On-line Dynamic Station Redeployments in Bike-Sharing Systems

Carlo Manna(✉)

Insight Research Centre for Data Analytics, University College Cork, Cork, Ireland
carlo.manna@insight-centre.org

Abstract. Bike-sharing has seen great development during recent years, both in Europe and globally. However, these systems are far from perfect. The uncertainty of the customer demand often leads to an unbalanced distribution of bicycles over the time and space (congestion and/or starvation), resulting both in a loss of customers and a poor customer experience. In order to improve those aspects, we propose a dynamic bike-sharing system, which combines the standard fixed base stations with movable stations (using trucks), which will able to be dynamically re-allocated according to the upcoming forecasted customer demand during the day in real-time. The purpose of this paper is to investigate whether using moveable stations in designing the bike-sharing system has a significant positive effect on the system performance. To that end, we contribute an on-line stochastic optimization formulation to address the redeployment of the moveable stations during the day, to better match the upcoming customer demand. Finally, we demonstrate the utility of our approach with numerical experiments using data provided by bike-sharing companies.

Keywords: On-line combinatorial optimization · Uncertainty · Smart cities

1 Introduction

Bike-sharing systems (BSS) are in place in several cities in the world, and are an increasingly important support for multi-modal transport systems [1]. BSS are widely adopted with 747 active systems, a fleet of over 772,000 bicycles and 235 systems in planning or under construction [2]. A BSS typically has a number of base stations scattered throughout a city. At the beginning of the day, each station is stocked with a pre-determined number of bikes. Users with a membership card can pick up and return bikes from any designated station, each of which has a finite number of docks. At the end of the work day, trucks are used to move bikes around so as to return to some pre-determined configuration at the beginning of the day. Due to the individual movement of customers according to their needs, there is often congestion (more than required) or starvation (fewer than required) of bikes at certain base stations. According to CapitalBikeShare Company [3], in a city like Washington, at a minimum, there are around 100

G. Adorni et al. (Eds.): AI*IA 2016, LNAI 10037, pp. 13–25, 2016.
DOI: 10.1007/978-3-319-49130-1_2

cases of empty stations and 100 cases of full stations per day and at a maximum there are about 750 cases of empty stations and 330 cases of full stations per day. As demonstrated in [4] this can result in a significant loss of customer activity. Such loss in demand can have two undesirable outcomes: (a) loss in revenue; (b) increase in carbon emissions, as people resort to less environmentally-friendly modes of transport. To address such a problem, the majority of the proposed solutions aim to find more efficient methods for dynamic rebalancing of the number of bicycles on the base stations taking into account the uncertainty of the customer demand or predicting the customer demand at each station [5,6]. Basically, by using trucks a number of bicycles are transferred from one station to another to accomplish the upcoming demand. This operation takes place ones a day (or more in some specific situations).

The aim of this paper is to study a totally different approach, in which the fixed base stations are augmented by a number of movable stations (using trucks) with the dual purpose of both: (1) dynamically adding/re-allocating dock stations in city areas to match customer demand in real-time and, (2) a dynamic re-balancing of the number of bicycles in some particular fixed station where a redeployment of a docking station is unnecessary.

Particularly, we consider a problem in which the day time is partitioned in time intervals of equal length. We suppose that for each of those time periods, the probability distributions of the travel demands between different locations are known. At start of the day, we compute the best possible locations for the dock stations in each time period, taking into account the stochastic nature of the demand, with the aim to maximize the number of customers. This is an on-line problem. That means, although the solution is a sequence of decisions for each time period, only the immediate decision is actually taken (i.e. the station allocations for the incoming time period). While this time period, a new computation is performed, using more updated travel demand predictions, and a new decision for the next period is taken from the re-computed decision sequence. This carry on until the termination of the time horizon.

The main advantage of such a system is that the bike stations configuration is not fixed, but it can change adaptively with the travel demand day by day and, in each day, it can change during a number of time periods, in the respect of specific time constraints (i.e. a new configuration must be computed in advance with enough time to allow the repositioning of the stations). Finally, updating the decisions each time periods, allow reducing the uncertainty in the predictions. This because the majority of the travel demand prediction techniques are based on auto-regressive models, which make use of the most recent known data to predict future outcomes [7].

The key distinction from existing research on bike sharing is that we consider the dynamic redeployment of bicycle stations (instead of just rebalancing the number of bicycles in the existing stations). This approach from one side does not exclude the possibility to rebalancing as in the existing research. However this extends such a research to a novel approach to the BSS by which is possible to dynamically change the configuration of the dock stations in real-time to maximize the potential customer demand. In doing this, while numerous models exist in literature, in this paper the potential customer demand is based on a primary concept: the *distance decay*. That is "the attenuation of a pattern or a process with distance" [8]. In other words, people are less willing to use a facility allocated too far from them. This is a focal concept in a variety of modelling contexts, such as transporation, migration and location theory [9].

Specifically, our key contributions are as follows:

- A mixed and linear programming (MILP) formulation to maximize the expected demand assigned to the moveable bicycle stations while simultaneously address the distance decay;
- An on-line stochastic optimization formulation to address the dock stations re-allocation problem during the day to accomplish the estimated customer demand in real-time;
- A potentially novel approach to the design of a bike-sharing system, along with numerical results for an initial investigation.

Extensive numerical simulations using datasets of two bike-sharing companies, namely Capital Bikeshare (Washington, DC) [10] and Hubway (Boston, MA) [11] show that the proposed approach can improve the customer usage of the bike-sharing system and that the computation time is reasonably fast to be used in real-time.

2 Related Work

Although bike sharing systems are relatively new, they have been studied extensively in the literature. For this reason, we only focus on threads of research that are of relevance to this paper. However, on the best of our knowledge, there is no any previous work for on-line stochastic redeployment of moveable stations in bike-sharing.

The first thread of research focus on the bicycles rebalancing between the stations. Particularly, [12–14] focus on the problem of finding routes at the end of the day for a set of carriers to achieve the desired configuration of bikes across the base stations. They have provided scalable exact and approximate approaches to this problem by either abstracting base stations into mega stations or by employing insights from inventory management or by using variable neighbourhood search based heuristics. Those works assume there is only one fixed redeployment of bikes that happens at the end of the day. In contrast, [15] predict the stochastic demand from user trip data of Singapore metro system using poisson distribution and provide an optimization model that suggests the best location of the stations and a dynamic bicycles redeployment for the model to minimize the number of unsatisfied customers. However, they assume that redeployment of bikes from one station to another is always possible without considering the routing of carriers, which is a major cost driver for the bike-sharing company. In [16] they overcome this problem, developing a mixed integer programming formulation which includes the carrier routes into the optimization model. Finally, other relevant works have been proposed in [17] to deal with unmet demand in rush hours. They provide a myopic redeployment policy by considering the current demand. They employed Dantzig-Wolfe and Benders decomposition techniques to make the decision problem faster. [18] also provides a myopic online decisions based on assessment of demand for the next 30 min.

The second thread of research is complementary to the work presented in this paper is on demand prediction and analysis. [19] provides a service level analysis of the BSS using a dualbounded joint-chance constraints where they predict the near future demands for a short period of time. Finally, in [20], the BSS is represented as a dual markovian waiting system to predict the actual demand.

As we already highlighted, all the aforementioned works differ from the one proposed in this paper as we consider a dynamic re-allocation of a certain number of bicycle stations during the day. This lead to a formulation of the problem which is different from all the previous provided in literature.

3 Problem Description

In this section we formally describe the bike-sharing system with dynamic redeployment. It is compactly described using the following tuple: $\langle T, A, K, \mathbf{S}, \mathbf{P}, \mathbf{D}, \mathbf{X}, \hat{\mathbf{X}}, \delta, l \rangle$, where A represents the set of areas for which the demand has to be covered, K represents the set of possible locations for the dock stations with $K \subseteq A$, T is the time horizon. $\mathbf{S}$ is a binary vector representing the totality of decisions on the allocated stations, with S_k^t denotes the decision on whether or not enabling a dock station in $k \in K$ at time $t \in T$. Furthermore, $\mathbf{P}$ is a binary vector representing the distribution of the service coverage between areas and dock stations. In particular, $P_{a,k}^t$ denotes whether or not the area $a \in A$ is served by the station in $k \in K$, at time $t \in T$. $\mathbf{D}$ is a vector of the distances (or travel time) between all areas and station locations, with $d_{a,a'}$ ($d_{a,k}$), the distance/travel time between the area a and a' (between the area a and the station in k). $\mathbf{X}$ is a vector of the probability distribution of the potential customer demand between different areas, with $x_{a,a'}^t$ denoting the potential travel demand from the area a to the area a' (with $(a, a') \in A$), at time $t \in T$. The potential customer demand at time t denotes the maximum number of possible customers to be served. This is expressed in form of probability distribution. Similarly, $\hat{\mathbf{X}}$ is a vector of the *expected* covered customer demand between different areas, with $\hat{x}_{a,a'}^t$ denoting the expected travel demand from the area a to the area a' (with $(a, a') \in A$), at time $t \in T$. Finally $\delta \in [0, 1]$ is a distance decay parameter [8], through which we take into account the aforementioned *distance decay* concept. Finally l is the maximum number of moveable stations. The expected covered customer demand denotes the number of customers which is expected to use the bike-sharing system. Hence, given a potential demand $x_{a,a'}^t$, it holds $\hat{x}_{a,a'}^t \leqslant x_{a,a'}^t$ for each $a \in A$ and $t \in T$. $\hat{\mathbf{X}}$ depends on various factors, such as the total distance/travel time between the customer location and the dock station, and from the distance decay parameter δ.

Given the potential customer demand $\mathbf{X}$ instantiated from a known probability distribution, at each time step $t \in T$, the goal is to maximize a profit function $W(\cdot)$ of the overall expected covered demand $\hat{\mathbf{X}}$ over the total time horizon $\mathbf{T}$, subject on some problem specific constraints $C(\cdot)$ (both are specified in the following paragraph). This is achieved through finding the best possible sequence of decisions $\langle S^1, \ldots, S^T \rangle$, concern the locations of the dock stations over the entire time horizon $\mathbf{T}$.

Finally, as reported in the relevant literature [21], in defining the solution approach, we have made the following assumptions:

1. the expected covered demand decrease with the total distance travelled on foot (i.e. the distance between the starting point and the pick up station, and the distance between the return station and the arrival point), given a predefined decay parameter δ.
2. each area can be served by only one station.

The assumption (1) states that the distance travelled on bicycle between the stations does not have any negative impact on the demand. This because the potential demand $\mathbf{X}$ is already estimated considering this factor. Conversely, the distance travelled on foot does have a negative impact on the service usage. The assumption (2) states that the totality of the demand in an area $a \in A$, considering a either starting or arrival point, can be served by only one station located at $k \in K$. This hypothesis is one of the most used in many facility location models, which assumes that the customer always use the closest facility.

4 Solution Approach

We propose an on-line stochastic combinatorial optimization approach for the system described in the last section. Consequently, we first outline the solution approach for the deterministic case (in which we consider no uncertainty into the customer demand) and then, we remove such hypothesis and extend this formulation to the on-line stochastic case.

4.1 Deterministic Case

The deterministic model is based on a single scenario. That means considering $\mathbf{X}$ as a single per-determined scenario values for the potential customer demand between different areas. The solution is expressed as a sequence of decisions to be taken at each time step. We do not need to recompute the solution at each time step, since the demand remain the same in each period. Hence the problem does not have any on-line feature.

To address the deterministic case we propose a MILP formulation with the following decision variables:

$$S_k^t = \begin{cases} 1 \text{ if a station is located in } k \text{ at time } t \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

$$P_{ak}^t = \begin{cases} 1 \text{ if the area } a \text{ is served by station in } k \text{ at time } t \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

The proposed MILP formulation is defined as follow:

$$\max \sum_{a,t} \hat{x}_{a,a'}^t \tag{3}$$

s.t.:

$$\hat{x}_{a,a'}^t = x_{a,a'}^t \left[1 - \delta \left(\sum_k d_{a,k} P_{a,k}^t + \sum_k d_{a',k'} P_{a',k'}^t \right) \right] \quad \forall\, a, t \tag{4}$$

$$\sum_k P_{a,k}^t = 1 \quad \forall\, a, t \tag{5}$$

$$\hat{x}_{a,a'}^t \geqslant 0 \quad \forall\, a, t \tag{6}$$

$$P_{a,k}^t - S_k^t \leqslant 0 \quad \forall\, a, k, t \tag{7}$$

$$2 \leqslant \sum_k S_k^t \leqslant l \quad \forall\, t \tag{8}$$

$$S_k^t \in \{0, 1\} \quad \forall\, k, t \tag{9}$$

$$P_{a,k}^t \in \{0, 1\} \quad \forall\, a, k, t \tag{10}$$

The objective (3) maximizes the total expected covered demand over the entire time horizon. Constraint (4) specifies the interaction between the expected covered demand with the potential demand and the station locations. The constraint (4) explicit the condition 1 reported in the previous section, which assumes the expected covered demand decreases linearly with the total distance travelled on foot. Further, beyond a certain distance (which scales with δ), the expected covered demand becomes 0. This according to a primary concept in the central place theory, namely *coverage range*, which denotes the maximum distance (or travel time) a user is willing to overcome to utilize a service facility

[21]. Constraint (5) specifies that the demand may only be assigned to one station at each time step, as mentioned in the condition 2 in the previous section. Constraint (6) ensures the expected covered demand be a non negative value. Constraint (7) limits assignment to open stations only, while constraint (8) specifies that the number of open stations, at each time step, must be in a predefined range (from a minimum of 2 up to l). Finally, constraints (9) and (10) are integrality conditions.

Among many possible models describing the relationships between expected covered demand and distance, we have chosen the one reported in (4) as the linear nature of such relation allows us to propose a linear formulation for the optimization problem.

4.2 On Line Stochastic Case

In this section we remove the deterministic hypothesis and introduce the on-line stochastic optimization model. We use a similar approach as in [22], with some more additional ideas to address the peculiarity of the BSS system. The general approach is to evaluate many scenarios draw from the probability distribution $\mathbf{X}$ of the potential customer demand. Then, each of those scenarios represents a deterministic problem as the one detailed in the previous section. Hence, we solve each scenario deterministically, using the MILP formulation reported in (1)–(10). Each solution represents the best dock stations configuration to maximize the (3), over the entire time horizon for each scenario. Once we have the best solution for each scenario, we need to combine all these solutions in order to find the best decision over all scenarios. In other words, we need to choose only one solution among the totality of the all computed solutions.

To that end, we propose the following heuristic: let $\mathbf{S}_i$ be the solution for the scenario i with $i = 1 \ldots, N$, over the entire time horizon, with S_i^t the dock station configuration at time t, and $S_i^t \subseteq \mathbf{S}_i$ for each i and $t = 1 \ldots, T$. Finally, s_{ik}^t denotes the decision on whether or not enabling a dock station in $k \in K$ at time $t \in T$ for the scenario i. s_{ik}^t represents a single element of the station configuration S_i^t. We define $V(t, k)$ a matrix of the location $k \in K$ and the time t, with t rows and k columns. The value of each cell $v(t, k)$ is a non negative integer.

We initialize each element of $v(t, k) = 0$. During the deterministic optimization step, for each time that is $s_{ik}^t = 1$, we increment the corresponding element $v(t, k)$ such that $v(t, k) = v(t, k) + 1$

Put in other words, we select the time t at which a location k has been chosen as dock station, and included in a deterministic solution $\mathbf{S}_i$ for the scenario i and increment those elements $v(t, k)$ by 1. We continue to update $V(t, k)$ after optimizing each sampled scenario i.

At the end of the sampling/optimization step, the final value of the all elements $v(t, k)$ denote the total number of times those choices have been selected by the totality of the solutions.

Then, we can evaluate each deterministic solution $\mathbf{S}_i$ individually by means of $V(t, k)$.

Algorithm 1. On line stochastic optimization procedure

1: **for** t from 1 to T **do**
2: $\quad 0 \leftarrow V(k,t)$
3: $\quad$ **for** i from 1 to N **do**
4: $\quad\quad \omega_i \leftarrow$ **sample**($\mathbf{X}$)
5: $\quad\quad S_i \leftarrow$ **solve**(ω_i)
6: $\quad\quad$ **store** S_i *in* Γ_t
7: $\quad\quad$ **for all** (t,k): $s_{ik}^t = 1$ **do**
8: $\quad\quad\quad v(t,k) \leftarrow v(t,k) + 1$
9: $\quad S_{best} \leftarrow \text{argmax}_{S_i} \quad \sum_{(t,k):s_{ik}^t=1} v(t,k)$

We choose as best solution $\mathbf{S}_{best}$ among all the plans $\mathbf{S}_i$ as follow:

$$S_{best} = \underset{S_i}{\text{argmax}} \sum_{(t,k):s_{ik}^t=1} v(t,k) \tag{11}$$

The on-line stochastic optimization procedure is reported in the box Algorithm 1. At each time step $t \in H$ (line 1), we initialize the matrix $V(t,k)$ (line 2). Then, we sample N different scenarios ω_i from the probability distribution $\mathbf{X}$ (line 4) and solve each of those deterministically using the MILP formulation (1)–(10) (line 5) finding the solution $\mathbf{S}_i$ and finally storing it in Γ_t. At the end of each deterministic optimization procedure, we update each element $v(t,k)$ of the matrix $V(t,k)$ according to the decisions included in the solution $\mathbf{S}_i$ (lines 7-8). Finally (line 9), once each deterministic scenario has been solved, we choose the best solution S_{best} according to the Eq. (11).

Notice that, in the spirit of the proposed on-line formulation, at each time step t, only the decisions $S_{best}^{t+1} \subseteq \mathbf{S}_{best}$ for the immediate next time step $t+1$ is taken.

5 Numerical Experiments

We evaluate our approach with respect to run-time and demand growth for the BSS on synthetic data based on real world data sets. In particular, for each generated instance, we have compared a BSS schema with fixed dock stations with the one with moveable stations.

5.1 Data Preparation

For the numerical experiments we have used data sets provided by bike-sharing companies [10,11]. These data sets contain numerous attributes. For the purpose of this paper, among all the attributes provided in the data, we used the followings: (1) Customer trip records and (2) Geographical locations of base stations.

In order to generate different instances, we assumed the demand follows a poisson distribution as in the model provided by [15]. We divided each day in 4 time slots (from $8:00$ up to $24:00$). For each time slot, we learn the parameter λ that governs the poisson distribution from real data. Then, we assumed such probability distribution as the demand $\mathbf{X}$ for each instance. Notice that we also learn a single parameter λ for the entire day and assumed this as our demand $\mathbf{X}$ for the fixed dock stations case.

5.2 Experimental Setting

We generated 3 different instances, and for each of those we have considered different case studies. Each case study consider a different combination of the following parameters: *(i)* number of areas $\mathbf{A}$ for which the demand has to be covered, *(ii)* number of total dock-stations (considering the sum of the fixed and the moveable stations) and *(iii)* the maximum number of moveable stations l. Basically, in each case study we consider a scenario with a fixed number of settled dock-stations and a number of moveable stations during the entire time horizon. In this, the location for the settled dock-stations were chosen randomly among the station locations provided by the solution found running the deterministic optimization problem (1)–(10) for the corresponding scenario.

We start from a smaller case in which we have 12 areas and a total of 6 stations, up to the biggest case of 40 areas and 20 total dock stations.

We have computed the fixed dock stations case using the deterministic model (1–10), and the Algorithm 1 for the moveable stations case. In this case the time horizon has 4 time slots of 4 h each (from $8:00$ up to $24:00$). Notice that, as the stations are moveable in each time slot, these stations are not available to the customers while moving. Hence, we have considered such stations only available in the first 2 h of each time slot, while moving (and not available) in the remain time. For this reason, to make the simulation as realistic as possible, the computation time never has to exceed the 2 h limit for each run. Hence, for any case study the number of samples N in the Algorithm 1 was 1000, except for the biggest case (40 areas and 20 total dock stations), where to limit the run time we set $N = 500$.

Finally, throughout the experiments the distance decay parameter δ is 0.1, following a general case reported in similar works related to facility location planning [21].

5.3 Results

The algorithms were written using ILOG CPLEX Optimization Studio V12.6 incorporated within C++ code on a 2.70 GHz Intel Core i7 machine with 8 GB RAM on a Windows 7 operating system.

The results for each case study are reported throughout the Tables 1, 2, 3, 4, and 5. We can see that for each instance and for each case study, using a dynamic re-deployment of the dock-stations bring to a significant increasing in term of customer demand. Although the results seem do not show some general rule, we can notice that the more the system growth in complexity (more areas to be covered and more moveable stations), the more the performance of the system

Table 1. Case study with **12** areas and **6** dock stations in total and with a set of moveable stations from **1** to **3**. The stochastic optimization procedure has been computed using $\mathbf{N} = \mathbf{1000}$ samples.

Instance	Moveable stations	Demand growth	Runtime (sec)
1	1	12.15 %	320
2	1	9.49 %	378
3	1	12.16 %	316
1	2	12.75 %	371
2	2	8.51 %	372
3	2	13.23 %	368
1	3	12.81 %	398
2	3	10.13 %	436
3	3	14.06 %	377

Table 2. Case study with **18** areas and **9** dock stations in total, and with a set of moveable stations from **3** to **5**. The stochastic optimization procedure has been computed using $\mathbf{N} = \mathbf{1000}$ samples.

Instance	Moveable stations	Demand growth	Runtime (sec)
1	3	13.61 %	961
2	3	8.75 %	999
3	3	15.53 %	989
1	4	13.81 %	1012
2	4	8.96 %	1069
3	4	16.81 %	1021
1	5	14.94 %	986
2	5	8.90 %	1107
3	5	16.87 %	1023

increase in term of customer demand. In particular, comparing the results from Table 1 (smallest case) and the one in the Table 4 we can see an improvement of the demand of a minimum of 9.26 % (which refers to the instance 2, using 3 moveable stations for the case in Table 1 and 8 moveable stations for the case in Table 4). Also, we can see that the maximum improvement (12.36 %) in the customer demand is achieved for the instance 1, with 3 moveable stations in the case study reported in Table 1 and with 8 moveable stations in the case reported in Table 4. According to this, the improvement seems more related

Table 3. Case study with **24** areas and **12** dock stations in total, and with a set of moveable stations from **4** to **6**. The stochastic optimization procedure has been computed using $\mathbf{N} = \mathbf{1000}$ samples.

Instance	Moveable stations	Demand growth	Runtime (sec)
1	4	17.52 %	2483
2	4	13.20 %	2519
3	4	21.06 %	2403
1	5	17.36 %	2528
2	5	13.79 %	2521
3	5	20.88 %	2422
1	6	18.16 %	2555
2	6	13.94 %	2506
3	6	20.99 %	2467

Table 4. Case study with **30** areas and **15** dock stations in total, and with a set of moveable stations from **6** to **8**. The stochastic optimization procedure has been computed using $\mathbf{N} = \mathbf{1000}$ samples.

Instance	Moveable stations	Demand growth	Runtime (sec)
1	6	23.86 %	4433
2	6	19.36 %	4592
3	6	23.61 %	4391
1	7	24.76 %	4504
2	7	19.21 %	4669
3	7	24.12 %	4394
1	8	25.17 %	4473
2	8	19.39 %	4872
3	8	25.72 %	4512

to the implementation of the dynamic station re-deployment on larger systems then related with the number of moveable platforms. Finally, for the largest case reported in Table 5 the result confirm such tendency, although the improvement does not seems too significant. However, it is difficult to say from this results if this may depends on the few number of samples used by the Algorithm 1, or by different reasons.

Table 5. Case study with **40** areas and **20** dock stations in total, and with a set of moveable stations from **8** to **10**. The stochastic optimization procedure has been computed using $\mathbf{N = 500}$ samples.

Instance	Moveable stations	Demand growth	Runtime (sec)
1	8	25.27 %	6609
2	8	20.14 %	6846
3	8	25.94 %	6531
1	9	25.85 %	6742
2	9	20.19 %	6930
3	9	25.88 %	6557
1	10	25.94 %	6858
2	10	20.97 %	7029
3	10	25.90 %	6704

6 Conclusions

In this paper, we investigated the problem of on-line dynamic redeployment in bike-sharing systems. We proposed an on-line stochastic optimization formulation to address this problem. Then, we carried out numerical experiments using data provided by bike-sharing companies to show the effectiveness of our approach.

When we consider the on-line station redeployments, the results show a significant improvement in the customer demand which go from a minimum around 8 % up to a maximum of a 25 %. We also found that the more the system growth in complexity (more areas to be served and more stations) the more the improvement in customer demand becomes significant.

As future works there are many points to deal with. First of all, in order to generalize the results, we need to formulate models for which the relationship between the expected customer demand and the distance between the facilities is non linear. This will lead to a different optimization formulation. Further, we need to integrate into the proposed formulation, the dynamic rebalancing of the bicycles and eventually, the cost for the truck routings into the optimization formulation. Finally, we need to test such an approach on bigger instances (300 areas with $100 - 150$ stations in total) which can be significant for large cities such as Barcellona, Madrid or Dublin [23].

References

1. Shaheen, S., Guzman, S., Zhang, H.: Bikesharing in europe, the americas, and asia. Transp. Res. Rec. J. Transp. Res. Board **2143**, 159–167 (2010)
2. Meddin: The bike-sharing world map (2015). http://www.bikesharingworld.com/
3. Capitalbikeshare (2015). www.capitalbikeshare.com/home

4. Fricker, C., Gast, N.: Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. EURO J. Transp. Log. **5**(3), 261–291 (2016)
5. O'Mahony, E., Shmoys, D.: Data analysis and optimization for (citi)bike sharing. In: AAAI Conference on Artificial Intelligence (2015)
6. Boucher, X., Brissaud, D., Tran, T.D., Ovtracht, N., dArcier, B.F.: 7th industrial product-service systems conference - pss, industry transformation for sustainability and business modeling bike sharing system using built environment factors. Procedia CIRP **30**, 293–298 (2015)
7. Bowman, J., Ben-Akiva, M.: Activity-based disaggregate travel demand model system with activity schedules. Transportation research part a: policy and practice **35**(1), 1–28 (2001)
8. Gregory, D., Johnston, R., Pratt, G., Watts, M., Whatmore, S.: The Dictionary of Human Geography, 5th edn. Wiley, Hoboken (2011)
9. Fotheringham, A.S.: Spatial Interaction Models: Formulations and Applications. Springer, Heidelberg (1988)
10. Various 1 (2016). http://www.capitalbikeshare.com/system-data
11. Various 2 (2016). http://hubwaydatachallenge.org/trip-history-data
12. Raviv, T., Kolka, O.: Optimal inventory management of a bike-sharing station. IIE Trans. **45**(10), 1077–1093 (2013)
13. Raviv, T., Tzur, M., Forma, I.A.: Static repositioning in a bike-sharing system: models and solution approaches. EURO J. Transp. Log. **2**(3), 187–229 (2013)
14. Rainer-Harbach, M., Papazek, P., Hu, B., Raidl, G.R.: Balancing bicycle sharing systems: a variable neighborhood search approach. In: Middendorf, M., Blum, C. (eds.) EvoCOP 2013. LNCS, vol. 7832, pp. 121–132. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37198-1_11
15. Shu, J., Chou, M.C., Liu, Q., Teo, C.P., Wang, I.L.: Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems. Oper. Res. **61**(6), 1346–1359 (2013)
16. Ghosh, S., Varakantham, P., Adulyasak, Y., Jaillet, P.: Dynamic redeployment to counter congestion or starvation in vehicle sharing systems. In: International Conference on Automated Planning and Scheduling (2015)
17. Contardo, C., Morency, C., Rousseau, L.M.: Balancing a dynamic public bike-sharing system (2012)
18. Pfrommer, J., Warrington, J., Schildbach, G., Morari, M.: Dynamic vehicle redistribution and online price incentives in shared mobility systems. IEEE Trans. Intell. Transp. Syst. **15**(4), 1567–1578 (2014)
19. Nair, R., Miller-Hooks, E.: Fleet management for vehicle sharing operations. Transp. Sci. **45**(4), 524–540 (2011)
20. Leurent, F.: Modelling a vehicle-sharing station as a dual waiting system: stochastic framework and stationary analysis. 19 p., November 2012
21. Farhan, B., Murray, A.T.: Distance decay and coverage in facility location planning. Ann. Reg. Sci. **40**(2), 279–295 (2006)
22. Hentenryck, P.V., Bent, R.: Online Stochastic Combinatorial Optimization. The MIT Press, Cambridge (2006)
23. Dublin bike sharing map (2016). http://www.dublinbikes.ie/all-stations/station-map

http://www.springer.com/978-3-319-49129-5

AI*IA 2016 Advances in Artificial Intelligence
XVth International Conference of the Italian Association
for Artificial Intelligence, Genova, Italy, November 29 –
December 1, 2016, Proceedings
Adorni, G.; Cagnoni, S.; Gori, M.; Maratea, M. (Eds.)
2016, XIV, 554 p. 86 illus., Softcover
ISBN: 978-3-319-49129-5