Computers & Industrial Engineering 169 (2022) 108214

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Task assignment in predictive maintenance for free-float bicycle sharing systems

Lan Lu [a], Shichen Zhao [b], Qiao-Chu He [b,*], Ning Zhu [c]

[a] International Institute of Finance, School of Management, University of Science and Technology of China, Hefei 230000, China
[b] School of Business, Southern University of Science and Technology, Shenzhen 518055, China
[c] College of Management and Economics, Tianjin University, Tianjin 300072, China

ARTICLE INFO

Keywords:
Free-float bike-sharing system
Faulty bike maintenance problem
Predictive decision analytics
Dedicated policy
Second-order conic programming

ABSTRACT

The growth of Free-Float Bike-Sharing Systems (FFBSs) is heavily impeded by faulty bike maintenance among other operational challenges. In this paper, we aim to improve the efficiency of faulty bike maintenance by predicting faulty bikes in order to make better maintenance assignment decisions. Inspired by industry practice, we identify the role of “black holes” in accurate predictions of faulty bikes: locations with morbidly high faulty rates, which can be characterized using data-driven approaches (clustering and convex hull). Based on the prediction result, we propose two maintenance policies, i.e., the pooling model and the dedicated model, for the faulty bike maintenance assignment problem with the objective of minimizing the sum of maintenance time cost and travel time cost. Finally, we provide a tractable reformulation via linear mix-integer Second-Order Conic Programming (SOCP) and conduct a case study with real data. Our analysis identifies the main trade-off between routing efficiency and maintenance efficiency in the different maintenance policies. We find that the pooling policy concentrates on routing efficiency while the dedicated policy emphasizes maintenance efficiency. Moreover, we demonstrate the importance of “black holes” in the prediction of faulty bikes. In the case study, we observe that bikes in “black holes” are about 70% more likely to be faulty than those out of “black holes.” We find that the improvement due to prediction is significant even when the prediction is imperfect. In our case study, when prediction accuracy exceeds 65%, we can observe the cost reduction by prediction in the faulty bike maintenance problem.

1. Introduction

The free-float bike-sharing system (FFBS) is becoming increasingly popular all over the world because it is convenient, inexpensive, efficient, and eco-friendly. In China, FFBS first appeared in 2015, and this convenient and flexible way of travel quickly became popular in different Chinese cities. Different from the traditional docked bike-sharing system (DBS) where users can park bikes only at the docked stations, in FFBS, users can park bikes almost anywhere. Therefore, because the users are able to rent the nearest bike and return it at any appropriate location after the ride, FFBS systems can efficiently solve the first and last-mile problem. However, the FFBS system brings operational challenges as well as all these benefits, especially with regard to the maintenance of faulty bikes. Faulty bikes may lead to the following problems: (a) seriously threaten users’ safety, (b) too many faulty bikes will affect the quality of service and further influence the company’s reputation, and (c) have a bad influence on the city’s image. However, in FFBS, the fact that the bikes can be parked at any location results in a wide dispersion of faulty bicycles away from the starting and endpoints. This feature provides bike-sharing companies with a completely different and challenging scenario to handle the faulty bike maintenance problem. Therefore, it is very important for bike-sharing companies to solve the faulty bike maintenance problem.

In this paper, we focus on improving the performance of faulty bicycle maintenance by predicting faulty bikes in order to make better maintenance assignment decisions. First, before making maintenance assignment decisions, we need to predict the health status of bikes from data. We use a data-driven approach and propose two features to help us predict faulty bikes. The first one is inactive days. When a bike remains unused for several days, it is likely to be either faulty or improperly parked. The greater the bicycle’s inactive days, the more likely it will be faulty. The second feature is the “black hole.” In daily life, people may

* Corresponding author.
E-mail addresses: lluae@ustc.edu.cn (L. Lu), 11611917@mail.sustech.edu.cn (S. Zhao), heqc@sustech.edu.cn (Q.-C. He), zhuning@tju.edu.cn (N. Zhu).

https://doi.org/10.1016/j.cie.2022.108214
Received 4 November 2020; Received in revised form 16 April 2022; Accepted 27 April 2022
Available online 10 May 2022

park the bikes in some “black holes” (such as villages in cities and dead ends), and these bikes will be faulty (for example, the solar locks on the bicycles cannot be charged by sunlight in some areas) after a while. Thus, finding these “black holes” is one of the keys to predicting faulty bikes. Inspired by state-of-art industry practices (“geo-fencing” technology), we characterize these “black holes” by clustering faulty bikes’ locations and derive convex hulls for these clusters. Then, we predict the health status of bikes based on these “black holes” and their inactive days. We consider two types of faulty bikes to be maintained, i.e., fully faulty bikes (broken) and minor faulty bikes (typically in need of charging or minor mechanical adjustments). These two types of faulty bikes need different maintenance services with distinct costs, which has an important impact on the maintenance assignment decisions. Therefore, we aim to propose a data-driven prediction model for the health status of bikes that also addresses the maintenance assignment problem.

After predicting faulty bikes, we consider how to assign repairpersons to maintain these bikes. The maintenance assignment problem can be described as follows: the maintenance base has to send a fixed number of maintenance trucks to maintain all faulty bikes, including fully faulty and minor faulty bikes. The objective is to minimize the total time cost (including maintenance time cost and travel time cost) under the condition of maintaining all faulty bikes. Since there are two types of faulty bikes, we establish two mathematical models of the maintenance assignment problem: the pooling policy, where any truck can maintain both two types of faulty bikes; and the dedicated policy, where one truck can only maintain a single type of faulty bikes but is more proficient. We provide tractable reformulations via second-order conic programming (SOCP) for both policies. We acquire a real data set of more than 480,000 bicycles and nearly 350,000 bicycling users from a leading industry partner. Based on the real data, large-scale calculations are performed to obtain business insights and executable operation and maintenance strategies.

We summarize our contributions as follows. First, inspired by industry practice, we identify the role of “black holes” in the prediction of faulty bikes: locations with morbidly high faulty rates, which can be characterized using data-driven approaches (clustering and convex hull). We demonstrate the importance of the “black holes” in the prediction of faulty bikes. By applying our prediction model to more than 480,000 bikes with real data, we find that bikes in “black holes” are about 70% more likely to be faulty than those out of “black holes.” This new angle of view for the faulty bikes prediction helps us to improve the prediction accuracy as well as the efficiency of the faulty bike maintenance problem.

Second, we identify the main trade-off between travel efficiency and maintenance efficiency in the maintenance assignment problem. We propose two different assignment policies, i.e., the pooling policy and the dedicated policy, to minimize the total cost. Comparing our pooling policy and dedicated policy helps to identify the optimal assignment strategy. We find that the maintenance cost differentiation and faulty bikes’ distributions have significant impacts on the optimal policy: (1) The dedicated policy is preferred when the specialization degree is high, while the pooling policy is preferred in low-specialization situations. (2) The dedicated policy is preferred when faulty bikes are distributed more uniformly, while the pooling policy is better off when faulty bikes are distributed in clusters.

Third, we use an end-to-end approach (by developing a machine learning model with some significant features) to approximate the travel distance for each truck instead of solving the routing problem for every possible assignment. Based on this approximation, we reformulate the two models as tractable Second-order Cone Programming (SOCP) that can be solved efficiently by the Gurobi solver. With this reformulation, we conduct numerical experiments. The case study emphasizes the importance of health status prediction in the maintenance problem. With perfect prediction, a more than 7% cost reduction can be achieved. The cost reduction due to prediction is significant even when the prediction is imperfect (in our case, our data-driven model performs better as long as the prediction accuracy exceeds 65%).

The remainder of the paper is organized as follows. Section 2 provides a literature review. We propose two basic models in Section 3. The analysis is in Section 4, including health status, travel time, and computational techniques. We illustrate numerical case studies in Section 5 to compare the two models and obtain business insights and maintenance strategies. Section 6 provides the conclusions and limitations of the paper.

2. Literature review

We will discuss several main streams of relevant literature from their research topics, methods, etc.

Bicycle sharing and in general shared-mobility OM. Bicycle sharing and in general shared-mobility OM have been discussed in previous literature. In this study, the repositioning problem receives attention because of its natural connection to inventory control. Reference Shu, Chou, Liu, Teo, and Wang (2013) examined particular bike sharing operations in a networked environment. The researchers developed a network flow model and tested the impact of bike redistribution in a network and the ability of stations to move traffic through the system. This approach has been widely accepted in the operational research literature. For example, for operational-level decisions, Wang, Agatz, and Erera (2018) considered the optimization of a ride-sharing match using incomplete information. Reference Pal and Zhang (2017) contributed to an efficient solution algorithm for the free-floating bike-sharing systems by treating granular neighborhoods as stations. There are also some well-known papers that considered both operational and strategic decisions. For instance, Kabra, Belavina, and Girotra (2019) constructed a structural demand model and estimated the impact of the station accessibility and bicycle availability. Reference He, Zheng, Belavina, and Girotra (2020b) estimated a structured demanding model for station networks and emphasized the importance of the network effect. Study He, Hu, and Zhang (2020a) considered a fleet rebalancing problem of dynamic matching vehicle supply with travel demand to a free-floating vehicle-sharing system. They formulated the problem as stochastic dynamic programming and then reformulated the distributionally robust optimization. Reference Benjaafar and Hu (2020) comprehensively reviewed the applications of shared-mobility OM. In our study, the focus is on the faulty bike maintenance problem in the FFBS. We consider the faulty bike maintenance problem as two subproblems: an upper-level problem involving the prediction problem of faulty bikes, and a lower-level problem that deals with maintenance.

Predictive maintenance The papers concerning predictive maintenance are closely related to our paper. For example, Cai, Wu, and Zhou (2009) studied a preemptive-repeating model with imperfect information for an individual machine’s stochastic faults. A stochastic model to check maintenance policies for both real and suspected faults was developed by Sleptchenko and Johnson (2015). They formulated a model based on linear programming to optimize maintenance priorities. Reference Abbou and Makis (2019) considered a problem of the dynamic allocation of available repairpersons in unreliable production equipment systems. The failure mode is also an important consideration in the predictive machine maintenance problem. Most of the existing literature focused on soft failure and hard failure. Soft failure occurs when system degradation reaches a certain threshold, while hard failure happens because of a random shock that is catastrophic to the system. Some papers considered soft and hard failures independently (Huang & Askin, 2003; Bocchetti, Giorgio, Guida, & Pulcini, 2009; Ye, Xie, Tang, & Shen, 2012). In practice, the two failures are always correlated. Various dependencies were also discussed in some studies (Liu et al., 2013; Tang, Yu, Chen, & Makis, 2015; Hu, Sun, & Ye, 2020). Although the maintenance problem considered in this paper is different from the classical machine maintenance problem, we are inspired by these papers and incorporate features mainly related to hard failures. We focus on using machine learning methods to predict the health status of bikes and then solve the maintenance assignment problem based on prediction results.

Order assignment and vehicle routing In terms of maintenance assignment problems, our paper is closely relevant to the order assignment problem and Vehicle Routing Problem (VRP). Vehicle Routing Problems have been extensively studied, and they comprise a large body of literature. There are some classical instances of VRPs developed in stochastic contexts (see Laporte, Louveaux, & Mercure (1992), Gendreau, Laporte, & Séguin (1996), Campbell & Thomas (2008), Erera, Morales, & Savelsbergh (2010), Jaillet, Qi, & Sim (2016)). As a special case of VRPs, the traveling repairperson problem (TRP) and its variants deal with the single routing problem. For example, Luo, Qin, and Lim (2014) considered both vehicles and the distance constraints in a new TRP variant. Reference Dewilde, Cattrysse, Coene, Spieksma, and Vansteenwegen (2013) focused on the TRP with returns and described a multineighborhood tabu search algorithm. Reference Tulabandhula and Rudin (2014) studied a combination of machine learning and decision-making in the traveling repairperson problem. Moreover, Cappanera and Scutellà (2015) solved a family care problem that is formulated by considering routing, assignment decisions, and scheduling together. The dynamic scheduling problem of orders arriving dynamically in one day was studied by Klapp, Erera, and Toriello (2018). Reference Liu, He, and Shen (2018) set up a framework to optimize the last mile delivery service's order assignment. They also used stochastic and robust methods to optimize the points for uncertain service time of customer location. Our approach is similar to Liu et al. (2018) in that we directly estimate the total travel distance from the real data without specifying the visiting sequence. In this way, the travel time estimator trained offline will be used as a proxy to the VRP solution online. Reference Restrepo, Semet, and Pocreau (2019) addressed driver scheduling and routing for attended home delivery. They combined a set-covering model with a queuing model to allocate the required location to the site and measure the relevant service level. In our research, we integrate prediction models with the maintenance assignment problem and provide a tractable reformulation via the mixed-integer Second-Order Conic Programming (SOCP).

Transportation literature. Bicycle-sharing research has received early attention in its dock-based format. For instance, Lin and Yang (2011) studied the network design based on docking systems. In the recent transportation literature, researchers focused primarily on the bike rebalancing problem or bike repositioning problem, for example, Regue and Recker (2014). Moreover, a dynamic bike rebalancing method that jointly considers the customer dissatisfaction forecasting, bike rebalancing,and vehicle routing was presented by Zhang, Yu, Desai, Lau, and Srivathsan (2017). Reference Bruck, Cruz, Iori, and Subramanian (2019) solved a static bicycle repositioning problem that prohibits temporary operations. It is of great significance to predict the dynamic characteristics of the bike-sharing network and further link bike rebalancing with system strategy design under stochastic demands (de Chardon & Caruso (2015), Dell'Amico, Iori, Novellani, & Subramanian (2018)). Moreover, Çelebi, Yörüsün, and Işık (2018) presented a method for designing a bicycle sharing system that takes into account capacity allocation and location decision. Neumann-Saavedra, Crainic, Gendron, Mattfeld, and Omer (2020) proposed a formulation of service network design to solve the rebalancing problem in bicycle sharing systems. Reference Qin, Wang, Chen, and Wang (2021) focused on optimizing CO2 emissions in bike sharing systems and proposed a repositioning optimization model on the basis of partitioning strategies. However, these papers rarely considered the faulty bike problems in a bike-sharing system. Among the very few works that focused on the bike maintenance problem, Du, Cheng, Li, and Tang (2020) is relevant to us topic-wise; however, the operational objective and target are completely different: they incorporated the recycling of faulty bikes into a rebalancing problem. In our research, we focus on the identification and predictive maintenance of these faulty bikes. We focus on the faulty bike maintenance problem in the FFBS and help companies to reduce operational costs.

3. Basic model

We now introduce the framework of the faulty bike maintenance problem. In this problem, the maintenance base has to send m maintenance trucks to maintain n faulty bikes, including fully faulty (type 1) and minor faulty (type 0) bikes. The objective is to minimize the total time cost (including maintenance time cost and travel time cost) under the condition of maintaining all faulty bikes. We denote the assignment decisions by x_{ik}, where $x_{ik} = 1$ if bicycle i is assigned to truck k and $x_{ik} = 0$ otherwise. Let $f_i = \{0, 1\}$ denote the type of faulty bike, $t(f_i)$ denote the maintenance time cost for bike i that depends on its type, and l_k denote the travel distance of truck k. The objective is to minimize the total time cost as in the following:

$$\sum_{k=1}^{m}\sum_{i=1}^{n} t\left(f_i\right) x_{ik} + \sum_{k=1}^{m} \frac{l_k}{v_k} \tag{1}$$

where v_k is the average travel speed, $\sum_{k=1}^{m}\frac{l_k}{v_k}$ is the total travel time cost for m trucks, and $\sum_{k=1}^{m}\sum_{i=1}^{n} t(f_i)x_{ik}$ is the total maintenance time cost for n faulty bikes. Before solving the assignment problem, the main challenges are how to depict the travel distance l_k for each truck and the health status f_i for each bike.

To identify f_i, i.e., the type of each bike, we build a prediction model based on two key features: inactive days and "black holes." The inactive days examine the latest activity of bikes, and the "black holes" are the gathering places for faulty bikes. We denote the inactive days of bike i as π_i and define b_i as a binary variable, where $b_i = 1$ if the bike is in a black hole and $b_i = 0$ otherwise. We consider two types of faulty bikes to be maintained, i.e., type-1 bikes that are fully faulty (broken) and type-0 bikes that have minor faults (typically in need of charging or minor mechanical adjustments). Then, $f(\pi_i, b_i)$ is the prediction function, where $f(\pi_i, b_i) = 1$ if bicycle i is of type 1 and $f(\pi_i, b_i) = 0$ if bicycle i is of type 0. Details of the prediction function $f(\pi_i, b_i)$ are discussed in Section 4.1.

For the characterization of l_k, we use a machine learning approach with some significant features to predict the travel distance for every truck. These features, including the service region area, number of faulty bikes, and latitudinal difference, are identified based on data analytics. Instead of solving the routing problem for every possible assignment, our approximation can avoid extensively complicated modeling and provide a tractable formulation for the assignment problem. The details will be discussed in Section 4.2.

Then, we can formulate the faulty bike maintenance problem. The total time cost includes the maintenance time cost and travel time cost. Due to the different degrees of fault, the two types of bicycles have different maintenance costs. The type-0 bicycles are not completely faulty, so we can maintain them at lower cost than type-1 bicycles. We denote T_0 as the maintenance time cost for type-0 bicycles (e.g., charging) and T_1 as the extra maintenance time cost for type-1 bicycles compared with type-0 bicycles (e.g., repairing).

To solve the maintenance assignment problem, we propose a pooling policy. In the pooling policy, each maintenance truck can recycle both type-0 and type-1 bikes. Since the number of faulty bikes is fixed, the maintenance time cost is also fixed. Therefore, the objective of the assignment problem is equivalent to minimizing the travel time cost. However, since the two types of faulty bikes need different maintenance operations, we further consider the specialization of maintenance operations. We propose a dedicated model, where each truck can only recycle one type of bike. The maintenance time cost, however, is lower than that in the pooling model because repairpersons are more professional for their corresponding maintenance operations. The pooling model and dedicated model have their own advantages and disadvantages. We are interested in a comparison of these two models and to identify the conditions where each model fits. Next, we introduce the two models for the maintenance assignment problem.

Following are the notations we used in the model:

x_{ik}: Decision variables. $x_{ik} = 1$ if bicycle i is assigned to truck k, and $x_{ik} = 0$ otherwise.
π_i: Inactive days of bicycle i.
$b_i : b_i = 1$ if bicycle i is in a black hole; otherwise, $b_i = 0$.
$f(\pi_i, b_i)$: Prediction function. $f(\pi_i, b_i) = 1$ if bicycle i is of type 1 and $f(\pi_i, b_i) = 0$ if bicycle i is of type 0.
T_0^P: Maintenance time for type-0 bicycle (e.g., charging) under the pooling policy.
T_1^P: Extra maintenance time for type-1 bicycle compared with type-0 bicycle (e.g., repairing) under the pooling policy.
T_0^D: Maintenance time for type-0 bicycle (e.g., charging) under the dedicated policy.
T_1^D: Extra maintenance time for type-1 bicycle compared with type-0 bicycle (e.g., repairing) under the dedicated policy.
C: Capacity for truck k.
l_k: Total travel distance for truck k.
v_k: Driving speed for truck k.

3.1. Pooling policy (P)

After we predict the health status of bikes, we have all the faulty bikes' exact locations and their predicted health status. In the pooling policy, each maintenance truck can take both type-0 and type-1 bicycles. Our objective is to allocate these trucks to maintain all faulty bikes with a minimum total time cost. We need to make decisions about trucks' maintenance assignments under the constraints of maintaining all faulty bikes and without exceeding the truck's capacity. The pooling policy can be formulated as follows:

$$\min_{x_{ik}} \sum_{k\in K}\left(\sum_{i\in I}\left[T_0^P + f\left(\pi_i, b_i\right)T_1^P\right]x_{ik} + \frac{l_k}{v_k}\right) \tag{2a}$$

$$s.t. \sum_{k\in K} x_{ik} = 1, \forall i \in I \tag{2b}$$

$$\sum_{i\in I} x_{ik} \leqslant C, \forall k \in K \tag{2c}$$

$$x_{ik} = \{0, 1\}, \forall i \in I, \forall k \in K \tag{2d}$$

The objective 2a is to minimize the total time cost, including the maintenance time cost and travel time cost. As the number of faulty bikes is fixed, the maintenance time cost is a constant. Therefore, the objective is equivalent to minimizing the travel time cost. Constraint 2b ensures that every bike will be maintained, and constraint 2c gives the capacity budget of each truck, where C is the capacity of each truck. Constraint 2d ensures x_{ik} is a binary variable.

3.2. Dedicated policy (D)

In the dedicated policy, each maintenance truck is assigned to recycle only one type of bike. Similarly, the objective of the dedicated policy is to minimize total time cost. The dedicated model is formulated as follows: The dedicated policy model:

$$\min_{x_{ik}} \sum_{k\in K}\left(\sum_{i\in I}\left[T_0^D + f\left(\pi_i, b_i\right)T_1^D\right]x_{ik} + \frac{l_k}{v_k}\right) \tag{3a}$$

$$s.t. \sum_{k\in K} x_{ik} = 1, \forall i \in I \tag{3b}$$

$$\sum_{i\in I} x_{ik} \leqslant C, \forall k \in K \tag{3c}$$

$$f(\pi_i, b_i)x_{ik} \leqslant t_k, \forall i \in I, \forall k \in K \tag{3d}$$

$$[1 - f(\pi_i, b_i)]x_{ik} \leqslant 1 - t_k, \forall i \in I, \forall k \in K \tag{3e}$$

$$x_{ik} = \{0, 1\}, \forall i \in I, \forall k \in K \tag{3f}$$

The objective function 3a and constraint 3b, 3c and 3f are the same as the pooling policy. In constraint 3d and 3e, we have one more decision variable $t_k = \{0, 1\}$ to determine the type for each truck, where $t_k = 0$ for type 0 and $t_k = 1$ for type 1. Constraints 3d and 3e ensure that each truck can only recycle one type of bike.

The pooling policy and dedicated policy have their advantages and disadvantages. In the dedicated policy, restricting the maintenance type of each truck may result in more travel time costs because the trucks have to go farther for special bikes assigned to them. However, this dedicated assignment policy may improve the maintenance efficiency because the trucks only need to focus on one type of bike and will be more proficient. In our model, the improvement of maintenance efficiency is characterized by $T_0^D < T_0^P$ and $T_1^D < T_1^P$. Therefore, the choice of these two models is actually a trade-off between maintenance time cost and travel time cost. Further discussion about these efficiency gaps is provided in the case study, where a more general framework that incorporates both pooling and dedicated policy is built.

4. Analysis

4.1. Predicting health status

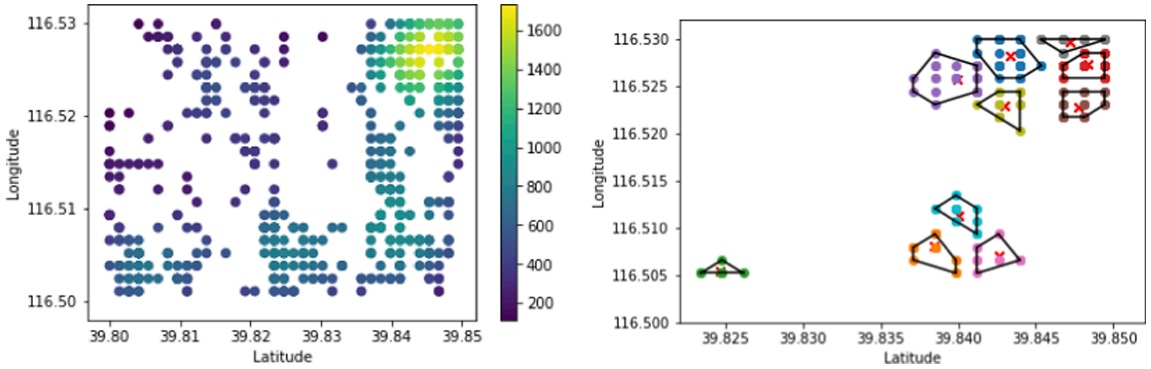
Recall that one of the challenges in our maintenance assignment problem is predicting the health status of each bike (the value of $f(\pi_i, b_i)$). We identify two important features, i.e., inactive days and black holes, to predict the health status of bikes. In this section, we introduce these two features and the prediction model in detail.

Data set. We have a real dataset of more than 480,000 bikes from a leading industry partner. For each bike, we have the data of the starting time, ending time, starting location, and ending location of each trip for 14 days. Each bike is labeled as healthy, type-0 faulty, or type-1 faulty (details are in the online supplement).

Inactive days. The "inactive days" π_i is the number of idle days from its latest use. In particular, trips of very short lengths (less than 120 meters) are regarded as invalid trips and do not influence the inactive days. Therefore, more inactive days imply a higher possibility for faulty bikes.

Black holes. To obtain the value of b_i, we need to depict the "black holes" first. A "black hole" can be imagined as a place where many faulty bikes are gathered. There are many reasons for "black holes" to form. For example, solar locks on the bicycles cannot be charged by sunlight in some areas (such as dead ends). Thus, if people park bikes in these areas, then these bikes will be faulty after a while. We aim to identify these "black holes" and use them as a key feature in predicting the health status of bikes. Moreover, with identification, we can even control these "black holes" with geofencing technology (a new technology that allows the company to control the parking area for bicycles by setting virtual boundaries). To depict the black holes, first, we use kernel density estimation (details are in the online supplement) to select faulty bicycles with high density. Next, we use a K-means clustering algorithm to identify high concentrations of faulty bikes; in particular, the k-means centers correspond to centers of "black holes." Finally, inspired by "geofencing" technology in industry practice, we identify the boundaries of "black holes" based on the convex hulls of the aforementioned clusters. Details for each step are below.

Since we have all locations of the faulty bikes, we can use kernel density analysis to find the places where the faulty bikes are most concentrated. These places are the "black holes" for which we are looking. After this, we plot a spatial distribution of faulty bicycles and



(a) Spatial distribution of faulty bicycles (kernel density estimation)

(b) Location of identified “black holes” using geofencing (convex hull)

Fig. 1. Generation of “geofences”.

Table 1
Comparison of prediction models.

Order	Model	Accuracy Score	Precision	Recall	F Score
1	Logistic Regression	0.92	0.92	0.92	0.92
2	K-Neighbors	0.86	0.86	0.86	0.86
3	Support Vector Classification	0.90	0.91	0.90	0.90
4	Desicion Tree	0.95	0.95	0.95	0.95
5	Bernoulli Naive Bayes	0.60	0.47	0.60	0.49
6	GBDT	0.96	0.96	0.96	0.96

Table 2
Black holes.

“Black holes”	Healthy	Type 0	Type 1	Faulty (Type 0 + Type 1)
Out	0.518	0.295	0.187	0.482
In	0.179	0.425	0.396	0.821

represent this information via kernel density estimation (details are given in the online supplement). Results are summarized in Fig. 1(a). The bicycle density increases from blue, green, to yellow, with a color bar indicating the scale of estimated density value.

Next, we sort these faulty bicycles by their density values and select 250 bicycles with the highest density. Then, we conduct a K-means clustering of these bicycles, and the clustering centers are marked by a red cross. Finally, we generate convex hulls for all clusters, which are the “black holes” that we will find [see Fig. 1(b)]. Each dot represents a 150 m-by-120 m area. The generated convex hulls can be used to predict the faulty bikes. As mentioned in the previous section, we define a binary variable b_i to denote whether bike i is in a “black hole.” If a bike i is parked in a “black hole” ($b_i = 1$), then it is more likely to be faulty, and this feature is captured in our prediction model. For further consideration, we can even create no-parking zones using geofencing technology to reduce the fault rate, which may be a direction for feature research.

Then, we can identify two features of bikes: the inactive days and whether the bikes are in “black holes.” To predict the health status of the bikes, we use a cross-validation method to evaluate different machine learning models. In our model, we divide bicycles into three statuses: healthy bikes that are in good condition, type 0 (minor faulty) bikes that are not yet completely damaged, and type 1 (fully faulty) bikes that are considered to be completely faulty. The results of different prediction models are listed in Table 1. From Table 1, we can see that the accuracy of most models is high, and Gradient Boosted Decision Trees (GBDT) (Hastie, Tibshirani, & Friedman, 2009) is the best one. Therefore, we choose GBDT as the prediction model. See Table 2 for illustrations of the prediction results. From Table 2 we can observe that the probability for bikes out of “black holes” to be faulty is 0.482, while it surges to 0.821 for bikes in “black holes,” indicating a 70% increase. This demonstrates the importance of the role of “black holes” in the prediction of faulty bikes. With the above prediction model, given the information of inactive days (π_i) and “black holes” (b_i) for each bike, we can predict the health status, i.e., $f(\pi_i, b_i)$, and integrate the results into our maintenance assignment problem.

4.2. Travel time prediction

Recall that the other challenge in the maintenance assignment problem is evaluating the travel time cost, or equivalently, the travel distance. The problem can be regarded as the Vehicle Routing Problem (VRP), which is computationally intractable. Because both the location of the bikes and each truck’s route behavior are different, it is difficult to model all practical constraints and behavior considerations. To overcome the computational difficulties, we use an end-to-end approach (by developing a machine learning model with some significant features) to approximate the travel distance of visiting all faulty bikes without specifying the visiting sequence. In this way, we do not need to solve a VRP problem whenever we want to evaluate the travel distance for a truck assignment solution. Instead, the travel distance estimator trained offline is used as a proxy to the VRP solution online.

The approximate formula of the traveling salesman problem (TSP) and the asymptotic results of VRP in various cases have been proposed. Assuming that bikes are distributed independently and evenly in a square zone with an area of A. The optimal TSP^* solution meets the following formula: (Beardwood, Halton, & Hammersley, 1959)

$$\lim_{n\to\infty}\frac{TSP^*}{\sqrt{n}} = \psi\sqrt{A} \tag{4}$$

where ψ is a constant, and n is the number of demand locations. Such an approximate formula requires a strong random assumption, and good results can only be obtained when n is sufficiently large(Shen & Qi, 2007). However, when collecting faulty bikes, a truck may not be able to get to so many destinations, which makes the approximate formula inappropriate. Thus, we will develop a prediction model of travel distance while keeping the computational tractability.

First, we need to choose appropriate prediction features. The main categories of features are the numeral feature (number of bikes), spatial features (area and distance), and interactive terms. On the basis of the approximation in Eq. 4, researchers proposed many uncomplicated regression models that contain prediction features such as the area and distance between the depot and locations. For example, $\bar{d}$ (mean distance between maintenance base and locations) and $R\sqrt{n-1}$ (R is the

Table 3
Test results for different models.

	OLS	Lasso	Ridge	Random Forest	SVR	Decision Tree
R2	0.998	0.998	0.998	0.986	0.998	0.909
F	29103.558	29039.152	29071.237	4068.116	26672.455	590.862
AIC	2076.896	2077.559	2077.228	2663.500	2103.009	3218.150
BIC	2099.118	2099.782	2099.451	2685.723	2125.232	3240.373
MSE	975.535	977.694	976.617	6893.498	1064.256	43790.291

* We apportion the data into training and test sets with a 7/3 split. Thus, the test result is derived from the model trained on the entire training set (i.e., 70% of the entire dataset). We tested other random splits and observed that the results might be slightly different, but the linear models (i.e., OLS, Lasso, and Ridge) always perform better than the other three nonlinear models. However, there is little difference between the three linear models, and they all might be best in a random split.

Table 4
Result of Lasso for feature selection.

Features	d	a	b	n	$\sqrt{n}$
Coefficient	0	1	0.47	0.01	0.02

* The coefficients are normalized. Lasso selects three features a, b, n and $\sqrt{n}$.

area of the smallest rectangle covering n positions) were proposed by Chien (1992). In the following, we present the definitions of basic features in our prediction models.

- n: Number of bikes
- d: Smallest distance among the maintenance base and bikes
- R: Area of the smallest rectangle that covers all bikes
- a: Largest latitude difference of two bikes
- b: Largest longitude difference of two bikes

However, not all features produce a tractable computational representation. For example, R can be computed as $a \times b$, but R does not produce a linear or convex representation, which presents a challenge to tractability. Moreover, since the interactive terms are difficult to linearize, we only select the features that are tractable. The tractable features contain d, a, b, n, and $\sqrt{n}$.

We then examine the prediction models with these tractable features. This involves several linear models [e.g., Ordinary Least Square (OLS), ridge, and LASSO regression] and piecewise linear models (e.g., decision tree). We sample a thousand groups of faulty bike locations from the dataset and calculate the optimal TSP results by Gurobi. Then, these results are used to fit six different models: Ordinary Least Squares (OLS) regression, ridge regression, Lasso regression, random forest, Support Vector Regression (SVR), and decision tree. Table 3 summarizes the performance evaluation metrics of the different prediction models on the test set.

Based on the test results, the linear models (OLS, Lasso, and Ridge regression) are better. Moreover, all three linear models work well, and their difference is very small. It is well-known that Lasso regression can produce sparse solutions so that it can find redundant features. Therefore, we use Lasso to select features and choose OLS regression as a representative prediction model.

The features selected by Lasso are shown in Table 4. The outcome of the OLS fitting is

$$l_k = -17.6273 + 204.6356 \times a_k + 97.1158 \times b_k + 1.6222 \times n_k + 6.1601 \times \sqrt{n_k} \tag{5}$$

4.3. Computational techniques

Since both models are second-order nonlinear programming problems, the computational complexity is too high. We need to transform the two models into tractable Second-Order Cone Programming (SOCP) to solve them. This section discusses the SOCP and transformation in detail.

Proposition 1. The basic model is equivalent to the following SOCP:

Pooling Policy:

$$\min_{x_{ik}} \sum_{k\in K}\left(\sum_{i\in I}\left[T_0^P + f(\pi_i, b_i)T_1^P\right]x_{ik} + \frac{l_k}{v_k}\right) \tag{6a}$$

$$s.t. \sum_{k\in K} x_{ik} = 1, \forall i \in I \tag{6b}$$

$$\sum_{i\in I} x_{ik} \leqslant C, \forall k \in K \tag{6c}$$

$$\sum_{i\in I} x_{ik}^2 \leqslant y_k^2, \forall k \in K \tag{6d}$$

$$x_{ik} = \{0,1\}, \forall i \in I, \forall k \in K \tag{6e}$$

where $l_k = \beta_0 + \beta_1 a_k + \beta_2 b_k + \beta_3 n_k + \beta_4 y_k$ and $n_k = \sum_{i\in I} x_{ik}$.

Dedicated Policy:

$$\min_{x_{ik}} \sum_{k\in K}\left(\sum_{i\in I}\left[T_0^D + f(\pi_i, b_i)T_1^D\right]x_{ik} + \frac{l_k}{v_k}\right) \tag{7a}$$

$$s.t. \sum_{k\in K} x_{ik} = 1, \forall i \in I \tag{7b}$$

$$\sum_{i\in I} x_{ik} \leqslant C, \forall k \in K \tag{7c}$$

$$f(\pi_i, b_i)x_{ik} \leqslant t_k, \forall i \in I, \forall k \in K \tag{7d}$$

$$[1 - f(\pi_i, b_i)]x_{ik} \leqslant 1 - t_k, \forall i \in I, \forall k \in K \tag{7e}$$

$$\sum_{i\in I} x_{ik}^2 \leqslant y_k^2, \forall k \in K \tag{7f}$$

$$x_{ik} = \{0,1\}, \forall i \in I, \forall k \in K \tag{7g}$$

where $l_k = \beta_0 + \beta_1 a_k + \beta_2 b_k + \beta_3 n_k + \beta_4 y_k$ and $n_k = \sum_{i\in I} x_{ik}$.

Proof of Proposition 1

First, we add new variables y_k, s.t. $\sqrt{n_k} \leqslant y_k$.

Since $n_k = \sum_{i\in I} x_{ik}$, we have $\sqrt{\sum_{i\in I} x_{ik}} \leqslant y_k$. Then, we can get the following constraint by squaring both sides of the equation:

$$\sum_{i\in I} x_{ik} \leqslant y_k^2 \tag{8}$$

Because $x_{ik} = \{0,1\}$, we have $x_{ik} = x_{ik}^2$, the constraint (8) equals the following form:

$$\sum_{i\in I} x_{ik}^2 \leqslant y_k^2 \tag{9}$$

Table 5
Impact of capacity C.

	n	k	v	C	Assignment	T	Cost
(1)	60	2	20	30	30,30	$\begin{pmatrix} 0.25 & 0.25 \\ 0.10 & 0.40 \end{pmatrix}$	37.51
(2)	60	2	20	32	32,28	$\begin{pmatrix} 0.25 & 0.25 \\ 0.10 & 0.40 \end{pmatrix}$	36.92
(3)	60	2	20	34	34,26	$\begin{pmatrix} 0.25 & 0.25 \\ 0.10 & 0.40 \end{pmatrix}$	36.35
(4)	60	2	20	36	36,24	$\begin{pmatrix} 0.25 & 0.25 \\ 0.10 & 0.40 \end{pmatrix}$	35.82
(5)	60	2	20	38	38,22	$\begin{pmatrix} 0.25 & 0.25 \\ 0.10 & 0.40 \end{pmatrix}$	35.25
(6)	60	2	20	40	40,20	$\begin{pmatrix} 0.25 & 0.25 \\ 0.10 & 0.40 \end{pmatrix}$	34.66

Table 6
Impact of maintenance cost **T**.

	n	k	v	C	Assignment	T	Cost
(1)	60	2	20	40	26,34	$\begin{pmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{pmatrix}$	39.52
(2)	60	2	20	40	40,20	$\begin{pmatrix} 0.2 & 0.3 \\ 0.25 & 0.25 \end{pmatrix}$	38.53
(3)	60	2	20	40	40,20	$\begin{pmatrix} 0.15 & 0.35 \\ 0.25 & 0.25 \end{pmatrix}$	37.53
(4)	60	2	20	40	40,20	$\begin{pmatrix} 0.1 & 0.4 \\ 0.25 & 0.25 \end{pmatrix}$	36.53
(5)	60	2	20	40	40,20	$\begin{pmatrix} 0.25 & 0.25 \\ 0.2 & 0.3 \end{pmatrix}$	38.16
(6)	60	2	20	40	40,20	$\begin{pmatrix} 0.25 & 0.25 \\ 0.15 & 0.35 \end{pmatrix}$	36.41
(7)	60	2	20	40	40,20	$\begin{pmatrix} 0.25 & 0.25 \\ 0.1 & 0.4 \end{pmatrix}$	34.66

Finally, we add the constraints (9) to the basic models and replace $\sqrt{n_k}$ in l_k formula with y_k. Thus, the Proposition is proven.

Proposition 1 reformulates the two models and provides tractable formulations. Then, the faulty bike maintenance problem can be solved by a Gurobi solver. In the following case study, we use a linear formulation of l_k to speed up the running rate and increase the tractable size. The OLS estimates the linear formula: $l_k = 0.1994 + 204.6116 \times a_k + 97.1377 \times b_k + 2.0917 \times n_k$. With this linear formulation of l_k, we can conduct extensive numerical experiments with real data.

5. Case study

In the numerical case study, we extend the basic model to consider that the cost of maintenance is related not only to the type but also the truck. This is a more general formulation that incorporates both pooling and dedicated policy. Specifically, our new maintenance costs are expressed as a matrix.

$$\mathbf{T} = \begin{pmatrix} T_{01} & T_{02} & \cdots & T_{0k} \\ T_{11} & T_{12} & \cdots & T_{1k} \end{pmatrix} \tag{10}$$

where T_{0k} is the maintenance cost of repairperson k riding in truck k for type-0 bikes, and T_{1k} is the maintenance cost of repairperson k riding in truck k for type-1 bikes. When the difference between the maintenance costs for different drivers is very large, which implies a high degree of specialization, this formulation degenerates to the dedicated policy. In other words, the dedicated policy is an extreme case of this general formulation, which allows us to evaluate a continuous spectrum of policies from the most dedicated ones to the pooling policy. Based on this general formulation, we investigate the impact of several important parameters and the prediction accuracy.

We consider three parameters that may have an important impact on the maintenance strategy, i.e., the truck's capacity C, maintenance costs **T**, and distribution of faulty bikes. We conduct several numerical experiments and make some interesting observations.

Capacity. We first consider the impact of the capacity C for trucks. We select 60 bikes as a sample from the dataset and assume two trucks in the numerical experiments. The truck capacity C varies from 30 to 40, and the maintenance cost **T** remains unchanged. The total cost is calculated for different C, and results are presented in Table 5.

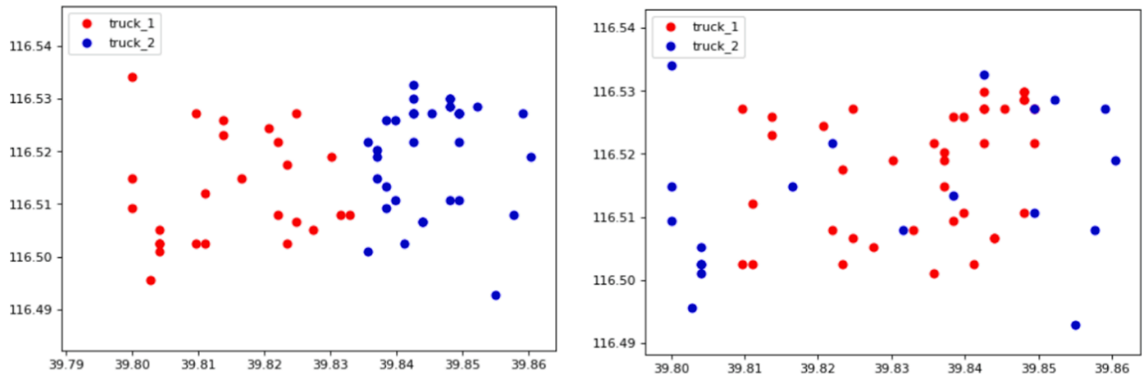
Observation 1. The total cost is decreasing with regard to truck capacity C.

This observation is easy to understand. Increasing the capacity of trucks allows more freedom for assignments so that the final cost is no more than the original cost.

Maintenance cost. Next, we consider the impact of the maintenance cost **T**. We use the same sample of 60 bikes and two trucks, where the capacity C is assumed to be 40. Therefore, in this numerical experiment, the maintenance cost **T** is a 2×2 matrix. We keep $T_{11} + T_{12}$ and $T_{21} + T_{22}$ constant and change their values to see the impact of maintenance cost differentiation. Table 6 shows the results for different structures of **T**.

Observation 2. If we keep $T_{11} + \cdots + T_{1k}$ constant, then as $\Delta f_1 = \max_k\{T_{1k}\} - \min_k\{T_{1k}\}$ increases, the optimal maintenance strategy will gradually prefer the dedicated strategy, and the total cost will decrease.

Observation 2 indicates that the dedicated strategy is more likely to be preferred as the gap of maintenance costs between trucks becomes large. Δf_1 characterizes the degree of truck differentiation. A high Δf_1 results in a significant difference in cost for choosing different trucks, implying a high degree of specialization in maintenance jobs. Inversely, the pooling strategy is preferred when the trucks are not specialized



(a) $\mathbf{T} = \begin{pmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{pmatrix}$ (b) $\mathbf{T} = \begin{pmatrix} 0.25 & 0.25 \\ 0.2 & 0.3 \end{pmatrix}$

Fig. 2. Assignments for different **T**.

Table 7
Comparison of two distributions.

Distribution	n(type0, type1)	k	v	T	Policy	Cost
two-dimensional Gaussian distribution	60 (28,32)	2	10	$\begin{pmatrix} 0.25 & 0.25 \\ 0.15 & 0.35 \end{pmatrix}$	Dedicated	49.46
Gaussian mixture distribution	60 (28,32)	2	10	$\begin{pmatrix} 0.25 & 0.25 \\ 0.15 & 0.35 \end{pmatrix}$	Pooling	48.90

Table 8
Result of two distributions.

	n	k	C	T	two-dimensional Gaussian distribution	Gaussian mixture distribution
(1)	60	1	60	$\begin{pmatrix} 0.25 \\ 0.25 \end{pmatrix}$	36.53	37.03
(2)	60	2	30	$\begin{pmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{pmatrix}$	36.71	37.20
(3)	60	3	20	$\begin{pmatrix} 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 \end{pmatrix}$	36.91	37.10

enough. See Fig. 2 as an illustration. Another observation of Table 6 is that the total cost decreases as the gap of costs increases. This result demonstrates the benefits of developing dedicated trucks. A simple and efficient assignment strategy can be easily obtained by assigning more type-0 bikes to the trucks that specialize in type-0 bikes (basic maintenance cost T_{0k} is low) and more type-1 bikes to the trucks that specialize in type-1 bikes (extra maintenance cost T_{1k} is low).***Distribution of faulty bikes.*** As travel time cost is an important part of the total cost, it is related to the routing problem, and the distribution of faulty bikes significantly affects the travel time. In the numerical experiments, we focus on two types of distributions, i.e., two-dimensional Gaussian distribution and Gaussian mixture distribution. We obtain the following interesting observation from the experiments. **Observation 3.** In the two-dimensional Gaussian distribution, the dedicated strategy is more likely to be preferred, while in the Gaussian mixture distribution, the pooling strategy is preferred.

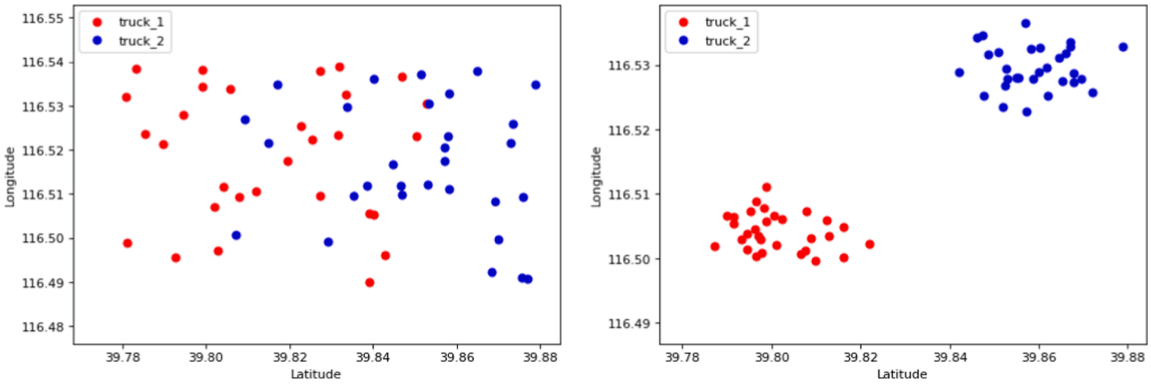
Table 7 lists the results, and Fig. 3 is an illustration of the optimal maintenance strategy. Obviously, the geographical distribution of bikes has a significant impact on choosing assignment strategies. Generally, if the bikes' distribution is more uniform, then maintaining bikes type by type can reduce the maintenance cost, and the dedicated strategy is likely better. However, the pooling strategy is more likely to be preferred when the bikes are distributed in several clusters, where the travel time cost between clusters is significant.

Observation 4. If we keep $k \times C$ (total capacity of trucks) constant, then as k increases, the total cost increases in the two-dimensional Gaussian distribution, while the total cost may decrease in the Gaussian mixture distribution.

Observation 4 reflects two patterns of recycling: centralized and decentralized. The decentralized pattern refers to using more trucks to complete the maintenance assignment problem, resulting in an average and fewer tasks for each truck. The centralized pattern needs every truck to recycle as many faulty bikes as possible so that we can reduce the number of trucks. The result of an example for two-dimensional Gaussian distribution is shown in Table 8 and Fig. 4(a). We find that the distribution of bicycles is relatively uniform without obvious clusterings. Therefore, using more trucks is not very efficient, so the total cost will decrease as k increases. However, the result of Gaussian mixture distribution is quite different. The total cost when k equals three is less than the total cost when k equals two. As we can see from Fig. 4 (b), the bicycles are divided into three clusters so that assigning three trucks is more appropriate and efficient. In summary, the insight from observation 4 is that with limited total capacity, the impact of the number of trucks is dependent on the distribution of faulty bikes. The decentralized pattern is more likely to be preferred when faulty bikes are distributed in several clusters, while the centralized pattern outperforms when faulty bikes are distributed uniformly.

Prediction. In our maintenance model, the prediction of bikes' health status is a key process. The accuracy of our prediction influences the efficiency of our maintenance assignment strategy. In this subsection, we study the influence of prediction accuracy. We divide the experiments into three cases. The first case is 100% accuracy, and we take it as the real case. The second case is that we do not predict, in which case we treat both types of bicycles as the same, and the maintenance cost is equal to a weighted average of two types of bicycles. This case can be regarded as using practical experience instead of data-driven prediction. The third case is that we make predictions with different accuracies. The results are listed in Table 9 and Fig. 5. Based on the predictions with different accuracies, we solve for the corresponding optimal maintenance assignment strategies (column 6). Then, we plug these assignment strategies into the real case and calculate the actual cost for each prediction accuracy (column 7).

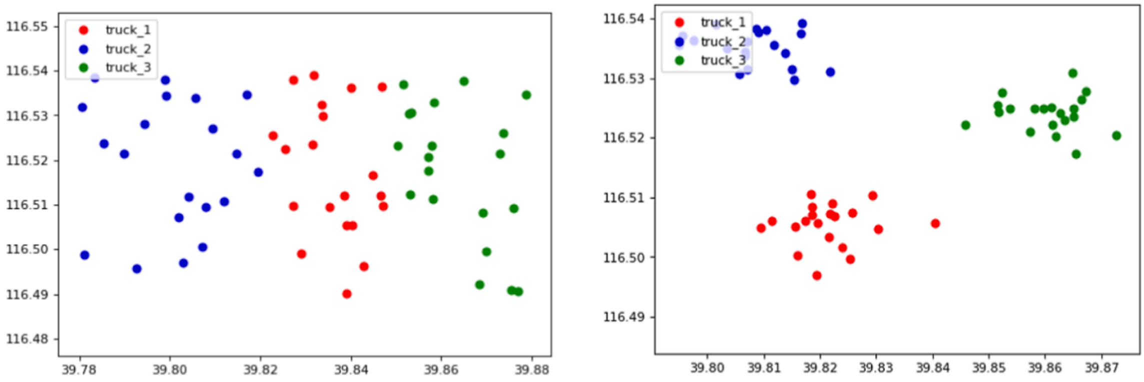
We use 60 faulty bikes (15 type-0 bikes and 45 type-1 bikes) in the numerical experiments to investigate the impact of prediction accuracy. The maintenance truck's capacity is 40 and maintenance cost matrix $\mathbf{T} = \begin{pmatrix} 0.25 & 0.25 \\ 0.40 & 0.10 \end{pmatrix}$, where T_{0k} is the maintenance cost of repairperson k riding in truck k for type 0 bikes, and T_{1k} is the maintenance cost of repairperson k riding in truck k for type 1 bikes. If we know the true types of the 60 faulty bikes, then we will assign as many type-1 bikes to truck 2 as possible (because it costs less for truck 2 to repair type-1



(a) two-dimensional Gaussian distribution

(b) Gaussian mixture distribution

Fig. 3. Results of two distributions.



(a) two-dimensional Gaussian distribution (b) Gaussian mixture distribution

Fig. 4. Example for observation 4.

Table 9
Influence of prediction accuracy.

Accuracy	Predicted Truck 1's Assignment	Predicted Truck 2's Assignment	Actual Truck 1's Assignment	Actual Truck 2's Assignment	Predicted Cost	Actual Cost	Cost Gap
No Prediction	20	40	10,30	10,30	-	37.28	-
50%	20,0	9.92,30.08	3.82,16.18	11.18,28.82	31.55	37.90	20%
60%	20,0.02	6.46,33.52	5.27,14.75	9.73,30.25	32.00	37.56	17%
70%	19.90,0.10	4.49,35.51	7.93,12.07	7.07,32.93	32.28	36.81	14%
80%	19.46,0.56	1.58,38.40	11.16,8.86	3.84,36.14	32.82	35.92	9%
90%	17.7,2.3	0.14,39.86	14.07,5.93	0.93,39.07	33.64	35.01	4%
95%	16.4,3.6	0,40	14.64,5.36	0.36,39.64	34.14	34.81	2%
Correct (100%)	15,5	0,40	15,5	0,40	34.66	34.66	0%

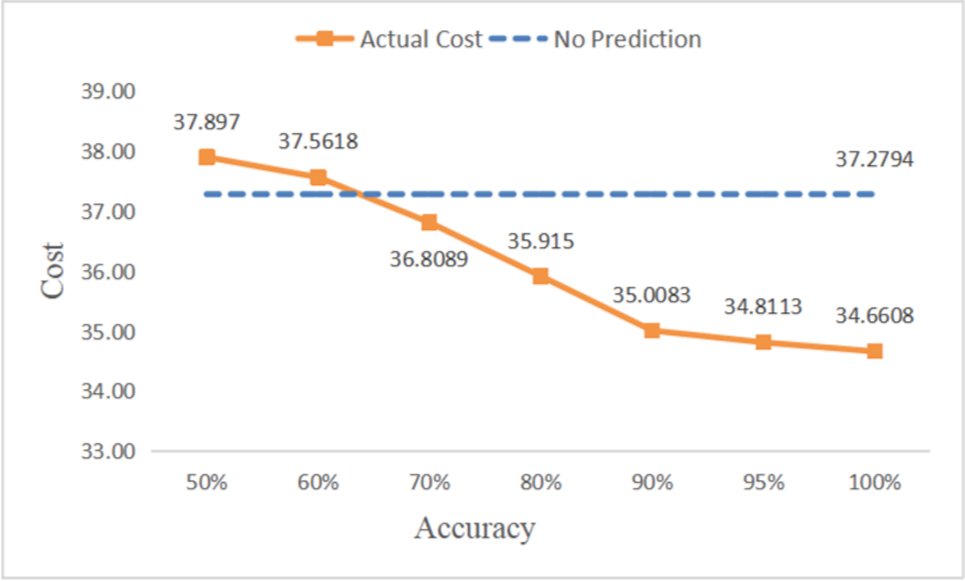
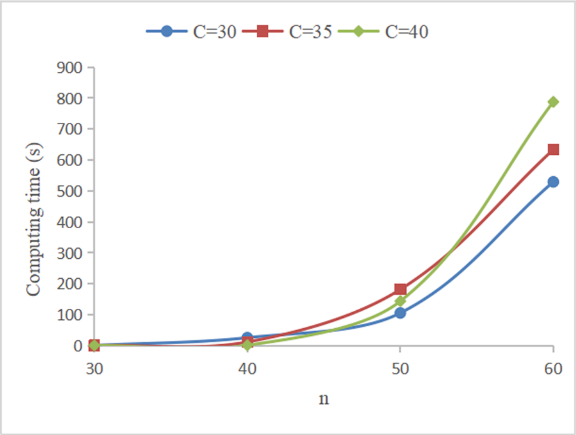


Fig. 5. Influence of prediction accuracy.

bikes). Therefore, the optimal plan with 100% accuracy is assigning 40 type-1 bikes to truck 2 and the other 20 faulty bikes (15 type-0 bikes and 5 type-1 bikes) to truck 1. The two-tuples in Table 9 are the numbers of type-0 and type-1 bikes. Columns 2 and 3 are the optimal assignment plan based on the prediction result. Columns 4 and 5 are the true results of the predicted assignment. For each prediction accuracy, we use 100 sets of data to do the experiments and take the average. The lower the prediction accuracy, the larger the gap between the actual cost and predicted cost.

From Fig. 5, we find that when the prediction accuracy reaches more than about 65%, the result of making the prediction is better than that of not making the prediction. Thus, as long as we can obtain a 65% accuracy prediction, our data-driven model can outperform practical experience and help to reduce the total maintenance cost. Since even a random guess can achieve 50% accuracy, a 65% accuracy prediction is easy to reach. Therefore, our model is quite applicable in practice.



n	k	C	t_{avg}(s)	t_{lb}(s)	t_{ub}(s)
		30	0.68	0.52	0.88
30	2	35	0.69	0.52	0.96
		40	0.85	0.56	2.43
		30	25.78	19.03	30.03
40	2	35	11.96	8.51	14.92
		40	1.44	0.86	3.48
		30	105.66	89.06	126.94
50	2	35	181.68	111.90	270.18
		40	143.84	108.63	198.09
		30	529.33	502.53	603.56
60	2	35	633.59	597.28	700.12
		40	787.73	750.23	846.38

* t_{avg}, t_{lb} and t_{ub} denotes the average, the best and the worst computing time in 50 runs.

Fig. 6. Computing time of SOCP.

Computation time The original problem is a nonconvex quadratic problem that is infeasible. Our contribution is to transform the intractable nonconvex quadratic problem into a SOCP that is tractable. The experimental results about the computing time of SOCP are shown in Fig. 6.

According to our experiments, the SOCP model is efficient when $k \leqslant 2$ and $n \leqslant 60$. Within this range, we change C and n and then record the results of the computing time. From Fig. 6, the computing time increases exponentially with n. There is no obvious rule about C.

6. Conclusion

In this paper, we constructed a framework that integrates a prediction model for faulty bikes and travel distance predictors with maintenance assignment optimization to solve the faulty bike maintenance problem in bike-sharing systems. The main contributions and key findings are summarized as follows.

First, inspired by industry practice, we identified the role of “black holes” in the prediction of faulty bikes: locations with morbidly high faulty rates, which can be characterized using data-driven approaches (clustering and convex hull). We demonstrated the importance of “black holes” in the prediction of faulty bikes. By applying our prediction model to more than 480,000 bikes with real data, we found that bikes in “black holes” are about 70% more likely to be faulty than those out of “black holes.” This new angle of view for faulty bike prediction helped us to improve the prediction accuracy as well as the efficiency of the faulty bike maintenance problem.

Second, we identified the main trade-off between travel efficiency and maintenance efficiency in the maintenance assignment problem. We proposed two different assignment policies, i.e., the pooling policy and the dedicated policy, to minimize the total cost. The comparison of our pooling policy and dedicated policy helped to identify the optimal assignment strategy. We found that the maintenance cost differentiation and faulty bike distributions have significant impacts on the optimal policy: (1) The dedicated policy is preferred when the specialization degree is high, while the pooling policy is preferred in low-specialization situations. (2) The dedicated policy is preferred when faulty bikes are distributed more uniformly, while the pooling policy is better off when faulty bikes are distributed in clusters.

Third, we used an end-to-end approach to approximate the travel distance for each truck instead of solving the routing problem for every possible assignment. Based on this approximation, we can reformulate the two models as tractable Second-order Cone Programming (SOCP) that can be solved efficiently by the Gurobi solver. Using this reformulation, we conducted numerical experiments. The case study emphasizes the importance of health status prediction in the maintenance problem. With perfect prediction, a more than 7% cost reduction can be achieved. Moreover, the cost reduction due to prediction is significant even when the prediction is imperfect (in our case, our data-driven model performs better as long as the prediction accuracy exceeds 65%).

However, there are still some limitations to this study. The classification of two faulty types, i.e., type 0 and type 1, is a simple approximation of different levels of damage. If more detailed data of faulty bikes can be obtained, then more accurate classifications of the faulty bikes in the basic model can improve the prediction accuracy. Moreover, although our numerical results show that even under imperfect prediction, the maintenance assignment of our model can help to reduce costs significantly, and inaccurate prediction can be addressed in the framework of a distributionally robust optimization (DRO) model which considers the uncertainty in predicting. The DRO model can be a promising future research direction to deal with inaccurate prediction and build a robust maintenance assignment process. Finally, we discovered from practice that bike impoundment is a special kind of “black hole” that is widely present in the operation of bike sharing. The Traffic Enforcement Department impounds bikes that accumulate on the roadside and sends them to undisclosed locations (“black holes”). The bike-sharing company then needs to find these impound locations before paying the required fines to release the impounded bikes.

CRediT authorship contribution statement

Lan Lu: Methodology, Writing-original draft. Shichen: Formal analysis, Writing-original draft, Writing-review & editing. **Qiao-Chu He:** Conceptualization, Methodology, Funding acquisition, Investigation, Project administration, Resources, Writing-review & editing. **Ning Zhu:** Supervision, Funding acquisition, Writing-review & editing.

Acknowledgements

The authors sincerely thank the editors and the anonymous reviewers for their detailed comments and many valuable suggestions. This work was partially supported by NSFC (National Natural Science Foundation of China) grants 72071101 and Shenzhen Natural Science Fund (Grant No. GXWD20201230110313001, Program Contract No.

20200925160442005).

References

Abbou, A., & Makis, V. (2019). Group maintenance: A restless bandits approach. *INFORMS Journal on Computing, 31*(4), 719–731.

Beardwood, J., Halton, J. H., & Hammersley, J. M. (1959). The shortest path through many points. In *Mathematical Proceedings of the Cambridge Philosophical Society* (Vol. 55, pp. 299–327). Cambridge University Press.

Benjaafar, S., & Hu, M. (2020). Operations management in the age of the sharing economy: what is old and what is new? *Manufacturing & Service Operations Management, 22*(1), 93–101.

Bocchetti, D., Giorgio, M., Guida, M., & Pulcini, G. (2009). A competing risk model for the reliability of cylinder liners in marine diesel engines. *Reliability Engineering & System Safety, 94*(8), 1299–1307.

Bruck, B. P., Cruz, F., Iori, M., & Subramanian, A. (2019). The static bike sharing rebalancing problem with forbidden temporary operations. *Transportation Science, 53*(3), 882–896.

Cai, X., Wu, X., & Zhou, X. (2009). Stochastic scheduling subject to preemptive-repeat breakdowns with incomplete information. *Operations Research, 57*(5), 1236–1249.

Campbell, A. M., & Thomas, B. W. (2008). Probabilistic traveling salesman problem with deadlines. *Transportation Science, 42*(1), 1–21.

Cappanera, P., & Scutellà, M. G. (2015). Joint assignment, scheduling, and routing models to home care optimization: A pattern-based approach. *Transportation Science, 49*(4), 830–852.

Çelebi, D., Yörüsün, A., & Işık, H. (2018). Bicycle sharing system design with capacity allocations. *Transportation Research Part B: Methodological, 114*, 86–98.

William Chien, T. (1992). Operational estimators for the length of a traveling salesman tour. *Computers & Operations Research, 19*(6), 469–478.

de Chardon, C. M., & Caruso, G. (2015). Estimating bike-share trips using station level data. *Transportation Research Part B: Methodological, 78*, 260–279.

Dell'Amico, M., Iori, M., Novellani, S., & Subramanian, A. (2018). The bike sharing rebalancing problem with stochastic demands. *Transportation Research Part B: Methodological, 118*, 362–380.

Dewilde, T., Cattrysse, D., Coene, S., Spieksma, F. C. R., & Vansteenwegen, P. (2013). Heuristics for the traveling repairman problem with profits. *Computers & Operations Research, 40*(7), 1700–1707.

Du, M., Cheng, L., Li, X., & Tang, F. (2020). Static rebalancing optimization with considering the collection of malfunctioning bikes in free-floating bike sharing system. *Transportation Research Part E: Logistics and Transportation Review, 141*, 102012.

Erera, A. L., Morales, J. C., & Savelsbergh, M. (2010). The vehicle routing problem with stochastic demand and duration constraints. *Transportation Science, 44*(4), 474–492.

Gendreau, M., Laporte, G., & Séguin, R. (1996). Stochastic vehicle routing. *European Journal of Operational Research,, 88*(1), 3–12.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer Science & Business Media.

He, L., Hu, Z., & Zhang, M. (2020a). Robust repositioning for vehicle sharing. *Manufacturing & Service Operations Management, 22*(2), 241–256.

He, P., Zheng, F., Belavina, E., & Girotra, K. (2020b). Customer preference and station network in the london bike-share system. *Management Science*.

Hu, J., Sun, Q., & Ye, Z.-S. (2020). Condition-based maintenance planning for systems subject to dependent soft and hard failures. *IEEE Transactions on Reliability*.

Huang, W., & Askin, R. G. (2003). Reliability analysis of electronic devices with multiple competing failure modes involving performance aging degradation. *Quality and Reliability Engineering International, 19*(3), 241–254.

Jaillet, P., Qi, J., & Sim, M. (2016). Routing optimization under uncertainty. *Operations Research, 64*(1), 186–200.

Kabra, A., Belavina, E., & Girotra, K. (2019). Bike-share systems: Accessibility and availability. *Management Science*.

Klapp, M. A., Erera, A. L., & Toriello, A. (2018). The one-dimensional dynamic dispatch waves problem. *Transportation Science, 52*(2), 402–415.

Laporte, G., Louveaux, F., & Mercure, H. (1992). The vehicle routing problem with stochastic travel times. *Transportation Science, 26*(3), 161–170.

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.

Liu, S., He, L., & Shen, Z.-J. M. (2018). On-time last mile delivery: Order assignment with travel time predictors. *Forthcoming in Management Science*.

Liu, X., Li, J., Al-Khalifa, K. N., Hamouda, A. S., Coit, D. W., & Elsayed, E. A. (2013). Condition-based maintenance for continuously monitored degrading systems with multiple failure modes. *IIE Transactions, 45*(4), 422–435.

Luo, Z., Qin, H., & Lim, A. (2014). Branch-and-price-and-cut for the multiple traveling repairman problem with distance constraints. *European Journal of Operational Research, 234*(1), 49–60.

Neumann-Saavedra, B. A., Crainic, T. G., Gendron, B., Mattfeld, D. C., & Omer, M. R. (2020). Integrating resource management in service network design for bike-sharing systems. *Transportation Science*.

Pal, A., & Zhang, Y. (2017). Free-floating bike sharing: Solving real-life large-scale static rebalancing problems. *Transportation Research Part C: Emerging Technologies, 80*, 92–116.

Qin, M., Wang, J., Chen, W. M., & Wang, K. (2021). Reducing Co_2 emissions from the rebalancing operation of the bike-sharing system in Beijing. *Frontiers of Engineering Management, 5*.

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.

Restrepo, M. I., Semet, F., & Pocreau, T. (2019). Integrated shift scheduling and load assignment optimization for attended home delivery. *Transportation Science, 53*(4), 1150–1174.

Shen, Z.-J. M., & Qi, L. (2007). Incorporating inventory and routing costs in strategic location models. *European Journal of Operational Research, 179*(2), 372–389.

Shu, J., Chou, M. C., Liu, Q., Teo, C.-P., & Wang, I.-L. (2013). Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems. *Operations Research, 61*(6), 1346–1359.

Sleptchenko, A., & Eric Johnson, M. (2015). Maintaining secure and reliable distributed control systems. *INFORMS Journal on Computing, 27*(1), 103–117.

Tang, D., Yu, J., Chen, X., & Makis, V. (2015). An optimal condition-based maintenance policy for a degrading system subject to the competing risks of soft and hard failure. *Computers & Industrial Engineering, 83*, 100–110.

Tulabandhula, T., & Rudin, C. (2014). On combining machine learning with decision making. *Machine Learning, 97*(1–2), 33–64.

Wang, X., Agatz, N., & Erera, A. (2018). Stable matching for dynamic ride-sharing systems. *Transportation Science, 52*(4), 850–867.

Ye, Z.-S., Xie, M., Tang, L.-C., & Shen, Y. (2012). Degradation-based burn-in planning under competing risks. *Technometrics, 54*(2), 159–168.

Zhang, D., Yu, C., Desai, J., Lau, H. Y. K., & Srivathsan, S. (2017). A time-space network flow approach to dynamic repositioning in bicycle sharing systems. *Transportation Research Part B: Methodological, 103*, 188–207.