Shared Bikes Scheduling Under Users’ Travel Uncertainty

ZHI-YONG ZHANG AND XIAO ZHANG
School of Economics and Management, Xidian University, Xi’an 710026, China
Corresponding author: Xiao Zhang (zhangxiao.neu@163.com)

This work was supported in part by the National Natural Science Foundation of China under Project 71401131, in part by the Natural Science Foundation of Shaanxi Province under Grant 2019JM-110, and in part by the Fundamental Research Funds for the Central Universities under Grant JB190604, Grant JBX180603, and Grant RW180173.

ABSTRACT With the rise of green concept, shared bikes are booming. The accompanying unbalanced scheduling problem is a scientific problem that needs to be solved urgently. Aiming at the problem of shared bikes scheduling with travel uncertainty, a multi-objective integer programming model is established based on the consideration of static demand of fix time period, station capacity limit, penalty cost and other practical factors. In addition, this paper gives the basic formula to calculate the parameters in the model. An algorithm based on “ant colony algorithm” is then given to solve the model. Taking the massive data provided by the “Mobike” company in 2017 as an example, this paper uses the program analysis data to prove the feasibility and effectiveness of the model and get the initial optimization plan. Finally, the data simulation is carried out to verify the feasibility and accuracy of the optimization scheme and the optimization scheme is adjusted accordingly to obtain the final optimization scheme. The research results show that the final optimization scheme proposed in this paper has certain reference value for the scheduling problem of Shanghai “Mobike.”

INDEX TERMS Shared bikes, travel uncertainty, scheduling, static demand interval, rebalancing.

I. INTRODUCTION

With the rise of green concept, shared bikes is booming. The emergence of shared bikes reflects another awakening of people’s ecological and environmental health and their own health. At the same time, with the rapid development of mobile Internet technology and smart devices, shared bikes have blossomed everywhere in China’s major capital cities since the second half of 2016, and their demand has exploded. According to statistical data in 2016, the number of domestic shared bikes users reached 18.864 million, with a growth rate of 700% [40]. In 2017, the number of domestic shared bikes users reached 210 million, with a growth rate of 646% [5]. According to the “2018-2023 shared bikes industry market prospects and investment and financing strategy research report” released by the China Business Research Institute, the domestic shared bikes market will exceed 23 billion yuan in 2019 and the number of shared bikes users is expected to reach 376 million. However, with the rapid development of shared bikes, the problem of unbalanced dispatching is also needed to be solved urgently, which has attracted the attention of academic circles and the industry. The problem of unbalanced dispatching is usually caused by inaccurate scheduling scheme, including inaccurate scheduling quantity, path and so on.

The shared-bike scheduling problem is also known as the shared-bike rebalancing problem or the bikes relocation problem. According to the research of Raviv et al. [37] and Forma et al. [21], the shared bikes scheduling problem is classified into the static problem and dynamic problem by the low and high activity level of the travel system. Therefore, in the problem of shared bikes scheduling, scholars have carried out the following research on static and dynamic problems according to their different research purposes and methods.

In the static problem, the shared bikes scheduling problem is a variant of “one traveling salesman problem for goods pickup and delivery” (1-PDTSP), which was studied by Hernandez-perez and Salazar-gonzalez [18]–[20]. Among the pioneer scholars, Chemla et al. [7] and Raviv et al. [37] defined and solved static rebalancing problems. Chemla et al. [7] studied a deterministic static single vehicle rebalancing problem where each station can be visited more than once and proposed branch-and-cut algorithms and
a TABU search algorithm for solving the problem. Moreover, Li et al. [42] considered multiple types of bikes to be rebalanced with a single vehicle and proposed a mixed-integer linear programming (MILP) model and a hybrid genetic algorithm to solve the problem. Szeto et al. [40] and Szeto and Shui [41] studied the problem that determines the routes, the loading and unloading quantities that minimize the positive deviation from the tolerance of total demand dissatisfaction and the service times. Based on the above research on single-vehicle static shared bikes rebalancing problem, a growing number of scholars have focused on the multi-vehicle static shared bikes rebalancing problem in recent years. Raviv et al. [37] considered this problem, where the objective is to minimize both operational costs and users' dissatisfaction. In addition, Forma et al. [21] developed a heuristic procedure to solve the similar problem, where stations can be visited multiple times, transshipment is allowed, and all routes can be operated for a long time every day. Ho and Szeto [34], [35] solved this problem based on the same model by proposing a hybrid large neighborhood search. Nowadays more scholars are willing to solve the static shared bikes scheduling problem in different situations because it matches the reality and easier to solve.

The research for the dynamic problem is rare since it is very difficult to consider scheduling problem of shared bikes in a dynamically changing system or from a dynamic perspective. Contardo et al. [3] considered a dynamic version of this problem where the fleet of vehicles is heterogeneous and put forward time-indexed formulations, settled via Dantzing-Wolfe and Benders’ decomposition based heuristics. Zhang et al. [9] proposed a MILP formulation and a math-heuristic approach to solve the dynamic version of this problem considering the inventory level and users’ arrival forecasting in a time-space network flow model. Shui and Szeto [5] introduced a dynamic green shared bikes rebalancing problem that minimizes the total dissatisfaction demand and the fuel and CO2 emission cost of the rebalancing vehicle over an operational period and they proposed a hybrid rolling horizon artificial bee colony algorithm to solve it.

It is worth mentioning that, nowadays, many scholars pay more attention to the users’ uncertain travel when studying the static and dynamic shared bikes scheduling problems. Raviv et al. [37] considered uncertain information implicitly in their penalty function. Regue and Recker [33] solved a dynamic problem, where routing and inventory are both accounted and the uncertain demand part is estimated using historical data feeding a forecasting model. Saharidis et al. [17] suggested a mathematical formulation to design the shared bikes infrastructure, deciding the location and the size of the bike stations and incorporating an hourly demand estimation in order to deal with the uncertainty of user travel. Schuĳbroek et al. [27] derived bounds on inventory quantities by modeling inventory as a non-stationary Markov chain and used them in mixed integer and constraint programming models to cope with the uncertainty of demand. Recently, Brinkmann et al. [23] considered an uncertain-dynamic inventory routing problem in shared bikes systems and presented a dynamic policy to anticipate future demands to minimize the expected unsatisfied demands.

Previous studies have significantly advanced the research on shared bikes scheduling problem from different perspectives and provided a lot of references for this paper. However, there are still some gaps in the existing research as follows. (1) Some scholars regard user travel as regular in order to simplify the calculations. However, it leads to a large deviation in the implementation of the scheduling scheme in reality. Wang and Wang [22] provided an analysis on bike repositioning strategies by conducting a simulation study where real time and historical data are considered, found that users’ travel is uncertain in the use of shared bikes. Therefore, it is necessary to consider the user uncertain travel when study the shared bikes scheduling problem. (2) Some previous studies have considered the shared bike station capacity is a fixed value [14] or a range with fixed value [22] to facilitate the calculation. However, it does not match the actual situation and often leads to a deviation in making the scheduling scheme. The capacity of a station should have a certain limit and different stations should have different capacity limit. The lower bound of capacity should be used to cope with the uncertain travel of users and the upper bound of capacity should be used to prevent excessive bikes from affecting road traffic. (3) When scholars consider the penalty cost to occur, it is usually assumed that the penalty cost is generated because the user’s demand is not satisfied. However, it does not match the actual situation because the penalty cost may also result in a fine due to the excessive number of vehicles in the station. Based on the above analysis, to solve the shared bikes scheduling problem from static perspective, (1) it is essential to take into account the users’ uncertain travel, including the uncertainty of route and travel, (2) it is important to calculate the upper and lower limits of the station capacity by the user’s historical travel data, and (3) it is necessary to consider two different penalty functions. This is the motivation of our study.

This paper aims to solve the problem of the shared bikes scheduling problem with users’ uncertainty travel. (1) This paper will consider the uncertainty of the users’ travel, including the uncertainty of route and travel time, and use the following two methods to deal with the uncertainty and dynamics of user travel. Firstly, this paper intends to divide the time of a day into four static and unchanging small periods. The intensity of users’ uncertain travel can be regarded as regular and continuous in each period. Secondly, the shared bikes capacity of each station will have a lower and upper bound to allow the uncertainty of users’ travel to be taken into account when scheduling the number of shared bikes. The uncertainty of the users’ travel is settled to the greatest extent in this paper by the determination of the lower and the upper bounds of the stations’ capacity. (2) This paper will
give the calculation formula of the upper and lower limits of the station capacity and combine the user’s historical travel data to calculate the upper and lower limits of the station capacity. In the research problem of this paper, the capacity of different stations is not necessarily the same and the capacity interval of each station is not fixed. The capacity interval of each station changes as the user’s travel data updates. (3) The penalty cost caused by the excess vehicle quantity will be considered so as to completely consider all the circumstances of the penalty cost of the shared bikes scheduling scheme. This paper gives an algorithm based on “ant colony algorithm” to solve the model. The massive data provided by “Mobike” in 2017 is used to show the feasibility and effectiveness of the model and obtain the initial optimization plan. Finally, the feasibility and accuracy of the optimization scheme are tested and adjusted by data simulation based on the original data.

The rest of this paper is organized as follow. Section 2 establishes the multi-objective integer programming model to solve the shared bikes scheduling problem with users’ uncertainty travel, and gives the basic formula to calculate the parameters in the model. In section 3, this paper will use the program analysis data provided by the “Mobike” company in 2017 to prove the feasibility and effectiveness of the model and get the initial optimization plan. Finally, we will modify the initial optimized scheduling scheme through data simulation. The paper concludes with a summary and an outlook in section 4.

II. MODEL AND FORMULATION
Based on the reality and the results of previous studies, the basic assumption, basic formulas and models of the research problem will be given in this part.

A. MODEL ASSUMPTIONS & PARAMETERS AND NOTATIONS
1) MODEL ASSUMPTIONS

Assumption 1: The process of user groups using the shared bikes for trips every day in different stations is uncertain.

Assumption 2: The travel path of the shared bikes is not considered, but only the starting point and the arrival point of the shared bikes are considered.

Assumption 3: The number of shared bikes in different stations is within a certain range, and there is an upper and lower boundary.

Assumption 4: All shared bikes in the stations can be used normally. At the same time, shared bikes will not be damaged during the trip.

Assumption 5: The real time scheduling is not considered, but only the case of a specific interval scheduling is considered.

Assumption 6: The capacity of the transport vehicle between station \( i \) and station \( j \) is capped and the upper limit is a fixed value.

Assumption 7: The total number of shared bikes in the station system is constant and there is neither increase nor decrease in the number of shared bikes.

2) PARAMETERS AND NOTATIONS
The symbol descriptions in the mathematical model are described in Table 1.

B. MATHEMATICAL MODEL AND SOLUTION
In this paper, the shared bikes scheduling problem with users’ uncertain travel is defined in a complete directed graph. The directed graph is \( G = (V, A) \), where \( V \) represents a set of stations and \( A \) represents a set of edges in this directed graph. In set \( V \), \( N_i (i = 1, 2, \ldots, k) \) represents the stations in the directed graph. In set \( A \), \( e \) represents the edges in the directed graph and \( e_{ij} \) represents the edge form station \( i \) to station \( j \). The capacity of each edge to transport shared bikes is fixed. Therefore, it is stipulated in this paper that the capacity of each edge transport shared bikes is \( U \). Each station in this digraph has three corresponding parameters \( (l_i, b_i, h_i) \). The meanings of these three parameters are the lower limit of the number of shared bikes at the station, the variation in the number of shared bikes at the station during a certain period and the upper limit of the number of shared bikes at the station. The average cost \( C_{ij} \) of scheduling shared bikes from station \( i \) to station \( j \) is related to \( e_{ij} \). For the convenience of calculation, the total cost of this scheduling is recorded as \( C_{ij} \). Before people start to travel every day, the company must calculate the number of trips at different times of the day at each station to arrange a reasonable shared bikes scheduling plan, and the number of shared bikes at each station after scheduling should conform to the given limit \( (l_i, h_i) \) in the previous hypothesis. Meanwhile, the goal of this scheduling scheme is to minimize scheduling cost, minimize scheduling distance and maximize users’ satisfaction.

Since the users’ travel rules using shared bikes are affected by work or learning arrangements, users’ travel is a dynamic and uncertain process. However, the modeling of the dynamic rebalancing process of shared bikes with users’ uncertain trip is difficult and does not conform to the actual demands. Therefore, this paper will use the following two methods to deal with the uncertainty and dynamics of user travel. Method 1, this paper intends to divide the time of a day into four static and unchanged small periods. The intensity of users’ travel can be regarded as regular and continuous in each period of time. For the convenience of statistics, this paper will take 6 hours as a basic time unit and divide the time of a day into the following four stages: early morning (0:00-6:00), morning (6:00-12:00), afternoon (12:00-18:00) and evening (18:00-24:00). Therefore, \( y = 4 \). Method 2, the capacity of each station will be set in this paper, and each station’s capacity has a lower bound \( l_i \) and upper bound \( h_i \). Therefore, it allows the uncertainty of users’ travel to be taken into account when scheduling the number of shared bikes. And the uncertainty of the users’ travel is settled to
the greatest extent in this paper by the determination of the lower bound \( l_i \) and the upper bound \( h_i \).

In summary, combined with the basic assumptions in 2.1.1 as constraints, with the highest users’ satisfaction, the lowest scheduling cost and the shortest scheduling distance as the objective function, a multi-objective integer programming model is established as follows.

\[
\max \sum_{i=1}^{k} S_i \quad (1)
\]

\[
\min \sum_{i=0}^{k} C_i \quad (2)
\]

\[
\min R \quad (3)
\]

\[
s.t. \sum_{i=0}^{k} \alpha_i(x, y) = L, \quad \forall x \in R,
\]

\[
y \in \{1, 2, 3, 4\}, \quad L \in R \quad (4)
\]

\[
\sum_{i=1}^{k} \beta_{ij} = 1 \quad (5)
\]

\[
\sum_{j=1}^{k} \beta_{ij} = 1 \quad (6)
\]

\[
d_{ij}(x, z) \leq \beta_{ij} \times U, \quad \forall i \in k, \quad j \in k, \quad i \neq j \quad (7)
\]
\[ \sum_{j=1}^{k} d_{ij}(x, z) \leq \alpha_i(x, z), \quad \forall i \in k, j \in k, \ i \neq j \]  
\[ \delta_{ij}^+(x, y) = \delta_{ij}^-(x, y) \]  
\[ \delta_{ij}^+(x, y) = \delta_{ij}^-(x, y) \]  
\[ \beta_{ij} = 0 \text{ or } \beta_{ij} = 1, \quad \forall i \in k, j \in k, \ i \neq j \]  
\[ \alpha_i(x, y) \geq 0 \text{ and } \alpha_i(x, y) \in Z \]  
\[ d_{ij}(x, z) \geq 0 \text{ and } d_{ij}(x, z) \in Z \]  

The equality constraint (4) constrains that the number of shared bikes in the overall space is a fixed value, and the number will not increase, decrease or change with time. Constraints (5) and (6) constrain that there is only one scheduling service vehicle to serve when scheduling from station \( i \) to station \( j \), and the number of participating in the service is only once. Constraint (7) constrains the limit of the number of shared bikes scheduled from station \( i \) to station \( j \), and the number of shared bikes scheduled for this path can not exceed the maximum measurement of the path. Constraint (8) constrains that the total number of shared bikes dispatched from station \( i \) during the scheduling period can not exceed the total number of shared bikes owned by station \( i \) during the scheduling period. Constraint (9) and (10) implement flow protection for the flow of shared bikes on each node, ensuring that each shared bike will not disappear or increase in the system. Constraint (11) constrains the selection of shared bikes scheduling path. If the path is selected, then the constraint (11) and constraint (13) are non-negative constraints.

### C. BASIC FORMULAS

Considering the users’ shared bikes travel uncertain scheduling process, the basic formulas are obtained as follows.

1. Variation in the number of shared bikes at station \( i \) during a certain period of time, \( b_i \), can be given by (Chemla et al., [6])

\[ b_i = \delta_i^+ - \delta_i^- . \]  

When \( b_i \geq 0 \), it indicates that the number of shared bikes in the station \( i \) increases during this period, which may exceed the upper limit of the number of shared bikes in this station. Therefore, at this point the station’s scheduled job should remove the shared bikes to make the station balanced. On the contrary, the result is reversed.

Variation in the number of shared bikes at station \( i \) during \( y \) period of \( x \) day, \( b_i(x, y) \), can be given by

\[ b_i(x, y) = \delta_i^+(x, y) - \delta_i^-(x, y). \]  

\( b_i(x, y) \) will be used to determine the specific scheduling plan, that is, the scheduling plan of shared bikes of station \( i \) on a specific day and time period.

2. According to Assumption 3, the number of shared bikes at each station is within a certain range, and this range has lower and upper bounds \((l_i, h_i)\). Therefore, before the scheduling of shared bikes, the number of shared bikes at each station should be within its range, otherwise, there will be a penalty cost. The upper boundary indicates that the station can accommodate the maximum number of shared bikes while satisfy the demands of users without disrupting traffic, while the lower boundary indicates that the station is only able to accommodate the minimum number of shared bikes.

The lower limit of the number of shared bikes at station \( i \) will be given below.

When station \( i \) has a missing in a certain period of time, it should provide station \( i \) at least a certain number of shared bikes during the time period. The number of shared bikes is the lower bound of station \( i \). When station \( i \) in the whole period of time are in excess, the lower bound of station \( i \) is the difference between “the difference between the maximum value of the number of shared bikes in and out” and “the difference between the minimum value of the number of shared bikes in and out” in a certain period of time. There are two cases as follows.

**Case 1:** If the station \( i \) is in missing during a certain period, i.e.,

\[ \sum_{y=1}^{2} \delta_i^+(x, y) - \sum_{y=1}^{2} \delta_i^-(x, y) \leq 0 \]

\[ \sum_{y=3}^{4} \delta_i^+(x, y) - \sum_{y=3}^{4} \delta_i^-(x, y) \leq 0 \]

For the convenience of formula display, let

\[ u_i(x, y) = \sum_{y=1}^{2} \min \delta_i^+(x, y) - \sum_{y=1}^{2} \min \delta_i^-(x, y) \]

\[ v_i(x, y) = \sum_{y=3}^{4} \min \delta_i^+(x, y) - \sum_{y=3}^{4} \min \delta_i^-(x, y) \]

Then,

\[ l_i = \min \{ |u_i(x, y)|, |v_i(x, y)| \} . \]  

**Case 2:** If station \( i \) is in excess throughout the period, i.e.,

\[ \sum_{y=1}^{2} \delta_i^+(x, y) - \sum_{y=1}^{2} \delta_i^-(x, y) \geq 0 \]

\[ \sum_{y=3}^{4} \delta_i^+(x, y) - \sum_{y=3}^{4} \delta_i^-(x, y) \geq 0 \]

For the convenience of formula display, let

\[ \varphi_i(x, y) = \sum_{y=1}^{2} \max \delta_i^+(x, y) - \sum_{y=1}^{2} \max \delta_i^-(x, y) \]

\[ \eta_i(x, y) = \sum_{y=1}^{4} \min \delta_i^+(x, y) - \sum_{y=1}^{4} \min \delta_i^-(x, y) \]

\[ \phi_i(x, y) = \sum_{y=3}^{4} \max \delta_i^+(x, y) - \sum_{y=3}^{4} \max \delta_i^-(x, y) \]
\[ \mu_i(x, y) = \sum_{y=3}^{4} \min \delta_i^+(x, y) - \sum_{y=3}^{4} \min \delta_i^-(x, y) \]

\[ \theta_i(x, y) = \sum_{y=1}^{2} \max \delta_i^+(x, y) - \sum_{y=1}^{2} \min \delta_i^+(x, y) \]

\[ \rho_i(x, y) = \sum_{y=3}^{4} \max \delta_i^-(x, y) - \sum_{y=3}^{4} \min \delta_i^-(x, y) \]

\[ \sigma_i(x, y) = \phi_i(x, y) - \eta_i(x, y) \]

\[ \xi_i(x, y) = \phi_i(x, y) - \mu_i(x, y) \]

Then, \[ l_i = |[\sigma_i(x, y)]|, |\xi_i(x, y)|, |\theta_i(x, y)|, |\rho_i(x, y)|. \tag{17} \]

The upper limit of the number of shared bikes at station \( i \) can be given as follows.

When station \( i \) is in missing during a certain period of time, the maximum number of shared bikes supplemented to station \( i \) during that period should be given under the condition that the lower bound of station \( i \) is satisfied. The number of shared bikes is the upper limit of this station. When station \( i \) is in excess state in the whole time period, the upper bound of station \( i \) is the difference between the “maximum number of shared bikes entering” and “minimum number of shared bikes leaving” in the day of this station under the condition that the lower bound is satisfied. There are two cases as follows.

For the convenience of formula display, let

\[ \alpha_i = \sum_{y=1}^{2} \delta_i^+(x, y) \]

\[ \kappa_i = \sum_{y=1}^{2} \delta_i^-(x, y) \]

\[ \nu_i = \sum_{y=3}^{4} \delta_i^+(x, y) \]

\[ \psi_i = \sum_{y=3}^{4} \delta_i^-(x, y) \]

**Case 1:** If station \( i \) is in missing during a certain period, i.e.,

\[ \sum_{y=1}^{2} \delta_i^+(x, y) - \sum_{y=1}^{2} \delta_i^-(x, y) \leq 0 \]

\[ \sum_{y=3}^{4} \delta_i^+(x, y) - \sum_{y=3}^{4} \delta_i^-(x, y) \leq 0. \]

Then,

\[ h_i = |\alpha_i| + l_i. \tag{18} \]

**Case 2:** If station \( i \) is in excess throughout the period, i.e.,

\[ \sum_{y=1}^{2} \delta_i^+(x, y) - \sum_{y=1}^{2} \delta_i^-(x, y) \geq 0 \]

\[ \sum_{y=3}^{4} \delta_i^+(x, y) - \sum_{y=3}^{4} \delta_i^-(x, y) \geq 0. \]

Then,

\[ h_i = |\max(\alpha_i, \nu_i) - \min(\kappa_i, \psi_i)| + l_i. \tag{19} \]

③ It can be concluded from Assumption 1 and Eqs. (14-15) in 2.2 that the travel of users is uncertain. In this paper, the demand for users’ uncertain travel is divided into static demand intervals, with 6 hours as a basic time unit, and a day is divided into four parts. At the same time, in order to conveniently calculate the shared bikes demand of each station in each different time period, this paper will use the average of the obtained travel data to indicate the shared bikes demand of the station during the time period.

The demand for shared bikes in different time periods of station \( i \) is as follows

\[ Q_i(y) = \frac{1}{n} \sum_{x=1}^{n} b_i(x, y), \quad y = 1, 2, 3, 4. \tag{20} \]

④ Since there are the upper and lower bounds for the number of shared bikes at each station, when the number of shared bikes in the station increases or decreases, the station may become a missing or excess station. According to Raviv et al. [37], when station \( i \) is a missing station, the missing number of shared bikes is as follow

\[ m_i = \max\{0, l_i - b_i - \sum_{y=z}^{z+1} \alpha_i(x, y) + \sum_{j=1}^{k} d_i(x, z)\}. \tag{21} \]

When station \( i \) is an excess station, the excess number of shared bikes is as follow

\[ n_i = \max\{0, b_i - h_i + \sum_{y=z}^{z+1} \alpha_i(x, y) - \sum_{j=1}^{k} d_i(x, z)\}. \tag{22} \]

⑤ The penalty cost of station \( i \) occurs in two cases. On the one hand, due to the insufficient number of shared bikes in station \( i \), users are dissatisfied with the merchants. This paper adopts missing penalty cost to quantify the satisfaction degree of users. On the other hand, due to the excessive number of shared bikes in station \( i \), the overcrowding in the station hinders the traffic and other situations. In this case, this paper will use the cost of excess punishment to show the excess of shared bikes caused by the social distress. According to Raviv et al. [37], the missing penalty cost of station \( i \) is

\[ P_i = p_i \times m_i, \tag{23} \]

and the excess penalty cost of station \( i \) is

\[ P_i = q_i \times n_i. \tag{24} \]
The scheduling cost of shared bikes from station \( i \) to station \( j \) is related to the number of shared bikes, the average scheduling cost and the fixed cost of shared bikes. According to Chemla et al. [7], the scheduling cost from station \( i \) to station \( j \) is as follows:

\[
C_{ij} = c_{ij} \times d_{ij} + f, \quad \forall i, j \in k, i \neq j, \quad (26)
\]

\[
C_{ij} = 0, \quad \forall i, j \in k, i = j. \quad (27)
\]

The total dispatching cost for all stations is as follows:

\[
C = \sum_{i=1}^{n} \sum_{j=1}^{n} (c_{ij} \times d_{ij} + f), \quad \forall i, j \in k, i \neq j. \quad (28)
\]

The scheduling distance between station \( i \) to station \( j \) is

\[
r_{ij} = \beta_{ij} \times r_{ij}, \quad \forall i, j \in k, i \neq j. \quad (29)
\]

The total dispatching distance for all stations is as follows:

\[
R = \sum_{i=1}^{k} \sum_{j=1}^{k} (\beta_{ij} \times r_{ij}), \quad \forall i, j \in k, i \neq j. \quad (30)
\]

An objective function for multi-objective integer programming model with the highest user satisfaction and the lowest shared bikes scheduling cost can be obtained. Owing to the complexity of solving this model, the following conversion is done.

As for objective function (1), Eqs. (21-22) show that the number of shared bikes in a certain period of time should be as much as possible to satisfy the demand in order to maximize users’ satisfaction. It can also be regarded as minimizing the punishment cost in Eqs. (23-25), which is as follow:

\[
\max \sum_{i=1}^{k} S_i = \min \sum_{i=1}^{k} P_i.
\]

For the objective function (3), aiming at the shared bikes scheduling distance, according to Eqs. (26-30), it can know that when the scheduling distance is shorter, the scheduling time cost and fuel consumption cost and so on will be lower. It can also be understood as converting the scheduling distance in Eq. (28) into a scheduling cost. In order to facilitate the establishment and calculation of the model, this paper will multiply the scheduling distance formula by a fixed coefficient to represent the scheduling cost at this distance, i.e.,

\[
R' = \gamma \times R = \gamma \sum_{i=0}^{k} \sum_{j=0}^{k} (\beta_{ij} \times r_{ij}), \quad \forall i \in k, j \in k, i \neq j,
\]

\[
\min R \rightarrow \min R'.
\]

Since \( \gamma \) is only a fixed coefficient, the value of \( \gamma \) has no effect on the distance \( r_{ij} \) between station \( i \) and station \( j \). It only converts the original distance formula into cost formula, which is convenient to establish and calculate the mathematical model.

As for objective function (2), the objective function of the scheduling cost of shared bikes, the number of shared bikes in a certain period of time should be as much as possible to satisfy the demand in order to minimize the scheduling cost. In other words, Eqs. (14-15) are used to make a more accurate prediction of the demand in each period of each day.

In summary, the objective function is replaced by the penalty cost as a substitute, the objective function (3) is quantified as the cost. Therefore, the original multi-objective integer programming model is converted into a single-objective integer programming model, and the weight factor is added to take into account decision action of decision maker.

\[
\min w = w_1 \sum_{i=0}^{k} \sum_{j=0}^{k} C_{ij} + w_2 \sum_{i=1}^{k} P_i + w_3 \gamma \sum_{i=0}^{k} \sum_{j=0}^{k} (\beta_{ij} \times r_{ij})
\]

\[
s.t. \ w_1 + w_2 + w_3 = 1
\]

\[
(4) \rightarrow (13)
\]

Three letters, \( w_1, w_2, w_3 \), indicate the weight of the three objective functions. Their value is between 0 and 1, and \( w_1 + w_2 + w_3 = 1 \). The specific value of each weight is determined by decision makers according to their own preferences and actual demands.

III. DATA ANALYSIS AND MODEL RESULTS

A. PROCESSING OF ORIGINAL DATA

Mobike has created the world’s first smart shared bikes model, using a new generation of Internet of Things technology, allowing users to locate and use the nearest Mobike anytime and anywhere. It also allows users to ride to the destination, parked nearby in the right area. Mobike was established in January 2015. On April 22nd, 2016, the Earth Day, the smart shared bikes service, Mobike, was launched officially in Shanghai. It has entered 19 countries including China, Singapore, Britain, Italy, Japan, Thailand and so on. In the city where it arrived, Mobike set off a wave of riding, promoted “Let the bicycle return to the city”, brought convenience to more people, and provided a sustainable solution for the city to promote green travel. This paper uses and analyzes the travel data of “Mobike” from May 10th to May 16th provided by “Mobike” in the “Mobike cup algorithm competition” in 2017. Firstly, we sort out more than 3 million pieces of travel data which is irregular to find out the travel rules of users. Secondly, we select the places that need to be scheduled in the sorted travel data, set them as stations.
These stations will wait for the arrangement of the scheduling plan. Finally, we obtain the preliminary scheduling plan by substituting the data into the mathematical model established in Section 2.2 for solution. Due to the different functional attributes of each station location, the peak period and the situation of missing and excess are also different.

1. In living areas, users usually use shared bikes to go to companies and public transport stations. Therefore, the number of shared bikes in the living areas presents the missing status in the morning, presents the balanced status in the afternoon, and presents the excess status in the evening.

2. In a public transportation station, the situation is relatively complicated. This paper divides public transport stations into the following two categories, user departure station and user arrival station. The status presented in both types is opposite. At the user departure station, the user usually uses a shared bike to reach the user departure station. Therefore, the number of shared bikes in the user departure station presents the excess status in the morning, presents the equilibrium status in the afternoon, and presents the missing status in the evening. The status of the user arrival station is opposite to the departure station.

3. In the work area or business district, users usually use shared bikes to go to the company or business district. As a result, the number of shared bikes in this region presents the excess status in the morning and afternoon, and presents the missing status in the evening.

In order to analyze the changing rules for users to use shared bikes more intuitively, based on the sorted original data and taking the variable $b_i$ in Eq. (14) as the parameter, line charts of users’ using shared bikes to travel at different time points throughout the day are shown by MATLAB software in Figs. 1-4.

Take Fig. 1 as an example. It is assumed that the number of initial vehicles in the station is 0. During peak hours between 6:00 and 8:00, users leave residential areas and use shared bikes to go to work. Therefore, the value of $b_i$ declines in a straight line, and the station is in in missing. The value of $b_i$ is fluctuating at the intersection of positive and negative numbers between 10:00 and 16:00. Since the fluctuation value of $b_i$ is small, it can be considered as an equilibrium state here. Users use shared bikes to return to residential areas from work.
areas between 17:00 and 19:00. Therefore, the value of \( b_i \) rises in a straight line, and the station is in an excess state. The situation described above is consistent with the nature of the residential area.

By sorting out and classifying more than 3 million pieces of travel data, this paper roughly divides the whole region into 6 station areas and 74 stations. Through data processing and analysis, 59 unbalanced stations and 15 balanced stations are finally identified. Among them, the data of these 59 unbalanced stations will be used as the input data in Section 3.2 to obtain the scheduling scheme.

### B. DATA SUBSTITUTION

According to the original data analysis results in Section 3.1, it can be concluded that the situations of stations with different natures in different time periods are different, but the situations of stations with the same natures in different time periods is basically consistent. At the same time, there is a certain volatility in the number of shared bikes on the same station at different dates and different times, indicating that the users’ travels are uncertain.

1) THE STATUS OF STATIONS AT A CERTAIN TIME

This paper sorts out the original data in Section 3.1 and combines it with Eqs. (14-15) to calculate the values of \( b_i \) and \( b_i(x, y) \). Then the meaning of values of \( b_i \) and \( b_i(x, y) \) can be interpreted according to Eqs. (14-15) and the status of 59 unbalanced stations in 4 time periods can be obtained according to the interpretation. As it is shown in Table 2 below.

It can be clearly seen from Table 2 that the 59 unbalanced stations are in the different status within the 4 fixed time periods. The specific unbalanced status of each station is shown in the Appendix Table 10. These 59 stations have been in an unbalanced status for four fixed periods, and they need to be artificially adjusted. At the same time, while imbalances of many stations change over time, only a few stations change remain constant.

2) CALCULATION OF THE UPPER AND LOWER BOUNDS OF STATIONS

This paper is based on the analysis results of the original data in Section 3.1, combined with the basic assumptions in Section 2.1.1 and the model in Section 2.2 for calculation. The upper and lower bounds of 59 unbalanced stations are determined as shown in Table 3 below.

It can be seen from Table 3 that the upper and lower bounds of the number of shared bikes can be accommodated by the 59 unbalanced stations. The specific upper and lower limits of the number of shared bikes that each station can accommodate are shown in the Appendix Table 11.

Although, the upper and lower bounds for each unbalanced station are explicitly determined in table 3. However, the upper and lower bounds for each station are not fixed values. Instead, it needs to be dynamically adjusted through the travel data record. Therefore, after the optimal scheduling scheme presented in this paper has been implemented for a period of time to obtain enough new data, the upper and lower bounds of each station are recalculated using the Eqs. (16-19). Then we modify the previously given optimal scheduling scheme by means of the new upper and lower bounds of each station so as to complete the dynamic adjustment of the shared bikes scheduling in considering of the uncertainty situation of the users’ traveling.

3) DETERMINATION OF SCHEDULING SCHEME

To determine the accurate optimal scheduling scheme, it is necessary to accurately determine two problems. First, the path problem is to determine the origin station and terminal station. Second, the quantity problem is how many shared bikes should be scheduled. Since it is difficult to solve the above two problems in one model, the scheduling path problem will be solved first, and then determine the optimal scheduling quantity based on the solved path.

4) SCHEDULING PATH PROBLEM SOLVING

Firstly, we find a point on the map as the origin and set up a rectangular coordinate system. Secondly, we determine the coordinates of the 59 unbalanced stations. Finally, we get the coordinates of each station. In order to facilitate the calculation, the origin selected in this paper makes all 59 stations in the first quadrant of the coordinate system. The coordinates of each station are shown in Table 4 below.

Table 4 shows the positions of the 59 unbalanced points in the coordinate diagram. The specific coordinates of each station are shown in Table 12 in Appendix. With the coordinate position of each station, the distance between each station can be calculated. The station distance formula is as follows.

\[
r_{ij} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}
\]

In this paper, “ant colony algorithm” [4] is used to calculate the shortest scheduling path. “Ant colony algorithm” is a heuristic algorithm for shortest path planning. The shortest path results can be obtained from the “ant colony algorithm” calculation in two cases. For Case 1, the number of iterations of the path calculation has reached the maximum iteration of the algorithm. For Case 2, the path calculation has reached the optimal solution within the maximum number of iterations.

In this paper, the following parameters are set to implement the ant colony algorithm for this problem, for example, the number of ants in the ant colony is 59, the maximum number of loops is 1000, the pheromones is 1, the heuristic factor is 5, the attenuation coefficient is 0.5 and the number of pheromones is 100 ([4], [38], [43]). Put the data in Table 4 into the operation code of “ant colony algorithm”, the following results can be obtained, as shown in Fig. 5 and Fig. 6.

According to Fig. 5, it can be clearly seen that the shortest path of dispatching shared bikes which is connected by line among 59 unbalanced stations. And the total length of the shortest path is 47.0042km through the operation program.
It can be clearly seen from Fig. 6 that the optimal path of this problem has been determined after about 140 iterations, and each subsequent iteration shows that the current path is the shortest scheduling path of this problem.

5) SCHEDULING QUANTITY PROBLEM SOLVING

Combined with the actual requirements, when the number of scheduling increase, although it can meet the demands of users, the company’s scheduling cost is too high and unrealistic. Nowadays, shared bikes companies schedule the shared bikes only once, at 0:00, and the rest of the time to schedule is random. Therefore, this paper integrates the actual situation, the lowest scheduling cost of the company, the highest users’ satisfaction and other requirements, and sets the scheduling times for two times according to the case that the day is divided into four time periods. In this paper, the scheduling time is set at 0:00 and 12:00 a.m., i.e., $z = 1, z = 3$.  

---

**TABLE 2.** Status of each station at different periods.

| Station | Early Morning | Morning | Afternoon | Evening |
|---------|--------------|---------|-----------|---------|
| wx4dn   | missing      | missing | missing   | missing |
| wx4dp   | excess       | excess  | missing   | missing |
| wx4dq   | excess       | excess  | missing   | excess  |
| wx4dr   | missing      | excess  | missing   | excess  |
| wx4ds   | missing      | missing | excess    | excess  |
| wx4dt   | excess       | excess  | missing   | excess  |
| wx4dv   | excess       | missing | missing   | missing |
| wx4sp   | excess       | missing | missing   | missing |

**TABLE 3.** Upper and lower bounds of each station.

| Station | Lower bound | Upper bound | Station | Lower bound | Upper bound |
|---------|-------------|-------------|---------|-------------|-------------|
| wx4dn   | 50          | 339         | wx4fb  | 120         | 1992        |
| wx4dp   | 40          | 273         | wx4fc  | 279         | 2327        |
| wx4dq   | 40          | 765         | wx4fd  | 16          | 568         |
| wx4dr   | 1           | 784         | wx4fe  | 80          | 441         |
| wx4ds   | 15          | 163         | wx4ff  | 113         | 2648        |
| wx4dt   | 46          | 605         | wx4fg  | 89          | 830         |
| wx4du   | 12          | 369         | wx4fh  | 114         | 710         |
| wx4dv   | 24          | 633         | wx4fk  | 88          | 509         |

**FIGURE 5.** Shared bikes scheduling path.

**FIGURE 6.** Path iteration result.
This paper takes Shanghai Mobike as an example and takes the average travel data of Mobike in one week as \( \delta^+ \) and \( \delta^- \). The number of initial shared bikes within the station is equal to the lower bound of the station, i.e., \( \alpha_i = l_i \). The exact values of the upper and lower bounds of each station, \( l_i \) and \( h_i \), are shown in Table 3. The total number of shared bikes is set at 60000, i.e., \( L = 60000 \). The maximum capacity of the scheduling path is 20000, i.e., \( U = 20000 \). The selection of scheduling paths for each station, the specific values of \( r_{ij} \) and \( \beta_{ij} \), are decided by Eqs. (29-30). Through the analysis of the practical situation of operating costs and profits of Mobike company, the average cost of shared bikes dispatching is 3 yuan per vehicle, i.e., \( c_{ij} = 3 \) (\( \forall i \in k, j \in k, i \neq j \)). The fixed cost of shared bikes dispatching is 20 yuan per time, i.e., \( f = 20 \). This paper considers that the decision maker holds an equilibrium attitude towards three decision variables, i.e., \( w_1 = w_2 = w_3 = 1/3 \). However, \( p_i \) and \( q_i \) will affect the decision of scheduling, so the following three cases will be considered.

**Case 1:** \( p_i = q_i \). For the convenience of formula display, let \( p_i = q_i = 1 \).

**Case 2:** \( p_i > q_i \). For the convenience of formula display, let \( p_i = 10, q_i = 1 \). At this point, the merchants will try their best to prevent the excessive number of shared bikes on the station even at the expense of users’ demands.

Put all the above parameter values into the mathematical model established in Section 2.2, the results can be obtained as shown in Table 5 below.

According to the value of \( Q_i \) in Table 5, it can be seen that these 59 stations have obvious unbalanced state in the two selected scheduling periods, which needs to be adjusted manually. Due to the excessive scheduling number of 59 stations, the detailed scheduling plan is shown in Table 13 in the Appendix. The value of \( Q_i \) reflects the number of unbalanced shared bikes in the station directly. At the same time, the positive or negative demand reflects the state of the station directly. By bringing the parameter values and the sorted data into the model, the number \( d \) of the shared bikes dispatched to the station in this case can be calculated. At the same time, the positive or negative of \( d \) represents the direction of dispatching and shared bikes. The positive represents the calling in and the negative represents the recalling.
TABLE 6. Data simulation results.

| Station | Precision | Accuracy | Adjustment scheme |
|---------|-----------|----------|-------------------|
| wx4dn   | 100%      | 100%     | -                 |
| wx4dp   | 100%      | 100%     | -                 |
| wx4dq   | 93%       | 100%     | -                 |
| wx4dr   | 100%      | 100%     | -                 |
| wx4ds   | 100%      | 100%     | -                 |
| wx4dt   | 93%       | 100%     | -                 |
| wx4du   | 100%      | 100%     | -                 |
| wx4dv   | 84%       | 93%      | Adjust the second schedule to 190 |

C. DATA SIMULATION TEST

According to the results shown in Table 5, a preliminary optimal scheduling scheme for “Mobike” shared bikes in Shanghai can be determined. In order to verify the correctness and accuracy of the scheme, R language and MATLAB 2018b are used to conduct data simulation to verify the proposed optimization scheduling scheme. This paper will simulate users’ uncertain travel conditions in 30 days to adjust the optimization scheme in Section 3.2 in order to reduce the error. Meanwhile, the correction is made for the place where there is a large deviation, and the final optimization plan is obtained by this. Data simulation results are shown in Table 6.

As the number of days for this data simulation is set at 30, the number of time samples for simulation is a bit small. The error of more than 2 days is relatively large in the whole and it can not be ignored. Therefore, when the error days of station are less than two days, it can be expressed that the optimization scheduling plan for the station has good correctness. On the contrary, it means that the correctness and accuracy of the optimized scheduling scheme are poor, and it is necessary to adjust the optimized scheduling scheme according to the data of simulation.

It can be intuitively reflected from Table 6 that the overall correctness and accuracy of the optimization scheduling and adjustment scheme proposed for Shanghai “Mobike” shared bikes is relatively high. It also shows that the optimal scheduling scheme calculated based on mathematical model and original data is reliable. However, in the initial optimal scheduling scheme, there are still four unbalanced stations where the scheduling scheme is not reasonable, and there is a significant deviation. Based on the results of data simulation, this paper makes reasonable adjustments to the scheduling schemes of the four unbalanced stations as shown in Table 6. Detailed simulation results are shown in Table 14 in the Appendix.

The three different cases are discussed as follow.

1. In the case of \( p_i = q_i \), the number of shared bikes that merchants are willing to dispatch basically satisfy the demands of users, and the scheduling scheme of 4 unbalanced stations is unreasonable. Correction schemes for unbalanced stations are given in Table 6. The number of shared bikes scheduled by this scheduling scheme is relatively small, so the scheduling cost is low.

2. In the case of \( p_i > q_i \), merchants are willing to increase the input of shared bikes in the station to ensure all the demands of users. Therefore, the accuracy of this scheduling scheme is the highest compared with the other two cases. In addition, according to the results of data simulation, this paper does not need to modify the scheduling scheme itself. However, the total number of vehicles in this scheduling scheme is larger than the other two schemes, so the scheduling cost is too high.

3. In the case of \( p_i < q_i \), merchants are willing to reduce the number of shared bikes and even sacrifice the demands of users to ensure that they do not reach the upper limit of the station. However, since the upper bound of each station calculated in Table 3 in this paper is basically greater than the scheduling number of scheduling schemes in case of \( p_i < q_i \), the scheduling scheme in case of \( p_i < q_i \) is basically the same as that in case of \( p_i = q_i \). Therefore, the result of this data simulation is the same as that in the case of \( p_i = q_i \). Table 6 shows the station scheduling plan that needs to be modified under this scheduling situation. At the same time, the number of shared bikes scheduled by this scheduling scheme is relatively small, so the scheduling cost is low.

In conclusion, in the case of considering users’ uncertain travel, the final optimal scheduling and adjustment scheme proposed in this paper for “Mobike” shared bikes in Shanghai is that the scheduling path shown in Fig. 5 and number of scheduled shared bikes shown in Table 6.

D. COMPARISON ANALYSIS AND DISCUSSION

Compared with the existing methods, this method proposed in this paper has the following differences. (1) Static demand interval is used to solve the shared-bike scheduling problem with users’ travel uncertainty whereas the existing methods which solve this problem seldom consider it. (2) This model takes into account the actual capacity of the stations for more
TABLE 7. Optimal scheduling quantities in the case of fixed station capacity.

| Station | Demand | Amount of scheduling |
|---------|--------|----------------------|
|         | 0:00   | 12:00                |
|         | P_i = q_i | P_i > q_i | P_i < q_i | P_i = q_i | P_i > q_i | P_i < q_i |
| wx4dn   | -117   | -103                | 117       | 141       | 117       | 103       | 124       | 103       |
| wx4dp   | 71     | -70                 | 0         | 0         | 0         | -1        | -1        | -1        |
| wx4dq   | 135    | 387                 | 0         | 0         | 0         | -135      | -135      | -135      |
| wx4dr   | 71     | -444                | 0         | 0         | 0         | 250       | 250       | 250       |
| wx4ds   | 1048   | 1183                | 19        | 23        | 19        | 0         | 0         | 0         |
| wx4dt   | 18     | 328                 | 0         | 0         | 0         | -250      | -250      | -250      |
| wx4dv   | 36     | -166                | 0         | 0         | 0         | 164       | 164       | 164       |

TABLE 8. Optimal scheduling quantities in the case of fixed stations capacity interval.

| Station | Demand | Amount of scheduling |
|---------|--------|----------------------|
|         | 0:00   | 12:00                |
|         | P_i = q_i | P_i > q_i | P_i < q_i | P_i = q_i | P_i > q_i | P_i < q_i |
| wx4dn   | -117   | -103                | 137       | 165       | 137       | 103       | 124       | 103       |
| wx4dp   | 71     | -70                 | 0         | 0         | 0         | 0         | 0         | 0         |
| wx4dq   | 135    | 387                 | 0         | 0         | 0         | -135      | -135      | -135      |
| wx4dr   | 71     | -444                | 0         | 0         | 0         | 179       | 179       | 179       |
| wx4ds   | 1048   | 1183                | 39        | 47        | 39        | 0         | 0         | 0         |
| wx4dt   | 18     | 328                 | 0         | 0         | 0         | -250      | -250      | -250      |
| wx4dv   | 36     | -166                | 0         | 0         | 0         | 150       | 180       | 150       |

TABLE 9. Data simulation comparison of scheduling scheme accuracy.

| Station | Scheduling scheme accuracy |
|---------|-----------------------------|
|         | Fixed Stations Capacity     | Fixed Stations Capacity interval |
|         | P_i = q_i | P_i > q_i | P_i < q_i | P_i = q_i | P_i > q_i | P_i < q_i |
| wx4dn   | 100%     | 100%     | 100%     | 100%     | 100%     | 100%     |
| wx4dp   | 93%      | 93%      | 93%      | 100%     | 100%     | 100%     |
| wx4dq   | 70%      | 70%      | 70%      | 83%      | 83%      | 83%      |
| wx4dr   | 100%     | 100%     | 100%     | 100%     | 100%     | 100%     |
| wx4ds   | 10%      | 17%      | 10%      | 30%      | 30%      | 30%      |
| wx4dt   | 47%      | 47%      | 47%      | 53%      | 53%      | 53%      |
| wx4dv   | 57%      | 57%      | 57%      | 60%      | 60%      | 60%      |
|         | 33%      | 33%      | 33%      | 39%      | 39%      | 39%      |

precise scheduling of shared bikes. Based on the historical travel data of the user, this paper calculates the upper and lower limits of each station to propose a more accurate shared bikes scheduling scheme.

Modifying the station capacity will have a certain impact on the scheduling scheme. Some previous studies have considered the shared bike station capacity as a fixed value [22] or a range with fixed value [14] to facilitate the calculation. Therefore, this paper will do the following simulation experiments to verify the advantages of this model compared with the previous model. First, we set all the factors to be the same except the station capacity. Secondly, according to the previous study, the station capacity was set to either a fixed value or a fixed interval. Therefore, two control groups
| Station | Early Morning | Morning | Afternoon | Evening |
|---------|--------------|---------|-----------|---------|
| wx4dn   | missing      | missing | missing   | missing |
| wx4dp   | excess       | excess  | missing   | missing |
| wx4dq   | excess       | excess  | excess    | missing |
| wx4dr   | missing      | excess  | missing   | excess  |
| wx4ds   | missing      | excess  | excess    | missing |
| wx4dt   | excess       | excess  | excess    | missing |
| wx4du   | missing      | missing | missing   | missing |
| wx4dv   | excess       | missing | missing   | missing |
| wx4dw   | excess       | missing | missing   | missing |
| wx4dx   | excess       | missing | excess    | missing |
| wx4dy   | missing      | missing | missing   | missing |
| wx4dz   | excess       | missing | missing   | missing |
| wx4e5   | excess       | excess  | missing   | missing |
| wx4eh   | missing      | excess  | missing   | excess  |
| wx4ej   | excess       | excess  | missing   | missing |
| wx4en   | missing      | missing | missing   | missing |
| wx4ep   | excess       | missing | missing   | missing |
| wx4eq   | missing      | excess  | missing   | missing |
| wx4ex   | missing      | missing | missing   | missing |
| wx4ey   | excess       | missing | missing   | missing |
| wx4f2   | missing      | missing | missing   | missing |
| wx4f3   | excess       | missing | missing   | missing |
| wx4f5   | missing      | excess  | missing   | missing |
| wx4f6   | missing      | excess  | missing   | missing |
| wx4f7   | excess       | excess  | missing   | missing |
| wx4f8   | excess       | missing | missing   | missing |
| wx4f9   | missing      | excess  | missing   | missing |
| wx4fb   | missing      | missing | missing   | missing |
| wx4fc   | missing      | missing | missing   | missing |
| wx4fd   | missing      | missing | excess    | missing |
| wx4fe   | missing      | missing | excess    | missing |
| wx4ff   | missing      | excess  | missing   | missing |
| wx4fg   | missing      | excess  | missing   | missing |
| wx4fh   | missing      | missing | missing   | missing |
| wx4fk   | excess       | excess  | missing   | missing |
| wx4fu   | missing      | missing | excess    | missing |
| wx4fv   | missing      | missing | missing   | balanced|
| wx4fy   | excess       | excess  | missing   | missing |
| wx4fz   | missing      | excess  | missing   | missing |
| wx4g0   | missing      | excess  | missing   | missing |
| wx4g1   | missing      | missing | missing   | missing |
| wx4g2   | missing      | missing | missing   | missing |
| wx4g3   | balanced     | missing | excess    | missing |
| wx4g4   | missing      | missing | excess    | missing |
| wx4g5   | missing      | missing | missing   | missing |
| wx4g6   | excess       | excess  | missing   | excess  |
| wx4g7   | excess       | missing | excess    | missing |
| wx4g8   | excess       | excess  | missing   | missing |
| wx4g9   | missing      | missing | missing   | missing |
| wx4gb   | missing      | excess  | missing   | missing |
| wx4gc   | excess       | missing | missing   | missing |
| wx4gd   | missing      | missing | missing   | missing |
| wx4gh   | excess       | missing | missing   | missing |
| wx4gn   | excess       | missing | missing   | missing |
| wx4sn   | missing      | excess  | missing   | missing |
| wx4sp   | excess       | missing | missing   | missing |
TABLE 11. Upper and lower bounds of each station.

| Station | Lower bound | Upper bound | Station | Lower bound | Upper bound |
|---------|-------------|-------------|---------|-------------|-------------|
| wx4dn   | 50          | 339         | wx4fb  | 120         | 1992        |
| wx4dp   | 40          | 273         | wx4fc  | 279         | 2327        |
| wx4dq   | 40          | 765         | wx4fd  | 16          | 568         |
| wx4dr   | 1           | 784         | wx4fe  | 80          | 441         |
| wx4ds   | 15          | 163         | wx4ff  | 113         | 2648        |
| wx4dt   | 46          | 605         | wx4fg  | 89          | 830         |
| wx4du   | 12          | 369         | wx4fh  | 114         | 710         |
| wx4dv   | 24          | 633         | wx4fk  | 88          | 509         |
| wx4dw   | 117         | 1380        | wx4fu  | 56          | 288         |
| wx4dx   | 43          | 779         | wx4fv  | 43          | 181         |
| wx4dy   | 114         | 1278        | wx4fy  | 1           | 292         |
| wx4dz   | 335         | 1946        | wx4fz  | 26          | 175         |
| wx4e5   | 161         | 1100        | wx4g0  | 8           | 1189        |
| wx4eh   | 93          | 1586        | wx4g1  | 33          | 2431        |
| wx4ej   | 93          | 1374        | wx4g2  | 78          | 1869        |
| wx4em   | 5           | 256         | wx4g3  | 1           | 1825        |
| wx4en   | 1           | 1384        | wx4g4  | 88          | 2720        |
| wx4ep   | 65          | 2682        | wx4g5  | 128         | 1457        |
| wx4eq   | 99          | 2078        | wx4g6  | 220         | 2030        |
| wx4ew   | 11          | 575         | wx4g7  | 3           | 141         |
| wx4ex   | 14          | 515         | wx4g8  | 6           | 462         |
| wx4ey   | 21          | 359         | wx4g9  | 81          | 632         |
| wx4ez   | 114         | 796         | wx4gb  | 83          | 268         |
| wx4f2   | 96          | 643         | wx4gc  | 86          | 734         |
| wx4f3   | 48          | 884         | wx4gd  | 9           | 865         |
| wx4f5   | 17          | 214         | wx4gh  | 286         | 1566        |
| wx4f6   | 11          | 361         | wx4gn  | 70          | 376         |
| wx4f7   | 25          | 478         | wx4sn  | 22          | 614         |
| wx4f8   | 337         | 2807        | wx4sp  | 54          | 565         |
| wx4f9   | 74          | 1975        |         |             |             |

will be set up. The station capacity is a fixed value in one control group, and the station capacity is a fixed interval in the other control group. When the fixed station capacity is set to a large size, although the users’ uncertainty travel is well handled, it does not necessarily match the actual situation of the number of vehicles that can be accommodated at the station. In addition, when the fixed station capacity is set to a small size, the users’ uncertainty travel cannot be well handled. Therefore, after using historical data for many trials and comparisons, we finally determined that the fixed capacity of the station will be set to 250 and the fixed capacity interval of the station will be set to [20, 250]. Finally, different scheduling schemes in different situations will be generated by the model proposed in this paper. We will conduct data simulation experiments on different scheduling schemes to verify the accuracy of the scheduling scheme.

The different scheduling schemes and the data simulation comparison for each case are shown in Table 7, Table 8 and Table 9 below.

Comparing Table 5, Table 7, and Table 8, we can easily find that the scheduling scheme of this problem has changed significantly under different cases of station capacity. And it can be intuitively reflected from Table 9 that the scheduling scheme that takes into account the actual station capacity is more accurate than others. The specific comparison of data simulation results are detailed in Table 15 in the Appendix. Among some stations, the randomness of the demand of several stations is not very strong, so the simulation results of the three scheduling schemes are accurate, such as the station of wx4dn and wx4dr. However, the users’ uncertainty travel is more obvious in some stations, such as the station of wx4ds, wx4dv and so on. The scheduling scheme that uses the station capacity calculated in this paper for these stations is more accurate than other scheduling schemes. Therefore, the model proposed in this paper can better solve the shared bikes scheduling problem with users’ travel uncertainty compared with the previous models. The model proposed in this paper and the calculated scheduling scheme are effective and reasonable.
### IV. CONCLUSION

In this paper, we have presented a new method for the shared bikes scheduling problem under users’ travel uncertainty. Solving this problem, a multi-objective integer programming model is established based on the consideration of static demand of fix time period, station capacity limit, penalty cost and other practical factors. In addition, this paper gives the basic formula to calculate the parameters in the model and an algorithm based on “ant colony algorithm” is also given to solve the model. Then taking the massive data provided by the “Mobike” company in 2017 as an example, this paper uses it to prove the feasibility and effectiveness of the model and get the initial optimization scheme. Finally, the optimization scheme is adjusted to obtain the final optimization scheme by the data simulation. The research results show that the final optimization scheme proposed in this paper has certain reference value for the scheduling problem of Shanghai “Mobike”.

This paper divides the time of a day into four static and unchanging periods, considering the reality and company’s scheduling cost. The users’ uncertain travel can be regarded as regular and continuous in each period. This paper predicts the variation of shared bikes in each time period of each station every day and then gives the reasonable scheduling plan according to the actual situation.

The upper and lower bounds for each station are clearly determined in this paper. In the research problem of this paper, the capacity of different stations is not the same and the capacity interval of each station is not fixed. The capacity interval of each station changes as the user’s travel data updates, and then the optimal scheduling scheme will be modified by the new upper and lower bounds of each station so as to complete the dynamic adjustment of the shared bikes scheduling in considering of the uncertainty situation of the users’ traveling.

The penalty cost caused by the excess vehicle quantity is considered so as to completely consider all the circumstances of the penalty cost of the shared bikes scheduling scheme. And the scheduling scheme of the shared bikes is adjusted to make the scheme more realistic through it.

However, the time of the second schedule is too regular in this paper. After all, the situation of each station is different in

| Station | Number | Coordinate(km)   | Station | Number | Coordinate(km)   |
|---------|--------|------------------|---------|--------|------------------|
| wx4dn   | 1      | (0.5, 6.4)       | wx4fb  | 31     | (4.2, 5.7)       |
| wx4dp   | 2      | (0.8, 5.2)       | wx4fc  | 32     | (3.8, 4.6)       |
| wx4dq   | 3      | (0.7, 3.8)       | wx4fd  | 33     | (4.0, 3.0)       |
| wx4dr   | 4      | (0.5, 2.5)       | wx4fe  | 34     | (4.2, 4.1)       |
| wx4ds   | 5      | (0.5, 0.6)       | wx4ff  | 35     | (4.5, 3.5)       |
| wx4dt   | 6      | (1.2, 4.2)       | wx4fg  | 36     | (4.6, 1.0)       |
| wx4du   | 7      | (1.0, 1.7)       | wx4fh  | 37     | (4.8, 2.8)       |
| wx4dv   | 8      | (1.7, 6.3)       | wx4fk  | 38     | (5.1, 2.1)       |
| wx4dw   | 9      | (1.6, 5.7)       | wx4fu  | 39     | (5.2, 3.3)       |
| wx4dx   | 10     | (1.7, 3.3)       | wx4fv  | 40     | (5.2, 4.1)       |
| wx4dy   | 11     | (1.3, 1.3)       | wx4fy  | 41     | (4.9, 4.9)       |
| wx4dz   | 12     | (2.1, 4.7)       | wx4fz  | 42     | (5.2, 6.4)       |
| wx4e5   | 13     | (2.1, 1.0)       | wx4g0  | 43     | (5.6, 3.7)       |
| wx4eh   | 14     | (2.0, 2.6)       | wx4g1  | 44     | (5.5, 2.6)       |
| wx4ej   | 15     | (2.5, 3.2)       | wx4g2  | 45     | (5.4, 1.2)       |
| wx4em   | 16     | (2.7, 2.3)       | wx4g3  | 46     | (5.3, 0.7)       |
| wx4en   | 17     | (3.0, 4.0)       | wx4g4  | 47     | (5.6, 1.9)       |
| wx4ep   | 18     | (3.0, 1.5)       | wx4g5  | 48     | (5.7, 5.8)       |
| wx4eq   | 19     | (3.2, 3.9)       | wx4g6  | 49     | (5.8, 4.8)       |
| wx4ew   | 20     | (3.1, 3.2)       | wx4g7  | 50     | (5.8, 2.7)       |
| wx4ex   | 21     | (3.2, 2.2)       | wx4g8  | 51     | (5.9, 1.5)       |
| wx4ey   | 22     | (3.6, 1.6)       | wx4g9  | 52     | (6.1, 3.3)       |
| wx4ez   | 23     | (3.9, 1.0)       | wx4gb  | 53     | (6.2, 6.3)       |
| wx4f2   | 24     | (2.5, 5.5)       | wx4gc  | 54     | (6.2, 4.2)       |
| wx4f3   | 25     | (2.7, 6.8)       | wx4gd  | 55     | (6.3, 6.9)       |
| wx4f5   | 26     | (3.1, 6.2)       | wx4gh  | 56     | (6.1, 5.4)       |
| wx4f6   | 27     | (3.0, 4.8)       | wx4gn  | 57     | (6.2, 2.1)       |
| wx4f7   | 28     | (3.7, 6.9)       | wx4sn  | 58     | (6.7, 3.3)       |
| wx4f8   | 29     | (3.7, 5.2)       | wx4sp  | 59     | (6.8, 4.0)       |
| wx4f9   | 30     | (4.3, 6.7)       |
### TABLE 13. Optimal scheduling quantities in three cases.

| Station | Demand | Amount of scheduling |
|---------|--------|----------------------|
|         | 0:00   | 12:00                |
|         | $p_i = q_i$ | $p_i > q_i$ | $p_i < q_i$ | $p_i = q_i$ | $p_i > q_i$ | $p_i < q_i$ |
| wx4dn   | -117   | -103                 | 117          | 141        | 117        | 103         | 124         | 103         |
| wx4dp   | 71     | -70                  | 0            | 0          | 0          | -1          | -1          | -1          |
| wx4dq   | 135    | 387                  | 0            | 0          | 0          | -135        | -108        | -135        |
| wx4dr   | 71     | -444                 | 0            | 0          | 0          | 373         | 448         | 373         |
| wx4ds   | 1048   | 1183                 | 0            | 0          | 0          | -1048       | -839        | -1048       |
| wx4dt   | 18     | 328                  | 0            | 0          | 0          | -324        | -260        | -324        |
| wx4du   | -94    | 20                   | 94           | 113        | 94         | 130         | 0           | 130         |
| wx4dv   | 36     | -166                 | 0            | 0          | 0          | -20         | -16         | -20         |
| wx4dw   | 151    | -133                 | 0            | 0          | 0          | -20         | -16         | -20         |
| wx4dx   | -25    | 76                   | 25           | 30         | 25         | 0           | 0           | 0           |
| wx4dy   | -146   | -147                 | 146          | 176        | 146        | 147         | 177         | 147         |
| wx4dz   | -486   | -737                 | 486          | 98         | 486        | 737         | 885         | 737         |
| wx4e5   | 177    | -224                 | 0            | 0          | 0          | 47          | 57          | 47          |
| wx4eh   | -19    | 17                   | 19           | 23         | 19         | 0           | 0           | 0           |
| wx4ej   | 166    | -175                 | 0            | 0          | 0          | 9           | 11          | 9           |
| wx4em   | -30    | -8                   | 30           | 36         | 30         | 8           | 10          | 8           |
| wx4en   | -18    | -79                  | 18           | 22         | 18         | 79          | 95          | 79          |
| wx4ep   | 378    | 60                   | 0            | 0          | 0          | -378        | -303        | -378        |
| wx4eq   | 116    | 51                   | 0            | 0          | 0          | -116        | -93         | -116        |
| wx4ew   | -74    | -45                  | 74           | 89         | 74         | 45          | 54          | 45          |
| wx4ex   | -9     | 44                   | 9            | 11         | 9          | 0           | 0           | 0           |
| wx4ey   | 114    | -89                  | 0            | 0          | 0          | -25         | -20         | -25         |
| wx4ez   | 171    | -300                 | 0            | 0          | 0          | 129         | 155         | 129         |
| wx4f2   | -196   | -164                 | 196          | 236        | 196        | 164         | 197         | 164         |
| wx4f3   | 1      | -122                 | 0            | 0          | 0          | 122         | 147         | 122         |
| wx4f5   | 115    | -7         | 0            | 0          | 0          | -118        | -95         | -118        |
| wx4f6   | 10     | -174                 | 0            | 0          | 0          | -164        | 197         | 197         |
| wx4f7   | 172    | -15                  | 0            | 0          | 0          | -152        | -122        | -152        |
| wx4f8   | -382   | -752                 | 382          | 459        | 382        | 752         | 903         | 752         |
| wx4f9   | 83     | -621                 | 0            | 0          | 0          | 538         | 646         | 538         |
| wx4fb   | -91    | -526                 | 91           | 110        | 91         | 526         | 632         | 526         |
| wx4fc   | -349   | -575                 | 349          | 419        | 349        | 575         | 690         | 575         |
| wx4fd   | -313   | 12                   | 313          | 376        | 313        | 0           | 0           | 0           |
| wx4fe   | -158   | -23                  | 158          | 190        | 158        | 23          | 28          | 23          |
| wx4ff   | 68     | -527                 | 0            | 0          | 0          | 459         | 551         | 459         |
| wx4fg   | 75     | -229                 | 0            | 0          | 0          | 154         | 185         | 154         |
| wx4fh   | -248   | -209                 | 248          | 298        | 248        | 209         | 251         | 209         |
| wx4fk   | 136    | 53                   | 0            | 0          | 0          | -136        | -109        | -136        |
| wx4fu   | -161   | -13                  | 161          | 194        | 161        | 13          | 16          | 13          |
| wx4fv   | -74    | -11                  | 74           | 89         | 74         | 11          | 14          | 11          |
| wx4fy   | 9      | -44                  | 0            | 0          | 0          | 35          | 42          | 35          |
| wx4fz   | 22     | -18                  | 0            | 0          | 0          | -4          | -3          | -4          |
| wx4g0   | 116    | 40                   | 0            | 0          | 0          | -116        | -93         | -116        |
| wx4g1   | 216    | -207                 | 0            | 0          | 0          | -9          | -8          | -9          |
| wx4g2   | -102   | -163                 | 102          | 123        | 102        | 163         | 196         | 163         |
| wx4g3   | -44    | -145                 | 44           | 53         | 44         | 145         | 174         | 145         |
| wx4g4   | -149   | -54                  | 149          | 179        | 149        | 54          | 65          | 54          |
| wx4g5   | -156   | -186                 | 156          | 188        | 156        | 186         | 224         | 186         |
| wx4g6   | 270    | 323                  | 0            | 0          | 0          | -270        | -216        | -270        |
| wx4g7   | -4     | 68                   | 4            | 5          | 4          | 0           | 0           | 0           |
| wx4g8   | 77     | -15                  | 0            | 0          | 0          | -62         | -50         | -62         |
| wx4g9   | -130   | -135                 | 130          | 156        | 130        | 135         | 162         | 135         |
| wx4gb   | -128   | 61                   | 128          | 128        | 128        | 0           | 0           | 0           |
| wx4gc   | 262    | -201                 | 0            | 0          | 0          | -61         | -49         | -61         |
| wx4gd   | 0      | -227                 | 0            | 0          | 0          | 227         | 273         | 227         |
| wx4gh   | 165    | -453                 | 0            | 0          | 0          | 288         | 346         | 288         |
| wx4gn   | 89     | -124                 | 0            | 0          | 0          | 35          | 149         | 35          |
| wx4sn   | -1     | -277                 | 1            | 2          | 1          | 277         | 333         | 277         |
| wx4sp   | -5     | -227                 | 5            | 6          | 5          | 227         | 273         | 227         |
TABLE 14. Data simulation results.

| Station | Accuracy | Adjustment scheme                        |
|---------|----------|-----------------------------------------|
|         | $p_i = q_i$ | $p_i > q_i$ | $p_i < q_i$ |
| wx4dn   | 100%     | 100%     | 100%       | -          |
| wx4dp   | 100%     | 100%     | 100%       | -          |
| wx4dq   | 93%      | 100%     | 93%        | -          |
| wx4dr   | 100%     | 100%     | 100%       | -          |
| wx4ds   | 100%     | 100%     | 100%       | -          |
| wx4dt   | 93%      | 100%     | 93%        | -          |
| wx4du   | 100%     | 100%     | 100%       | -          |
| wx4dv   | 84%      | 93%      | 84%        | Adjust the second schedule to 190 |
| wx4dw   | 93%      | 100%     | 93%        | -          |
| wx4dx   | 100%     | 100%     | 100%       | -          |
| wx4dy   | 100%     | 100%     | 100%       | -          |
| wx4dz   | 93%      | 100%     | 93%        | -          |
| wx4e5   | 93%      | 100%     | 93%        | -          |
| wx4eh   | 100%     | 100%     | 100%       | -          |
| wx4ej   | 100%     | 100%     | 100%       | -          |
| wx4em   | 93%      | 100%     | 93%        | -          |
| wx4en   | 93%      | 100%     | 93%        | -          |
| wx4ep   | 87%      | 100%     | 87%        | -          |
| wx4eq   | 93%      | 100%     | 93%        | -          |
| wx4ew   | 100%     | 100%     | 100%       | -          |
| wx4ex   | 93%      | 100%     | 93%        | -          |
| wx4ey   | 100%     | 100%     | 100%       | -          |
| wx4ez   | 100%     | 100%     | 100%       | -          |
| wx4f2   | 93%      | 100%     | 93%        | -          |
| wx4f3   | 93%      | 100%     | 93%        | -          |
| wx4f5   | 87%      | 93%      | 87%        | -          |
| wx4f6   | 93%      | 100%     | 93%        | -          |
| wx4f7   | 100%     | 100%     | 100%       | -          |
| wx4f8   | 93%      | 100%     | 93%        | -          |
| wx4f9   | 100%     | 100%     | 100%       | -          |
| wx4fb   | 93%      | 100%     | 93%        | -          |
| wx4fc   | 93%      | 100%     | 93%        | -          |
| wx4fd   | 93%      | 100%     | 93%        | -          |
| wx4fe   | 93%      | 100%     | 93%        | -          |
| wx4ff   | 100%     | 100%     | 100%       | -          |
| wx4fg   | 93%      | 100%     | 93%        | -          |
| wx4fh   | 93%      | 100%     | 93%        | -          |
| wx4fk   | 87%      | 100%     | 87%        | -          |
| wx4fu   | 93%      | 100%     | 93%        | -          |
| wx4fv   | 93%      | 100%     | 93%        | -          |
| wx4fy   | 93%      | 100%     | 93%        | -          |
| wx4fz   | 93%      | 100%     | 93%        | -          |
| wx4g0   | 87%      | 100%     | 87%        | -          |
| wx4g1   | 87%      | 93%      | 87%        | -          |
| wx4g2   | 80%      | 87%      | 80%        | Adjust the first schedule to 202 |
| wx4g3   | 80%      | 87%      | 80%        | Adjust the second schedule to 160 |
| wx4g4   | 77%      | 87%      | 77%        | Adjust the first schedule to 74 |
| wx4g5   | 93%      | 100%     | 93%        | Adjust the second schedule to 150 |
| wx4g6   | 100%     | 100%     | 100%       | -          |
| wx4g7   | 100%     | 100%     | 100%       | -          |
| wx4g8   | 93%      | 100%     | 93%        | -          |
| wx4g9   | 100%     | 100%     | 100%       | -          |
| wx4gb   | 93%      | 100%     | 93%        | -          |
| wx4gc   | 93%      | 100%     | 93%        | -          |
| wx4gd   | 93%      | 100%     | 93%        | -          |
| wx4gh   | 87%      | 93%      | 87%        | -          |
| wx4gn   | 100%     | 100%     | 100%       | -          |
| wx4sn   | 100%     | 100%     | 100%       | -          |
| wx4sp   | 93%      | 100%     | 93%        | -          |
| Station | Fixed Stations Capacity | Fixed Stations Capacity interval | Capacity interval calculated in this paper |
|---------|-------------------------|----------------------------------|--------------------------------------------|
|         | $p_i = a_i$ | $p_i > a_i$ | $p_i < a_i$ | $p_i = a_i$ | $p_i > a_i$ | $p_i < a_i$ |
| wx4dn   | 100% | 100% | 100% | 100% | 100% | 100% |
| wx4dp   | 93%  | 93%  | 93%  | 93%  | 100% | 93%  |
| wx4dq   | 70%  | 70%  | 70%  | 83%  | 83%  | 83%  |
| wx4dr   | 100% | 100% | 100% | 100% | 100% | 100% |
| wx4ds   | 10%  | 17%  | 10%  | 27%  | 30%  | 27%  |
| wx4dt   | 47%  | 47%  | 47%  | 53%  | 53%  | 53%  |
| wx4du   | 57%  | 57%  | 57%  | 60%  | 60%  | 60%  |
| wx4dv   | 33%  | 33%  | 33%  | 39%  | 39%  | 39%  |
| wx4dw   | 60%  | 67%  | 60%  | 73%  | 77%  | 73%  |
| wx4dx   | 100% | 100% | 100% | 100% | 100% | 100% |
| wx4dy   | 87%  | 93%  | 87%  | 97%  | 100% | 97%  |
| wx4dz   | 37%  | 37%  | 37%  | 50%  | 60%  | 50%  |
| wx4e   | 87%  | 87%  | 87%  | 93%  | 93%  | 93%  |
| wx4e   | 7%   | 7%   | 7%   | 27%  | 33%  | 27%  |
| wx4ej   | 0%   | 0%   | 0%   | 7%   | 10%  | 7%   |
| wx4em   | 87%  | 87%  | 87%  | 93%  | 93%  | 93%  |
| wx4en   | 87%  | 87%  | 87%  | 93%  | 93%  | 93%  |
| wx4eq   | 67%  | 70%  | 67%  | 77%  | 77%  | 77%  |
| wx4er   | 87%  | 87%  | 87%  | 93%  | 93%  | 93%  |
| wx4es   | 87%  | 87%  | 87%  | 93%  | 97%  | 93%  |
| wx4ey   | 83%  | 83%  | 83%  | 83%  | 87%  | 87%  |
| wx4ez   | 73%  | 73%  | 73%  | 83%  | 83%  | 83%  |
| wx4f   | 63%  | 50%  | 47%  | 93%  | 93%  | 93%  |
| wx4f   | 83%  | 83%  | 83%  | 93%  | 93%  | 93%  |
| wx4f   | 87%  | 87%  | 87%  | 93%  | 93%  | 93%  |
| wx4f   | 77%  | 80%  | 77%  | 83%  | 83%  | 83%  |
| wx4f   | 87%  | 87%  | 87%  | 93%  | 93%  | 93%  |
| wx4f   | 93%  | 93%  | 93%  | 93%  | 100% | 93%  |
| wx4f   | 7%   | 7%   | 7%   | 17%  | 23%  | 17%  |
| wx4f   | 13%  | 13%  | 13%  | 27%  | 27%  | 27%  |
| wx4f   | 37%  | 37%  | 37%  | 47%  | 50%  | 47%  |
| wx4f   | 0%   | 0%   | 0%   | 7%   | 10%  | 7%   |
| wx4f   | 73%  | 73%  | 73%  | 80%  | 80%  | 80%  |
| wx4f   | 83%  | 87%  | 83%  | 93%  | 93%  | 93%  |
| wx4f   | 47%  | 47%  | 47%  | 60%  | 60%  | 60%  |
| wx4f   | 83%  | 83%  | 83%  | 93%  | 93%  | 93%  |
| wx4f   | 70%  | 70%  | 70%  | 83%  | 83%  | 83%  |
| wx4f   | 87%  | 87%  | 87%  | 93%  | 93%  | 93%  |
| wx4f   | 87%  | 87%  | 87%  | 93%  | 93%  | 93%  |
| wx4f   | 93%  | 93%  | 93%  | 93%  | 93%  | 93%  |
| wx4fl  | 90%  | 90%  | 90%  | 93%  | 93%  | 93%  |
| wx4f  | 80%  | 83%  | 80%  | 93%  | 93%  | 93%  |
| wx4f  | 47%  | 50%  | 47%  | 83%  | 83%  | 83%  |
| wx4f  | 77%  | 77%  | 77%  | 83%  | 83%  | 83%  |
| wx4f  | 70%  | 70%  | 70%  | 93%  | 93%  | 93%  |
| wx4f  | 83%  | 83%  | 83%  | 93%  | 93%  | 93%  |
| wx4f  | 40%  | 40%  | 40%  | 93%  | 93%  | 93%  |
| wx4f  | 73%  | 93%  | 93%  | 93%  | 93%  | 93%  |
| wx4f  | 93%  | 93%  | 93%  | 93%  | 93%  | 93%  |
| wx4f  | 83%  | 83%  | 83%  | 93%  | 93%  | 93%  |
| wx4f  | 83%  | 83%  | 83%  | 93%  | 93%  | 93%  |
| wx4f  | 87%  | 87%  | 87%  | 87%  | 93%  | 87%  |
| wx4f  | 83%  | 83%  | 83%  | 93%  | 87%  | 93%  |
| wx4f  | 63%  | 67%  | 63%  | 77%  | 77%  | 77%  |
| wx4f  | 93%  | 93%  | 93%  | 93%  | 93%  | 93%  |
| wx4f  | 83%  | 83%  | 83%  | 93%  | 93%  | 93%  |
| wx4f  | 93%  | 93%  | 93%  | 93%  | 93%  | 93%  |
reality, so it needs to be adjusted reasonably in the following studies. Meanwhile, this paper calculates the lower and upper bounds of each station. Among them, the calculation of the lower bound $l_i$ is accurate, but the calculation of the upper bound $h_i$ does not take into account the station’s actual situation. For example, the station location, the station’s actual maximum accommodation and so on should be considered for the station’s upper bound. The accurate upper bound $h_i$ remains to the further research. Finally, we will consider other factors such as market competition and we will also collect different vendors’ data to test the method proposed in this paper for future research.

APPENDIX

See Tables 10–15.

REFERENCES

[1] A. J. King and S. W. Wallace, Modeling With Stochastic Programming, vol. 3. New York, NY, USA: Springer, 2012, pp. 153–165.

[2] B. P. Bruck, F. Cruz, M. Iori, and A. Subramanian, “The static bike sharing rebalancing problem with forbidden temporary operations,” Transp. Sci., vol. 53, no. 3, pp. 882–896, 2017.

[3] C. Contardo, C. Morency, and L. M. Rousseau, Balancing a Dynamic Public Bike-Sharing System, vol. 4. Montreal, QC, Canada: CIRRELT, 2012, pp. 181–196.

[4] C. Rajendran and H. Ziegler, “Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs,” Eur. J. Oper. Res., vol. 155, no. 2, pp. 426–438, Jun. 2004.

[5] C. Shui and W. Szeto, “Dynamic green bike repositioning problem-a hybrid rolling horizon artificial bee colony algorithm approach,” Transp. Res. D, vol. 60, pp. 119–136, May 2018.

[6] H. Dell’Amico, E. Hadjicostantinou, M. Iori, and S. Novellani, “The bike sharing rebalancing problem: Mathematical formulations and benchmark instances,” Omega, vol. 45, pp. 7–19, Jun. 2014.

[7] D. Chemla, F. Meunier, and R. W. Calvo, “Bike sharing systems: Solving the static rebalancing problem,” Discrete Optim., vol. 10, no. 2, pp. 120–146, 2013.

[8] D. J. Bertsimas and D. Simchi-Levi, “A new generation of vehicle routing research: Robust algorithms, addressing uncertainty,” Oper. Res., vol. 44, no. 2, pp. 286–304, 1996.

[9] D. Zhang, C. Yu, J. Desai, H. Y. K. Lau, and S. Srivathsan, “An integer l-shaped optimization modeling approach for the establishment of a bike sharing network: A case study of the city of Athens,” in Proc. Int. MultiConf. Eng. Comput. Sci., vol. 2, 2014, pp. 7–23.

[10] H. Hernández-Pérez and J. J. Salazar-González, “The one-commodity pickup and delivery traveling salesman problem,” Discrete Appl. Math., vol. 145, pp. 89–104, Jan. 2003.

[11] H. Hernández-Pérez and J. J. Salazar-González, “A branch-and-cut algorithm for a traveling salesmen problem with pickup and delivery,” Discrete Appl. Math., vol. 149, pp. 126–139, Dec. 2005.

[12] F. Louveaux and J. J. Salazar-González, “Heuristic algorithm for the split-demand one-commodity pickup-and-delivery travelling salesman problem,” Comput. Oper. Res., vol. 97, pp. 1–17, Sep. 2018.

[13] I. A. Forma, T. Raung, and M. Trau, “A 3-step math heuristic for the static repositioning problem in bike-sharing systems,” Transp. Res. B, Methodol., vol. 71, pp. 230–247, Jan. 2015.

[14] I. L. Wang and C.-W. Wang, “Analyzing bike repositioning strategies based on simulations for public bike sharing systems: Simulating bike repositioning strategies for bike sharing systems,” in Proc. 2nd IAI Int. Conf. Adv. Appl. Inform. (HATAI), Aug./Sep. 2013, pp. 306–311.

[15] I. L. Wang and J. Liu, “Dynamic lookahead policies for stochastic-dynamic inventory routing in bike sharing systems,” Comput. Oper. Res., vol. 106, pp. 260–279, Jun. 2019.

[16] J. J. Salazar-González and B. Santos-Hernández, “The split-demand one-commodity pickup-and-delivery travelling salesman problem,” Transp. Res. B, vol. 75, pp. 58–73, May 2015.

[17] J. Oyola, H. Arentz, and D. L. Woodruff, “The static vehicle routing problem, a literature review, part 1: Models,” Eur. J. Transp. Logist., vol. 7, pp. 193–221, Sep. 2018, doi: 10.1007/s10332-016-0100-5.

[18] J. R. Birge and F. Louveaux, Introduction to Stochastic Programming, vol. 3. New York, NY, USA: Springer, 2011, pp. 43–58.

[19] J. Schrijbroek, R. C. Hampshire, and W.-J. van Hoeve, “Inventory rebalancing and vehicle routing in bike sharing systems,” Eur. J. Oper. Res., vol. 257, no. 3, pp. 992–1004, Mar. 2016.

[20] L. D. Gaspero, A. Rendl, and T. Urli, “A hybrid ACO+CP for balancing bicycle sharing systems,” in Proc. Int. Workshop Hybrid Metaheuristics, New York, NY, USA: Springer, 2013, pp. 198–212.

[21] M. Gendreau, O. J. Ibarbi, and W. Rei, “Stochastic vehicle routing problems,” in Vehicle Routing: Problems, Methods, and Applications, Philadelphia, PA, USA: SIAM, 2014, pp. 213–239.

[22] M. Rainer-Harbach, P. Papazek, G. R. Raidl, B. Hu, and C. Klotzmüller, “PILOT, GRASP, and VNS approaches for the static balancing of bicycle sharing systems,” J. Global Optim., vol. 63, no. 3, pp. 597–629, 2015.

[23] J. Oyola, H. Arntzen, and D. L. Woodruff, “The stochastic vehicle routing problem,” in Routing: Problems, Methods, and Applications. Philadelphia, PA: SIAM, 2014, pp. 181–213.

[24] M. Iori, “Optimizing the level of service quality of a bisesharing system,” Omega, vol. 62, pp. 163–175, Jul. 2016.

[25] J. Oyola, H. Arntzen, and D. L. Woodruff, “The stochastic vehicle routing problem,” in Vehicle Routing: Problems, Methods, and Applications, Philadelphia, PA: SIAM, 2014, pp. 213–239.

[26] M. Rainer-Harbach, P. Papazek, G. R. Raidl, B. Hu, and C. Klotzmüller, “PILOT, GRASP, and VNS approaches for the static balancing of bicycle sharing systems,” J. Global Optim., vol. 63, no. 3, pp. 597–629, 2015.

[27] J. Oyola, H. Arntzen, and D. L. Woodruff, “The stochastic vehicle routing problem,” in Vehicle Routing: Problems, Methods, and Applications, Philadelphia, PA: SIAM, 2014, pp. 213–239.

[28] J. Oyola, H. Arntzen, and D. L. Woodruff, “An exact algorithm for solving the static repositioning problem in bike-sharing systems,” Transp. Res. E, vol. 72, pp. 192–209, Dec. 2014.

[29] S. C. Ho and W. Szeto, “Solving a static repositioning problem in bike-sharing systems using iterated tabu search,” Transp. Res. E, Logistics Transp. Res., vol. 69, pp. 180–198, Sep. 2014.

[30] J. Oyola, H. Arntzen, and D. L. Woodruff, “An exact algorithm for solving the static repositioning problem in bike-sharing systems,” Transp. Res. B, vol. 95, pp. 340–363, Jun. 2017.

[31] T. Bulhôes, A. Subramanian, G. Erdőgan, and G. Laporte, “The static bike relocation problem with multiple vehicles and visits,” Eur. J. Oper. Res., vol. 264, no. 2, pp. 508–523, 2018.

[32] T. Raviv, M. Tzur, and I. A. Forma, “Static repositioning in a bike-sharing system: Models and solution approaches,” EURO J. Transp. Logistics, vol. 2, no. 3, pp. 187–229, 2013.

[33] W.-H. Liao, Y. Kao, and C.-M. Fan, “Data aggregation in wireless sensor networks using ant colony algorithm,” J. Network Comput. Appl., vol. 31, no. 4, pp. 387–401, 2008.

[34] W. R. Stewart and B. L. Golden, “Stochastic vehicle routing: A comprehensive approach,” Eur. J. Oper. Res., vol. 14, no. 4, pp. 371–385, 1993.

[35] W. Szeto, Y. Liu, and S. C. Ho, “Chemical reaction optimization for solving a static bike repositioning problem,” Transp. Res. D, vol. 47, pp. 104–135, Aug. 2016.

[36] W. Szeto and C. Shui, “Exact loading and unloading strategies for the static multi-vehicle bike-repositioning problem,” Transp. Res. B, vol. 95, pp. 176–211, Mar. 2018.

[37] Y. Li, W. Szeto, J. Long, and C. Shui, “A multiple type bike repositioning problem,” Transp. Res. B, vol. 90, pp. 263–278, Aug. 2016.

[38] Z. Jiao, K. Ma, and Y. Li, “A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs,” J. Comput. Sci., vol. 25, pp. 50–57, Mar. 2018.
ZH-YONG ZHANG received the B.E. degree in logistics management from Southeastern University (SEU), Nanjing, China, in 2018. He is currently pursuing the M.S. degree in logistics engineering with Xidian University, Xi’an, China. His current research interests include decision analysis and operations research in the sharing economy.

XIAO ZHANG received the B.E. degree in industrial engineering and the M.S. and Ph.D. degrees in management science and engineering from Northeastern University (NEU), Shenyang, China, in 2007, 2009, and 2012, respectively. She is currently an Associate Professor with the Department of Management Engineering, School of Economics and Management, Xidian University, Xi’an, China. She is the author or coauthor of more than 30 articles published in international and local journals. Her current research interests include decision analysis and operations research.