Effective Recycling Planning for Dockless Sharing Bikes

Cong Zhang
Beijing Uni. of Posts and Tele.
cong1126@bupt.edu.cn

Yanhua Li
Worcester Polytechnic Institute, USA
yli15@wpi.edu

Jie Bao
JD Finance
baojie@jd.com

Sijie Ruan
Xidian University
ruansijie@jd.com

Tianfu He
Harbin Institute of Technology
Tianfu.D.He@outlook.com

Hui Lu, Zhihong Tian
Guangzhou University
{luhui,tianzhihong}@gzhu.edu.cn

Cong Liu
The University of Texas at Dallas
cong@utdallas.edu

Chao Tian
Tencent
astortian@tencent.com

Jianfeng Lin, Xianen Li
Mobike
{linjianfeng,lixianen}@mobike.com

ABSTRACT

Bike-sharing systems become more and more popular in the urban transportation system, because of their convenience in recent years. However, due to the high daily usage and lack of effective maintenance, the number of bikes in good condition decreases significantly, and vast piles of broken bikes appear in many big cities. As a result, it is more difficult for regular users to get a working bike, which causes problems both economically and environmentally. Therefore, building an effective broken bike prediction and recycling model becomes a crucial task to promote cycling behavior. In this paper, we propose a predictive model to detect the broken bikes and recommend an optimal recycling program based on the large scale real-world sharing bike data. We incorporate the realistic constraints to formulate our problem and introduce a flexible objective function to tune the trade-off between the broken probability and recycled numbers of the bikes. Finally, we provide extensive experimental results and case studies to demonstrate the effectiveness of our approach.

CCS CONCEPTS

• **Applied computing** → *Transportation*; *Forecasting*; *Transportation*; • **Information systems** → *Spatial-temporal systems.*

KEYWORDS

bike-sharing systems, predictive model, optimal recycling program

ACM Reference Format:
Cong Zhang, Yanhua Li, Jie Bao, Sijie Ruan, Tianfu He, Hui Lu, Zhihong Tian, Cong Liu, Chao Tian, and Jianfeng Lin, Xianen Li. 2019. Effective Recycling Planning for Dockless Sharing Bikes. In *SIGSPATIAL '19: 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, November 5–8, 2019, Chicago, Illinois, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3347146.3359340

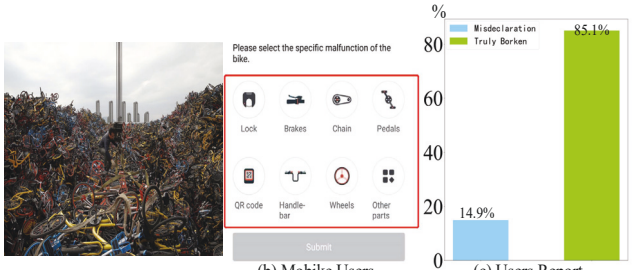
SIGSPATIAL '19, November 5–8, 2019, Chicago, Illinois, USA

ACM ISBN 978-1-4503-6909-1/19/11…$15.00
https://doi.org/10.1145/3347146.3359340

1 INTRODUCTION

Bike-sharing system is a popular transportation system in modern cities, as it not only provides an environment friendly choice for short-distance travelling, but also eases the traffic congestion. Currently, there are over 1,000 deployed bike-sharing systems world wide, and more than 300 systems are in the progress of deployment [29]. In recent years, station-less bike-sharing services, like Mobike[1], which allow users to pick up and drop off bikes at any locations they want, become more popular.

Due to the sharing nature of the bike-sharing systems, the sharing bikes have much higher broken possibilities compared with private bikes due to the high ridden frequency and open-air parking problem. For example, the bike sharing system in New York saw 3.6 daily rides per bike [2]. As a result, as shown in Figure 1(a), thousands of broken station-less sharing bikes are being kept in a bike graveyard.



(a) Bike Graveyards (b) Mobike Users Malfunction Report (c) Users Report Fault Bike Review

Figure 1: Issues with broken Sharing Bike.

Since the number of bikes put in the market is limited, without the proper maintenance, the number of bikes in good condition is continuously decreasing. The broken bikes not only cause economic losses to the companies but also lead to environmental pollution. Therefore, an effective bike recycling plan should be conducted. Currently, Mobike develops a broken bike report function in the app, so that the broken bikes can be discovered in a crowdsourcing way. As shown in Figure 1(b), users can report different types of bike problems in the mobile app, so that the company can arrange workers to collect and recycle them.

[1]https://en.wikipedia.org/wiki/Mobike
[2]https://bit.ly/2T6q5SE

However, there are *three* challenges to conduct such a broken bike recycling task:

Inaccurate and Inadequate Labels. Though the broken bike report function can help the company to quickly locate the broken bike, the report cannot be fully trusted. As shown in Figure 1(c), we manually exam the status of the reported broken bikes. Only 85.1% bikes are truly broken. Furthermore, not all of the users are willing to report the broken status of the bikes, as the broken report function is not a required step.

Arbitrary Spatial Distribution. Different from the station-based systems, the parking location of each individual station-less bike is totally arbitrary, which makes the recycling routes vary from day to day.

Limited Recycle Capacity. Given a set of bikes to be recycled, the worker can only collect the limited number of broken bikes within the working hour. Besides, the capacity of the collecting vehicle is limited, and the worker has to drive back to the recycling site as soon as the vehicle is full of broken bikes.

In this paper, we design a broken bike recycling route planning system for the worker. This system consists of two main modules: 1) broken bike inference, which infers the broken probability of each sharing bike using its inherent characteristics and the user trajectories associated with it; and 2) recycling route planning, which plans multiple closed recycling routes for the worker to conduct in each day.

The contributions of the paper are summarized as follows:

(1) We propose a novel broken sharing bike recycling problem, which takes the broken probability, working time constraint, and vehicle capacity into consideration.

(2) We build a broken bike inference model using inherent features and trajectory features extracted from the sharing bike so that the status of every single bike can be accurately inferred.

(3) We propose a scatter search-based heuristic algorithm for the broken sharing bike recycling problem.

(4) Experiments show the recycling efficiency of the broken bikes recommended by scatter search algorithm is 2.5 times that of the regional random search method and 1.5 times that of the Nearest neighbor routing search method. At the same time, the result of the algorithm is twice the efficiency of Mobike employees' broken bikes recycling.

The rest of the paper is organized as follows: Section 2 describes the problem and the system overview. Broken sharing bike inference model is discussed in Section 3. Section 4 gives the solution of broken sharing bike recycling routing problem. Experiments and case studies are given in Section 5. Related works are summarized in Section 6. Section 7 concludes the paper.

2 OVERVIEW

In this section, we define the broken prediction and recycling routing problem for Sharing Bike, and outline our solution framework.

2.1 Preliminaries

We define p_i as the inferred broken probability of sharing bike b_i. In the recycling task, we only consider bikes which are inferred as broken, i.e., $p_i > 0.5$. The bike with high broken probability is preferred to collect in the priority given limited working time. However, the bikes with high broken probability can distribute unevenly in the given region, which introduces large traveling time, and finally leads to less number of broken bike collected. As a result, we define a beneficial score $score_i$ below to characterize the worthiness of collecting a particular bike b_i. In the broken bike recycling mission, the dockless sharing bike can be at any location in the city, e.g., hiding in the residential area or close to the road network, where the parking location of collecting vehicles is usually along with the road network. As a result, the distance between them varies significantly, which we define v_w (walking speed) and rt (registration time) below to better characterize the individual bike collecting events.

Definition 1. ***(Beneficial score)*** *$score_i$ captures the overall benefit to recycle bike b_i, which characterizes the trade-off between the broken likelihood and the recycling cost of b_i.*

$$score_i = \alpha^{\frac{p_i}{\min p}} \qquad \alpha \geq 1 \tag{1}$$

where the parameter α represents the trade-off preference on the broken probability p_i vs recycling cost. $\min p$ is the minimum broken probability over all the bike in the region, which serves as a normalization term.

Each bike b_i has a broken probability p_i, i.e., the likelihood of being a broken bike. In practice, the trade off when choosing a bike is: If we seek for only bikes of high broken probability p_i, we may end up with a small number of bikes collected (less efficient); on the other hand, if we seek for a large number of collected bikes, many bikes collected may not be broken (false positive). The beneficial score defined in definition 1 captures such a trade-off by the parameter α. The reason for designing a score function using the exponential function is that the bike with higher broken probability will have a higher score ($\alpha > 1$). When α is close to 1, the efficiency is highly considered, leading to a large number of collected bikes; on the other hand, when $\alpha \gg 1$ is large, the broken probability p_i is highly considered, thus only bikes with high p_i will be collected. Especially, $\alpha = 1$ means that we do not care about the broken probability of the bike, and every broken bike has the same beneficial score. The α is a tunable parameter (chosen by the service operators), which provides them the flexibility between the efficiency (i.e., the number of collected bikes) and the likelihood of the collected bike being broken. From the operator's perspective, there are different objectives under various circumstances, for example, in regions hard to access, the efficiency should be highly considered (i.e., choosing α close to 1), while in areas with bikes densely populated, e.g., downtown, accurately collecting each broken bike is preferred, thus the likelihood of broken bikes needs to be considered more (i.e., choosing a large α). As a result, the beneficial score measures the practical "benefit" of collecting each bike.

Definition 2. ***(Sub-route)*** *Each closed route, which starts and ends at the collection site s, is considered as a sub-route.*

Definition 3. ***(Time Cost)*** *The time cost of sub-route R_j is composed of the vehicle travelling time between consecutive locations and the visiting time at each broken bike. Given a sub-route $R_j = s \rightarrow b_{r_1} \rightarrow \cdots \rightarrow b_{r_n} \rightarrow s$, the time cost T_j is calculated as follows:*

$$T_j = T_{travel}(R_j) + \sum_{i=1}^{n} T_{visit}(b_{r_i}). \tag{2}$$

Let us denote the shortest road network distance between broken bike b_i and b_j as $dist(b_i, b_j)$, and the vehicle driving speed as v_d, then the travelling time cost is calculated as follows:

$$T_{travel}(R_j) = \frac{dist(s, b_{r_1}) + \sum_{i=1}^{n-1} dist(b_{r_i}, b_{r_{i+1}}) + dist(b_{r_n}, s)}{v_d}. \tag{3}$$

The broken bike visiting time includes the walking time between the vehicle on the main road and the location of the broken bike, and the broken bike registration time rt. We denote the walking speed as v_w, and the perpendicular distance of the broken bike b_i to the nearest road segment as $shift_i$, then the visiting time cost can be represented as

$$T_{visit}(b_{r_i}) = \frac{2shift_{r_i}}{v_w} + rt. \tag{4}$$

Problem Definition. Given the road network RN, driving speed v_d, walking speed v_w, collection site s, broken bike registration time rt, working hour T, vehicle capacity M, and a broken sharing bike distribution graph $G = (V, E)$. The vertex set $V = \{b_1, b_2, \cdots, b_n\}$ represents all the broken bikes in the given service region of s, each of which is associated with a spatial location and a collection score $score_i$, and the edge set E denote the road network connectivity of broken bike pairs.

The objective of the broken bike recycling route planning problem aims to plan multiple traveling routes for the worker, so that the total score collected is maximized. The recycling route planning problem fulfills three constraints: (1) each broken bike is collected at most once; (2) the working time of the personnel is no more than T; and (3) the broken bikes collected in each sub-route are no more than the vehicle capacity M. If we use δ_{ij} to denote whether the broken bike b_i is collected during sub-route R_j, the problem can be formulated as follows:

$$\max_{\mathcal{R}} \quad \sum_{b_i \in V} \sum_{R_j \in \mathcal{R}} \delta_{ij} score_i \tag{5}$$

$$str. \quad \sum_{R_j \in \mathcal{R}} \delta_{ij} \leq 1, \quad \forall b_i \in V \tag{6}$$

$$\sum_{R_j \in \mathcal{R}} T_j \leq T \tag{7}$$

$$\sum_{b_i \in V} \delta_{ij} \leq M, \quad \forall R_j \in \mathcal{R} \tag{8}$$

Such a problem of finding k budget constrained connected components with a maximum beneficial score is NP-hard as proven in Lemma 1 below.

Lemma 1 (NP-difficulty). *When time and capacity constrained, collecting broken-sharing-bikes with a maximal beneficial score is NP-hard.*

Proof. The broken sharing bikes collection problem is a combination of broken sharing bike vertex selection and determining the shortest path between the selected vertices. As a consequence, We can reduce our problem of collecting broken-sharing-bikes with maximal beneficial score from the Knapsack Problem (KP) and the Travelling Salesperson Problem (TSP), when time and capacity constrained. We can view each broken sharing bike $b_i \in V$ as an item, with an item size (i.e., Collecting time cost), and an item profit (e.g., a beneficial score contribution). The set V of selected broken sharing bikes is viewed as a knapsack, with a fixed size T (i.e., total working time constraint). Furthermore, not all broken sharing bike $b_i \in V$ have to be visited in the problem. Determining the shortest path between the selected vertices $b_i \in V'$ will be helpful to visit as many vertices as possible in the available time. our goal is to maximize the total score collected. If a recycling worker with not enough time and capacity to collect all possible broken sharing bikes. He knows the number of beneficial scores to expect in each broken bike and wants to maximize the total beneficial score, while keeping the total travel time limited to T. Our problem boils down to an Orienteering Problem problem (OP), which is known to be NP-complete [41]. □

Given it is an NP-hard problem, we develop a heuristic-algorithm to tackle the issue.

2.2 System Overview

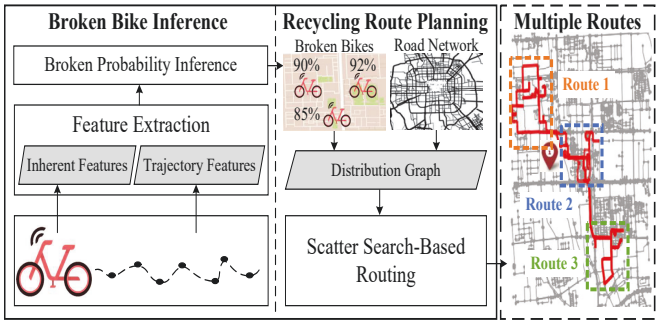


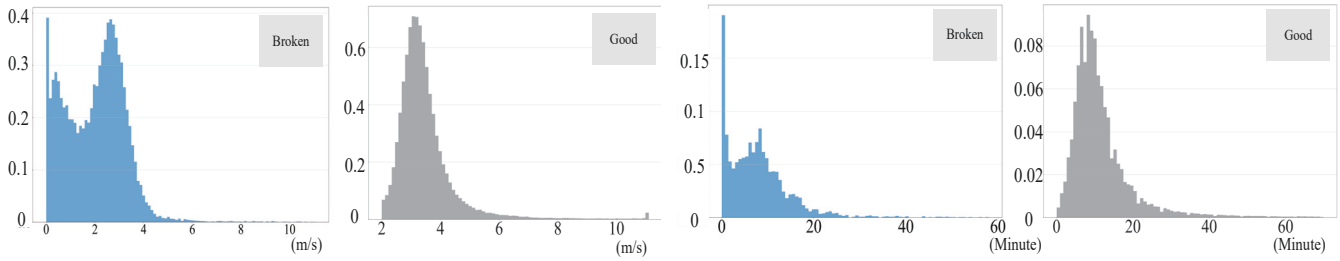
Figure 2: System Overview.

Figure 2 gives an overview of our system, which consists of two main components: (1) *Broken Bicycle Inference*, which calculates broken probability for each sharing bike, which takes the sharing bike's parameters, e.g., the bike inherent feature, and trajectory features, and outputs the bike broken probability and current status (detailed in Section 3) and (2) *Recycling Route Planning* component takes the results of the prediction model, the road network data and the recycling of historical data as input. It establishes the distribution graph of the broken bikes (detailed in Problem Definition) and recommends the optimal route for recycling the broken bike (detailed in Section 4).

3 BROKEN BIKE INFERENCE

Due to the fact that there is only a small proportion of sharing bikes reported as broken by the users, and not all of the reported bikes are truly broken, a broken bicycle inference model is required to detect the real broken bikes for the worker to collect. An inference model under the supervised-learning paradigm is used to assign a broken probability to each bicycle. In the later routing algorithm, the bike with high broken possibility is preferred to collect.

The training bike samples are selected as follows: 1) If a bike is reported as broken by Mobike user and the broken status is confirmed by the worker, we regard it as a broken bike sample; 2) If



(a) Difference of speed probability density distributions (b) Difference of duration probability density distributions

Figure 3: Mobike Trip Characteristics.

a bike is rode repeatedly in a time period (i.e., one month), and the user does not report the status of the bike as a broken, we regard it as a good bike sample.

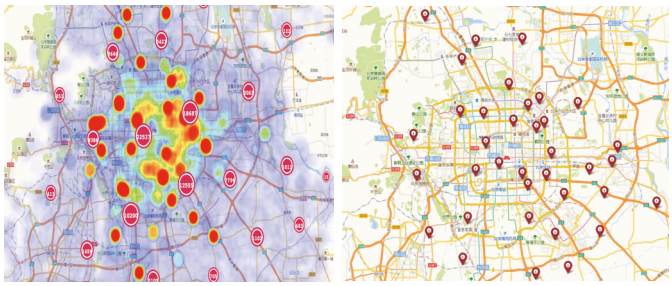
Feature Extraction. Whether a bike is broken can be inferred mainly from two aspects: *1) inherent features*, such as the life time of the bicycle, the number of ridden times, the total duration of cycling, and the number of maintenance; and *2) trajectory features*, which include the average travel speed and trip duration distributions. The selected trajectory features are derived from the analytics of Mobike trajectories. As shown in Figure 3(a), the probability of the average riding speed less than $1m/s$ for the broken bike is much higher than the good bike. This may be because some broken bikes are more cumbersome, the cycling speed will be slower. And from Figure 3(b), the trip duration of the broken bike is much shorter compared with the good one. This may be because the user finds that there is some problem with the bike after scanning the bicycle to ride, thereby terminating the cycling behavior. This phenomenon of user riding helps to determine the state of the sharing bike.

Broken Probability Inference. Since the sharing-bike status takes two values: good or broken (not good), we use a 0-1 valued binary variable y to denote the status outcome, where 1 stands for broken and 0 stands for good. We use p_i to denote the broken probability of the bike b_i. The probability depends on many factors, such as the trip duration and speed of a bike, etc. Such information can be encoded into a feature vector X_i, which is associated with the inherent features and the trajectory features of sharing bikes. Given extracted feature vector X_i, we can estimate the acceptance probability as: $p_i = p(y = broken|X_i)$. Since then, the broken inference task can be formulated as a typical binary classification problem, and the traditional classification model, such as Logistic Regression [11], can be employed.

4 RECYCLING ROUTE PLANNING

After the broken probability of each bike in the service region of a collection site is obtained, the distribution graph is constructed using bike locations with broken probabilities and the road network data. In this section, we describe the *scatter search-based routing algorithm* for the broken bike recycling problem using the constructed distribution graph.

In broken bike recycling problem, the instance size is surely beyond the solvability of standard solver, for example, as shown in Figure 4, there are typically hundreds of broken bikes in some regions, and 39 broken sharing bike collection site in Beijing. The collection



(a) Distribution of Broken Sharing Bikes (b) Distribution of Collection Site

Figure 4: Boken Mobike sharing Bike and collection site Distribution in Beijing

site of broken sharing bike need to occupy certain resources, so each collection site has its own service range. The departure and return locations of the workers are the same collection site in the area. If there is no limit to the capacity of the recycling vehicle and there are no restrictions on the working hours of the recycling workers. Our problem of recycling broken bikes with maximal beneficial score can be converted into a problem of recycling all broken sharing bikes and minimizing the overall recycling path, which can be converted into a tsp problem. However, in the case of working hours and the limited capacity of the recovered vehicle. The problem can be described as workers with not enough time and vehicle capacity to collect all possible broken bikes. He knows the beneficial score which is uniquely defined by the practical broken-bike collection problem (detailed in 2) of each broken bikes, and wants to maximize beneficial scores, while with the working hours and vehicle capacity limited.

Main Idea. Due to the capacity limitation M of the recycling vehicle, the worker can only collect the limited number of bikes during one sub-route. The main idea is that during each sub-route, we first try to collect at most M bikes with high broken probabilities (i.e., high beneficial scores), which are spatially close to each other, and then carefully plan the visiting order, so that the traveling time in each sub-route is minimized. We continuously find such sub-route until the working time is used up. The discovery of each sub-route is explained in following three stages: 1) *broken bike clustering*; 2) *sub-route selection*; 3) *status update*.

Stage 1: Broken Bike Clustering. In this stage, the bikes inferred to be broken are clustered using spatial clustering algorithm, e.g., kMeans [18], so that the broken bikes in each cluster are spatially close to each other. The number of clusters k is computed

according to both recycling vehicle capacity M and The total number of broken bikes in the area n. We initialize k as $k = round(n/M)$.

Stage 2: Sub-route Selection. In this stage, the algorithm finds the best sub-route in each cluster, and the best sub-route over all the clusters is selected. The goodness of a sub-route is defined as the beneficial score per time cost. The sub-route selection in each cluster is conducted in an iteratively way following the scatter search idea. We first select bikes with top M high scores as the initial bike set to recycle, and then design recycling route for it using TSP algorithm. Then we randomly replace a bike in the sub-route with a bike outside the sub-route but inside the cluster, to check whether there is any improvement. This process is repeated N times to obtain a stable sub-route in each cluster.

Stage 3: Status Update. In each iteration, the algorithm puts the best sub-route R_j into the final recycling route set $\mathcal{R}$, and updates the working time by subtracting the time spent recycling broken bikes in R_j and broken bike vertex set V by subtracting the recycled broken bikes in sub-route R_j. The algorithm terminates when working time T is used up, and then returns the recycling route set $\mathcal{R}$ as the recommended broken bikes recycling plan.

Algorithm Design. Algorithm 1 gives the pseudo-code of our scatter search-based heuristic algorithm. In each iteration of the *Scatter Search stage*(Line 2), the algorithm first partition the vertex set of broken bike nodes V into k clusters. The value of K is determined by the number of broken sharing bikes and the capacity of the recycling vehicle. Then, optimal broken sharing bikes collection scheme in the cluster is then selected separately in each independent cluster. When initializing the sub-route set in each cluster, two initialization strategies are employed depending on the value of a α. If the number of broken bikes in the candidate set S_i is greater than recycling vehicle capacity M, the initial recovery of the bicycle is selected using two methods. If tuning parameter α is equal to 1, the algorithm random select M broken bikes point in set S_i. otherwise, the algorithm select M broken bike in set S_i by the probability value of each broken bike as the candidate set C_i (Line 4-10). After selecting the initial result set C_i in cluster i, we use Function RecyRoute to solve the optimal recycling order of the broken bike in the result set and calculate the gain of recycling benefit score g_i. In the set S_i in which the number of each broken bicycle is larger than the recycling vehicle capacity, Take the broken bicycle not included in the set C_i which are randomly selected from the set S_i to replace random replace a broken bike in the set C_i. During the process, we keep track of the set C_i' and C_i, which has the maximum score gain in the iteration. If the number of broken bikes in the candidate set S_i is less than recycling vehicle capacity M, we just calculate the corresponding beneficial score gain. Select the best set C_i which has the maximum score gain from all clusters, and puts the best set R_j in recycling route set $\mathcal{R}$ base on Function RecyRoute. Then, R_i is removed from broken sharing bikes set V, the remaining working time is updated by subtracting the time cost $R_i.time$.At the same time, due to the reduction of the number of broken bikes, the number of clusters is also reduced (Line 11- 19).

Finally, when all the working time budget is used up, the algorithm terminates, and broken sharing-bikes recycling route set $\mathcal{R}$ is returned as the recommended broken bike recycling plan.

Algorithm 1 Scatter Search-based Routing Algorithm

Input: Broken sharing-bikes distribution graph $G = (V, E)$, working time T, parameter α, capacity M, initial number of clusters k and the maximum number of iterations N.
Output: Recycling route set $\mathcal{R}$.

1: **while** $T > 0$ **do**
//Stage 1: Broken Bike Clustering
2: $(\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_k) \leftarrow$ Kmeans(V, k)
//Stage 2: Sub-route Selection
3: **for** $i \leftarrow 1$ to k **do**
4: **if** $\mathcal{S}_i > M$ **then**
5: **if** $\alpha = 1$ **then**
6: Random select M points in $\mathcal{S}_i$ as C_i
7: **else**
8: Select the top M of broken probability in $\mathcal{S}_i$ as C_i
9: **else**
10: Select all point in $\mathcal{S}_i$ as C_i
11: $R_i, g_i \leftarrow$ RecyRoute(C_i)
12: **for** $l \leftarrow 1$ to N **do**
13: Randomly swap $b_m \in C_i$ by $b' \in \mathcal{S}_i - C_i$ as C_i'
14: $R_i', g_i' \leftarrow$ RecyRoute(C_i')
15: **if** $g_i' > g_i$ **then**
16: $C_i \leftarrow C_i'$; $R_i \leftarrow R_i'$; $g_i \leftarrow g_i'$
17: $j \leftarrow_i g_i$
//Stage 3: Status Update
18: $\mathcal{R} \leftarrow \mathcal{R} \cup \{R_j\}$; $T \leftarrow T - R_j.time$; $V \leftarrow V - R_j$
19: $k \leftarrow k - 1$
20: return $\mathcal{R}$

Function RecyRoute(C_i)
$R_i \leftarrow$ TSP(C_i); $g_i \leftarrow \frac{R_i.score}{R_i.time}$
return R_i, g_i

5 EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the effectiveness of our system. We first describe the real dataset used in the paper. Then, we present comparison results with other baseline methods over different values of α and working time constraints. Finally, we present real-world case studies to evaluate our broken bike detection and recycling route planning algorithm.

5.1 Datasets

Road Networks. The road network data in Beijing and Guangzhou, China is collected from Open Street Map [3].

Mobike Order Data. Each Mobike order contains a bike ID, a user ID. The dataset used in the paper includes the entire Mobike orders in the City of Beijing and Guangzhou from 01/08/2018 to 12/31/2018.

Mobike Recycling Data. Each Mobike recycling record contains a bike ID, a worker ID, the start time and the end time to recycle the bike. The dataset is collected in the City of Beijing, with the time span of 01/06/2017 - 12/31/2018.

Mobike Trajectories. Each Mobike trajectory contains a bike ID, a user ID, the time interval of the trajectory, the start/end locations, and a sequence of intermediate GPS points. The dataset includes the

[3] https://www.openstreetmap.org/

entire Mobike trajectory data in the City of Beijing and Guangzhou from 01/08/2018 to 12/31/2018.

5.2 Data Pre-Processing

Data Pre-processing takes the road network, the Mobike order data, Mobike recycling data, and the Mobike trajectories as input, and performs the following three tasks to prepare the data for further processing:

Data Cleaning. *Data Cleaning* cleans the raw order data, trajectories, and recycling data from Mobike. Essentially as a type of crowdsensing data, Mobike trajectories are generated by the GPS modules from mobile phones. As a result, a noticeable portion of trajectories has different data errors, which significantly affect the accuracy of the broken bike inference model. This step cleans the raw trajectories from Mobike users by filtering the noisy GPS points with a heuristic-based outlier detection method [43].

Map-Matching. In this module, we map the GPS points onto the corresponding segments in road networks, which is crucial for the broken sharing bike collection. The Mobike sharing bike can be at any location in the city, e.g., hiding in the residential area or close to the road network, where the parking location of collecting vehicles is usually along with the road network. As a result, we should employ v_w (walking speed) to better characterize the individual bike collecting events. This step evaluates the distance of each broken sharing bikes to the nearest corresponding segments in road networks with a global map matching method [28].

Map Griding. For the ease of assessing the regional rt (registration time), we adopt the griding based method, which simply partitions the map into equal side-length grids [23, 24]. our approach divides the urban area into equal-size grids with a pre-defined side-lengths in 100 meters.

5.3 Effectiveness Evaluation

In this subsection, we study the effectiveness of both broken bike prediction and recycling. Unless mentioned otherwise, the default parameters used in the experiments are: recycling vehicle capacity $M = 20$, the average speed of the worker's walking is $v_w = 1m/s$, and the average speed of the worker's driving is $v_d = 25km/h$.

5.3.1 Broken Bike Prediction.

In the broken bike prediction model, we tried two popular models: logistic regression (LR) [11] and random forest (RF) [2] algorithms. We train the models for different cities and evaluate both methods in terms of Accuracy (ACC) and Area under the Curve of ROC (AUC). Experimental results for Beijing and Guangzhou are shown in Table 1, where we observe that 1) LR outperforms RF slightly and 2) both models get good results, which validates the effectiveness of our feature extraction scheme.

5.3.2 Performance of Different TSP Methods in Recycling Route Planning.

We study the effect of different TSP methods in our recycling route planning. The test data select from Haidian District, Beijing, which the inference model give 537 broken bikes in this area as shown in Figure 7. In this work, we tried five popular models: Simulated Annealing Algorithms (SA) [13], Genetic Algorithms (GA) [31], Ant Colony Optimizations Algorithms (ACO)[9], Lin-Kernighan(LK)[14] and Self-organizing Feature Maps (SOFM)[3].

Table 1: Results of LR and RF

Beijing				
Model	ACC	AUC	Recall	F-score
LR	0.9768	0.9965	0.9763	0.97796
RF	0.9750	0.9934	0.9746	0.97608
Guangzhou				
Model	ACC	AUC	Recall	F-score
LR	0.9757	0.9948	0.9759	0.9756
RF	0.9746	0.9933	0.9745	0.9750

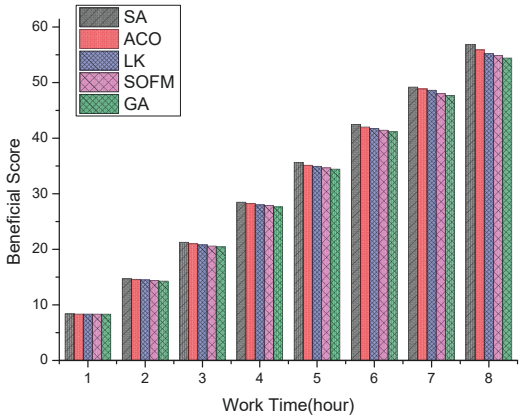


Figure 5: The Evaluation of Different TSP Methods.

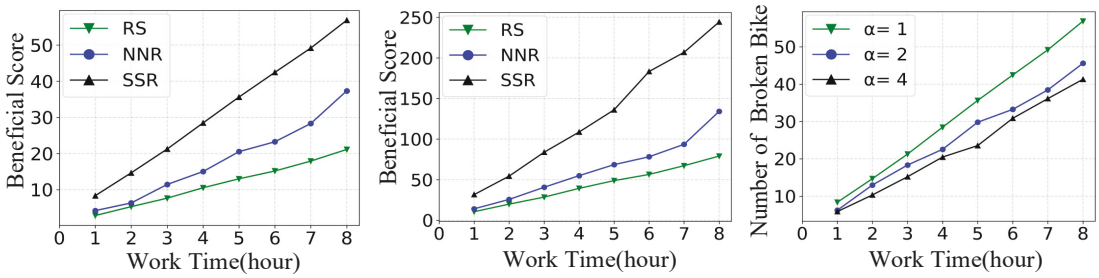
We evaluate five methods in terms of the total beneficial scores in our recycling broken sharing bike problem. Experimental results for Beijing are shown in Figure 5. Our experiments show that for our test data, these TSP algorithms do not make a significant difference as well. SA is slightly more accurate. Therefore, we choose SA as the TSP method in our recycling route planning model.

5.3.3 Recycling Route Planning.

We study the effect of different parameter settings of α and working time, and we compare our method, i.e. Scatter Search-based Routing (SSR), with two other baselines.

• **Baseline 1: Random selection (RS).** If there is no inference model in the collection problem, we assume that workers collect broken bikes according to user reports which occur randomly. We directly take the next car after each collection of a broken bike for collection. When the number of broken bikes collected reaches the vehicle capacity, return to the broken bike station in the area and repeat the random collection process for the next round. The collection process terminates when the total collection time exceeds the working time.

• **Baseline 2: Nearest neighbor routing (NNR).** The location where the broken bike is relatively densely distributed is selected as the starting area for collecting the first broken bike. In NNR, the recycling vehicle starts at the recycling parking spot, repeatedly visits the nearest broken bikes node until the capacity of the vehicle and the working hours of the workers exceed the constraint and returns back to the parking spot.



(a) Results with α =1 (b) Results with α =2 (c) Result of Recycling of Different α

Figure 6: Effectiveness Evaluation.

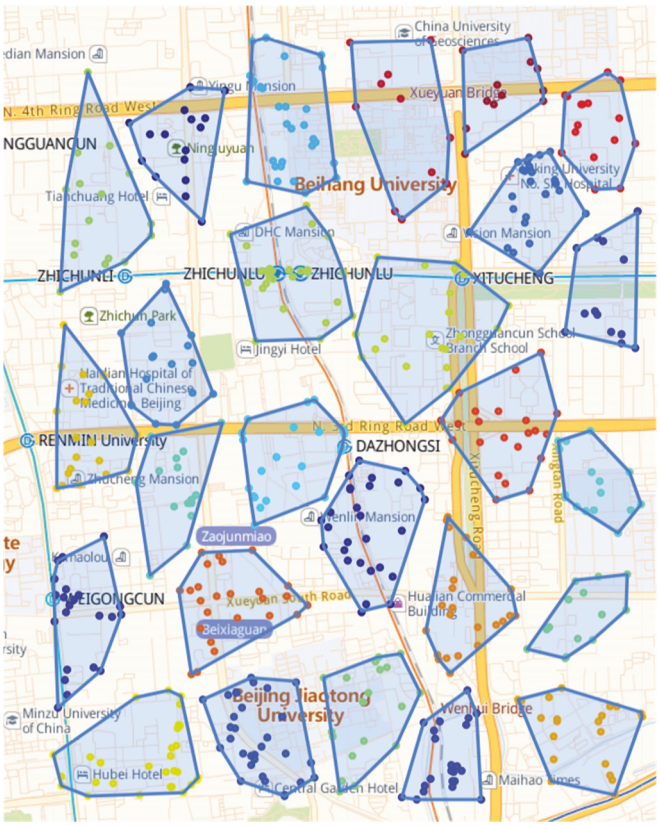


Figure 7: Broken sharing bike distribution cluster in Haidian District.

is not quite sensitive to it and the bike moves within a certain range. This has resulted in a higher recovery cost for workers. The SSR method can adjust the recovery of the beneficial score by controlling the parameter α, which can solve the phenomenon and make the overall recovery more effective as shown in Figure 6(b). Figure 6(c) provides the results with different α settings, It is interesting that, when α is large, the number of recycling broken sharing bikes will be reduced to some extent. Moreover, with a higher α, the number of recycling broken bikes is smaller, but the degree of reduction will gradually decrease. The reason behind these phenomena is that a bicycle with a higher probability of failure prediction has a higher score. Where the value of α is larger, and it is more preferable to collect a bicycle which has higher broken probability when collecting broken sharing bikes. However, the distance between bikes also affects the time cost of recycling, so when the value of a is larger, the number of broken bikes collected will become smaller.

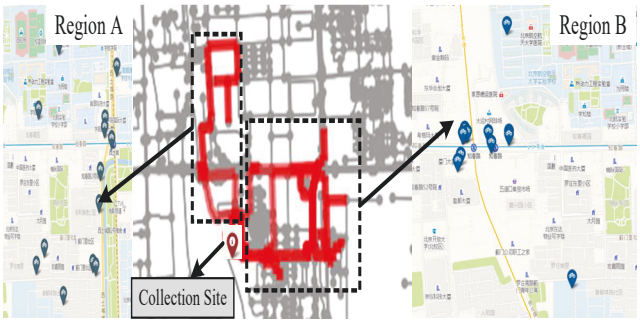


Figure 8: A Real Case Study in Haidian District, Beijing.

Effects on Total Working Time Budget. Figure 6 illustrates the total beneficial scores with different total working time budgets for a worker with Mobike, from 1 Hour to 8 Hour. The experimental results of finding a broken bike based on a random walk of RS is the average value of the income score after the algorithm solves the problem 1000 times independently. From the figure, we make the following observations: 1) the scatter search-based heuristic SSR method performs better than other baseline models. 2) When working hours are between 5 hours and 6 hours, the NNR method will have a useful period of slow growth. This is because, during this time, the NNR method took a long time in a broken bicycle. It is interesting that, because of the slight damage to the bike, the user

5.4 Case Studies

To better understand the effectiveness of our bike prediction and recycling model, we conduct a field case study. We choose to visit the area near Zhichun Road, Dazhongsi, and Beitucheng subway station in Haidian District, Beijing.

Figure 8 gives the path that Mobike operators use to recycle broken bikes in this area. The workers recovered a total of 32 broken bikes in the vicinity in 8 hours, and mainly concentrated near the temporary parking spots. The traditional recycling methods of worker are similar to the NNS method. The worker first finds the area where the broken bikes are densely distributed near the temporary parking point through the location reported by bikes. Then,

a broken bike in the dense area is selected and recycled according to the scheme of the shortest travel distance of the map navigation. When the target bicycle is found, another broken bike which is closest to the current location is selected for recycling. However, this will make recyclers tend to pay more attention to broken bikes that are closer to temporary parking spots. The distribution of broken bikes in the area is not fully considered, resulting in low efficiency and high cost of recycling. Figure 9 shows the results of the broken recovery path recommended by the SSR method. Mobike's worker recycles the broken bikes in a given area three times and collect 59 broken bikes in an 8 hour working time limit. It is interesting that, the recommended recovery path of the algorithm is not only the broken bikes concentrated near the temporary parking point, but also the broken bikes far from the temporary parking point are also included in the recommended collection of recycling. This is because the algorithm parameters take into account of the overall distribution of the bikes and the optimal recovery sequence in the actual recycling process, and the parameters are tuned according to the efficiency gain of each bike recovery. This makes the recycling of bikes more efficient. At the same time, the two broken bikes at the bottom left of Figure 9 are the broken bikes that are seriously broken in the recovery. One is the chain is broken, and the other is the seat is lost. The confidence of the two cars in the model is 0.99988294 and 0.99961954 respectively. These two bikes belong to the broken bike that is preferentially collected when the value of α is greater than one. This shows that the model is sensitive to bike with a relatively high degree of failure.

Figure 9: A Real Case Study Base on Scatter Search Model Result in Haidian District, Beijing.

6 RELATED WORK

The research of fault sharing bikes recycling can be summarized in two main areas: 1) Urban Crowd Sourcing, and 2) Route Planning.

Urban Crowd Sourcing. Essentially, we take advantage of the massive Mobike users in a city to perform the fault bike detection task. Similar problems are addressed with the crowdsourcing techniques[19, 26]. For example, The literature [42] quantifies the fragility of cities through detecting the delay in commuting activities using GPS data collected from smartphones. The literatures [34, 36] infer noise levels for locations by smartphone users. The literature [27] proposes a bike sharing network optimization approach by extracting fine-grained discriminative features from human mobility data, point of interests (POI), as well as station network structures. The literatures [7, 15] identify potholes or classify road quality from vehicle's accelerometer data. Differing from the above works, we focus on the problem of broken sharing bikes detection and collecting path planning.

Route Planning. The fault sharing bike recycling problem is related to the multiple Traveling Salesman Problem (mTSP) [1, 38] and orienteering Problem (OP) [4, 39, 41]. mTSP and OP can be considered as a relaxation of our problem, with the capacity or working time restrictions removed. The solutions for these two problems are primarily in two fold: 1) optimal algorithms and 2) heuristic algorithms. In literatures [21, 35], the authors use branch-and-bound to solve instances with less than 20 and 150 vertices, respectively. The authors in [22] use a cutting plane method to obtain better upper bounds. In literatures [10, 12], the authors propose branch-and-cut algorithms. However, the branch-and-cut procedure with instances up to 500 vertices cannot be performed. GAs are relatively stochastic search algorithms based on evolutionary biology and computer science principles [16]. Using GAs to the mTSP problem have several representations, like one chromosome technique [33], the two chromosome technique [32] and the latest two-part chromosome technique. The authors in [25] propose an ant colony optimization approach and a tabu search algorithm. In literature [37], the authors develop a Pareto ant colony optimization algorithm and a multi-objective variable neighborhood search algorithm. In [40], the authors propose a Variable Neighbourhood Search (VNS) algorithm and embed an exact algorithm to deal with a path feasibility subproblem. In [20], the authors present two polynomial size formulations for OP. The authors in [30] discuss several vehicle routing algorithms, and present a heuristic method which searches over a solution space formed by the large number of feasible solutions to an mTSP. The authors in [17] study the adaptive stochastic knapsack problem with deterministic size and stochastic rewards. Their problem objective is to find a sequential inserting policy to maximize the probability of the reward exceeding a threshold value without violating the capacity constraint. In [5], the authors study the adaptive stochastic knapsack problem with items of deterministic reward and stochastic size. Their goal is to maximize expected value while fitting all the items in the knapsack. The authors demonstrate the benefit of an adaptive policy and provide an approximation approach. In [6], the authors study an orienteering problem with stochastic travel times and present adaptive path planning methods to take advantage of dynamically updating data; combine the orienteering problem and optimal path finding into a single model. The authors in [8] discuss the vehicle routing problem with hard time windows and stochastic service times (VRPTW-ST). They adopt the dynamic programming algorithm to account for the probabilistic resource consumption by extending the label dimension and by providing new dominance rules. In this paper two recourse strategies are proposed and the resulting problems are solved by branch-price-and-cut algorithms. However, all of these works cannot be directly used for broken sharing bikes recycling, because these works simply test on benchmark instances and fail to consider the realistic constraints and road network distance.

7 CONCLUSION

In this paper, we introduce a novel approach to detect broken sharing bikes and recommend the appropriate bicycle recycling path to the worker based on the real sharing bikes data collected from Mobike (a major station-less bike sharing system). Our system can address the problem of recycling efficiency of broken sharing bikes in a more realistic fashion, considering the constraints and requirements from sharing bike worker's perspective: 1) working time limitations, 2) vehicle capacity constraints, and 3) broken sharing bike recovery benefit. We also propose a flexible beneficial score function to adjust preferences between the number of bikes recovered and the predicted probability of damage to bikes. The formulated problem is proven to be NP-hard, thus we propose a *scatter search-based heuristic* algorithm. We perform extensive experiments on a large scale Mobike data and demonstrate the effectiveness of our proposed broken sharing bike predict model and bike recycling routing model, where our model can predict the broken sharing bikes with above 97% accuracy and recommends that the number of real broken bikes recovered by the recycling path of the broken bikes is two to three times that of the Mobike traditionally recycling broken bikes.

REFERENCES

[1] Tolga Bektas. 2006. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega* 34, 3 (2006), 209–219.

[2] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[3] Łukasz Brocki and Danijel Koržinek. 2007. Kohonen self-organizing map for the traveling salesperson problem. In *Recent Advances in Mechatronics*. Springer, 116–119.

[4] I-Ming Chao, Bruce L Golden, and Edward A Wasil. 1996. The team orienteering problem. *European journal of operational research* 88, 3 (1996), 464–474.

[5] Brian C Dean, Michel X Goemans, and Jan Vondrdk. 2004. Approximating the stochastic knapsack problem: The benefit of adaptivity. In *45th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 208–217.

[6] Irina Dolinskaya, Zhenyu Edwin Shi, and Karen Smilowitz. 2018. Adaptive orienteering problem with stochastic travel times. *Transportation Research Part E: Logistics and Transportation Review* 109 (2018), 1–19.

[7] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. 2008. The pothole patrol: using a mobile sensor network for road surface monitoring. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*. ACM, 29–39.

[8] Fausto Errico, Guy Desaulniers, Michel Gendreau, Walter Rei, and L-M Rousseau. 2018. The vehicle routing problem with hard time windows and stochastic service times. *EURO Journal on Transportation and Logistics* 7, 3 (2018), 223–251.

[9] Jose B Escario, Juan F Jimenez, and Jose M Giron-Sierra. 2015. Ant colony extended: experiments on the travelling salesman problem. *Expert Systems with Applications* 42, 1 (2015), 390–410.

[10] Matteo Fischetti, Juan Jose Salazar Gonzalez, and Paolo Toth. 1998. Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing* 10, 2 (1998), 133–148.

[11] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2001. *The elements of statistical learning*. Vol. 1. Springer series in statistics New York.

[12] Michel Gendreau, Gilbert Laporte, and Frederic Semet. 1998. A branch-and-cut algorithm for the undirected selective traveling salesman problem. *Networks: An International Journal* 32, 4 (1998), 263–273.

[13] Xiutang Geng, Zhihua Chen, Wei Yang, Deqian Shi, and Kai Zhao. 2011. Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. *Applied Soft Computing* 11, 4 (2011), 3680–3689.

[14] Keld Helsgaun. 2009. General k-opt submoves for the Lin–Kernighan TSP heuristic. *Mathematical Programming Computation* 1, 2-3 (2009), 119–163.

[15] Marius Hoffmann, Michael Mock, and Michael May. 2013. Road-quality classification and bump detection with bicycle-mounted smartphones. In *Proceedings of the 3rd International Conference on Ubiquitous Data Mining-Volume 1088*. CEUR-WS. org, 39–43.

[16] John Holland. 1975. Adaptation in natural and artificial systems: an introductory analysis with application to biology. *Control and artificial intelligence* (1975).

[17] Taylan İlhan, Seyed MR Iravani, and Mark S Daskin. 2011. The adaptive knapsack problem with stochastic rewards. *Operations research* 59, 1 (2011), 242–248.

[18] Anil K Jain. 2010. Data clustering: 50 years beyond K-means. *Pattern recognition letters* 31, 8 (2010), 651–666.

[19] Shenggong Ji, Yu Zheng, and Tianrui Li. 2016. Urban Sensing Based on Human Mobility. UbiComp 2016. https://www.microsoft.com/en-us/research/publication/urban-sensing-based-human-mobility/

[20] Imdat Kara, Papatya Sevgin Bicakci, and Tusan Derya. 2016. New formulations for the orienteering problem. *Procedia Economics and Finance* 39 (2016), 849–854.

[21] Gilbert Laporte and Silvano Martello. 1990. The selective travelling salesman problem. *Discrete applied mathematics* 26, 2-3 (1990), 193–207.

[22] Adrienne C Leifer and Moshe B Rosenwein. 1994. Strong linear programming relaxations for the orienteering problem. *European Journal of Operational Research* 73, 3 (1994), 517–523.

[23] Yanhua Li, Jun Luo, Chi-Yin Chow, Kam-Lam Chan, Ye Ding, and Fan Zhang. 2015. Growing the charging station network for electric vehicles with trajectory data analytics. In *2015 IEEE 31st International Conference on Data Engineering*. IEEE, 1376–1387.

[24] Yanhua Li, Moritz Steiner, Jie Bao, Limin Wang, and Ting Zhu. 2014. Region sampling and estimation of geosocial data with dynamic range calibration. In *2014 IEEE 30th International Conference on Data Engineering*. IEEE, 1096–1107.

[25] Yun-Chia Liang, Sadan Kulturel-Konak, and Alice E Smith. 2002. Meta heuristics for the orienteering problem. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, Vol. 1. IEEE, 384–389.

[26] Dongyu Liu, Di Weng, Yuhong Li, Jie Bao, Yu Zheng, Huamin Qu, and Yingcai Wu. 2016. Smartadp: Visual analytics of large-scale taxi trajectories for selecting billboard locations. *IEEE transactions on visualization and computer graphics* 23, 1 (2016), 1–10.

[27] J. Liu, Q. Li, M. Qu, W. Chen, J. Yang, H. Xiong, H. Zhong, and Y. Fu. 2015. Station Site Optimization in Bike Sharing Systems. In *2015 IEEE International Conference on Data Mining*. 883–888. https://doi.org/10.1109/ICDM.2015.99

[28] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. 2009. Map-matching for low-sampling-rate GPS trajectories. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*. ACM, 352–361.

[29] Russell Meddin and Paul DeMaio. 2015. The bike-sharing world map. (2015). URLhttp://www.bikesharingworld.com

[30] RH Mole, DG Johnson, and K Wells. 1983. Combinatorial analysis for route first-cluster second vehicle routing. *Omega* 11, 5 (1983), 507–512.

[31] Yuichi Nagata and Shigenobu Kobayashi. 2013. A powerful genetic algorithm using edge assembly crossover for the traveling salesman problem. *INFORMS Journal on Computing* 25, 2 (2013), 346–363.

[32] Yang-Byung Park. 2001. A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines. *International Journal of Production Economics* 73, 2 (2001), 175–188.

[33] Jean-Yves Potvin, Guy Lapalme, and Jean-Marc Rousseau. 1989. A generalized k-opt exchange procedure for the MTSP. *INFOR: Information Systems and Operational Research* 27, 4 (1989), 474–481.

[34] Zhaokun Qin and Yanmin Zhu. 2016. NoiseSense: A crowd sensing system for urban noise mapping service. In *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 80–87.

[35] R Ramesh, Yong-Seok Yoon, and Mark H Karwan. 1992. An optimal algorithm for the orienteering tour problem. *ORSA Journal on Computing* 4, 2 (1992), 155–165.

[36] Rajib Kumar Rana, Chun Tung Chou, Salil S Kanhere, Nirupama Bulusu, and Wen Hu. 2010. Ear-phone: an end-to-end participatory urban noise mapping system. In *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks*. ACM, 105–116.

[37] Michael Schilde, Karl F Doerner, Richard F Hartl, and Guenter Kiechle. 2009. Metaheuristics for the bi-objective orienteering problem. *Swarm Intelligence* 3, 3 (2009), 179–201.

[38] Joseph A Svestka and Vaughn E Huckfeldt. 1973. Computational experience with an m-salesman traveling salesman algorithm. *Management Science* 19, 7 (1973), 790–799.

[39] Tommy Thomadsen and Thomas K Stidsen. 2003. The quadratic selective travelling salesman problem. (2003).

[40] Fabien Tricoire, Martin Romauch, Karl F Doerner, and Richard F Hartl. 2010. Heuristics for the multi-period orienteering problem with multiple time windows. *Computers & Operations Research* 37, 2 (2010), 351–367.

[41] Pieter Vansteenwegen, Wouter Souffriau, and Dirk Van Oudheusden. 2011. The orienteering problem: A survey. *European Journal of Operational Research* 209, 1 (2011), 1–10.

[42] Takahiro Yabe, Kota Tsubouchi, and Yoshihide Sekimoto. 2017. CityFlowFragility: Measuring the Fragility of People Flow in Cities to Disasters using GPS Data Collected from Smartphones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 117.

[43] Yu Zheng. 2015. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 3 (2015), 29.