A Learning-based Bike Rebalancing Scheme with Adaptive User Incentive

Yubin Duan and Jie Wu

Abstract Nowadays, major cities have deployed the Bike Sharing Systems (BSSs), attracted by their environmental and economic benefits. The asymmetric bike demand in temporal and spatial domains causes an imbalance in bike distribution which further leads to overflow and underflow events, for both docked and dock-less BSSs. The overflow and underflow events may lead to congestion in the city and a lower profit to BSSs, respectively. While challenging, it is necessary to efficiently rebalance the BSS in a timely and cost-efficient manner. We investigate the rebalancing problem in this paper, and propose to provide both source and/or destination incentives to users to let them rent and/or return bikes at alternative locations. Our objective is to maximize the number of bike usages during a day and maintain the summation of incentives under a given budget. The complex user dynamics in both spatial and temporal domains make it challenging to adaptively determine the prices for both source and destination incentives. We use reinforcement learning techniques to find patterns of user dynamics and to determine the incentive prices. Specifically, we adapt the state-of-the-art reinforcement learning framework for dock-less BSS rebalancing. Different from existing research, we make full use of the benefits of destination incentives. In addition, we further extend the reinforcement learning framework to docked BSSs by adding station capacities to the state space of the reinforcement learning agent. We examine the performance of our schemes based on real-world datasets. Our experiment results reveal that the hybrid incentive scheme outperforms the source-incentive-only scheme.

Yubin Duan
Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA 19122 e-mail: `yubin.duan@temple.edu`

Jie Wu
Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA 19122 e-mail: `jiewu@temple.edu`

1 Introduction

The Bike Sharing Systems (BSSs) have been deployed in major cities across the world. The environmental and economic benefits brought by the BSS speed up the deployment of the BSS [10]. The easy accessibility of BSSs could motivate users to ride bikes for short distance travel and drive less. About 40% of BSS users drive less after joining the BSS in 4 American cities. Riding bikes is more environmental friendly and helps to reduce the CO_2 emission. The BSS also brings economic benefits to the public. Deploying a BSS in a local neighborhood could increase accessibility with local businesses. In addition, BSSs address the "last mile-first mail" issues in cities by decreasing journey times and increasing users' mobility.

However, efficiently rebalancing the BSS becomes necessary and challenging with the expansion of the BSS. Without bike rebalancing, the asymmetric users' demands for bikes in temporal and spatial domains would cause *overflow* and *underflow* events as shown in Fig. 1. In docked BSSs, such as the *Citi bike* system in NYC, the overflow events occur at stations that are full of bikes. Users cannot return bikes to those overflow stations, which increases the detour distances of users. The underflow events occur at stations that have no bikes. Users cannot rent bikes at those underflow stations, and the BSS loses the potential profit. Although there are no concepts of station capacities for dock-less BSSs, such as the *Mobike* in China, too many bikes clustered in a small region would also cause overflow events, since it would cause congestion in the city. The underflow events where there are no bikes in an area also make the BSS operators lose users. Those negative impacts caused by bike overflow and underflow events motivate the BSS operators to rebalance their systems in a timely and cost-efficient manner.

In this paper, we investigate the bike rebalancing problem. We follow the user incentive approach to rebalance the BSSs. Specifically, the BSS operator would provide a monetary incentive for users if they rent/return bikes at locations specified by the operator. We consider both *source incentive* and *destination incentive* for renting and returning bikes, respectively. Compared with truck-based rebalancing approaches, the user-based approach is more flexible and has been implemented in real-world BSSs, such as the *Bike Angle project*[1] in NYC. Our objective is to maximize the number of bike usages during a day. Maximizing the number of bike usages is critical since it could benefit both system operators and users. A larger number of bike usages means a better service level of the system. It could satisfy more user demands for bikes and enlarge the profit of BSS operators. Our constraint is the budget constraint which means the summation of incentives provided to users in a day is limited by a constant value. The budget constraint is essential for BSS operators since they need to make profits for long-term operation.

The problem we investigated is different from existing research. The state-of-the-art user-based bike rebalancing scheme [25] only considers the source incentives but ignores the power of destination incentives. Specifically, their scheme only considers encouraging users to rent bikes from nearby regions with a source incentive. We

[1] https://www.citibikenyc.com/bikeangels/

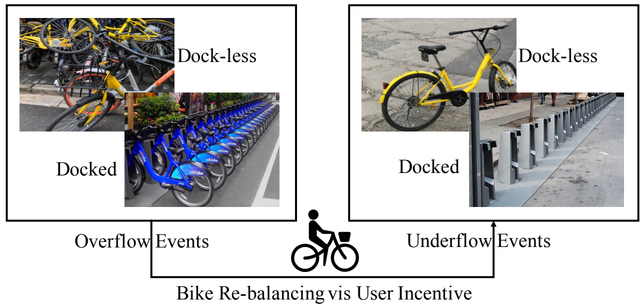


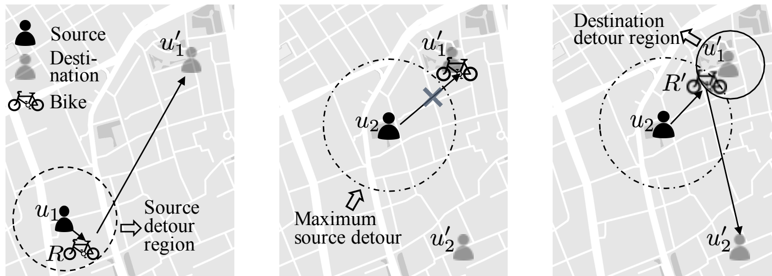
Fig. 1 Resolving underflow/overflow through bike rebalancing.

notice that destination incentives that let users return bikes at alternative locations can also help to rebalance the system. Adaptively combining those two incentives could bring extra benefits for bike balancing. Besides, we extend the problem scenario of [11] which only considers the dock-less BSSs. Both dock-less and docked rebalancing problems are considered in this paper.

The benefits of the destination incentive are shown by the example in Fig. 2. In a dock-less BSS, there are two users u_1 and u_2 and only one available bike located at R on the map. We assume u_2 arrive at the system right after u_1 reaches its destination u_1'. Firstly, we only consider source incentives. If a user cannot find bikes nearby his/her source location, the BSS would incentivize the user to enlarge its source detour region and pick-up a bike, as shown in Fig. 2(a). Note that users have maximum detour distances [27] since their mobility is limited by walking. Therefore, as illustrated in Fig. 2(b), user u_2 cannot rent the bike, and the BSS operator loses the user. However, we notice that the detour distance of user u_1 is relatively small and does not exceed the detour distance limitation. u_1 could have another detour near its destination if an incentive is provided. If we also consider destination incentive, the user u_1 could return the bike at location R' instead of its destination, as shown in Fig. 2(c). Then, the user u_2 could successfully rent the bikes with source incentives. The example shows that the number of bike usage or the service level is increased from 1 to 2 by providing destination incentives along with source incentives.

Inspired by the motivation example, we propose a user-incentive-based rebalancing scheme that considers both source and destination incentives. However, it is not trivial to design such a scheme. The first challenge is the complex user dynamics. [12] and [25] have shown the user dynamics in both temporal and spatial domains in docked and dock-less BSSs, respectively. For dock-less BSSs, another challenge is the extremely large number of bikes in a city. For example, Mobike plans to deploy hundreds of thousands of bikes in Guangzhou, China. Determining the incentive price for each bike is computationally complex. Another challenge for both dock-less and docked BSSs is to adaptively adjust the source and destination incentive prices for each time-slot in a day.

We propose to use reinforcement learning approaches to solve those challenges. Although the user dynamics are complex in both temporal and spatial domains,



(a) u_1 accepts the source incentive

(b) u_2 cannot rent the bike w/o dest. incentive

(c) Dest. detour of u_1 helps u_2

Fig. 2 An illustration of the benefit of destination incentive.

there exist usage patterns. Liu et. al. [23] have shown that the user demands could be predicted by using machine learning techniques. The reinforcement learning agent could learn the pattern from its exploring experience. Besides, the reinforcement learning agent could adaptively adjust its pricing policy according to the reward function. Reinforcement learning algorithms fit our problem scenario. In addition, we could divide the city into multiple regions and let bikes in the same region have identical incentive prices. The problem scale could be reduced, as well as the complexity of training a reinforcement learning agent. Therefore, in this paper, we extend the reinforcement learning framework proposed by [25] that only considers source incentives, and propose a hybrid incentive scheme that makes use of both source and destination incentives.

The contributions of our paper are summarized as follows:

- We propose to rebalance the BSS by offering users both source and destination incentives, which brings extra benefits compared with the source-incentive-only schemes.
- We analyze the advantage of the destination incentives, and propose to combine the source and destination incentives by splitting the rebalancing budget.
- We adapt the state-of-the-art reinforcement learning framework for rebalancing dock-less BSSs to determine the destination incentive prices.
- We further extend our scheme to docked BSSs by adding the capacities of each station into the state space of the reinforcement learning agent.
- We test the performance of our hybrid incentive schemes by experiments on a real-world dataset.

The remainder of the paper is organized as follows. Section 2 presents our problem statement which contains the notations and system models. Section 3 presents our hybrid incentive scheme which adaptively adjusts source and destination incentive strength. Section 4 discusses the extended scheme for docked BSS rebalancing. Section 5 illustrates the experiment that is conducted on a real-world dataset. Section 6 reviews the related works. Section 7 concludes the paper.

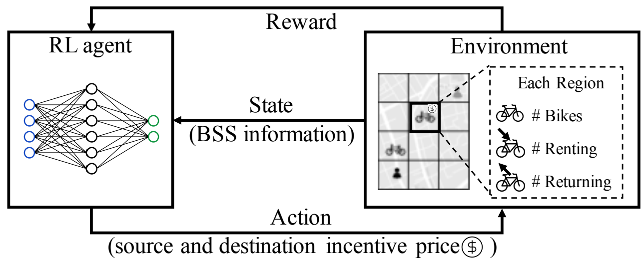


Fig. 3 An overview of the reinforcement learning framework.

2 Problem Statement

2.1 Overview

In our model, we propose an adaptive approach for rebalancing dock-less BSS. Given a limited budget, which is not sufficient enough to totally balance the BSS, our approach adaptively allocates it to incentive users to conduct a detour at source and/or destination based on the underflow/overflow distribution across time and space. The objective is to maximize the overall service level of the system over a day. The service level is quantified by the number of satisfied users or the number of bike usages.

Specifically, the incentive used to encourage users to rent bikes at neighbor regions of their sources is denoted as a source incentive, while the incentive is called a destination incentive on the other side. For a source incentive, the BSS operator provides locations of available bikes to each user along with incentive prices of bikes in neighbor regions. For a destination incentive, the operator suggests users return bikes to neighbor regions of the user's destination. The price of source and destination incentive is determined by the incentive scheme. A reinforcement learning based price scheme for source incentive has been studied in [25]. We propose to jointly consider source and destination incentives inspired by the benefits of destination incentive we observed. Users' choice of accepting incentives or not is simulated by the environment model. The performance of the rebalance is evaluated via the service level which equals the number of satisfied users or the number of bike usages. The reinforcement learning framework of our hybrid incentive scheme is shown in Fig. 3.

2.2 Incentive Scheme Model

In the incentive scheme, we discretize time and space into time-slots and square regions respectively. The BSS operator provides differential source and/or destination

Table 1 Table of Notations

Notations	Description
H	The region set and $H = \{h_i, \ldots, h_n\}$
T	The slotted time set and $T = \{t_i, \ldots, h_m\}$
U	The user set and $U = \{u_i, \ldots, u_o\}$
$\varphi_i(t)$	Number of bikes in h_i at the beginning of t
$D_i(t)$	Number of bike rented from h_i during timeslot t
$\Lambda_i(t)$	Number of bike returned to h_i during timeslot t
$N(h_i)$	Neighbor regions of h
$c_k(i, j, \delta)$	u_k's cost of moving from h_i to h_j with distance δ
$\tau_{ij}(t)$	Number of users in h_i who rent bike from h_j and return to h_j during time t
$p_i^+(t)/p_i^-(t)$	Source/destination incentive price of region h_i at t
B^+/B^-	Budget provided by the BSS operator for the source/destination incentive.

incentive price for each region at each time-slot. Specifically, each day is separated into m time-slots in the time domain, denoted by $T = \{t_1, t_2, \ldots, t_m\}$. In the spatial domain, a city H is divided into n square regions, i.e., $H = \{h_1, h_2, \ldots, h_n\}$. The neighbors of a region h_i are defined as the four regions that are directly adjacent to h_i, and the set of neighbor regions for h_i is denoted as $N(h_i)$. Users to the BSS system are denoted by $U = \{u_1, u_2, \ldots\}$. Although the actual user demands vary across time and space, the patterns on their demands in both temporal and spatial domain provide basis for our discretization. Our statistic on traces data from *Mobike* shows the existence of rush hour and demand hot spots. The number of users' rent events and return events at region h_i during time-slot t are modeled as random variables $D_i(t)$ and $\Lambda_i(t)$ respectively. The number of bikes in h_i at the beginning of time-slot t is denoted as $\varphi_i(t)$.

To deal with the imbalance of the BSS, we assume that the provider is willing to provide a budget B for user incentive, including a source incentive budget B^+ and a destination incentive budget B^-. Our incentive scheme helps BSS operators to decide the differential price of source incentive $p_i^+(t)$ and destination incentive $p_i^-(t)$ for each region h_i at each time-slot t. That is, if a user rents bikes at a neighborhood region h_i of his/her source region during time-slot t, he/she can obtain an incentive $p_i^+(t)$. Each neighbor region may contain more than one bike, and the bikes in the same region have the same incentive price. Similarly, a destination incentive $p_i^-(t)$ is given to users who return bikes to h_i that are adjacent to users' destination region during time-slot t. Different from the source incentive, we assume that each region only contains one potential return location which is the center of the region. This simplification is to reduce the complexity of the model.

2.3 Environment Model

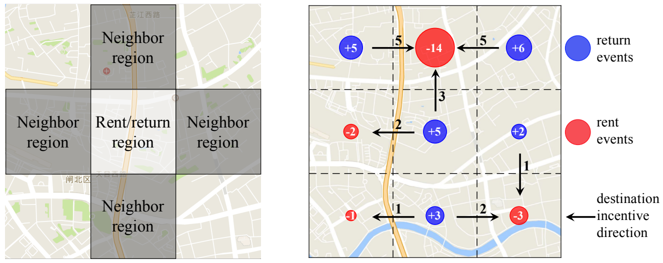
The environment mainly models user dynamics and provides feedbacks to the incentive scheme. Based on the source and destination incentive price vector generated by the scheme, the environment simulated each user choice of accepting the incentive or not. We assume each user has a cost if he/she goes to alternative locations to rent bikes (with source detours) or return bikes (with destination detours). We follow the user cost model in [25, 27]. Specifically, a user u_k has an initial cost C for either source or destination detour. Besides, the cost is also relevant to the detour distance δ. Specifically, let $c_k(h_i, h_j, \delta)$ and $c'_k(h_i, h_j, \delta')$ denote the source and destination detour cost respectively. h_i and h_j represent regions where u_k rents and returns a bike respectively. δ and δ' are the corresponding source and destination detour distance. If the user u_k rents (or returns) a bike at a region which is the neighbor of his/her source (or destination), his/her source detour cost $c_k(h_i, h_j, \delta) = C + \eta\delta^2$ (or destination detour cost $c'_k(h_i, h_j, \delta') = C + \eta\delta'^2$), where η is a constant coefficient. We assume users are not willing to rent or return bikes at regions further than neighbor regions, and the cost of renting or returning bikes in these regions is infinity. If the user u_k rents (or returns) bikes in the same region as his/her source (or destination), there is no cost. Note that if a user detours at both source and destination, he/she will receive both source and destination incentives, which helps to resolve overflow and underflow problem of the BSS, in one trip.

Users make decisions on whether to accept source and/or destination incentive before they start riding. A user first decides whether to accept the source incentive first and then makes decision on the destination incentive. If a user in h_i requests a bike during time-slot t, and he/she decides to rent a bike in h_j and to return it in h_k due to the incentives, then we increment $\tau_{ij}(t)$ by one to record this trace.

2.4 An Existing Pricing Scheme for Source Incentive

A pricing algorithm for source incentive is proposed by Pan et. al [25]. Their pricing scheme is based on a Markov Decision Process (MDP) and is optimized by using a reinforcement learning approach inspired by the hierarchical reinforcement learning [8, 9, 1] and Deep Deterministic Policy Gradient algorithm [20]. The pricing algorithm is briefly stated in the section and our adaptive incentive scheme is built upon it.

The MDP is used to model the interaction between the pricing scheme and the environment. Specifically, the MDP is a 5-tuple (S, A, P, r, γ), where S is the set of states $\{s_t\}$, A is the set of actions $\{a_t\}$, P describes the transition possibility between states under an action, r denotes the immediate reward and γ is the discount factor. The weights of future rewards and the present reward is determined by the discount factor $\gamma \in [0, 1]$. $\gamma = 1$ represents that the future rewards share the same importance as the present reward, i.e. the overall reward is the additive sum of the reward from each time-slot. The pricing scheme takes source incentive prices given to all regions as an action, and the number of satisfied users as a reward. The MDP ends when the budget B is used up. The pricing scheme finds a policy π_θ, which maps states to



(a) Neighbor regions (b) Rebalancing BSS via dest. incentive

Fig. 4 An illustration of destination incentive.

actions, through optimizing the MDP based on reinforcement learning. The number of bikes rented from h_i and returned to h_j during time slot t is denoted by $\tau_{ij}(t)$

Formally, their pricing scheme is defined as

$$\begin{aligned} \max \quad & \textstyle\sum_{t=1}^{m} \sum_{i,j=1}^{n} \tau_{ij}(t) && (1)\\ \text{s.t.} \quad & \textstyle\sum_{t=1}^{m} \sum_{i=1}^{n} p_i^{+}(t) < B && (2)\\ & \textstyle\sum_{j=1}^{n} \tau_{ji}(t) - \sum_{j=1}^{n} \tau_{ji}(t) \le \varphi_i(t), \forall i, t && (3)\\ & \textstyle\varphi_i(t+1) = \varphi_i(t) + \sum_{j=1}^{n} (\tau_{ji}(t) - \tau_{ji}(t)) \forall i, t. && (4) \end{aligned}$$

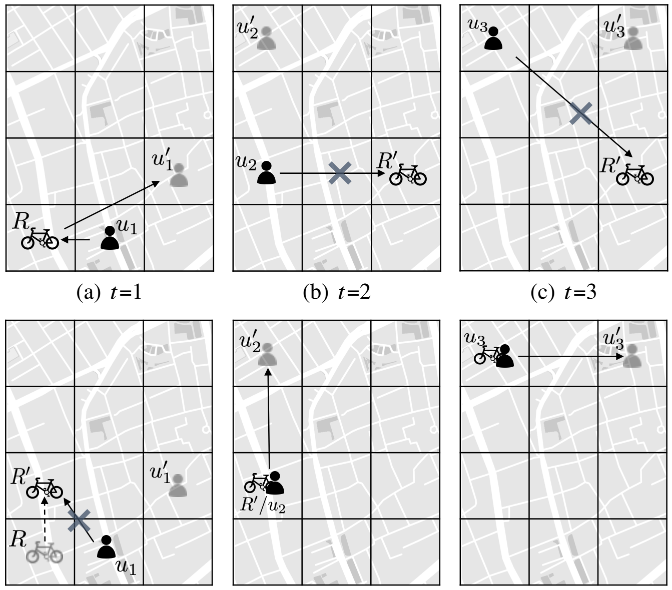
The objective is to maximize the number of valid traces over all regions and time-slots. The constraint (2) is the incentive budget limitation. The constraint (3) means that the number of bikes in each region should not be less than zero at any region during any time-slot. Eq. (4) represents the evolution of the number of bikes in each slot among different time-slots.

The MDP is optimized by applying reinforcement learning algorithms. The reinforcement learning aims to train a parameter set θ in π_θ such that the overall rewards can be maximized by following a policy π_θ. Formally, the overall reward brought by the policy π_θ is:

$$J_{\pi_\theta} = E[\sum_{t=0}^{\infty} \gamma^k r(a_t, s_t) | \pi_\theta, s_0].$$

2.5 Problem Formulation

Based on the system and environment models, the BSS Rebalancing problem is proposed. We aim to maximize the service level of a BSS in a one-day service circle. In each service circle, the BSS operator provides budget B^+ and B^- for source and destination incentives. Formally, our problem can be expressed as:



(a) t=1 (b) t=2 (c) t=3

(d) t=1' (e) t=2' (f) t=3'

Fig. 5 An example showing the benefit of returning at alternative destinations.

$$\max \quad \sum_{t=1}^{m} \sum_{i,j=1}^{n} \tau_{ij}(t) \tag{5}$$

$$\text{s.t.} \quad \sum_{t=1}^{m} \sum_{i=1}^{n} p_i^{+}(t) < B - B^{-} \tag{6}$$

$$\sum_{t=1}^{m} \sum_{i=1}^{n} p_i^{-}(t) \leq B^{-} \tag{7}$$

$$\sum_{j=1}^{n} \tau_{ji}(t) - \sum_{j=1}^{n} \tau_{ji}(t) \leq \varphi_i(t), \forall i, t \tag{8}$$

$$\varphi_i(t+1) = \varphi_i(t) + \sum_{j=1}^{n} (\tau_{ji}(t) - \tau_{ji}(t)) \forall i, t \tag{9}$$

Note that the difference with the existing price scheme can be found in Eq. (6) and (7), where we consider two kinds of incentives. The overall budget of source and destination incentives remains as B. The difference is that some part of the budget B^{-} is assigned to conduct the destination incentive.

3 Hybrid Incentive Scheme

3.1 Benefits of destination incentive

Although the source incentive is a straighforward way to increase the service level, it cannot fully unitize the power of user incentive. Besides source incentive, we use

Algorithm 1 The hybrid incentive schema.

Input: The source and destination of user u_k
Output: Alternative bike b to rent and alternative location b' to return
1: $h_i, h_j) \leftarrow$ index of the region of location u_k and u'_k
2: $N(h_i), N(h_j)) \leftarrow$ neighboring regions of h_i and h_j
3: Incentive price set $I = (p_i^+(t), p_i^-(t)) \leftarrow$ the pricing scheme learned by the reinforcement learning agent
4: **for** all bikes in $N(h_i)$ and return locations in $N(h_j)$ **do**
5: u_k calculates the net profit of source and destination detours, i.e., incentive prices minus detour costs
6: u_k chooses the bike b and return location b' with maximum net profit which is denoted as $p_{\max}$
7: **if** $p_{\max} < 0$ **then**
8: u_k refuses incentives.
9: **return** b, b' as the user's alternative pick-up and drop-off locations.

Fig. 4 to illustrate the dock-less BSS can also be balanced through the destination incentive. In the figure, the blue nodes represent areas whose return events are more frequent (i.e. the number of bikes on the area increases) in the given time window. The amount of extra return events for each region is shown in the figure and each node's area is proportional to the extra value. The red nodes have the opposite meaning. By applying the destination incentive, i.e., incentivizing users to return bikes to neighbor regions, the imbalance usage can be greatly eased. A possible assignment for users is shown in the figure. The arrows indicate the destination incentive direction and the number of users needed is shown along the arrows. However, the imbalanced demand cannot be totally satisfied since the number of extra return events could be different from the number of extra rent events. Although the spatial distribution cannot be fully balanced, the service level of the system can be improved because more users are able to rent bikes.

The unique benefit of the destination incentive is illustrated through the toy example in Fig. 5. By applying the destination incentive, the service level can increase to 2. Only applying the source incentive cannot achieve such a service level. In the example, we consider that there are three users appearing in time slot 1, 2, 3 at location u_1, u_2, u_3 shown in the figure respectively. Their corresponding destination is u'_1, u'_2 and u'_3. There is only one bike in the map and it is returned at location R at $t = 1$ without considering destination incentive. As shown in Figs. 5 (a), (b) and (c), only user u_1 can successfully finish her trip and users at u_2 and u_3 cannot rent a bike since the detour distance exceeds the limitation. In this case, the number of satisfied users is 1. However, if the destination incentive is allowed at $t = 1$, the bike can be returned to location R' instead of R. Although it is too far for user u_1 to rent the bike, users u_2 and u_3 can finish their journeys without any detour. That is, the number of satisfied users increases to 2 with a well-designed alternative return location suggestion, and the users' total walk distance is the same as following the source or destination incentive scheme.

Although the destination incentive may have a better control on the trend of the bike flow, deciding the return location is complex due to the large size of the

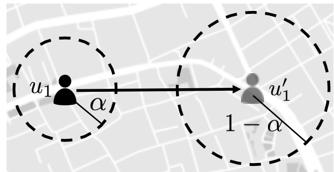


Fig. 6 Adaptively adjusting source and destination incentive.

potential returning points. Especially for dock-less BSSs, a bike could be returned to any location as along as it does not block others. The large solution space makes the learning algorithm hard to converge. To cast off that complexity, we propose to sample the metric middle point of each region as the potential returning point. It means that if a user accepts the destination incentive, the system would suggest the metric middle point of the destination region as the returning point. This additional constraint not only reduces the action space for reinforcement learning algorithms, but also reduce management difficulties for the system operator.

The trade-off is that the efficiency of the destination incentive might be reduced. That is because, choosing the middle point as returning point may cause the unnecessary detour which may increase the detour distance of a user. If the pay off brought by the destination detour is not more than detour distance wasted on a certain user dynamic, the destination detour is not useful anymore.

3.2 A Hybrid Incentive Scheme

Either the source incentive or destination incentive itself has its own shortage on certain user dynamics. Therefore, besides considering incentivizing users to just pick up or just drop off bikes in neighbor regions, we propose to combine these two kinds of incentives, and build a hybrid incentive scheme. The intuition behind the hybrid incentive scheme is that the scheme could adaptively adjust the source and destination incentive based on different imbalance situations.

In the hybrid incentive scheme, the state and action space in the MDP is enlarged because of the destination detour budget B^- and price p^-. Specifically, a state vector s_t is constructed by $\sum_{h_i} \varphi_i(t)$, $\sum_{h_i} D_i(t-1)$, $\sum_{h_i} \Lambda_i(t-1)$, $B^+ - \sum_{h_i,t} p_i^+(t)$, $B^- - \sum_{h_i,t} p_i^-(t)$ and unserviced events in previous time-slots. The first term represents the number of unused bikes over the city at the beginning of t. The total amount of bikes over the city is constant, but the number of unused bikes may vary over time because of the fluctuated usage of users. The $\sum_{h_i} D_i(t-1)$, $\sum_{h_i} \Lambda_i(t-1)$ represents the total number of rent and return events over the city, which captures the temporal bike usage information to the MDP. The $B^+ - \sum_{h_i,t} p_i^+(t)$ calculates the remaining budget for the source incentive, and $B^- - \sum_{h_i,t} p_i^-(t)$ calculates the remaining budget for the source and destination incentive respectively. The MDP ends either when t reaches the time-slot upper bound or any of remaining budget for source or destination incentive is empty.

An action vector a_t in A for time-slot t is constructed by the source incentive price vector $(p_i^+(t), i = 1, \ldots, n)$ and source incentive price vector $(p_i^-(t), i = 1, \ldots, n)$. Although the transmission probability P is built on the enlarged state and action spaces, the state transmission still can be simulated via our environment model. The reward r of the hybrid incentive scheme is constructed by rewards from source incentive $r^+(s_t, p^+)$ and rewards from the destination incentive $r^-(s_t, p^-)$.

To adapt the modification to the MDP, we extend the actor-critic framework in [25]. The size of the actor network is enlarged as shown in Fig. 7. The actor network 1 is used to learn the source incentive prices $p^+(t)$, and the actor network 2 is used to learn the destination incentive prices $p^-(t)$. As for the critic network, the sub-Q-value of each region h_i at step t is evaluated based on $(p_i^+(t), p_i^-(t))$ instead of just considering $p_i^+(t)$, and the estimation of Q-value changes correspondingly.

3.3 Adaptively Adjust Source and Destination Incentive

Besides adjusting the learning framework, we also propose two different ways to adjust the ratio of source and destination incentive price. One way is to adjust the strength of source and destination incentive by controlling the ratio of source and destination incentive budget. It is achieved by splitting the total budget into source and destination incentive budgets based on a ratio ρ.

Definition 1 (Budget division) Assume the total budget available is B, and the budget division ratio is ρ. Then the budget appointed to source incentive is ρB and the remaining $(1-\rho)B$ is used for the destination incentive.

Under this scheme, the remaining budget of source and destination incentive in the initial state s_0 becomes:

$$B^+ = \rho B,\ B^- = (1-\rho)B$$

The overall reward during a day under policy π_θ becomes:

$$J_{\pi_\theta} = E[\sum_{k=0}^{\infty} \gamma^k (r^+(a_k, s_k) + r^-(a_k, s_k)) | \pi_\theta, s_0]$$

The other way is to adjust the ratio between detour distances of source and destination incentive. It is achieved by adding the maximum detour constraint to users in the environment model. Let l denote the maximum detour distance that a user can accept, including source and destination detour. The value of l can be extracted from a user survey when applying the scheme in the real world. We split l into two parts l^s and l^d which correspond to the maximum source detour and maximum destination detour respectively. Let α denote the adjust parameter between l^s and l^d.

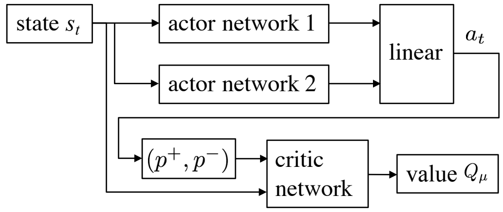


Fig. 7 The learning framework for hybrid incentive scheme.

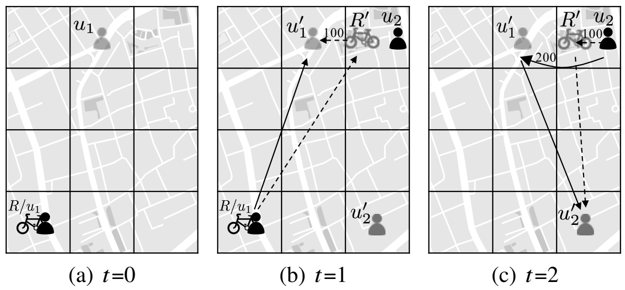


Fig. 8 Illustration of combining source and destination incentive.

Definition 2 (Detour distance division) Given the maximum detour distance l of each user and parameter α, the maximum detour under source incentive is $l^s = \alpha l$ and the detour under destination incentive is $l^d = (1-\alpha)l$.

To keep the consistence with the environment model, we assume the user rejects to detour either if his/her detour distance exceeds the limitation or if he/she cannot gain profit from the detour. Through this setting we try to limit each region's source and destination incentive among all time-slots.

Formally, based on α, we attempt to limit the source and destination incentive as:

$$p_i^+(t) < C + \eta\alpha^2 l^2 \text{ and } p_i^-(t) < C + \eta((1-\alpha)l)^2 \;\; \forall t \in T$$

The budget division strictly imposes restrictions on budgets of source and destination incentives, while the detour distance division restricts the source and destination incentive price on estimation. Either kind of incentive is adaptive among regions, and the sum of incentive prices cannot exceed the corresponding budget. The budget division is applied to the initial state of the MDP. The detour distance division is applied to the environment, the incentive greater than the limitation cannot bring benefits to the scheme.

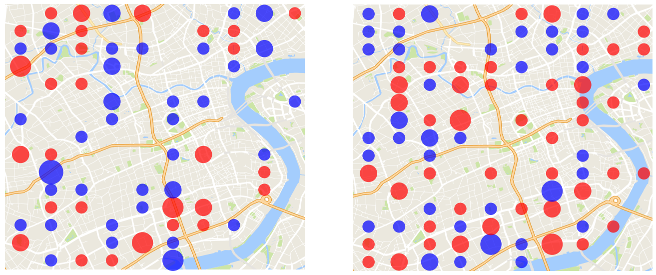
3.4 Properties

Actually, besides the adaptive adjustment, the hybrid incentive scheme can also help to break a long detour distance of one user into two short detour distances of different users. More clearly, we use the example in Fig. 8 to show the benefits brought by applying source and destination incentives at the same time. We consider there are two users arriving at time slots 0 and 1 respectively. Their origins are u_1 and u_2 and destinations are u_1' and u_2' respectively. There is one bike located at R at the beginning time slot $t = 0$. The solid lines show users' paths if only the source incentive is allowed. By following the solid lines, user u_1 rents the bike from R and returns it to u_1'. Then, the user u_2, who comes after u_1 returns the bike, has to rent the bike at u_1' with 200m detour, while the detour distance of user u_1 is 0. In contrast, the dashed lines show paths when both the source and destination incentives are allowed. User u_1 could return the bike to R' with 100m destination detour. In this case, user u_2 could rent the bike from R' with 100m source detour. The relatively long detour of user u_2 is equally shared by u_1 and u_2 in our hybrid incentive scheme. According to the survey [27], the growth rate of each user's detour cost is proportional to the square of his/her detour distance. Our hybrid system provides a probability that let users with relatively shorter detour distance help to share the long detour distance. It mitigates the burden for users with relatively longer detour distances and helps to attract more users to accept the incentive. It may help to incentive more users to involve in the rebalancing and increase the number of satisfied users to the system. With more potential users joining rebalancing, the system is easier to choose proper users for rebalancing.

4 Hybrid Incentive in Docked BSS

In this section, we consider the docked BSS in which users must rent/return bikes at bike stations deployed by the BSS operator. In the docked BSS scenario, our objective is also maximizing the service level of the BSS in the daily service period.

In the docked BSS rebalancing, let H denote the set of stations rather than regions, and let h_i denote the station i. The docked BSS suffers more from the imbalance bike distribution. Specifically, it would cause *overflow* and *underflow* stations. The overflow stations are the stations that are full of bikes. Users cannot return bikes to overflow stations. The underflow stations have no bikes, and users cannot rent bikes. We also follow the user incentive approach to resolve the overflow and underflow issues. Specifically, at each time slot t, the BSS operator would assign source and destination incentives for each station h_i. The source incentive price is denoted as $p_i^+(t)$ which is the amount of the monetary incentive for a user who rents bikes from the station. We provide source incentives to resolve the overflow issues. Similarly, the destination incentive price $p_i^-(t)$ means the incentive for a user who returns bikes to h_i. It is used to resolve the underflow issues of a station. The source and destination



(a) Pick-up events in 8 a.m. - 9 a.m. (b) Pick-up events in 6 p.m. - 7 p.m.

Fig. 9 The temporal and spatial imbalanced distribution in the dock-less BSS dataset.

incentive prices would not be provided at the same station. That is, if there is a source incentive at station h_i at time t, the destination incentive price $p_i^-(t) = 0$.

The challenge of rebalancing a docked BSS is the capacity limitation of each bike station. Let c_H denote the vector of station capacities of stations in H. Specifically, even if a station is located in a popular area and has a large number of bike renting demands, the station cannot hold more bikes than its capacity. The destination incentive might be infeasible for these fully occupied stations. The reinforcement learning agent needs to know the current number of bikes in each bike station at each time slot, along with the capacity of the station.

Besides the capacity limitations, there is no concept of neighborhood region in the docked BSS. For the dock-less scenario, we use neighborhood regions to reduce the complexity for the reinforcement learning agent. The incentive price for each region is affected by its four neighborhood regions rather than all regions in the map. For the docked scenario, we also need to reduce the state space for the agent. In particular, there are usually hundreds of stations in a city. When determining the incentive price for a station, taking states of all stations into consideration would also lead to a large solution space. Therefore, similar to using neighborhood regions, we only consider states of k *nearby stations* when determining the incentive price for each station in a docked BSS. The k nearby stations of station h_i are defined as the first k nearest stations to h_i. Same as neighborhood regions, the k nearest stations of h_i are determined and could be hard coded into the reinforcement learning agent.

Considering the differences between docked and dock-less BSSs, we extend our reinforcement framework to learn the source and destination incentive price for docked BSSs by enlarging the state space (adding station capacities as features) and replacing neighborhood regions with k nearby stations. The state and actor vectors of the reinforcement learning agent are updated to:

$$s'_t = (s_t, c_H), a'_t = (b(t) \in \{1, -1\}, p_i(t), i = 1, \ldots, n).$$

Specifically, the state vector of the reinforcement learning becomes $s'_t = (s_t, c_H)$ where s_t is the state vector used for dock-less BSS rebalancing and c_H is the vector of capacities of stations in H. To make sure the source and destination incentive

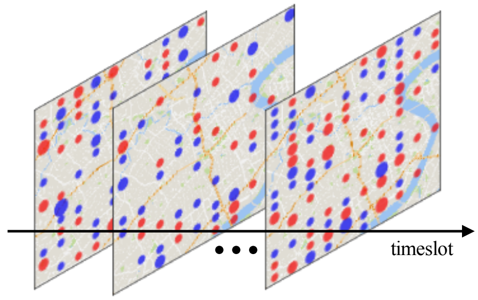


Fig. 10 Rebalancing among multiple timeslots.

prices at a station would not be positive simultaneously, we update the action vector to $a_t' = (b(t) \in \{1, -1\}, p_i(t) \in \mathbb{R}^+, i = 1, \ldots, n)$ where $b(t)$ is used to specify whether it is the source incentive or the destination incentive, and $p_i(t)$ represents the incentive price for station h_i.

The environment model is also modified to fit the docked BSS rebalancing scenario. The detour cost model used for dock-less BSSs cannot be used for docked BSSs since there are no neighborhood regions in docked BSSs. As a result, we update the cost model for docked BSSs and remove the limitations caused by neighborhood regions. Instead, we set maximum detour distances for users based on their original trip lengths. Assume a user plans to rent bikes from the station h_i and returned to h_i'. The alternative pick-up station is h_j. Then, in docked BSSs, the source detour cost of the user is:

$$c(h_i, h_j) = \begin{cases} C{+}\eta \cdot dis(h_i,h_j)^2 & dis(h_i,h_j){<}\kappa \cdot dis(h_i,h_i'). \\ +\infty & \text{otherwise.} \end{cases}$$

where the parameter κ is used to adjust the maximum detour distances. For example, when we setting $\kappa = 1$, we assume a user would not willing to take a detour whose length is longer than the user's original journey. The cost model for destination detour is updated in the same way.

By updating the state and action spaces of the reinforcement learning agent as well as the environment model, we extend our hybrid incentive scheme for dock-less BSSs to docked BSSs. The performance of the extended scheme is tested on real-world datasets and the results are illustrated in Section 5.

5 Experiment

5.1 Dataset

We use the data published by *Mobike* to construct the dock-less dataset, and use the trip history data published by NYC to construct the docked dataset. The NYC dataset contains more than 1.5 million trip records for 328 bike stations. The Mobike dataset contains more than 100k trip records of Shanghai. The record of each trip includes trip duration (in seconds), trip start (end) time and date, start (end) latitude and longitude, etc. The summery of the Mobike dataset is shown in Table II. We first visualize the imbalanced bike distributions in those datasets.

The spatial imbalance of the Mobike dataset is shown in Fig. 9. To illustrate the spatial imbalance we plot the usage of region in Fig. 7 for AM rush hours (7:00 - 9:00 AM) and PM rush hours (6:00PM - 8:00 PM). The blue nodes represent stations whose return events are more frequent (i.e. the number of bikes on stations increase) in the given time window. The node's diameter is proportional to demands of the corresponding station. The red nodes have the opposite meaning. Fig. 10 shows the variation of spatial imbalance over multiple time-slots.

In addition, we investigate the bike imbalance of the NYC dataset. Fig. 11(a) shows the statistics on bike user's trip duration. Fig. 11(b) illustrates the statistics of the temporal usage distribution of trips on 08/01/2016 (Monday). It shows that more than 55% of trips are shorter than 10 minutes. Fig. 11(b) further shows that the demands of bikes are not even during a day. There also exist morning and evening peak hours.

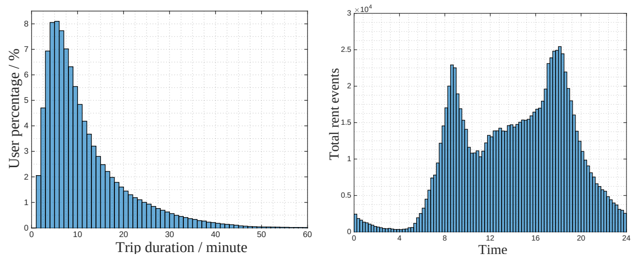
Table 2 The Mobike Dataset

Data Source	Mobike traces in Shanghai	
Time Span	8/1/16 to 9/1/16	
Weekdays (Weekends)	24 (8) days	
Bike Data	# Bikes 79,063	# Trips 102,361

5.2 Experiment Setup

In our experiment, the environment model is built on *OpenAI Gym*, a toolkit for comparing reinforcement learning algorithms. Specifically, a day is temporally divided into 24 time-slots and the Shanghai city is spatially divided into 20×40 regions. The effective area of the city is bounded by [30.841°N, 31.477°N] and [120.486°E 121.971°E]. Users' request time, locations and destinations are extracted from the Mobike trace data. Through the statistics of unique bike ID, there are totally 79, 063 bikes used in the dataset. Considering the retirement of broken bikes, the actual number of bikes may be less than that amount. Users' riding speed is chosen as the mean speed of all users and the walking speed is assigned as 5 km/h. The cost of user detour obeys the cost model introduce in Subsection 2.3.

When training the hybrid incentive scheme, the Adam algorithm [18] is used to optimize both actor and critic networks. The learning rates for training both parts are set as 10^{-4}. In each step, to explore the more action space, a Gaussian noise is added to each action generated from the actor network. Although [20] proposed to



(a) Duration distribution (b) Rent event distribution

Fig. 11 Statistics of the NYC dataset.

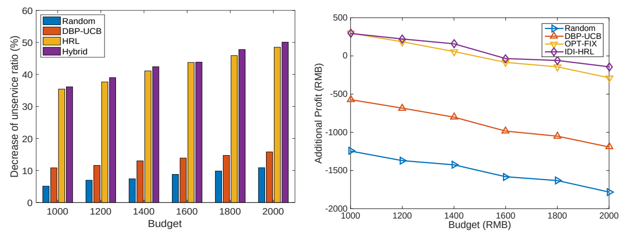
add Uhlenbeck-Ornstein noise to actions, the Gaussian noise is used for simplicity. The discount factor γ in the MDP is chosen as 0.99.

In the first set of experiments, we compare the performance of our algorithm with others under different budget. The budget is varied from 1000 to 2000 and the performance is quantified by the decreased unserviced ratio defined in [25]. The number of unserviced users increases by one if the user cannot find a bike to ride and he/she is not satisfied with any source incentives which is offered by the system. Let N_1 denote the number of unsatisfied users with no incentive, and let N_2 denote the number of unsatisfied users with incentive. Then, the corresponding unserviced ratio is defined as $(N_1 - N_2)/N_1$.

The second set of experiments focuses on the number of satisfied users. The number of satisfied users is proportional to the income of the BSS. We assume the BSS operator charges 1 for each user who rents the bike. Therefore, the profit of the operator can be calculated by subtracting the budget spent for incentive from the overall income of a day.

We also test the influence of the initial bike amount. If the initial bikes are sufficient enough, then each user can find a bike without a detour and the maximum service level is achieved. However, the number of bikes is limited in each region. Therefore, wisely spending the budget to achieve a better service level is important. The last set of experiments focus on the rebalance performance across multiple days. Our first comparison algorithm is the source-incentive-only scheme [25] which is denoted as HRP. The second comparison algorithm is the DBP-UCB [27] which is one of the state-of-the-art bike rebalancing approaches based on user incentive. A randomized incentive scheme is used as a baseline.

Then, we tested our docked BSS rebalancing scheme. We compare our leaning-based scheme with a fixed incentive scheme used by the operator of the NYC Citi bike. Specifically, the Citi bike launched the "Bike Angels" project and would give users fixed points if they rent/return bikes at specific locations. Each point worth about 0.1$. During the experiment, we denote this rebalancing scheme as *Fixed*. Besides, we use the *Random* scheme, which assigns incentive prices to stations randomly, as the baseline.



(a) Comparison on DUR. (b) Comparison on profit.

Fig. 12 Comparison on DUR and additional profit with of varying budget.

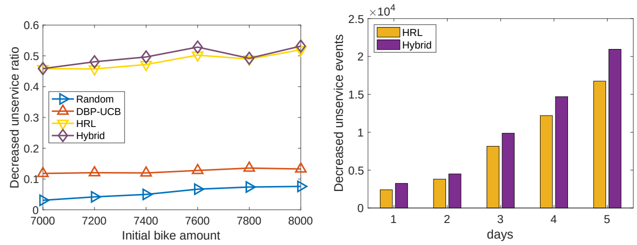
5.3 Results

We illustrate our experiment result on decreased unserviced ration in Fig. 12(a). From the figure we can conclude that the performance of our hybrid approach achieves better performance than other approaches. Comparing with the HRL that just considers the source incentive, we can conclude that adaptively allocating incentive on source detour and destination detour can bring additional benefits on the service level. It is reasonable since the source and destination incentives are included in the action spaces of the hybrid incentive scheme. By comparing HRL and DBP-UCB we can conclude that the reinforcement learning can greatly improve the service level since the reinforcement learning considers further reward when choosing the action for each state. The performance trend of all approaches shows that more user requests can be satisfied with a higher budget, even for the randomized policy.

The additional profit brought by the incentive is illustrated in Fig. 12(b). As stated in [25], the HRL can bring additional benefits to the BSS operator when the budget is not too large. The hybrid incentive scheme also can gain profits from the incentive, which is arguably one of the most important feathers to BSS operators. However, with the increase of budgets, the profit decreases. It illustrates that the number of satisfied users increases more slowly with the increasing budget. That is to say, the BSS operator may not find it worthwhile to totally rebalance. The totally rebalanced system means that all user requests can be satisfied. The DBP-UCB and randomized scheme can bring additional profits to the system with a budget less than 1000 within the Mobike dataset.

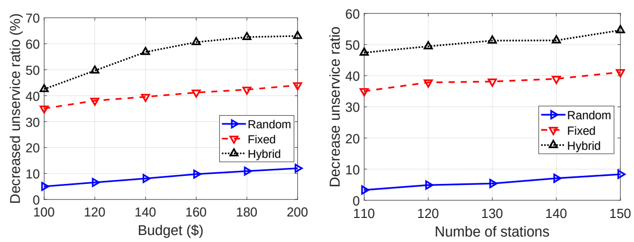
Fig. 13(a) shows the influence of the initial bike amount. With more initial bikes placed in the city, incentive schemes are more likely to achieve better performance. The increasing ratio in the figure is not as sufficient, and the reason could be that the initial bikes are uniformly distributed among the city, and adding bikes to regions with nearly no user request may cause the waste of bikes. If the distribution of initial bike could fit user request, the increased initial bike amounts may greatly improve the service level.

Fig. 13(b) shows the rebalance performance over multiple days. We count the decreased unservice events since it is additive. The difference between the HRL and the hybrid incentive increases with the increase of the number of days. It shows that



(a) Comparison on bike amount. (b) Long-term comparison.

Fig. 13 Cumulative density function and long-term performance comparison.



(a) Varying the budget amount. (b) Varying the number of stations.

Fig. 14 Performance comparison on docked BSS dataset.

the hybrid incentive scheme keeps a better distribution than the HRL. As we have shown in the previous section, the destination incentive, to a certain extent, is eager to place bikes in regions with more request. These bikes are more likely to be used when the number of time-slots increases. It may explain that the advantage of the hybrid scheme is more obvious with a larger number of time-slots.

The experiment results on the docked BSS dataset— the NYC dataset is shown in Fig. 14. Fig. 14(a) shows the performances of different schemes under different incentive budget. Not surprisingly, all schemes could achieve a higher service level when a larger budget is provided. Among them, our hybrid incentive scheme makes better use of the budget and have the highest service level. It shows the reinforcement learning agent could learn a better way of allocating incentives among bike stations, compared with a fixed incentive scheme. Fig. 14(b) shows the experiment results under different number of stations. The amount of budget per station is fixed during the experiment. This result shows that our hybrid incentive scheme is robust when the scale of the docked BSS is expanded. The learning agent is still capable to allocate incentives wisely to keep the service level at a high level, and it outperforms the fixed incentive scheme by about 30.7%-35.3%.

6 Related Work

With the booming of BSSs, more and more researchers devote their effort to related issues including user demand prediction [6, 19, 24, 29] , bike rebalance strategy [26, 22, 27, 31, 16, 13], station location optimization [21, 5], bike lane planning [2], suggestion of user's journeys [30, 32, 7] . We focus on the studies that have been conducted on rebalance strategy designing issues, which are closely related with our work.

Before designing efficient bike rebalancing scheme, accurately predicting user demands for BSSs is critical. The existing demand predication methods could be group as station level and cluster level prediction approaches. The station level prediction is designed to predict the number of rent/return events at each bike station in docked BSSs, such as [15, 17]. However, it ignores the potential depends among bike stations, and may not generate inaccurate demand predictions [19]. To overcome this, Li et al. [19] proposes to cluster similar stations. Specifically, they proposed to use transition patterns and station locations to cluster bike stations first. Instead of predicting demand of each bike station, they predict the demand of each cluster. Chen et al. [6] further improve the prediction accuracy by considering more features such as traffic and social event. Du et al.[**?**] adapt a Convolutional Neural Network for the demand prediction. Their algorithm could find virtual stations by using density-peak based clustering. By utilizing demand prediction, our approach aims to optimize the worker assignment during rebalancing.

Rebalancing strategies designed for docked BSSs have been widely studied. Typically, there are two major approaches which are the truck-based and the user-based approach. The truck-based approach such as [4, 14] means the BSS operator hires a fleet of trucks to transport bikes from overflow stations to underflow stations. Chemla [4] proposed a single-vehicle rebalancing problem, where each station could be used as a buffer to temporarily store some bikes. Their rebalancing algorithm is based on brand-and-bound, which can be used for small size BSS systems. When the number of stations exceeds 100, the time cost is significant. Liu et al. [22] proposed a method that first clusters bike stations according to geographic information and station status, and then assigns a truck to each cluster. They model the bike rebalancing as an integer programming problem and use integer programming solvers to optimize route for trucks used for bike rebalancing. Different from those works, we follow the user-based approach which is more flexible and cost-efficient.

As for the user-based approach like [28, 27], the BSS operator gives incentive to users and suggest them to rent or return bikes at certain stations. User-based approaches expect the BSS can achieve self-balance. They improve the overall service level by controlling user's dynamics through incentive. The user-based approach is more flexible. Not like truck-based approaches which could only apply a few rounds of rebalancing in a day, the user-based rebalancing lasts continuously only if the budget is sufficient. Designing the pricing mechanism is the key problem in these approaches. Waserhole [28] presented a dynamic pricing mechanism that incentivizes users to redistribute bikes by providing alternative rental prices. Singla et al. [27] proposed a pricing mechanism to incentivize users via crowdsourcing. For dock-less BSSs, besides the source incentive scheme based on reinforcement learning proposed by Pan et. al [25]. Liu et. al [24] propose a demand prediction method. Their inference model combines Convolutional Neural Network and factor analysis techniques. Based on an precise demand prediction, some docked rebalanced scheme may be extended to the dock-less scenario. Caggiani et. al [3] proposed a dynamic bike rebalance method including a prediction scheme of the number and position of bikes and a relocation decision system. Our hybrid scheme is an end-to-end system and the incentive price can be given without demand prediction.

7 Conclusion

We investigate the bike rebalancing problem for both dock-less and docked BSSs in this paper. We illustrate that the imbalanced bike distribution in BSSs might cause bike overflow and underflow events. Those events may bring congestions to the city or decrease the service level of BSSs, and rebalancing BSSs timely is necessary. We follow the user-based approach for rebalancing, and propose to adaptively provide both source and destination incentives to users with the objective of maximizing the service level. We adapt a reinforcement learning framework in [25] to overcome the complex user dynamics for dock-less BSSs. In addition, we extend the learning framework to the docked BSS rebalancing problem. The capacity of each station is added to the state space of the reinforcement learning agent and the environment model is also updated to fit the docked BSS scenario. We use real-world trace data from Mobike and NYC Citi bike to test our dock-less and docked rebalancing scheme, respectively. Experiment results show that providing both source and destination incentives could achieve a higher service level compared with the state-of-the-art source-incentive-only scheme. Our extended scheme outperforms the fixed incentive scheme which is currently implemented by the City bike in NYC.

Acknowledgement

This research was supported in part by the National Science Foundation grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1618398, CNS 1651947, and CNS 1564128.

References

1. Ashar, R.: Hierarchical learning in stochastic domains. Ph.D. thesis, Citeseer (1994)
2. Bao, J., He, T., Ruan, S., Li, Y., Zheng, Y.: Planning bike lanes based on sharing-bikes' trajectories. In: Proc. of ACM SIGKDD, pp. 1377–1386 (2017)

3. Caggiani, L., Camporeale, R., Ottomanelli, M., Szeto, W.Y.: A modeling framework for the dynamic management of free-floating bike-sharing systems. Transportation Research Part C: Emerging Technologies **87**, 159 – 182 (2018)
4. Chemla, D., Meunier, F., Calvo, R.W.: Bike sharing systems: Solving the static rebalancing problem. Discrete Optimization **10**(2), 120–146 (2013)
5. Chen, L., Zhang, D., Pan, G., Ma, X., Yang, D., Kushlev, K., Zhang, W., Li, S.: Bike sharing station placement leveraging heterogeneous urban open data. In: Proc. of ACM Ubicomp, pp. 571–575 (2015)
6. Chen, L., et al.: Dynamic cluster-based over-demand prediction in bike sharing systems. In: Proc. of ACM Ubicomp, pp. 841–852 (2016)
7. Cheng, P., Xu, C., Lebreton, P., Yang, Z., Chen, J.: Terp: Time-event-dependent route planning in stochastic multimodal transportation networks with bike sharing system. IEEE Internet of Things Journal **6**(3), 4991–5000 (2019)
8. Dayan, P., Hinton, G.E.: Feudal reinforcement learning. In: NIPS, pp. 271–278 (1993)
9. Dean, T., Lin, S.H.: Decomposition techniques for planning in stochastic domains. In: Proc. of IJCAI, vol. 2, p. 3 (1995)
10. DeMaio, P.: Bike-sharing: History, impacts, models of provision, and future. Journal of Public Transportation **12**(4), 3 (2009)
11. Duan, Y., Wu, J.: Optimizing rebalance scheme for dock-less bike sharing systems with adaptive user incentive. In: 20th IEEE MDM, pp. 176–181. IEEE (2019)
12. Duan, Y., Wu, J.: Optimizing the crowdsourcing-based bike station rebalancing scheme. In: 39th IEEE ICDCS, pp. 1559–1568. IEEE (2019)
13. Duan, Y., Wu, J.: Spatial-temporal inventory rebalancing for bike sharing systems with worker recruitment. IEEE Transactions on Mobile Computing pp. 1–1 (2020)
14. Duan, Y., Wu, J., Zheng, H.: A greedy approach for vehicle routing when rebalancing bike sharing systems. In: IEEE GLOBECOM, pp. 1–7. IEEE (2018)
15. Froehlich, J., Neumann, J., Oliver, N., et al.: Sensing and predicting the pulse of the city through shared bicycling. In: Proc. of IJCAI, vol. 9, pp. 1420–1426 (2009)
16. He, S., Shin, K.G.: Dynamic flow distribution prediction for urban dockless e-scooter sharing reconfiguration. In: Proceedings of The Web Conference 2020, pp. 133–143 (2020)
17. Kaltenbrunner, A., Meza, R., Grivolla, J., Codina, J., Banchs, R.: Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. Pervasive and Mobile Computing **6**(4), 455–466 (2010)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Li, Y., et al.: Traffic prediction in a bike-sharing system. In: Proc. of ACM SIGSPATIAL, p. 33 (2015)
20. Lillicrap, T.P., et al.: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015)
21. Liu, J., Li, Q., Qu, M., Chen, W., Yang, J., Xiong, H., Zhong, H., Fu, Y.: Station site optimization in bike sharing systems. In: Proc. of IEEE ICDM, pp. 883–888 (2015)
22. Liu, J., Sun, L., Chen, W., Xiong, H.: Rebalancing bike sharing systems: A multi-source data smart optimization. In: Proc. ACM SIGKDD, pp. 1005–1014 (2016)
23. Liu, J., Sun, L., Li, Q., Ming, J., Liu, Y., Xiong, H.: Functional zone based hierarchical demand prediction for bike system expansion. In: Proc. of ACM SIGKDD, pp. 957–966 (2017)
24. Liu, Z., Shen, Y., Zhu, Y.: Inferring dockless shared bike distribution in new cities. In: Proc of ACM WSDM, pp. 378–386 (2018)
25. Pan, L., Cai, Q., Fang, Z., Tang, P., Huang, L.: Rebalancing dockless bike sharing systems. arXiv preprint arXiv:1802.04592 (2018)
26. Singla, A., Krause, A.: Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In: Proc. ACM WWW, pp. 1167–1178 (2013)
27. Singla, A., et al.: Incentivizing users for balancing bike sharing systems. In: Proc. of AAAI, pp. 723–729 (2015)
28. Waserhole, A., Jost, V.: Pricing in vehicle sharing systems: Optimization in queuing networks with product forms. EURO Journal on Transportation and Logistics **5**(3), 293–320 (2016)

29. Yang, Y., Heppenstall, A., Turner, A., Comber, A.: Using graph structural information about flows to enhance short-term demand prediction in bike-sharing systems. Computers, Environment and Urban Systems **83**, 101521 (2020)
30. Yoon, J.W., et al.: Cityride: a predictive bike sharing journey advisor. In: 13th IEEE MDM, pp. 306–311. IEEE (2012)
31. Yoshida, A., Yatsushiro, Y., Hata, N., Higurashi, T., Tateiwa, N., Wakamatsu, T., Tanaka, A., Nagamatsu, K., Fujisawa, K.: Practical end-to-end repositioning algorithm for managing bike-sharing system. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 1251–1258. IEEE (2019)
32. Zhang, J., Lu, P., Li, Z., Gan, J.: Distributed trip selection game for public bike system with crowdsourcing. In: INFOCOM. IEEE (2018)