Dynamic Vehicle Redistribution and Online Price Incentives in Shared Mobility Systems

Julius Pfrommer, Joseph Warrington, *Member, IEEE*, Georg Schildbach, *Member, IEEE*, and Manfred Morari, *Fellow, IEEE*

***Abstract*—This paper considers the efficient operation of shared mobility systems via the combination of intelligent routing decisions for staff-based vehicle redistribution and real-time price incentives for customers. The approach is applied to London's Barclays Cycle Hire scheme, which the authors have simulated based on historical data. Using model-based predictive control principles, dynamically varying rewards are computed and offered to customers carrying out journeys, based on the current and predicted state of the system. The aim is to encourage them to park bicycles at nearby underused stations, thereby reducing the expected cost of redistributing them using dedicated staff. In parallel, routing directions for redistribution staff are periodically recomputed using a model-based heuristic. It is shown that it is possible to trade off reward payouts to customers against the cost of hiring staff to redistribute bicycles, in order to minimize operating costs for a given desired service level.**

***Index Terms*—Bicycle sharing, dynamic pricing, dynamic vehicle routing, model predictive control (MPC), shared mobility systems.**

I. Introduction

PUBLIC bicycle sharing (PBS) schemes offer the rental of bicycles as a means of public transportation in urban areas. They allow registered users to pick up a bicycle from one of many docking stations throughout the entire city, without any prior notice. The bicycle is returned to another station, after which the user's intended destination is usually reached on foot. Short journeys are encouraged by charging users only a small fee for a short rental period (typically less than 1 h), but then ramping up the cost significantly for longer use.

Such schemes have been introduced in major cities as an alternative to often slow and crowded mass transportation. Many have considerably grown in size in recent years [1], [2] and are becoming a major component of their cities' public transportation systems. In 2008, for example, 120 000 daily journeys were being made using shared bicycles in Paris [3].

Most PBS schemes are still unable to recoup their full operational and investment costs solely from customer fees. According to [4], capital costs can be up to \$4500 per bicycle, and annual operational costs can be up to \$1700 per bicycle. Sometimes, additional revenues from advertising can be used to mitigate this cost gap. However, in almost all cases, additional funding from public sources is required [1], [5].

One of the major contributors to operational costs is the need to operate staffed trucks for manual relocation of bicycles, in order to balance the difference between supply and demand at various stations. If this effort were not made, the arrival and departure of customers would cause many stations to run full or empty, and the customer service rate would drop below acceptable levels [6], [7].

The goal of this paper is to explore a novel method for improving the service level while lowering the need for manual relocations and, hence, to save on operational costs. The method consists of two components. First, we devise an algorithm to optimize the dynamic route planning of multiple trucks for bicycle relocation. Second, on top of this manual redistribution, we propose a scheme that offers users price incentives based on the current and predicted state of the system, in order to encourage them to change the endpoint of their journeys. These incentives are set to shift bicycle drop-offs away from stations that are overfilled and toward nearby stations that may have empty spaces. Both components are combined into an effective strategy for operating the PBS, in which the truck routes and the price incentives are recomputed online at periodic time instances.

This paper's main contributions can be summarized as follows.

1) A tailored routing algorithm that plans how trucks will be used to redistribute bicycles among stations. Redistribution is performed in the dynamic setting, i.e., while the system is in operation. The heuristic chooses the actions of multiple trucks, with the aim of enabling as many extra journeys as possible to take place.
2) A dynamic incentives scheme where customers are encouraged to change their target station in exchange for a payment. Changes to journey length may be inconvenient, and we assume customers accept or reject such incentives based on the value of their time and the payment offered.

For both components, a predictive model of the expected near-future evolution of the system state is used to optimize their actions over a finite receding horizon. The optimization goal is to maximize the number of additional journeys enabled via redistribution, taking into account available resources and cost tradeoffs. At each reoptimization step, up-to-date information on the current state of the system is used to plan all future operational decisions. This is schematically shown in Fig. 1.

Manuscript received June 5, 2013; revised October 1, 2013; accepted January 9, 2014. Date of publication March 3, 2014; date of current version August 1, 2014. The Associate Editor for this paper was E. Kosmatopoulos.

J. Pfrommer is with the Vision and Fusion Laboratory (IES), Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany.

J. Warrington, G. Schildbach, and M. Morari are with the Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH) Zürich, 8092 Zürich, Switzerland.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2014.2303986

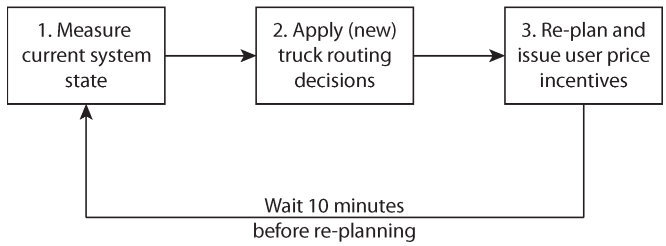

Fig. 1. Schematic of the online optimization scheme presented in this paper. In step 2, truck routes are planned based on a model of how the bicycle movements will evolve from the current state, and if necessary, new orders are issued. In step 3, the current system state and the future truck actions are taken into account, and new price incentives for users are computed, based on a tradeoff between payouts and system performance.

TABLE I
RIDE INFORMATION SAMPLES

bicycle-id	start (date, station-id)	end (date, station-id)
3340	{2010-07-30 06:00:00, 47}	{2010-07-30 06:22:00, 47}
3870	{2010-07-30 06:00:00, 234}	{2010-07-30 06:14:00, 203}
1627	{2010-07-30 06:01:00, 149}	{2010-07-30 06:29:00, 293}
1695	{2010-07-30 06:02:00, 152}	{2010-07-30 06:06:00, 324}

We evaluate the tradeoff between these two methods using a Monte Carlo model of the London Cycle Hire PBS scheme, which is constructed from detailed historical usage information. Our results suggest that service level improvements may be attained using price incentives alone, and that increases in either the customer payouts or the number of redistribution trucks deliver diminishing returns to service levels. Unsurprisingly, higher service levels can be reached on weekends in comparison to weekdays.

This paper is organized as follows: In Section II, we explain how a model of the London PBS scheme was derived using historical data. In Section III, we develop a metric for the utility of redistribution actions based on the expected ability to serve additional future customers. The results are used in Section IV to develop a heuristic for determining the routes of redistribution trucks. In Section V, we develop a model-based controller for computing the price incentives offered to customers. The two approaches for redistribution are compared using a Monte Carlo simulation framework in Section VI. Section VII draws conclusions on the performance of the scheme.

II. SYSTEM MODEL

A. Historical Data

The PBS system model used in this paper is based on London's Barclays Cycle Hire scheme. For modeling, we used three data sets made publicly available by the Transport for London authority.[1]

1) 1.42 million rides spanning a period of 97 days (examples in Table I).
2) Location and capacity of 354 stations actively used during the recorded period (examples in Table II).
3) An initial station fill level recorded during nighttime when all bicycles were docked. In total, we estimate that the system contained 3708 bicycles at the time for which historical data are available.

TABLE II
STATION INFORMATION SAMPLES

id	name	position (lat, lon)	capacity
1	River St, Clerkenwell	{51.5291, -0.1099}	18
2	Phillimore Gardens, Kensington	{51.4996, -0.1975}	34
3	Christopher St, Liverpool St	{51.5212, -0.0846}	33
4	St. Chad's Street, King's Cross	{51.5300, -0.1200}	22

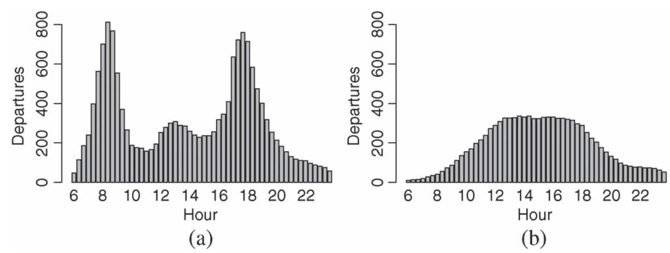

Fig. 2. (a) Average departure rates on weekdays. (b) Average departure rates on weekends.

Analysis of the historical journeys reveals regular daily flow patterns, with a substantial difference between weekdays and weekends. Journeys are allowed between 6 A.M. and midnight. As expected, many customers commute to the city center in the morning and ride back to the outer districts in the late afternoon. This pattern, absent on weekends, causes two spikes in daily rental activities that are illustrated in Fig. 2(a).

B. Model Parameters

In our model, we define a set S containing all stations $s \in S$. Time t is assumed to be discrete and indexed on a 1-min level, where T_{hist} denotes all time steps of the observed period. We distinguish between workdays and days on weekends by the binary variable $w \in \{\text{weekday, weekend}\}$ and split every day into 72 slices $k \in K$ of 20 min each. Time is mapped to day-type and timeslice using $w(t)$ and $k(t)$, respectively. All customer departure and arrival events are counted in matrices of dimension $|S| \times |S|$. The sum of departing customers going from station i to j in a timeslice k and on a day w is $D_{i,j}(k,w)$; similarly, the sum of customers who arrive at station j coming from i is $A_{i,j}(k,w)$. The average number of such arrival events $\Lambda_{i,j}(t)$ and departure events $M_{i,j}(t)$ at time t in the historical data can thus be expressed as

$$M_{i,j}(t) = \frac{D_{i,j}\left(k(t), w(t)\right)}{|\{t' \in T_{\text{hist}} : k(t') = k(t), w(t') = w(t)\}|} \quad (1)$$

$$\Lambda_{i,j}(t) = \frac{A_{i,j}\left(k(t), w(t)\right)}{|\{t' \in T_{\text{hist}} : k(t') = k(t), w(t') = w(t)\}|}. \quad (2)$$

Here, the denominator gives the duration of the recorded history for a given timeslice and day-type indicated by t.

Based on these average numbers of events happening per time step, customer departures are exponentially distributed with time-varying parameter $M_{i,j}(t)$. This parameter fit is based on the following implicit assumptions.

- 100% service rate for departures in the historical data. Potential customers who could not rent a bicycle due to

[1] http://www.tfl.gov.uk/businessandpartners/syndication/default.aspx.

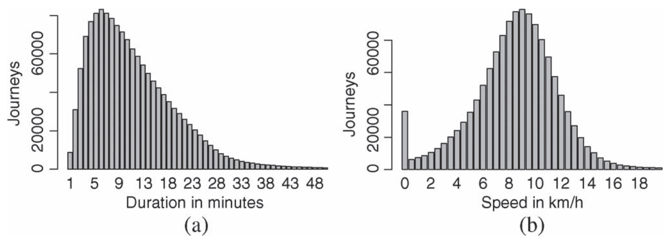

Fig. 3. (a) Distribution of the journey duration in the historical data. (b) Journey speed distribution (Euclidean station distances).

an empty station are excluded, as they are not recorded in the historical data. This assumption is justified to some degree by the considerable redistribution effort made by the operator of the London PBS scheme [8].

- Independence of customers. Departure rates of customers vary with time and type of the day, but the distribution of departures within a time frame is memoryless. As a caveat, this does not accurately model customer groups, for example, tourists.
- Effects of season, weather, events, etc., are disregarded but could easily be included in a more detailed model.

If a customer has departed at a station, the probability distribution of his destinations is given by their relative frequency in the historical data, as recorded in $M_{i,j}(t)$. The total expected departure $\mu_s(t)$ and arrival $\lambda_s(t)$ at each station and the *net arrival* $\eta_s(t)$ during time step t is therefore

$$\mu_s(t) = \sum_{\tilde{s}\in S} M_{s,\tilde{s}}(t), \quad \lambda_s(t) = \sum_{\tilde{s}\in S} \Lambda_{s,\tilde{s}}(t) \tag{3}$$

$$\eta_s(t) = \lambda_s(t) - \mu_s(t) \tag{4}$$

given that enough bicycles and free slots are available.

In addition to the customer's arrival and departure rate, the following assumptions about their behavior are needed to simulate the system.

- Customers who want to depart from a station that turns out to be empty leave without starting a journey. They do not wait for a bicycle to be returned, nor do they walk on to a neighboring station.
- The travel time between any two stations i and j is always equal to the average travel time extracted from the historical data. Fig. 3 depicts the historical distribution of travel speeds and journey durations.
- Customers who arrive at their target station wanting to return their bicycle when the station is full ride on to one of the neighboring stations (chosen according to his perceived utility, as described in Section II-C). If this station is also full, a customer will go on to the next station, but he does not return to any station already visited.

C. Customer Decision Model

In order to investigate the effects of offering price incentives, a model of how the customer reacts is required. We assume all customers place a value on the additional time they would spend traveling if they were to accept an incentive. This is equivalent to penalizing a longer travel distance. The additional distance a customer has to travel if he changes his target station consists of the additional distance he has to bicycle, plus an additional walking distance to his final destination. For simplicity, let us assume that the final destination of a customer traveling to station s_i is indeed the center of mass m_i of the corresponding Voronoi partition around each station (see Fig. 4). The Voronoi region is the polytope that contains all points closer to a given station than to any other. Moreover, let us assume that the walking speed is half the cycling speed and that the distance on the map is given by Euclidean distance d_{eucl}. The *effective distance* $\tilde{d}_{i,j}$, which adds to a customer's travel time when switching from station s_i to s_j, can be expressed as

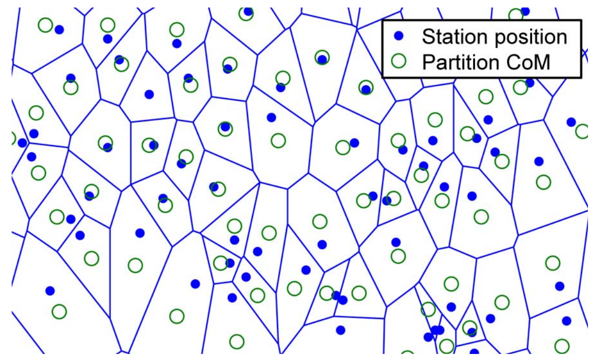

Fig. 4. Voronoi partitioning with centers of mass (CoM) of the London bicycle-sharing system.

$$\tilde{d}_{i,j} = d_{\mathrm{eucl}}(s_i, s_j) + 2 \cdot (d_{\mathrm{eucl}}(s_j, m_i) - d_{\mathrm{eucl}}(s_i, m_i)). \tag{5}$$

The incentives to go to a neighboring station are offered to customers upon their arrival. Each customer decides whether to take an incentive by maximizing his personal benefit based on the incentive payout and the customer's perceived cost of additionally traveled distance. This implies customer rationality and makes the choice independent from the original pricing of journeys. For each end station $s \in S$, the set of neighboring stations to which a price incentive could be offered is N_s, and $p_{s,n}$ denotes the amount of money offered to go from station s to neighbor $n \in N_s$. In addition, let $\tilde{N}_s$ be the set of stations having s in their respective neighbor set. The following model of customer reactions is used.

1) The marginal cost of travel c for each arriving customer is drawn from a uniform distribution $C \sim U[0, c_{\max}]$, where we have used $c_{\max} = £20/\mathrm{km}$ in our simulations.[2]
2) The customer selects the best offer of maximum value as

$$n^* = \arg\max_{n\in N_s} (p_{s,n} - d_{s,n}c). \tag{6}$$

3) If the original target station is full, i.e., the customer cannot return his bicycle there, he always chooses the best incentive by going to n^*. If there is space left, he takes an incentive only if the perceived value of the best incentive is positive with $p_{s,n^*} - d_{s,n^*}c > 0$.

[2]Future work could incorporate more detailed models of how customers value their time (see [9] and [10]). In addition, instead of having a single distribution, one could also differentiate between customer types (e.g., those commuting to work and those riding during leisure times) by introducing a time-varying component.

The probability $\pi(s, n, p_s)$ of an arriving customer taking an incentive to neighbor $n \in N_s$ for a given payout vector p_s depends on the distribution of perceived travel costs c. First, the offering to go to n must have the highest perceived value among all incentives offered to neighboring stations. Second, assuming that station s is not full, the perceived cost for traveling the additional distance must be lower than the relevant payout $p_{s,n}$. Thus

$$\pi(s, n, p_s) = P\left(p_{s,n} \geq c \cdot \tilde{d}_{s,n} \wedge p_{s,n} - (c \cdot \tilde{d}_{s,n}) \geq p_{s,n'} - (c \cdot \tilde{d}_{s,n'}), \forall n' \in N_s\right). \quad (7)$$

III. Utility of Changes in Station Fill Level

In this paper, two methods are considered for influencing the distribution of bicycles in the PBS: manual redistribution (see Section IV) and price-led redistribution via incentives (see Section V). However, as a basis for both algorithms, it is necessary to estimate the positive or negative effects any change in the stations' fill levels will bring about. Since the system is stochastic, these are not straightforward to assess. Here, we introduce a function that estimates the utility of changes in fill levels for a given station.

Raviv and Kolka [11] have done related work in order to determine the best fill level of each station in a static redistribution setting. Their approach tracks the probability of all possible fill levels based on a discrete-time approximation of the underlying continuous birth–death process. However, if we were to adopt this method, the dimension of the resulting optimization problem would significantly increase the computational complexity of the proposed approaches in Sections IV and V.

We make the simplifying assumption that arrivals and departures are deterministic and given by the expected net change $\eta_s(t)$. This results in a coarser model than, for example, the probabilistic approach in [11]. However, this simple parameterization is attractive in that it leads to tractable optimization problems, as will be seen in Sections IV and V.

Furthermore, we define the utility of changes to a station's fill level as the difference in the number of customers successfully served if arrivals and departures over a limited time horizon still occur exactly at their expected rates, but with the station initialized at this changed fill level. The benefit of any redistribution action (adding or taking away bicycles at a single station) at the current time can then be evaluated based on this notion.

For a given station s and starting time t_0, we precompute the expected future fill level f_t^s over a time horizon with $t = t_0, \ldots, t_0 + T_{\text{util}}$, where $T_{\text{util}} = 24$ h is the look-ahead period considered. The expected fill level is governed by the following dynamics:

$$f_{t+1}^s = \max\left(0, \min(f_t^s + \eta_s(t), f_{\max}^s)\right) \quad (8)$$

where $f_{\max}^s$ denotes the maximum capacity of the station, and the $\max$ and $\min$ functions represent the fact that the station never becomes "more than full" or "less than empty." The quantity $\eta_s(t)$ is the net arrival rate defined by (4).

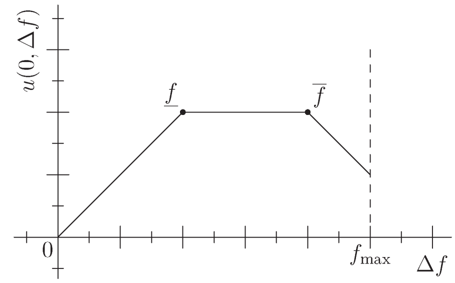

Fig. 5. Example for an empty station ($f = 0$), showing the utility "plateau" between the fill levels $\underline{f}_t^s$ and $\overline{f}_t^s$, based on the expected net arrival rate over the time horizon.

Adding or taking away bicycles from the station at the current time changes how many customers can be served later on in the time horizon, since the station will become empty or full at different times in the future. For a current fill level f_t^s and a change in fill level Δf_t^s to be made at time t, Algorithm 1 computes the utility $u(s, t, f_t^s, \Delta f_t^s)$ by comparing the two scenarios of different initial fill levels. The algorithm steps through the time horizon, measuring the difference in the number of customers served under the nominal pattern. If the amount of change is lower in the original case, a station capacity constraint is being hit earlier than in the adapted case, and vice versa. The difference between Δ and $\tilde{\Delta}$ is the difference in customers that were successfully served in that time period. The procedure aborts if the end of the time horizon has been reached or if the fill level of the station becomes equal in both cases (e.g., both are full or empty).

Algorithm 1 Utility of a change to the fill level

Require: $s \in S, T_{\text{util}}$ ▷ Relevant station and horizon length,
$f_{\max}^s$ ▷ Maximum capacity of station s,
$\eta_s(t)$ ▷ Expected net arrival of customers,
$f_t^s \; \forall t \in \{t_0, \ldots, t_0 + T_{\text{util}}\}$ ▷ Precomputed fill levels in the original case according to (8),
$\tilde{f}_{t_0}^s = f_{t_0}^s + \Delta f_{t_0}^s$ ▷ Starting fill level in the case with redistribution
procedure RedistributionUtility$(t, \tilde{f}_t^s)$
  if $\tilde{f}_t^s = f_t^s$ **or** $t \geq T_{\text{util}}$ **then**
    return 0
  end if
  $\tilde{f}_{t+1}^s \leftarrow \max(0, \min(\tilde{f}_t^s + \eta_s(t), f_{\max}^s))$
  $\Delta \leftarrow |f_t^s - f_{t+1}^s|$
  $\tilde{\Delta} \leftarrow |\tilde{f}_t^s - \tilde{f}_{t+1}^s|$
  return $\tilde{\Delta} - \Delta +$ RedistributionUtility$(t + 1, \tilde{f}_{t+1}^s)$
end procedure

Although the worst case time complexity of Algorithm 1 is only linear with respect to the horizon length, it would take too long to use online in the later optimization steps. However, as formalized in Theorem 1, the utility function always takes the "plateau" shape shown in Fig. 5. It can therefore be efficiently stored in a lookup table for each station and time period.

Theorem 1: For any station $s \in S$ at time t, there exist two fill levels $\underline{f}_t^s, \overline{f}_t^s \in [0, f_{\max}^s]$ independent from the initial fill level f_t^s such that for the utility of change in fill level Δf_t^s, we have

$$\frac{\partial u\left(s,t,f_t^s,\Delta f_t^s\right)}{\partial \Delta f_t^s} = \begin{cases} 1, & \text{if } f_t^s + \Delta f_t^s < \underline{f}_t^s \\ 0, & \text{if } \underline{f}_t^s \leq f_t^s + \Delta f_t^s \leq \overline{f}_t^s \\ -1, & \text{if } f_t^s + \Delta f_t^s > \overline{f}_t^s. \end{cases} \quad (9)$$

The utility of no change of the station's fill level is understood to be zero, i.e., $u(s,t,f_t^s,0) := 0$. The (possibly empty) interval $[\underline{f}_t^s, \overline{f}_t^s]$ is called the utility "plateau" of constant maximum utility.

Proof: Assume the station will become full within time horizon T_{util} for initial fill level $f_t^s + \Delta f_t^s$. Adding δ bicycles to the station implies that δ additional expected customers who want to return their bicycles have to be rejected. Therefore, utility $u(s,t,f_t^s,\Delta f_t^s + \delta)$ monotonically decreases with slope -1 for any $\delta \geq 0$, $f + \Delta f_t^s + \delta \leq f_{\max}^s$. Thus

$$u\left(s,t,f_t^s,\Delta f_t^s + \delta\right) = u\left(s,t,f_t^s,\Delta f_t^s\right) - \delta.$$

A similar case can be made for fill levels where the station is running empty. Only, the utility decreases when more bicycles are removed with a slope of u equal to 1. It naturally follows that for a given initial fill level, there are thresholds $\underline{f}_t^s$, $\overline{f}_t^s$, with $\underline{f}_t^s < \overline{f}_t^s$, where the station first starts to run empty or full within the horizon. Within interval $[\underline{f}_t^s, \overline{f}_t^s]$, the station's capacity constraints are not hit, and the utility function must therefore be constant, leading to a "plateau" of the type shown in Fig. 5. ∎

Sections IV and V will make use of this characterization of station fill utilities in order to choose how trucks redistribute bicycles and which price incentives are offered to customers.

IV. Dynamic Truck-Routing Algorithm

This section describes an algorithm for intelligent operation of a fleet of R trucks, which move bicycles between stations as needed. Their objective is to increase the system utility (as defined in Section III) and, hence ultimately, the system's service level, as much as possible.

The problem of how best to redistribute vehicles manually within shared mobility schemes is not new. It originated in pilot car-sharing projects, such as *Praxitèle* [12], [13], *Intellishare* [14], and *Honda ICVS* [15]. However, the redistribution planning algorithms used in car-sharing projects do not directly translate to the public bicycle-sharing scheme considered in this paper. First, each of these algorithms exhibits certain characteristics that are specific to its corresponding vehicle-sharing system, for example, charging times of electric vehicles. Second, car-sharing systems tend to be much smaller in their network size than the PBS considered in our paper, and the proposed algorithms cannot be easily scaled to several hundred stations.

Intelligent redistribution in bicycle-sharing schemes has also received prior attention in the literature [1]. The proposed approaches can be separated into static and dynamic redistribution.

In the static redistribution approach, an optimal route is computed in order to attain a predefined fill level for each station, prior to customers interacting with the system (e.g., during the night). For example, Benchimol *et al.* [16] presented a solution for the routing of a single truck, and Raviv *et al.* [17] considered the case of multiple trucks. Expanding on this, Schuijbroek *et al.* [18] partitioned the area for redistribution to speed up search for multiple trucks. The advantage of static redistribution is that there is ample time to compute a good truck-routing solution. However, static redistribution cannot react to unforeseen variations in the demand pattern during the day, caused for example by unexpected weather conditions. In addition, nighttime redistribution of bicycles has been restricted for some PBS (see [19] for the London case).

In the dynamic redistribution approach, truck routing is planned in a receding horizon fashion while the system is in full operation. This allows the planner to react online to unexpected changes in the system's state. As such, it is a more suitable approach to our problem. Nair and Miller-Hooks [20] used a stochastic formulation for dynamic redistribution based on stationary distributions for customer arrivals and departures. However, their approach is not applicable to the PBS model employed in this paper, which is driven by strongly time-varying customer patterns. Contardo *et al.* [21] presented another dynamic redistribution approach with time-varying deterministic future flow patterns. However, the computational complexity of their approach is prohibitive for our system, because it is too expensive to simulate the system with multiple trucks over a long time horizon. Nair *et al.* [22] considered the stochastic demand for bicycles and free space during the next time period in order to decide on redistribution. However, they do not take time and load constraints of truck usage into account and restrict their approach to choosing "swaps" of bicycles between pairs of stations.

This paper considers the dynamic redistribution case only. It is a variation of the routing problem with pickups and deliveries for one commodity, taken from [23]. In order to determine a truck route, time is discretized into 5-min intervals, and the planning problem is considered on the time-expanded network (see Section IV-A). During each interval, customer behavior is assumed to be time varying but deterministic. A receding planning horizon is considered, the length of which is the maximum of the truck visiting four stations and $T_{\text{truck}} = 40$ min.[3] The period for reoptimizing of truck routes (implementation horizon) is chosen as $T_{\text{impl}} = 30$ min, which is based on a tradeoff between computation time and performance. Note that as shown in Fig. 6, the planning horizon is longer than the reoptimization cycle. This improves the performance of truck journeys beginning shortly before the next reoptimization, as these journeys are likely to end within the longer planning horizon where subsequent opportunity is still considered.

To solve the routing problem for a single truck (see Section IV-B) over the finite planning horizon, we adopt a two-step approach. First, for each truck, we construct a tree of "promising route candidates" using a greedy heuristic. This

[3]Using the utility measure in Section III implies a horizon $T_{\text{util}} = 24$ h in which individual redistribution actions work out their effects.

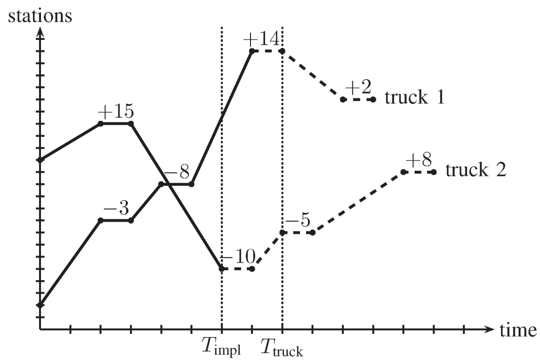

Fig. 6. Dynamic truck-routing algorithm for $R = 2$ trucks. The time axis is discretized into 5-min intervals. The planning horizon is the maximum of $N_{\mathrm{truck}} = 4$ stations visited by the truck and $T_{\mathrm{truck}} = 40$ min; the implementation horizon is $T_{\mathrm{impl}} = 30$ min. Each line represents the route of one truck, where journeys are indicated by a solid line if they start within the implementation horizon (i.e., they are definitely executed) or by a dashed line if they start with the planning horizon (i.e., they may be subject to change at future replanning). The trucks wait at each stop for 5 min in order to load or unload bicycles. The number of bicycles loaded and unloaded at each stop is indicated as well.

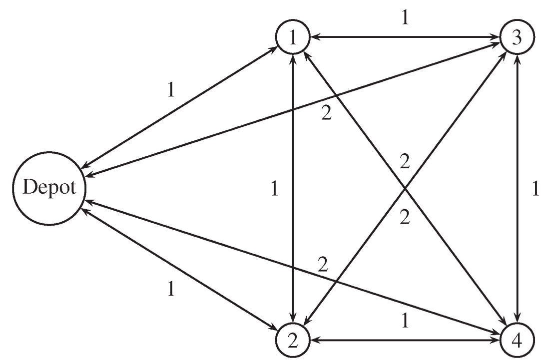

Fig. 7. Example station graph, with arcs weighted according to distance.

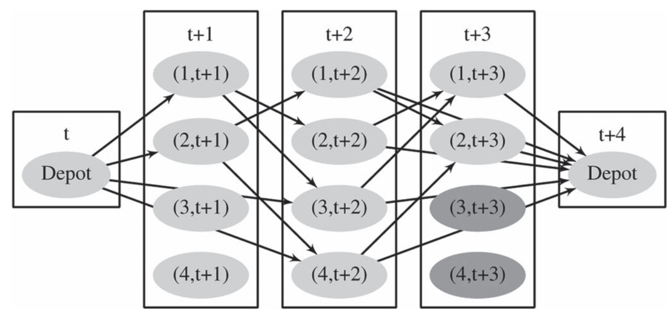

Fig. 8. Example time-expanded network. Dark-gray vertices are marked inaccessible due to the terminal condition.

means that truck routes are extended by stations that promise high ratios of utility added per time to travel. Second, for each of the promising routes, the optimal number of bicycles to be loaded or unloaded at each stop is optimized as the complete routes have become known. Then, the route providing the highest utility improvement is selected. Finally, we extend this algorithm in a simple manner to multiple trucks (see Section IV-C).

A. *Modeling Redistribution Truck Routes on a Time-Expanded Network*

Our truck-routing algorithm employs a time-expanded network of graph $G = (V, A)$ [15], [21], which is the basis of our truck-routing algorithm. The vertices V of this graph consist of tuples $v = \{(s, t), s \in S, t \in T\} \in V$.[4] For each vertex v, the (expected future) fill level $f_{t(v)}^{s(v)}$ is generated according to (8). The arcs $a = (v_1, v_2) \in A$ correspond to possible journeys a truck is able to take.

In our model, the time it takes for a truck to traverse an arc is discretized to multiples of 5 min. It is computed based on the Euclidean distance (in kilometers) $d_{\mathrm{eucl}}(s_i, s_j)$ between two stations s_i, $s_j \in S$, and the assumption that trucks have an average speed of 15 km/h in city traffic. Including an additional 5 min for bicycle handling after reaching the station, the resulting *effective journey time* (in time steps of 5 min) for a truck to go from station s_i to s_j is

$$\bar{d}(s_i, s_j) := \lceil d_{\mathrm{eucl}}(s_i, s_j)/1.25 \rceil + 1. \tag{10}$$

Note that the dividing factor of 1.25 results from converting distance (in kilometers) into time steps (of 5 min). The redistribution trucks have limited operation hours during the day, which is set to 8 A.M.–10 P.M. All trucks are constrained to start at a maintenance depot in the morning and to finish at this depot at the end of the working day. As a consequence, vertices from which no combination of arcs leads back to the depot on time are excluded from the graph. An example of a network of stations and a corresponding time-expanded network are shown in Figs. 7 and 8.

B. *Computing Single-Truck Routes*

Here, we present an algorithm for finding a redistribution route for a single truck in the dynamic case. In contrast to the static redistribution case, the goal is not to attain a defined system state using as little resources as possible but rather to optimally invest the available resources, i.e., operational hours of the truck. Thus, based on the utility of fill level changes defined in Section III, the ratio of added utility per invested time of truck r is maximized.

a) Constructing a Tree of Promising Candidate Routes: Consider a single truck $r \in R$ with a load capacity of $l_{\max}$ bicycles. The goal is to determine the truck's future *redistribution actions* $\omega_i = (v_i, \Delta f_i, l_i) \in P : v_i \in V,\ \ l_i \in \{0, \ldots, l_{\max}\}, i \in \{1, \ldots, N_{\mathrm{truck}}\}$. The load l_i is defined as the number of bicycles in the truck after performing the redistribution action Δf_i at the station and time indicated by the vertex of the time-expanded network v_i. In particular, for every pair of actions ω_i, ω_{i+1} with their respective time-indexed stations v_i, v_{i+1}, there must exist an arc $a \in A$ with

$$v_1(a) = v_i \ \wedge\ v_2(a) = v_{i+1}, \quad \forall i \in \{1, \ldots, N_{\mathrm{truck}} - 1\}. \tag{11}$$

[4]The notation $s(v)$ is used to access the component s of tuple $v = (s, t)$.

Moreover, the following consistency constraints for station fill levels and truck loads must hold:

$$f_{t(v_i)}^{s(v_i)} = f_{t(v_i)-1}^{s(v_i)} + \Delta f_i \in \left[0, f_{\max}^{s(v_i)}\right], \quad \forall\, i \in \{1, \ldots, N_{\text{truck}}\} \tag{12a}$$

$$l_{i+1} = l_i - \Delta f_{i+1} \in \{0, \ldots, l_{\max}\}, \quad \forall\, i \in \{1, \ldots, N_{\text{truck}} - 1\}. \tag{12b}$$

Equation (12a) incorporates the redistribution actions in the predicted station fill levels, otherwise governed by (8). With (12b), the truck's capacity constraint is ensured to hold.

Starting from an initial position ω_1, the possible next steps can be represented by a tree graph Φ, where each node ϕ represents a specific redistribution action, and each leaf node determines a unique truck route. The tree of all possible routes has a branching factor of $|S - 1|$, since a route could possibly lead to any of the other stations for the next redistribution action. Hence, there are $|S - 1|^{N_{\text{truck}}-1}$ possible combinations of stations for a route of length N_{truck} (where the initial position ω_1 is already known). For systems consisting of several hundred stations, it is thus not viable to test every possible combination. The complexity of the problem is reduced by concentrating on a subset of possible routes corresponding to the most promising redistribution actions. It works by constructing a pruned version of the route tree. Starting from the truck's initial position (at the root), the tree is recursively extended at each of its leaf nodes until it has reached the desired depth of N_{truck} (or the journey has a minimum duration of T_{truck}):

1) The current leaf node is $\phi = (v_\phi, \Delta f_\phi, l_\phi)$. First, we compute the "value per unit distance" the truck might bring by going to any of the other stations. The set $\tilde{V} := \{v \in V : \exists a \in A, v_1(a) = v_\phi, v_2(a) = v\}$ contains the vertices of the time-expanded network the truck would reach by going to any of the other stations. Since we know the current load of bicycles l_ϕ, the best action to be performed at $\tilde{v} \in \tilde{V}$ can be computed with a "greedy" approach as

$$\Delta f^*(\tilde{v}, \phi) = \begin{cases} \max\left(l_\phi - l_{\max}, \left\lceil \overline{f}_{t(\tilde{v})}^{s(\tilde{v})} - f(\tilde{v}) \right\rceil\right), & \text{if } f(\tilde{v}) > \overline{f}(\tilde{v}) \\ \min\left(l_\phi, \left\lfloor \underline{f}_{t(\tilde{v})}^{s(\tilde{v})} - f(\tilde{v}) \right\rfloor\right), & \text{if } f(\tilde{v}) < \underline{f}(\tilde{v}) \\ 0, & \text{else.} \end{cases} \tag{13}$$

We choose the K vertices with the best ratio $\Delta f^*(\tilde{v}, \phi)/\bar{d}(v_\phi, \tilde{v})$ and add them as leaves of ϕ in the form of route steps. The set of new leaf nodes is Φ_ϕ.

2) In addition, we add stations that could serve as an intermittent depot. Going there may not yield a direct utility. However, the possibility to bring or take bicycles may be of use at other stations of the route. We choose

$$\tilde{v}_{\text{store}} = \underset{\tilde{v} \in \tilde{V} : s(\tilde{v}) \neq s(v_\phi)}{\arg\max} \frac{\min\left(l_{\text{depot}}, \overline{f}_{t(\tilde{v})}^{s(\tilde{v})} - f(\tilde{v})\right)}{\bar{d}(v_\phi, \tilde{v})} \tag{14a}$$

$$\tilde{v}_{\text{pick}} = \underset{\tilde{v} \in \tilde{V} : s(\tilde{v}) \neq s(v_\phi)}{\arg\max} \frac{\min\left(l_{\text{depot}}, f(\tilde{v}) - \underline{f}_{t(\tilde{v})}^{s(\tilde{v})}\right)}{\bar{d}(v_\phi, \tilde{v})} \tag{14b}$$

and add them to the set of leafs Φ_ϕ with a redistribution action of zero. To prevent that $\tilde{v}_{\text{store}}$, $\tilde{v}_{\text{pick}}$ are only set to very large stations, we cap the maximum intermittent depot capacity considered to some $l_{\text{depot}} \leq l_{\max}$. How stations that may serve as intermittent depots are incorporated into the actions at other steps of the route is explained in the following.

3) If the depth of the recursive procedure has not yet reached the final depth N_{truck}, it is repeated for every $\phi' \in \Phi_\phi$. Before entering the recursive procedure at ϕ', the corresponding redistribution action $\Delta f_{\phi'}$ is incorporated into the predicted future fill levels f_t^s. These changes, of course, have to be unwound between the $\phi' \in \Phi_\phi$.

If even more aggressive tree pruning is necessary to comply with computational constraints, the similar beam search [24], [25] can be applied. It leads to linear complexity in the route length, but at the expense of discarding more potentially optimal solutions. In beam search, a greedy approach is used to determine promising next steps as well, but only K leaf nodes are added in total to all nodes of the same height. Resorting to beam search was, however, not necessary for the route length horizon used in the sample setting of this paper.

b) Refining Truck Loading Actions: The greedy heuristic used to construct the tree of candidate routes Φ could not know about stations visited later in each route. Knowing the complete routes, their respective action profiles can be further refined. As a motivating example, it may be beneficial to pick up more bicycles than the utility function of a single station $u(s, t, f_t^s, \Delta f_t^s)$ (see Section III) originally indicated, that is, if taking more bicycles has zero local utility (the fill level remains within the utility plateau); however, a subsequent station in the route can make use of the additional bicycles. The problem of choosing optimal actions can be formulated as a manageable quadratic program (QP).

$s(i), t(i)$	The station and time at the ith step in the route for $i \in \{1, \ldots, N_{\text{truck}}\}$.
f_i	The expected fill level $f_{t(i)}^{s(i)}$ of station $s(i)$ at time $t(i)$.
Δf_i	The action performed at step i. This is the optimization variable.
$\underline{f}(i), \overline{f}(i)$	Beginning and end of the utility plateau of station $s(i)$ at time $t(i)$, as described in Section III.
$\Delta \underline{f}(i), \Delta \overline{f}(i)$	Difference between the new fill level $f_i + \Delta f_i$ and the plateau beginning/end $\underline{f}(i), \overline{f}(i)$. The difference is defined to grow positively going outward from the respective side of the plateau.
$\Delta \underline{f}'(i), \Delta \overline{f}'(i)$	Auxiliary variables containing the absolute difference from the plateau beginning $\Delta \underline{f}'(i) = \lvert\Delta \underline{f}(i)\rvert$ or end $\Delta \overline{f}'(i) = \lvert\Delta \overline{f}(i)\rvert$. They are correctly set by the solver minimizing costs within the bounds set in (15e) and (15f).
$l_{\text{init}}, l_{\max}$	Initial fill level of the redistribution truck and the maximum truck load capacity.

$q \gg 2l_{\max}^2 + 1$ Scaling factor for penalizing redistribution actions.

$$\min \sum_{i=1}^{N_{\text{truck}}} \Delta \underline{f}(i) + \Delta \underline{f}'(i) + \Delta \overline{f}(i) + \Delta \overline{f}'(i) + \Delta f_i^2/q \tag{15a}$$

such that

$$0 \leq l_{\text{init}} - \sum_{i'=1}^{i} \Delta f_{i'} \leq l_{\max}, \quad \forall i \in \{1, \ldots, N_{\text{truck}}\} \tag{15b}$$

$$\Delta \underline{f}(i) = \underline{f}(i) - f_i - \Delta f_i, \quad \forall i \in \{1, \ldots, N_{\text{truck}}\} \tag{15c}$$

$$\Delta \overline{f}(i) = -\overline{f}(i) + f_i + \Delta f_i, \ \forall i \in \{1, \ldots, N_{\text{truck}}\} \tag{15d}$$

$$-\Delta \underline{f}'(i) \leq \Delta \underline{f}(i) \leq \Delta \underline{f}'(i), \quad \forall i \in \{1, \ldots, N_{\text{truck}}\} \tag{15e}$$

$$-\Delta \overline{f}'(i) \leq \Delta \overline{f}(i) \leq \Delta \overline{f}'(i), \quad \forall i \in \{1, \ldots, N_{\text{truck}}\}. \tag{15f}$$

The linear part of objective function (15a) evaluates redistribution actions according to the utility definition in Section III. The quadratic part of (15a) minimizes the action of the truck operators (they take the fewest bicycles possible). It is scaled such that actions with a positive utility will still be performed. However, it prevents "neutral" actions causing negative utility at one station and the equal positive utility at another step in the route. Since the plateau is always within the station capacity constraints, this also ensures that actions remain feasible. Equation (15b) ensures that the fill level of trucks stays within the capacity constraints. To speed up this often-repeated refinement step, we relax variables representing bicycle quantities to continuous (rather than integer) values. The solution is then manually fitted by clipping any noninteger parts. Computational tests showed that this relaxation makes little difference to the solution.

We now determine the best set of redistribution actions for all promising routes and choose the route that results in the best overall utility increase per unit time.

C. Routing Multiple Trucks

Cooptimized routes for several trucks are too difficult to compute online within the time constraints of the PBS system. Therefore, the route search for multiple trucks is interleaved but independently computed for each single truck. Of course, the actions of prior trucks can be treated as known. They are manifest in the station fill levels f_s^t considered for subsequent trucks.

For every truck $r \in R$, Ω_r gives its current route as an ordered set of future redistribution actions. Initially, each Ω_r contains only a single redistribution action, indicating the next known position and fill level of the truck (it may be that the truck is currently under way when the routes are updated).

Choosing one of the trucks that have a minimum time index for the last action in Ω_r, we append a single redistribution action to its route. For this, we compute a route of length N_{truck} starting at the last entry of Ω_r according to the procedure described in the previous section. However, only the first step of this route is appended to Ω_r. Later steps are discarded since they are prone to being suboptimal given the not-yet-known actions of the remaining trucks. This step is repeated until all routes have a duration larger than T_{truck}.

However, situations may arise leading truck routes to "collide." Assume, without loss of generality, that the trucks $r \in R$ are ordered according to the time index of the last redistribution action in Ω_r. If truck r chooses to take an action at a time-indexed station v to which truck $r' < r$ has already planned to go but arrives at a later time with $t(v') > t(v)$, $s(v') = s(v)$, then r' has made his choice based on false assumptions about the station's fill level upon his arrival. These collisions can be handled by removing the offending action from $\Omega_{r'}$ (as well as eventual subsequent actions by r') and updating the fill levels f_t^s to reflect the change.

V. Dynamic Price Incentives for Users

Customers themselves might contribute to the rebalancing of a PBS scheme if offered an appropriate incentive (monetary or otherwise). In many cases, it would already be possible to communicate and pay these incentives to users via existing infrastructure, such as kiosk terminals or mobile phone applications. Here, we consider a mechanism that dynamically chooses payments offered to customers to change the endpoint of their journey to a nearby station in a way that improves the overall service level. To this end, we take the model of how customers accept (or choose between) price offers, as described in Section II-C, and then formulate an optimization problem trading off the expected payouts and the expected improvement in service level. The solution is a set of incentives for each station, assigning price incentives pointing toward a limited number of neighboring stations where customers might leave their bicycles instead.

Using price incentives to induce a desired behavior in the users of shared mobility systems has been examined in multiple contexts in recent literature. The work of Barth *et al.* [14] examines user-based redistribution in a shared mobility system. However, the approach of splitting and merging rides can only be applied to cars and not to public bicycle hire schemes. Incentives for bicycle-sharing schemes are investigated by [26], where users pick two stations at random and go to the more empty.

Next, we propose a novel scheme for setting incentives based on model predictive control (MPC) [27]. The advantages of MPC are the ability to respond to future events predicted by the system model and time-dependent changes to the system dynamics within the considered time horizon. Furthermore, the system model can incorporate constraints on the states and inputs of the system. In the case of the bicycle-sharing scheme, this allows us to bound the price incentives offered to customers. Importantly, MPC is used for online control where the solution is periodically recomputed with an updated initial system state.

A. Simplified Model of Customer Behavior

We now derive an approximate model of customer behavior that can be used in the context of MPC. Customer behavior described in (7) means their response to price incentives and, therefore, enters into the system dynamics $f_t(x(t), u(t))$. However, this response is nonlinear, which leads to a nonconvex MPC problem that cannot be efficiently solved.

We first explain the origin of this nonlinearity. Consider two neighbor stations n', $n'' \in N_s$ with equal distance to s, for which the incentives offered from station s are equal, i.e., $p_{s,n'} = p_{s,n''}$. If there are customers equally willing to go to n' or n'', an infinitesimally small increase in $p_{s,n'}$ would cause all those customers to choose n' if we assume they act totally rational.

To formulate a tractable MPC problem, we approximate the customer behavior $\pi(s, n, p_s)$ from Section II-C in a linear fashion. To this end, we choose a convenient set N_s of nearby stations with $N_s = N$ and compute the linearization $\bar{\pi}(s, n, p_s)$ using Algorithm 2. It creates samples of customer reactions to random incentive offers (to all neighboring stations) and performs a least squares fit between observed behavior and the linear model. Note that the probability of a customer to choose an incentive depends not only on the incentive itself but also on the alternative incentives offered to the other nearby stations. The linear approximation of the customer's probability to reject station s and go to neighbor n instead is defined by vectors $\tilde{\pi}_{s,n}$ of size N, for each $s \in S$, $n \in N_s$. Thus

$$\bar{\pi}(s, n, p_s) = \tilde{\pi}_{s,n}^{\top} p_s. \tag{16}$$

Algorithm 2 Fitting the linear customer behavior model

Require: $s \in S$,
N_s, $\triangleright |N_s| = N$ nearest neighbors around station s
$\text{TAKEN\_INCENTIVE}(s, N_s, p)$, $\triangleright$ Neighbor chosen by the customer as described in Section II-C
$p_{\max}$, $\triangleright$ Maximum payout
$P \gg 0$, $\triangleright$ Number of generated payout vectors (samples)
$C \gg 0$, $\triangleright$ Number of customer behavior samples
Ψ $\triangleright$ Set of samples. Each sample is a tuple of two vectors: the offered payouts p to the N neighbors and the percentage of customers taking a certain incentive δ.
for $i = 1$ to P **do**
  $p \leftarrow p_{\max} \cdot \text{RAND}(N) \in [0, p_{\max}]^N$ $\triangleright$ Payout vector
  $e \leftarrow \{0\}^N$ $\triangleright$ Initialize behavior count
  for $c = 1$ to C **do**
    $n' \leftarrow \text{TAKEN\_INCENTIVE}(s, N_s, p) \in \{\mathbb{N}^+, \emptyset\}$
    $e_{n'} \leftarrow e_{n'} + 1$
  end for
  $\delta \leftarrow e/C$ $\triangleright$ Fraction taking a certain incentive
  $\Psi_i \leftarrow (p, \delta)$
end for
for $n = 1$ to N **do**
  $\tilde{\pi}_{s,n} \leftarrow \arg\min_{\pi} \sum_{i \in \{1,\ldots,P\}} \left(\pi^{\top} p(\Psi_i) - \delta(\Psi_i)_n\right)^2$
end for

B. Computing Dynamic Price Incentives

The state of the system in each time period of the horizon $t \in [0, \ldots, T_{\text{price}}]$ is defined as the number of bicycles present at each station. The control inputs are the price incentives offered at each station for diversions to neighboring stations. The objective function defines a tradeoff between the quality of the achieved system state and the cost for paying out incentives. The solution gives the price incentives $p_s(t)$ to be offered to customers, where the prices $p_s(0)$ are immediately issued, and $p_s(1), \ldots, p_s(T-1)$ are prices planned for subsequent time periods. However, since prices are recomputed after every time period (with a shifted horizon), only the prices $p_s(0)$ are ever really applied.

The number $f_s(t)$ of bicycles present at station s at time t evolves according to the original arrival rate $\lambda_s(t)$ and net change $\eta_s(t)$ described in Section II-B, along with a modification $\gamma(s, \lambda_s(t), p_s(t))$ due to customers taking price incentives and another, $\Delta f_s(t)$ due to trucks adding or taking away bicycles from the stations. $\tilde{N}_s$ denotes the set of stations having s as one of their nearest neighbors, i.e.,

$$\begin{aligned}\gamma(s, \lambda_s(t), p_s(t)) = &\sum_{\tilde{n} \in \tilde{N}_s} \bar{\pi}(\tilde{n}, s, p_{\tilde{n}}(t)) \cdot \lambda_{\tilde{n}}(t) \\ &- \sum_{n \in N_s} \bar{\pi}(s, n, p_s(t)) \cdot \lambda_s(t)\end{aligned} \tag{17}$$

$$f_s(t+1) = f_s(t) + \eta_s(t) + \gamma(s, \lambda(t), p(t)) + \Delta f_s(t). \tag{18}$$

Note that $\sum_{s \in S} \gamma(s, \lambda_s(t), p_s(t)) = 0$ since the total number of bicycles in the system must be constant. In addition, the controller assumes that customers who take an incentive go from their originally intended destination to the new one within the same time step.

Under the assumption of a linear customer response to prices, the expected total incentive payouts are a quadratic function of the prices (the amount of customers taking an incentive multiplied by the payout for the incentive). The cost for suboptimal system states aims to keep every station close to its optimum fill level (based on the utility function). The resulting MPC problem is a QP and can be stated as

$$\min_{p(t)} \sum_{t=1}^{T_{\text{price}}} \sum_{s}^{S} Q_s(t) \tilde{f}_s(t)^2 + \sum_{t=0}^{T_{\text{price}}-1} \sum_{s}^{S} R_s(t) p_s(t)^2 \tag{19a}$$

such that

$$\tilde{f}_s(t) = f_s(t) - \frac{1}{2}\left(\underline{f}_t^s + \overline{f}_t^s\right), \qquad \forall s \in S, \forall t \tag{19b}$$

$$\begin{aligned}f_s(t+1) = &f_s(t) + \eta_s(t) + \Delta f_s(t) \\ &+ \sum_{\tilde{n} \in \tilde{N}_s} \left(\tilde{\pi}_{\tilde{n},s}^{\top} p_{\tilde{n}}(t)\right) \lambda_{\tilde{n}}(t) \\ &- \sum_{n \in N_s} \left(\tilde{\pi}_{s,n}^{\top} p_s(t)\right) \lambda_s(t), \quad \forall s \in S, \forall t\end{aligned} \tag{19c}$$

$$\sum_{n \in N_s} \tilde{\pi}_{s,n}^{\top} p_s(t) \leq 1, \quad \forall s \in S, \forall t \tag{19d}$$

$$0 \leq p_{s,n}(t) \leq p_{\max}, \quad \forall s \in S, \forall t, \; n \in \{1, \ldots, N\}. \tag{19e}$$

The cost weights $Q_s(t)$ and $R_s(t)$ in cost function (19a) are used to penalize deviation $\tilde{f}(t)$ from the optimal state $1/2(\underline{f}_t^s + \overline{f}_t^s)$, and the cost caused by the incentives payout, respectively, i.e.,

$$Q_s(t) = 1/\left(\overline{f}_t^s - \underline{f}_t^s\right) \tag{20}$$

$$R_s(t) = \alpha \sum_{n \in N_s} \tilde{\pi}_{s,n}^\top \lambda_s(t). \tag{21}$$

A lower value of weighting factor α leads to a lower relative penalty for paid incentives, likely leading to higher price incentives applied.

Equation (19b) transforms the number of bicycles at each station to a quantity measured relative to the "best" fill level, i.e., the middle of the station's utility plateau. The predicted system states within the horizon are defined by (19c). It includes the expected arrival and departure rates, manual redistribution with trucks and the linearized model of customer behavior. Equation (19d) limits the payouts such that no more than 100% of arriving customers take an incentive to one of the neighbors, and (19e) ensures that payouts are, at most, $p_{\max}$.

VI. Simulation

A. Simulation Setting

Based on the assumptions and the system model developed in Section II, a Monte Carlo simulation was used to compare the two approaches for bicycle redistribution. It is important to note that although simplified models of the system were used to choose truck actions and incentive prices, we used the *full* model derived from historical data, as described in Section II, to simulate the actual behavior of customers. We simulated a sequence of both weekdays and weekend days, bearing in mind that demand patterns significantly differ between the two. Every simulation run consisted of a 24 h burn-in period, in order to reduce the dependence of our results on the initial condition supplied, followed by three consecutive days for which the statistics gathered are presented below. In accordance with [19], redistribution with trucks was part of the simulation only between the hours of 8 A.M. and 10 P.M.

B. Simulation Results

The resulting service level was computed as

$$\text{Service level} = \frac{\text{Potential customers} - \text{No - service events}}{\text{Potential customers}}. \tag{22}$$

When simulating three consecutive weekdays, about 49 800 potential customers were generated on average, and for three consecutive weekend days (e.g., a bank holiday weekend), about 29 900 potential customers were generated. The number of total no-service events gives the sum of customers who could not rent a bicycle at an empty station and customers who wanted to return their bicycle at a full station. Fig. 9 shows the effect on service level of varying the number of redistribution trucks used and the volume of price incentive payouts. The latter was varied by choice of parameter α in (21).

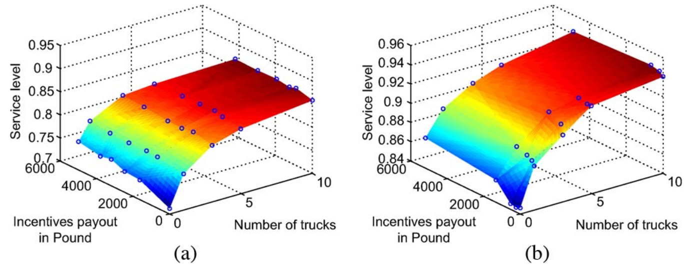

Fig. 9. (a) Service level for weekdays, as a function of number of trucks and total payouts. (b) Service level for weekend days, as a function of number of trucks and total payouts.

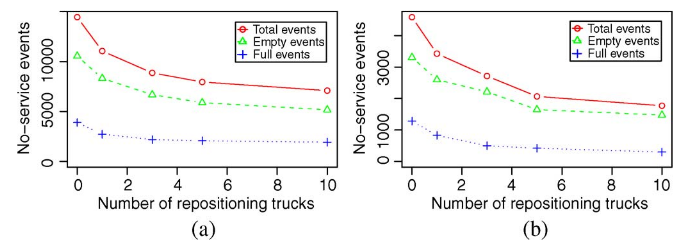

Fig. 10. (a) No-service events for different numbers of redistribution trucks (no incentives) for three consecutive weekdays. (b) No-service events for different numbers of redistribution trucks (no incentives) for three consecutive weekend days.

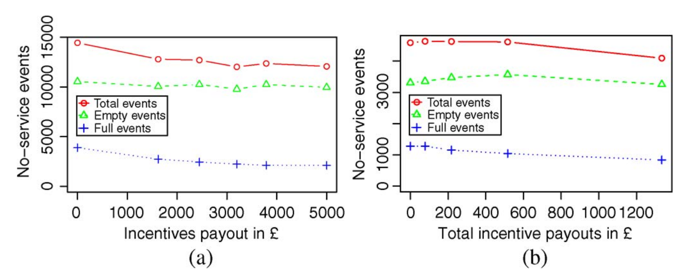

Fig. 11. (a) No-service events during simulations for weekdays with 0 redistribution trucks. Over the course of 72 h ca. 49 800 potential customers arrive. (b) No-service events during simulations of weekend days with 0 redistribution trucks. Over the course of 72 h ca. 29 900 potential customers arrive.

As expected, adding more trucks as well as paying out more incentives both had a positive effect on the service level. However, with increasing service level, adding trucks and incentive payouts become less efficient. Figs. 10 and 11 show the split of no-service events into "empty events" (where customers wanting to rent a bicycle arrive at an empty station) and "full events" (where customers wanting to return a bicycle arrive at a full station). Since the number of full events is considerably lower than the number of empty events, it seems plausible that adding more bicycles could have a positive effect on the service rate.

VII. Conclusion

This paper has considered how a PBS scheme could be managed using a combination of intelligently routed redistribution trucks and price incentives for customers. The truck routes and price incentives were computed using model-based receding horizon optimization principles, which took account of expected future customer behavior. As the number of trucks was increased, diminishing gains to service level were reported for added trucks and customer incentive payouts. Customer payments were shown to be a means of reducing service shortfalls, particularly when few redistribution trucks were in operation.

Our results suggest that price incentives are viable for redistribution bicycles in a PBS when the commuting rush hour is less prominent. For the London PBS, price incentives alone were shown to be enough to keep the service level above 87% on weekends without the use of staff for redistribution. On weekdays, however, when many customers use the PBS to commute to work, price incentives alone are not sufficient to lift the service level substantially.

The price control algorithm could be developed further in several ways. First, a field trial could be used to improve the accuracy of the customer decision model upon which our controller is based. This would reveal the range of price elasticities customers exhibit and also indicate to what extent customer responses to prices are irrational. Second, some simplifying assumptions (for example, deterministic customer arrival and linearized customer reaction to incentives) could be replaced by more detailed models. The main difficulty in designing a more detailed optimization-based control approach lies in considering the behavior of stochastically generated customers without incurring prohibitive computational costs.

REFERENCES

[1] S. Shaheen, S. Guzman, and H. Zhang, "Bikesharing in Europe, the Americas, and Asia," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2143, no. 1, pp. 159–167, 2010.

[2] P. DeMaio and L. MetroBike, "Bike-sharing: History, impacts, models of provision, and future," *J. Public Transp.*, vol. 12, no. 4, pp. 41–56, 2009.

[3] S. Erlanger, "A new fashion catches on in Paris: Cheap bicycle rentals," The New York Times, Jul. 13, 2008.

[4] P. Midgley, "Bicycle-sharing schemes: Enhancing sustainable mobility in urban areas," United Nations, Dept. Econ. Social Affairs, New York, NY, USA, Tech. Rep., 2011.

[5] S. Wang, J. Zhang, L. Liu, and Z. Duan, "Bike-sharing-a new public transportation mode: State of the practice & prospects," in *Proc. IEEE ICEMMS*, 2010, pp. 222–225.

[6] P. Vogel and D. Mattfeld, "Modeling of repositioning activities in bike-sharing systems," in *Proc. WCTR*, Lisbon, Portugal, 2010, pp. 1–13.

[7] *Optimising Bike Sharing in European Cities—A Handbook*, Optimising Bike Sharing in European Cities, Germany, 2011.

[8] A. Stannard, Building the best public cycle hire scheme in the world. Accessed in August 2012, 2011. [Online]. Available: http://www.scribd.com/doc/104443894

[9] P. Mackie, S. Jara-Dýaz, and A. Fowkes, "The value of travel time savings in evaluation," *Transp. Res. E, Logist. Transp. Rev.*, vol. 37, no. 2/3, pp. 91–106, Apr.–Jul. 2001.

[10] S. Hess, M. Bierlaire, and J. W. Polak, "Estimation of value of travel-time savings using mixed logit models," *Transp. Res. A, Policy Pract.*, vol. 39, no. 2/3, pp. 221–236, Feb./Mar. 2005.

[11] T. Raviv and O. Kolka, "Optimal inventory management of a bike-sharing station," *IIE Trans.*, vol. 45, no. 10, pp. 1077–1093, Oct. 2013.

[12] M. Dror, D. Fortin, and C. Roucairol, "Redistribution of self-service electric cars: A case of pickup and delivery," INRIA, Rocquencourt, France, Tech. Rep., 1998.

[13] C. Duron, M. Parent, and J. Proth, "Analysis of the balancing process in a pool of self-service cars," INRIA, Rocquencourt, France, Tech. Rep., 2000.

[14] M. Barth, M. Todd, and L. Xue, "User-based vehicle relocation techniques for multiple-station shared-use vehicle systems," presented at the Transportation Research Board 80th Annual Meeting, Washington, DC, USA, 2004, Paper 04-4161.

[15] A. Kek, R. Cheu, Q. Meng, and C. Fung, "A decision support system for vehicle relocation operations in carsharing systems," *Transp. Res. E, Logist. Transp. Rev.*, vol. 45, no. 1, pp. 149–158, Jan. 2009.

[16] M. Benchimol, P. Benchimol, B. Chappert, A. De La Taille, F. Laroche, F. Meunier, and L. Robinet, "Balancing the stations of a self service "bike hire" system," *RAIRO–Oper. Res.*, vol. 45, no. 1, pp. 37–61, Jan. 2011.

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

[18] J. Schuijbroek, R. Hampshire, and W.-J. van Hoeve, Inventory rebalancing and vehicle routing in bike sharing systems, Tepper School of Business, Pittsburgh, PA, USA, Working Paper. [Online]. Available: http://repository.cmu.edu/tepper/1491

[19] *Freedom for Information Request: Barclays Bicycle Redistribution*, Transport for London, London, U.K., 2012.

[20] R. Nair and E. Miller-Hooks, "Fleet management for vehicle sharing operations," *Transp. Sci.*, vol. 45, no. 4, pp. 524–540, 2011.

[21] C. Contardo, C. Morency, and L. Rousseau, "Balancing a dynamic public bike-sharing system," CIRRELT, Montréal, QC, Canada, Tech. Rep., 2012.

[22] R. Nair, E. Miller-Hooks, R. C. Hampshire, and A. Buši, "Large-scale vehicle sharing systems: Analysis of Vélib," *Int. J. Sustainable Transp.*, vol. 7, no. 1, pp. 85–106, Jan. 2013.

[23] G. Berbeglia, J. Cordeau, and G. Laporte, "Dynamic pickup and delivery problems," *Eur. J. Oper. Res.*, vol. 202, no. 1, pp. 8–15, Apr. 2010.

[24] B. Lowere, "The harpy speech recognition system," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, USA, 1976.

[25] P. S. Ow and T. E. Morton, "Filtered beam search in scheduling," *Int. J. Prod. Res.*, vol. 26, no. 1, pp. 35–62, Jan. 1988.

[26] C. Fricker and N. Gast, "Incentives and regulations in bike-sharing systems with stations of finite capacity," *Arxiv preprint arXiv:1201.1178*, 2012.

[27] J. M. Maciejowski, *Predictive Control : With Constraints*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.

Julius Pfrommer received the M.S.-equivalent degree in industrial engineering and economics from Grenoble Institute of Technology, Grenoble, France, in 2012 and the Dipl. Wirtsch.-Ing. degree from Karlsruhe Institute of Technology, Karlsruhe, Germany, in 2013. He is currently working toward the Ph.D. degree in computer science at the Karlsruhe Institute of Technology.

For his diploma thesis, he was with the Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH) Zürich, Zürich, Switzerland. His research interests include logistics and automation as well as planning and control in highly concurrent discrete-event systems.

Joseph Warrington (M'10) received the B.A. and M.Eng. degrees in mechanical engineering from University of Cambridge, Cambridge, U.K., in 2008. He is currently working toward the Ph.D. degree in the Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH) Zürich, Zürich, Switzerland.

In 2007, he held a SURF Fellowship at the California Institute of Technology, Pasadena, CA, USA. From 2008 to 2009, he was a Control Design Engineer with Wind Technologies Ltd., Cambridge. His research interests include distributed and robust control and optimization, with applications in power systems and market mechanism design.

Georg Schildbach (M'11) received the Dipl.-Ing. degree in mechanics and the Dipl. Wirtsch.-Ing. degree in industrial engineering from Technical University of Darmstadt, Darmstadt, Germany. He is currently working toward the Ph.D. degree in the Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH) Zürich, Zürich, Switzerland.

From 2004 to 2005, he was a Visiting Student in mechanical engineering with University of California at Berkeley, Berkeley, CA, USA, with a fellowship from the German Academic Exchange Association (DAAD). From 2008 to 2009, he was with the London branch of Bank of America Merrill Lynch. His research interests include methods of optimal control and control of uncertain systems, as well as their practical applications to systems in engineering and finance.

Manfred Morari (F'05) received the Diploma degree in chemical engineering from Swiss Federal Institute of Technology (ETH) Zürich, Zürich, Switzerland, and the Ph.D. degree in chemical engineering from University of Minnesota, Minneapolis, MN, USA.

In 1994, he was appointed Head of the Automatic Control Laboratory, ETH Zürich. Prior to this, he was the McCollum–Corcoran Professor of chemical engineering and the Executive Officer for Control and Dynamical Systems with California Institute of Technology, Pasadena, CA, USA. His research interests include hybrid systems and control of biomedical systems.

Dr. Morari has held appointments with Exxon and Imperial Chemical Industries PLC and serves on the technical advisory board of several major corporations. In recognition of his research contributions, he has received numerous awards, including the Hendrik W. Bode Lecture Prize of the IEEE Control Systems Society, the Donald P. Eckman Award and the John R. Ragazzini Award of the Automatic Control Council, the Allan P. Colburn Award and the Professional Progress Award of the American Institute of Chemical Engineers, the Curtis W. McGraw Research Award of the American Society for Engineering Education, and the IEEE Control Systems Field Award. He is a Fellow of the International Federation of Automatic Control; was conferred the Doctor Honoris Causa by Babes-Bolyai University, Cluj-Napoca, Romania; and was elected to the National Academy of Engineering.