobility-on-Demand: Learning-based approach and its performance in the presence of traffic congestion

Maxime Guériau*, Federico Cugurullo†, Ransford A. Acheampong‡, Ivana Dusparic*

*Enable – CONNECT Research Center, School of Computer Science and Statistics, Trinity College Dublin, Ireland

†Department of Geography, Trinity College Dublin, The University of Dublin, Ireland

‡Department of Planning and Environmental Management, University of Manchester, United Kingdom

Abstract—Mobility-on-demand systems consisting of shared autonomous vehicles (SAVs) are expected to improve the efficiency of urban mobility through reduced vehicle ownership and parking demand. However, several issues in their implementation remain open, such as unifying the vehicle and ride-sharing assignment with rebalancing non-occupied vehicles. Furthermore, proposed SAV systems are evaluated in isolation from other traffic; no congestion is taken into account when assigning requests or calculating routes. To address this gap, we present Shared Autonomous Mobility-on-Demand system (SAMoD), a reinforcement learning-based approach to vehicle relocation and ride-sharing request assignment. Each vehicle learns its pick-up and rebalancing behaviour based on local current and observed historical demand. We evaluate SAMoD on Manhattan network using NYC taxi data in microsimulator SUMO. We investigate SAMoD performance in the presence of congestion generated by private vehicles, as well as investigate impact of different percentages of SAMoD vehicles in the system on overall traffic network performance.

Index Terms—ride-sharing, mobility on demand, traffic congestion, shared autonomous vehicles

I. INTRODUCTION

The advent of autonomous cars and car-sharing systems revealed the potential of Shared Autonomous Vehicles (SAV) to act as a promising Mobility-on-Demand (MoD) system in cities. Reduced waiting and travel times in such systems are expected [1] under the assumption that citizens are willing to share rides, and that autonomous cars can relocate themselves to the areas with high demand. Efficient relocation strategies are of crucial importance, as travel patterns in cities significantly vary during the day but can also vary daily.

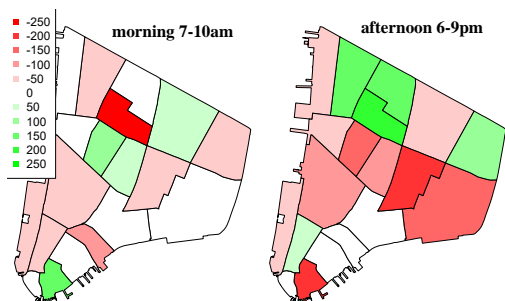


Fig. 1: Pick-ups and drop-offs number imbalance

To illustrate this, Figure 1 shows an example of an observed imbalance in New York City (NYC) taxi trip data [2] during the morning and evening rush hours. Ride-sharing in MoD systems has a strong potential, since most of the vehicles travel with only a few passengers, as illustrated in Figure 2.

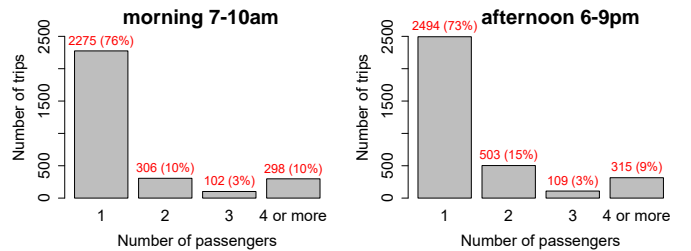


Fig. 2: Sample taxi trip occupancies

Using the same data as in the previous figure, we observed an average occupancy of only 1.6 passenger per car. While the morning and evening rush hours follow a similar pattern, the majority of trips (over 70%) served 1 passenger only, which highlights the potential of ride-sharing enabled systems. Recent ride-sharing research [3], [4] confirms the importance of ride-sharing to significantly increase the level of service using high-occupancy shared vehicles.

However, a number of gaps are observed in existing work. Ride-sharing and rebalancing strategies are addressed using separate approaches, *i.e.*, only unoccupied vehicles can rebalance. In combined approaches, partially-occupied vehicles could also be taken into account as they can serve further requests along their route, effectively rebalancing to the areas they are travelling through. Another factor that needs to be considered is the willingness of citizens to adopt ride-sharing systems. In a survey of 1233 residents of Dublin, Ireland, which we conducted, only 5% were willing to give up their private vehicle in favour of shared MoD systems. However, 54% of respondents were willing to replace some trips with the shared ones. Further, 14% of respondents were willing to adopt autonomous vehicles as soon as they become available, while 49% indicated they would wait 5 years or longer. Therefore, evaluation of autonomous ride-sharing systems needs to take different adoption rates into account. Another major issue is that the performance of ride-sharing systems has so far been evaluated only in isolation from other traffic; either analytical

models are used to calculate travel/waiting times assuming straight-line distances, or simulation is used, but no other types of vehicle is simulated. Therefore, it is uncertain whether performance of the existing models holds true in the presence of other traffic on real road networks, as congestion will directly impacts SAV travel times, and therefore their ability to arrive to the location of the requests in time.

In our previous work [5], we introduced a decentralized Shared Autonomous Mobility-on-Demand system (SAMoD), a reinforcement learning (RL) based decentralized approach to vehicle rebalancing as well as ride request assignment in shared mobility-on-demand systems. In this paper we reiterate the main features of this approach, but go further by extending SAMoD algorithm to make request assignment and rebalancing decisions based on real-world road network and in the presence of congestion simulated in micro-simulator SUMO. We investigate the behaviour of SAMoD in the presence of congestion generated by private vehicles, as well as investigate impact of different percentages of SAMoD vehicles in the system on overall traffic network performance.

Therefore, the main contributions of this paper are:

- Extending SAMoD algorithm to take congestion into account when making request assignment and vehicle routing decisions
- Evaluating SAMoD in the presence of congestion in SUMO micro-simulator using the real-world map of Manhattan. As ride-sharing systems are usually evaluated in isolation, this work represents the first evaluation of a ride-sharing and rebalancing system in a microsimulation with other traffic present.
- Investigating the impact that shared vehicles have on overall congestion, by varying the numbers and ratio of shared vehicles and private vehicles in the simulation.

II. RELATED WORK

Algorithms for on-demand ride-sharing are increasingly investigated in literature with the aim to increase the efficiency of SAV systems. Table I shows the most relevant approaches.

While the initial work, e.g., [6], [7], [8], [9] focuses only on ride-sharing, approaches in [10], [4], [3], [11], [12] combine rebalancing and ride sharing as their mutual impact has been recognized. Rebalancing in these is, however, done either at fixed intervals rather than dynamically [10], or using a centralized approach [4], [13], [12], [14]. A decentralized rebalancing approach is presented in [11], but in combination with a centralized request and ride-sharing assignment. Apart from computational efficiency (and therefore more responsive real-time adaptivity to new requests), a benefit of decentralizing both dispatch and rebalancing is that vehicles do not have to be part of the same fleet, and the city-wide system can consist of individual either driver-operated or autonomous vehicles.

Only a minority of research uses a microscopic traffic simulation to evaluate the performance of an AMoD system [15], [16]. However it still does not feature dynamic ride-sharing, and when rebalancing is available, it relies on human operators [16].

The only approaches that consider congestion with ride-sharing and/or rebalancing are [9], [12], [13], [14]. However

TABLE I: Main dynamic ride-sharing (DRS) and rebalancing (RB) approaches, using microscopic simulation for Shared Autonomous Vehicles and Congestion-Aware Routing (CAR) strategies

Reference	DRS	RB	Micro-sim	CAR
Maciejewski and Bischoff [15]	✗	✗	✓	✓
Hörl <i>et al.</i> [13]	✗	✓	✗	✗
Zhang <i>et al.</i> [6]	✓	✗	✗	✗
Simonetto <i>et al.</i> [7]	✓	✗	✗	✗
Lu <i>et al.</i> [8]	✓	✗	✗	✗
Levin <i>et al.</i> [9]	✓	✗	✗	✓
Martinez <i>et al.</i> [16]	✗	✓	✓	✗
Fagnant and Kockelman [3]	✓	✓	✗	✗
Fiedler [10]	✓	✓	✗	✗
Alonso-Mora <i>et al.</i> [4]	✓	✓	✗	✗
Wen <i>et al.</i> [11]	✓	✓	✗	✗
Vosooghi <i>et al.</i> [12]	✓	✓	✗	✗
Ruch <i>et al.</i> [14]	✓	✓	✗	✗
SAMoD Guériau and Dusparic [5]	✓	✓	✗	✗
This paper	✓	✓	✓	✓

none uses a congestion-aware routing strategy to react to observed congestion level. [9], [12] only account for congestion generated by the fleet of SAVs themselves rather than other vehicles, and in [14], [13] the agent-based simulation used relies on a simplified queue-based model for traffic congestion instead of more realistic car-following behaviours. In other approaches, simplifications were used to mimic the behaviour in the presence of congestion, e.g., modifying the speed of SAVs per simulated period: one speed at rush hour, and another during the quiet period [6]. Some approaches aim to estimate travel times more realistically, e.g., by using a microscopic simulator to estimate finer travel time per link [16], or OpenStreetMap (OSM) data to simulate time-varying traffic condition by applying a hourly velocity ratio [7]. While those contributions allow for more realistic SAV travel times, the effect of traffic on SAVs remains to be assessed. The only work that models traffic and SAVs at the same time is [15], which simulated private vehicles mixed with autonomous taxis and assessed the potential impact on traffic congestion, however taxis did not feature ride-sharing or relocation strategies. With ride-sharing and rebalancing enabled, SAV systems could have a strong impact on regular traffic and vice versa, effect of which still remains to be evaluated.

To address this gap, we evaluate the influence of regular traffic, *i.e.*, private cars, on the behaviour of a fleet of SAVs, as well as the influence of varying sizes of SAV fleets on regular traffic. We extend our previous work SAMoD [5] in which we proposed a fully decentralized RL approach to ride-sharing and vehicle rebalancing. We investigate whether SAMoD's achieved behaviour holds in the presence of congestion, and what effect does it have on the overall traffic flow. We use microscopic simulation to reproduce a realistic traffic flow in NYC network, and modify the algorithm to take the congestion into account when calculating routing and pick up decisions.

III. ON-DEMAND AUTONOMOUS RIDE-SHARING ENABLED MOBILITY

This section presents the details of a single SAMoD agent design, and the overall system architecture and behaviour aris-

ing from individual agents operating in the same environment.

A. SAMoD agent

Each vehicle in SAMoD is controlled by an intelligent learning-based SAMoD agent. To implement the learning process, we use Q-learning [17], a widely used RL algorithm. In Q-learning, for each state (set of environment conditions), an agent, in interaction with environment from which it receives rewards (reinforcements), learns the most suitable actions to perform in the long-term.

SAMoD agent’s main goal (Q-learning policy) is to serve ride requests, *i.e.*, to pick up passengers. RL reward is given to an agent if the vehicle has passenger(s) and no reward is given otherwise. This goal can be achieved in two ways: serving current pending requests, including additional ride-sharing requests, and, if there are no requests in agent’s vicinity, by rebalancing to the areas where there are requests or from where historically the most requests originate. Each agent, by receiving this reward signal, learns its own way to best maximise the cumulative reward, by learning the actions for each of its states that lead to maximizing it.

The environment is divided into zones, provided during the system implementation, and requests are classified depending on the origin and destination zones. Each agent makes action decisions considering only a subset of zones: the zone it is currently in, and its direct neighbouring zones. The full RL agent state consists of (i) vehicle state (*empty*, *hasPassengers*, *full*), (ii) presence of current requests in agent’s zone (*yes*, *no*) and (iii) presence of current active in neighbouring zones (*yes*, *no*). We make a distinction between *full* and *hasPassengers* rather than just mark the vehicle as occupied, to denote if vehicle is available for further ride-sharing requests. An agent can execute three basic actions: *pickUp*, *rebalance* and *doNothing*. Pick-up action can refer to picking up the first passenger, or picking up a ride-sharing request. With respect to rebalancing, agents learn to choose between a number of strategies: (i) rebalance to a neighbouring zone with the most current pending requests, (ii) rebalance to a neighbouring zone with the biggest gap between vehicle supply and number of requests, (iii) rebalance to a neighbouring zone which historically has the most requests, or (iv) rebalance to a neighbouring zone which historically has the biggest gap between vehicle supply and number of requests. At the start of the learning process, agent does not have historical information; as learning episodes are executed, an agent records the observed requests in each zone it operates in and builds up the required information for the more sophisticated rebalancing strategies.

SAMoD learning process is summarized in Algorithm 1. A new learning episode is triggered by either dropping off passengers or by finishing rebalancing to a different zone.

B. SAMoD system

SAMoD agents are designed to be a part of fully flexible stationless shared on-demand mobility system, in which ride-sharing requests are considered in real-time as they come up; no knowledge of future requests is assumed, although

Algorithm 1 SAMoD Agent Learning Process

```

1: // execute on learning event trigger
2: state ← MAPLOCALENVTOSTATE(environment)
3: action ← QLEARNING.PICKACTION(state)
4: // if picking up new (ride-sharing) passengers
5: if action == "pickUp" and state! = "full" then
6:   request ← NEARESTREQUEST(roadNetwork)
7:   if state! = "empty" then
8:     request ← RIDESHARINGREQUEST(roadNetwork)
9:   end if
10:  route ← CALCROUTE(roadNetwork, trafficData)
11:  PICKUPPASSENGERS(request)
12: // if rebalancing, determine the zone to rebalance to
13: else if action == "rebalance" then
14:   current ← GETPENDINGREQUESTS(zones)
15:   historical ← GETHISTORICALREQUESTS(zones)
16:   zone ← SELECTZONE(current, historical)
17:   route ← CALCROUTE(roadNetwork, trafficData)
18:   REBALANCETO(zone, route)
19:   TRIGGERNEWEVENT(Rebalanced)
20: end if
21: // serve requests and finish learning episode
22: if state! = "empty" then
23:   route ← CALCROUTE(roadNetwork, trafficData)
24:   DROPOFFPASSENGERS(request, route)
25:   TRIGGERNEWEVENT(DropOff)
26: end if
27: UPDATEENVIRONMENTHISTORICALDATA(request)
28: UPDATEQLEARNINGPROCESS(request)

```

historical data is gathered throughout an agent’s lifetime to learn where requests typically originate.

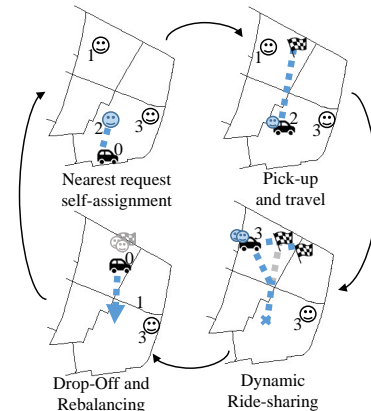


Fig. 3: Example learnt behaviour of a SAMoD. Numbers refer to passengers count per request and car occupancy.

Figure 3 depicts the learnt SAMoD behaviour. At the start of an episode, agent receives information on available requests in its own zone and picks up the nearest request. After pick-up, the vehicle drives to the rider destination (top right image). The original route can be modified if the vehicle detects further requests eligible for ride sharing. Action *pickUp* is then executed to pick another passenger, route and destination are updated to include additional passengers, and passengers

are dropped off to their respective destinations (bottom right image). Upon completion of this episode, if there are requests in its own zone, vehicle can assign itself to the nearest one, and go through the same cycle again. If there are no requests, as is the case in the depicted example, vehicle learns to either wait in the same zone or relocate itself to one of the neighbouring zones. No direct collaboration or coordination between vehicles exist. Vehicles instead coordinate indirectly through the environment; each agent has visibility of how many other vehicles are present in their zone, so can decide to rebalance to the zone with the biggest gap between requests and vehicle supply (or be purely guided by requests and to remain "competing" with other vehicles in the same zone).

An advantage of this decentralized approach is that no adjustment is needed to the system or algorithm if a vehicle joins or leaves the system (unlike in centralized approaches where recalculation/reassignment is needed).

Figure 4 depicts the SAMoD environment and the interactions between agents and the environment, from which the overall SAMoD system performance arises in a decentralized bottom-up manner. System provides the map of the operating environment, which includes the road network as well as its division into "zones", smaller segments of the map introduced to discretize the environment. Upon joining the system, each agent/vehicle initializes its learning process; first time it joins it starts from no knowledge and builds it up as it operates. Each agent has access to the information about its own local environment only - number of the requests and other vehicles in its own zone and neighbouring zones.

The framework is designed to allow a straightforward porting of SAMoD to a real fleet of SAVs. The update of vehicles position and speed is performed by a separate process, in this case a microscopic traffic simulation. While SAMoD environment simulates the requests, SAMoD cars behave and learn independently. Their current position is given by the traffic simulation at every time step, and allow them to compute distances to requests in their zone. The key difference here, when compared to most of state of

the art approaches, is that our modelling allows vehicles to compute distances to potential requests based on the network representation, *i.e.*, based on the shortest path between the vehicle position and the waiting rider (Algorithm 1, lines 6 and 8). In addition, our framework allows SAVs to update their route dynamically based on current traffic conditions. SAVs receive information about the current average travel time on each link of the network, and based on this compute the fastest route (in terms of estimated travel time) towards their next destination. In Algorithm 1, we show how congestion-awareness is incorporated into agents actions: the dynamically computed route is passed as a parameter into pick-up (line 11), drop-off (line 24) and rebalance (line 18) methods. This allows SAMoD cars to adapt their route to current traffic, hence avoiding the most congested links on the network.

IV. EXPERIMENTAL SETUP

The evaluation of SAMoD is implemented in a microsimulator SUMO [18]. Vehicle speed and position are updated using a car-following model, reproducing high-level traffic behaviour such as congestion.

We used the open NYC taxi dataset [2] to generate ride requests in the lower Manhattan area. The road network, its division into zones, speed limits and traffic lights positions were extracted from Open Street Maps data and include 1450 nodes, 2846 links and 877 traffic lights, divided into 19 provided zones.

To generate the ride-sharing requests, we extracted the trips from 50 consecutive Tuesdays between July 2015 and June 2016 to represent typical weekday demand. We chose to examine the morning rush hour (7-10am), which in the data set is the highest demand period. We obtained a total of 184,938 trips (with a total of 294,977 passengers). Each SAV has a capacity of 4 passengers. Customers request trips, indicating their origin location (GPS coordinates) and the number of passengers. They wait for a maximum of 10 minutes before a SAV is assigned (or self-assigns) to their request. After that time, unassigned requests are cancelled

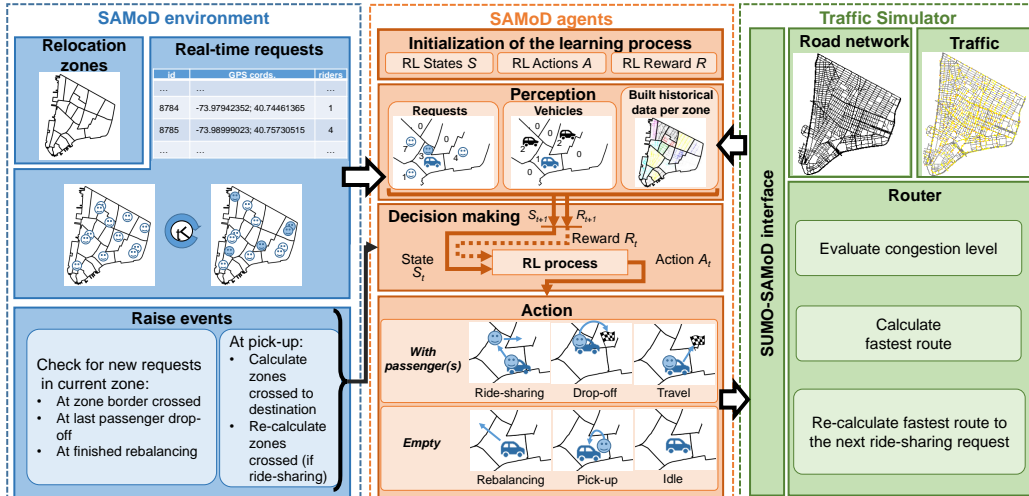


Fig. 4: SAMoD system

from the system and marked as "not served". After pick-up, based on desired passenger destination, SAV dynamically recomputes the route to the nearest destination among its current passengers.

To implement a realistic evaluation of SAMoD in the presence of traffic congestion, personal vehicle trips were generated using a uniform distribution, ensuring spread-out network occupancy. SAVs use SUMO's routing component to receive updated information on average travel time and continually recompute their routes towards pick-up requests and drop-offs. Routes for privately owned vehicles are randomly generated. Even though AVs and human-driven vehicles will highly likely exhibit different longitudinal behaviours in the future, currently proposed AV car-following models are not validated in urban scenarios, so in this work both AVs and private vehicles are ruled by SUMO's default car-following Krauss model. However, behaviour of SAMoD vehicles differs from that of the privately owned ones in other aspects: private cars drive from an origin to a destination, when AVs continuously circulate through the system dynamically changing their route to satisfy requests. In addition, each pick up and drop off action of a SAV is associated with a 3 second stop on the road to allow passengers to board and leave the car. In addition, we assume SAVs can park on the side of the road after they drop their last passenger and when they reach a rebalancing location, an assumption in line with all of the other related work [19], [3], [4], [11].

V. EVALUATION, RESULTS AND ANALYSIS

Evaluation of SAMoD approach focuses on two aspects: (i) evaluating the performance of a learnt car- and ride-sharing strategy applied by a fleet of AVs in the presence of other traffic, to observe any potential differences between naive evaluations used by state of the art approaches (and our previous work) which simulate only SAVs (Section V-A); and (ii) evaluating the overall performance of traffic with varying SAV fleet sizes and private vehicles vs. SAV vehicles ratios (Section V-B).

A. SAMoD performance in the presence of congestion

In this section, we compare the performance of a fleet composed of SAMoD vehicles in the original simulation framework [5] which did not consider congestion, and in the new framework which integrates traffic and a realistic road network. Travel times were computed by assuming a speed of 21 mph (33.8 km/h) for peak hours and of 30 mph (as in [6]). We use SUMO microsimulation to simulate actual speeds of SAVs and private vehicles taking congestion and traffic controllers in account. Therefore, current simulation gives more realistic travel and waiting times. We use a fixed fleet size of 200 SAVs.

In order to evaluate different features of SAMoD, we compare our proposed approach to several baselines combining centralized assignment, decentralizing assignment, rebalancing and ride sharing, defined as follows:

- *Centralized assignment* (C): a central controller assigns nearest vehicle to the request with highest waiting time.

- *Decentralized assignment* (D): vehicles self-assign to the request with the highest waiting time.
- *Rebalancing* (RB): the vehicle drives towards the centre of its "home" zone.
- *Ride sharing* (RS): a vehicle can serve multiple requests in the same or neighbouring zone simultaneously, until reaching maximum occupancy.

TABLE II: Simulated configurations: 4 baseline scenarios and 3 configurations of SAMoD

Summary	Assignment	Rebalancing	Ride sharing
C	Centralized	No	No
D	Decentralized	No	No
C_RB_RS	Centralized	Yes	Yes
D_RB_RS	Decentralized	Yes	Yes
S_RB	Learnt	Learnt	No
S_RB_RS	Learnt	Learnt	Learnt current zone
S_RB_RS+1	Learnt	Learnt	Learnt current zone+1

We combined these behaviours into 4 baseline scenarios and we evaluate 3 different configurations of SAMoD agent. The scenarios are summarized in Table II.

Ride-share requests can be served only in the zone vehicle is already due to travel through, or diversions of maximum one zone are allowed. Rebalancing actions are learnt by an agent, and vehicles are allowed rebalance as many times as needed to move to desired area. For all SAMoD scenarios, SAVs Q-learning process uses an ϵ -greedy policy with 50% exploration rate (probability of taking a random action) during training, performed on the first 40 days of the dataset, and 0% during the last 10 days, *i.e.* for the presented results. Q-learning parameters learning rate α and the discount factor γ are both set to 0.1.

As a measure of performance, we compare three sets of indicators: system metrics, rider metrics, and vehicle metrics. We evaluate the overall system performance with respect to the number and percentage of served and timed-out (not served after 10 minutes) requests. For the riders, we measure the waiting time t_w between the request is sent and the rider picked up, detour time t_d , *i.e.*, the time of the detour required to serve additional ride-sharing requests, and the total travel time TT . For the vehicles, we measure a number of Vehicle Miles Travelled (VMT) metrics: empty VMT, engaged VMT, and shared VMT, as well as the vehicle occupancy.

The results obtained are summarized in Table III. For completion and consistency the table includes the performance of centralized and decentralized baselines without ride-sharing or rebalancing, however, as these are significantly outperformed by approaches which include ride-sharing and rebalancing, we only focus on those in our discussion. First, we observe the performance of SAMoD against the baselines. SAMoD with ride-sharing and rebalancing outperforms both centralized (C_RB_RS) and decentralized (D_RB_RS) baselines in terms of vehicle occupancy (3.2 versus 3.1 and 2.9 in decentralized and decentralized baselines). It also achieves significantly lower waiting times (3.8 min vs 4.31 and 4.1 in decentralized and decentralized baselines), and a higher distance travelled with more than one passenger. This improvement, however, is achieved at the cost of a higher empty VMT (67 vs. 52 for both

		Original version of SAMoD [5]						SAMoD-Sumo							
		No RB, No RS		RB and RS		SAMoD			No RB, No RS		RB and RS		SAMoD		
		C	D	$C_{-RB_{-RS}}$	$D_{-RB_{-RS}}$	S_{-RB}	$S_{-RB_{-RS}}$	$S_{-RB_{-RS+I}}$	C	D	$C_{-RB_{-RS}}$	$D_{-RB_{-RS}}$	S_{-RB}	$S_{-RB_{-RS}}$	$S_{-RB_{-RS+I}}$
System	Satisfied requests	29667	35388	38346	38407	35691	37790	37679	24062	31607	35879	36092	32089	33959	34244
	% of total requests	76.4	91.13	98.75	98.91	91.91	97.32	97.03	62.34	81.89	92.96	93.51	83.14	87.98	88.72
	Not served requests	8675	3098	0	11	2903	693	726	14049	6565	2037	1818	6203	4106	3700
	% of total requests	22.34	7.98	0	0.03	7.48	1.78	1.87	36.4	17.01	5.28	4.71	16.07	10.64	9.59
Riders	Avg t_w (min)	11.63	5.48	2.1	2.6	2.87	2.46	2.27	16.44	7.9	4.65	4.11	4.31	4.12	3.78
	Avg TT (min)	5.8	5.69	10.19	8.73	5.69	9.11	12.03	6.62	6.5	16.95	15.72	6.49	13.29	17.88
	Avg t_d (min)	0	0	4.44	2.99	0	3.39	6.31	0.13	0.06	10.5	9.28	0.05	6.89	11.48
Vehicles	Avg VMT	863.8	735.79	760.06	845.02	882.85	865.94	869.94	315.62	282.3	302.91	303.35	320.28	303.7	308.35
	Avg empty VMT	428.48	228.29	181.56	268.52	371.95	352.6	335.81	171.39	97.94	52.06	52.83	133.31	81.73	67.12
	Avg engaged VMT	435.32	507.5	578.5	576.5	510.91	513.34	534.13	144.23	184.36	250.86	250.52	186.97	221.97	241.23
	Avg shared VMT	103	120.55	376.86	301.96	115.84	330.3	433.86	33.31	43.6	197.66	183.89	40.4	149.19	199.76
	Avg occupancy	1.47	1.48	2.63	2.27	1.45	2.52	3.13	1.46	1.47	3.1	2.93	1.41	2.57	3.2

TABLE III: SAMoD [5] original version (left) and new Sumo framework (right) for 7-10am period (10 days)

baselines), as the dynamic relocation strategy makes the vehicles travel a lot more. In addition, SAMoD serves 5% less requests than the baselines. Therefore, SAMoD outperforms the baselines in terms of passenger metrics (shorter waiting time) and vehicle/driver metric (increased vehicle occupancy resulting in increased profits), while it performs worse with respect to overall system metrics (number of requests served). The improvement on passenger and vehicle side is not surprising, as our implementation of RL-based agents receives rewards for picking up passengers. Potential real-world deployments might require a more fine-tuned balance between these metrics using more sophisticated multi-objective optimization strategies, and could be modified to include rewarding agents for reaching service targets to increase the number of requests served.

The second conclusion arising from the results is that the SAV fleet exhibits a similar general behaviour in both original naive simulation and SUMO framework using real road network and congestion. However, the impact of traffic congestion is clearly visible. For instance, while a fleet of 200 SAVs was sufficient to serve nearly all requests in the original framework with enabled ride-sharing and rebalancing, a significant proportion of requests (around 10 %) are now missed by both SAMoD and the baselines. The explanation lies in the average distance travelled by the vehicles, which is significantly reduced in the presence of traffic congestion, resulting from lower average speeds (the original version of SAMoD was assuming fixed speed values).

Also, as the objective of SAMoD vehicles is to travel with passengers, ride-sharing requests are preferred, and adapting the reward could result in a better balance between level of service and share rides. Similarly, the detour time for ride-sharing enabled configuration is now almost doubled (e.g., from 6.31 to 11.48 min when picking additional passengers in neighbouring zones is allowed), as vehicles travel the same distance but with a lower average speed. Note that within the SUMO framework, detour time is computed from the expected travel time is estimated from current traffic conditions, hence it shows some variation with actual measurements (i.e., it should be zero for both C and D baselines). The observed values of travel and detour time in SUMO framework suggests that the system would benefit from an adjusted behaviour accounting

for congestion, or that vehicles should learn improved assignment and rebalancing strategies which include the current congestion level. We further discuss this in Section VI, as we believe that such congestion-awareness in assignment and rebalancing is a crucial feature SAVs need to have for real-life deployments, and as such will be investigated in a further development of SAMoD.

B. Impact of SAMoD system on traffic

In this section we investigate the impact of SAMoD on private vehicle traffic, by simulating several adoption scenarios for SAVs, as summarized in Table IV.

TABLE IV: Different SAV + private vehicles scenarios

Scenario	No of private vehicles (obs.)	No of SAVs	Total no of vehicles (obs.)	SAV %	No of people
1	2000 (1991.547)	0	2000 (1991.55)	0%	3000
2	1900 (1905.706)	100	2000 (2005.71)	5%	3150
3	1800 (1787.693)	200	2000 (1987.69)	10%	3300
4	2000 (1956.172)	200	2200 (2156.17)	9%	3600

The Table specifies the targeted number of cars in the simulation at the same time, but since traffic generation is done stochastically by SUMO, the observed number of vehicles slightly differs (the average number of vehicles is listed in parenthesis in the table). Designed scenarios investigate different penetration rates (i.e., percentage) of SAVs and several levels of demand. The total number of people travelling in the network is calculated by assuming an occupancy of 1.5 for private cars (rounding down the actual figure of 1.6 observed in real data, Section I) and 3 for SAVs (rounding down the figure of 3.2 observed in experiments in Section V-A).

Scenario 1 shows a baseline configuration where there are no SAVs present and 2000 private vehicles travel the network at any point in time. Scenarios 2 and 3 represent the 5% and 10% penetration rates for SAVs, i.e., scenarios in which 100 and 200 vehicles out of 2000 private ones were replaced by a SAV. Scenario 3 represents the same configuration used in experiments in Section V-A. Scenarios 1, 2, and 3 are therefore designed to evaluate the impact of different private vs. SAV vehicle ratios on the overall traffic. Scenario 4 shows

a pessimistic case where SAVs do not replace any private vehicles, but are added to the existing private vehicles from Scenario 1. This represents the scenario where additional 10% of SAVs are added but this increased demand is generated by people previously using other non-private vehicle modes of transport (e.g., public transport, cycling, walking). The results were obtained by performing 10 runs for each scenario. As in Section V-A, SAMoD vehicles were given 40 days to learn the behaviours, and the metrics recording started from day 40 and ended at day 50 (the recording period represents 30 hours of simulated requests).

To compare the performance between the different scenarios for private vehicles we measure average speed, average travel time and average delay (i.e., the time that was added on to the optimal travel time due to congestion). For SAVs we focus on rider metrics, and, same as in Section V-A, present average travel time and average detour time (i.e., the time that was added on to the trip due to ride-sharing).

TABLE V: Average travel time, delay time per mile and speed of normal traffic for all scenarios (10 simulations)

Scenario	1	2	3	4
Average travel time per mile (min)	1.810	1.813	1.807	1.817
Average delay time per mile (min)	0.943	0.946	0.94	0.951
Average speed in mph (km/h)	12.80 (20.59)	12.78 (20.56)	12.82 (20.64)	12.75 (20.52)

Table V shows the private vehicle metrics across the four scenarios, averaged over 10 runs. The differences between the scenarios are very minor, and follow the overall number of vehicles rather than the ratio of private vehicles and SAVs. The lowest travel time of 1.807 minutes is observed in Scenario 3 (with 1987 vehicles), followed by Scenario 1 (1991 vehicles, 1.81 minutes), Scenario 2 (2005 vehicles, 1.812 minutes), and finally Scenario 4 (2156 vehicles, 1.817 minutes). Scenario 3, with 10% of SAVs has lower travel time than the baseline scenario with no SAVs, while Scenario 2, with 5% of SAVs has longer travel time than the baseline.

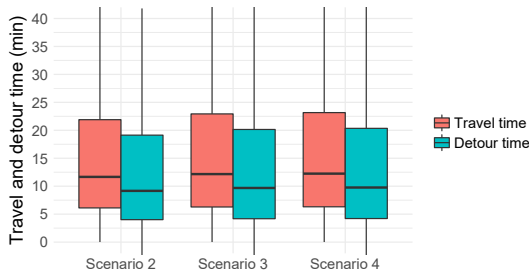


Fig. 5: Travel and detour times for SAVs trips (10 runs)

Figure 5 shows the SAV metrics across three scenarios containing SAVs (Scenarios 2, 3 and 4). Similarly to private vehicle metrics, the differences between scenarios are negligible, and mean travel time for all scenarios is about 11.5 minutes, with about 8 minutes spent on detour time to meet additional ride-sharing requests. This is not surprising as the number of private vehicles between scenarios ranges only between 1800 and 2000, and the biggest impact has already been observed in Section V-A, when private vehicles have

been added to the simulation. Based on the lack of pattern in ratio of SAVs and private vehicles relationship with travel time, we conclude that, at low penetration rates of 5-10% of SAVs, private vehicles and SAVs can co-exist without impact on the overall congestion, despite SAVs making more trips for rebalancing. They do however carry more passengers, due to ride-sharing capabilities, and as such improve the passenger capacity of the network. In Section VI we discuss the need for further evaluation of different penetration rates and congestion levels to estimate if, and at which point, SAVs would have a significant impact on overall traffic.

VI. CONCLUSIONS AND FUTURE WORK

This paper presents an extended version of SAMoD: a decentralized RL-based shared autonomous mobility-on-demand system with dynamic ride-sharing and rebalancing strategies where vehicles take congestion into account in their decisions. SAMoD was evaluated in the simulation of a real road network in SUMO, using NYC taxi data, and in the presence of congestion generated by private vehicles. This use of a realistic microsimulation framework fills an important gap in literature by being the first to carry out an evaluation of shared mobility in the presence of other traffic, as well as investigation of the impact of SAVs on congestion and vice versa. We observe that overall pattern of SAMoD behaviour with and without congestion is similar (i.e., it achieves an increased occupancy and significantly lower waiting times than the baselines but at the expense of higher distance travelled to allow for ride-sharing). However, the impact of congestion is clear when looking at absolute performance of both baseline and SAMoD; up to 10% of the requests are not served and detour time for ride-sharing is doubled in the presence of congestion. Implications of this result are twofold. Contrary to the standard in the current literature, this paper shows that the evaluation of SAV approaches needs to take congestion into account to accurately estimate level of service. In addition, results highlight the need for SAV assignment and rebalancing algorithms to be congestion-aware. In the future, vehicles should incorporate congestion information into its learning processes, to learn over time the most congested areas and/or times of the day (using techniques akin to RL-based traffic lights control responding to congestion [20] or by dynamically building the RL state space [21]). We have also evaluated the impact of SAMoD on travel times of private vehicles. At low penetration rates expected in the early deployments (5-10%), SAVs have no negative impact on the overall congestion, despite making more trips for rebalancing and having frequent stops, and are therefore unlikely to affect traffic, while having the benefit of carrying more passengers. Potential real-world deployments of SAMoD will require using fine-tuned multi-objective techniques (e.g., DWL [22]) to balance the trade-offs between passenger, vehicle and system objectives, as literature pointed out the importance of these factors for ensuring AMoD systems acceptability [6], [10]. Knowledge sharing between SAMoD agents could be used to improve the learnt behaviours and increase the speed of learning, via, e.g., transfer learning [23]. While this paper presents an important

first step in integrating ride-sharing and congestion evaluation, more extensive evaluation of impact of SAVs on traffic and vice versa needs to be performed, with more realistic traffic patterns using real traffic data, taking into account higher penetration rates, specific car-following models for AVs, and removing parking availability assumptions.

ACKNOWLEDGEMENTS

This research has been sponsored in part by a research grant from Science Foundation Ireland (SFI) under Grant Number 16/SP/3804 and by the Irish Research Council through "Surpass: how shared autonomous cars will transform cities" New Horizons award.

REFERENCES

- [1] J. Miller and J. P. How, "Predictive positioning and quality of service ridesharing for campus mobility on demand systems," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1402–1408.
- [2] NYC Taxi and Limousine Commission, "Tlc trip record data," 2017. [Online]. Available: <http://www.nyc.gov>
- [3] D. J. Fagnant and K. M. Kockelman, "Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in austin, texas," *Transportation*, vol. 45, no. 1, pp. 143–158, 2018.
- [4] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," *Proceedings of the National Academy of Sciences*, vol. 114, no. 3, pp. 462–467, 2017.
- [5] M. Guérliau and I. Dusparic, "Samod: Shared autonomous mobility-on-demand using decentralized reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1558–1563.
- [6] W. Zhang, S. Guhathakurta, J. Fang, and G. Zhang, "The performance and benefits of a shared autonomous vehicles based dynamic ridesharing system: An agent-based simulation approach," in *Transportation Research Board 94th Annual Meeting*, January 2015.
- [7] A. Simonetto, J. Monteil, and C. Gambella, "Real-time city-scale ridesharing via linear assignment problems," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 208–232, 2019.
- [8] M. Lu, M. Taiebat, M. Xu, and S.-C. Hsu, "Multiagent spatial simulation of autonomous taxis for urban commute: Travel economics and environmental impacts," *Journal of Urban Planning and Development*, vol. 144, no. 4, p. 04018033, 2018.
- [9] M. W. Levin, K. M. Kockelman, S. D. Boyles, and T. Li, "A general framework for modeling shared autonomous vehicles with dynamic network-loading and dynamic ride-sharing application," *Computers, Environment and Urban Systems*, vol. 64, pp. 373 – 383, 2017.
- [10] D. Fiedler, M. Certický, J. Alonso-Mora, and M. Cáp, "The impact of ridesharing in mobility-on-demand systems: Simulation case study in prague," in *21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1173–1178.
- [11] J. Wen, J. Zhao, and P. Jaillet, "Rebalancing shared mobility-on-demand systems: A reinforcement learning approach," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Oct 2017, pp. 220–225.
- [12] R. Vosooghi, J. Puchinger, M. Jankovic, and A. Vouillon, "Shared autonomous vehicle simulation and service design," *Transportation Research Part C: Emerging Technologies*, vol. 107, pp. 15 – 33, 2019.
- [13] S. Hörl, M. Balac, and K. W. Axhausen, "Dynamic demand estimation for an amod system in paris," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 260–266.
- [14] C. Ruch, S. Hörl, and E. Frazzoli, "Amodeus, a simulation-based testbed for autonomous mobility-on-demand systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3639–3644.
- [15] M. Maciejewski and J. Bischoff, "Congestion effects of autonomous taxi fleets," *Transport*, vol. 33, no. 4, pp. 971–980, 2018.
- [16] L. M. Martínez, G. H. d. A. Correia, F. Moura, and M. Mendes Lopes, "Insights into carsharing demand dynamics: Outputs of an agent-based model application to lisbon, portugal," *International Journal of Sustainable Transportation*, vol. 11, no. 2, pp. 148–159, 2017.

- [17] C. J. Watkins and P. Dayan, "Q-Learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.
- [18] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018.
- [19] J. Bischoff, M. Maciejewski, and K. Nagel, "City-wide shared taxi: A simulation study in berlin," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Oct 2017, pp. 275–280.
- [20] I. Dusparic, J. Monteil, and V. Cahill, "Towards autonomic urban traffic control with collaborative multi-policy reinforcement learning," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 2065–2070.
- [21] M. Guérliau, N. Cardozo, and I. Dusparic, "Constructivist approach to state space adaptation in reinforcement learning," in *2019 IEEE 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, June 2019, pp. 52–61.
- [22] I. Dusparic and V. Cahill, "Autonomic multi-policy optimization in pervasive systems: Overview and evaluation," *ACM Transactions on Autonomous and Adaptive Systems*, April 2012.
- [23] A. Taylor, I. Dusparic, M. Guérliau, and S. Clarke, "Parallel transfer learning in multi-agent systems: What, when and how to transfer?" in *2019 International Joint Conference on Neural Networks (IJCNN)*, July 2019, pp. 1–8.



Maxime Guérliau is a Research Fellow in the School of Computer Science and Statistics, Trinity College Dublin. He obtained his Ph.D. in Computer Science in 2016 from Lyon 1 University, France. His main projects include Irish Research Council research project "Surpass: how shared autonomous cars will transform cities" in 2018. His research interests are distributed intelligent complex systems, unsupervised machine learning, vehicular simulation, and traffic modelling for C-ITS applications.



Ransford A. Acheampong is a Presidential Academic Fellow in Urban Studies at the University of Manchester, United Kingdom (UK). He earned his MPhil (2012) and PhD (2017) degrees from the University of Cambridge, UK. Ransford's research themes include, Land Use and Transport Interaction modelling, socio-spatial impacts of emerging mobility paradigms and travel behaviours, focusing on the interaction between built-environment and psychological factors in sustainable mobility choices.



Federico Cugurullo is Assistant Professor in Smart and Sustainable Urbanism at Trinity College Dublin. He earned his PhD in Urban Geography and Ecological Urbanism in 2014 at King's College London. Federico's research is positioned at the intersection of urban geography, political philosophy and experimental urbanism, and explores how ideas of sustainability are cultivated and implemented across geographical spaces, with a focus on projects for eco-cities and smart cities.



Ivana Dusparic is an Ussher Assistant Professor in the School of Computer Science and Statistics at Trinity College Dublin since 2016. She holds a BSc from La Roche College, PA, USA (2003), and MSc (2005) and PhD (2010) from Trinity College Dublin. Her expertise and research interests lie in the use of Artificial Intelligence techniques (machine learning, intelligent agents and multi-agent systems) to achieve autonomous optimization of large-scale heterogeneous infrastructures, applied to smart cities and sustainable urban mobility.