2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)

Online Trip Planning for Public Bike Systems

Jiawei Mu, Jianhui Zhang, Tianyu Zhang, Bei Zhao*, Wanqing Zhang
Key Laboratory of Brain Machine Collaborative Intelligence of Zhejiang Province,
College of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018 China
*Corresponding author *e*-mail: zhaobei@hdu.edu.cn

2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS) | 978-1-7281-9866-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/MASS50613.2020.00069

***Abstract*—Public Bike System (PBS) not only provides convenient travel service but also alleviates the last-mile problem. With the increasing awareness of environmental protection and green commuting, people prefer to use the public bike as transportation for short-distance travel. However, the explosion of users in PBS leads to new congestion problems. To relieve the pressure of PBS, there are many types of research on system prediction, operation, and trip planning. However, there is few work focusing on the online trip planning problem. To study the case, we propose an Online Matching Trip Planning algorithm (OMTP), and we prove the theoretical lower bound of OMTP is 1 - 1/e. And then, we consider the short-term conflicts among users and design an Online Group Trip Planning algorithm (OGTP). We design two kinds of experiments- Generated Data Based and Real Data Based. In the generated data based experiment, we reveal the impact of different parameters with the generated trip data. In the real data based experiment, we validate our proposed algorithms with the real trip data set in New York City. The results show that OMTP and OGTP save time per trip on average.**

***Index Terms*—Public Bike System, Online Matching, Network Flow**

I. Introduction

Public Bike System (PBS) has become popular in many cities, which provide tremendous convenience for people's daily life [1]. As a new part of the urban public transportation system, PBS has eased the pressure of urban traffic congestion and air pollution problem. However, the explosive growth of bicycles in PBS also brings some potential problems to the cities.

Compare to other types of vehicles, the public bicycle has its unique characteristics. First of all, unlike traditional public transportation (e.g. taxis and buses), the public bicycle is unattended and shared. As time goes on, the public bicycles in PBS are assembled in a group of stations. Second, the usage frequency of public bicycles is susceptible to other external factors (e.g. weather), which makes it hard to predict in advance. These two features bring challenges to the management and the optimization of PBS.

Many researchers have paid attention to PBS, with the aim to promote its efficiency. In general, researches on public bike systems can be classified into four groups, *i.e.* system prediction, system operation, trajectory data analysis and bike trip planning. System prediction includes the bike availability predict and rent demand predict [2] [3] [4], which mainly relies on new technologies such as machine learning, and data mining. System operation mainly focuses on the bike reposition problem, which can be further divided into two camps: regularly transport by trunks [5] and incentive users to help with bike redistribution [6]. Further, with the emergence of wireless communication tools, there are a large number of bike trajectories generated every day. In recent years, some researchers try to excavate the value of these trajectories, e.g. detecting illegal parking vehicle [7], helping to plan bike lanes [8]. As far as we know, there is few work focusing on the bike trip planning problem. Unlike system prediction, system operation, and trajectory data analysis, the bike trip planning problem focuses on providing a convenient riding experience for each user. Hu propose a fine-grained prediction model and present a trip advisor that helps to balance bike usage [9]. Zhang *et al.* model the bike trip planning problem as a Bike Trip Selection (BTS) game and prove that the game is equal to a symmetric network congestion game, and develop a distributed algorithm to help users to select bike trip [10].

Li *et al.* focus on the large-scale bike trip planning problem that takes the three-segment of each bike trip into consideration [11]. In his model, all users' information is given and the resources at each station can be used only once, which is impractical since users' information is unknown in advance and resources at each station in real scenario change dynamically. To overcome the shortcomings of his model, this paper addresses a more practical problem, called the online bike trip planning problem (OTP). And we propose two algorithms: an Online Matching Trip Planning algorithm (OMTP) and an Online Group Trip Planning algorithm (OGTP), to solve this problem. In OMTP, the OTP problem is formulated as the well-known Online Matching problem, and we prove the theoretical lower bound of the OMTP algorithm is $1 - \frac{1}{e}$. In OGTP, we consider the short-term conflicts among users and reformulate the OTP problem based on the network flow model. Next, we design the OGTP algorithm to solve it. In the experiment part, we design two kinds of experiments- Generated Data Based and Real Data Based. In the generated data based experiment, we test the impact of different parameters based on the generated data, and then analyse the possible reason. In the real data based experiment, we analyze the real-world human mobility data set in New York City and validate our proposed algorithms with it.

Contribution. The main contributions of this paper are listed as follows:

1) Online bike trip model construction. We propose the online bike trip planning model and give the corresponding problem formulation, in which the complete three-segment bike trip is considered.

2155-6814/20/$31.00 ©2020 IEEE
DOI 10.1109/MASS50613.2020.00069

2) Online algorithm design. First, we design the OMTP algorithm and prove the theoretical lower bound of OMTP. Next, we consider the short-term conflicts among users and reformulate the problem based on the network flow model, and design the OGTP algorithm to solve it.

3) Experiment design and result analysis. We design two kinds of experiments and do extensive simulations to test the performance of two algorithms. Under each experiment result, we analyse the possible reasons.

Paper Structure. In Section II, we give the system model and formulate the online trip planning problem. In Section III and IV, we introduce the OMTP algorithm and the OGTP algorithm respectively. In Section V, we design two kinds of experiments and analyse the possible reasons. In Section VI and Section VII, we give some discussions and review some related work respectively. In Section VIII, we make a conclusion of the whole paper.

II. Problem Formulation

This section presents the system model in PBS when users arrive online, which is quite close to reality. We then formulate the online trip planing problem.

A. System Model

In this paper, we focus on the online public bike trip planning problem where users arrive online. We consider the situation of PBS over a time period T. Let $B = \{b_1, \cdots, b_N\}$ denote the the bike station set and $U = \{u_1, \cdots, u_M\}$ denote the user set, where N is the number of bike stations in PBS and M is the number of users who request the server to help them plan bike trip during the time period.

User. Each user u_i contains three parameters: a start location l_i^o, a terminal location l_i^t and the appear time ξ_i, denoted as $u_i = \{l_i^o, l_i^t, \xi_i\}$. Note that the appearance of users is stochastic and user can occur at any location at any time. But, the appear time of all users is in the range of the time period T. And we assume that the user's information will not change or cancel after uploading.

Bike Station. Each bike station b_i contains three parameters: its location l_i, the number of available bikes A_i^c and the number of empty docks A_i^s. In real life, A_i^c and A_i^s change dynamically as the user borrows or returns a bicycle. But the sum of A_i^c and A_i^s is a fixed number, which is equal to the total docks of station b_i. For user u_i, he can borrow a bike from one station and return it to another station. The bike station where user u_i can borrow one bike is the start station b_i^o, while the station where user u_i can return it is the target station b_i^t. For user u_i, he may have more than one start station and target station, we use B_i^o and B_i^t to represent his start station set and target station set, respectively.

Bike Trip. To make PBS more intelligent, we consider that each bike trip contains three segments. As shown in Fig. 1, one user issues a request at the start location, and then he borrows a bike in the start station. After that, he returns the bike to the target station and walks to the terminal location. And we give the definition of bike trip as follows:

Definition 1(Bike Trip): A bike trip h_i for u_i contains three segments: one riding segment from the start station b_i^o to the target station b_i^t and two walking segments, one from user's start location l_i^o to the start station b_i^o and the other from the target station b_i^t to his terminal location l_i^t. The bike trip can be denoted as $h_i = (l_i^o, b_i^o, b_i^t, l_i^t)$. And we use the trip set $H_i = \{h_1, h_2 \dots h_K\}$, to represent all potential trips.

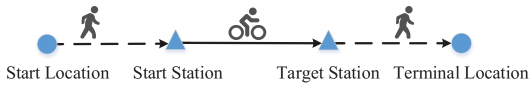


Fig. 1. The Three Segments of Trip

Time Cost. The time cost of each segment is associated with two parameters: its distance and velocity. For each segment, there may have more than one available path. However, different path might have different distance. To simplify the problem, we suppose that users are so lazy that they tend to choose the shortest path to save time, thus the minimal distance among all available paths is set as the segment distance.

For user u_i, the bike trip h_i contains three segments: one riding segments and two walking segments. Therefore, the time cost $E(h_i)$ of trip h_i contains three parts: the first one is the walking time w_i from start location to start station, the second one is the riding time v_i from start station to target station, and the third one is the walking time w_i' from target station to terminal location. The time cost $E(h_i)$ can be calculated by the following equation.

$$E(h_i) = w_i + v_i + w_i' \tag{1}$$

B. Problem Formulation

In this paper, the process of online planning trip for users, is called Online Trip Planning problem (OTP). To be specific, the OTP problem is how to online plan the optimal trip h_i for the arrived user, with aim to improve his trip quality.

For each user, his trip quality is related with two factors: 1) the three-segment trip time; 2) the success of three-segment trip. In the OTP problem, each trip time contains three parts and both the bike borrowing and returning should be successful. However, the available bikes and empty docks at bike stations are time-varying, which makes the problem complex.

Table I lists the major symbols used throughout this paper.

TABLE I
Symbol and Description

Sym.	Description	Sym.	Description
u	User	U	User set
b	Bike station	B	Bike station set
p	Resource pair	P	Resource pair set
h	Trip	H	Trip set
A_i^c	b_i's available bikes	A_i^s	b_i's available docks
E	Trip time	w	Walk time
v	Ride time	d_{max}	Maximal walking distance
T	Time period	τ	Time slot
G	Bike route graph		

III. Online Matching Trip Planning

In this section, we introduce the Online Matching Trip Planning algorithm (OMTP) and prove the theoretical bound of OMTP.

A. Basic Idea

Intuitively, the success of complete three-segment bike trip can be determined by giving the bike to borrow and the dock to return. Thus, we regard the bike in the start station and the dock in the target station as one bike-dock pair. And, we give the definition of bike-dock pair set as follows:

Definition 2(Bike-Dock Pair Set): A bike-dock pair set P_i of user u_i is the collection of all available bike-dock pairs. Each bike-dock pair p_j contains two parameters: the first one is the available bike c in the start station b_i^o, and the second one is the empty dock s in the target station b_i^t.

$$P_i = \{p_j = (c, s) | c \in b_i^o, b_i^o \in B_i^o, s \in b_i^t, b_i^t \in B_i^t\} \quad (2)$$

In Fig. 2, there are two bikes in the station b_1 and two docks in the station b_2. Thereby, the bike-dock pair set P_i of user u_i can be denoted as $P_i = \{p_1, p_2, p_3, p_4\}$. And the complete trip h_i for user u_i can be determined with the selected bike-dock pair. For example, if pair p_1 is selected, the complete trip of user u_i can be denoted by $h_i = (l_i^o, b_1(c_1), b_2(s_2), l_i^t)$.

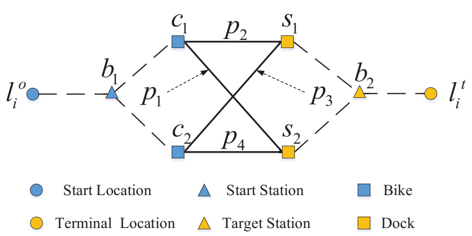


Fig. 2. The Bike-Dock Pair

To improve the trip quality, the quality $Q(p_j)$ of the bike-dock pair p_j is set to the inverse of three-segment trip time, which is denoted by the following equation:

$$Q(p_j) = \frac{1}{w + v + w'} \quad (3)$$

With the above analysis, the OTP problem can be formulated as the well-known online matching problem, which can be described as: the problem is how to match the optimal bike-dock pair p_j with user u_i. The online trip planning problem based on online matching can be formulated as:

$$\max \sum_{j=1}^{|P_i|} Q(p_j) x_{p_j} \quad (4)$$

$$\textbf{s.t.} \quad \sum_{j=1}^{|P_i|} x_{p_j} = 1 \quad (5)$$

$$|P_i| \geq 1 \quad (6)$$

Here, Eq. (5) means that only one bike-dock pair can be allocated to u_i. x_{p_j} is a decision variable: x_{p_j} = 1 if p_j is allocated to u_i; otherwise x_{p_j} = 0. Eq. (6) means that there should be at least one existing bike-dock pair for user u_i, or u_i is failed.

B. Algorithm Design

Intuitively, in order to achieve high trip quality, OMTP matches the bike-dock pair which has the maximal quality with the online-arrived user. That is the main idea of OMTP. Note that each bike-dock pair is locked after allocating, which means each bike-dock pair can be allocated for only once. OMTP is detailed in Algorithm 1.

Algorithm 1 Online Matching Trip Planning

Input: User: u_i;
Output: Trip h_i

1: Get start station set B_i^o and target station set B_i^t for u_i;
2: Construct bike-dock pair set P_i for u_i;
3: **for** bike-dock pair p_j in P_i **do**
4:     Calculate the bike-dock pair quality $Q(p_j)$ for p_j;
5:     Record the bike-dock pair quality $Q(p_j)$ for p_j;
6: **end for**
7: Find the bike-dock pair p_j^* which has the maximum quality $Q(p_j^*)$ among P_i;
8: Construct the trip h_i based on p_j^*;
9: Output trip h_i;

Specifically, OMTP finds the start station set B_i^o and the target station set B_i^t for user u_i. In the meanwhile, OMTP constructs the bike-dock pair set P_i for user u_i based on Eq. (2). For each bike-dock pair p_j in P_i, OMTP calculates its bike-dock quality $Q(p_j)$ based on Eq. (3) and record it. After that, OMTP finds the bike-dock pair p_j^* in P_i which has the maximum quality $Q(p_j^*)$. Then, OMTP constructs the trip h_i based on the bike-dock pair p_j^*. In the end, OMTP outputs the trip h_i, then the algorithm stops.

C. Time Complexity Analysis and Pruning Strategy

In this part, we analyze the time complexity of this OMTP algorithm, and then we present the pruning strategy to reduce time complexity.

In Algorithm 1, get the start station set and the target station set takes $2N$ steps. And we denote the size of the start station set, the size of the target station set by S, T. In line 2, construct bike-dock pair takes $S \times T$ steps because each start station and each target station are traversed. After constructing, we suppose the size of the bike-dock pair is K. Next, in line 3-6, calculate and recode the quality takes total $2K$ steps due to it traverses the whole set. Similarly, it takes K steps to find the maximal bike-dock pair in line 7. And construct the trip in line 8 and output the trip in line 9 take 2 steps. Thus, the sum of the OMTP algorithm takes $2N + S \times T + 3K + 2$ steps. Therefore, the time cost of this OMTP algorithm is given by $O(S \times T)$.

Through the above analysis, we know the time cost of the OMTP algorithm is related to the value of S and T. Combined with real-life experience, users prefer to choose the adjacent bike stations to borrow and return. Therefore, we set the maximal walking distance d_{max} for users. And users can only look for the start stations and target stations within the range of d_{max}. Consequently, the time cost of the OMTP algorithm can be reduced rapidly.

D. Algorithm Performance Analysis

In OMTP, each bike-dock pair doesn't take part in the subsequent request processing after allocating. Suppose the number of initial bike-dock pairs in PBS is n. And the number of users that OMTP can satisfy in the ideal case is also n.

In the ideal case, there are n users and n available bike-dock pairs, and they form a matching matrix. The k^{th} user's request corresponds to the $n-k+1$ column of matrix. By the Karp's proving, the expected matching result of the online matching algorithm is same as that of the random matching algorithm on this matrix [12]. And the strategy of the random matching algorithm is to randomly select one bike-dock pair for current user. Therefore, the worst case's performance of OMTP can be evaluated by proving the expected performance of the random matching algorithm, and the theoretical lower-bound of OMTP is described as follows.

Suppose that the matching matrix of the random matching algorithm in the worst case is M. When the x^{th} user comes, there are still y eligible bike-dock pairs in M, which are likely be any set of y lines in the first $n-k+1$ rows of M. At time t, the number of remaining users and the number of remaining bike-dock pairs are denoted by $\alpha(t)$ and $\beta(t)$, where $\Delta\alpha = \alpha(t+1) - \alpha(t)$ and $\Delta\beta = \beta(t+1) - \beta(t)$. OMTP algorithm processes one user's request each time, thus $\Delta\alpha = -1$. If the diagonal entry in the $t+1^{st}$ column is eligible but is not matched, $\Delta\beta = -2$, otherwise $\Delta\beta = -1$. Due to the set of remaining eligible bike-dock pairs is randomly chosen from among the first $n-t$, we get the following equation:

$$E[\Delta\beta] = -1 - \frac{\beta(t)}{\alpha(t)} \times \frac{\beta(t)-1}{\beta(t)} = -1 - \frac{\beta(t)-1}{\alpha(t)} \quad (7)$$

Next, we get the new equation:

$$\frac{E[\Delta\beta]}{E[\Delta\alpha]} = 1 + \frac{\beta(t)-1}{\alpha(t)} \quad (8)$$

The possibility of having k available bike-dock pairs at time t of OMTP is the same as that of the random matching algorithm. Based on the proof proposed by Karp [12], when the possibility tends to 1 and n tends to infinity, Eq. (8) can be transformed into the following equation:

$$\frac{d\beta}{d\alpha} = 1 + \frac{\beta - 1}{\alpha} \quad (9)$$

Meanwhile, suppose $\alpha = \beta = n$, we get

$$\beta = 1 + \alpha\left(\frac{n-1}{n} - \ln\frac{\alpha}{n}\right) \quad (10)$$

Obviously, when only one bike-dock pair is eligible, the remaining user number is $\frac{n}{e} + o(n)$. Therefore, the expected matching number is $n(1-\frac{1}{e}) + o(n)$. Thus, the lower-bound of the OMTP algorithm is $1 - \frac{1}{e}$.

Theorem 1: The theoretical lower-bound of OMTP is $1-\frac{1}{e}$.

IV. Online Group Trip Planning

In this section, we introduce the Online Group Trip Planning problem (OGTP) and give the problem formulation based on the network flow model. And then, we design a heuristic algorithm to solve it.

A. Discussion

In the public bike services, the users keep coming and the system online allocates bike-dock pair for the arrived user. However, the resources at bike stations are restricted, including the available bikes and empty docks. Besides, the resources at each bike station may become unbalanced as time goes on, due to each bike station has different inflows and outflows. Therefore, the conflicts among users are inevitable, especially in the rush hour.

Take an example, there are two users uploading the request to the system simultaneously, namely u_1, u_2. User u_1 and user u_2 have similar start location and they share the same start station b_1. Unfortunately, user u_2 has only one avaible station to borrow which is b_1. However, there is only one bike in station b_1. If we pre-allocate the bike in station b_1 to user u_1, there are no available bikes for user u_2, and u_2 is failed. Therefore, there exists conflicts among u_1 and u_2. In this problem, we call the station b_1 is the conflicting station and user u_1, u_2 are two conflicting users. Similarly, if user u_1 and u_2 share the same target station which has only one dock, they can also become conflicting users.

Intuitively, if we allocate trips for a group of user at one time, we can improve overall trip quality. Thus, we divide the time period T into L equal-length time slots and each slot $\tau_l \in T, l = 1, 2, \cdots, L$. And the user set U_l is the collection of appeared users in one time slot τ_l.

B. Bike Route Graph

In this part, we introduce the bike route graph and the specific process of drawing a bike route graph. And then, we give the OGTP problem based on the bike route graph.

Bike Route Graph. Having the bike station set B, the user set U_l and the maximal walking distance d_{max}, we can draw users' potential bike routes from their start locations to target locations on a new graph, called the bike route graph G_l. In Fig. 3, there is an example of bike route graph, which contains three users. The solid circles and triangles are the nodes in the graph, in which the triangles represent the bike station. And we use the dotted lines and the solid lines to represent users' walking segments and riding segments, respectively. The line from user's start location to his target location is one potential trip. In the figure, u_1 has four potential trips and u_2 has two potential trips. For each bike station node b_i on G_l, we add one virtual node b'_i and a virtual edge $e(b_i, b'_i)$ connecting them.

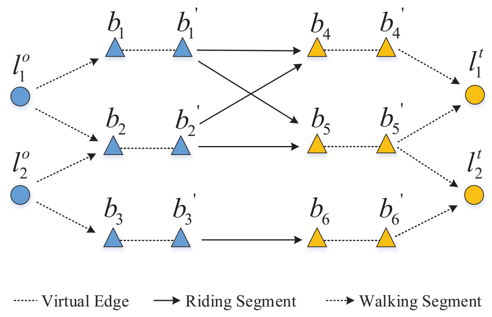


Fig. 3. An example of bike route graph

Bike Route Graph Drawing. The bike route graph shows all the users' potential routes among stations. Next, we introduce the specific process of drawing a bike route graph:

1) During each time slot, online-arrived users upload their personal information to the server, including start location, target location and appear time. And the server collects users' uploaded information, records their appear time and adds them into the user set U_l.

2) At the end of time slot, we calculate and find user's start station set and target station set within the range of the maximal walking distance d_{max}. Besides, we filter out those users who can't find start station or target station. Next, we update the resources at each bike station according to the current situation in PBS, including the available bikes and the empty docks.

3) Next, we get all potential bike trips for filtered users and draw their potential bike trips on G_l. In the following context, we set the capacity A_e and cost C_e of edge $e(i,j)$:

Capacity. The capacity of the virtual edge connecting the start station and its virtual node is set as the number of available bikes of this bike station. While the capacity of the virtual edge connecting the target station and its virtual node is set as the number of empty docks. For example, in Fig. 3, the capacity of edge $e(b_1, b_1')$ is assigned with the available bike station of station b_1. The capacity of edge $e(b_5, b_5')$ is assigned with the empty docks of station b_5. The initial capacity A_e of edge $e(i,j)$ is defined as follows:

$$A_e = \begin{cases} A_s^c & i = b_s, j = b_s' \\ A_t^s & i = b_t, j = b_t' \\ \infty & Otherwise \end{cases} \tag{11}$$

Cost. The initial cost of each edge is set to the time cost between two vertices, which can be calculated by dividing the trip distance by the walk (ride) velocity. For example, in Fig. 3, the cost of edge $e(l_1^o, b_1)$ and edge $e(b_5', b_1^t)$ is set as walking time, while the cost of edge $e(b_1', b_5)$ is set as riding time. The initial cost C_e of edge $e(i,j)$ is defined as follows:

$$C_e = \begin{cases} w & i = l_k^o, j = b_s \\ w' & i = b_t', j = l_k^t \\ v & i = b_s', j = b_t \\ 0 & Otherwise \end{cases} \tag{12}$$

where b_s and b_t represent the start station symbol and the target station symbol, respectively. For example, in Fig. (4), b_1, b_2 and b_3 belong to b_s, while b_4, b_5 and b_6 belong to b_t. l_k^o and l_k^t are user u_k's start location and target location, respectively. The time cost $E(h)$ of one potential trip h for u_k from the start location l_k^o to the target location l_k^t on G_l is redefined by the following equation:

$$E(h) = C_{(l_k^o, b_k^o)} + C_{(b_k^o, b_k^o{}')} + C_{(b_k^o{}', b_k^t)} + C_{(b_k^t, b_k^t{}')} + C_{(b_k^t{}', l_k^t)} \tag{13}$$

After the above steps, we have the filter user set U_l, the bike route graph G_l, the time cost $E(h)$ of each potential trip. This can lead to the OGTP problem as follows.

Given the initialized bike route graph G_l, the OGTP problem is how to plan trips for users on G_l, with aim to improve overall trip quality under the resource restricts.

C. Problem Formulation

The OGTP problem is an integer multi-commodity flow (IMCF) problem [13]. The bike route graph G_l is a directed graph. On the graph G_l, the size of nodes is n and the number of edges is m. There are K users needed to be sent from their start locations to terminal locations.

The formulation of the OGTP problem based on IMCF is given as follows:

$$\min C^T \sum_{k=1}^{K} X^k \tag{14}$$

$$\textbf{s.t.} \sum_{k=1}^{K} X^k \leq A \tag{15}$$

$$DX^k = d^k, k = 1, 2, ..., K \tag{16}$$

$$X^k \geq 0, k = 1, 2, ...K \tag{17}$$

where Eq. (14) is a target function with aim to minimize overall trip time. Eq. (15) is the edge maximum flow constraint. Eq. (16) is the node flow equilibrium equation. Eq. (17) is the flow non-negative constraint.

Here, $X^k = [x_1, x_2, \cdots, x_m]$ is the flow vector of user k, $C = [C_1, C_2, \cdots, C_m]^T$ and $A = [A_1, A_2, \cdots, A_m]^T$ are the cost and capacity vectors of all edges, respectively. $D = [D_{il}]_{n \times m}$ is the incidence matrix between nodes and edges. For the l^{th} edge $e(i,j)$, setting $D_{il} = 1$ and $D_{jl} = -1$. The node flow vector $d^k = [d_1^k, d_2^k, ..., d_n^k]$, is defined as follows:

$$d_i^k = \begin{cases} 1 & i = l_k^o \\ -1 & i = l_k^t \\ 0 & Otherwise \end{cases} \tag{18}$$

D. Algorithm Design

In the OGTP problem, we plan trips for a group of users at one time, which eases short-term conflicts to some extent. Besides, the target of OGTP is to minimize trip time of all users, with aim to improve the overall trip quality. OGTP is a NP-C problem. And we design a heuristic algorithm to find a feasible solution, named the OGTP algorithm.

Intuitively, with aim to minimize the overall trip time, we greedily and iteratively plan the shortest-time trip for each user. For user u_k, we find his shortest-time trip based on Eq. (13), which is $(l_k^o, b_k^o, {b_k^o}', b_k^t, {b_k^t}', l_k^t)$. Once the capacity A_e of edge $e(b_k^o, {b_k^o}')$ is equal to 0, we increase C_e of edge $e(b_k^o, {b_k^o}')$ and replan the affected trips. The same operation for edge $e(b_k^t, {b_k^t}')$. The algorithm keeps on running until find a feasible solution. Finally, the allocated bikes and docks will be reserved for users. OGTP is detailed in algorithm 2.

Algorithm 2 Online Group Trip Planning

Input: Bike Route Graph: G_l;
Output: Trip Set: H_l;

The Unallocated User Set $UU = U_l$;
The Allocate Trip Set $H_l = \emptyset$;
while $UU \neq \emptyset$ **do**
  for user u_k in UU **do**
    Find the trip h_k: $(l_k^o, b_k^o, {b_k^o}', b_k^t, {b_k^t}', l_k^t)$ for u_k on G_l;
    if $A_{(b_k^o, {b_k^o}')}$ or $A_{(b_k^t, {b_k^t}')} \leq 0$ **then**
      Trip Subset $H = \emptyset$;
      for trip $h_i : (l_i^o, b_i^o, b_i^{o\prime}, b_i^t, b_i^{t\prime}, l_i^t)$ in H_l **do**
        if $\{b_i^o == b_k^o\}$ or $\{b_i^t == b_k^t\}$ **then**
          $A_{(b_i^o, {b_i^o}')}$ += 1, $A_{(b_i^t, {b_i^t}')}$ += 1;
          Add h_i into H;
        end if
      end for
      Delete H from H_l;
      $C_{(b_k^o, {b_k^o}')}$ or $C_{(b_k^t, {b_k^t}')}$ += o;
    else
      $A_{(b_k^o, {b_k^o}')}$ -= 1, $A_{(b_k^t, {b_k^t}')}$ -= 1;
      Add h_k into H_l and delete u_k from UU;
    end if
  end for
end while
Output the trip set H_l;

Note that the increasing number o is set by us. And we set a cost threshold D for each virtual edge to avoid no solution problem. Once C_e of edge e exceeds D, we will stop the algorithm and greedily allocate trips for unallocated users until the resources are exhausted.

Take an example, in Fig. 4, the cost and capacity of edge e are represented by $C_e(A_e)$. We find the shortest trip for user u_1, which is $(l_1^o, b_1, b_1', b_4, b_4', l_1^t)$. And the available bike in b_1 and the empty dock in b_4 are pre-allocated for u_1. Then, we find the shortest trip for u_2, which is $(l_2^o, b_1, b_1', b_6, b_6', l_2^t)$. And the bike in b_1 has been allocated to u_1, which indicate that there exists conflict between u_1 and u_2. Intuitively, this conflict can be solved by increasing $C_{(b_1,b_1')}$ of edge (b_1, b_1'). If $C_{(b_1,b_1')}$ of edge (b_1, b_1') increase to 2, the shortest path of user u_1 is changed to $(l_1^o, b_2, b_2', b_4, b_4', l_1^t)$. Then, the conflict between u_1 and u_2 is solved. Similarily, if user u_2 chooses her shortest trip, there are no empty docks for user u_3. And this problem can be solved by increasing $C_{(b_6,b_6')}$ of edge (b_6, b_6').

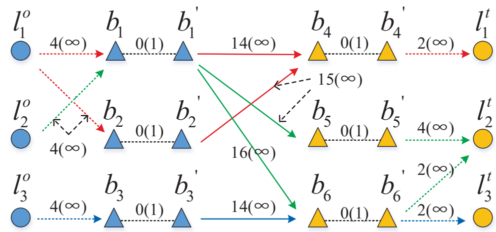


Fig. 4. An example of OGTP algorithm

V. Experiment Results

In this section, we design two kinds of experiments- Generated Data Based and Real Data Based, to test the performance of the above two algorithms.

A. Methodology

In this part, we introduce the bike station set and preset several relative parameters. Besides, we give the experimental measurements of this paper.

The bike station data set utilizes the real data in New York City. The data can be crawled from the net, which contains the information of each bike station, e.g. ID, location, total capacity. The number of bike stations in the set is 936. For each user, the walking speed is set as $5km/h$ [14], and the riding speed is set as $20km/h$ [15].

Measurements. We apply two experimental measurements to test the performance of our algorithms.

- **Average trip time (ATT)**: It's the average trip time cost of successful users who arrive at their target locations.
- **Successful Service Ratio (SSR)**: It's the ratio of successful users among all users in time period T, which is calculated by the number of successful users divided by the total user number.

B. Generated Data Based Experiment and Analysis

In this part, we show the simulation results of OMTP and OGTP based on the generated data. Besides, we test the influence of different parameters and analyze the possible reasons.

Before the simulation, the length of time period T is set as 4 hours. The length of each time slot τ_l is set as 30s. The increasing number o is set as 50 and the cost threshold D is set as 1000.

In each simulation, the users' trip data is generated randomly, including start location, terminal location and appear time, where the appear time is subject to normal distribution. And the number of available bikes and empty docks at each bike station is equally initialized.

In the experiment, we mainly test the impact of two different parameters. The first parameter is the user number $|U|$, and the second parameter is the maximal walking distance d_{max}. We consider 18 different user numbers and 5 different maximal walking distances. The specific value of parameters is presented in Table II.

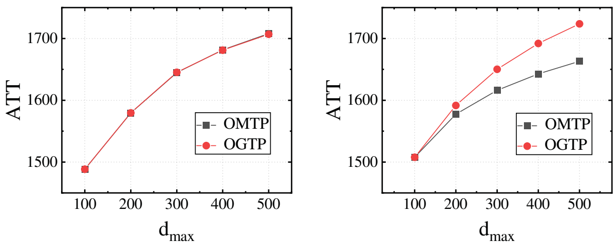
TABLE II
EXPERIMENT SETTINGS

Factor	Setting
Station number	936
Walking speed	$5km/h$
Riding speed	$20km/h$
User number	$\{[1k,12k]\|\Delta = 1k\} \cup \{[15k,40k]\|\Delta = 5k\}$
d_{max}	$100m$, $200m$, $300m$, $400m$, $500m$

Each simulation with different user number and maximal walking distance repeats 10 times and records the average result.

Average trip time. Fig. 5 illustrates the ATT of OMTP and OGTP. The bigger d_{max} results in higher ATT. The reason is that users need to walk longer to borrow or return bikes as d_{max} increases.

In Fig. 5(a), we show the ATT of OMTP and OGTP when $|U| = 10k$. Obviously, the ATT of OMTP and OGTP is almost the same. It is because that the conflicts among users are virtually nonexistent when $|U|$ is small. Both OMTP and OGTP allocate the optimal trip for each user. However, the difference between OMTP and OGTP increases with the increment of user number. When $|U| = 40k$, the ATT of OGTP is bigger than that of OMTP, shown in Fig. 5(b). As $|U|$ gets bigger, the conflicts among users become serious. OGTP sacrifices some users' trip time to avoid conflicts, while OMTP always tries to allocate the maximum-quality bike-dock pair to each user. Therefore, in the scope of ATT, OMTP outperforms OGTP when d_{max} is big enough.

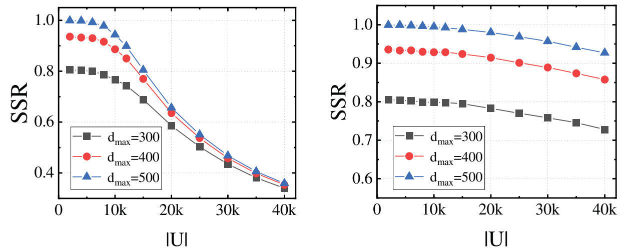


(a) $|U|$ = 10k (b) $|U|$ = 40k

Fig. 5. The ATT of OMTP and OGTP

Successful Service Ratio. Fig. 6 shows the SSR of OMTP and OGTP. The SSR changes along under different d_{max}. Under the same $|U|$, the SSR increases along with d_{max}, which is because users have more potential trips as d_{max} increases.

In Fig. 6(a), we show the SSR of OMTP under different d_{max}. When $|U|$ becomes bigger, the SSR of OMTP reduces rapidly. The reason is that the number of initial bike-dock pairs in PBS is finite, which is not sufficient to support all users' demand as $|U|$ increases. In Fig. 6(b), the SSR of OGTP decreases slightly with the increment of user number, which is because the bikes and docks will be reserved for users in OGTP. When $|U|$ exceeds a certain threshold, the SSR decreases.



(a) OMTP Algorithm (b) OGTP Algorithm

Fig. 6. The SSR of OMTP and OGTP

C. Real Data Based Experiment and Analysis

In this part, we give the experimental result based on the real-world human mobility data set in New York City and analyze the possible reasons.

We get the bike station data and the trip data on the Internet. The trip data utilizes the data from 2020-02-01 to 2020-02-29, which contains 29 days of data. In this experiment, the time period T chooses the peak period for bike borrowing: from 15:30 to 19:30. And we analyze the trip data and the bike station data in Fig. 7 and Fig. 8, respectively.

Fig. 7 shows the average number of real trip data from 15:30 to 19:30. Obviously, the number of trips in working days is higher than on weekends, which indicates that users tend to use bikes as a commuting tool. By calculation, we get that the average number of trips in this period is 9865. In reality, the actual demand is larger than that in record because the existing data set doesn't contain those who fail to borrow bikes.

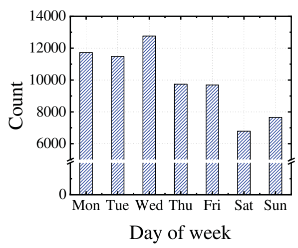


Fig. 7. The average number of users

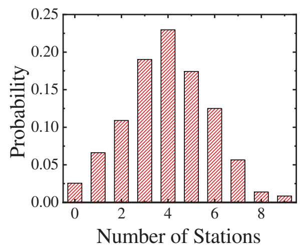


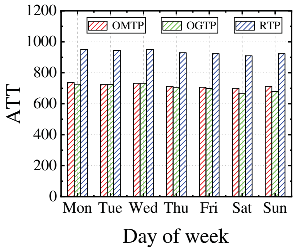
Fig. 8. Station distribution

Fig. 8 shows the probability distribution function of the number of adjacent stations when the range of each station is $500m$. From the picture, we know that over three-fourths of the stations have more than 2 adjacent stations. Therefore, the maximal walking distance d_{max} is also set as $500m$, which can ensure that there exists stations around the start location and target location of each user.

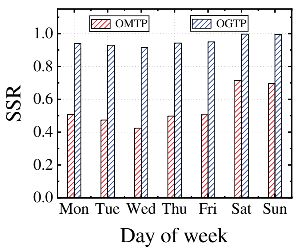
In this experiment, the length of each time slot and the increasing number are set as same as that in the generated based experiment. Since each trip record only contains the information from the start station to the target station, there is no user's start location and target location. So we randomly set two locations around the start station and target station as the start location and the target location.

We propose a benchmark algorithm, called Real Trip Planning algorithm (RTP). In RTP, we allocate the recorded pair of stations to borrow and return for each user. Besides, the trip time of each user is recalculated by Eq. (1). In the experiment, we plan trips for the users with three algorithms and record the daily results.

Fig. 9(a) shows the ATT of the three algorithms. From the picture, we can see that the fluctuation of ATT of three algorithms is relatively small. The average ATT of OMTP, OGTP and RTP is 11.96 minutes, 11.72 minutes and 15.55 minutes, respectively. The ATT of OMTP and OGTP is pretty close, while the ATT of OMTP and OGTP is nearly 23.1% lower than that of RTP. Fig. 9(b) shows the SSR of OMTP and OGTP. The RTP algorithm plans the real trip for each user so the SSR of RTP is 100%. The average SSR of OGTP is 95.28%, which is 4.72% lower than that of RTP and is 40.71% higher than that of OMTP.



(a) Average Trip Time



(b) Successful Service Ratio

Fig. 9. The Performance of OMTP and OGTP

VI. Discussion

Algorithm application. This paper designs the OMTP and OGTP algorithm to address the OTP problem. To ensure the success of each trip, we give the deterministic bike-dock pair for each user. In the real scenario, the available bikes and the empty docks at bike stations are shared and unattended, and each user has different characteristics and preferences. These features pose challenges to apply online trip planning algorithm in reality.

Algorithm analysis. In the Generated Data Based experiment, we know that the conflicts have a great impact on the above two algorithms. OMTP only considers one user's interest, while OGTP considers a group of users' interest which eases the short-term conflicts. However, both two algorithms don't consider the long-term returns in PBS. Thus, better algorithms should be designed for PBS.

VII. Related Work

In this paper, we introduce two algorithms to address the OTP problem. Next, we review some related work in three categories: Public Bike System, Online Matching, and Network Flow.

A. Public Bike System

The public bike system was first launched at 1965 in Amsterdam [16]. After two times of evolution, now the third generation city bike system has been launched in 712 cities in 2014 all over the world [17].

In recent years, the explosive growth of bicycles also brings some potential problems to the city. To solve it, a mass of researchers have paid attention to PBS problem, such as the redistribution problem [5] [6], system prediction problem [2] [3] [4] and station location determination [18] [19]. Moreover, the effective trip planning of bike-sharing schemes for users has also been proposed. Li *et al.* focus on the static trip planning problem and illustrate the complexity of this problem [11]. Zhang *et al.* model the static trip planning problem as a bike trip selection game and propose a distributed algorithm [10].

As far as we know, there is few work focusing on the online trip planning problem. Yoon *et al.* put forward a prediction model based on ARIMA and design a trip advisor system [20]. Hu *et al.* present a fine-grained probabilistic forecast method and design a novel architecture which helps to the balance bike usage [9]. These existing works are probability-based. However, the probabilistic predict model may not very accurate and timely for each user, and it will impose a heavy burden on the server.

B. Online Matching

Nearly three decades age, Karp and Vazirani *et al.* introduce the online matching problem [12]. For decades, there has been a mass of literature on the online matching problem and its variants [21] [22] [23].

In recent years, the problem of spatial crowdsourcing is becoming popular. The problem is how to allocate tasks to suitable crowd workers. To more fit the actual situation, online matching has been recently widely used in some spatial crowdsourcing problem [24] [25] [26]. In the domain of transportation, online matching is widely used in the taxi dispatching problem [27] [28].

In this problem, we should consider the unique characteristics of the public bicycle and dynamics of resources at the bike stations, which is different from the above problem.

C. Network Flow

The Network Flow Problem is an old problem, which has been introduced in 1956 by Ford *et al.* [29]. The network flow problem has been extensively studied in a variety of domains including wireless sensor, route planning *etc.* [30] [31].

Zhang *et al.* consider the complete bike route may contain the exchanging station and solve the bike route planning problem with the minimum cost flow algorithm. [32]. However, his works did not take the conflicts among users into consideration.

Different from these works presented above, this paper studies the online trip planning problem, which meets the actual demand.

VIII. Conclusion

This paper mainly focuses on the online trip planning problem (OTP) for PBS where users arrive online. To address it, we design two algorithms, called OMTP and OGTP. And we conduct two different experiments- Generated Data Based and Real Data Based. In the Generated Data Based experiment, we reveal the impact of different parameters with the generated trip data. And the simulation result shows that OGTP helps more people find trips than OMTP, while OMTP can better save users' trip time. In the real data based experiment, we validate our proposed algorithms with the real trip set in New York City. The results show that both OMTP and OGTP save 23.1% time per trip on average. And OGTP helps nearly 95.28% of users successfully receive the service of PBS.

IX. Acknowledge

This work is under the support of National Natural Science Foundation of China (NSFC) under Grant No.61473109, 61672198, 61572164 and 61877015, Zhejiang Major Science and Technology Programme (2018C04012), the General Research Foundation of the Education Department of Zhejiang Province under Grant No.Y201840731, and the Graduate Scientific Research Foundation of Hangzhou Dianzi University under Grant No.CXJJ2018053.

References

[1] Andreas Kaltenbrunner, Rodrigo Meza, Jens Grivolla, Joan Codina, and Rafael Banchs. Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. *Pervasive and Mobile Computing*, 6(4):455–466, 2010.

[2] Zidong Yang, Ji Hu, Yuanchao Shu, Peng Cheng, Jiming Chen, and Thomas Moscibroda. Mobility modeling and prediction in bike-sharing systems. In *Proc. of ACM MobiSys*, pages 165–178, June 26-30 2016.

[3] Divya Singhvi, Somya Singhvi, Peter I Frazier, Shane G Henderson, Eoin O'Mahony, David B Shmoys, and Dawn B Woodard. Predicting bike usage for new york city's bike sharing system. In *AAAI 2015 Workshop on Computational Sustainability*, Austin, Texas, USA, January 25-26 2015.

[4] Romain Giot and Raphaël Cherrier. Predicting bikeshare system usage up to one day ahead. In *IEEE CIVTS*, pages 22–29, San Diego, CA, USA, October 5-8 2014.

[5] Yexin Li, Yu Zheng, and Qiang Yang. Dynamic bike reposition: A spatio-temporal reinforcement learning approach. In *ACM KDD*, pages 1724–1733, London, UK, August 19-23 2018.

[6] Adish Singla, Marco Santoni, Gábor Bartók, Pratik Mukerji, Moritz Meenen, and Andreas Krause. Incentivizing users for balancing bike sharing systems. In *AAAI*, pages 723–729, Austin, Texas, USA, January 25-29 2015.

[7] Tianfu He, Jie Bao, Ruiyuan Li, Sijie Ruan, Yanhua Li, Chao Tian, and Yu Zheng. Detecting vehicle illegal parking events using sharing bikes' trajectories. In *ACM KDD*, pages 340–349, London, UK, August 19-23 2018.

[8] Bao Jie, Tianfu He, Sijie Ruan, Yanhua Li, and Zheng Yu. Planning bike lanes based on sharing-bikes' trajectories. In *ACM KDD*, pages 1377–ÍC1386, Halifax, Nova Scotia, Canada, Aug 13-17 2017.

[9] Ji Hu, Zidong Yang, Yuanchao Shu, Peng Cheng, and Jiming Chen. Data-driven utilization-aware trip advisor for bike-sharing systems. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 167–176. IEEE, 2017.

[10] Jianhui Zhang, Pengqian Lu, Zhi Li, and Jiayu Gan. Distributed trip selection game for public bike system with crowdsourcing. In *IEEE INFOCOM*, pages 2717–2725, Honolulu, HI, USA, April 15-19 2018.

[11] Zhi Li, Jianhui Zhang, Jiayu Gan, Pengqian Lu, Zhigang Gao, and Wanzeng Kong. Large-scale trip planning for bike-sharing systems. *Pervasive and Mobile Computing*, 54:16–28, 2019.

[12] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358. ACM, 1990.

[13] Cynthia Barnhart, Christopher A Hane, and Pamela H Vance. Integer multicommodity flow problems. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 58–71. Springer, 1996.

[14] Raymond C Browning, Emily A Baker, Jessica A Herron, and Rodger Kram. Effects of obesity and sex on the energetic cost and preferred speed of walking. *Journal of applied physiology*, 100(2):390–398, 2006.

[15] Bicycle performance. http://en.wikipedia.org/wiki/Bicycle. Accessed Jan 1, 2020.

[16] Paul DeMaio. Bike-sharing: History, impacts, models of provision, and future. *Journal of public transportation*, 12(4):3, 2009.

[17] Susan A Shaheen, Elliot W Martin, Adam P Cohen, Nelson D Chan, and Mike Pogodzinski. Public bikesharing in north america during a period of rapid expansion: Understanding business models, industry trends & user impacts, mti report 12-29. 2014.

[18] Juan Carlos García-Palomares, Javier Gutiérrez, and Marta Latorre. Optimizing the location of stations in bike-sharing programs: A gis approach. *Applied Geography*, 35(1-2):235–246, 2012.

[19] Junming Liu, Qiao Li, Meng Qu, Weiwei Chen, Jingyuan Yang, Hui Xiong, Hao Zhong, and Yanjie Fu. Station site optimization in bike sharing systems. In *2015 IEEE International Conference on Data Mining*, pages 883–888. IEEE, 2015.

[20] Ji Won Yoon, Fabio Pinelli, and Francesco Calabrese. Cityride: a predictive bike sharing journey advisor. In *IEEE MDM*, pages 306–311, Bengaluru, India, Jul 23-26 2012. IEEE.

[21] Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science*, 8(4):265–368, 2013.

[22] Itai Ashlagi, Maximilien Burq, Chinmoy Dutta, Patrick Jaillet, Amin Saberi, and Chris Sholley. Maximum weight online matching with deadlines. *arXiv preprint arXiv:1808.03526*, 2018.

[23] Buddhima Gamlath, Michael Kapralov, Andreas Maggiori, Ola Svensson, and David Wajc. Online matching with general arrivals. *arXiv preprint arXiv:1904.08255*, 2019.

[24] Yajun Wang and Sam Chiu-wai Wong. Two-sided online bipartite matching and vertex cover: Beating the greedy algorithm. In *International Colloquium on Automata, Languages, and Programming*, pages 1070–1081. Springer, 2015.

[25] Yongxin Tong, Jieying She, Bolin Ding, Libin Wang, and Lei Chen. Online mobile micro-task allocation in spatial crowdsourcing. In *ICDE*, pages 49–60. IEEE, 2016.

[26] Yansheng Wang, Yongxin Tong, Cheng Long, Pan Xu, Ke Xu, and Weifeng Lv. Adaptive dynamic bipartite graph matching: A reinforcement learning approach. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1478–1489. IEEE, 2019.

[27] John P Dickerson, Karthik A Sankararaman, Aravind Srinivasan, and Pan Xu. Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. In *AAAI*, pages 1007–1014, New Orleans, USA, Feb 2-7 2018.

[28] Boming Zhao, Pan Xu, Yexuan Shi, Yongxin Tong, Zimu Zhou, and Yuxiang Zeng. Preference-aware task assignment in on-demand taxi dispatching: An online stable matching approach. In *AAAI*, Hawaii, USA, 2019.

[29] Lester R Ford Jr. Network flow theory. Technical report, Rand Corp Santa Monica Ca, 1956.

[30] José Maurício Costa, Pedro P Paniago, Joaquim de Andrade, Thiago F Noronha, and Marcos AM Vieira. Integer linear programming formulations for the variable data rate and variable channel bandwidth scheduling problem in wireless networks. *Computer Networks*, 165:106939, 2019.

[31] Christian Vad Karsten, David Pisinger, Stefan Ropke, and Berit Dangaard Brouer. The time constrained multi-commodity network flow problem and its application to liner shipping network design. *Transportation Research Part E: Logistics and Transportation Review*, 76:122–138, 2015.

[32] Jiawei Zhang and Philip S. Yu. Trip route planning for bicycle-sharing systems. In *IEEE CIC*, pages 381–390, Pittsburgh, PA, USA, Nov 1-3 2016.