Journal of Organizational and End User Computing
Volume 33 • Issue 3 • May-June 2021

Towards Smart Transportation System:
A Case Study on the Rebalancing Problem of Bike Sharing System Based on Reinforcement Learning

Guofu Li, Ping An Asset Management, Shanghai, China
Ning Cao, University College Dublin, Dublin, Ireland
Pengjia Zhu, State Street Corp., Zhejiang, China
Yanwu Zhang, Qingdao Binhai University, Qingdao, China
Yingying Zhang, University of Shanghai for Science and Technology, Shanghai, China
Lei Li, Qingdao Binhai University, Qingdao, China
Qingyuan Li, Qingdao Binhai University, Qingdao, China
Yu Zhang, Qingdao Binhai University, Qingdao, China

ABSTRACT

Smart transportation system is a cross-field research topic that involves both the organizations that manage the large-scaled system and individual end-users who enjoy these services. Recent advancement of machine learning-based algorithms has either enabled or improved a wide range of applications due to its strength in making accurate predictions for complex problems with a minimal amount of domain knowledge and great ability of generalization. These nice properties imply potential to be explored for building smart transportation system. This paper studies how deep reinforcement learning (DRL) can be used to optimize the operating policy in modern bike sharing systems. As a case study, the authors demonstrate the potential power of the modern DRL by showing a policy-gradient-based reinforcement learning approach to the rebalancing problem in a bike sharing system, which can simultaneously improve both the user experience and reduce the operational expense.

KEYWORDS

Bike Sharing System, Machine Learning, Optimal Transportation Problem, Rebalancing Problem, Reinforcement Learning, Smart Transportation

INTRODUCTION

The newly emerged concept of smart (or intelligent) transportation system brings benefits to both the individual citizens as end-users, and the large-scaled organizations that act as service providers. Building smart transportation system involves the construction of the physical infrastructures, refining laws and regulations, improving management and operation policies, etc. (Albino, Berardi & Dangelico, 2015). Thus, services of this kind require more than just the network connections of

DOI: 10.4018/JOEUC.20210501.oa3

the involved objects, when the intelligence built inside the system at the software and management level (Zhang, Thomas, Brussel & van Maarseveen, 2016) is also crucial. For instance, to optimize a bike-sharing system may involve a long list of decisions like the best number and locations of the bike stations (Lin & Yang, 2011).

The recent achievements of smart transportation system are largely attributed to the development of the IoT technology, which connects the scattered pieces of physical objects into a large-scaled network. The scale of this kind of networks then attracts the interests from the research fields of BigData (Hashem, Chang, Anuar et al., 2016; Chourabi, Nam, Walker et al., 2012). Based on these infrastructures, the question now becomes how to enable the "smart" behaviors. The conventional approaches that rely on the techniques to implement the "smartness" require delicate human labors on problem modeling and variable selection (Villani, 2008; Zhang, Wang, Wang, et al., 2011; Lippi, Bertini & Frasconi, 2013), which are hard to generalize and transfer. Comparatively, modern deep learning based approach enables a unified solution to a wide range of problems due to its ability of universal function approximation and end-to-end learning (Lecun, Bengio & Hinton, 2015), thus offering a great opportunity to make up for this missing piece. In this paper, we are especially interested in the policy optimization problem in smart transportation systems, which is critical to both the users' experience and operation cost of the organization.

Reinforcement learning (RL), as a branch of machine learning, aims to optimize a long-term overall return, and has been studied in optimizing the transportation system (Arel, Liu, Urbanik & Kohls, 2010; Khamis & Gomaa 2014; Zolfpour-Arokhlo, Selamat, Hashim, & Afkhami, 2014). This paper argues that DRL has its special strengths in optimizing the resource allocation and schedule in large transportation system, and is tested on the rebalancing problem in bike sharing system (BSS) (Demaio, 2009; Shaheen, Guzman & Zhang, 2010). Compared with the previous methods like *Optimal Transportation* (OT) (Villani, 2008; Courty, Flamary, Remi et al., 2015) or *Pickup and Delivery Problem* (PDP) (Savelsbergh, Sol & 1995), RL approach requires very little prior knowledge on the environment dynamics, and is more flexible to meet different objectives.

BACKGROUND

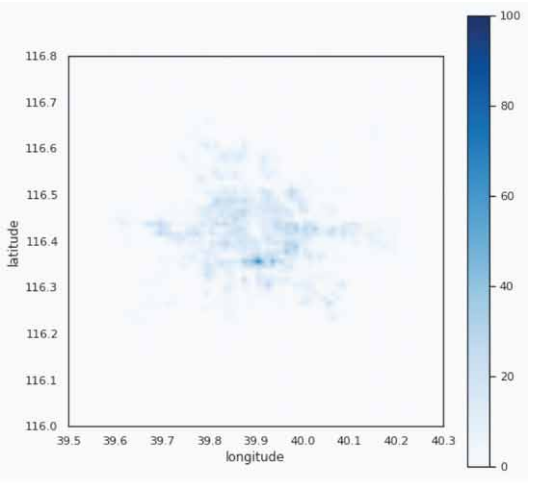
One emerging class of smart transportation services is based on the idea of sharing economy, typified by the Bike-Sharing system (BSS). A bike-sharing system is a service in which bicycles are public available for shared use by individual users on a temporary basis for a comparatively low price. Its recent success is mostly driven by the modern tracking technology, empowered by IoT and wireless sensor networks (O'Brien, Cheshire & Batty, 2014). Meanwhile, the management policy of the big system also has huge impact on various aspects of the service, especially its operation cost.

The Rebalancing Problem

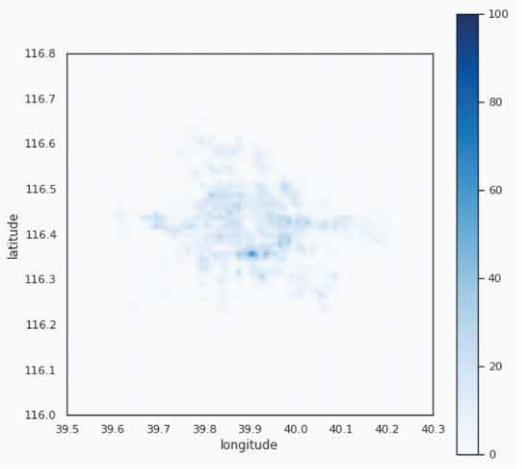
To understand the characters of the users' renting the returning behaviors, we explore the data of the Mobile BSS open data repository, which consists of about two weeks' usage-trace log of the Beijing city. Figure 1 illustrates the overall distribution of the user rental requests and return actions side-by-side. There are a few obvious patterns that we can infer from Figure 1:

1. There are several outstanding hubs that occupy a large portion of entire mass, for both rent requests and return actions. Assumedly, these hubs should comply with the city landmarks (e.g., shopping malls, metro line terminals, etc.) (Figure 2).
2. The locations of the request hubs are usually very close to the locations of the return hubs, but not identical.

Figure 1. The distributions of the rent requests and returns of the Mobike BSS in Beijing over the period between 10/May/2017 and 25/May/2017



(a) The bike rent request distribution

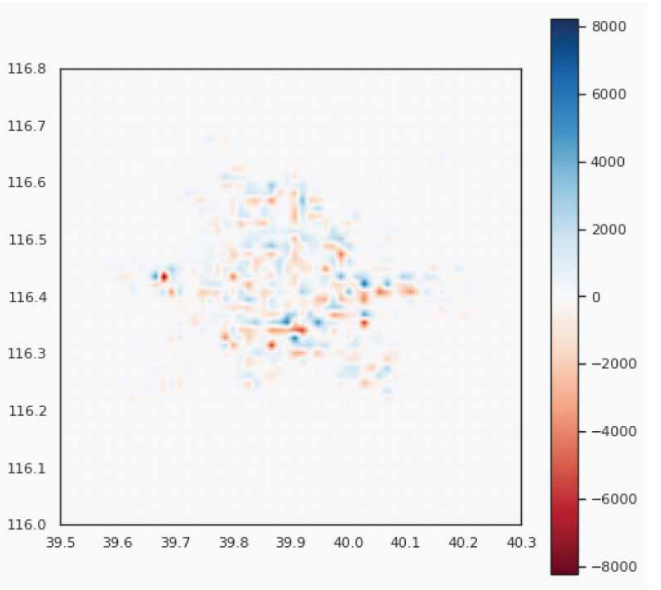


(b) The bike return distribution.

Researchers suggest that there are two broad ways to categorize the rebalancing problem in a bike sharing system (Contardo, Morency, & Rousseau 2012):

1. According to the type of the entity who conducts the reposition behavior:
   a. *Operator-based rebalancing* is conducted by the organization that runs the service;
   b. *User-based rebalancing* is conducted by the end user, encouraged by some incentive to return the bike to a requested near-by position;
2. According to the time of repositioning:
   a. *Static rebalancing* is conducted when the system is relatively static, e.g., in the midnight;
   b. *Dynamic rebalancing* is conducted during the day when the business is still running and the distribution of the bike is constantly changing.

Figure 2. The accumulated size of the gap between the rent request and returns of the Mobike BSS in Beijing over the period between 10/May/2017 and 25/May/2017. Positive value means the amount of demand that is above the amount of bike returns.



In this paper, we are interested in the simple case of operator-based static rebalancing scenario, so inter-day behavior of the distribution is more meaningful. We decrease the time window to a shorter period to study the gaps between rents and returns on a daily basis shown in Figure 3. A similar pattern can be easily spotted. This allows us to form some basic intuitions on solving the rebalancing problem.

There may exist infinite ways to rebalance the BBS repository state, each with different cost and return. The most important point that defines a "good rebalancing plan" is the balance of the users' satisfactory and the cost, such that the users can almost always find an available bike nearby when necessary, with an acceptable amount of cost of the transportation work for moving the bikes.

EXISTING METHODS

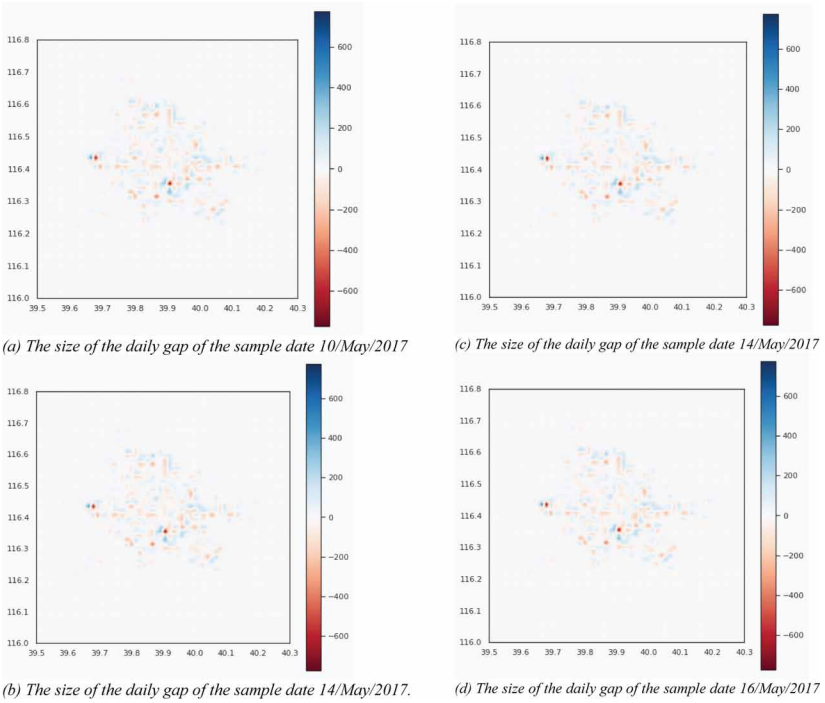
Optimal Transportation Theory

Suppose we have good knowledge of the distribution of the bike rents and returns. Then, the most natural way to formulate the problem is to find a transportation plan that can pushes the return distribution to the rental distribution with the minimal cost, which is exactly the research goal of the classic optimal transportation. The optimal transportation (OT) theory, firstly formalized by the Gaspard Monge (Villani, 2008) to solve a class of planning problems, aims to find the optimal way to move a pile of earth to a pit by minimum effort. The solution to this problem is associated with a large class of problems within or outside the transportation domain.

Monge formulates this problem as, given two probability measures μ_S on Ω_S and μ_T on Ω_T, find a transport $T : \Omega_S \to \Omega_T$ to minimize the overall moving cost:

$$\inf\left\{\int c\big(x,T(x)\big)d\mu_S(x), \quad T\#\mu_S = \mu_T\right\} \tag{1}$$

Figure 3. The size of the daily gap between the rent request and returns of the Mobike BSS in Beijing. Positive value means the amount of demand that is above the amount of bike returns.



(a) The size of the daily gap of the sample date 10/May/2017

(c) The size of the daily gap of the sample date 14/May/2017

(b) The size of the daily gap of the sample date 14/May/2017.

(d) The size of the daily gap of the sample date 16/May/2017

where $\Omega_S \in \mathbb{R}^{d_S}$ and $\Omega_T \in \mathbb{R}^{d_T}$, and $c : \Omega_S \times \Omega_T \rightarrow [0, \infty]$ is a cost function. One major flaw of the Monge's formulation is its non-convexity and intractability. Moreover, in Monge's formulation, the existence of the map is not always guaranteed (e.g., when μ_S is a Dirac and μ_T is not, or when μ_S and μ_T are supported on a different number of Diracs). Later, the Kantorovitch's version of the OT problem fixes these flaws by using a formulation with convex relaxation, and can be expressed as finding the coupling between Ω_S and Ω_T:

$$\gamma_0 = \arg\min_{\gamma \in \Pi} \int_{\Omega_S \times \Omega_T} c(x^s, x^t) d\gamma(x^s, x^t) \tag{2}$$

where π is the set of all probabilistic couplings in $P(\Omega_S \times \Omega_T)$, the space of all joint distributions with marginal μ_S and μ_T.

The study on the optimal transportation came from the physical world, but its abstract forms of the problem modeling and its neat solution exerts much more impact outside the transportation area (Courty, Flamary, Tuia & Rakotomamonjy, 2015). On the other hand, applying the Kantorovitch's method directly to the rebalancing task may encounter several real world difficulties. For instance, the road network does not form a metric space.

PDP-Based Approaches

A large portion of the current researches on the BSS rebalancing problem model it as a type of a pickup and delivery problem (PDP) (Benchimol, Benchimol, Benoit et al., 2011; Raviv & Kolka,

2013; Forma, Raviv & Tzur, 2015). A general form of the pickup and delivery problem (also known as the General Pickup and Delivery Problem, or GPDP) is a typical resource allocation problem, in which the system is required to allocating a set of vehicles, construct a route for each, in order to meet a number transportation requests, with certain constrains and optimization objectives. In the scenario of transportation system, *general pickup and delivery problem* (GPDP) (Savelsbergh & Sol, 1995) covers a wide range of real-world problems that related to the schedule optimization. There are several sub-types of the GPDP, and one simple classification scheme proposed by (Savelsbergh & Sol, 1995). Besides, Parragh, Doerner & Hartl (2008) proposes a finer classification scheme, which involves a hierarchy of four levels.

Under this framework, Chemla, Meunier & Calvo (2013) propose a branch-and-cut algorithm for static rebalancing, which finds an upper bound of the optimal solution via a tabular search. Raviv & Kolka (2013) tackle the problem by predicting the optimal inventory level at the beginning of the day, with respect to the minimal day-time dissatisfaction, and propose a solution based on the continuous time Markov chain. On the dynamic rebalancing side, (Benchimol, Benchimol, Chappert et al., 2011) formulate the problem by an arc-flow formulation on a space-time network, and compare it to the 1-PDP, then uses Dantzig-Wolfe decomposition and Benders decomposition to obtain a tractable lower bounds. Pfrommer et al. (Pfrommer, Warrington, Schildbach & Morari, 2014) use model-based receding horizon optimization techniques to combine operators relocation operations and user-based relocations. In fact, a number of researches on operation strategy notice that the predictive analysis can make important contributions to the rebalancing problem (Barth & Todd, 1999). Caggiani and Ottomanelli (2012) use a neural network to forecast the arrivals and departures of the bikes to assist the decision support system for the bike-sharing rebalancing task. Regue and Recker (2014) introduce the idea of *proactive* vehicle routing, and propose a framework to tackle the dynamic bike rebalancing problem in BSS, which contains a demand forecasting model as one of its four components.

AN RL'S PERSPECTIVE

We propose a reinforcement learning approach that can solve the bike-rebalancing problem by learning through experience, without prior knowledge of the rent or return distribution. This approach features in:

1. The policy function directly maps the observed state to the action to be taken, without any online fitting process;
2. Due to the non-convexity nature of deep neural network, and the discrete space optimization that RL allows, a much wider range of forms of objective function is available;
3. The learning process does not require the prior knowledge or explicit prediction of the destination distribution of the bikes.

This paper simplifies the scenario and only considers the rebalancing among the central hubs, which has also been adopted by other previous researches (Singhvi, Singhvi, Frazier et al., 2015). The distances between the locations of any pair of hubs are encoded in a distance matrix $\mathbf{D}$, in which $\mathbf{D}_{i,j}$ represents the logic distance between hub point i and j.

Policy Gradient for Continuous Action

We model the rebalancing problem as a RL for continuous action policy problem (Lillicrap, Hunt, Pritzel et al., 2016). Since it is intractable to find the argmax of the Q-function in continuous action space, policy gradient method (Sutton, McAllester, Singh, & Mansour, 2000) is usually used. Suppose the policy function π family is parameterized by θ, the goal of the policy gradient is to update θ

in the direction that will improve the performance (i.e., $\nabla_\theta J(\pi_\theta)$), which usually follows the *policy gradient theorem* (Sutton, 1999) for stochastic policy:

$$\begin{aligned}\nabla_\theta J(\pi_\theta) &= \int_S \rho^\pi(s) \int_A \nabla_\theta \pi_\theta(a \mid s) Q^\pi(s,a) da ds \\ &= \mathbb{E}_{s\sim\rho^\pi, a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a \mid s) Q^\pi(s,a)\right]\end{aligned} \tag{3}$$

where $\rho^\pi(s)$ is the state distribution dependent on the policy π. Then the difficult part is approximating the Q function, which can be estimated from the sample return of r_t^γ, in a way similar to the REINFORCEMENT algorithm (Williams, 1992).

In this research, we use a deterministic version of the continuous policy gradient method, in which the $\mu_\theta : S \to A$. The performance objective J_θ thus does not require the integration of the action space (Silver, Lever, Heess et al., 2014):

$$\begin{aligned}J(\mu_\theta) = &\int_S \rho^\mu(s) r\left(s, \mu_\theta(s)\right) ds \\ &= \mathbb{E}_{s\sim\rho^\mu}\left[r(s, \mu_\theta(s))\right]\end{aligned} \tag{4}$$

and the gradient can be calculated by the chain-rule:

$$\begin{aligned}\nabla_\theta J(\mu_\theta) &= \int_S \rho^\mu(s) \nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s,a)\Big|_{a=\mu_\theta(s)} ds \\ &= \mathbb{E}_{s\sim\rho^\mu}\left[\nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s,a)\Big|_{a=\mu_\theta(s)}\right]\end{aligned} \tag{5}$$

Objective Function

An action $\mathbf{A} \in \mathcal{A} \subseteq \mathbb{R}^{k\times k}$ as an instance of the rebalancing plan, represented by a $k \times k$ matrix, where $\mathbf{A}_{i,j}$ is the amount of bikes to be repositioned from *i* to *j*. The virtual environment uses two *k*-dimensional real-valued vectors **u** to represent the number newly received rent requests, and **v** the number of bikes after all day's service at hub. At each time-step, when the agent is to generate a plan $\pi(s) = \mathbf{A}$, it can observe the end-of-day repository state **v** (i.e., $s^{(t)} = \mathbf{v}^{(t)}$ for any of the day *t*), but is ignorant of the incoming rent request amount **u** due to the model-free assumption.

Intuitively, the reward at each time-step of the policy is made up of two components:

$$r_\theta = r_{\text{income}} - cost_{\text{reposition}} \tag{6}$$

Here, r_{income} is the amount of income that users pay to rent the bikes, and $cost_{\text{reposition}}$ is the reposition cost, approximated by:

$$w_{\text{reposition}} \times \sum_{i,j} \left\{\mathbf{A}_{i,j} \times \mathbf{D}_{i,j}\right\} \tag{7}$$

But to give a stronger signal of the potential loss of the income, we replace this income term r_{income} with a loss value l_{miss}, proportional to the gap between the supply and demand:

$$\ell_{\text{miss}} = w_{\text{miss}} \times \max(\mathbf{u} - \mathbf{z}, 0) \tag{8}$$

where w_{miss} is the rate of the potential cost if a rent request cannot be satisfied, max is the element-wise clamp function. Here, $\mathbf{z}_i^{(t+1)} = \mathbf{v}_i^{(t)} + \sum_j \mathbf{A}_{i,j}^{(t)}$ is the supply amount of hub *i* at the beginning of the day $t + 1$, made up of two parts: the end-of-day amount of hub *i* at day *t*, and the sum of reposition amount from all other hubs at day *t*. This is similar to the idea of UDF (User Dissatisfaction Function) proposed by (Raviv & Kolka, 2013), and encourages us to reform the objective as a loss function, which is positive number to be minimized:

$$\ell_\theta = \ell_{\text{miss}} + cost_{\text{reposition}} \tag{9}$$

Regularization

There is one constrain that the policy must satisfy: the sum of the moving amount from the source hub *i* should add up the current repository of that hub. That is:

$$\sum_{j=1}^{k} \mathbf{A}_{i,j} = v_i \tag{10}$$

To address this constrain, we regard the policy for any hub *i* in a problem with k hubs in total to have *k* - 1 degree-of-freedom, such that our policy network would only product only *k* - 1 values for each hub, and the *k*-th value is implied by:

$$\mathbf{A}_{i,k} = v_i - \sum_{j=1}^{k-1} \mathbf{A}_{i,j} \tag{11}$$

If we regard the possible negative value as the "back-flow", and change the rebalancing plan to:

$$w_{\text{reposition}} \times \sum_{i,j} \left\{ \left| \mathbf{A}_{i,j} \right| \times D_{i,j} \right\} \tag{12}$$

such that the back-flow reposition amount will always bring extra costs. Thus, the agent can atomically learn to avoid such "back-flowing" behavior. To accelerate the learning, we add an extra high-penalty to such back-flow behavior:

$$\Omega_{\text{back-flow}} = w_{\text{back-flow}} \times \sum_{i,j} \min\left(\mathbf{A}_{i,j}, 0\right) \tag{13}$$

where $w_{\text{back-flow}}$ is a large number (e.g., 10^8). Using an extra regularization term with high penalty weight is commonly seen in the implementation of the support vector machine (SVM), which transforms a hard constraint into a soft version, and also provides the convenience for optimization.

Ω_{idle} is another regularization term that discourages the systems to have idle bikes stay in the docker. This regularization term may reflect the extra cost that the operation company would pay to the municipal authority:

$$\Omega_{\text{idle}} = w_{\text{idle}} \times \min(\mathbf{u} - \mathbf{z}, 0) \tag{14}$$

This would allow us to estimate the optimal delivery volume of the entire system, which is set constant for now. Add the two regularization terms into the previous loss function gives the following objective:

$$\ell_\theta = cost_{\text{reposition}} + \ell_{\text{miss}} + \Omega_{\text{back-flow}} + \Omega_{\text{idle}} \tag{15}$$

Empirical Results

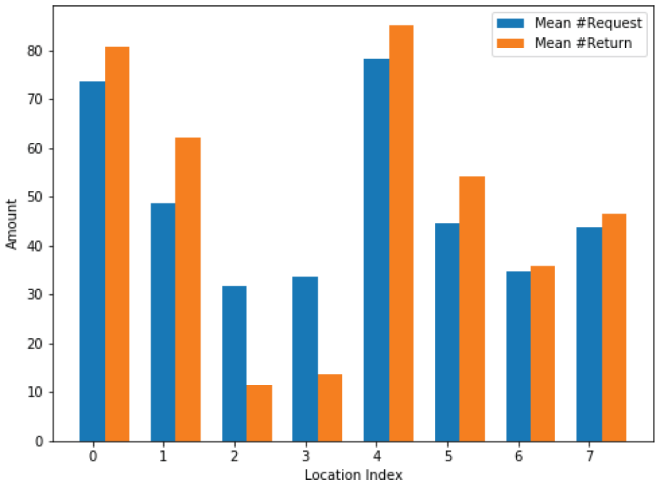
We setup a virtual environment in which there are $k = 8$ central hubs, whose pair-wise distance matrix is randomly initialized. Each of hubs has its own character of the rent and return behaviors. The choice of $\mathbf{u}$ and $\mathbf{v}$ should obey that the total amount of bikes should roughly remains constant, and we further assume that the total supply of the bikes is roughly the total demand of the bikes, such that $\mathbb{E}[|| \mathbf{v} ||_1] = \mathbb{E}[|| \mathbf{u} ||_1]$. This is challenging setting, since any mis-alignment of the supply would result to heavy penalty. We randomly generate $\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$ as shown in Figure 4.

The actual daily rent and return patterns follow a normal distribution with the variance if $\Sigma_{\mathbf{u}}$ and $\Sigma_{\mathbf{v}}$ respectively:

$$\begin{aligned} p(\mathbf{u}) &\sim \mathcal{N}(\bar{\mathbf{u}}, \Sigma_{\mathbf{u}}) \\ p(\mathbf{v}) &\sim \mathcal{N}(\bar{\mathbf{v}}, \Sigma_{\mathbf{v}}) \end{aligned} \tag{16}$$

We set the variances of the distributions are both isomorphic $\Sigma_{\mathbf{u}} = \Sigma_{\mathbf{v}} = 5 \cdot \mathbf{I}$, and the hyper-parameters $w_{\text{reposition}} = 0.2$, $w_{\text{miss}} = 10$, and $w_{\text{idle}} = 0.1$. We follow the deterministic policy gradient

Figure 4. The histogram of the distribution mean of the number of rent requests and the number of bike returns

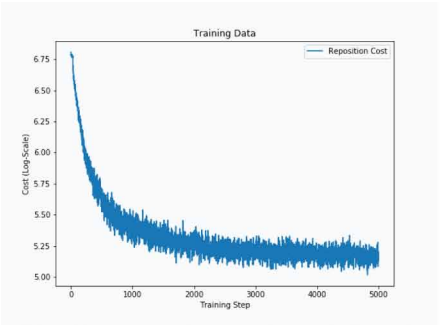


method, and use the batch expected return as the estimation of the state-action value. Figure 5(a) and 5(b) depict the learning progress of the policy in terms of two important part of the loss function: the reposition cost and the missing request loss. These two terms are actually competing with each other, but both decrease steadily along with the learning process, and finally reach a dedicated balance. Figure 5(c) and 5(d) depict the same loss values of the policy learned, when applied on a set of unseen test dataset. They follow similar patterns as those in Figure 5(a) and 5(b), indicating that the policy function can generalize well on the unseen new scenarios, can directly generate a good action decision without the need to make on-line fitting. We also notice that the $cost_{\text{reposition}}$ has a gentler learning curve than $loss_{\text{miss}}$. This is arguably because it takes more effort to learn to let most of repository remains unmoved, than to learn to respond to the rent distribution for a deep neural network.

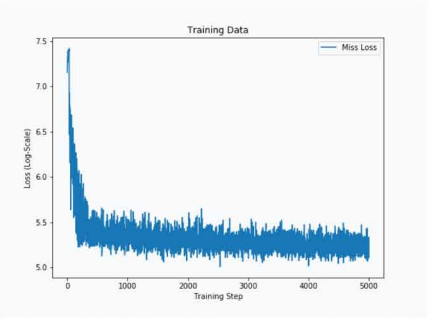
The overall loss of the system is shown in Figure 6. We use a horizontal line in Figure 6(a) to indicate the value of $loss_{\text{miss}}$ of the baseline *lazy policy* (without any reposition cost). This value is relatively low compared with the early iterations of the learnt policy, since the improper rebalancing work would cause considerable cost, which may far above the loss due to the missed requests and the idle cost. As learning progresses, the agent begins to find the sub-optimal policy to reposition the bikes for each of the hubs. The fluctuation of loss value reflects the unpredictability of the dynamic process, possibly because the total amount of request is tightly equal to the total amount of supplies, such as any mis-alignment would cause high penalty of $loss_{\text{miss}}$. At each of the iterations, the policy is tested on a separate set of data. We compare the loss values between the training data and the test data in Figure 6(b), which shows no degeneration of the performance caused by overfitting of the learned policy.

There are several points this approach can improve. First, in our simulations, it obtains a large amount of experiences; while in real-world scenarios, the agent has to be able to learn from a very limited amount of experience, leading to extra challenge on the ability of generalization, especially when the distribution is not stationary. Second, the amount of the bikes to transport is modeled as

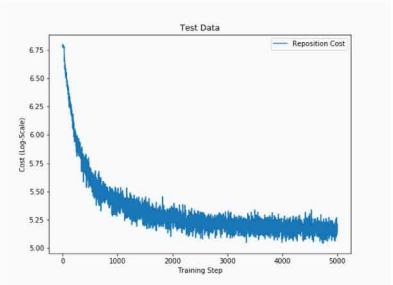
Figure 5. The value of two important parts of the loss functions on the training data and test data



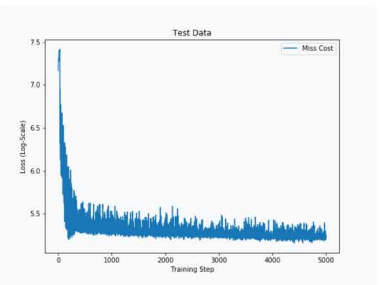
(a) The log-scaled reposition cost at each training step, when measured on the training data.



(b) The log-scaled loss of the missing requests at each training step, when measured on the training data.



(c) The log-scaled reposition cost at each training step, when measured on the test data.



(d) The log-scaled loss of the missing requests at each training step, when measured on the test data.

Figure 6. The curve of the loss function on training data and test data

(a) The curve of the loss function on the training data. The horizontal line is for the baseline policy.

(b) A comparison between the value of the loss function on training data and test data.

continuous action, whereas the real-world scenarios may require discrete action space. Third, this example neither considers the vehicles' backhaul trips, nor the multi-vehicle settings, which would require additional parameters and items in both the action space and objective functions. And lastly, in our simulation, the only variable to describe the state is the repository state, while leaving all other factors to the perturbations as random. In real-world scenarios, features like weekend, weather can make important contributions to the final policy.

CONCLUSION

To enable a better smart transportation system, the genuine "smart" behaviors of the system, featured by the machine learning algorithms built on top of the IoT infrastructures should be our next focus of research. This paper serves to demonstrate the considerable power of the DRL in policy optimization, through a case study on the rebalancing problem in a bike sharing system. The optimization goals of the policy functions in these problems are often subject to a combination of several criteria, which

can be modeled well by DRL framework. In addition, DRL allows the system to generate sensible policies with the minimum amount of prior knowledge over the world environment, and have a much stronger capability to generalize from the training data, when compared with the previous approaches. These nice properties encourage us to explore the potentials of DRL in smart transportation systems more extensively in future works.

ACKNOWLEDGMENT

This research was supported by National social science fund project [grant numbers 16BXW031]; Shanghai college and university young teacher training subsidy program [grant number ZZslg16054].

REFERENCES

Albino, V., Berardi, U., & Dangelico, R. M. (2015). Smart cities: Definitions, dimensions, performance, and initiatives. *Journal of Urban Technology*, *22*(1), 3–21.

Arel, I., Liu, C., Urbanik, T., & Kohls, A. G. (2010). Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems*, *4*(2), 128–135.

Barth, M., & Todd, M. (1999). Simulation model performance analysis of a multiple station shared vehicle system. *Transportation Research Part C, Emerging Technologies*, *7*(4), 237–259.

Benchimol, M., Benchimol, P., Chappert, B., De La Taille, A., Laroche, F., Meunier, F., & Robinet, L. (2011). Balancing the stations of a self service "bike hire" system. *Operations Research*, *45*(1), 37–61.

Caggiani, L., & Ottomanelli, M. (2012). A modular soft computing based method for vehicles repositioning in bike-sharing systems. *Procedia: Social and Behavioral Sciences*, *54*, 675–684.

Chemla, D., Meunier, F., & Calvo, R. W. (2013). Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization*, *10*(2), 120–146.

Chourabi, H., Nam, T., Walker, S., Gil-Garcia, J. R., Mellouli, S., Nahon, K., . . . Scholl, H. J. (2012, January). Understanding smart cities: An integrative framework. In *System Science (HICSS), 2012 45th Hawaii International Conference on* (pp. 2289-2297). IEEE.

Contardo, C., Morency, C., & Rousseau, L. M. (2012). *Balancing a dynamic public bike-sharing system* (Vol. 4). Cirrelt.

Courty, N., Flamary, R., Tuia, D., & Rakotomamonjy, A. (2017). Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(9), 1853–1865.

DeMaio, P. (2009). Bike-sharing: History, impacts, models of provision, and future. *Journal of Public Transportation*, *12*(4), 3.

Forma, I. A., Raviv, T., & Tzur, M. (2015). A 3-step math heuristic for the static repositioning problem in bike-sharing systems. *Transportation Research Part B: Methodological*, *71*, 230–247.

Gong, W., Qi, L., & Xu, Y. (2018). Privacy-Aware Multidimensional Mobile Service Quality Prediction and Recommendation in Distributed Fog Environment. *Wireless Communications and Mobile Computing*.

Hashem, I. A. T., Chang, V., Anuar, N. B., Adewole, K., Yaqoob, I., Gani, A., & Chiroma, H. et al. (2016). The role of big data in smart city. *International Journal of Information Management*, *36*(5), 748–758.

Hu, C., Li, R., Mei, B., Li, W., Alrawais, A., & Bie, R. (2018). Privacy-preserving combinatorial auction without an auctioneer. *EURASIP Journal on Wireless Communications and Networking*, *2018*(1), 38.

Hu, C., Li, W., Cheng, X., Yu, J., Wang, S., & Bie, R. (2018). A secure and verifiable access control scheme for big data storage in clouds. *IEEE Transactions on Big Data, 4*(3), 341-355.

Khamis, M. A., & Gomaa, W. (2014). Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. *Engineering Applications of Artificial Intelligence*, *29*, 134–151.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., . . . Wierstra, D. (2015). *Continuous control with deep reinforcement learning.* arXiv preprint arXiv:1509.02971.

Lin, J. R., & Yang, T. H. (2011). Strategic design of public bicycle sharing systems with service level constraints. *Transportation Research Part E, Logistics and Transportation Review*, *47*(2), 284–294.

Lippi, M., Bertini, M., & Frasconi, P. (2013). Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, *14*(2), 871–882.

Liu, B. S., Li, Y. J., Yang, H. T., Sui, X. S., & Niu, D. F. (2006, October). Research on forecasting model in short term traffic flow based on data mining technology. In Null (pp. 707-712). IEEE.

O'Brien, O., Cheshire, J., & Batty, M. (2014). Mining bicycle sharing data for generating insights into sustainable transport systems. *Journal of Transport Geography*, *34*, 262–273.

Pan, Z., Lei, J., Zhang, Y., & Wang, F. L. (2018). Adaptive fractional-pixel motion estimation skipped algorithm for efficient HEVC motion estimation. *ACM Transactions on Multimedia Computing Communications and Applications*, *14*(1), 12.

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008). A survey on pickup and delivery problems. *Journal für Betriebswirtschaft*, *58*(1), 21–51.

Peng, K., Leung, V., Zheng, L., Wang, S., Huang, C., & Lin, T. (2018). Intrusion Detection System Based on Decision Tree over Big Data in Fog Environment. *Wireless Communications and Mobile Computing*.

Peng, K., Leung, V. C., & Huang, Q. (2018). Clustering Approach Based on Mini Batch Kmeans for Intrusion Detection System over Big Data. *IEEE Access: Practical Innovations, Open Solutions*.

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

Qi, L., Zhang, X., Dou, W., & Ni, Q. (2017). A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data. *IEEE Journal on Selected Areas in Communications*, *35*(11), 2616–2624.

Raviv, T., & Kolka, O. (2013). Optimal inventory management of a bike-sharing station. *IIE Transactions*, *45*(10), 1077–1093.

Regue, R., & Recker, W. (2014). Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem. *Transportation Research Part E, Logistics and Transportation Review*, *72*, 192–209.

Savelsbergh, M. W., & Sol, M. (1995). The general pickup and delivery problem. *Transportation Science*, *29*(1), 17–29.

Shaheen, S., Guzman, S., & Zhang, H. (2010). Bikesharing in Europe, the Americas, and Asia: Past, present, and future. *Transportation Research Record: Journal of the Transportation Research Board*, (2143), 159–167.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014, June). Deterministic policy gradient algorithms. ICML.

Singhvi, D., Singhvi, S., Frazier, P. I., Henderson, S. G., O'Mahony, E., Shmoys, D. B., & Woodard, D. B. (2015, January). Predicting Bike Usage for New York City's Bike Sharing System. *AAAI Workshop: Computational Sustainability*.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.

Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In Advances in neural information processing systems (pp. 1057-1063). Academic Press.

Villani, C. (2008). *Optimal transport: old and new* (Vol. 338). Springer Science & Business Media.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, *8*(3-4), 229–256.

Xu, Y., Qi, L., Dou, W., & Yu, J. (2017). Privacy-Preserving and Scalable Service Recommendation Based on SimHash in a Distributed Cloud Environment. *Complexity*.

Yan, C., Cui, X., Qi, L., Xu, X., & Zhang, X. (2018). Privacy-aware data publishing and integration for collaborative service recommendation. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 43021–43028.

Zhang, J., Wang, F. Y., Wang, K., Lin, W. H., Xu, X., & Chen, C. (2011). Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, *12*(4), 1624–1639.

Zhang, Y., & Li, S. (2017). Distributed biased min-consensus with applications to shortest path planning. *IEEE Transactions on Automatic Control*, *62*(10), 5429–5436.

Zhang, Y., Thomas, T., Brussel, M. J. G., & van Maarseveen, M. F. A. M. (2016). Expanding bicycle-sharing systems: Lessons learnt from an analysis of usage. *PLoS One*, *11*(12), e0168604.

Zolfpour-Arokhlo, M., Selamat, A., Hashim, S. Z. M., & Afkhami, H. (2014). Modeling of route planning system based on Q value-based dynamic programming with multi-agent reinforcement learning algorithms. *Engineering Applications of Artificial Intelligence*, *29*, 163–177.

Guofu Li is a machine learning scientist in Ping An Asset Management. He received his BSc in Software Engineering from the Fudan University in 2007, and his PhD in Computer Science and Informatics from the University College Dublin in 2014. He was a Post-doctoral Research Fellow at the CNGL research centre (ADAPT now) and Computer Science and Informatics of the University College Dublin from 2014 to 2015. Then he worked as a Lecturer in College of Communication and Art Design at the University of Shanghai for Science and Technology. His research interests include machine learning and natural language processing.

Pengjia Zhu is focused on natural language processing.

Yanwu Zhang is a teacher for the Department of Information Engineering at Qingdao Binhai University.

Lei Li is a teacher for the Department of Information Engineering at Qingdao Binhai University.

Qingyuan Li says "happiness is the absence of striving for happiness."

Yu Zhang is affiliated with Qingdao Binhai University in the Computer Science and Technology Department.