Rebalancing Shared Mobility-on-Demand Systems: a Reinforcement Learning Approach

Jian Wen*, Jinhua Zhao† and Patrick Jaillet‡
* Department of Civil and Environmental Engineering
† Department of Urban Studies and Planning
‡ Department of Electrical Engineering and Computer Science
Laboratory for Information and Decision Systems, Operations Research Center
Massachusetts Institute of Technology, Cambridge, MA 02139-4307
Emails: wenj@mit.edu, jinhua@mit.edu, jaillet@mit.edu

***Abstract*—Shared mobility-on-demand systems have very promising prospects in making urban transportation efficient and affordable. However, due to operational challenges among others, many mobility applications still remain niche products. This paper addresses rebalancing needs that are critical for effective fleet management in order to offset the inevitable imbalance of vehicle supply and travel demand. Specifically, we propose a reinforcement learning approach which adopts a deep Q network and adaptively moves idle vehicles to regain balance. This innovative model-free approach takes a very different perspective from the state-of-the-art network-based methods and is able to cope with large-scale shared systems in real time with partial or full data availability. We apply this approach to an agent-based simulator and test it on a London case study. Results show that, the proposed method outperforms the local anticipatory method by reducing the fleet size by 14% while inducing little extra vehicle distance traveled. The performance is close to the optimal solution yet the computational speed is 2.5 times faster. Collectively, the paper concludes that the proposed rebalancing approach is effective under various demand scenarios and will benefit both travelers and operators if implemented in shared mobility-on-demand systems.**

***Index Terms*—rebalancing, mobility-on-demand, reinforcement learning, deep Q network, ride-sharing, agent-based simulation.**

I. Introduction

The concept of shared mobility-on-demand (shared MoD) systems describes an emerging mode of transportation in which vehicles are no longer personally owned while services are tailored as per the immediate requests. Despite many of the debates with regards to regulation and societal impact, this idea is believed to be one of the most promising approaches to reform urban transportation. On the one hand, the on-demand mobility is able to connect more people with timely and flexible service. On the other hand, sharing, taking a variety of forms including car-sharing and ride-sharing, helps reduce travel costs and enables more affordable trips.

Researchers have been seeking solutions to optimize the service design and improve its operational performance. The efforts have led to advanced assignment algorithms [1], [2], dynamic prediction tools [3], [4], system design evaluation [5]–[7], dynamic pricing [8], [9], and the pursuit of full autonomy. Rebalancing (or interchangeably repositioning, relocating), which consists of actively distributing idle vehicles to regain the demand-supply balance, also appears among the list recently [10]–[15].

Stemming from the spatial and temporal mismatch between demand (trips, travelers or requests) and supply (vehicles), the rebalancing problem can often be seen in transportation systems like car rental [16], [17] and public bike sharing [18], [19]. Rebalancing MoD systems distinguishes itself from the aboves in the following aspects: (1) MoD systems are very demand responsive thus rebalancing should be done in real time; and (2) free-floating MoD systems provide door-to-door service, making the decision space continuous. In addition, if rides are shared, measuring the supply availability becomes even more critical, since a partially-occupied vehicle is still (conditionally) available to new requests as long as the dispatching constraints are satisfied.

Existing works adopting network-based optimization approaches are usually computationally demanding. In addition, as far as the authors are aware, none of them has been aiming for the free-floating systems and ride-sharing has also been omitted for the sake of simplicity. However, free-floating and ride-sharing are indeed two key elements that ensure the connectivity and affordability of the MoD service. For this matter, this paper will incorporate both of the missing elements into the shared MoD system and propose a reinforcement learning approach that applies deep Q network (DQN) for fast and effective solution. The contribution of the paper is therefore twofold: (1) it makes a transition from the station-based systems (discrete space) to free-floating systems (continuous space) and adds ride-sharing features to the model; (2) the proposed DQN-based approach for rebalancing shared MoD systems is model-free, real-time, demand predictive, computationally scalable and has good performance in terms of both level of service and operational cost.

The rest of the paper is organized as follows. Section II will review some of the most relevant works in the literature and identify the research areas to which we can contribute. Section III will formulate the optimal rebalancing problem (ORP). Two benchmark methods - the heuristic optimal rebalancing (HOR) and the simple anticipatory rebalancing (SAR) - are presented

978-1-5386-1526-3/17/$31.00 ©2017 IEEE

and we develop the reinforcement learning approach. In Section IV, we will simulate and compare the effectiveness of the methods under various demand scenarios and map sizes. A case study in London will then demonstrate the algorithmic performance in the realistic urban setting. Finally, Section V will draw the conclusion and point out the directions to the future works.

II. Literature Review

Within the realm of urban transportation, the existing research works on rebalancing have been largely focused on car rental systems [16], [17] and public bike sharing systems [18], [19]. Rebalancing MoD systems, on the contrary, is a relatively new topic. Early MoD works often draw on the experience of the car rental and bike sharing counterparts and adopt network-based optimization approaches.

[10] is among the first. Based on the fluid model, this paper proposes an optimal rebalancing model and simulates it with a 12-station autonomous MoD (AMoD) system. In this system, every station reaches an equilibrium so that there are excess vehicles and no waiting customers. However, under the influence of its car rental predecessors, the proposed method is only limited to simplified station-based networks. In addition, it is only focused on the ideal equilibrium and does not touch upon the stochastic fluctuations of demand-supply interplay. In continuation to this work, [11] transforms the model into an analytical guideline for fleet sizing in conceptual AMoD systems and validates it in a Singapore case study. This strategical work still remains at the static level and provides little insights to real-time operation.

[12] extends the idea of the fluid model and presents a queueing-theoretical approach within the framework of Jackson networks. Many efforts in this paper have been made to prove that, as a closed Jackson network, the system is most efficient when inward and outward vehicle flows (including rebalancing flows) are equal at each station. The solution to an offline optimal rebalancing problem is given, and the paper argues that, if taking only current information at a specific time point, the problem could be adopted to online applications. A case study in New York City with around 8000 non-shared vehicles demonstrates the effectiveness of the method. Unfortunately, trips in the simulation still have to be clustered to fit in the station-based model.

[13] tests both offline and online policies with an agent-based simulation platform using Singapore travel data as input. The results show that about 28% and 23% less vehicles are required to guarantee the same service rate when offline and online rebalancing are in use respectively. Moreover, online policy outperforms the offline one by reducing the average wait time from 11 minutes to 9. Using a similar approach, [14] tackles the rebalancing issues from the perspective of the fleet operators. It quantifies the operational cost as a function of fleet size, customer walk aways and the utilization rate and reveals that rebalancing can reduce the cost significantly.

The problem formulation in this paper extends the online model in [12] in order to incorporate both door-to-door service and ride-sharing. It also introduces a probabilistic objective function to describe the stochasticity in request arrivals. In the next section, we will first give the formulation to the generic shared MoD system. Multiple solutions are provided thereafter, including the proposed DQN approach and HOR/SAR as two benchmarks.

III. Methodology

Consider a shared MoD system covering a predefined service area. For operational purposes, the area is divided into S zones $\mathcal{S} = \{1, 2, \ldots, S\}$ which are mutually exclusive and collectively exhaustive. The set of fleet $\mathcal{V}$ consists of V shareable vehicles of capacity K. Travelers with origins and destinations in $\mathcal{S}$ send requests and queue up. Travelers are then assigned to vehicles dynamically by the central dispatcher on a first-come-first-serve basis and no traveler will "walk away" despite the possible long wait.

A vehicle having been assigned to travelers is "in service". Otherwise, it is "idle" and available to be rebalanced. We assume that the online rebalancing algorithm is run every ΔT time units and at time $t = T$, one run is triggered. The period of study is therefore $[T, T + \Delta T]$. We also assume that, over the period, the number of incoming requests A_i in zone i follows the Poisson process with predicted arrival intensity λ_i, that is to say, $A_i \sim \text{Poisson}(\lambda_i \Delta T)$. Knowing both the demand distribution and the vehicle status, the objective of rebalancing is to maximize the service availability while limiting the rebalancing cost. Service availability is evaluated by the average wait time of requests emerging within the period, in which "wait time" is defined as the time difference between a traveler sends out the request and s/he is picked up by a vehicle; rebalancing cost is represented by total vehicle rebalancing distance traveled, that is, the total distance covered by all vehicles in the fleet due to rebalancing.

A. Optimal Rebalancing Problem

The decision variables in the rebalancing problem are $r_{i,j}$ for $i, j \in \mathcal{S}$. $r_{i,j}$ represents the number of rebalancing vehicles sent from zone i to zone j at $t = T$, i.e., the beginning of the period. Specially, if $i = j$, it represents the number of idle vehicles in zone i that remain unmoved during the period. The decision variables satisfy $\sum_{j \in \mathcal{S}} r_{i,j} = r_i$, in which r_i is the number of idle vehicles available to be rebalanced in zone i when $t = T$.

The cost of rebalancing one vehicle from i to j is noted as $c_{i,j}$. In this paper, $c_{i,j}$ is defined as below:

$$c_{i,j} = \begin{cases} cd_{i,j} & \text{if } j \text{ is accessible from } i \text{ within } \Delta T \quad \text{(1a)} \\ \bar{c} & \text{otherwise} \quad \text{(1b)} \end{cases}$$

In equation 1a, $d_{i,j}$ is the distance from i to j and the cost is proportional to the distance with a multiplier c; in equation 1b, the cost is set to be a large constant $\bar{c}$ when j is too far away from i. This is to discourage long rebalancing and guarantee that all rebalanced vehicles can arrive at their destinations before the period ends.

We assume that both in-service and idle vehicles follow the shortest routes when picking up/dropping off travelers and rebalancing. We also assume that the planned routes are not influenced by the incoming requests over the period and travel time is deterministic. Consequently, the level of supply at $t = T + \Delta T$, i.e., the end of the period, could be measured by the availability of both idle and in-service vehicles at that time. Note, a vehicle is classified as "idle" or "in-service" only according to its status before rebalancing starts.

The availability of idle vehicles at $t = T + \Delta T$ is measured by r'_j, the number of idle vehicles that zone j will receive at the end. $r'_j \triangleq \sum_{i \in \mathcal{S}} r_{i,j}$. An in-service vehicle may also become available (if all on-board passengers have been dropped off) or partially available (if it still has passenger(s) on board but admits ride-sharing) at the end. Similarly, we define s'_j as the availability of in-service vehicles in zone j at $t = T + \Delta T$. s'_j could be expressed as:

$$s'_j = \sum_{v \in \mathcal{V}} I'_j(v) w(l'_v), \forall j \in \mathcal{S} \tag{2}$$

$I'_j(v)$ is the indicator function that equals 1 if v is in zone j when $t = T + \Delta T$ and 0 otherwise. $w(l'_v)$ is the load-availability factor, defined as:

$$w(l'_v) = \frac{\hat{p}(l'_v)}{\hat{p}(0)}, \text{ for } 0 \leq l'_v \leq K \tag{3}$$

l'_v is the load of v when $t = T + \Delta T$. In a shared MoD system with fixed settings, $\hat{p}(l'_v)$ is the empirical probability that a vehicle of load l'_v will be assigned to the new request based on simulation results. Similarly, $\hat{p}(0)$ is the empirical probability that an empty vehicle will be assigned to the new request. For a vehicle of capacity 4, we have $w(0) = 1.0$ and $w(\cdot) = 0.4, 0.2, 0.1, 0.0$ when load is $1, 2, 3, 4$ respectively. The estimated total supply v'_j when $t = T + \Delta T$ is therefore:

$$v'_j = \lfloor r'_j + s'_j \rfloor, \forall j \in \mathcal{S} \tag{4}$$

The objective of maximizing the areawide service availability is translated to maximizing the total expected number of requests b that can be served by vehicles from the same zone at the end of the period. $b = \sum_{j \in \mathcal{S}} b_j(v'_j)$ and $b_j(v'_j)$ represents the expected number of served requests in zone j during ΔT, knowing that v'_j vehicles are in supply:

$$b_j(v'_j) = \sum_{k=0} \min(k, v'_j) \mathbb{P}(A_j = k) \tag{5}$$

Based on the definitions and assumptions, the optimal rebalancing problem (ORP) is as follows:

$$\begin{aligned} \max_{r_{i,j}} \quad & \textstyle\sum_{j \in \mathcal{S}} b_j(v'_j) - \sum_{i,j \in \mathcal{S}} c_{i,j} r_{i,j} \\ \text{s.t.} \quad & \textstyle\sum_{j \in \mathcal{S}} r_{i,j} = r_i, \forall i \in \mathcal{S} \\ & r_{i,j} \in \mathbb{N}, \forall i, j \in \mathcal{S} \end{aligned} \tag{6}$$

The equation 6 is a Mixed Integer Nonlinear Programming (MINLP) problem. When problem size is large, solving MINLP is extremely computationally burdensome and in this paper, we use a combination of incremental-optimal and branch-and-bound methods to give a close approximation to the optimum. This approximated solution is referred to as heuristic optimal rebalancing (HOR).

B. Simple Anticipatory Rebalancing

Despite the good quality in solutions, the wide use of optimal rebalancing policies is still constrained by the limit of computational capacity. Large-scale applications and simulations often opt for locally executable algorithms which decentralize the decision-making and solve the problem empirically. By bounding the problem to a small area, it reduces the complexity in real-time vehicle operation and also naturally caps the induced rebalancing distance without explicitly describing the cost. One representative example is [15], which gives an intuitive method for local rebalancing: if a zone's supply exceeds its expected demand or vice versa, system pushes or pulls idle vehicles to or from adjacent zones. The paper then justifies this empirical method through simulation.

We extend this method and develop here a simple anticipatory rebalancing (SAR) approach: a vehicle makes decision based on local knowledge within its neighboring area $\bar{\mathcal{S}}$ of $\bar{S}$ zones; the probability that it moves to zone j is proportional to the number of predicted requests in that zone, that is:

$$\mathbb{P}(\text{vehicle moves to } j) = \frac{\lambda_j}{\sum_{j' \in \bar{\mathcal{S}}} \lambda_{j'}} \tag{7}$$

Under this policy, a vehicle could rebalance itself with its local knowledge and avoid causing increased workload on the central dispatcher. Decisions can be made in parallel.

C. Deep Q Network

Solving the rebalancing problem requires modeling the shared MoD system deliberately. However, delicate models are barely solvable and transferable from system to system, which unfortunately discourages its use in practice.

Reinforcement learning provides a very different approach to tackling the rebalancing problem since it's model-free and requires little adjustment of the generic architecture. Recent advances in deep Q networks [20], [21] have also made it possible to handle the delay between actions and rewards as well as the sequences of highly correlated states. Research works have demonstrated that DQN has the ability to master difficult control policies including traffic controls and taxi dispatching [22], [23].

In this paper, the neural network is trained with a variant of the Q-learning algorithm [20]. The neighboring area $\bar{\mathcal{S}}$ of a specific vehicle is divided into grids and thus makes the reinforcement learning environment. The distribution of idle vehicles, in-service vehicles together with the predicted demand around it (i.e. r_j, s'_j, λ_j for all $j \in \bar{\mathcal{S}}$) build up the state. Based on an action-value function, the DQN agent takes the current state as input and returns the policy by simply selecting the action with the highest value from the set of {noop, ne, e, se, s, sw, w, nw, n}. "noop" indicates no rebalancing operation; "ne" indicates rebalancing to the

northeast adjacent zone and so on and so forth. The vehicle then executes the action and returns the reward.

The rewards is evaluated under the following rules: (1) if the vehicle is assigned to traveler(s) during the rebalancing period, we compare the environment to the one without rebalancing and calculate the save in wait time as reward; the episode terminates and the system moves on until the vehicle becomes idle again; (2) if the vehicle remains idle during the rebalancing period, we use a penalty (a negative constant) as reward to "punish" this rebalancing action; the episode continues. The penalty is designed as such to discourage empty-running rebalancing distance and limit the operational cost. The rewarded episodes are stored into a replay memory and we update action-value function using batched samples drawn from the memory. After the training, this action-value function is used to guide the rebalancing actions of all idle vehicles in the online algorithm.

The architecture and parameterization of DQN are described in the next section, where we'll compare the effectiveness of DQN with HOR and SAR and cast them into a case study.

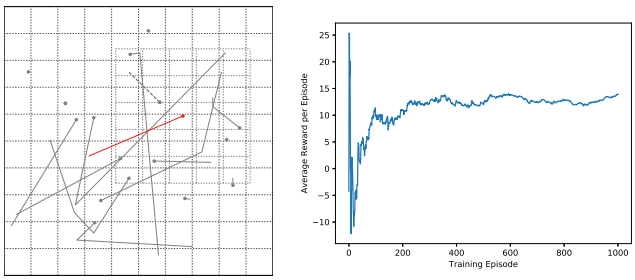
IV. Simulation and Case Study

The simulator in this section is built upon an agent-based modeling platform and details can be found in [7]. It evaluates the performance of the shared MoD system with the aid of a series of indicators, among which the interesting ones are: (1) wait time, representing the service availability from the traveler's perspective; (2) total/vehicle rebalancing distance traveled, indicating the rebalancing cost from the operator's perspective; and (3) computational time, implying it feasibility for large-scale real-time applications.

A. Benchmarking

For the sake of training efficiency, we begin the test with an abstract 5km×5km map with no road networks. Requests are drawn from a pool of origin-destination pairs with arrival rate of 100 trips/h, following the exponential distribution. 20 vehicles of capacity 4 forms the fleet, moving straight from point to point based on Euclidean distance with a constant speed of 21.6 km/h. Despite its simplification in describing the road network and the traffic, this presentation is sufficient to evaluate the effectiveness of the algorithms.

As shown in figure 1a, HOR discretizes the entire map into 10×10 fixed cells of 0.5km×0.5km. As for the local algorithms, each vehicle in both SAR and DQN has knowledge of a neighboring area of 2.5km×2.5km, which is centered at the location of the vehicle and discretized into 5×5 moving cells of the same size. For each of the rebalancing methods, simulation runs 50 times with a simulation time of 3 hours. Rebalancing is performed every 150 seconds. The DQN is trained with a three-layer neural network beforehand for 6000 steps, using ϵ-greedy behavior policy and a replay memory of 2000 most recent steps in batches of size 32. The learning rate is set to be 0.001 with no decay for the Adam optimizer [24] and the penalty for empty rebalancing is -5. Figure 1b shows that the average reward increases and reaches its height



(a) discretization (b) average reward

Fig. 1. (1a) Illustration of the discretization: points represent vehicles; lines (dashed line) originating from points represent planned routes (rebalancing routes); dotted grids represent the discretized cells for both HOR (10×10) and SAR/DQN (5×5 with red vehicle as center). (1b) Average reward per episode during one typical training.

after around 400 episodes during one typical training (irregular ups and downs are due to randomness in training). The best DQN in terms of the shortest average wait time is used in the following analysis.

We distinguish three demand scenarios: (1) the balanced scenario, in which the trip ODs are uniformly distributed on the map at random; (2) the imbalanced scenario, in which the trip origins are concentrated to two production areas and destinations to two attraction areas; and (3) the first-mile scenario, in which trip origins are uniformly distributed while the destinations are fixed to one point (e.g. an access station). The level of imbalance increases from scenarios 1 to 3 as the distributions of origins (where requests are sent, i.e., demand) and destinations (where vehicles become idle, i.e., supply) mismatch each other to a greater and greater extent. The necessity of rebalancing is therefore expected to increase.

1) Performance Comparison: Figure 2 compares the performance of the rebalancing methods presented in section III under the balanced scenario. When HOR is applied, the average traveler wait time according to 50 three-hour runs is 146.2 seconds. This is a great leap from 170.6 seconds (+16.7%) in the case with no rebalancing policy as shown in the rightmost box. With the same system settings, SAR scores 155.6 (+6.4%) and DQN scores 150.1 (+2.7%). It indicates that DQN has superiority over its local counterpart SAR with regard to high service accessibility, yet it falls behind HOR. The suboptimality of DQN could be explained by its design of individual rewarding. Without coordination with other vehicles in the training the agent (vehicle) tends to overreact to the imbalance. It is also noticed that both HOR and DQN produce smaller wait time variance. SAR, in contrast, performs in a rather random manner and the results are more dispersed. When it comes to the vehicle distance traveled, the simulation points out that all rebalancing methods would induce average vehicle distance traveled by around 30% to 35%. The rebalancing distance could be controlled by adjusting $c_{i,j}$ in HOR and the size of the neighboring area in SAR and DQN.

2) Various Demand Patterns: Table I shows how rebalancing responds to different demand patterns. As the level of

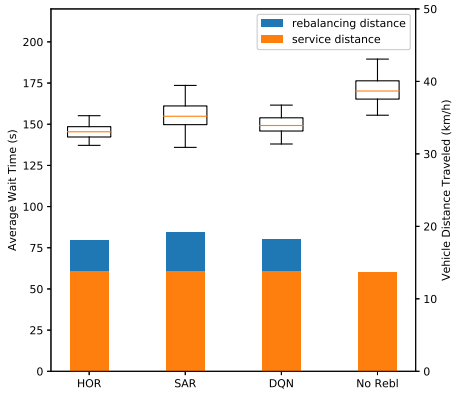


Fig. 2. Comparison of rebalancing methods: the balanced scenario.

TABLE I
COMPARISON OF AVERAGE WAIT TIMES ACROSS SCENARIOS

	HOR	SAR	**DQN**	No Rebl
Scenario 1	146.2	155.6	**150.1**	170.6
(balanced)		(+6.4%)	(+2.7%)	(+16.7%)
Scenario 2	138.5	172.5	**155.8**	211.4
(imbalanced)		(+24.5%)	(+12.5%)	(+52.6%)
Scenario 3	129.2	161.9	**151.6**	232.8
(first-mile)		(+25.3%)	(+17.3%)	(+80.2%)

* Wait times (in seconds) on top; changes compared to HOR (in percentage) in the middle.

TABLE II
COMPARISON OF AVERAGE WAIT TIMES AND COMPUTATIONAL TIMES ACROSS MAP SIZES: THE BALANCED SCENARIO

	HOR	SAR	**DQN**	No Rebl
Map 1	146.2	155.6	**150.1**	170.6
(5km×5km)		(+6.4%)	(+2.7%)	(+16.7%)
	0.033	*0.027*	*0.034*	
Map 2	147.9	164.3	**154.4**	176.3
(10km×10km)		(+11.1%)	(+4.4%)	(+19.2%)
	1.893	*0.682*	*0.822*	
Map 3	147.2	182.4	**158.2**	240.7
(20km×20km)		(+23.9%)	(+7.5%)	(+63.5%)
	259.185	*42.976*	*41.209*	

* Wait times (in seconds) on top; changes compared to HOR (in percentage) in the middle; computational times (in seconds) on bottom in italics.

imbalance increases from scenario 1 to scenario 3, the wait time with no rebalancing soars up from 170.6 to 232.8. If HOR is in action, the system performs surprisingly better when productions/attractions are more unevenly distributed, owing to the fact that agglomerated trip distribution reduces the routing difficulties in ride-sharing. The performance of DQN still resides in between HOR and SAR. However, limited to the local knowledge, SAR and DQN gradually loss their competitiveness when the demand-supply imbalance is significant only at the areawide level.

3) Scalability: We enlarge the map from 5km×5km to 10km×10km and 20km×20km and augment the demand intensity proportionally under the scenario 1. The fleet size also increases from 20 to 125 and 810 to maintain the same level of service under HOR. As shown in Table II, the increasing wait time from 170.6 to 240.7 shows a manifest necessity for vehicle rebalancing when map is large. DQN stays very close to the optimal solution, demonstrating its robust performance over different map sizes.

The computational times are also shown in Table II. Each value represents the average running time for one rebalancing solution using a machine with 2.7 GHz Intel Core i5 and 8 GB memory. HOR is undoubtedly the most computationally demanding one. The computational time increases drastically as the map expands since its complexity is proportional to the product of the fleet size and the number of cells in the area. This might raise a challenge when the scale of application grows. SAR and DQN, in contrast, perform much faster when the area is large and could be further distributed and computed in parallel. This structure evidently facilitates the application to large-scale networks, especially when autonomous vehicles are used.

B. Case Study: London Shared MoD

Now we test our rebalancing methods on a 150km^2 road network in Orpington, London.

According to the travel data over the past 10 years, residents within this area make over 40,000 trips during the morning peak hours of a typical workday. The demand intensity for shared MoD is estimated to be 1145 trips/h, which accounts for around 12% of the motorized trips [7]. In this series of simulations, we use these trips as the demand pool and adopt a more realistic system setting: the request time window is 8 minutes (traveler walks away if wait time exceeds this value) and the maximum detour factor to be 1.5 (ride-sharing is acceptable only if the actual travel time is less than 1.5 times of the shortest travel time). We further assume that the system does not reject travelers unless the above constraints cannot be satisfied and the "walk away rate" should not exceed 10% to ensure the service availability.

As shown in Figure 3, the average wait time decreases as the fleet size grows, making the decreasing curves. If the operator promises the average wait time to be less than 3 minutes, about 230 vehicles should be put in service when no policy is adopted to deal with the imbalance of origins and destination (morning trips usually feature "home to work" or "suburb to downtown" asymmetry). If it chooses HOR, SAR or DQN to rebalance during operation, the fleet size could be largely reduced to 96, 122 and 105, while the 3-minute wait time is guaranteed. The computational time for one rebalancing solution is on average 118.8, 55.1 and 42.4 seconds.

Since the vehicles are used much more efficiently under DQN, the average vehicle distance traveled also increases from around 14km per hour to around 29km, of which 23km are for servicing travelers and 6km for rebalancing. However, because of ride-sharing, the total distance traveled by the fleet does not

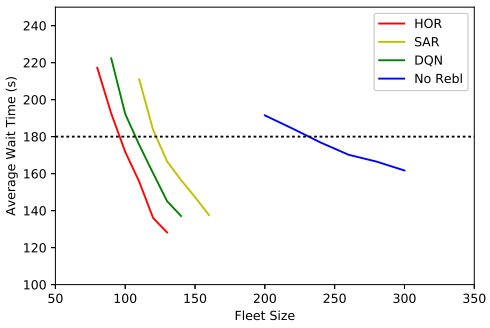


Fig. 3. Fleet Sizing under Different Rebalancing Policies. Each curve represents the relation between fleet size and average wait time under a specific rebalancing method. The dashed line defines the level of service.

grow as much (∼3538km vs. ∼3220km), although rebalancing induces around 30% more empty miles.

V. Conclusion and Future Work

This paper develops a DQN-based reinforcement learning approach for rebalancing the shared mobility-on-demand system and proves its effectiveness through an agent-based simulation platform. Results show that DQN performs effectively by reducing the wait time of travelers and limiting the distance traveled by vehicles. The London case study also demonstrates that the proposed approach is robust in a realistic setting. Although its performance is still second to the network-based optimal, the model-free DQN has revealed its potential in this field originally dominated by operation research models, particularly when scales are large and stochasticity hinders the quality of the optimal formulation.

The very next step of this work is to customize the reinforcement learning architecture to better serve the shared MoD systems. Several directions might be worthy to follow up:

1) moving from discrete action space to continuous one to avoid discretizing rebalancing actions;
2) extending the deep Q network to multi-agent models to correct overreacting;
3) redefining the reward to represent different performance metrics from the perspective of both operators and travelers.

Modeling traffic and customer behaviors in a stochastic manner might also be of interest to get closer to the real world.

Acknowledgment

The authors would like to thank Transport for London for providing trip data and road network used in this study. We also wanted to express our gratitude to Neema Nassir, Yuxin Leo Chen, Han Qiu and many other members from MIT JTL Mobility Lab and MIT Transit Lab for their comments and suggestions during this study.

References

[1] 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*, p. 201611675, 2017.

[2] A. Prorok and V. Kumar, "Privacy-preserving vehicle assignment for mobility-on-demand systems," *arXiv preprint arXiv:1703.04738*, 2017.

[3] J. Chen, K. H. Low, Y. Yao, and P. Jaillet, "Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 901–921, 2015.

[4] J. Miller, A. Hasfura, S.-Y. Liu, and J. P. How, "Dynamic arrival rate estimation for campus mobility on demand network graphs," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 2285–2292.

[5] Y. Shen, H. Zhang, and J. Zhao, "Embedding autonomous vehicle sharing in public transit system: An example of last-mile problem," Tech. Rep., 2017.

[6] D. J. Fagnant and K. M. Kockelman, "Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in austin, texas," *Transportation*, pp. 1–16, 2016.

[7] J. Wen, Y. Chen, N. Nassir, and J. Zhao, "Transit-oriented autonomous vehicle operation with integrated demand-supply interaction: a case study in london," 2017, working paper.

[8] M. K. Chen and M. Sheldon, "Dynamic pricing in a labor market: Surge pricing and flexible work on the uber platform." in *EC*, 2016, p. 455.

[9] H. Qiu, R. Li, and J. Zhao, "Dynamic pricing in shared mobility on demand service and itssocial impacts." Elsevier, 2017.

[10] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, "Robotic load balancing for mobility-on-demand systems," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 839–854, 2012.

[11] K. Spieser, K. Treleaven, R. Zhang, E. Frazzoli, D. Morton, and M. Pavone, "Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in singapore," in *Road Vehicle Automation*. Springer, 2014, pp. 229–245.

[12] R. Zhang and M. Pavone, "Control of robotic mobility-on-demand systems: a queueing-theoretical perspective," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 186–203, 2016.

[13] K. A. Marczuk, H. S. Soh, C. M. Azevedo, D.-H. Lee, and E. Frazzoli, "Simulation framework for rebalancing of autonomous mobility on demand systems," in *MATEC Web of Conferences*, vol. 81. EDP Sciences, 2016, p. 01005.

[14] K. Spieser, S. Samaranayake, W. Gruel, and E. Frazolli, "Shared-vehicle mobility-on-demand systems: a fleet operators guide to rebalancing empty vehicles," in *Transportation Research Board 96th Annual Meeting*, 2016.

[15] D. J. Fagnant and K. M. Kockelman, "Dynamic ride-sharing and optimal fleet sizing for a system of shared autonomous vehicles," in *Transportation Research Board 94th Annual Meeting*, no. 15-1962, 2015.

[16] B. Boyacı, K. G. Zografos, and N. Geroliminis, "An optimization framework for the development of efficient one-way car-sharing systems," *European Journal of Operational Research*, vol. 240, no. 3, pp. 718–733, 2015.

[17] G. Alfian, J. Rhee, M. F. Ijaz, M. Syafrudin, and N. L. Fitriyani, "Performance analysis of a forecasting relocation model for one-way carsharing," *Applied Sciences*, vol. 7, no. 6, p. 598, 2017.

[18] M. Dell'Amico, E. Hadjicostantinou, M. Iori, and S. Novellani, "The bike sharing rebalancing problem: Mathematical formulations and benchmark instances," *Omega*, vol. 45, pp. 7–19, 2014.

[19] S. Ghosh, P. Varakantham, Y. Adulyasak, and P. Jaillet, "Dynamic repositioning to reduce lost demand in bike sharing systems," *Journal of Artificial Intelligence Research*, vol. 58, pp. 387–430, 2017.

[20] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[21] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning." in *AAAI*, 2016, pp. 2094–2100.

[22] E. Walraven, M. T. Spaan, and B. Bakker, "Traffic flow optimization: A reinforcement learning approach," *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 203–212, 2016.

[23] M. Han, P. Senellart, S. Bressan, and H. Wu, "Routing an autonomous taxi with reinforcement learning," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 2421–2424.

[24] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.