Crowd-based ecofriendly trip planning

Dimitrios Tomaras and Vana Kalogeraki
Athens University of Economics and Business
Athens, Greece
tomaras,vana@aueb.gr

Thomas Liebig
TU Dortmund
Dortmund, Germany
thomas.liebig@tu-dortmund.de

Dimitrios Gunopulos
University of Athens
Athens, Greece
dg@di.uoa.gr

***Abstract*—In recent years we have witnessed a growing interest in trip planning systems aiming at organizing daily travel schedules in smart cities. Such systems use specialized engines to find optimal means of transport between two geospatial endpoints to provide recommendations to citizens for short routes across the city. At the same time, alternative means of transportation, such as bike sharing systems, have enjoyed tremendous success since they offer a green and facile solution for daily commuters and tourists. However, one major challenge of the bike sharing systems is that the distribution of bikes among the stations can be quite uneven during rush hours or due to topography. This often results in shortage of bikes and increasing numbers of disappointed users. Existing works in the literature are limited since they only focus on predicting the demand or apply a-posteriori methods for balancing the load of stations. Furthermore, none of these works consider the benefit of these systems in concert. In this work, we present "MOToR" (MultimOdal Trip Rebalancing), a system that builds upon the OpenTripPlanner framework to incorporate dynamic transit schedule data while balancing the availability of bikes among the bike stations. Our experimental evaluation shows that our approach is practical, efficient and outperforms state-of-the-art methods for route planning.**

1. Introduction

In the recent years, the concept of smart transportation systems has gained increasing interest as a mean of enhancing the quality of daily commute for the citizens. A wide range of smart transportation applications have been developed to address various aspects of the transportation systems such as crowd based traffic information [1], [2], emergency response systems [3], trip planning systems [4], [5] etc. Moreover, recent types of transportation systems, such as bike sharing systems, have initiated a new era in the research community. It was recently shown that bike sharing systems act as sensors of human mobility across the city [6], since people resort, with an increasing rate, to using such systems as an alternative, eco-friendly and entertaining way to commute to destinations across the city.

The rapid development of applications that implement multi-modal trip planning, combining transit, pedestrian, bike and car segments, has generated significant scientific interest recently. People increasingly utilize smart trip planning systems to facilitate their commutes across the city. Real-time information about transit schedules, availability of transportation modes and the estimated time of arrival to a preferred destination, are only some of the aspects studied in the bibliography [7], [8], [9], [10]. One of the most interesting problems in smart transportation systems, is how to appropriately combine various and different types of transportation networks for route planning, in concert, not only for the benefit of the users but also for the benefit of the operation of the urban structures. The performance of an urban structure depends not only on the city's static infrastructure facilities but also on the quality and availability of such systems. For instance, a proposed route from a route planning systems may combine multiple modes of transportation, including cycling as part of the trip. In cases that no bikes or docks are available, users can easily be frustrated, for both the trip planning system and the bike network, since users will consider both of them inadequate to use for their daily needs.

The concept of multi-modal transportation comes with an additional effort which is by far more difficult than solely accommodating car drivers. Each transportation medium has different priorities and needs for street space allocation. On every street planning process, trade-offs between the corresponding benefits arise when choosing among supporting one transportation medium instead of another. For example, in a given geographical area, setting up a bike station instead of a bus stop, may reduce costs and serve more people at the same time instead of a bus stop. When designing intelligent transportation systems, it is essential to optimize more than a single objective. From one side we aim at enhancing user satisfaction from publicly available services, while on the other side we aim at reducing the operational costs and providing a healthier environment. Thus, the key challenge is to succeed in optimizing concurrently these multiple objectives with the least cost, but it is not a trivial task. Since humans are intrinsically social creatures, their mobility needs have a strong connection and impact route planning. Our goal is to enhance the multi-modal route planning process and exploit it accordingly so that it facilitates rebalancing of bikes among bike stations in the city. Better balancing of the bikes at the bike stations results to more satisfied users that will select this transportation medium. Consequently,

the environmental footprint of the city will be reduced as users choose more eco-friendly means of transportation.

This paper proposes a crowd-based eco-friendly multi-modal trip planning approach that explores solutions that optimize the route planning process while balancing bikes at the bike stations across the city. This is a challenging problem: First, when designing multi-modal route planning approaches, it is necessary to provide tractable travel recommendations to users. For this case, dynamic information about temporal constraints, such as transit times and departure schedules, need to be seamlessly incorporated, since in real-life, schedules are not always met [10]. The second challenge is to predict the bike demand changes at the stations. Rebalancing can be a costly procedure [11] and therefore it is necessary to have an accurate view for each one of the bike stations of the network. Predicting the bike demand change is attributed to several factors including the location of bike station, day, time, etc. However, as we illustrate extensively in the following sections, the bike demand change does not follow a known mathematical distribution (despite the fact that bike demand itself can be modeled using known statistical models). The bike demand change is an appropriate metric that captures a bike station's tendency for requiring bike redistribution over the day [6]. Therefore, it is necessary to build a per-station bike demand change model that can predict this behavior more accurately, instead of incorporating a single model for every station. Finally, the last challenge is how to design a crowd-based system that despite users' personal preferences will succeed in satisfying multiple objectives such as the better distribution of bikes, users' satisfaction for route planning using multiple means of transportation and reduction of the environmental footprint of the city.

Prior work limitations. Despite some recent works that apply prediction methodologies to estimate the bike trip demands [12], [13], [14], [15], [16], the aforementioned studies are limited as they aim at minimizing only the relocation cost based on the predicted demand without proposing a strategy that could exploit the bike sharing system's users instead of city operator's rebalancing trucks. Moreover, predicting only the bike demand is not adequate, since it fails to capture bike stations' trend for requiring bikes to be dropped-off or picked-up and its respective level concurrently. Furthermore, existing works in the literature still suffer from the cost of redistributing bikes using trucks or rely on price-design mechanisms [17] that require additional fees in order to succeed. Finally, related works in trip planning do not employ dynamic transit information about delays [18] but only make simple assumptions on how to cope with them, in contrast with the multi-modal approach proposed in our work. Therefore, recognizing the unique bike demand trends presented in each bike station and exploit them appropriately to redistribute bikes, while optimizing the route planning process, is the focus of our work.

Contributions. In this work, we propose "MOToR", a fast multi-modal trip planning approach that optimizes both the route planning process and facilitates better balancing of the load of bike stations across the city. Our contributions are summarized as follows:

- We propose a multi-modal trip planning algorithm that incorporates dynamic travel information, captures real-time delays and provides travel recommendations in a time-efficient manner in real-time.
- We propose a prediction scheme, based on random walk method and model the factors that affect bike demand, so as identify the bike demand change at bike stations.
- We implement our solutions on the OTP framework, a widely utilized engine for multi-modal trip planning.
- We evaluate the performance of our proposed scheme using real-world datasets from the city of Warsaw.

The rest of the paper is structured as follows: In section 2, we provide some preliminaries including our System Model, model the factors that affect the bike demand at stations and we define our dual objective problem. In section 3, we extensively describe our approach. In section 4, we present our experimental evaluation. Section 5 describes related work and finally, in section 6, we conclude with lessons learnt from this work.

2. System Model

In this section we first describe our system model, model the factors that affect the bike demand at bike stations and then we define our dual optimization problem.

2.1. Preliminaries

Transportation Systems. Typically, smart cities employ various and different means to server the human need for transportation along the city. Such systems, typically consist of *bike sharing systems*, where users commute from place to place using bikes, *transit systems*, where users utilize tram or metro lines for their transportation needs or other traditional means such as *buses or taxis*. In our work, we assume that citizens utilize multi-modal transportation solutions for their trip planning, combining transit, pedestrian, bike and car segments, for their commutes. Multi-modal transportation systems are typically [9] defined as graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each vertex of $\mathcal{V}$, can be either a bike station or a transit stop and each edge of $\mathcal{E}$, annotates the connection from a place to another using different means of transportation.

Users. A user $u \in \mathcal{U}$ of our system is a citizen that travels along the city. Users typically utilize existing means of transportation such as the transit network or the bike sharing systems to move from one place to another. A user $u \in \mathcal{U}$ of the system can be defined by her age age_u, her personal preferences over the available means of transportation $pref_u$ and her personal tolerance to time delays $thres_u$. The tolerance to time delays of the user expresses

the percentage of the maximum amount of time the user is willing to spend travelling around the city. Moreover, users are typically characterized by their current location, which is defined as the pair of geospatial coordinates (lat_u, lon_u), which annotate their presence in the area of the smart city.

Trip Plan Query. Users $u \in \mathcal{U}$ of our system issue queries q_u in order to receive directions for commuting from a specific geospatial point A to a destination geospatial point B. A trip plan query is characterized by the following tupple:

$$q_u : \langle o^{q_u}, d^{q_u}, t^{q_u} \rangle$$

where o^{q_u} is the origin point (i.e. the geospatial coordinates where the user u begins his trip), d^{q_u} is the destination point (i.e. the geospatial coordinates of the destination of the user) and t^{q_u} is the starting time of the user's u trip.

Route Leg. In trip planning systems, once a user u issues a trip plan query q_u, he receives a set of consecutive path segments to follow until reaching his final destination. We denote such path segments as route legs $tl^j_{Tr_k}$. A route leg $tl^j_{Tr_k}$ is characterized by the following tupple:

$$tl^j_{Tr_k} : \langle Tr_k, o_{tl}, d_{tl}, dur_{o_{tl} \rightarrow d_{tl}}, mode_{tl} \rangle$$

where Tr_k annotates the id of the trip which consists of this route leg, o_{tl} is the origin point of the specific route leg, d_{tl} is the destination point of the specific route leg, $dur_{o_{tl} \rightarrow d_{tl}}$ reflects the amount of travel time required from point o_{tl} to point d_{tl} and $mode_{tl}$ annotates the means used for traversing the specific route leg, such as "WALK", "BICYCLE" or "TRANSIT".

Route. A typical response of a trip planning system is a set of consecutive route legs $tl^j_{Tr_k}$ that the user should follow to reach her final destination as reflected by her issued query q_u. We define this set as a route R_{q_u}. More formally, a route R_{q_u} is defined as a list of consecutive route legs $tl^j_{Tr_k}$ as follows:

$$R_{q_u} : \{tl^1_{Tr_k}, tl^2_{Tr_k}, ..., tl^n_{Tr_k}\}$$

where each $tl^j_{Tr_k}$ annotates the $j-th$ route leg the user should traverse before arriving to her final destination.

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: $\langle i, lat_i, lon_i, cap_i, avail_i, dock_i \rangle$, where i is the unique id of each bike station, lat_i, lon_i represent the geographical coordinates of the station, latitude and longitude, cap_i denotes the bike capacity of the station (*i.e.,* the amount of bike racks of the station), $avail_i$ reflects the current bike availability at the station and $dock_i$ denotes the number of free docks for parking bikes.

Bike Station NetFlow. A key characteristic of bike stations is their demand, expressed either 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 notion of netflow has been used in recent works [6], [19] to reflect how balanced the bike stations are. When the value of net flow at a station is zero this indicates that the entire bike demand, *i.e.,* bikes that arrive and depart, is balanced. We also annotate a bike station that has positive, negative or zero netflow at timeslot t as $b^+_{i,t}$, $b^-_{i,t}$ & $b^0_{i,t}$ respectively.

Netflow Unbalance Score. A key aspect of a bike sharing system's operation is how much unbalanced it is. We need to introduce an appropriate metric that will allow us to successfully capture and illustrate the performance of a bike sharing system. While selecting pairs of bike stations that could be used so that a bike could be picked-up from the first one and be moved to the latter, this selection has a cost, in comparison with another combination of bike stations. We define this cost as the sum of netflow that occurs after selecting the specific station for either picking up from or dropping-off a bike to. More formally,

$$score(b_i) = \begin{cases} (nf^t_{i'} - 1) + \sum_{\forall b^+_i} nf^t_i, i \neq i', \ if \ nf^t_i > 0 \\ (nf^t_{i'} + 1) + \sum_{\forall b^-_i} nf^t_i, i \neq i', \ if \ nf^t_i < 0 \end{cases}$$

Having define the score function, we need to calculate the netflow unbalance score $nus(b^+_i, b^-_{i'})$ that captures how much the system is unbalanced when the specific combination of bike stations are chosen. More formally,

$$nus(b^+_i, b^-_{i'}) = |score(b^+_i)| + |score(b^-_{i'})|$$

2.2. Modeling Bike Station Demand

Our work is motivated by the fact that there are several factors that may affect the bike usage across bike stations during the day [6], [15].

In the following, we aim at identifying the key features that affect the bike usage focusing on Warsaw City, where we are witnessing a growing interest in utilizing bikes for daily commuting. We have implemented MOToR and deployed it in the VaVeL system[1] to provide route recommendations in the City of Warsaw. We graphically illustrate the average number of bike trips per station conducted by users on both weekdays and weekends. As we observe from Figure 1, users resort to using bikes for their commuting during the morning and the afternoon rush hours (8 to 10am and 4-6pm). However, we observe that the number of trips per station is higher during the afternoon rush hours. We reason this to the fact that users prefer bicycles instead of other traditional means as an alternative, healthier and recreational way to commute back to their personal space. Additionally, during weekends, we observe from Figure 2

1. http://www.vavel-project.eu/

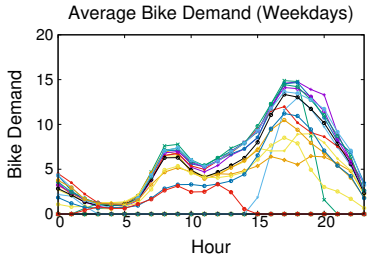


Figure 1. Hourly Bike Demand on weekdays

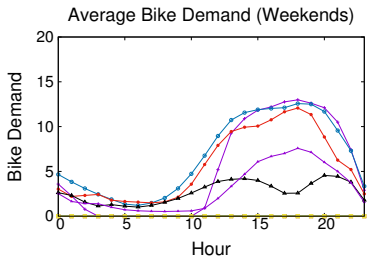
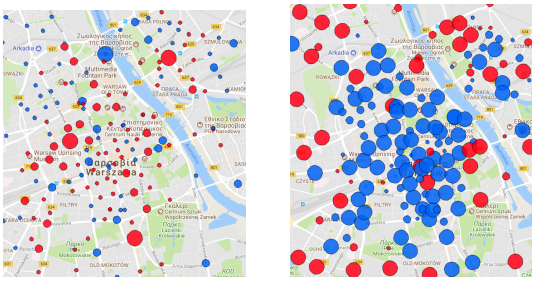


Figure 2. Hourly Bike Demand on weekends



(a) 8:00-10:00am (b) 16:00-18:00

Figure 3. Bike demand at bike stations on morning and afternoon rush hours

that people resort to bikes for their activities for a greater amount of time (higher bike demand during 12pm to 8pm). Therefore, we may conclude that the type of day (weekday or weekend) plays a significant role when trying to model bike demand. Finally, in Figures 3a & 3b, we draw the bike station netflow for station around the Warsaw City center. As we observe, stations around the city center present higher outgoing bike demand instead of ingoing bike demand during the afternoon. This is an expected phenomenon, known as complementarity effect [6], but quite important to understand how citizens behave while using bikes during the day. For the case of Warsaw city, we may conclude that users are highly interested on using bikes for leaving the city center during the afternoon.

2.3. Problem Definition

Hereby, we present our dual optimization problem. Our goal is twofold: maximize the number of bike stations that are balanced and optimize the route planning process by incorporating dynamic real-time information about schedule delays. We aim at incorporating the route planning process in concert with a rebalancing procedure in order to support certain demand levels at bike stations across the city. Having predicted the change of bike demand at bike stations near the origin and destination points, we aim at selecting the best route so that the number of bike stations that are balanced is maximized and the trip time duration is minimized by the incorporation of the dynamic transit schedule data.

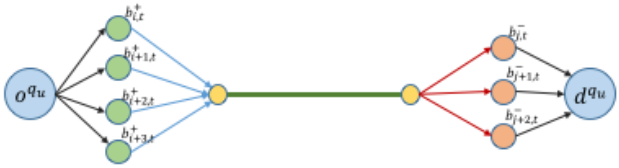


Figure 4. Problem overview

An overview of our optimization problem is illustrated in Figure 4. We assume a set of bike stations $b_i \in \mathcal{B}$. We also assume a route R_{q_u} generated from the trip planning system from an issued user query q_u. We identify the bike stations which are near the origin and destination points of the route R_{q_u} and form a set $\mathcal{B}_{check}$. Utilizing the origin and destination point of the user query, o^{q_u}, d^{q_u} respectively, the bike stations $b_i \in \mathcal{B}_{check}$ and the start and destination points of the route leg $tl^j_{Tr_k}$ of route R_{q_u} where $mode_{tl} = "TRANSIT"$, we form a weighted directed graph $G' = (V', E')$, where each vertex $v_w \in V'$ annotates the respective geospatial point and each edge $e_{wz} \in E'$ annotates the path from vertex v_w to vertex v_z. Moreover, each edge e_{wz} has a respective weight f_{wz} which is defined as the time required to traverse edge e_{wz}. Furthermore, since we aim at maximizing the number of stations that are balanced (having netflow equal to zero), our goal is to minimize the netflow unbalance score, defined in section 2.1. Therefore, our dual optimization problem can be expressed formally as follows:

$$\max \quad F(x) = -\sum_{i}\sum_{i'}(x_{ii'} * nus(b_i^+, b_{i'}^-)), \tag{1}$$

$$\min \quad G(x) = \sum_{w}\sum_{z}(x_{wz} * f_{wz}) \tag{2}$$

$$x_{ii'} \in \{0,1\}, \forall i \neq i' \tag{3}$$

$$x_{wz} \in \{0,1\}, \forall w \neq z \tag{4}$$

where F(x) in Equation 1 annotates the netflow unbalance optimization objective, which we need to minimize, whereas G(x) in Equation 2 annotates the travel time duration which we need to minimize as well. Equation 3 denotes whether we should select the specific pair of bike stations or not, so that the travel time duration is minimized and the netflow unbalance score is also minimized. Finally, Equation 4 denotes whether we should select the specific pair of nodes or not so that the trip time is minimized.

3. Our approach

In this section, we describe our approach towards solving our dual optimization problem. First, we introduce the necessary methods for predicting the change of demand at bike stations. Then, we proceed on describing how our route planning process incorporates dynamic transit data so as to generate the shortest route for a given user query. Finally, we present how these two different components are used in

concert for implementing the "MOToR" algorithm and we give a brief analysis of the algorithm's complexity.

3.1. Bike Demand Change Prediction

Our primary goal is to estimate which of the bike stations near the origin and destination endpoint of the user query require to be balanced. A few recent techniques have been proposed in the bibliography for predicting the demand at bike stations using non-parametric statistical regression techniques [20], random forest techniques on mobility models per station [21] and multi-similarity weighted KNN approaches [15] for the prediction process. However, these approaches are limited, since, our goal is to select bikes from stations that have positive netflow (users drop-off bike with higher rate than picking up) to be moved to stations with negative netflow (where users pick-up bikes with higher rate than dropping-off). These methods fail to capture this change on bike stations and cannot fit in our setting. For this purpose, we resort to the random-walk method [22], which is a forecasting technique that captures the change of a quantity based on its value on a previous time period.

In our approach, we use a modified random-walk method to estimate the bike demand change on each bike station, given historical data of bike demand, for a given timewindow. Our goal is to estimate the bike station's netflow and identify the stations that are unbalanced and for which, it is required to either pick-up bike from or drop-off bikes to.

Modeling NetFlow Change. A necessary step for the prediction process is to model the netflow behavior. During our experimental analysis, we observed that bike stations present different patterns of netflow changes on a per-hour basis. Therefore, we argue that it is not possible to incorporate a known statistical distribution that could successfully capture the bike station netflow change for all stations. In order to predict the bike station netflow change, we propose a generic linear model that helps capture this change. More specifically, the netflow nf_i^t of a bike station at timeslot t can be computed as the sum of the netflow of the previous timeslot nf_i^{t-1} with an estimated bike change $c^*_{t,dw}$ for the timeslot t and the type of day dw from the historical data. More formally:

$$nf_i^t = nf_i^{t-1} + c^*_{t,dw}$$

Demand Change Prediction. As aforementioned, we utilize a modified random-walk method to capture this change of demand, which is illustrated by the netflow metric. The random-walk-with-drift method we utilize, requires the estimation of the parameter $c^*_{t,dw}$, which is the step size.

Therefore, the first step of the prediction process, is to estimate the value $c^*_{t,dw}$, given the historical data, for the timeslot t. To do so, we generate the average value of $c^*_{t,dw}$ utilizing the observed changes of netflow at the bike station for the timeslots t and $t-1$ in the historical data and for the same type of day dw. Since, this prediction process is conducted on a per-bike-station basis, it helps capture the unique behavior of the bike station's netflow and therefore, it can provide more accurate results of the bike demand change.

The final step of the prediction process is to estimate the bike station's netflow at timeslot t. The traditional random-walk method requires *tossing a coin* in order to select whether, we should move up or down. However, in our case, we loose this constraint, since the sign of the estimated bike demand change encapsulates this behavior(positive-going up and negative-going down). Given the actual bike station netflow at the previous timeslot $t-1$, the value of nf_i^t is computed as the sum of nf_i^{t-1} and the estimated bike netflow change from the previous step $c^*_{t,dw}$ for the required timeslot t and the specific type of day dw.

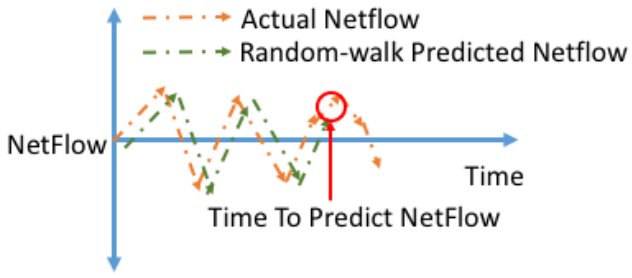


Figure 5. Modified Random-walk Method

In Figure 5, we illustrate how the prediction method is conducted at each bike station. Given estimated step size $c^*_{t,dw}$ from the historical data and the previous netflow value nf_i^{t-1}, the modified random-walk method estimates the netflow nf_i^t at timeslot t.

3.2. Enhanced Dynamic Travel Planning

During the travel planning our trip computation method searches the shortest path from the origin o^{q_u} to the destination d^{q_u} using the standard cost metrics included in OTP. OTP supports multi-modal trip computation, and usage of different trip computation algorithms on the travel modes. Initially, the origin and destination locations are mapped to the traffic network (a process also known as *MapMatching*), with the resulting vertices a shortest path search along the traffic network is performed as follows. Suppose we have a graph $G = (V, E)$, with $E \subseteq V \times V$ and a cost function mapping $c : E \mapsto \mathbb{R}^+$. A path of length n from a vertex $v_0 \in V$ to a vertex $v_n \in V$ in this network is an alternating sequence of vertices and edges $P(v_0, v_n) = (v_0, e_0, \ldots, v_n)$, with $\forall v_s : v_s \in V$ and $\forall e_g = (v_g, v_{g+1}) : (v_g, v_{g+1}) \in E$. The cost of such a path is the sum of the edgewise costs $c(P(x_0, x_n)) = \sum_{e_g \in P} c(e_g)$. Trip computation from v_0 to v_n is the search for the path $P^*(v_0, v_n)$ with the minimal costs:

$$P^*(v_0, v_n) = \underset{P(v_0,v_n)}{\arg\min}\, c(P(v_0, v_n)) \; .$$

Standard solution to the problem is using Dijkstra's algorithm [23]. Given the graph $G = (V, E)$ and $v_0, v_n \in$

TABLE 1. TRANSFER PATTERN EXAMPLE

line L17	S154	S097	S987	S111	···
trip 1	8:15	8:22 8:23	8:27 8:29	8:38 8:39	···
trip 2	9:14	9:21 9:22	9:28 9:28	9:37 9:38	···

TABLE 2. TRANSFER PATTERN EXAMPLE (CONTINUED)

S097:	(L8,4)	(L17,2)	(L34,5)	(L87,17)	···
S111:	(L9,4)	(L13,5)	(L17,4)	(L55,16)	···

V, it initializes a queue of nodes $Q = V$ and a distance function over $V \times V$ with $dist(v_0, v_0) = 0$ and $dist(v_0, v_n) = \infty, \forall v \neq v_0, v \in V$. Until the queue is empty the node v_w with the smallest distance $dist(v_0, v_w)$ is picked and removed from Q. For each neighboring node of v_w the distance is updated as follows: $dist(v_0, v_n) := dist(v_0, v_w) + c(P(v_w, v_n))$, if the latter is smaller than the former. Dijkstra's algorithm can be sped up by running it simultaneously from both v_0 and v_n until a common node v_w is hit. In OpenTripPlanner [4], street based trip computation is performed by a A* algorithm, a possible speedup could be usage of Contraction Hierarchies [24]. Public transport routing poses additional requirements to the trip computation as trip availability becomes time-dependent and also waiting times at a particular transfer stop must be incorporated in the algorithm. Besides the time-extended or time-depending A*, Transfer Patterns [18] provide a state-of-the-art method to speed trip computation in public transport networks up. However, transfer pattern do not necessarily provide correct answers in the likely case of delays of the transit vehicles. Thus, we utilize the Dynamic Transfer Pattern proposed in [10]. The trip planning requires a preprocessing of the transit schedules as follows. For each public transport line a table is stored denoting in the columns the stops along the line. In this way it holds the maximal possible route without changes. The rows of the table represent the actual trips of the line and Table 1 provides an example.

In addition, for every station a list is stored with the passing lines and their position in the trips, see an example in Table 2.

With Transfer Pattern a route from S_{start} to S_{stop} at time t is calculated by the intersection of the lists for S_{start} and S_{stop}, the first connection after time t is the desired result. As an example, the route from S097 to S111 at 9:03 is computed by intersecting the lines in Table 2, we find that line 17 connects the stops on positions 2 and 4. Earliest possible trip is, according to Table 1 departing 9:22 at S097 and arrives at S111 at 9:37.

Besides this temporal information, Transfer Pattern also store the graph structure of the transit network (neglecting temporal information). In the pre-processing phase the shortest paths amongst all pairs of stops are constructed, and intermediate stops are stored in a directed acyclic graph (DAG). The approach in [10] incorporates potential delay information already in the pre-computation phase, and adds

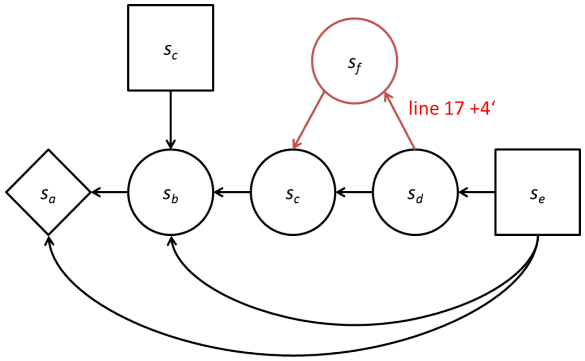


Figure 6. DAG structure of the dynamic transfer pattern.

additional transfer possibilities to the DAGs created during transfer pattern creation, see Figure 6.

In case of dynamic schedule information, the time table information (Table 1 in our example) can be updated and the route computation could be performed as in the regular case [18]. However, it might be that the optimal transit connection differs as new connections could be enabled by the delay itself, especially if multiple lines arrive belated incorporation of the delays in trip computation provides useful benefit to travellers.

3.3. MOToR Algorithm

Hereby, we present "MOToR" strategy towards balancing the bikes of the bike station network comprising the bike demand change prediction and the enhanced dynamic travel planning in concert.

Given a user query q_u, the enhanced dynamic travel module of the OTP instance generates an initial route R_{q_u}, which incorporates real-time delays for the travel time cost calculation. This is performed by the method $GenerateInitialTransitRouteFromOTP()$. The generated route consists of a route leg $tl^j_{Tr_k}$ where $mode_{tl} =$ "$TRANSIT$", which fulfills the shortest in time criterion of the user using transit as the transportation medium. Then, the route R_{q_u} is forwarded to the bike station prediction module. At this step, we identify all the bike stations near the endpoints of the transit route leg $tl^j_{Tr_k}$ using the method $NearByBikeStations()$ for both the first and the last transit stop. For all these stations, we predict the netflow change in the next future timeslot using the method $PredictNetFlows()$. This step helps identify the stations near the origin endpoint of the tram route that a bike could possibly be picked-up and which stations near the destination endpoint of the tram route require more bikes to be dropped-off. Given the origin and destination points of the user query, this module generates all the possible combinations of routes that could be proposed to the user utilizing both bike and transit network using the method $GeneratePossibleRoutes()$.

The final step of the "MOToR" algorithm is to select the appropriate route that fulfills certain time criteria, but

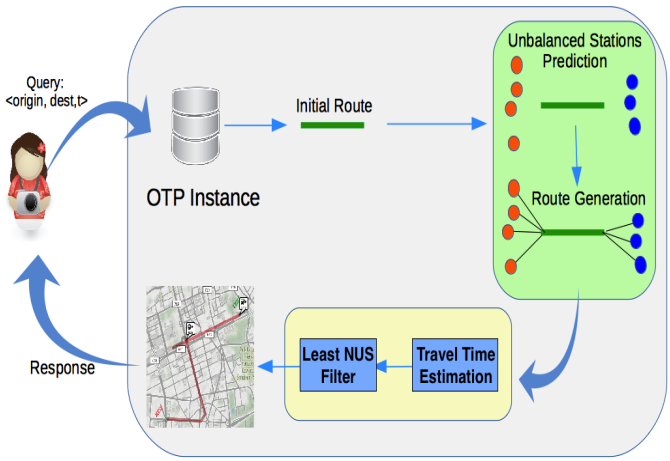


Figure 7. Flow of our approach

Input: $User\ Query\ q_u$
Output: Shortest route satisfying constraints
1: $R_{init} = GenerateInitialTransitRouteFromOTP(q_u)$;
2: $L_1 = NearByBikeStations(FirstTransitStop, R_{init})$;
3: $L_2 = NearByBikeStations(LastTransitStop, R_{init})$;
4: $[B_+, B_-] = PredictNetFlows(L_1, L_2)$;
5: $PR = GeneratePossibleRoutes(B_+, B_-)$;
6: $CR = RunShortestPath(PR, thres_u)$;
7: **for** (cr : CR) **do**
8: $nus = findNUS(cr)$;
9: **if** (nus $\leq$ optimal) **then**
10: $optimal = nus$;
11: $bestRoute = cr$;
12: **end if**
13: **end for**
14: **return** $bestRoute$

Figure 8. MOToR algorithm

also helps rebalance the bike station network. At this point, we first need to identify the shortest in time route, which can be proposed to the user. We utilize a very well known technique [23] to identify the shortest path from the origin to the destination endpoint of the user query using the method $RunShortestPath()$. Given the user's tolerance to travel duration time, we also acquire the set of paths that are within the given duration time constraints. Given the second part of our optimization problem, we need to optimize the bike sharing system's netflow unbalance. For this purpose, we utilize the Netflow Unbalance Score in order to filter the candidate routes and select the respective one that achieves the least Netflow Unbalance Score. Finally, the route that optimizes the route planning process, as illustrated in the respective travel time duration and has the least Netflow Unbalance Score is returned to the user. A graphical illustration of our approach can be seen in Figure 7 and the steps of our algorithm are summarized in Figure 8.

Worst-case Complexity. The algorithmic complexity for deriving the route using the enhanced dynamic travel planning component is equal to the complexity of the A^*-algorithm ($\mathcal{O}(b^d)$). Let us assume m bike stations with positive netflow and n bike stations with negative netflow. We form a graph $G' = (V', E')$, using the origin and destination endpoints of the user query, the bike stations and the transit stops, as illustrated in Figure 4. Running the Dijkstra algorithm over the graph G' has an algorithmic complexity of $\mathcal{O}((|V'|+|E'|)*log|V'|)$. Given that we run this for k iterations so as to identify the sets of other possible routes that are within the acceptable user threshold, the complexity is $\mathcal{O}(k*(|V'|+|E'|)*log|V'|)$. Finally, since we have to check the set of the generated routes for optimizing the systems unbalance, we need to check k routes, therefore the final complexity is $\mathcal{O}(k*(|V'|+|E'|)*log|V'|)+\mathcal{O}(b^d)$.

4. Experimental 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 using datasets from real systems. Our goal was to identify the advantages of our approach with respect to the following metrics: *a) prediction performance*, *b) rebalancing performance* and *c) trip time duration optimization*.

4.1. Experimental Setup

Datasets. We use two heterogeneous sources of real-world data from the city of Warsaw in order to evaluate our proposed algorithm. The datasets provide information regarding the bike availability of bike stations and the possible routes between different pairs of source and destination endpoints using only transit network.

Bike availability dataset. We utilized a real bike availability dataset from the city of Warsaw. The available bike data expand from August 6th to August 31st. The dataset provides information about, the bike station id, the geospatial coordinates of the station, latitude and longitude, the number of bikes available to pick up, the number of docks available to park a bike, the total number of bike racks, which denotes the capacity of the bike station, and, finally, a list of bike ids for bikes that are available for pick-up. The data where collected using a two minute sampling frequency, which has been used [25] to better capture the activity of bike stations. Then, the data were preprocessed so as to find the necessary bike demand values on a timeslot of 1-hour length.

Routes dataset. In order to evaluate the trip time duration performance, we synthetically generate a dataset of origin and destination pairs for the city of Warsaw. Given eight different types of routes, (i.e., crossing the city center (for variable route length), going from suburban areas to city center, from suburban areas to other suburban areas e.t.c), we generated a set of perturbated origin and destination endpoints for each one of these eight different types. Each pair of the set was used to generate the appropriate routes

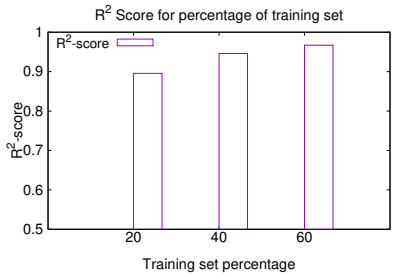


Figure 9. $R^2 - score$

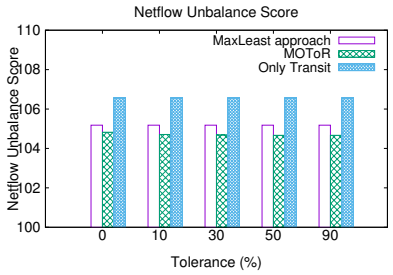


Figure 10. Netflow Unbalance Score Reduction

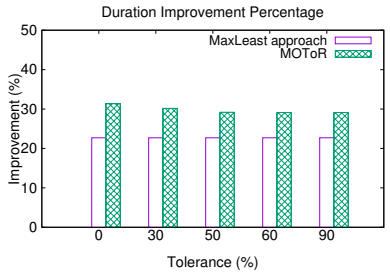


Figure 11. Travel time duration improvement

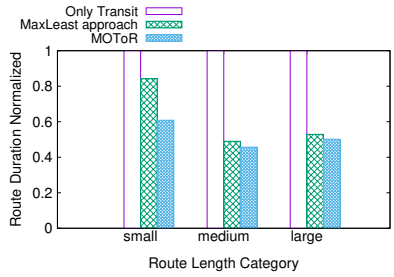


Figure 12. Duration Improvement: 0% Tolerance

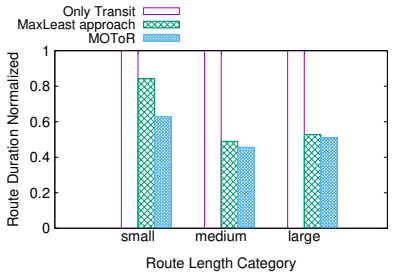


Figure 13. Duration Improvement: 30% Tolerance

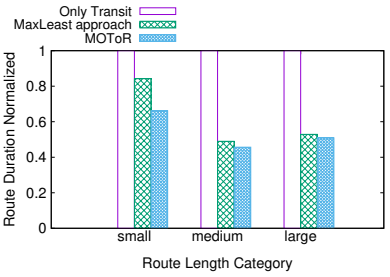


Figure 14. Duration Improvement: 60% Tolerance

using the transit network. To achieve that, we have set up an OpenTripPlanner instance using real GTFS data from the city of Warsaw, upon which we queried and derived the necessary route information. The route response from OTP contains information about the start and destination endpoints of the user query, the geospatial coordinates of the transit stops along the route and the time required for each route leg (walking from origin, transit, and walking to destination).

4.2. Evaluation

Evaluation Scenario. We have set up the following scenario. For each route in the routes dataset, we identify and construct a set of stations that are within a walking distance of 500m (5 minutes) from the first and the last tram stop respectively. We aim at optimizing bike stations very close to the transit stops. Then, for each bike station in the sets, we predict its bike demand change, and identify whether this station is appropriate to pick-up a bike from or requires bikes to be dropped-off. After having predicted the bike demand change for all the bike stations in the sets, we keep only those that have a bike station netflow value greater or less than zero. Given the sets of stations, we use a path generator in order to get the routes to check, which contain cycling as part of the trip. Finally, for the constructed set, we identify those that satisfy both objectives, least trip duration time and less unbalance of the system. In order to evaluate the performance of "MOToR", we have developed a set of baseline approaches. The first baseline approach we developed was the *MaxLeast* approach. This method returns the route that bypasses a bike station with the maximum value of positive netflow near the first tram stop of the route and bypasses a bike station with the least value of negative netflow near the last tram stop of the route. The second baseline approach we evaluated "MOToR" against, is an *Only Transit* approach, in which the returned route to the user contains only "transit" as the transportation medium.

Prediction Performance. We evaluated the performance of "MOToR" prediction method for each one of the routes in our scenario. We utilized the R^2 metric [26], which is a common metric for evaluating the performance of a prediction method, given the size of the training dataset. This metric provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model. Values near to one are considered to be better. As we observe from Figure 9, "MOToR" prediction method achieves very good performance even for low percentage of the training set. We conclude that the proposed linear model based on a modified version of the random-walk method is appropriate for predicting the bike demand change, since it succeeds in capturing the unique trends of bike demand change of each bike station.

Rebalancing performance. Next, we evaluated the performance of "MOToR" regarding its efficiency to improve the netflow unbalance score at bike stations near the first and the last tram stop of each route. We conclude from Figure 10 that as the percentage of user tolerance increases, "MOToR" outperforms its competitors. We also observe that the average netflow unbalance score decreases as the tolerance to delays increases. This is an expected outcome, since "MOToR" considers more candidate routes, and therefore favours routes with lower netflow unbalance score. Finally,

we observe that our approach is beneficial for urban areas with high number of bike stations available nearby the transit stops, since we optimize the "first" and "last" mile of the user's route while at the same time balancing the bikes at nearby stations (for distances up to 500m from the transit stops). However, this distance is a tunable parameter and if we increase the distance we expect that even more stations will be balanced.

Travel time duration optimization. Finally, we evaluate the performance of "MOToR" regarding its efficiency to minimize the travel time routes. As we observe from Figure 11, "MOToR" outperforms the "MaxLeast" and "Only Transit" approaches and can achieve improvement of at least 30%, even for users that do not tolerate any additional trip time at all. We conclude that "MOToR" succeeds in identifying routes that minimize the travel time duration, instead of using only transit. Additionally, in Figures 12,13 & 14, we draw the normalized average travel time duration for the three route length categories of our dataset (small, medium and large) and for a given user tolerance of 0%, 30% and 60% respectively. We observe that "MOToR" outperforms its competitors, even for the highest tolerance. For instance, for 30% tolerance, "MOToR" derives routes with values equal to 331sec, 855sec and 915sec for each category, whereas "MaxLeast" derives 443sec, 917sec and 950sec for the same categories. This result is achieved as we take care of the "first" and "last" mile problem of the user's trip. Finally, it is clear that "MOToR" succeeds in satisfying both objectives of route planning and bike rebalancing.

5. Related Work

Bike Rebalancing. Existing state-of-the-art methods focus either on *prediction methods* [13], [14], [15], [16], [20], in which, authors propose systems that do not consider user route planning as a means of rebalancing bikes and *vehicular optimization methods* [27], [28], in which algorithms solve the relocation from other aspects such as optimal station-to-station route design and inventory management. In [13] authors model the parameters that affect the bike demand during the day and predict it for a given future timewindow. However, their work is limited since, they do not focus on how to handle unbalanced stations, as we do in our work. In [14], authors mainly focus on identifying over-demand stations during large events. However, their approach is limited since their rebalancing procedure relies solely on vehicular methods, whereas, in our approach, we optimize both the route planning process, but also rebalance the system, without application of vehicular strategies. In our previous works [6], [19], we illustrated the benefits of bike rebalancing during large scale events and have illustrated how human mobility can be extracted using bike sharing systems. However, in this work, we focus on a totally different approach, where the objective is to generate trip recommendations combining route planning techniques with bike rebalancing.

Route planning. There has been extensive work in the bibliography regarding the route planning problem [7], [8], [9]. However, the proposed methods are limited since they solely focus on optimizing the travel time duration parameter, without focusing on optimizing the performance of another means of transportation, such as bike sharing systems, as we do in our work. In [7], authors solely focus on identifying the trip that satisfies user time constraints based on their location. In contrast, in our work, our goal is not only to find the optimal trip, but also to optimize the bike sharing system of the city as well. Authors of [9] focus on identifying overcrowded transit stop stations, and propose an unobstructed route planning strategy. However, they solely focus on optimizing the travel time duration, without improving the performance of other means of transportation concurrently, as we do in our work.

Dynamic Transfer Planning. In a realistic route planning scenario, various delays occur among the public transport vehicles. In contrast to vehicular traffic, trams and trains can not overtake, and vehicles in transit networks wait for each others (e.g. connecting trains), this causes delays to propagate differently than vehicular traffic jams. In addition, two modes of transportation may share the same physical resource (e.g. buses or trams riding on vehicular street). Thus, two forms of delays in transit networks are distinguished in literature: 1) a vehicle is late due to own reasons, and 2) other vehicles are late caused by the former [29]. Several models for transit delays are reported in literature. The work in [30] assumes independence. In contrast, [31] allows delays to cumulate. Sophisticated models incorporate dependencies among the vehicles into the delay [32]. In [33] the delays are analyzed visually. In a trip planning application real-time predictions of delays are a main benefit as future delays may influence the route choice. Thus, we highlight few recent works on delay prediction and delay recognition: [34] applies queueing theory and assumes delays to aggregate, [35] detects delays and unexpected vehicle movement in real-time from the GPS traces, and [36] predicts delays per line based on its previous delay and the delay information of earlier trips at surrounding locations.

6. Conclusions

In this paper, we presented "MOToR", a novel eco-friendly framework, that helps rebalance a bike station network and fulfill user needs for reliable and efficient travel time route planning despite any observed real-time delays. Our contributions are summarized as follows:

- We proposed a multi-modal trip planning algorithm that incorporates dynamic travel information, captures real-time delays and provides travel recommendations in a time-efficient manner in real-time.
- We proposed a prediction scheme, based on random walk method and modeled the factors that affect bike demand, so as identify the bike demand change at bike stations.
- We implemented our solutions on the OTP framework, a widely utilized engine for multi-modal trip planning.

- We evaluated the performance of our proposed scheme using real-world datasets from the city of Warsaw.
- We illustrated that our approach is beneficial for citizens in urban areas, since they can improve their trip times taking advantage of the large number of bike stations located nearby transit stops rather than walking.

References

[1] "Crowdalert application," http://crowdalert.aueb.gr.

[2] "Waze," https://www.waze.com/.

[3] I. Boutsis, D. Tomaras, and V. Kalogeraki, "Towards real-time emergency response using crowdsourcing," in *PETRA*, Rhodes, Greece, May 2014.

[4] "Opentripplanner," http://www.opentripplanner.org/.

[5] "Google maps," http://maps.google.com.

[6] D. Tomaras, I. Boutsis, and V. Kalogeraki, "Lessons learnt from the analysis of a bike sharing system," in *PETRA*. ACM, 2017, pp. 261–264.

[7] E. H.-C. Lu, C.-Y. Lin, and V. S. Tseng, "Trip-mine: An efficient trip planning approach with travel time constraints," in *Mobile Data Management (MDM), 2011 12th IEEE International Conference on*, vol. 1. IEEE, 2011, pp. 152–161.

[8] O. Vedernikov, L. Kulik, and K. Ramamohanarao, "The hitchhiker's guide to the optimal route planning," in *Mobile Data Management (MDM), 2017 18th IEEE International Conference on*. IEEE, 2017, pp. 234–239.

[9] S. Shang, D. Guo, J. Liu, and K. Liu, "Human mobility prediction and unobstructed route planning in public transport networks," in *Mobile Data Management (MDM), 2014 IEEE 15th International Conference on*, vol. 2. IEEE, 2014, pp. 43–48.

[10] T. Liebig, S. Peter, M. Grzenda, and K. Junosza-Szaniawski, "Dynamic transfer patterns for fast multi-modal route planning," in *International Conference on Geographic Information Science*. Springer, 2017, pp. 223–236.

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

[12] 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.

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

[14] 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.

[15] 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.

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

[17] A. Singla, M. Santoni, G. Bartók, P. Mukerji, M. Meenen, and A. Krause, "Incentivizing users for balancing bike sharing systems." in *AAAI 2015*, Austin, Texas, USA, 2015, pp. 723–729.

[18] H. Bast, J. Sternisko, and S. Storandt, "Delay-robustness of transfer patterns in public transportation route planning," in *ATMOS-13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems-2013*, vol. 33. Schloss DagstuhlLeibniz-Zentrum fuer Informatik, 2013, pp. 42–54.

[19] D. Tomaras, I. Boutsis, and V. Kalogeraki, "Modeling and predicting bike demand in large city situations," in *PERCOM*. Athens, Greece: IEEE, 2018.

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

[21] 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.

[22] W. Estes, "A random walk model for choice behavior," *Mathematical methods in the social sciences*, vol. 1960, 1959.

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

[24] R. Geisberger, P. Sanders, D. Schultes, and C. Vetter, "Exact routing in large road networks using contraction hierarchies," *Transportation Science*, vol. 46, no. 3, pp. 388–404, 2012.

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

[26] J. Strauss and L. Miranda-Moreno, "Spatial modeling of bicycle activity at signalized intersections jillian strauss, luis f miranda-m," 2013.

[27] 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.

[28] 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.

[29] M. Müller-Hannemann and M. Schnee, "Efficient timetable information in the presence of delays," in *Robust and Online Large-Scale Optimization*. Springer, 2009, pp. 249–272.

[30] J. Dibbelt, T. Pajor, B. Strasser, and D. Wagner, "Intriguingly simple and fast transit routing," in *International Symposium on Experimental Algorithms*. Springer, 2013, pp. 43–54.

[31] M. Goerigk, M. Knoth, M. Müller-Hannemann, M. Schmidt, and A. Schöbel, "The price of robustness in timetable information," in *OASIcs-OpenAccess Series in Informatics*, vol. 20. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2011.

[32] A. Higgins and E. Kozan, "Modeling train delays in urban networks," *Transportation Science*, vol. 32, no. 4, pp. 346–357, 1998.

[33] J. D. Mazimpaka and S. Timpf, "A visual and computational analysis approach for exploring significant locations and time periods along a bus route," in *Proceedings of the 9th ACM SIGSPATIAL International Workshop on Computational Transportation Science*. ACM, 2016, pp. 43–48.

[34] A. Gal, A. Mandelbaum, F. Schnitzler, A. Senderovich, and M. Weidlich, "Traveling time prediction in scheduled transportation with journey segments," *Information Systems*, 2015.

[35] N. Zygouras, N. Zacheilas, V. Kalogeraki, D. Kinane, and D. Gunopulos, "Insights on a scalable and dynamic traffic management system." in *EDBT*, 2015, pp. 653–664.

[36] L. Heppe and T. Liebig, *Real-Time Public Transport Delay Prediction for Situation-Aware Routing*. Cham: Springer International Publishing, 2017, pp. 128–141. [Online]. Available: https://doi.org/10.1007/978-3-319-67190-1_10