Rebalancing Bike Sharing Systems: A Multi-source Data Smart Optimization

Junming Liu[1], Leilei Sun[2], Weiwei Chen[3], Hui Xiong[1]*
[1]Management Science and Information Systems, Rutgers University, USA, {jl1433, hxiong}@rutgers.edu
[2]Institute of Systems Engineering, Dalian University of Technology, China, leisun@mail.dlut.edu.cn
[3]Supply Chain Management, Rutgers University, USA, wchen@business.rutgers.edu

ABSTRACT

Bike sharing systems, aiming at providing the missing links in public transportation systems, are becoming popular in urban cities. A key to success for a bike sharing systems is the effectiveness of rebalancing operations, that is, the efforts of restoring the number of bikes in each station to its target value by routing vehicles through pick-up and drop-off operations. There are two major issues for this bike rebalancing problem: the determination of station inventory target level and the large scale multiple capacitated vehicle routing optimization with outlier stations. The key challenges include demand prediction accuracy for inventory target level determination, and an effective optimizer for vehicle routing with hundreds of stations. To this end, in this paper, we develop a Meteorology Similarity Weighted K-Nearest-Neighbor (M-SWK) regressor to predict the station pick-up demand based on large-scale historic trip records. Based on further analysis on the station network constructed by station-station connections and the trip duration, we propose an inter station bike transition (ISBT) model to predict the station drop-off demand. Then, we provide a mixed integer nonlinear programming (MINLP) formulation of multiple capacitated bike routing problem with the objective of minimizing total travel distance. To solve it, we propose an Adaptive Capacity Constrained K-centers Clustering (AdaCCKC) algorithm to separate outlier stations (the demands of these stations are very large and make the optimization infeasible) and group the rest stations into clusters within which one vehicle is scheduled to redistribute bikes between stations. In this way, the large scale multiple vehicle routing problem is reduced to inner cluster one vehicle routing problem with guaranteed feasible solutions. Finally, the extensive experimental results on the NYC Citi Bike system show the advantages of our approach for bike demand prediction and large-scale bike rebalancing optimization.

*Contact Author.

KDD '16, August 13-17, 2016, San Francisco, CA, USA
 ISBN 978-1-4503-4232-2/16/08…$15.00
DOI: http://dx.doi.org/10.1145/2939672.2939776

CCS Concepts

•**Applied computing** → **Transportation;** *Forecasting;*
•**Information systems** → Clustering;

Keywords

Bike Sharing System; Optimization; Clustering

1. INTRODUCTION

Recent years have witnessed worldwide prevalence of public bike sharing systems [5, 21], which provide short-term bike rental services with many bike stations scattering over an urban city. Indeed, these bike sharing systems offer an environment-friendly solution for the first-and-last mile connection and help bridge the gap between existing transportation modes such as subways and bus systems.

However, despite the significant benefits from bike sharing systems, it is very challenging to operate these systems in an effective and efficient way. The dynamics of human mobility often lead to inevitable bike supply/demand imbalance. Thus, it is crucial for service providers to redistribute bikes among stations in a proactive and economical way in order to ensure the system functioning effectively. To this end, in this paper, we study the bike rebalancing problem, where the assumption is that the number of bikes at each station is known in advance and will not be changed during the rebalancing operation (when system is closed or during midnight). There are two major challenges for this problem. First, it is necessary to determine the desired inventory level at each station. However, this depends on the accurate prediction of station-level pick-up and drop-off demands, which can be impacted by multiple factors, such as time, location, weather conditions, and traffic situations. Second, the bike rebalancing problem is a large-scale multiple capacitated vehicle routing problem, which is intractable due to the existence of the outlier stations with extremely large target inventory levels. Also, the scales of bike sharing systems, consisting of hundreds of stations, are too large to be solved by traditional route optimization techniques.

A number of recent researches have studied the bike rebalancing problem. Most studies on bike demand prediction are based on historical demand mean [10] or stochastic process [20, 1] with historical pick-up (drop-off) rate, without considering the impact of other influential factors. A promising way to improve bike demand prediction accuracy is to leverage a variety of data that is directly or indirectly related to the public bike sharing service. For instance, Yexin, etc (2015) [14] propose a multi-factor hierarchical predic-

tion model to predict the bike demands of the clusters of stations in a future period. However, it requires an accurate station-level bike usage prediction to determine the desired inventory level. For the problem of station rebalancing optimization problem, the small-scale (up to 60 stations) bike rebalancing problem has been investigated by solving feasible optimization models with the assumption that there exists at least one route for covering all targeted stations [4, 6, 3]. In practice, especially in large scale bike sharing systems, it is expected to determine the optimal number of the relocated bikes and to deal with large number of stations with some outlier stations which make traditional routing optimization infeasible.

Indeed, the emergence of multi-source big data enables a new paradigm for enhancing bike sharing services. Along this line, we exploit multi-source data related to bike sharing services, such as trip records, station status records, and weather reports, for developing data smart bike rebalancing solutions. Specifically, we first propose a Meteorology Similarity Weighed K-Nearest-Neighbor (MSWK) regression model to predict the hourly bike pick-up demand at each station. The weight of different factors are carefully learned to improve the accuracy of bike pick-up demand prediction. Also, for predicting the drop-off demand at each station, we provide an inter station bike transition (ISBT) predictor which considers the surrounding station pick-up information and the connection information between stations (station-station trip frequencies and trip durations). The predicted pick-up and drop-off demands can then be used to determine the station inventory targets.

Moreover, to solve the large scale multi-capacitated bike routing optimization problem for rebalancing, a hierarchical optimization model is proposed by exploring multi-source data. The optimization model first performs Adaptive Capacity Constrained K-centers Clustering (AdaCCKC) which separates outlier stations and groups the rest stations into different clusters, within which the stations are rebalanced by one vehicle with optimal route. Here, a mixed integer nonlinear programming (MINLP) model with lazy constrains is conducted to optimize the inner cluster rebalancing routes.

Finally, we carry out extensive experiments on a real world data set including trip records of 698 days collected from the NYC Citi Bike system and the experimental results clearly show the effectiveness of the proposed methods.

2. PROBLEM FORMULATION

In this section, we first define some preliminaries. Then, we introduce the rebalancing problem of multiple capacitated vehicles route optimization in bike sharing systems.

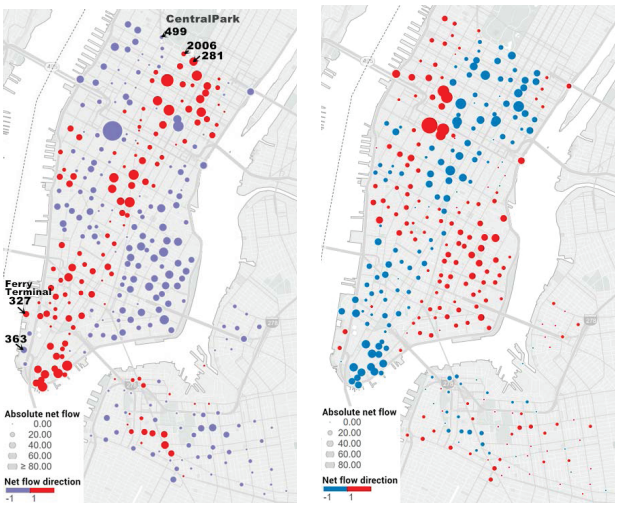
2.1 Preliminaries

Here, we first introduce some preliminaries, which will be used throughout this paper.

2.1.1 Station network.

A bike station network is represented by a directed graph $G = (S, E)$. With each station $s \in S$ as a node, the edges in E are directed connections of bike stations $e_{ij} = (s_i, s_j) \in E$. Each node and edge have several attributes. For example, $e_{ij}.f$ represents the frequency of bike trip records from station i to station j.

The station network is constructed by tracking a set of trip records. A trip record $tr = (s_0, s_d, \tau_0, \tau_0)$ is a bike usage record from an origin station s_0 to a destination station s_d. τ_0 is the pick-up time and and τ_d is the drop-off time. Only trips with duration $\tau_d - \tau_0$ larger than 1 min are recorded.



(a) Net flow 9 am ∼ 10 am (b) Net flow 6pm ∼ 7 pm

Figure 1: A net flow distribution of the NYC Citi Bike System during rush hours.

2.1.2 Station pick-up/drop-off demand.

The station bike demand is defined as the pick-up (drop-off) frequency per unit time when the station is available. Station availability means the station is in service and there are bikes available for pick-up (drop-off). Station unavailability is usually due to maintenance, empty dock (for pick-up) and full dock (for drop-off). We do not consider the station demand during its unavailable period.

Definition 1: *Station pick-up (drop-off) demand.* Let $s_i.pf(t)$ ($s_i.df(t)$) and $s_i.pa(t)$ ($s_i.da(t)$) represent the pick-up (drop-off) frequency and the pick-up (drop-off) available time of station i during time slot t. Each day is separated into 24 time slots of one hour duration, $t \in \{0, 1, ..., 23\}$. The station pick-up (drop-off) demand $s_i.pd(t)$ ($s_i.dd(t)$) and the bike net flow $s_i.nf(t)$ is defined as follows:

$$s_i.pd(t) = \frac{s_i.pf(t)}{s_i.pa(t)} \quad (s_i.dd(t) = \frac{s_i.df(t)}{s_i.da(t)}) \\ s_i.nf(t) = s_i.dd(t) - s_i.pd(t) \tag{1}$$

The one month (July. 2014) averaged station net flow distribution during weekday rush hours (9 am ∼ 10 am and 6 pm ∼ 7 pm) of the NYC CitiBike sharing system is presented in Figure 1 as an example. In Figure 1, each dot represents a bike station in NYC with its size representing the absolute value of net flow. The red color represents a positive net flow (drop-off frequency is larger than pick-up frequency) and the blue color represents a negative net flow. As we can see from Figure 1, the station demand distribution is unbalanced both geographically and temporally.

2.1.3 Station rebalancing target

Definition 2: *Station Rebalancing Target.* Given the station capacity $s_i.c$, the current number of bikes $s_i.cn$ and the station bike net flow $s_i.nf(t)$, the station rebalancing target $s_i.rt$ is defined as the number of bikes that need to be picked-up or delivered by rebalancing vehicle. The op-

timal inventory level aims at maximizing the time duration T within which the station remains balanced during its in-service time period. Thus the station rebalancing target $s_i.rt$ is decided by the following calculation:

$$\arg\max_{s_i.rt}\{T||s_i.cn + s_i.rt + \sum_{t=0}^{T} s_i.nf| \leq s_i.c\} \quad (2)$$

2.2 Problem Definition

2.2.1 Station level bike demand prediction

Given a set of bike trip records $TR_H = \{tr_1, tr_2, ...tr_H\}$ and weather report R_t, the problem of station level bike pick-up (drop-off) prediction is to predict the hourly pick-up (drop-off) demand $s_i.pd(t)$ ($s_i.dd(t)$) defined by Definition 1 of each station during a day.

2.2.2 Static bike rebalancing optimization

Given a set of bike station locations s_i of size m, rebalancing target $s_i.rt$, their inter vehicle transportation distance TD_{ij} and the number of operating vehicles $|V|$ with specified capacity limit VC, the problem of bike rebalancing optimization is to decide the optimal vehicle rebalancing routes, which is mathematically formulated as follows:

$$\min \quad \mathcal{F}(\mathbf{x}) = \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{v\in V} x_{ijv} TD_{ij} \quad (3)$$

$$s.t. \quad y_i + \sum_{j\neq i}\sum_{v\in V} x_{ijv} s_j.rt = \sum_{j\neq i}\sum_{v\in V} x_{ijv} y_j \quad \forall i \quad (4)$$

$$0 \leq y_i \leq VC \qquad \forall i \quad (5)$$

$$x_{ijv} \in \{0,1\} \qquad \forall i \neq j, \forall v \in V \quad (6)$$

$$\sum_{j\neq i, v\in V} x_{ijv} = 1 \quad \forall i \quad (7)$$

$$\sum_{i\neq j, v\in V} x_{ijv} = 1 \quad \forall j \quad (8)$$

Binary decision variable $x_{ijv} = 1$ indicates vehicle $v \in V$ redistributes bikes from station i to station j, otherwise $x_{ijv} = 0$. Integer decision variable y_i indicates the number of bikes carried by a vehicle after visiting station i. The objective function (3) minimizes the total traveling distance of vehicles for re-balancing. Quadratic constrain (4) implies bike flow conservation. Constrain (5) is for the vehicle capacity limit and (7, 8) indicate that a vehicle must enter and leave a station only once (no self loop). This MINLP formulation for vehicle route optimization is not completed because there exists outlier stations making this model infeasible and subtours have not been eliminated. We could add subtour elimination constrains (SEC) directly into our MINLP formulation. However, the number of SEC increases exponentially with the number of stations. Thus, for a large scale bike sharing system, considering all SEC is unsolvable.

2.3 An overview of the framework

Figure 2 shows the framework of our proposed method, which consists of two major phases: the station level bike demand (pick-up and drop-off) prediction and the AdaCCKC based bike rebalancing optimization.

Station level bike demand prediction. We first extract bike history records and station inventory status data from bike sharing systems, from which the station level bike demands defined by Definition 1 are calculated as benchmark data. The weather reports of each time slot are extracted, and the unbalanced weather condition records and missing records are preprocessed. In this step, weather conditions are clustered into four categories according to their suitability for outdoor bicycling: U_1={heavy snow, heavy rain}, U_2={snow, rain, light snow, light rain}, U_3={fog, mist, haze}, U_4={Cloudy, Sunny}. A meteorology similarity function is learned to build the meteorology similarity weighted KNN (MSWK) regression for station bike pick-up demand prediction. The station bike drop-off demand is predicted by the Inter Station Bike Transition (ISBT) predictor which estimates the drop-off station and drop-off time after a pick-up event according to the station-station connected network. Based on the predicted station bike net flow defined by Definition 1 and current inventory level, the station inventory target is then determined by Equation 2.

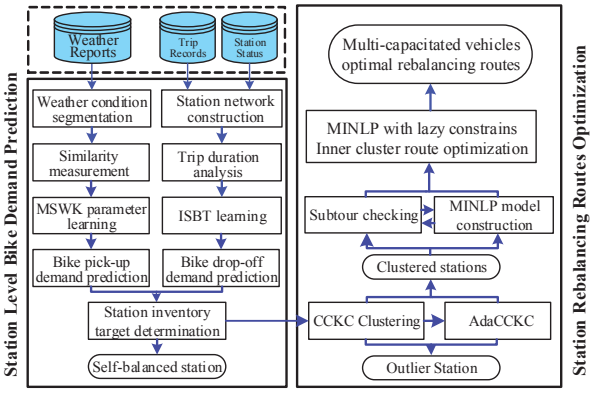


Figure 2: An overview of the framework.

Bike station rebalancing optimization. According to the station level target, several self-balanced stations (stations with net-flow close to zero) are excluded. The remaining stations are clustered by (Adaptive) Capacity Constrained K-centers Clustering (CCKC or AdaCCKC) and some outlier stations are discovered. We then implement one-vehicle routing MINLP model with lazy constrains to optimize the vehicle rebalancing routes for the rest stations within each cluster.

3. THE REBALANCING APPROACH

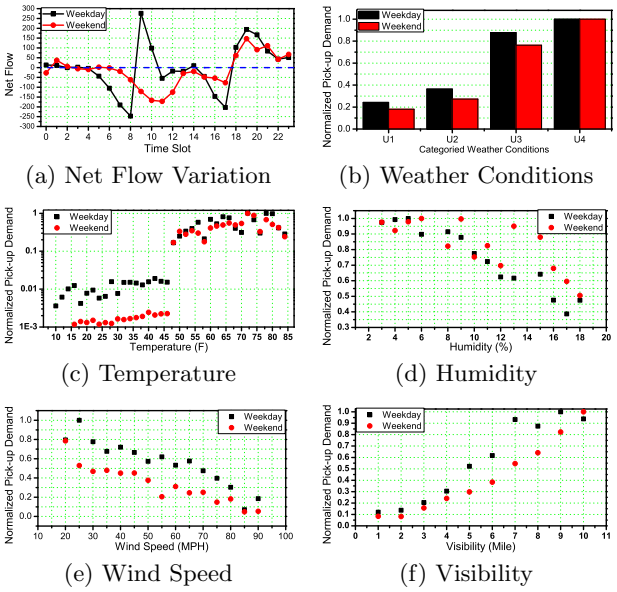
In this section, we provide the technical details of our bike demand prediction model and AdaCCKC based bike rebalancing optimization model. Weekdays and weekends are studied separately since the bike demand and the net flows are very different (see the example of CitiBike 24-hour net flow as an example in Figure 3(a)).

3.1 Station Bike Pick-up Prediction

The MSWK regressor is built to predict the station level bike pick-up demand $s_i.pd(D^t)$ during time slot t of any given day D that is on the basis of a meteorology multi-similarity function.

3.1.1 Similarity measurement

Given the weather reports R_D^t of each time slot t, which contains weather condition $W_{D_p^t}$ (sunny, raining, etc.), temperature $F_{D_p^t}$, humidity $H_{D_p^t}$, wind speed $S_{D_p^t}$ and visibility



(a) Net Flow Variation (b) Weather Conditions

(c) Temperature (d) Humidity

(e) Wind Speed (f) Visibility

Figure 3: Demand net flow (a) and the effect of multiple factors on bike pick-up demand (b)-(f).

$V_{D_p^t}$ of time slot t on day D_p, the similarity between 2 different days D_p^t and D_q^t is calculated as the linear combination of three units: weather similarity, temperature similarity and humidity-wind speed-visibility similarity. Each unit is associated with an effective coefficient a that is learned to improve the prediction accuracy.

Weather similarity. The weather conditions are first manually segmented into different levels according to their suitability for outdoor bicycling (see Figure 3(b)): ((heavy snowy, heavy rainy), (snowy, rainy), (hazy, foggy), (clear, cloudy))=(1, 0.75, 0.5, 0.25). Then the weather similarity is defined as follows:

$$\lambda_1(W_{D_p^t}, W_{D_q^t}) = \frac{1}{2\pi\sigma_1} e^{-\frac{(W_{D_p^t} - W_{D_q^t})^2}{\sigma_1^2}} \tag{9}$$

Temperature similarity. As can be seen from Figure 3(c), the bike pick-up demand is sensitive to the temperature, especially when temperature is below 47 F during the weekends. We extract the temperature information and calculate its similarity based on a Gaussian Kernel function:

$$\lambda_2(F_{D_p}^t, F_{D_q}^t) = \frac{1}{2\pi\sigma_2} e^{-\frac{(F_{D_p^t} - F_{D_q^t})^2}{\sigma_2^2}} \tag{10}$$

Humidity, wind speed and visibility similarity. Different from the effect of temperature, the humidity, wind speed and visibility affect bike pick-up demand with similar effects (see Figure 3(d)-3(f)). We choose a 3-D Gaussian Kernel to calculate the similarity between $(H_{D_p^t}, S_{D_p^t}, V_{D_p^t})$ and $(H_{D_q^t}, S_{D_q^t}, V_{D_q^t})$:

$$\lambda_3 = \frac{1}{2\pi\sigma} e^{-(\frac{(H_{D_p^t} - H_{D_q^t})^2}{\sigma_3^2} + \frac{(S_{D_p^t} - S_{D_q^t})^2}{\sigma_4^2} + \frac{(V_{D_p^t} - V_{D_q^t})^2}{\sigma_5^2})} \tag{11}$$

Similarity function. To uniform these similarity calculations, we normalize the temperature, humidity, wind speed and visibility into range $[0, 1]$ and simplify equation (9)-(11) by setting $\sigma_k = 1$ $(k = 1, 2, 3, 4, 5)$. The similarity function is then defined as a linear combination of λ:

$$M(D_p^t, D_q^t; a) = \delta_w(D_p, D_q) \sum_{i=1}^{3} a_i \lambda_i \tag{12}$$

where $\delta_w(D_p, D_q)$ is the delta function. $\delta_w(D_p, D_q) = 1$ if D_p and D_q are both weekdays or weekends, otherwise $\delta_w(D_p, D_q) = 0$.

3.1.2 MSWK learning

Given K and a, we select top K days $\{D_1^t, D_2^t, ..., D_K^t\}$ with the highest similarity to our target day D_n^t according to the similarity function. Then the $s_i.pd(D_q^t)$ is predicted by a similarity weighed KNN:

$$s_i.pd(D_q^t; a) = \frac{\sum_{p=1}^{K} M(D_p^t, D_q^t; a) s_i.pd(D_p^t)}{\sum_{p=1}^{K} M(D_p^t, D_q^t; a)} \tag{13}$$

The weight of different similarity function a in equation (12) is trained to reach the minimum prediction absolute error of predicted value $\hat{s}_i.pd(D_q^t; a)$ and ground truth $s_i.pd(D_q^t)$ by brute force searching:

$$a* = \arg\min_a \frac{1}{N} \sum_{i=1}^{N} |\hat{s}_i.pd(D_q^t; a) - s_i.pd(D_q^t)| \tag{14}$$

3.2 Station bike drop-off prediction

The inter station bike transition (ISBT) predictor for bike drop-off demand prediction investigates that for a pick-up event at station i during time slot t, the possibility that it will be dropped-off at station j during the same time slot t and next time slot $t + 1$.

3.2.1 Station-station inter transportation analysis

Given predicted station bike pick-up demand during time slot t ($s_i.pd(t)$), the number of bikes that will be dropped-off at station s_j from s_i is estimated from trip history records:

$$e_{ij}^t = s_i.pd(t) \frac{e_{ij}.f}{s_i.pd} \tag{15}$$

where $s_i.pd$ is the total pick-up demand at station s_i and $e_{ij}.f$ is the total number of trips from station s_i to s_j.

Among all the trips from station s_i to s_j which start during time slot t, not all of them will be dropped-off during the same time slot due to the unestimated trip duration. For commuters, they will find the fastest path to reach station s_j. For tourists or bike exercisers, the trip duration will vary. Figure 4 presents 3 typical station-station connections (the station IDs are indexed by the NYC Citi Bike System). The trip duration distributions are fitted by a 2-Gaussian functions (see fitting results in Table 1). Among the 3 typical station-station trip connections, the commuters construct the first left gaussian peak with a mean trip duration close to the fastest route estimated by Google Maps (second column of Table 1). The second right gaussian peak reveals the tourists or bike exercisers. Most trips from station 2006 to 281 (the large dot marker with black fit line) are not for the fastest route commuting and about half trips from station 281 to 499 (the diamond marker with red fit line) are for commuting. The trips from station 327 to 363 (the small dot marker with blue line) are mostly commuting trips. It is interesting that the station 2006 and 281 are near Central Park where most bike users are tourists or exercisers. The

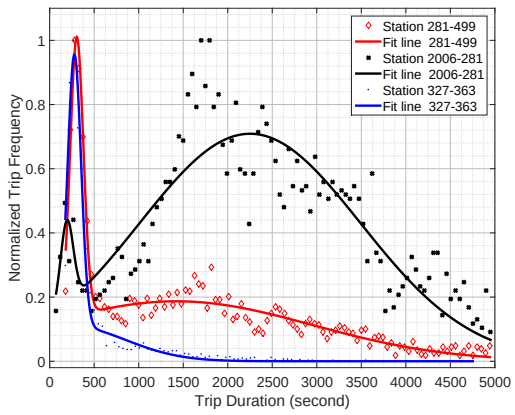


Figure 4: Trip duration distributions of three typical station-station connections.

Table 1: Fitting Results and P_{ij}^T, P_{ij}^{T+1} estimation function: $f(x) = a_1 exp^{-(\frac{x-x_{c1}}{b_1})^2} + a_2 exp^{-(\frac{x-x_{c2}}{b_2})^2}$

Edges	Map	x_{c1}(95% CI)	x_{c2}(95% CI)	R^2	P_{ij}^T
281-499	300	304.5±5	1424±237	0.9569	0.89
2006-281	120	194.3±55.5	2250±92	0.7532	0.73
327-363	180	212.9±30.5	278.1±6.1	0.9728	0.96

station 327 and 363 are located near World Financial Center Ferry Terminal where most bike users are commuters (see Figure 1 for station locations). Therefore, we estimate the trip duration (in unite of second) of each two pair of stations s_i and s_j by a 2-peak Gaussian function:

$$D_{ij}(t) = g_1(t; \mu_1, \sigma_1) + g_2(t; \mu_2, \sigma_2) \tag{16}$$

Let t_1 represent the actual pick-up time and t_2 represent trip duration from station s_i to s_j during time slot t of length $|t|$, the possibility that the picked-up bike will arrive at station j at the same time slot is calculated as:

$$\begin{aligned} P_{ij}^t &= P(t_1 + t_2 \leq |t|) \\ &= \int_0^{|t|} dt_1' P(t_2 \leq |t| - t_1 | t_1 = t_1') P(t_1 = t_1') \end{aligned} \tag{17}$$

Assuming the pick-ups arrive with uniform distribution, the same slot arrival possibility P_{ij}^T and next slot arrival possibility P_{ij}^{T+1} are calculated as following:

$$\begin{aligned} P_{ij}^t &= \frac{1}{|t|} \int_0^{|t|} \int_0^{|t|-t_1'} dt_1' dt_2 D_{ij}(t_2) \\ P_{ij}^{t+1} &= \frac{1}{|t|} \int_0^{|t|} \int_{|t|-t_1'}^{+\infty} dt_1' dt_2 D_{ij}(t_2) \end{aligned} \tag{18}$$

The slot arrival probability has the property of $P_{ij}^t + P_{ij}^{t+1} = 1$ since we neglect the possibility that a pick-up will be dropped off after 2 time slot periods (2 hours).

3.2.2 *Station level drop-off prediction*

The drop-off at station j is then estimated as:

$$s_j.dd(t) = \sum_{i \neq j} e_{ij}^t P_{ij}^t + e_{ij}^{t-1} P_{ij}^{t+1} \tag{19}$$

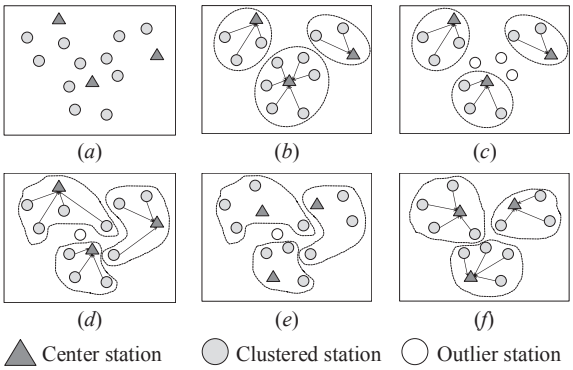


Figure 5: Constrained K-Centers Clustering procedure illustration.

The first term represents the estimated drop-offs during the same time slot as their pick-ups and the second term represents the estimated drop-offs one time slot later than their origin pick-ups from other stations.

3.3 Station rebalancing optimization

After we predict the station bike demand and determine their station inventory targets from Equation (2), the MINLP model for capacitated vehicle route optimization (Equation (3)-(8)) is still not solvable due to the following challenges: 1. the optimization problem is not feasible if the number of vehicles is not large enough to cover all stations; 2. the problem scale is too large to be solved efficiently; 3. direct solution of our MINLP model formulation contains subtours. In this section, we propose an AdaCCKC based hierarchial optimization model to solve the aforementioned challenges.

3.3.1 *Station clustering*

Clustering can be used to reduce the complexity of large-scale optimization problem. However, it has not arouse enough attention to reduce the scale of rebalancing optimization problem in bike sharing systems. Recently, Forma (2015) [9] proposed a heuristic algorithm and Schuijbroek (2013) [20] implemented the Maximum Spanning S-tar approximation for station clustering. However, in real large scale bike sharing systems, none of their methods are practical to deal with outlier stations. The implementation of clustering algorithms for bike rebalancing optimization problem has two challenges: 1) not only distances between stations should be taken into account, the station inventory target should also be considered for inventory constrains (Constrain (5)); 2) outlier stations should be discovered to guarantee inner cluster feasible routes. Although constrained clustering algorithms have been detailed studied, the constrained conditions in previous studies are mainly in the manner of *must-link* or *cannot link* pairs under the name of semi-supervised clustering [23, 2]. In this problem, whether two stations belong to the same cluster is not determined by themselves, but is affected by the total balance of stations in the same cluster. To the best of our knowledge, such kind of constrained clustering algorithm has not been studied ever before. In this paper, we propose a constrained K-Centers Clustering algorithm for bike stations to fill the research gap.

Algorithm 1 presents the proposed algorithm. It begins

Algorithm 1 $CCKC(\mathbf{TD},\mathbf{b},VN,VC,\delta,E)$

Input: TD_{ij}, b_i, VN, VC, δ, E;
Output: $\mathbf{c}$;

1: **for all** stations **do**
2: $c(i) = \arg\min_{j \in E}(TD_{ij})$;
3: **for each** cluster $C_k = \{i|c(i) = E(k)\}$, **if** $B(C_k) > VC$ **do**
4: **while** $B(C_k) > VC$ **do**
5: $q = \arg\min_{i \in C_k^*}(\sum_{j \in E} TD_{ij})$, where $C_k^* = \{i|B(C_k \setminus i) < B(C_k)\}$;
6: $c(q) = 0, C_k = C_k \setminus q$;
7: **find** $l = \arg\max_{i,c(i)=0}(\sum_{j \in E} TD_{ij})$
8: **if** $\exists\ S_l = \{k|B(C_k \bigcup l) < VC, \min_{j \in C_k} TD_{lj} < \delta\}$
9: **then do** $c(l) = E(k)$, $k = \arg\min_{q \in S_l}(\min_{j \in C_q} TD_{lj})$
10: **else do** $c(l) = -1$;
11: **go to** step 7 **till** $\nexists\ i$ that $c(i) = 0$;
12: **for each** cluster C_k
13: $E'(k) = \arg\min_{i,c(i)=E_k} \sum_{j,c(j)=E_k} TD_{ij}$;
14: **if** $E' \neq E$ **then** $E = E'$ **go to** step 1, **else go to** step 15;
15: **return** clustering result $\mathbf{c}$.

with an initial center set E, and assigns each station to its nearest stations. Then for a cluster, if the balance condition is not satisfied ($B(C_k) > VC$, where $B(C_k) = |\sum_{i \in C_k} b_i|$), we pick some stations out of the cluster. The stations, which are able to reduce the total balance of the cluster and close to other centers, are firstly picked out. In step 7∼11, the unlabeled stations are assigned with new cluster label. For each unlabeled station, the new cluster label is determined by the total balance of a cluster and the distance between the station and its nearest station in the cluster. The unlabeled outlier stations that are far from cluster centers are preferentially processed. This step ensures these outliers scattered at the central region of the studied area, and can be easily covered by other clusters. After adjusting clustering result according to balance conditions, new centers are selected in Step 12∼13. Step 1∼13 are iterated until convergence (centers are unchanged). Step 15 outputs the clustering result.

Figure 5 presents a toy example of the capacity constrained K-centers algorithm. Fourteen stations are given in Figure 5(a), and three stations are marked as initial centers by triangles. All other stations are assigned to their nearest centers in (b). Capacity condition is examined in (c). If the capacity of a cluster is over the vehicle capacity, then some stations are pressed out as temporary outliers. Stations near other centers are preferentially excluded as they are much easier to be visited by vehicles from other clusters. The outliers are assigned to other centers with respect to cluster capacity and traveling distance in (d). New centers are generated according to the current clustering result in (e) and stations are assigned to new centers in (f). This procedure is repeated till convergence.

The clustering result obtained by Algorithm 1 has the following features: 1) the total balance of a cluster is under the capacity of vehicle; 2) stations in a same cluster are close to each other; 3) clusters are overlapped, their boundaries are not discriminative; 4) outliers (if any) are at the central region of the studied area. All the features mentioned above are very helpful for designing bicycle station rebalancing routes. Feature 1) guarantees that there exists feasible solutions in the optimization of inner cluster, while Feature 2) means the traveling cost can be largely reduced as most of the routes are internal/short-term travels. Feature 3) shows vehicles can travel in an overlapping manner to serve as many stations as possible. Feature 4) means outliers are surrounded by a lot of vehicles. Therefore, the stations can be easily served by adding the capacity of vehicles nearby or by assigning two or more vehicles from other clusters to serve them. Although these features show advantages of Algorithm 1 in solving the bicycle station rebalancing problem, this algorithm still has some shortages as a K-Centers based method. On the one hand, the number of clusters needs to be specified. On the other hand, the clustering result is influenced by the initial center set. We further improve Algorithm 1 by proposing an Adaptive Capacity Constrained K-centers Clustering (AdaCCKC) to overcome these shortages.

Algorithm 2 $AdaCCKC(\mathbf{TD},\mathbf{b},VC,\delta,VNmax,NI)$

Input: TD_{ij}, b_i, VC, δ, $VNmax$, NI;
Output: $\mathbf{c}$;

1: $VNbest = VNmax$, $zbest = \sum_p \sum_q TD_{pq}$;
2: **for** i **from** 1 **to** $VNmax$ **do**
3: Generate initial center set E;
4: **for** j **from** i **to** $VNmax$ **do**
5: $\mathbf{c}$=$CCKC(\mathbf{TD}, \mathbf{b}, VN, VC, \delta, E)$;
6: **if** $\nexists\ h$ that $c(h) = 0$ **then break**
7: **else** $l = \arg\min_{p,c(p)=0} \sum_{q,c(q)=0} TD_{pq}$,
8: $E = unique(\mathbf{c})$, $E = E \bigcup l$;
9: **end**
10: $VN = |E|$, $z = \sum_{k=1}^{|E|} \sum_{p,q \in C(k)} TD_{pq}$;
11: **if** $(VN < VNbest)\&(z < zbest)$ **then**
12: $VNbest = VN$, $zbest = z$, $\mathbf{c}^* = \mathbf{c}$;
13: Repeat Step 2∼12 NI times;
14: **return** $\mathbf{c}^*$.

Algorithm 2 presents the proposed AdaCCKC algorithm. In each round, it begins with a randomly generated initial center set in Step 3. In step 4∼9, Algorithm 1 CCKC is implemented to get a temporary clustering result. If there exists unlabeled bicycle stations, a new cluster center is added to the current center set. The new added center is determined by all unlabeled stations as shown in Step 7. The break condition in Step 6 can also be activated if the number of outliers is below a specified threshold instead of 0, which makes the proposed algorithm more flexible for bicycle stations rebalancing problem. Considering the effect of initial center set on the final clustering result, the number of initial centers is set to vary from 1 to $VNmax$ in Step 2, where $VNmax$ is the maximum number of available vehicles. Step 12 picks out the best clustering result. Steps 2∼12 are repeated many times to reduce the influence of initial center set. As a result, Algorithm 2 can automatically determine the optimal number of vehicles in a smarter way, and users do not need to provide an initial center set.

3.3.2 *Inner cluster route optimization*

After our AdaCCKC clustering, the large scale multiple capacitated vehicles route optimization problem is simplified to one capacitated vehicle route optimization problem

of small or median problem sizes:

$$\min \quad \mathcal{F}_C(\mathbf{x}) = \sum_{i=1}^{m}\sum_{j=1}^{m} x_{ij} TD_{ij} \tag{20}$$

$$s.t. \quad y_i + \sum_{j\neq i} x_{ij} s_j.rt = \sum_{j\neq i} x_{ij} y_j \quad \forall i \tag{21}$$

$$y_i \leq VC \qquad \forall i \tag{22}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \neq j \tag{23}$$

$$\sum_{j\neq i} x_{ij} = 1 \quad \forall i \tag{24}$$

$$\sum_{i\neq j} x_{ij} = 1 \quad \forall j \tag{25}$$

However, this formulation is not complete since it allows multiple cycles (subtours) rather than one big route through all stations within one cluster. To handle this condition, subtour elimination constrains (SEC) should be included to provide one complete feasible rebalancing route.

$$\sum_{i\in S_C}\sum_{j\notin S_C} x_{ij} \geq 2 \qquad \forall S$$

where S_C is a subset of stations in cluster C. The SEC enforces that at least one route is going from a station in subset S_C to a station outside S_C. The number of SEC increases exponentially with the number of stations. For the bike sharing system with clusters of size up to 50 stations, generating all SECs is therefore impossible.

To solve this problem, we build SECs as lazy constrains. Unlike normal constrains which are generated in advance, lazy constrains are activated and added when a violated SEC is found from a feasible solution. We add the lazy constraints inside a callback function on the platform of Gurobi 6.5.0 Optimizer [17]. The callback function checks whether the candidate feasible solution is one complete tour. If not, a SEC is added for each subtour.

4. EXPERIMENTAL RESULTS

To validate the efficiency and effectiveness of our proposed method, extensive experiments are performed on real world NYC CitiBike trajectory data of 328 stations from July 2013 to May 2015. Among the 499 weekdays and 199 weekends, three sets of 40 weekdays and 40 weekends are randomly selected as testing set and the rest are selected as training set for bike usage demand prediction. For the problem of vehicle route optimization, a set of randomly selected 100 days records are selected to test the effectiveness and efficiency of our AdaCCKC algorithm and one day case study is conducted to test our optimization method. All experiments are conducted on a PC 7 with an Intel(R) Core i7-4790 CPU, 3.6 GHz, and 16 GB RAM running 64-bit Windows 10 system.

4.1 Experimental Data

We conduct our experiments on trip history data, station status data and meteorology data from NYC with their statistics presented in Table 2.

Citi Bike Data. Citibike transactions are generated by NYC Bike Sharing System which is public available from Citibike official website. 15.24 million transactions are extracted from July 1^{st} 2013 to May 28_{th} 2015. This data set contains the following information: station id, bicycle pick-up station, bicycle pick-up time, bicycle drop-off station and bicycle drop-off time. The station status is monitored every 5 minutes which contains the information of station in service status, current available bikes and station capacity. The capacity of all rebalancing vehicles is set to be $VC = 25$ in our experiments.

Table 2: Details of the datasets

Data Source		New York City
Time Span		7/1/13 to 5/28/15
weekdays (weekends)		499 (199) days
Bike Data	# Stations	328
	# Bikes	6,000
	# Trips	15+ million
Meteorology Data	Heavy snowy (rainy)	60 hours
	snowy (rainy)	927 hours
	Foggy (mist)	739 hours
	Cloudy (sunny)	13,495 hours
	Temperature	[1.9, 97] ^oF
	Visibility	[0.2, 10] mile
	Wind Speed	[3.5, 65] mph
	Humidity	[11%, 100%]

Hourly Weather Reports. The NYC weather report data covers the time period from July 2013 to May 2015 which consists of the hourly weather report formatted as (time, weather condition, temperature, humidity, wind speed, visibility). The missing meteorology data is completed according to the previous weather report and the missing wind speed is estimated by the average value of its previous and next report. Due to the unbalanced weather condition records, weather conditions are clustered into four categories based on their suitability for outdoor bicycling (see Section 3.1.1).

4.2 Baselines & Metric

The methods proposed in our work to predict the station level pick-up demand and drop-off demand are denoted as multi-similarity-weighed KNN (**MSWK**) predictor and inter station bike transition (**ISBT**) predictor. In order to confirm the effectiveness of our models, we conduct experiments to compare our methods with the following baselines:

Multi-similarity-based inference (MSI) [14]: The MSI considers the similarity of weather, temperature, wind speed and time. Its similarity function is the multiplication of these three similarities and the weight of different factors are not studied.

Multi-similarity-equally-weighted KNN (MSEWK): The MSEWK method treats different factors ($\lambda_1, \lambda_2, \lambda_3$ see Section 3.1.1) with equally weight. This baseline is conducted to test the effectiveness of factors weight learning (Equation (14)).

Inhomogeneous Poisson Process Inference (IPPI) [1]: The IPPI method simulates the time between pick-up (or drop-off) events based on inhomogeneous Poisson process, choosing the estimated average rate of pick-ups and drop-offs from history trip records.

Historical Mean (HM) [10]: The HM method takes the average bike demand of historical records as prediction value without considering other influential factors.

The multi-similarity-weighed KNN (MSWK) predictor is also studied as a baseline to compare our inter station bike transition (ISBT) predictor for drop-off demand prediction.

Metric: The metric we adopt to measure the performance is the mean absolute error of bike demand of each time slot.

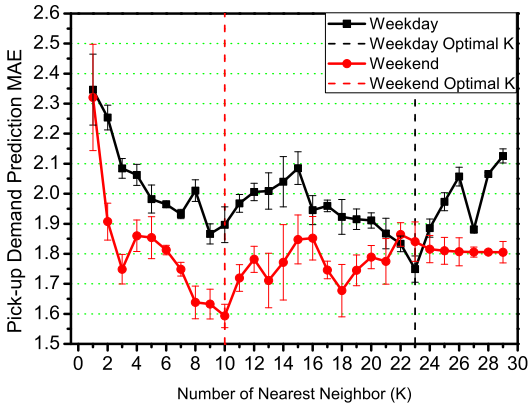


Figure 6: Performance comparison of MSWK predictor with different K.

The metric is denoted as MAE which is defined as follows:

$$MAE = \frac{1}{T}\sum_{t=1}^{T}\frac{\sum_{i=1}^{m}|\hat{s}_i.d(t) - s_i.d(t)|}{m}$$

Here $s_i.d(t)$ is our ground truth of bike pick-up or drop-off demand of station i during time slot t while $\hat{s}_i.d(t)$ and is the corresponding prediction value.

4.3 Performance Comparison

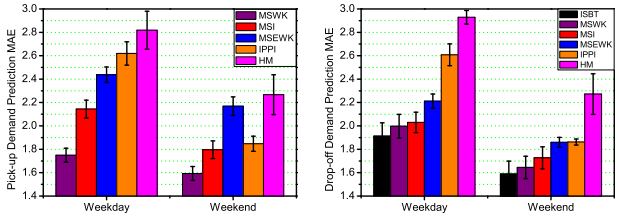
4.3.1 *Station level bike demand prediction.*

Station level bike pick-up demand prediction. As can be seen from Figure 6, the optimal number of nearest neighbor K for weekday pick-up demand prediction is 23 with minimum MAE of 1.74905 and the optimal number of nearest neighbor for weekend pick-up demand prediction is 10 with minimum MAE of 1.59323. A small K takes insufficient records to provide an accurate prediction result while a large K introduces unnecessary abnormal records. The performance comparison for pick-up demand prediction between our proposed MSWK and baselines is summarized in Figure 7(a). From Figure 7(a), we can see that the error rate obtained under our proposed MSWK is much lower than all the baselines with a significant margin. The performance comparison between MSWK and MSBI indicates that it is not necessary to consider the weighted records of all training data pool, which introduces abnormal records and lowers prediction accuracy. The comparison between MSWK and MSEWK proves the effectiveness of our similarity weight training process. Moreover, the multi-source prediction models (MSWK, MSBI and MSEWK) are better than the traditional one-factor statistical prediction models (IPPI, HM) by introducing multiple meteorology factors.

Station level bike drop-off demand prediction. The proposed ISBT method for bike drop-off demand prediction lowers the error rate for weekdays prediction with $MAE = 1.9133$ and for weekends prediction with $MAE = 1.5889$ compared to MSWK and other baseline algorithms (see Figure 7(b)). The performance of ISBT can be further improved by collecting more sufficient trip data to generate better 2-Gaussian fits for station-station connections.

4.3.2 *Station inventory rebalancing optimization.*

Constrained K-centers Clustering. The effectiveness and efficiency of our proposed CCKC are presented in Figure 8. As can be seen, For a given number of vehicles, we are able to cluster the optimizable stations and discover outlier station efficiently with CPU running time less than 0.1 second. As the number of scheduled vehicles increases, the number of outlier stations decreases fast. The CCKC algorithm can help decide the minimum number of vehicles to cover all target stations or to balance the operation cost and the number of outlier stations.



(a) Pick-up demand (b) Drop-off demand

Figure 7: A performance comparison of bike pick-up and drop-off demand prediction.

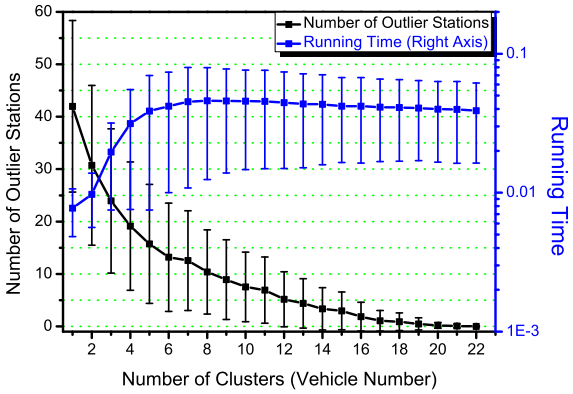
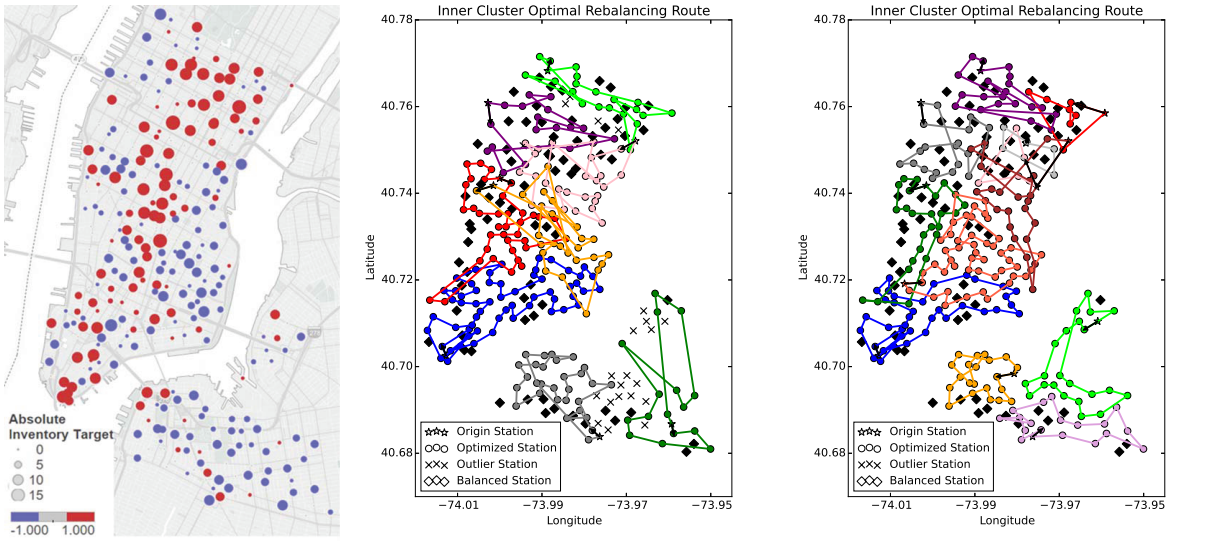


Figure 8: The efficiency of CCKC clustering

Inner Cluster route optimization (A case study). We present one day NYC Citi Bike sharing system of 328 stations with their station inventory target calculated (see Figure 9(a)) as an example. In Figure 9(a), the red dots represent stations that need vehicle deliveries and blue dots represent stations that need vehicle pick-ups. 70 stations with zero inventory targets marked as gray dots in Figure 9(a)) are self-balanced stations and are not studied in our clustering and optimization progress. Figure 9(b) presents our optimization results based on CCKC algorithm with specified 8 vehicles. The CCKC algorithm clusters stations into different groups but 8 vehicles can not cover all stations. 22 outlier stations are discovered and marked as "X" when only 8 vehicles are scheduled (see Figure 9(b)). The self-balanced stations are marked as "diamond". The stations belong to the same cluster have the same color and are rebalanced by one vehicle with its routes represented by the lines. The first visited stations are marked as "star" from which an arrow is pointed out toward the second visited station that specifies the rebalancing route direction. Figure 9(c) represents the situation when all stations are optimized with 12 vehicles based on our AdaCCKC clustering and optimization models. Table 3 summarizes the optimization results including number of Lazy Constrains (LC), optimal objectives (Obj), CPU running times (CT) and objective gap (Gap) between baselines and MINLP model for inner cluster route optimiza-



(a) Station Target Distribution (b) Route with outliers ($VN = 8$) (c) Route without outliers ($VN = 12$)

Figure 9: Vehicle Rebalancing Route Optimization (Case Study)

tions of 12 vehicles in Figure 9(c). The sizes of the station clusters range from 5 to 48. As can be seen from our case study, the optimization is very efficient for cluster size under 30 and it is able to solve for cluster of size up to 48. The effectiveness of our optimization model is compared to Nearest Neighbor Insertion Algorithm (NNIA) [12] and Genetic Algorithm (GA) [8], which are supposed to be efficient for vehicle routing optimization problems. In our case, our proposed MINLP model with lazy constrains is much more effective than NNIA and GA by providing optimal routs with much smaller traveling distances.

Table 3: Efficiency of inner cluster optimization

Cluster Size	MINLP with LC			NNIA		GA	
	LC	Obj (mile)	CT (s)	Obj (mile)	Gap (%)	Obj (mile)	Gap (%)
5	0	4.48	<1	4.48	0	4.48	0
6	2	4.23	<1	4.75	**12.3**	4.23	0
8	6	4.91	<1	5.07	3.3	4.91	0
21	10	9.58	40	14.75	5.4	9.84	2.7
20	37	10.21	210	10.39	1.8	10.97	7.4
25	26	11.21	271	11.73	4.6	12.2	8.8
22	36	8.32	401	9.49	**14.1**	8.87	6.6
23	35	17.83	752	21.4	**20.0**	18.28	2.5
19	29	12.16	1299	13.78	**13.3**	12.38	1.8
23	47	15.25	3469	16.64	9.1	15.84	3.9
38	69	18.96	7623	20.17	6.4	21.45	**13.1**
48	67	16.25	10163	19.44	**19.6**	18.81	**15.8**

5. RELATED WORK

There is an increasing interest in optimization problems arising in bike sharing systems. Below we describe some related studies that have been accomplished on bike demand prediction and rebalancing optimization.

Bike Demand Prediction. The early research on bike sharing system focused on the studies of daily bike demand forecasting using data mining techniques and classical empirical statistical methods. Yutaka [16] and Juan [11] built multi-factor statistical models for bicycle demand prediction by considering the influence of weather and geography. Liu et al.(2015) [15] built a spatial station demand prediction model by combining multiple static factors of surrounding environment and public transportation networks. The dynamic bike demand prediction for station inventory target requirements were investigated by implementing statistical models on historical trip data [1, 20]. The weather conditions were considered as influence factors on bike demands and integrated into prediction models [24, 14, 24]. However, among these multi-factor prediction models, the effectiveness of different influential factors has not been carefully studied and the different types of trips (commuters and exercisers) with different trip durations have not been considered to further improve the bike demand prediction accuracy.

Bike Rebalancing Optimization. For re-balancing the inherent asymmetry bicycle demand with minimum operation cost, Kaspi [13] explored bicycle reservation policies and suggested users visit the least loaded stations. Reservation could be denied or the destination could be diverted if no vacant lockers were expected to be available at the original destination. Also, Waserhole [25] and Singla [22] presented different dynamic pricing mechanisms that incentivized users to redistribute bikes by providing alternate rental prices.

Recent work on bike station re-balancing problem mainly focused on minimizing the total dispatching travel distance. Most bike station re-balancing problems focused on the static cases. Chemla [3] proposed a heuristic method for the single-vehicle static re-balancing problem, where each station could be visited more than once and could be used as a buffer in which bicycles are stored for a later visit. Their algorithm was based on a solution of the relaxed problem solved by a branch-and-cut procedure for system scale up to 60 stations. Erdogan [7] investigated a single capacitated vehicle routing problem that allowed the final bicycle inventory at each station to be between given lower and upper bounds. Rainer et. al [18] presented neighborhood search heuristics

that considered optimal loading operations. For large scale problems, Forma [9] proposed a 3-step model. The stations were first clustered according to geographic information as well as inventory capacity, and then the routing problem was solved within each cluster and consecutive clusters. In addition, Raviv (2013) [19] studied an extended objective that minimized a weighted sum of expected number of un-served stations and traveling distance simultaneously. However, rebalancing optimization with outlier stations for large scale bike systems has not been studied.

6. CONCLUSION

In this paper, we developed a multi-source data smart optimization approach for addressing the station rebalancing problem in bike sharing systems. Specifically, we first proposed a Meteorology Similarity Weighted KNN (MSWK) regressor for predicting station pick-up demand. Also, we provided an Inter Station Bike Transition (ISBT) model to predict the drop-off demand. Then, the station inventory target levels were determined based on the predicted traffic flows, and were used for the following rebalancing task. Moreover, we developed an AdaCCKC clustering algorithm that can separate infeasible outlier stations and group the remaining stations into different clusters. With AdaCCKC, the large-scale multiple vehicle routing problem is reduced to inner cluster one vehicle routing problem with guaranteed feasible solutions. The vehicle rebalancing route is then optimized by a MINLP model with lazy constrains. Finally, the extensive experiments on real-world weather data as well as the trip data from the NYC Citi Bike System of 328 stations showed the advantages of our approach for bike demand prediction and large-scale bike rebalancing optimization.

7. ACKNOWLEDGMENTS

This research was partially supported by Futurewei Technologies, Inc. Also, it was supported in part by Natural Science Foundation of China (71329201) and the Rutgers 2015 Chancellor's Seed Grant Program.

8. REFERENCES

[1] R. Alvarez-Valdes, J. M. Belenguer, E. Benavent, J. D. Bermudez, F. Muñoz, E. Vercher, and F. Verdejo. Optimizing the level of service quality of a bike-sharing system. *Omega*, 2015.

[2] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *SIGKDD 2004*, pages 59–68. ACM, 2004.

[3] D. Chemla, F. Meunier, and R. W. Calvo. Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization*, 10(2):120–146, 2013.

[4] M. Dell'Amico, E. Hadjicostantinou, M. Iori, and S. Novellani. The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. *Omega*, 45:7–19, 2014.

[5] P. DeMaio. Bike-sharing: History, impacts, models of provision, and future. *J PUBLIC TRANSPORT*, 2009.

[6] G. Erdoğan, M. Battarra, and R. W. Calvo. An exact algorithm for the static rebalancing problem arising in bicycle sharing systems. *EJOR*, 2015.

[7] G. Erdoğan, G. Laporte, and R. W. Calvo. The one-commodity pickup and delivery traveling salesman problem with demand intervals. 2013.

[8] Z. Fanggeng, L. Sujian, S. Jiangsheng, and M. Dong. Genetic algorithm for the one-commodity pickup-and-delivery traveling salesman problem. *COMPUT IND ENG*, 2009.

[9] I. A. Forma, T. Raviv, and M. Tzur. A 3-step math heuristic for the static repositioning problem in bike sharing systems. *TRANSPORT RES B-METH*, 2015.

[10] J. Froehlich, J. Neumann, and N. Oliver. Sensing and predicting the pulse of the city through shared bicycling. In *IJCAI' 2009*.

[11] J. C. GarcÍła-Palomares, J. GutiÍęrrez, and M. Latorre. Optimizing the location of stations in bike-sharing programs: A {GIS} approach. *Applied Geography*, 35(1ÍC2):235 – 246, 2012.

[12] S. Joshi and S. Kaur. Nearest neighbor insertion algorithm for solving capacitated vehicle routing problem. In *INDIACom*, pages 86–88. IEEE, 2015.

[13] M. Kaspi, T. Raviv, and M. Tzur. Parking reservation policies in one-way vehicle sharing systems. *TRANSPORT RES B-METH*, 2014.

[14] Y. Li, Y. Zheng, H. Zhang, and L. Chen. Traffic prediction in a bike-sharing system. In *23rd SIGSPATIAL*, GIS '15, 2015.

[15] J. Liu, Q. Li, M. Qu, W. Chen, J. Yang, X. Hui, H. Zhong, and Y. Fu. Station site optimization in bike sharing systems. In *ICDM 2015*. IEEE, 2015.

[16] Y. Motoaki and R. A. Daziano. A hybrid-choice latent-class model for the analysis of the effects of weather on cycling demand. *Transportation Research Part A: Policy and Practice*, 75:217 – 230, 2015.

[17] G. Optimization et al. Gurobi optimizer reference manual. *URL: http://www.gurobi.com*, 2016.

[18] M. Rainer-Harbach, P. Papazek, B. Hu, and G. R. Raidl. *Balancing bicycle sharing systems: A variable neighborhood search approach*. Springer, 2013.

[19] T. Raviv, M. Tzur, and I. A. Forma. Static repositioning in a bike-sharing system: models and solution approaches. *EURO Journal on Transportation and Logistics*, 2(3):187–229, 2013.

[20] J. Schuijbroek, R. Hampshire, and W.-J. van Hoeve. Inventory rebalancing and vehicle routing in bike sharing systems. 2013.

[21] S. A. Shaheen, S. Guzman, and H. Zhang. Bikesharing in europe, the americas, and asia. *Transportation Research Record: Journal of the Transportation Research Board*, 2143(1):159–167, 2010.

[22] A. Singla, M. Santoni, G. Bartók, P. Mukerji, M. Meenen, and A. Krause. Incentivizing users for balancing bike sharing systems. In *29th AAAI Conference on Artificial Intelligence*, 2015.

[23] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001.

[24] W. Wang. Forecasting bike rental demand using new york citi bike data. 2016.

[25] A. Waserhole and V. Jost. Pricing in vehicle sharing systems: Optimization in queuing networks with product forms. *EURO Journal on Transportation and Logistics*, pages 1–28, 2014.