J. Appl. Numer. Optim. 3 (2021), No. 1, pp. 105-114
Available online at http://jano.biemdas.com
https://doi.org/10.23952/jano.3.2021.1.07

LEARNING INCENTIVIZATION STRATEGY FOR RESOURCE REBALANCING IN SHARED SERVICES WITH A BUDGET CONSTRAINT

SHEN-SHYANG HO*, MATTHEW SCHOFIELD, NING WANG

Department of Computer Science, Rowan University, Glassboro, NJ, USA

Abstract. In this paper, we describe the problem of learning an optimal incentivization strategy that maximizes the service level given a fixed budget constraint for a sharing service such as bike-sharing, car-sharing, etc. in a spatiotemporal environment. The service level can be affected due to an imbalance in supply and demand at different locations during a specific time period. We describe and present our study and comparison of various reinforcement learning algorithms on a 1-D problem setting in a simulated bike-share system with a budget constraint on the incentives. We empirically study the performance of three policy gradient based reinforcement learning algorithms, namely: Proximal Policy Optimization (PPO), Trust Region Policy Optimization (TRPO), and Actor Critic using Kronecker-Factored Trust Region (ACKTR).

Keywords. Markov decision process; Reinforcement learning; Rebalancing problem; Incentivization.

1. INTRODUCTION

Resource rebalancing in shared systems is an important real-world problem, as shared mobility services such as bike-sharing and car-sharing have become very popular especially in urban areas. These systems are promoted to reduce traffic congestion, reduce pollution, and enhance public transportation. Imbalance in supply and demand may occur at different locations during a specific time period. These systems often rely on slow to respond and resource intensive methods to overcome the imbalance problem. A common method used to rebalance bike-sharing systems is to have employees go to areas with a high supply of bicycles and manually ride them or load them onto a van to take them to an area with low supply.

In this paper, we study how a dynamic incentivization system based on reinforcement learning that provides small incentives to slightly influence the choices of individual users (e.g., where to pick up the bike) thereby accumulating small changes can result in a more balanced bike-share system over time. The objective of such a user incentivization system is to increase the efficiency of bike-share systems, promote their usage and expansion, and increase their profitability. We describe and present our study and comparison of using different reinforcement learning algorithms on a simulated bike-share system in a 1-D environment with a budget constraint. We compare three policy-gradient based reinforcement learning methods: Proximal

*Corresponding author.
E-mail addresses: hos@rowan.edu (S.S. Ho), schofieldm0@students.rowan.edu (M. Schofield), wangn@rowan.edu (N. Wang).

Received October 2, 2020; Accepted December 13, 2020.

Policy Optimization (PPO) [1], Trust Region Policy Optimization (TRPO) [2], and Actor Critic using Kronecker-Factored Trust Region (ACKTR) [3]. We describe research issues and our future extension to 2-D problem setting and real-world applications.

2. Problem Definition

We define our environment as having a 1-D ordered spatial layout defining our bike-share system's layout. In this paper, we focus on dockless bike-share systems, where the 1-D space is partitioned into a sequence of n regions R=$\{r_1, r_2, \ldots, r_n\}$ and bikes can be distributed in each region freely. Docked bike-share systems can be modeled in a similar way and it is a special case of our proposed model where bikes can only be distributed at designated locations, i.e., bike stations. At time instance t, the number of bikes in a region r_i may changed due to bike removal or arrival. Upon initialization m bikes (or any shared resource) will be generated and distributed throughout the generated regions according to a defined distribution. Every hour a set of customers will be generated. Each customer is to appear at a particular region in the system and is interested in traveling to another region.

Definition 2.1 (Service level)**.** [4] Service level s is defined as the proportion of potential customers whose requests are satisfied within a predefined time period T as follows.

$$s = \frac{c-n}{c}, \tag{2.1}$$

where c is the number of potential customers and n is the number of potential customers failed to be allocated the requested resource.

Service level defined in Definition 2.1 is the performance metric to evaluate algorithms seeking to solve the rebalancing problem.

Definition 2.2 (Budget)**.** Budget b is the total amount of currency available to a system to fund incentivizations to customers over a fixed period T to make small changes in their resource selection behavior.

Definition 2.3 (Incentive)**.** Incentive o is the amount offered to a customer to move to a neighboring region with the requested resource.

Note that $o = 0$ if the requested resource is at the customer's location.

Definition 2.4 (Incentive Utility)**.** Incentive utility u_i is a measure of how beneficial an incentive to move to a neighboring region r_i is to a customer.

$$u_i = o_i - w_i$$

such that o_i is the offered incentive for the customer to move to the resource at location r_i and w_i is the customer's privately known walking cost from his/her location to the resource at location r_i (See Definition 2.6 and walking cost, $cost(i, j, d)$).

Budget b decreases by o when incentive o is paid to a customer. Offering a customer incentive is no longer possible if $o > b$.

Definition 2.5 (Request Satisfied)**.** A customer's request is satisfied if

(1) there exists a resource within the region r_i that he/she is located in and therefore is able to obtain the requested resource OR

(2) there exists a resource in the adjacent regions r_{i-1} and/or r_{i+1} such that the customer decides to obtain the requested resource based on the incentive utility u_{i-1} and/or u_{i+1}.

Section 2.1 describes the customer decision model when the resource is not available at the customer location in (2) of Definition 2.5.

For a bike-share system, the satisfied customer will occupy the requested resource (i.e., bike), removing it from the region as available for use. To simulate the movement from the start region to the destination region, the resource is moved to an in-transit buffer for the next hour until the trip is completed and the resource will appear as ready for use in the customer's destination region.

2.1. **Customer decision model and assumptions.** If there exists no requested resource at the customer's initialized location, r_i a customer can be offered incentives to move to a nearby region either r_{i-1} or r_{i+1} where the requested resource exists in the 1-D environment. The customer's decision to move to either region is determined by u_{i-1} and u_{i+1}. If $\max(u_{i-1}, u_{i+1}) > 0$, then the customer will take the incentive corresponding to the greater of u_{i-1} and u_{i+1} and move to the corresponding region to claim the available resource.

Definition 2.6 (Movement cost model)**.** A movement cost model describes either the cost for a customer to move from his/her original location to the resource location.

In this paper, we only incentivize a customer to move from his/her original location r_i to the bike location r_j. We use the movement cost model described in [5]. The walking cost

$$cost(i,j,d) = \begin{cases} 0, & r_i \text{ equals to } r_j, \\ \eta d^2, & r_i \text{ is a neighboring region of } r_j, \\ +\infty, & \text{otherwise.} \end{cases}$$

such that η is a positive weight and d is the straight line (Euclidean) walking distance from the customer's current location in r_i to the bike's location in r_j.

We only consider bikes in the adjacent regions of the customer. In other words, the customer at r_i will only accept bikes at r_{i-1} or r_{i+1} in our 1-D study. If no bike is in either location, the customer will not be satisfied. Hence, when there is no bike in r_i and the incentive utilities u_{i-1} and u_{i+1} are both negative, the customer will not be satisfied. Note that the above model is applicable to a 2-D problem with four adjacent regions.

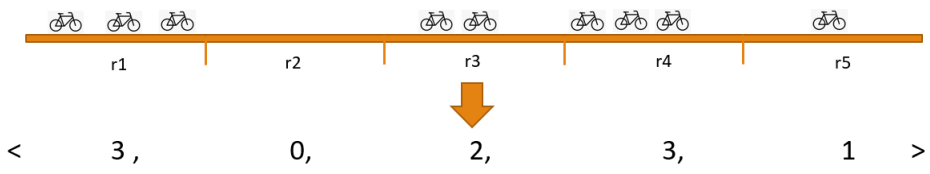

FIGURE 1. An example of a state x for our 1-D problem

2.2. **Problem formulation.** The incentive strategy for resource rebalancing for a shared service can be formulated as an online learning problem modeled as an infinite-horizon, discounted Markov Decision Process $(X, A, \gamma, P, \mathscr{R})$ such that X is the set of states, A is the set of actions, $\gamma \in (0,1)$ is the discount factor, P is the transition probability distribution, and $\mathscr{R} : X \to [0,1]$ is the reward function.

- A state $x \in X$ represents the number of bikes in each region. A state for a $n-$region problem setting can be represented by a n-element vector of non-negative integers that are the number of bikes in each region (see Figure 1). Note that we utilize a much simplified state representation for our 1-D problem compared to [5] which consists of current remaining budget, the demand, arrival and the expense in the last timestep for each region, and unservice level for each region for a fixed number of past timesteps for their 2-D scenario.
- An action $a \in A$ represents an incentive vector that assigns an incentive to each region at every hour. This action is represented by a n-element vector of non-negative integers. Note that the temporal granularity is hourly in our problem.
- Reward $\mathscr{R}(x)$: At the end of each hour $\mathscr{R}(x)$ is the service level for that hour. For successful rebalancing of resources for the hour, service level s needs to be high (i.e., near to 1).

At time t, an agent chooses an action a_t based on the policy $\pi_\theta(a|x_t)$ (θ is the policy parameter) given the current state $x_t \in X$ and a reward $\mathscr{R}(x_t)$ is calculated. The environment is then transit to the next state based on the transition probability $P(x_{t+1}|x_t, a_t)$. The goal of our problem is to maximize the expected γ-discounted cumulative return $\mathscr{J}(\theta) = E_\pi[\sum_{i\geq 0}^{\infty} \gamma^i \mathscr{R}(x_{t+i})]$.

3. Policy Gradient Methods

According to [2], there are three categories of policy optimization algorithms: (1) policy iteration methods, (2) derivative-free optimization methods, and (3) policy gradient methods. In this paper, we focus on the use of recently proposed policy gradient methods that have shown competitive performance against derivative-free policy optimization methods and have better sample complexity. We study empirically the following three policy gradient methods:

- Trust Region Policy Optimization (TRPO) [2]: This method optimizes large nonlinear complex policies by estimating local expected returns and uses the 'trust regions' to improve the policies. TRPO approximates a policy iteration scheme (see Algorithm 1 in [2]) replacing the KL divergence with a penalty in the objective function by a constraint on the KL divergence.
- Proximal Policy Optimization [1] (PPO2-GPU-enabled implementation used here): PPO improves upon TRPO achieving similar performance as TRPO by using a clipped surrogate objective function to provide a lower bound of the performance of the policy. Multiple (simple) first-order stochastic gradient ascents are used for each policy updating step using sampled data from the policy. Hence, it achieves the same performance faster (in terms of the number of episodes) and using less data as compared to TRPO (see Figure 2).
- Actor Critic using Kronecker-Factored Trust Region (ACKTR) [3]: ACKTER uses the Kronecker-factored approximation to estimate the so-called nature gradient (taking information geometry of the parameter space into account using a Riemannian metric to

adjust the direction of conventional gradient [6]) for the Actor Critic methods [7]. While Wu et al. [3] claimed that ACKTR is more efficient than the second order TRPO, our results in Figure 2 show that even though ACKTR improves faster than TRPO initially, it does not reach a good performance compared to TRPO in the long run.

4. Results and Comparisons

4.1. **Simulation setting.** We simulate the 1-D problem setting as defined in Section 2. The initial bike layout and user bike departure and arrival interests at n regions are modeled using a binomial distribution $bin(n,p)$ such that p is randomly drawn from a beta distribution with x on the interval $(0,1)$. We use a beta distribution as it can take many shapes as parameterized by α and β.

$$Beta(\alpha,\beta):\ f_X(x:\alpha,\beta)=\frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}, \tag{4.1}$$

where B is the beta function

$$B(\alpha,\beta)=\int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt.$$

This is to allow flexibility in defining variations in hourly departure and arrival counts of bike in a day during weekends and weekdays.

Specifically, we selected 5 Beta-Variate distributions and gave each an identifier B_i these were: $B_1=B(\alpha=0.5,\beta=0.5), B_2=B(\alpha=5,\beta=1), B_3=B(\alpha=1,\beta=3), B_4=B(\alpha=2,\beta=2), B_5=B(\alpha=2,\beta=5)$. All 5 Beta Variate distributions are used to determine three distributions: user arrival location, user destination locations, and initial bike layout. Thus, we have a total of 125 different simulated problem scenarios.

A user object is created with a continuous arrival location and a continuous intended destination, both within the bounds of the system [1, n]. A user will appear at his/her arrival location and if his/her request is satisfied he/she will 'leave' with the nearest bike within the region from that location and enter into a list of currently traveling users. A user's request is satisfied either when there is bike within his/her region or through an agent offering an incentive to move to a neighboring region that generates a positive incentive utility u. After the following hour he/she will 'arrive' at his/her intended destination and his/her bike will become available for use at that location.

4.2. **Results and discussions.** In our experimental results, we compare the average performance improvement in the service level for a system utilizing a learned incentive strategy from a baseline system without optimization.

4.2.1. *General performance comparison of the three policy gradient methods.* We ran each experimental trial for 20,000 episodes recording the service level. Figure 2 shows the average performance over time of each agent under 125 problem scenarios and 3 budgets (100, 200, 300). We observe both PPO2 and TRPO improve to a roughly 10% service level improvement, while ACKTR improves to a roughly 8% service level improvement. A possible explanation for this result is the similarity between PPO2 and TRPO, leading them to improve to similar performance. Another observation is that TRPO performs the most slowly and consistently, whereas PPO2 and ACKTR rapidly improve initially.

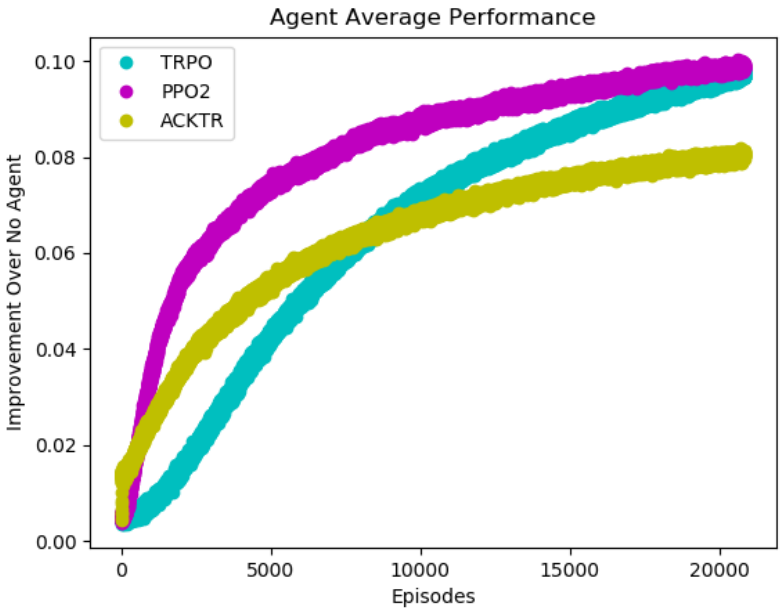

FIGURE 2. Average performance over all problem scenarios at 3 different budgets increases over time for each algorithm.

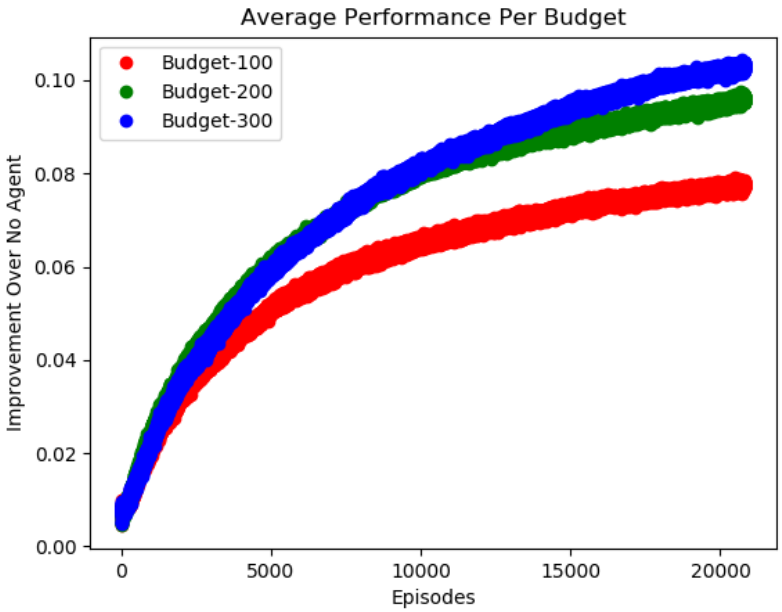

FIGURE 3. Average performance increase over three different budgets for each budget across all tested policy gradient methods and problem scenarios.

4.2.2. *Effects of budget on performance.* In this subsection we discuss the impacts of a varying budget and the result is shown in Figure 3. The experimental results show the average performance of each of the three budgets (100, 200, 300) across all three methods. We observe the expected result that each increment of 100 units to the budget improves the performance. This is the case as with more budget an agent is able to aggressively incentivize more users to promote the balance of the shared system. However, it is worth noting that the performance improvement slows down with increasing budget. Particularly, a budget of 100, 200, and 300 have a service ratio improvement of 0.07, 0.09, and 0.1, respectively. The results show that no excessive budget is needed to achieve good performance. One just needs an appropriate budget to achieve the most cost-effective rebalancing result.

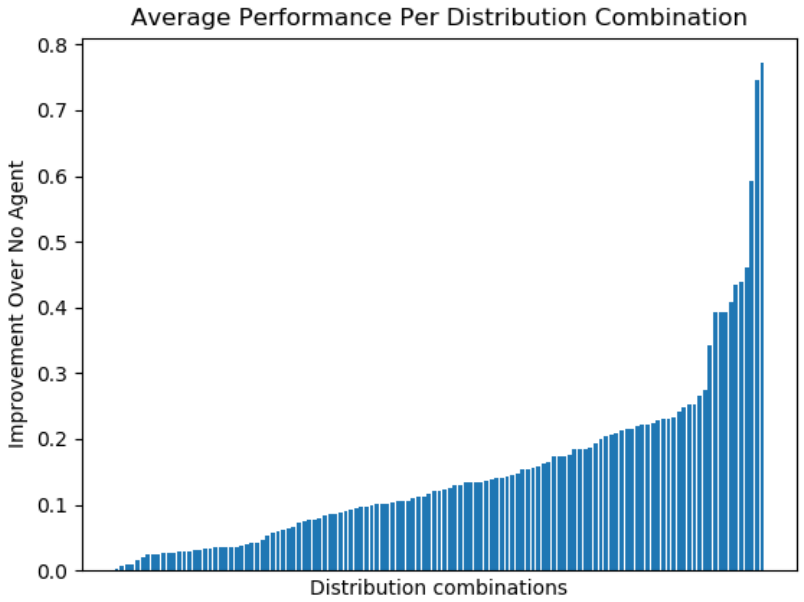

FIGURE 4. Average ending performance of every combination of three of the five Beta distributions across all test policy gradient methods and budgets.

4.2.3. *Impacts of different unbalanced problem scenarios on performance.* In this subsection we discuss the impact of various problem scenarios using the 5 Beta distributions to define the user arrival locations, user intended destinations, and initial bike layouts, described in Section 4.1, on the system performance. In Figure 4, the x-axis is the set of all combinations of three of the five Beta distributions used to defined the user arrival locations, user intended destinations, and initial bike layouts, across each algorithm (PPO2, ACKTR, TRPO) and budget (100, 200, 300). The set is sorted according to the average performance improvement in the service level for a system utilizing a learned incentive strategy from a baseline system without optimization.

We seek to show here the impact of different unbalanced situations introduced by different distribution combinations. It is worth noting that the four distribution combinations with the worst service-level improvement are in order B_2-B_2-B_2, B_5-B_5-B_5, B_4-B_4-B_4, B_3-B_3-B_3, this is the case as when all three distributions overlap the system naturally maintains a state of nearly perfect balance. The fifth complete distribution overlap is B_1-B_1-B_1 with the eighth

worst service-level improvement. The insight behind this result is that since the demand and supply of each region is perfectly balanced, there is little to no need to conduct rebalancing. The trained agents successfully learned this knowledge and did not make the situation worse. On the other hand, the following distribution combinations have the highest service-level improvement, B_5-B_2-B_5 (0.46), B_4-B_2-B_5 (0.59), B_2-B_5-B_2 (0.75), B_1-B_5-B_2 (0.77), which matches with our expectation that an unbalanced arrival and leaving distribution set leads to a large service-level improvement.

5. Related Work

Pfrommer et al. [4] is the earliest work on learning dynamic incentive mechanism to encourage users to slightly modify their destinations in exchange for a monetary incentive to solve the rebalancing problem in a shared mobility system. Later, Singla et al. [8] proposed an incentive mechanism that encourages users to pick up or return bikes from other locations using regret minimization in online learning to achieve an optimal pricing policy and it was the first time the incentive-based bike re-balancing strategy ever deployed in a real-world bike sharing system. Ghosh et al. [9] proposed an optimization model to generate the re-positioning tasks using bike trailers to rebalance the bikes and design a bidding mechanism to incentivize (with a budget constraint) those interested in taking up the re-positioning tasks using bike trailers. Lv et al. [10] proposed formulating the bike rebalancing problem as a crowd-sourcing task using a reverse auction model. To obtain the mechanism to incentivize users to park bicycles at locations towards rebalancing supply and demand, an auction model with a budget constraint is used to determine the incentive using allocation rule and the payment rule for the auction. They show that even when the budget is tight, the total revenue still exceeds or equals the budget.

Others have proposed solving the rebalancing problem without any explicit incentive mechanism. Chahchoub et al. [11] proposed the use of identifying outliers (nearly empty or nearly full bike stations). Users are proposed an alternative destination or starting points within a small area to ensure the stations are no longer outliers. Chiariotti et al. [12] proposed to solve the rebalancing and incentive problem as a joint optimization problem given the current state, arrival, and departure rate. A three-step procedure is used to approximate the solution compute the new state, adjusted arrival and departure rates. The approach requires users to pick up or drop off their bikes to enable rebalancing. There are also similar work on trip planning. Li et al. [13] discussed the optimal trip plan of users in a static setting and tried to maximize the number of the served users and minimize their trip time (similar to minimize the budget). They formulated the optimization problem as the weighted k-set packing problem and solve it via a heuristic algorithm with an approximation ratio of $1/3$. Tomaras et al. [14] proposed to maximize the number of states with equal amount of bike rent and bike return and minimize the user trip.

There are increasing research on using reinforcement learning to learn the incentive mechanism. An et al. [15] proposed an actor-critic reinforcement learning approach which uses neural network for function approximation to learn rewarding mechanism (picking up/parking bonus) for car-sharing system and allows continuous action space. The rewarding mechanism are used to guide the users' behaviors through price leverage to ensure cars are parked where needed and pick up where cars are available, and thus, boosting the company's profit and service level. Pan et al. [5] proposed using the hierarchical reinforcement learning driven by a deep deterministic

policy gradient algorithm to learn an incentive mechanism to encourage users to help rebalancing the bikes by renting from nearby locations instead of the intended locations when the service provider has a budget constraint and has an objective to maximize the daily service level. Duan and Wu [16] extended Pan et al.'s approach by learning an adaptive incentive mechanism for both renting from and returning to specific locations to solve the bike rebalancing problem under similar problem setting. Ji et al. [17] proposed an incentive mechanism that maximizes service level and users' utility while encouraging users to return bikes to regions that have a shortage of bikes within an acceptable walking distance to destinations.

Incentive mechanisms can be used in other similar applications. Mobility on Demand (MOD) service providers (e.g., Uber, Lyft, Didi, etc) proactively dispatch vehicles towards ride-seekers. He and Shin [18] proposed a spatio-temporal reinforcement learning framework for MOD coordination. Spatial distributions of demands and supplies, and the temporal factors such as events and weather conditions, are also considered. The objectives are to lower request rejection rates, shorter waiting time, and higher incentive profitability. Incentive mechanism can also be utilized to ensure appropriate driver distribution based on demand at different location and time period. Logistics is one important component of supply chain management. It consists of managing and coordinating resource movements in a timely, cost-effective and reliable manner. Li et al. [19] proposed formulating the resource balancing problem in a logistics network as a stochastic game and introduced a cooperative multi-agent reinforcement learning framework to solve the problem.

6. Future Research and Application Opportunities

We have described the problem of learning an optimal incentivization strategy in resource sharing systems through reinforcement learning. An effective incentivization strategy for rebalancing will increase the efficiency of many resource sharing systems such as popular bike-share systems that are being deployed in cities around the world. Future work for this research includes improved evaluation metrics and reward functions, improved state representations, etc. Improved evaluation metrics would provide more insights in a resource-sharing system and experimentation, and would inform the creation of improved reward functions. Budget usage should be included as an evaluation of the agent, and likely would improve agent performance if included in its reward function. An improved state representation is critical for enhancing an agent's performance. It is likely that incorporating more relevant information about the environment into the state representation and giving the agent enough time to learn on the richer input data, will increase agent performance. Including in the state representation information such as time of day, user locations, remaining budget, predicted future events, and specific bike locations, would be the appropriate next steps to increase the complexity of state space. A more flexible action representation (e.g., the movement range) to provide incentives will lead to more sophisticated incentivization strategies. We plan to further explore heterogeneous incentive strategies for different regions. Including real-world data is an important next step in our research. We have collected real user data relating to bike-share trips, we intend to use this data to perform simulations that are closer to a real-world 2-D environment. Using real-world city layouts and clustering algorithms to capture spatial constraints and obstructions is also a valuable next step to improve our simulation's reflection of reality. In addition, we will investigate new methods to evaluate the system status and figure out when to provide incentives to users.

REFERENCES

[1] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347, 2017.

[2] J. Schulman, S. Levine, P. Moritz, M. Jordan, P. Abbeel, Trust region policy optimization, International Conference on Machine Learning, ICML, Lile, vol. 3, pp. 1889-1897, 2015.

[3] Y. Wu, E. Mansimov, S. Liao, R. Grosse, J. Ba, Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation, Advances in Neural Information Processing Systems 2017 (2017), 5280-5289.

[4] J. Pfrommer, J. Warrington, G. Schildbach, M. Morari, Dynamic vehicle redistribution and online price incentives in shared mobility systems, IEEE Transactions on Intelligent Transportation Systems 15 (2014), 1567-1578.

[5] L. Pan, Q. Cai, Z. Fang, P. Tang, L. Huang, A deep reinforcement learning framework for rebalancing dock-less bike sharing systems, 33rd AAAI Conference on Artificial Intelligence, pp. 1393-1400, 2019,

[6] S.I. Amari, S.C. Douglas, Why natural gradient? Proceedings of the 1998 IEEE International Conference Acoustics, Speech, Signal Processing (ICASSP 1998), volume II, pp. 1213–1216, 1998.

[7] V.R. Konda, J.N. Tsitsiklis, Actor-critic algorithms, Advances in Neural Information Processing Systems, pp. 1008-1014, 2000.

[8] A. Singla, M. Santoni, G. Bart´ók, P. Mukerji, M. Meenen, A. Krause, Incentivizing users for balancing bike sharing systems, Proceedings of the National Conference on Artificial Intelligence, vol. 1, pp. 72-729, 2015.

[9] S. Ghosh, P. Varakantham, Incentivizing the use of bike trailers for dynamic repositioning in bike sharing systems, 27th International Conference on Automated Planning and Scheduling, ICAPS 2017; Pittsburgh, 2017.

[10] H. Lv, C. Zhang, Z. Zheng, T. Luo, F. Wu, G. Chen, Mechanism design with predicted task revenue for bike sharing systems, 34th AAAI Conference on Artificial Intelligence, AAAI 2020, United States, 2020.

[11] Y. Chabchoub, R. Ei Sibai, C. Fricker, Bike sharing systems: a new incentive rebalancing method based on spatial outliers detection, Int. J. Space-Based Situated Comput. 9 (2019), 99-108.

[12] F. Chiariotti, C. Pielli, A. Zanella, M. Zorzi, A bike-sharing optimization framework combining dynamic rebalancing and user incentives, ACM Transactions on Autonomous and Adaptive Systems, 14 (2020), 11.

[13] Z. Li, J. Zhang, J. Gan, P. Lu, Z. Gao, W. Kong, Large-scale trip planning for bike-sharing systems, Pervasive and Mobile Computing 54 (2019), 16-28.

[14] D. Tomaras, V. Kalogeraki, T. Liebig, D. Gunopulos, Crowd-based ecofriendly trip planning, 19th IEEE International Conference on Mobile Data Management (MDM), Aalborg, pp. 24-33, 2018.

[15] L. An, C. Ren, Z. Gu, Y. Wang, Y. Gao, Rebalancing the car-sharing system: A reinforcement learning method, 2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC), Hangzhou, pp. 62-69, 2019.

[16] Y. Duan, J. Wu, Optimizing rebalance scheme for dock-less bike sharing systems with adaptive user incentive, 20th IEEE International Conference on Mobile Data Management (IEEE MDM), Hong Kong, pp. 176-181, 2019.

[17] Y. Ji, X. Jin, X. Ma, S. Zhang, How does dockless bike-sharing system behave by incentivizing users to participate in rebalancing? IEEE Access 8 (2020), 58889-58897.

[18] S. He, K.G. Shin, Spatio-Temporal capsule-based reinforcement learning for mobility-on-demand network coordination, Proceedings of the World Wide Web Conference, pp. 2806-2813, 2019

[19] X. Li, J. Zhang, J. Bian, Y. Tong, T.Y. Liu, A cooperative multi-agent reinforcement learning framework for resource balancing in complex logistics network, arXiv preprint arXiv:1903.00714, 2019.