2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)

Modeling and Predicting Bike Demand in Large City Situations

Dimitrios Tomaras, Ioannis Boutsis, Vana Kalogeraki
Department of Informatics
Athens University of Economics and Business, Athens, Greece
{tomaras, mpoutsis, vana}@aueb.gr

***Abstract*—Bike-sharing systems have enjoyed tremendous success in many major cities around the world today as a new means of urban public transportation offering a green and facile solution for daily commuters and tourists. One common problem featured in these systems is that the distribution of bikes among stations can be quite uneven, due to topography, rush hours or during the occurrence of major events around the city. This often results in shortage of bikes or bike parking racks. An unbalanced bike system means an unreliable form of transportation and disappointed users. Existing works in the literature are limited as they are not designed to handle fluctuating, high or unpredictable demand during large city events that typically affect multiple stations and require rebalancing in real-time, during the event, to ensure seamless operation. In this work, we present "SmartBIKER", a cost-effective framework for bike sharing systems focusing on major city events. SmartBIKER models bike demand trends during major events, identifies bike stations with low or high demand using a trend forecasting model and determines a relocation strategy that minimizes the relocation cost while maximizing the utility of the stations. Our experimental evaluation shows that our approach is practical, efficient and outperforms state-of-the-art relocation and prediction schemes.**

***Index Terms*—Bike Sharing Systems**

I. Introduction

In recent years we are witnessing a growing interest in smart transport and traffic management systems in urban cities. These systems aim at providing a better environment for citizens using environmental-friendly ways of transportation and improving the quality of life by reducing travel times and eliminating traffic congestion [1], [2]. Furthermore, they can present important guidance to city operators for making efficient usage of city resources and urban planners for further improving the infrastructure management [3]–[5]. One of the most representative systems towards this direction is *bike sharing systems* which have shown significant success in several major cities, such as New York City [6] and Dublin [7].

Bike Sharing Systems. Bike sharing systems allow citizens and visitors to rent bikes for short-term using bike stations scattered around an urban city in return for a low-cost fee or no charge at all [8]. This way users can move across the city by picking up a bike from a station near their origin and cycle up to their destination where they drop-off the bikes at the nearest station. Encouraging such cutting-edge transportation options for getting around the city not only benefits residents and visitors, it highly reduces the CO_2 emissions in the city, cuts down parking costs and improves the inhabitants' health.

One of the main challenges of these systems is the issue of modeling and predicting the bike inventory demand to achieve efficient operation. The problem in bike sharing systems is that the stations commonly experience fluctuating demand that leads to an imbalance between supply and demand. For instance, bike stations near universities are typically empty in the afternoon as several students use bikes to return home. On the other hand, it might be impossible to find empty racks to park the bikes in the morning. This is a challenge for service providers as they have to deploy several trucks to redistribute bikes among stations so that the stations are neither full nor empty and ensure that the system functions smoothly. However, redistributing bikes can be an extremely costly procedure [9] as it requires both workforce and large vehicles that move across the city to make sure that stations needing bikes are constantly resupplied.

Large City Situations. With the extensive growth of technology involved in the city infrastructure, bike sharing systems act as sensors of the pulse of human activity, since citizens increasingly resort to bikes during *large city situations* as an alternative means of transportation. Examples of such large city situations are, but not limited to, large social events (*i*.e., concerts, exhibitions, marathons), where constraints imposed by factors such as shutdowns of parts of the city, changes in the other local modes of transport, changes in the user demands, etc., can all affect unexpectedly the bike stations' inventory and demand, despite the regulation processes applied by bike sharing system operators. Thus, the key challenge is how to sustain the system operation when such large city situations occur and to continue operating seamlessly, without leading to a **situation** of *"bike shortage"* in the city.

Why modeling bike demand in large city situations is important? The first question we need to answer is *what is the key reason to model and predict bike demand during such situations*. Typically, bike sharing systems operators sign contracts with the respective city authorities in order to employ and setup bike stations in the city area. However, they face financial penalties when adjacent station outages occur, for instance, for more than one hour [10]. Therefore, it is important for these systems to operate robustly even when an unexpected situation occurs, where there are dynamic changes in the demand and the distribution of the demand cannot be

978-1-5386-3224-6/18/$31.00 ©2018 IEEE

captured by adopting well-known statistical distributions, as we show in our experimental evaluation. Consequently, it is of great importance to model and predict the bike demand during such city situations.

This paper tackles the modeling and prediction of bike demand problem during large city situations. This is a challenging problem: First, we need to understand bike usage and model the supply-demand behaviour at the bike stations when such large and dynamic city situations occur. The second challenge is to predict bike demand at the stations based on the city situation. Bike demand prediction is attributed to several factors including type of event, location, day, time, etc. However, as we illustrate extensively in section II.B, bike demand does not follow a known mathematical distribution and therefore, we need to resort to a more sophisticated model for predicting demand. Finally, our goal is to exploit the modeling and prediction processes to implement efficient relocation strategies that fulfill the city requirements, considering both operational costs and system performance.

Prior work limitations. Despite some recent works that apply prediction methodologies to estimate the bike trip demands [11]–[15], the aforementioned studies are limited as they aim at minimizing only the relocation cost based on the predicted demand without concurrently considering the costs for the bike operator. Moreover, they focus on modeling and predicting the demand on typical days which is different from our problem where we focus on major or unpredictable city situations. City situations are more complex and may provoke stress conditions nearby bike stations, since there is no information available *a priori* for the bike demand. Some of studies assume that bike demand follows known statistical distributions, such as the Poisson Distribution during city situations [13], which as we show experimentally this is not correct. A few recent works on situation recognition in the city using hierarchical multimedia data [16] or exploring spatio-temporal correlations in the streaming data [17] have also been proposed. The problem of situation recognition is orthogonal to our work and our approach builds upon these works. However, they focus solely on recognizing a city situation rather than providing policies to handle such stress conditions for a disruption tolerant bike system service under large city situations, which is the focus of our work.

Contributions. In this work, we propose "SmartBIKER", a framework to address the problem of modeling and predicting bike demand at stations during large city situations. The key idea is to use historic bike sharing data during city situations to model and predict the per bike station demand, and propose an effective bike relocation route strategy so that bike availability is maximized while minimizing the respective travel costs for the operator. Our contributions are summarized as follows:

- We propose "SmartBIKER"(Smart BIKE Relocation), a novel framework for bike sharing systems that aims to model, predict and propose appropriate strategies to handle unexpected bike demand during large city situations.
- We model the factors that represent accurately a city situation.
- We propose a prediction model incorporating Holt's trend forecasting technique for the demand behaviour at bike stations when large city situations occur.
- We prove that the bike relocation problem is NP-hard and propose an efficient polynomial algorithm that enables efficient relocation of the bikes among stations and reduces the solution search space.
- We perform an extensive experimental evaluation of our approach and show that "SmartBIKER" effectively meets the requested demands by eliminating the stations that suffer from empty bikes while maximizing the benefit for the operator, has low overhead, increases the utility of the bike stations and outperforms its competitors by 46% in terms of accuracy.

II. Preliminaries

In this section we describe our system model and then present real-world examples that illustrate the factors affecting bike usage during major city situations.

A. System Model

Bike Stations. We assume a Bike Sharing System comprising a number of bike stations $b_i \in \mathcal{B}$. Each bike station b_i is characterized by the tuple: $\langle lat_i,\ lon_i,\ cap_i,\ avail_i,\rangle$, where lat_i, lon_i represent the geographical coordinates of the station, latitude and longitude, and cap_i denotes the bike capacity of the station (*i.e.,* the amount of bike racks of the station) and $avail_i$ reflects the current bike availability at the station.

Bike Network. Similar to related works [14] we use a weighted directed graph $\mathcal{G} = (\mathcal{B}, \mathcal{E})$ to represent the bike network. Each vertex in $\mathcal{B}$ represents a bike station $b_i \in \mathcal{B}$. The respective edges $e_{ij} = (b_i, b_j) \in \mathcal{E}$ represent the connections between bike stations b_i and b_j. Each edge e_{ij} is characterized by w^t_{ij}, which denotes the fraction of bike trajectories from station b_i to station b_j at time instance t(these can vary depending on the hour of the day), and d_{ij}, which determines the harvesine distance between the bike stations b_i, b_j.

Bike Station Net Flow. A key feature of bike stations is their demand, expressed as pick-up or drop-off. We denote the bike drop-off demand at each station b_i at timeslot t as dd^t_i. The drop-off demand is calculated as the frequency df^t_i with which bikes become available at station b_i during time period t, *i.e.,* number of bikes that arrive at b_i within time interval t, divided by the time during which there are available parking racks at the station: $dd^t_i = \frac{df^t_i}{da^t_i}$. Similarly, we denote the pick-up demand as pd^t_i, as the frequency with which bikes depart from the bike station b_i, *i.e.,* number of bikes that depart from the station divided by the time that there are bikes available during the time interval, as: $pd^t_i = \frac{pf^t_i}{pa^t_i}$. Finally we denote the difference between the drop-off demand and the pick-up demand as net-flow $nf^t_i = dd^t_i - pd^t_i$. The net flow metric denotes how balanced the bike stations are. When the value of net flow at all stations is zero this indicates that the entire bike demand, *i.e.,* bikes that arrive and depart, is balanced.

Large City Situations. A city situation[1] is defined as a set of spatio-temporal events that affect bike demand at multiple stations concurrently [16]. Such events include music festivals, concerts, marathons, sports events, road closures, etc., since a large portion of the users move toward these stations. Each city situation z_e is characterized by the following features: i) the time domain $timestamp_e$ which is the situation starting time, ii) the geo-spatial bounds of the city situation, denoted by the tuple $\langle lat_e, lon_e, R\rangle$, where lat_e, lon_e are the geospatial coordinates of the city situation and R is the walking distance from the city situation, and finally, iii) an appropriate metric for quantifying the difference in bike demand.

Influence factor. We introduce the *station influence* metric $l^{t_e}_{i,z_e,R}$ to represent whether a station is influenced during a city situation, estimated as the ratio of the sum of the drop-off and the pick-up demand when the city situation occurs divided by the sum of the drop-off and pick-up demand on typical days for the same time domain t_c. High station influence values indicate that a greater number of bike trips is expected to begin and end at this station. This is computed as:

$$l^{t_e}_{i,z_e,R} = \frac{dd^{t_e}_i + pd^{t_e}_i}{dd^{t_c}_i + pd^{t_c}_i} \tag{1}$$

Then, we define the *influence factor* of a city situation, as the average station influence value (Eq. 1) of all bike stations $b_i \in \mathcal{I} \subset \mathcal{B}$, where $\mathcal{I}$ is the set of all bike stations within walking distance R from the city situation. Greater influence factor values denote high demand for bikes at stations near the city situation. More formally,

$$l^{t_e}_{z_e,R} = \frac{\sum_i^{|\mathcal{I}|} l^{t_e}_{i,z_e,R}}{|\mathcal{I}|} \tag{2}$$

Essentially, the influence factor captures the percentage increase in the stations' bike demand when the situation occurs compared to a typical day.

B. Modeling Real-world City Situations

Our work is motivated by the fact that there are several factors in the real world that affect bike usage in large cities which can cause significant imbalances across the bike stations [13]. Our analysis has shown, that, even under normal operation, the station usage in terms of bike demand significantly varies even during the day. For example, bike stations nearby metropolitan railway stations have higher drop-off supply during morning rush-hours, since commuters use bikes for travelling from the suburban areas to the center of the town. Furthermore, different usage patterns are exhibited on weekdays and weekends [18]. However, when a city situation occurs, the demand behaviour at the bike stations in the geospatial area of the situation is significantly affected. In order to quantify the influence of each city situation on the demand of the nearby stations, we utilize the *influence factor* metric.

We draw our examples from large city situations occurring in two major cities, New York City and Dublin City, where bike sharing systems are extensively utilized. Table I illustrates the influence factor for different types of city situations in the cities of New York City and Dublin. Essentially, the influence factor of a city situation captures the percentage increase in the stations' bike demand at the situation day over a typical day; the higher the IF value the higher is the bike demand at stations within walking distance R from the city situation. As it may be observed, parades and exhibitions increase the demand in bike up to 4.1 and 3.8 times respectively compared to days with regular usage and no major city events.

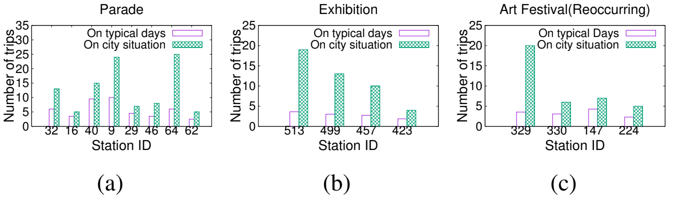
City situation	I.F.	Date	City
Parade	4.16	17 Mar. 2017	Dublin
Exhibition	3.839	22 Nov. 2013	NYC
New York Career Fair	2.966	4 Nov. 2013	NYC
Festival (Re-occurring)	2.849	Oct.-Nov. 2013	NYC
Marathon	2.762	3 Nov. 2013	NYC
Exhibition(Re-occurring)	2.6	Oct.-Nov. 2013	NYC
Music Event	2.3	5 Nov. 2013	NYC

TABLE I: Influence factor for different types of city situations

In the following, we look closely at four types of city situations with the highest influence factor. We graphically illustrate the number of trips in New York city and Dublin City on casual days and on four different city situations: (a) a parade day, (b) an exhibition day, (c) an art festival day and (d) a reoccurring exhibition event. These were chosen as they exhibited the highest influence factors, as indicated in table I. The parade occurred on March 17, 2017, namely ”St' Patricks Day Celebration Parade” in Dublin City, the art festival, namely ”Marfa Dialogues”, was a reoccurring city situation between October and November 2013, the exhibition occurred on November 22nd, 2013 namely ”Melt To Earth”, and the reoccurring exhibition event, namely ”Why We Fight: Remembering AIDS Activism” occurred between October and November 2013. In Figures 1a,1b,1c & 3a, we display the number of trips for the above city situations, while in Figures 2a,2b,2c & 3b we display the corresponding station influence. We plot the typical number of trips at these stations by drawing the average number of trips (that either end or start at each station) for the same hour of day the city situation starts. Due to lack of space, we also plot the per hour bike demand only for the reoccurring exhibition event for two of the bike stations that are highly affected (station 525 & 72), as illustrated in Figures 4a & 4b.

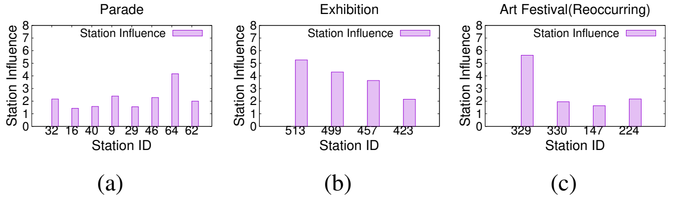
There are several important observations that we can draw. First, different bike stations exhibit different behaviour during large city events. For example, Figure 5 illustrates the effect of the St' Patricks parade on the bike usage at the roads nearby St' Patricks Church. Red circles denote stations with higher demand during the celebration day, whereas blue circles denote unaffected stations and black circles denote stations that were closed due to the event. As we observe, the volume of trips at bike stations near the route of the parade is higher compared to other stations in the wider city area. We also

[1] In the paper we use the terms situations and events interchangeably to represent large city situations



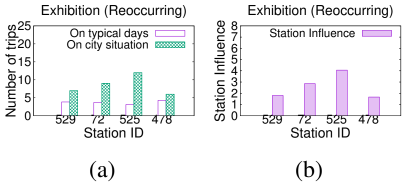
(a) (b) (c)

Fig. 1: Number of trips on various city situations: (a) Parade, (b) Exhibition, (c) Art Festival(Reoccurring).



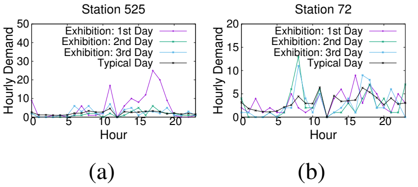
(a) (b) (c)

Fig. 2: Station influence on city situations: (a) Parade, (b) Exhibition, (c) Art Festival(Reoccurring).



(a) (b)

Fig. 3: (a) Number of trips, (b) Station Influence for exhibition event.



(a) (b)

Fig. 4: Hourly bike demand on exhibition days for two bike stations.

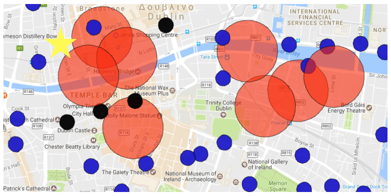


Fig. 5: Volume of bike trips on Parade day. I.F. = 4.16.

observe from Figures 1a & 2a that the impact of the situation ranges from 1.5 to 4.16 at the bike stations, which indicates that the stations that exhibit high demand, compared to the typical days, highly varies depending on the time and location of the city event. Also, in Figures 4a & 4b we observe that the bike demand at the stations follows a different distribution on every event day. This makes it difficult to adopt a known statistical distribution that could precisely model and therefore predict the bike demand at stations during a large city situation, and that is the reason why existing prediction techniques that assume known distributions [13] are limited. Finally, we may conclude that in cases where large city situations occur, the demand at the stations (either for dropping-off or picking-up bikes) varies significantly. This finding indicates that citizens commuted to city situation location using bikes, rather than other means of transportation (since road closures have been applied in specific areas nearby the city situation location).

III. SmartBiker Approach

A. Demand Prediction in City Situations

Our goal is to estimate which of the stations in the geospatial area of the event will be affected and rebalance these only, rather than the entire set of stations. A few recent techniques proposed for predicting the demand at bike stations using non-parametric statistical regression techniques [19], random forest techniques on mobility models per station [20] and multi-similarity weighted KNN approaches [14] for the prediction process, are limited, as city situations do not present *well-defined seasonality* due to oscillations in bike demand which cannot be captured by these techniques. For this purpose, we use the Holt's Model [21], a trend forecasting technique. It applies a double exponential smoothing process to *capture changing trends* in data (unlike regression techniques) and *predict a data value* in a future time period (as used in recent works [22]). In our approach, we use the Holt's model for each station to compute the similarity of demand with historical events occurring within the same time window, in order to identify if they are affected by the city situation.

Net Flow Estimation: We estimate the net flow for each bike station b_i to identify bike stations that are unbalanced. Positive netflow denotes that users drop-off more bikes than picking-up at the specific station, while negative netflow denotes that users pick-up more bikes than dropping-off. The net flow estimation step reduces the search space of our optimization problem. We aim at pruning self-balanced stations and consider only stations with high demand. The pruning process is further supported by the usage of the influence factor at each bike station, in order to identify which stations are affected by the city situation and therefore focus on rebalancing only them in real-time.

The netflow at each station is computed using the Holt's Model considering the features already defined (the similarity of netflow with respect to previous city situations experienced in the station in the corresponding time window). We denote as $pred\_nf_i^{t_e}$ the predicted net flow at time domain t_e, which depends on the city situation occurring at that time window ($t_e \in [timestamp_e, timestamp_e + \Delta t], \forall e$). Using a collected list of netflows from historical data during previous city situations, the Holt's Model computes the predicted netflow, as shown in Figure 6. The Holt's model requires setting up two constant parameters: α, the data smoothing factor, and γ, the trend smoothing factor, which are used for fine tuning the model.

The first step to find the predicted netflow is the calculation of the smoothed value of netflow $sm\_nf_i^t$. The computation process takes into consideration the current netflow nf_i^t, the previous smoothed value of netflow $sm\_nf_i^{t-1}$, and the previous trend tr_i^{t-1} of the netflow. Equation 3 describes how the smoothed value of netflow based on previous city situations is estimated. Equation 4 suggests that the estimation of the trend of the netflow value (whether it will increase or decrease and at which degree) is based on previous smoothed values of

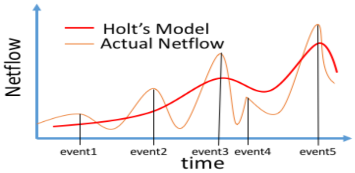


Fig. 6: Holt's Model.

netflow and helps better capture its behaviour. Since our goal is to identify the netflow on a short-term period, we utilize the Formula 5 to derive the predicted netflow of the bike station. Equations 6 & 7 just ensure that our constants α and γ are positive and less than 1.

As aforementioned, the Holt's model requires the setup of two constant parameters, which are used for fine tuning of the model. The initialization of the model with two constant values that remain unchanged would result in poor performance of the prediction model. In order to further improve it, we applied a feedback mechanism for estimating the best values for constants α and γ, $\alpha*$ and $\gamma*$ respectively. For this purpose, while training our model, we apply an iterative method, which, by sampling and checking the mean square error for the different values of α and γ, identifies the best values for $\alpha*$ and $\gamma*$. This helps us to identify the appropriate values for constants α and γ that would result in a more accurate estimation of the predicted netflow. Therefore, we use these values in Equations 3 and 4 to predict the netflow value in the time domain t using Equation 5.

$$sm\_nf_i^t = \alpha \cdot nf_i^t + (1-\alpha) \cdot (sm\_nf_i^{t-1} + tr_i^{t-1}) \tag{3}$$

$$tr_i^t = \gamma \cdot (sm\_nf_i^t - sm\_nf_i^{t-1}) + (1-\gamma) \cdot tr_i^{t-1}), \tag{4}$$

$$pred\_nf_i^t = sm\_nf_i^t + m \cdot tr_i^t, m = 1 \tag{5}$$

$$0 \leq \alpha \leq 1 \tag{6}$$

$$0 \leq \gamma \leq 1 \tag{7}$$

B. Relocation Strategy during city situations

Hereby, we present SmartBiker's relocation solution for the bike repositioning problem. We aim at minimizing the relocation cost (i.e. the distance required for the relocation fleet to travel) while maximizing the number of balanced stations based on their actual usage during large city situations. Having identified the bike stations that need rebalancing, we derive the appropriate route that will satisfy both objectives based on these stations. Finally, we prove that our problem is NP-Hard and provide a complexity analysis.

1) Problem Definition: We assume a set $U \subset \mathcal{B}$ of bike stations b_i. The travel cost among bike stations b_i, b_j is defined as the distance (in kilometers), denoted as DC_{ij}. A van, with a given capacity $Van\_Capacity$, can travel among stations and redistribute the bikes. Thus, our problem is to decide the optimal rebalancing routes that will minimize: (1) the total cost of the rebalancing process, subject to a predefined budget C, and (2) the number of unbalanced bike stations (bike stations that are empty or full) expressed as the *utility* of the system when rebalancing the stations. We formulate the problem as follows:

$$\begin{aligned}
\min \quad & F(x) = \sum_i \sum_j x_{ij} * DC_{ij}, && (8)\\
& G(x) = -\sum_j |nf_j^t * x_{ij}| && (9)\\
\text{s.t.} \quad & v_i + \sum_{j \neq i} nf_j^t * x_{ij} = \sum_{j \neq i} v_j * x_{ij}, \forall i && (10)\\
& 0 \leq v_i \leq Van\_Capacity \forall i && (11)\\
& F(x) < C && (12)\\
& \sum_{i \neq j} \sum x_{ij} \leq 1, \forall i && (13)\\
& \sum_{i \neq j} \sum x_{ij} \leq 1, \forall j && (14)\\
& x_{ij} \in \{0,1\}, \forall i \neq j && (15)
\end{aligned}$$

The binary variable x_{ij} indicates whether the edge e_{ij} has been selected for redistributing bikes from station i to station j, while, variable v_i reflects the number of bikes carried by a van after visiting station i. Our first objective (Equation 8) aims to minimize the travel cost among the edges selected for rebalancing. Our second objective (Equation 9) aims to maximize the amount of nodes that need rebalancing (balanced nodes are going to have a netflow of zero). The first constraint in Equation 10 represents the flow conservation for the bikes carried by the vans and Equation 11 states that the vans cannot exceed a certain capacity. The constraint in Equation 12 states that the total cost of the routes should not exceed a budget C. Finally, Equations 13,14 ensure that each station will be visited at most once. We provide a NP-Hardness proof for this problem at the end of this section.

2) Relocation Strategy: We propose a greedy relocation strategy. Given the bike network $\mathcal{G}$, defined in section II-A, a weighted graph is constructed where the weights w_{ij}^t are generated from historical data for the specific time window t for which we would like to predict the netflow at each bike station. In order to set the distance weights d_{ij}, we apply a K-Nearest Neighbor algorithm [23] to select the top-K Nearest Neighbors to the bike station and set their distance as weight. In this step we also perform a precomputation of the travel cost DC_{ij} between nodes in the network, which is a well-known speedup technique for any shortest path algorithm [24].

Our first objective is to minimize the relocation cost $F(x)$. We generate the set of feasible paths for a set of stations S that are influenced by the city situation and prune all those that violate the flow conservation, the van capacity and the budget constraint. We compute the travel cost for each pair of nodes in these paths offline, using the Dijkstra algorithm [25], so that the relocation cost can be estimated instantly.

The second objective is to maximize the total system utility $G(x)$, which is computed by summing up the absolute predicted net flow values for the set of the selected nodes S to traverse, given that the respective values satisfy the defined constraints. For the given set of nodes S, we aim at maximizing the total number of nodes that will be balanced with our strategy. Our approach aims at selecting those routes for which the total system utility will be the maximum despite the high demand.

Relocation Strategy Steps. Our goal is to provide a feasible solution in real-time. Computing all feasible solutions among stations for the size of the graphs we consider (real bike sharing systems consist of hundreds of bike stations) using techniques such as branch and bound, is a costly task. The idea of our relocation strategy is, that, based on an iteration over each unassigned station b_i, in which we alter the selection of the stations from set S, we select the appropriate bike stations which ensure that the constraints are fulfilled and both objectives are concurrently optimized. The pseudocode of the SmartBIKER relocation strategy is summarized in Algorithm 1 and consists of three major steps which we present below:

Step 1: As a first step, we aim at identifying which stations are affected from the city situation. We have already predicted the netflow at bike stations using our prediction method and we generate a list $\mathcal{R}$ of the affected bike stations $\mathcal{B}$ from the graph $\mathcal{G}$. We sort the list $\mathcal{R}$ we extracted, based on the predicted netflow $pred\_nf_i^t$ as: Sort($\mathcal{R}$) by $(pred\_nf_i^t)$. Then, we create an empty set $S = \{\emptyset\}$ that represents the bike stations that will be selected for rebalancing. This initialization step enables us to instantiate the needed structures for the rebalancing process.

Step 2: In the next step we aim at identifying the nodes with high drop-off demand, i.e. more bikes than usual, which can be used to balance nodes that are short of bikes and have high pick-up demand. Therefore, we use two iterators that go over the list $\mathcal{R}$ of bike stations from top to bottom and from bottom to top and we sum the netflow values of each set of stations b_i in S that they fulfill the defined constraints, as long as the budget has not been exceeded.

Step 3: In the final step we aim at improving both objectives. We continue iterating through the bike stations list $\mathcal{R}$. Upon reaching the budget constraint, any new addition of more stations will result in exceeding the budget. Thus, we examine whether we can replace the current node with one of the selected nodes to improve one of the objectives. At each iteration of our algorithm, we select the feasible routes given that they improve the utility of the system, or given that a new selected route results to the same utility as a previous, we select the route for which we have the shortest total travel cost.

Algorithm 1 SmartBIKER Relocation Strategy

```
$\mathcal{R}$ = FindAffected($\mathcal{B}, z_e$)
$\mathcal{R}$ = Sort($\mathcal{R}$) by $(pred\_nf_{r.n}^t)$;
$S = \emptyset$;
$balance = 0$;
while($\mathcal{R}_i > 0$ && $R_j < 0$ ) do
S = S∪ChooseStations($balance, b_i, b_j, pred\_nf_i^t, pred\_nf_j^t$);
$RelocationCost = F(S)$;
if($RelocationCost < C$) return $S$;
else
for (Station S.n : S) do
    $RelocationCost = F(S \cup \{b_i\} \setminus S.n)$;
    $SystemUtility = G(S \cup \{b_i\} \setminus S.n)$;
    if($RelocationCost < C$ && $SystemUtility > G(S)$)
    then $S = S \cup \{b_i\} \setminus S.n$;
return $S$;
end while
```

NP-hardness. We prove that our optimization problem is NP-Hard as the well-known 0-1 Knapsack problem can be reduced to it. Let us consider an instance of our problem. Consider a set of stations that the van will pass along in order to relocate the bikes, each one associated with a predicted netflow value (which represents the station's utility upon relocation) and a relocation cost. We identify a set of stations that maximizes the system utility and minimizes the relocation cost, i.e. we select the appropriate objects that maximize the sum of the values of the items in the knapsack so that the sum of relocation cost is smaller than or equal to a given budget. Each object can be selected one time or not at all. As it can be seen our problem can be reduced to the well-known 0-1 Knapsack Problem. Given an instance of the 0-1 Knapsack problem, we select the appropriate items that maximize the total sum of values and the sum of weights is less than or equal to a given budget. If we set the utility of bike stations as values and the relocation costs as weights, and given that we pass from each station only once, then the 0-1 Knapsack-problem is reduced to our problem. Given that 0-1 Knapsack problem is NP-Hard, then our problem is also NP-Hard.

Worst-Case Complexity. Assume n stations. First, we sort the stations according to their demand, that costs $\mathcal{O}(nlogn)$ and we iterate through the list. In the case that we have not reached the balance constraint by adding m stations, we add a new station to the set which costs $\mathcal{O}(1)$, thus it costs $\mathcal{O}(m)$. Since we have already calculated the shortest distance between stations, then the travel cost can be determined in $\mathcal{O}(m)$ and the system utility can be also determined in $\mathcal{O}(m)$. Given that we have already generated k sets of feasible routes using a path generator, then the worst case complexity of our algorithm is $\mathcal{O}(k * nlogn + k * 2m) = \mathcal{O}(k * nlogn + k * m)$. That is $\mathcal{O}(k * nlogn + k * n) = \mathcal{O}(k * nlogn)$.

IV. Experimental Evaluation

A. Dataset

1) Bike Network Dataset: We conducted our experiments using a real world dataset that includes bike trip data in New York City from the Citi Bike project [6]. The dataset contains all the necessary information regarding the bike trips, such as, the duration(in seconds), the start time and date, the stop time and date, the start station name, the destination station name, the unique station ID, the geospatial coordinates of each station (latitude and longitude) and the bike ID that performs the specific trip. Moreover, it provides demographic

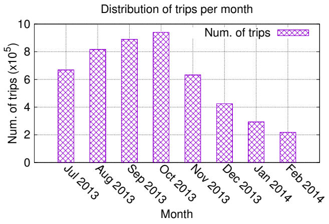


Fig. 7: Distribution of bike trips per month of the provided dataset

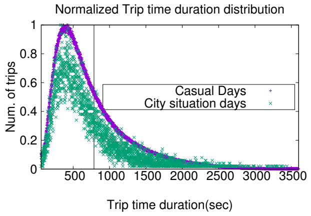


Fig. 8: Trip time duration distribution on casual days and events

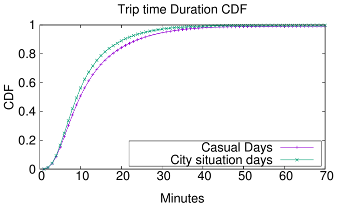


Fig. 9: Trip time duration Cumulative distribution function on casual days and events

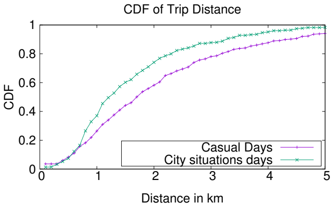


Fig. 10: Trip distance Cumulative distribution function on casual days and events

Bike Data	**Data Source:**	Citi Bike Project
	Time Span:	07/2013 to 02/2014
	#Stations #Bikes #Trips	330 6,000 4,881,484
	#Customer Trips #Subscriber Trips Average Trip Time	74 4,881,410 786.11 sec
City sit. Data	**Data Source:**	CityOfNewYork API
	Time Span:	09/2013 to 01/2014
	#City Sit. in the NYC	1522

TABLE II: Details of evaluation datasets

information regarding the users, such as the type of user (Customer = 24-hour pass or 7-day pass user; Subscriber = Annual Member), the gender (female, male, unknown) and year of Birth. The bike operator has already processed the data in order to remove trips that are taken by staff as they service and inspect the system, trips that are taken to/from any of their ”test” stations, as long as any trips that were below 60 seconds in length (potentially false starts or users trying to re-dock a bike to ensure it's secure).

The data used in our experiments expand from July 2013 to February 2014. Figure 7 shows the distribution of the number of trips in the months of our dataset. Figure 8 shows the normalized trip time distribution, from which we can observe that trips in city event days have smaller duration time than on typical days. Figure 9 illustrates that bikes are used for less time during city situation days. Figure 10 shows the trip distance cumulative distribution, from which we can observe that users are utilizing bikes for shorter distances during city situations, since the curve is higher in comparison to casual days for the same percentage of trips.

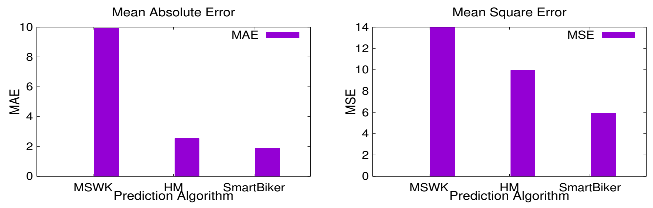
2) City situation Dataset: In order to quantify the influence of the city situations on the bike station demand, we used the API of CityOfNewYork [26] to get historical data of city situations in New York City for the respective time period of our bike station network dataset. It contains information about the name of event, the geospatial coordinates, and the corresponding time window. We acquired events that will actually have geospatial coordinates in order to quantify their influence given a geospatial bound. For the evaluation process, we acquire the city situations for September 2013 to January 2014, since these records have the appropriate form to evaluate SmartBiker's performance. Table II summarizes the datasets we used in our experiments.

B. Evaluation

We conducted a set of experiments to illustrate the benefits of our approach in terms of the quality of our prediction model and the quality of our proposed algorithm for the dual objective optimization problem. Our goal was to identify the advantages of our approach with respect to the following metrics: *a) Prediction Performance, b) Budget Constraint, c) Van Capacity Constraint, d) Influence factor, e) Number of Impacted Stations and f) Execution time*. We compare our approach with MINLP [14], [27], [28], a state-of-art method to generate the route that a van will pass to rebalance the system. MINLP minimizes the travel distance for traversing among all nodes without considering the distance that the operator is willing to tolerate per event, and the respective system utility. Thus, since MINLP may exceed this constraint we also compare our approach with a constrained version, called Constrained MINLP, to derive fairer results. Constrained-MINLP returns the route of MINLP until the constraints have been exceeded.

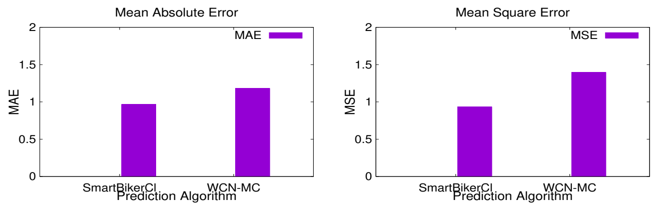
Evaluation Scenario. We have set up the following scenario. For each city situation in our dataset, we identify the influenced stations, given a bound for the influence factor value and walking distance to search for. Then, if the number of influenced stations is greater than one, we use a path generator in order to get the routes to check, given the budget, the van capacity, the influence factor value and the walking distance constraints.

1) Prediction Performance: We evaluated the performance of SmartBiker's prediction method when large city situations occur in comparison with (a) a baseline and two state-of-the-art methods: (b) MSWK, proposed in [14] and (c) WCN-MC, proposed in [13]. As a baseline, we used the historical mean(HM) of netflow as an estimator for each station. We use two common prediction performance metrics [12], the Mean Absolute Error(MAE) and the Mean Square Error(MSE). MAE illustrates how close the predicted netflow is to the eventual outcome. Furthermore, the MSE illustrates the quality of each estimator. In both metrics, values near zero are considered to be better.



(a) (b)

Fig. 11: Station level accuracy: (a) MAE & (b) MSE.



(a) (b)

Fig. 12: Cluster level accuracy: (a) MAE & (b) MSE.

Mean Absolute Error: Figure 11a illustrates the performance of our prediction method. We observe that SmartBiker outperforms both techniques in the performance of the prediction process. This is attributed to the fact that our proposed model captures better the trends of the bike demand during city situations, than when using a historical mean. Figure 12a illustrates the performance of our prediction method on a cluster approach(noted as SmartBikerCl), in order to make a fair comparison with the WCN-MC method. We observe that SmartBiker still outperforms the state-of-the-art method.

Mean Square Error: Figure 11b illustrates the quality of the estimators. We observe that the SmartBiker prediction method outperforms both techniques in terms of quality, which further validates that our proposed prediction method better captures the trends of netflow when city situations occur. Figure 12b illustrates the comparison of the quality of estimators. As we may observe, our proposed method outperforms the state-of-the-art method WCN-MC.

2) Budget Constraint: Figures 13 & 14 show the performance of our approach towards the MINLP and Constrained MINLP regarding the utility and the relocation cost as we vary the budget constraint from 500 to 1500 monetary units (illustrated with horizontal lines). For this experiments we set the van capacity to 15 and the influence factor to 3, based on the rest of our experiments. As can be observed from the figure, “SmartBIKER” has similar results in terms of utility with MINLP, although MINLP exceeds the budget constraint, as it tries to traverse all the affected nodes, providing an infeasible solution for our problem. SmartBIKER also produces better solutions for both objectives compared to the Constrained-MINLP. Moreover, SmartBIKER always produces feasible solutions that strongly outperform the other techniques in terms of relocation cost which are due to the fact that we get the best route in terms of utility among feasible solutions and also identify the proper route which minimizes the travel cost.

3) Van Capacity Constraint: Next we evaluate the performance of SmartBIKER as we vary the van capacity *i.e.*, the number of bikes the van transfer, from 5 to 25. For this experiment we set the budget to 1500 to provide a fair comparison. As can be observed from Figures 15 & 16, all approaches have a small benefit when we increase the van capacity as they can transfer more bikes. Similar to the previous experiment our approach has both better utility and reduced relocation cost for all feasible solutions. Furthermore, even for the lowest value of van capacity (van_capacity=5), we can clearly see the benefits in terms of utility maximization since even when the van capacity has such a low value our algorithm is tolerant and we are able to identify the best choice of stations that can satisfy the van capacity constraint, and therefore select the shortest route that has the best utility.

4) Influence factor: Next, we quantify the performance of SmartBIKER when the influence factor varies, thus, restricting the amount of affecting stations to relocate bikes. As can be seen in Figures 17 & 18, when the influence factor increases the utility also increases, but the travel distance remains the same. We reasoned this result to the fact that we select stations with higher demands and our algorithms succeed in identifying the best route for which the utility will be higher. At the same time, the distance still remains the same in both cases, which clarifies that our algorithm, when the influence of the city situation is high, succeeds in selecting the route that has the same distance but better utility. We also note that due to the reduced amount of stations, the MINLP approach managed to produce feasible solutions when the influence factor is set to 5. However, our approach outperformed MINLP for both objectives.

5) Number of Impacted Stations: In Figure 19 we illustrate the average amount of stations which are affected per city situation. As can be observed the amount of affected stations when the influence factor is 3, are 2.7 and 4.26 for walking distances of 500m and 1000m respectively. Respectively, when we set the influence factor to 5 the affected stations are reduced to 2.2 and 2.57. We can conclude that there are many stations that are slightly affected from the city situation, but there are also stations which are highly affected, and thus, we need a relocation procedure to rebalance them.

6) Execution time: Finally, we present the execution times of our approach over the amount of affected stations as we vary the budget of the relocation cost. As we observe from Figure 20, when the amount of affected stations increases the execution time also increases as we need to consider additional nodes. We limit the amount of affected stations to 9 as this was the maximum amount of affected nodes that we encountered during our experiments. Moreover, we observe that when the budget increases this also affects the execution time as it enables us to examine additional feasible paths among the nodes. However, as can be observed the execution time remains low, which is less than 4 seconds in all cases. Such an execution time is acceptable for the systems that

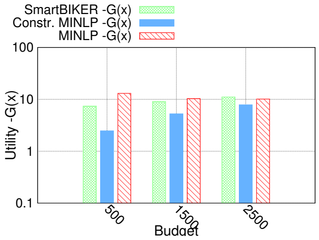


Fig. 13: Utility over Budget Constraint

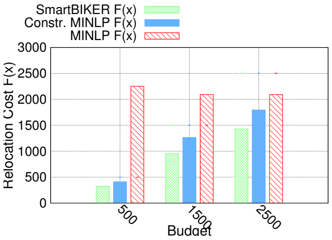


Fig. 14: Distance over Budget Constraint

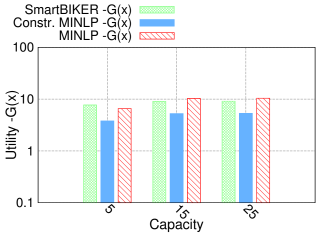


Fig. 15: Utility over Van Capacity

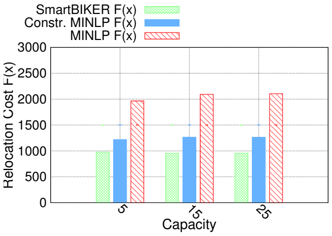


Fig. 16: Distance over Van Capacity

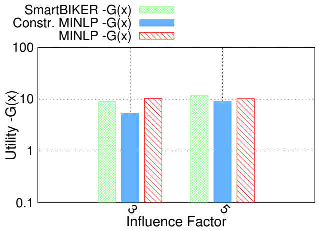


Fig. 17: Utility with Varying Influence Factor

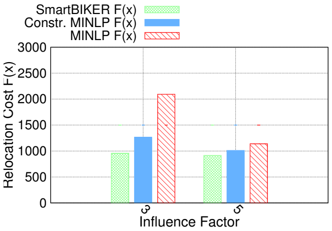


Fig. 18: Distance with Varying Influence Factor

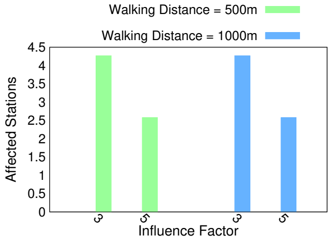


Fig. 19: Average Influenced Stations per Event

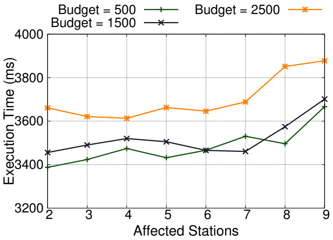


Fig. 20: Execution Times over Budget Constraint

we consider as the relocation will be performed in terms of minutes or hours.

V. Related Work

Existing state-of-the-art methods focus on either on *prediction methods* [13]–[15], [19], in which, authors propose systems that do not consider city situations and *vehicular optimization methods* [28], [29], in which, algorithms solve the relocation from other aspects such as optimal station-to-station route design and inventory management. However, these approaches are limited and cannot efficiently handle large city situations.

Vehicular Optimization methods: A few recent approaches study the relocation problem [28], [29]. The work presented in [29] focuses on minimizing the total relocation cost. However, they do not consider the utility of stations or how to tolerate high demands, as we do in our work. The authors in [28] propose an online repositioning approach that is based on Mixed Integer Linear Programming. However, their work focuses on minimizing only the number of unbalanced stations rather than minimizing the relocation cost as well, which we study in our work.

Prediction Methods: Prediction methods in literature [13]–[15], [19] address the bike relocation problem from a proactive aspect. Authors in [19] propose a hierarchical prediction model to predict the number of bikes that will be rented in a city, so that relocation can be executed in advance. However, they model and predict the bike demand during normal operation of the system without focusing on handling large city situations. The authors of [15] propose a stochastic model based on Markov chains in order to predict the demands at each station, but do not focus on modeling and handling city situations, as we do in our work. The work of [14] focuses on predicting the demand at each bike station of the system and applying a linear programming solution for the relocation optimization procedure without considering cases such as city situations. The authors in [13] propose a dynamic clustered model for predicting demands at stations. However, their model fails to capture the trends of demand during city situations, in contrast with our work.

VI. Conclusions

In this paper we presented ”SmartBIKER”, a novel cost-effective real-time framework for bike sharing systems whose objective is to model, predict and propose relocation strategies considering operational costs and system utility during large city situations. Our contributions are summarized as follows:

- We illustrated that city situations can have a significant effect on bike usage at stations in their event’s geographic proximity and cause significant imbalances.
- Modeling the stations’ bike demand during city situations can greatly help in predicting trends of demand for future city situations.
- We proved that the bike relocation problem is NP-hard and proposed an efficient polynomial algorithm that effectively reduces the search space of the solutions.
- We have shown that by considering the effect of the city situation when designing relocation algorithms it can greatly help us to efficiently handle sudden changes in urban dynamics.
- In our detailed experimental evaluation we illustrated that ”SmartBIKER” is successful in modeling, predicting and minimizing relocation costs compared to state-of-the-art methods.

Acknowledgment

This research has been financed by the European Union through the FP7 ERC IDEAS 308019 NGHCS project and the Horizon2020 688380 VaVeL project.

REFERENCES

[1] C. Boldrini, R. Bruno, and M. Conti, "Characterising demand and usage patterns in a large station-based car sharing system," in *INFOCOM Workshops*. San Francisco, CA, USA: IEEE, April 2016.

[2] M. Handte, S. Foell, S. Wagner, G. Kortuem, and P. Marron, "An internet-of-things enabled connected navigation system for urban bus riders," in *IEEE Internet of Things Journal*, vol. 3, no. 5. IEEE, 2016, pp. 735–744.

[3] C. Chen, Z. Wang, and B. Guo, "The road to the chinese smart city: Progress, challenges, and future directions," in *IT Professional*, vol. 18, no. 1. IEEE, 2016, pp. 14–17.

[4] 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," in *Omega*, vol. 62. Elsevier, 2016, pp. 163–175.

[5] A. M. Burden, R. Barth *et al.*, "Bike-share opportunities in new york city," *Department of City Planning, New York*, 2009.

[6] "Citi bike project," 2017, https://www.citibikenyc.com/system-data.

[7] "Dublin bikes," 2017, http://www.dublinbikes.ie/.

[8] "Aarhus city bikes," 2017, https://cibi.dk/aarhus-pendlercykel/.

[9] "Rebalancing cost," 2017, https://bikeportland.org/2016/09/07/portland-now-using-pedal-powered-trikes-to-help-rebalance-bike-share-stations-191007.

[10] S. M. Kaufman, L. Gordon-Koven, N. Levenson, and M. L. Moss, "Citi bike: The first two years," 2015.

[11] D. K. George and C. H. Xia, "Fleet-sizing and service availability for a vehicle rental system via closed queueing networks," in *European journal of operational research*, vol. 211, no. 1. Elsevier, 2011, pp. 198–207.

[12] J. Zhang, X. Pan, M. Li, and P. S. Yu, "Bicycle-sharing system analysis and trip prediction," in *MDM*. Porto,Portugal: IEEE, June 2016.

[13] L. Chen, D. Zhang, L. Wang, D. Yang, X. Ma, S. Li, Z. Wu, G. Pan, and J. J. Thi-Mai-Trang Nguyen, "Dynamic cluster-based over-demand prediction in bike sharing systems," in *UbiComp*. Heidelberg, Germany: ACM, September 2016, pp. 841–852.

[14] J. Liu, L. Sun, W. Chen, and H. Xiong, "Rebalancing bike sharing systems: A multi-source data smart optimization," in *SIGKDD*. San Francisco, USA: ACM, August 2016, pp. 1005–1014.

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

[16] V. K. Singh, M. Gao, and R. Jain, "Situation recognition: an evolving problem for heterogeneous dynamic big multimedia data," in *Proceedings of the 20th ACM international conference on Multimedia*. ACM, 2012, pp. 1209–1218.

[17] M.-S. Dao, K. Zettsu, S. Pongpaichet, L. Jalali, and R. Jain, "Exploring spatio-temporal-theme correlation between physical and social streaming data for event detection and pattern interpretation from heterogeneous sensors," in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2690–2699.

[18] J. W. Yoon, F. Pinelli, and F. Calabrese, "Cityride: a predictive bike sharing journey advisor," in *MDM*. Bengaluru,India: IEEE, July 2012, pp. 306–311.

[19] Y. Li, Y. Zheng, H. Zhang, and L. Chen, "Traffic prediction in a bike-sharing system," in *SIGSPATIAL*. Seattle, Washington, USA: ACM, November 2015.

[20] Z. Yang, J. Hu, Y. Shu, P. Cheng, J. Chen, and T. Moscibroda, "Mobility modeling and prediction in bike-sharing systems," in *MobiSys*. Singapore,Singapore: ACM, June 2016, pp. 165–178.

[21] D. L. Holt, "Dislocation cell formation in metals," *Journal of Applied Physics*, vol. 41, no. 8, pp. 3197–3201, 1970.

[22] R. E. De Grande, A. Boukerche, and R. Alkharboush, "Time series-oriented load prediction model and migration policies for distributed simulation systems," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 1. IEEE, 2017, pp. 215–229.

[23] K. Fukunaga and P. M. Narendra, "A branch and bound algorithm for computing k-nearest neighbors," in *IEEE transactions on computers*, vol. 100, no. 7. IEEE, 1975, pp. 750–753.

[24] D. Wagner and T. Willhalm, "Speed-up techniques for shortest-path computations," in *Annual Symposium on Theoretical Aspects of Computer Science*. Springer, 2007, pp. 23–36.

[25] E. W. Dijkstra, "A note on two problems in connexion with graphs," in *Numerische mathematik*, vol. 1, no. 1. Springer, 1959, pp. 269–271.

[26] "Nyc developers portal," 2017, https://developer.cityofnewyork.us/api.

[27] S. Ghosh, M. Trick, and P. Varakantham, "Robust repositioning to counter unpredictable demand in bike sharing systems," in *IJCAI*. New York City, USA: AAAI, 9 - 15 July 2016.

[28] M. Lowalekar, P. Varakantham, S. Ghosh, S. D. Jena, and P. Jaillet, "Online repositioning in bike sharing systems," in *ICAPS*. Pittsburgh, USA: AAAI, June 2017.

[29] D. Chemla, F. Meunier, and R. W. Calvo, "Bike sharing systems: Solving the static rebalancing problem," in *Discrete Optimization*, vol. 10, no. 2. Elsevier, 2013, pp. 120–146.