Author's Accepted Manuscript

www.elsevier.com

Optimizing the level of service quality of a bike-sharing system

Ramon Alvarez-Valdes, Jose M. Belenguer, Enrique Benavent, Jose D. Bermudez, Facundo Muñoz, Enriqueta Vercher, Francisco Verdejo

PII: S0305-0483(15)00200-5
DOI: http://dx.doi.org/10.1016/j.omega.2015.09.007
Reference: OME1597

To appear in: *Omega*

Received date: 15 October 2014
Revised date: 22 September 2015
Accepted date: 22 September 2015

Cite this article as: Ramon Alvarez-Valdes, Jose M. Belenguer, Enrique Benavent, Jose D. Bermudez, Facundo Muñoz, Enriqueta Vercher and Francisco Verdejo, Optimizing the level of service quality of a bike-sharing system *Omega,* http://dx.doi.org/10.1016/j.omega.2015.09.007

Optimizing the level of service quality of a bike-sharing system

Ramon Alvarez-Valdes[a,*], Jose M. Belenguer[a], Enrique Benavent[a], Jose D. Bermudez[a], Facundo Muñoz[a], Enriqueta Vercher[a], Francisco Verdejo[a]

[a] *Department of Stastistics and Operations Research, University of Valencia, Doctor Moliner 50, 46100 Burjassot, Spain. Telephone: +34 96354308*

Abstract

Public bike-sharing programs have been deployed in hundreds of cities worldwide, improving mobility in a socially equitable and environmentally sustainable way. However, the quality of the service is drastically affected by imbalances in the distribution of bicycles among stations. We address this problem in two stages. First, we estimate the unsatisfied demand (lack of free lockers or lack of bicycles) at each station for a given time period in the future and for each possible number of bicycles at the beginning of the period. In a second stage, we use these estimates to guide our redistribution algorithms. Computational results using real data from the bike-sharing system in Palma de Mallorca (Spain) are reported.

Keywords: Bike-sharing systems, Demand forecasting, Routing, Heuristics
2010 MSC: 90B06, 90B90

1. Introduction

A public bike-sharing system consists of a set of stations scattered over the city and a set of bicycles available to the system users. A user can take a bicycle

* Corresponding author
Email addresses: `ramon.alvarez@uv.es` (Ramon Alvarez-Valdes), `jose.belenguer@uv.es` (Jose M. Belenguer), `enrique.benavent@uv.es` (Enrique Benavent), `bermudez@uv.es` (Jose D. Bermudez), `facundo.munoz@uv.es` (Facundo Muñoz), `enriqueta.vercher@uv.es` (Enriqueta Vercher), `francisco.verdejo@uv.es` (Francisco Verdejo)

at a station, use it for a short journey, and leave it at the same or any other station. Since the first system was established in Amsterdam in 1965, there has been a rapidly increasing number of cities providing their citizens with this type of service, which has many advantages of various kinds: it is an environmentally sustainable and socially equitable mode of transportation, it can be used as part of an intermodal public transport system, it reduces motorized traffic and therefore emissions of contaminants, and it promotes a healthier way of life. According to the consultancy company MetroBike LCC ([1]), in July 2014 721 cities had a public bike-sharing system, with a total of approximately 814000 bicycles, and 228 were planned or under construction. These systems range from less than one hundred bicycles in small towns to many thousands in cities like Paris (20600), Hangzhou (78000), or Wuhan (90000).

The most important factor for the success of a public bike-sharing system is its ability to satisfy the varying demands of the users. Underlying the random variations of everyday demands, there are patterns of demand that have to be identified and estimated and the system has to be planned and managed to maximize the level of customer satisfaction. Situations in which the user arrives at a station to take a bicycle and finds the station empty, and those in which he/she arrives at the station to leave the bicycle and the station is completely full, have to be avoided as far as possible. For the bike-sharing system to become a sensible alternative to other modes of transportation, it has to be reliable. The everyday users have to be confident that they will find bicycles to start their trips and available lockers to leave them when the trips are finished wherever and whenever they need them. This can be achieved in the three phases of design and operation. First, at a strategic level, the number of stations and their location and size have to be decided. Second, at a tactical level, the number of bicycles in the system has to be determined. Third, at an operational level, a bike-repositioning system has to be adopted for moving bicycles from stations with an excess to stations with a shortage in order to satisfy the demands forecast for the next periods.

Repositioning is done by means of light trucks based at one or several depots, that pick up bicycles from stations at which there are too many and move them to stations where there are too few. Sometimes there are bicycles at the depots, for instance those that were damaged and have been repaired, and these can also be used when constructing the repositioning routes.

There are two types of repositioning systems. In the static case, the system is considered closed, so the users do not interact with it, its initial state is considered known and fixed, and the aim of the repositioning is to get the system to a desired, predefined state. In the dynamic case, the repositioning system operates while the bike-sharing system is being used. Therefore, users are continuously taking and leaving bicycles at the stations, modifying their states. The dynamic repositioning system has to take these changes into account and adapt its decisions to the actual state of the stations.

In this study we focus on the static repositioning system at the operational level. In the city of Palma de Mallorca (Spain), whose bike-sharing system gave rise to this study, the system operates every day from 07:00 until 24:00. Every night, when the system closes, the states of the stations do not correspond to the desired states for the next morning. Therefore the trucks of the repositioning system move some bicycles between stations in order to leave the system as close as possible to the ideal state when it opens in the morning.

There are several features of the Palma system and of our approach that make this study different from previously published studies. First, the desired state of each station is not given but calculated from the system information. Most of the previous papers on the static repositioning problem consider the desired or target state of each station as a given constant, assuming implicitly that these targets are attainable. In our approach, the ideal state of each station is calculated from the database in which the system has recorded every single bicycle move. Using these data, we can estimate the demands (in both directions, taking and returning bicycles) for every time interval and compute the unsatisfied demand for each possible station state. Using these estimated unsatisfied demands and the number of available bicycles in the system we calculate the state of each station that minimizes the overall cost of unsatisfied demands.

If there are enough bicycles to attain for each station the state that minimizes its estimated unmet demand, that will be the solution. However, if the number of available bicycles is lower than the sum of these individual optimal states, the available bicycles have to be placed at the stations where they minimize the overall unsatisfied demands.

A second special characteristic concerns the design of the truck routes. The city is not split beforehand into as many zones as there are trucks available. In our proposal, all the trucks work jointly and the routes are constructed simultaneously, taking into account balance criteria such as the total distance or total time of each route. There is also flexibility about the initial and final points of the truck routes and about the number of bicycles a truck can carry when it starts its route.

A third distinctive characteristic of our proposal is the management of damaged bicycles. These damaged, out-of-service bicycles are detected by the system and we were asked to include their collection in the repositioning routes. In our system, when the trucks visit the stations these damaged bicycles are collected, whenever possible, and taken to the depot. In this way, no special collection routes for damaged bicycles are needed.

2. Previous work

There are a number of existing papers in the literature about the static case (Benchimol et al. [2], Chemla et al. [3], Dell'Amico et al. [4], Raviv et al. [5], Rainer-Harbach et al. [6], Raidl et al. [7], Papazek et al. [8], Schuijbroek et al. [9], Erdoğan et al. [10] and Ho and Szeto [11]), but very few on the dynamic case (Nair and Miller-Hooks [12], Contardo et al. [13] and Caggiani and Ottomanelli [14]). Although all of them deal with the problem of repositioning, the objectives, constraints, and solution techniques are different.

Among the different objectives considered, we find minimization of total traveling cost or time ([2], [3], [4]), total unmet demand ([13]), maximum tour length ([9]), the sum of travel time and holding cost ([10]), the weighted sum of total time of the routes, deviation from the targeted number of bicycles at each station, and number of moves between stations ([8], [15, 16]).

The characteristics of the problem considered in each study are also different. For example, in some of them only one vehicle is available ([2], [10], [11]). In others, a limit to the number of visits to the stations is imposed: only one ([4], [11]), or a maximum fixed number ([3]). In some cases, the perfect balance requirement is a hard constraint ([2], [3]) but in most of them the imbalances are penalized. A key question is how the target state for each station is computed. In most of the studies, this quantity is fixed, generally at half the station capacity, as in Rainer-Harbach et al. [6], Raidl et al. [7], and Papazek et al. [8]. Schuijbroek et al. [9] compute a lower and upper bound on the service level requirement of each station by using a queuing system and Raviv and Kolka [17] compute a measure of dissatisfaction for different replenishment periods, given an initial inventory, the station size, and stochastic demand patterns.

Several types of heuristics have been proposed. For instance, a 9.5 approximate algorithm ([2]), a cluster-first, route-second algorithm ([9]), tabu search ([3], [11]), variable neighborhood search ([6], [7]), PILOT/GRASP ([8]), ant colony and constraint programming ([15, 16]), and matheuristics ([18]). In three studies ([4], [5], [19]), MIP formulations and/or relaxations of the problem are solved. To our knowledge, only four exact methods have been proposed: one branch-and-prize ([13]) and three branch-and-cut algorithms ([3], [10], [19]).

3. Description of the problem and data analysis of the Palma system

A bike-sharing system is composed of a set of stations, from which the bicycles are taken and to which they are returned. The registered users can take a bicycle from any station at any time, use it, and return it to another station. Each city has its own rules of usage. In Palma de Mallorca, the first 30 minutes are free of charge and a small fee is paid for extra time. Each movement is recorded and stored in the system's database. The quality of the service can be measured as the amount of time a station remains empty or completely full, and either the withdrawal demands cannot be satisfied or the bicycles cannot be returned. Another alternative measure would be to count or estimate the number of unsatisfied demands of both types at each station. The repositioning system includes one or several depots to which damaged bicycles are brought to be repaired and from which repaired or new bicycles are incorporated into the system, and a set of vehicles that take bicycles from the depots or from the stations and carry them to other stations in order to improve the quality of the service.

The system can be represented by a complete graph $G = (V, E)$, where $V = \mathcal{S} \cup \mathcal{D} \cup \mathcal{V}$ contains a vertex for each depot, each station, and each vehicle, representing its initial location.

For each station $i \in \mathcal{S}$ (set of stations) we know:

- C_i, capacity (number of lockers),
- bs_i, current number of bicycles ready to be used,
- bs_i^{dam}, current number of damaged bicycles.

For each vehicle $l \in \mathcal{V}$ (set of repositioning vehicles):

- P_l, capacity (number of places),
- bv_l, number of bicycles in good condition on the vehicle,
- bv_l^{dam}, number of damaged bicycles on the vehicle.

For each depot $k \in \mathcal{D}$ (set of depots):

- bd_k, number of bicycles stored in the depot.

We also know:

- t_{ij}, travel time from location i to location j; $i, j \in \mathcal{S} \cup \mathcal{D} \cup \mathcal{V}$,
- t^{park}, parking time,

- t^{load}, time to load/unload one bicycle.

As of November 2013, in *bicipalma*, the bike-sharing system of Palma de Mallorca (Spain), there were 28 stations and over 200 bicycles. The capacity of each station varies from 10 to 30 bicycles. Figure 1 shows the distribution of the stations on the Palma map.

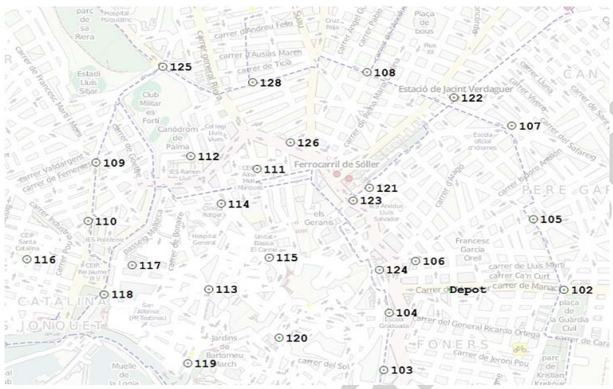


Figure 1: Map of Palma stations

The usage of each station is neither uniform nor correlated with its capacity. Figure 2 shows the capacity and the average number of movements on working days. The horizontal axis shows the stations' capacities (10, 15, 20, 30 lockers) and the vertical axis the average daily movements at each station. A large variability can be observed. Small stations with 10 lockers may have up to 60 movements, while large stations with 30 lockers may have far fewer, less than one move per locker. It could be argued that the system is not well designed, but in this study we are only concerned with attaining the maximum quality of service for a system in which the capacity of the stations is fixed. Rather than the total number of daily moves, we are interested in the distribution of withdrawals and returns over the course of the day, because the difference between the two types of moves will produce intervals in which the stations are empty or completely full.

Figure 3 shows the average number of withdrawals and returns for each

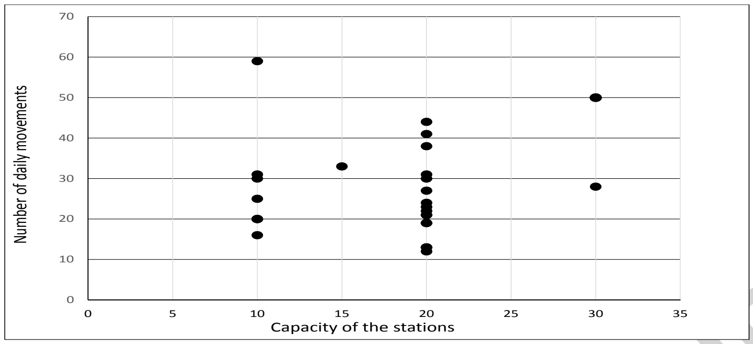


Figure 2: Capacity and average daily movements of the stations

hour of the working day at four selected stations. In November 2013, when the data were taken, withdrawals were only allowed in the Palma system from 07:00 to 22:00 hours on weekdays (07:00 to 24:00 on weekends), while returns were possible at any time. The stations have been chosen to show different behaviors. At Station 111 (top right) withdrawals and returns follow a very similar pattern and we could say that the station is self-regulating. Station 116 (bottom left) shows an imbalance only at the beginning of the day, which could easily be offset by having the station full at opening time. Stations 104 and 122 are more unbalanced. There is one period in the day in which there are many more returns than withdrawals and another in which the situation is just the opposite. As these two imbalances are not too different in magnitude, they can be offset if when the service starts in the morning Station 104 is almost empty to allow many returns, while Station 122 should be full or almost full to satisfy the many withdrawal demands. Note that if we define the netflow as the difference between the number of withdrawals and the number of returns, there is an initial state in which all unsatisfied demands would be avoided if the difference between the maximum and the minimum netflow values is lower than or equal to the station capacity. In the cases in Figure 3, if instead of average values they represented the values for a specific day, as the differences between minimum and maximum values of the netflow at each station were lower than the station capacities (10 lockers at Stations 104 and 116; 20 lockers at Stations 111 and 122), it would be possible to find an initial state for which there would not be any unsatisfied demand.

Figures 2 and 3 clearly indicate that each station works in a different way and the analysis has to be done separately for each of them.

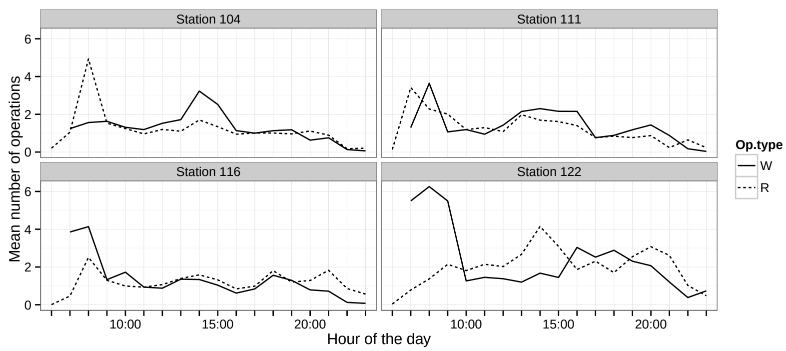


Figure 3: Withdrawals and returns over the course of the day for selected stations

Figure 4 indicates another source of variability between working and weekend days. For each station the corresponding line in the figure shows the average number of withdrawals and returns for each day of the week. The behavior of the users at each station is very similar on weekdays, while it is clearly different on Saturdays and on Sundays. Therefore, the analysis has to separate these three types of day.

In this study we address the static repositioning problem and divide it into two phases. In the first phase, described in Section 4, using the data stored in the system's database, we estimate the demands for bicycles and lockers at every station for each hour of each day of the week and use them to calculate the expected costs of the unsatisfied demands for each initial state of each station. In a second phase, described in Section 5, we decide first which is the best set of initial states attainable with the available bicycles and then we construct the

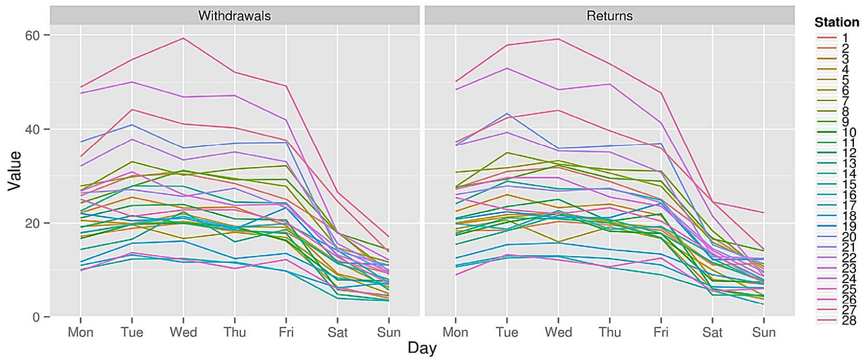


Figure 4: Average number of movements for each station and each day of the week

routes for the repositioning vehicles. In Section 6 we present the computational experience and Section 7 contains the conclusions.

4. Forecasting unsatisfied demand

The measure of service quality of the bike-sharing system considered in this study is a measure of *cost* that combines the two possible situations of unsatisfied demand, namely:

(a) A user wishing to withdraw a bicycle finds the station empty.

(b) A user returning a bicycle finds the station full.

Let η^{w} and η^{r} be the unknown number of thwarted withdrawals and returns within a temporal window $(t_0, t_0 + H)$, respectively. We define the measure of service-quality of the bike-sharing system in that time interval as:

$$\kappa = \eta^{\mathrm{w}} + \lambda\eta^{\mathrm{r}}$$

where λ is a weighting parameter to balance the relative importance assigned to the undesired situations (a) and (b). In what follows we have used $\lambda = 1$, but the methodology allows for any $\lambda \geq 0$.

Note that the measure of service-quality of the system as defined above is in fact the sum of the service-quality measures of each station in the system. Note also that the number of thwarted withdrawals and returns in any given station, $i \in S$, are random variables whose distributions depend on the number of available bicycles j at time t_0; i.e., the initial state of the station. Therefore, we will denote this stochastic cost by κ_{ij}.

The goal of this first phase of the study is to compute, for any given initial time t_0 and any temporal horizon H, a two-way table giving a forecast $\hat{\kappa}_{ij}$ of the measure of service-quality for station i assuming there are j bicycles available at the initial time. This table can then be used by the routing algorithm to evaluate each of the possible repositioning schemes.

The number of thwarted withdrawals and returns is predicted by fitting a statistical model, described below, using historical records of withdrawals and returns. The parameters of that model can be estimated and updated periodically, in order to feed the predictive algorithm. Using these estimated parameters we can simulate the probabilistic process of withdrawals and returns, and derive the expected number of failures-of-service as the mean of the thwarted withdrawals and returns across a sufficiently large number of simulations. As this is computationally demanding, we also propose a very accurate approximate iterative algorithm that reduces the computation time by an order of magnitude.

4.1. Statistical inference on withdrawal and return rates

We conducted an exploratory analysis of the historical records in order to identify the factors that drive the behavior of users. We found very clear differences between the dynamics of different stations, with some working mostly as either *providers* or *receivers* at different times of day. We also found clear differences with respect to the days of the week; we distinguish three day types:

weekdays, Saturdays, and Sundays. So we fit independent models to each station $i \in S$ and each type of day d.

We assume that the withdrawals and returns at a station i for all days of type d are realizations of the same stochastic processes

$$\mathcal{W}_t^{(id)} \quad \text{and} \quad \mathcal{R}_t^{(id)},$$

respectively. These are assumed to be independent and *inhomogeneous Poisson Processes* with rate parameters $\omega_{id}(t)$ and $\rho_{id}(t)$, respectively. Inhomogeneous Poisson processes are widely used for modeling nonstationary demands ([20]). In what follows, we omit the notation (id) on the understanding that everything will depend on the specific station i and the type of day d.

For practical computational purposes, we discretize the operational temporal window $(t_0, t_0{+}H)$ into M small intervals of length s minutes, assuming that the rate parameter functions $\omega(t)$ and $\rho(t)$ are constant in those intervals; i.e., $\omega(t) = \omega_m$ and $\rho(t) = \rho_m$ if $t_0{+}(m{-}1)s < t < t_0{+}ms$, for $m = 1, \ldots, M$, where ω_m and ρ_m are the mean of the functions $\omega(t)$ and $\rho(t)$ in the corresponding interval m. This is equivalent to assuming that the number of withdrawals and returns, both satisfied and unsatisfied, in the mth interval follow two independent Poisson distributions with parameters $s\omega_m$ and $s\rho_m$, respectively.

The parameter ω_m can be estimated by maximum likelihood from the historical records. Let n be the number of days in the historical records of the same type we are considering, let W_{lm} be the observed number of withdrawals during interval m of day $l = 1, \ldots, n$, and let T_{lm} be the time in minutes the station was available for withdrawals (i.e. not empty) within interval m of day l. Then, the maximum likelihood estimator of ω_m is given by:

$$\hat{\omega}_m = \frac{\sum_{l=1}^{n} W_{lm}}{\sum_{l=1}^{n} T_{lm}}.$$

The maximum likelihood estimator of the parameter ρ_m, $\hat{\rho}_m$, is obtained in a similar way.

4.2. Predictive algorithm of unsatisfied demand

For any given station i and day type d we will compute a point forecast of the number of thwarted withdrawals and returns in a temporal window (t_0, t_0+H), assuming that the station begins with j available bicycles. Our forecasts are based on the previous inhomogeneous Poisson processes using the estimated mean rates of withdrawals and returns: $\hat{\omega}_m$ and $\hat{\rho}_m$, $m = 1, \ldots, M$.

4.2.1. Simulation approach

The most straightforward approach to this computation is by simulating the processes of withdrawals and returns during the time span H while keeping track of the number of available bicycles and lockers in the station. For this, we use the fact that the time between arrivals in a Poisson process follows an Exponential distribution with the same rate. Thus we can simulate every single arrival and update the station status in continuous time. We can then simply count the number of thwarted withdrawals and returns in that period for each one of the realizations. Their *average* values give a Monte Carlo approximation to the point forecast of thwarted withdrawals and returns η^w and η^r, which we then combine linearly to calculate the forecasted *cost* $\hat{\kappa}_{ij}$ of having j available bicycles at station i.

This approximation can be made as accurate as desired by increasing the number of simulations. However, the computational burden increases rapidly.

4.2.2. Approximate approach

To speed up the forecasting computation we propose an approximate method for which we need a finer partition of the temporal window $(t_0, t_0 + H)$ than the partition used for the inference process. Let K be the number of time subintervals in this new partition, each of length $r \leq s$, r being a divisor of s. As this new partition is finer than the one used in the inference process, the subinterval k, for $k = 1, \ldots, K$, will be included in one of the intervals used for the inference process, say interval m; hence, the number of withdrawals and returns in subinterval k are Poisson variables with estimated intensities $r\hat{\omega}_m$ and

$r\hat{\rho}_m$, respectively. Therefore, the number X_k of additional bicycles in the station at the final point of subinterval k could be approached by the difference between the two independent Poisson variables, which follows a Skellam distribution [21], $X_k \sim \mathrm{Sk}(r\hat{\rho}_m,\ r\hat{\omega}_m)$.

The Skellam distribution allows us to compute analytically the probability of each possible outcome after interval k in a given station of capacity C. Let j be the number of available bicycles at the beginning of subinterval k. There are three different scenarios at the final point of the subinterval: in case $X_k < -j$ the station empties and there were $-X_k - j$ thwarted withdrawals; if $-j \leq X_k \leq C - j$ the station ends with $X_k + j$ bicycles, and there was no unmet demand; finally, in case $X_k > C - j$ the station fills and there were $X_k - C + j$ thwarted returns. In summary, the station may end up in any state from empty to full with conditional probabilities given by:

$$\begin{aligned} \mathrm{p}_k(0, j) &= \Pr(X_k \leq -j) \\ \mathrm{p}_k(j', j) &= \Pr(X_k = j' - j) \quad j' = 1, \ldots, C-1 \\ \mathrm{p}_k(C, j) &= \Pr(X_k \geq C - j) \end{aligned}$$

where $\mathrm{p}_k(h, j)$ is the probability that the station changes its state from j to h bicycles during interval k.

Similarly, the conditional expected numbers of thwarted withdrawals and returns during interval k are:

$$\begin{aligned} \mathrm{E}(\eta_k^w | j) &= \sum_{h=1}^{\infty} h \Pr(X_k = -(j+h)) \\ \mathrm{E}(\eta_k^r | j) &= \sum_{h=1}^{\infty} h \Pr(X_k = h + C - j) \end{aligned}$$

Using those results, the marginal expected numbers of thwarted withdrawals and returns during each interval k, from 1 to K, can be computed iteratively with the following algorithm. Initialize $\mathrm{p}_0(l) = 1$ for $l = j$ and $\mathrm{p}_0(l) = 0$ for

$l \neq j$. Then, for $k = 1, \ldots, K$ do:

$$\mathrm{E}(\eta_k^w) = \sum_{l=0}^{C} \mathrm{E}(\eta_k^w | l) \; \mathrm{p}_{k-1}(l)$$

$$\mathrm{E}(\eta_k^r) = \sum_{l=0}^{C} \mathrm{E}(\eta_k^r | l) \; \mathrm{p}_{k-1}(l)$$

$$\mathrm{p}_k(l') = \sum_{l=0}^{C} \mathrm{p}_{k-1}(l', l)\mathrm{p}_{k-1}(l) \quad \text{for } l' = 1, \ldots, C$$

Afterwards, the approximations $\hat{\eta}^w$ and $\hat{\eta}^r$ for the expected total numbers of thwarted withdrawals and returns with an initial state of j available bicycles are given by:

$$\hat{\eta}^w = \sum_{k=1}^{K} \mathrm{E}(\eta_k^w); \quad \hat{\eta}^r = \sum_{k=1}^{K} \mathrm{E}(\eta_k^r)$$

Finally, we combine these results into the final cost $\hat{\kappa}_{ij}$ for station i with initial configuration j.

With this approach we miss the cases where X_k is the result of combined withdrawals and returns, such that, considered continuously, the station would have been either filled or emptied with possibly some more losses. However, this approximation error is negligible if we take sufficiently small intervals. How small depends on the intensities of the underlying processes. In any case, it is possible to *calibrate* the method empirically using the simulation approach as a gold standard, and find a good compromise between speed and error.

5. Routing Algorithms

We present here the algorithms designed for the static repositioning problem. This repositioning is done when the usage rate of the system is negligible over the whole time planning horizon, typically at night. Our algorithm consists of two phases:

1. Calculating the target number of bicycles per station.
2. Constructing the repositioning routes for the vehicles.

5.1. Calculating the target number of bicycles per station

Let T_b be the total number of available bicycles, i.e. $T_b = \sum_{i\in\mathcal{S}} bs_i + \sum_{l\in\mathcal{V}} bv_l + \sum_{k\in\mathcal{D}} bd_k$. For each station $i \in \mathcal{S}$, b_i^{opt} will denote the number of bicycles that minimizes the expected unsatisfied demand, more precisely, b_i^{opt} is defined as the minimum j^* such that $\kappa_{ij*} = \min_{j=0,\ldots,C_i}\{\kappa_{ij}\}$, where C_i is the capacity of station i. If possible, this is the number of bicycles at each station that must be attained by the repositioning system. Nevertheless, it may happen that there are not enough bicycles to attain this level because $T_b < \sum_{i\in\mathcal{S}} b_i^{opt}$. In this case, it is impossible to achieve the optimal level, so the first decision we take is to determine the number of bicycles that each station should have after the repositioning is done so that the global expected dissatisfaction is minimal for the number of bicycles that are really available. We determine these numbers b_i^t, called *target* numbers, by solving an integer program that is formulated as follows. Let $x_{ij} = 1$ if station i has a target number of bicycles equal to j, and 0 otherwise; and let y_i be the number of bicycles that have to be moved to or from station i.

$$\begin{aligned}
min \quad & \sum_{i\in\mathcal{S}}\sum_{j=0}^{C_i} \kappa_{ij}x_{ij} + \alpha\sum_{i\in\mathcal{S}} y_i && (1)\\
\text{s.t.} \quad & \sum_{j=0}^{C_i} x_{ij} = 1 \quad i \in \mathcal{S} && (2)\\
& \sum_{i\in\mathcal{S}}\sum_{j=0}^{C_i} j\, x_{ij} \le T_b && (3)\\
& bs_i - y_i \le \sum_{j=0}^{C_i} j\, x_{ij} \le y_i + bs_i && (4)\\
& x_{ij} \in \{0,1\}, y_i \ge 0, i \in \mathcal{S}, j \in \{0,\ldots C_i\} && (5)
\end{aligned}$$

Constraints (2) mean that each station must be assigned to one target level, constraints (3) require that the sum of target levels cannot be greater than the total number of available bicycles, and constraints (4) force that $y_i \ge |bs_i - \sum_{j=0}^{C_i} j\, x_{ij}|$. The first term in the objective function (1) is obviously the total unsatisfied demand that has to be minimized as the first objective. The second term is weighted by a small number α and counts the number of bicycles that should be moved by the repositioning system, that is, the sum of the absolute differences between the current number, bs_i, and the target number of bicycles for each station. It has been observed that, generally, there are several values of j for which κ_{ij} is very near to $Min\{\kappa_{ij} : j = 0, \dots, C_i\}$ (see Table 1 in Section 6.2). Therefore, there are several target levels that produce optimal or near-optimal solutions in terms of the total unsatisfied demand. As it is clear that a solution involving a lower number of movements is preferable, we include, as a secondary objective, the total number of bicycles that have to be transported. We set $\alpha = 0.01$.

Let $\bar{x}_{ij}$ represent the solution of the above integer program. We define the target value for each $i \in \mathcal{S}$ as $b_i^t = j$ for the unique value j such that $\bar{x}_{ij} = 1$. Obviously, as mentioned before, if $T_b \geq \sum_{i \in \mathcal{S}} b_i^{opt}$, the integer program is not solved and the target value is defined as $b_i^t = b_i^{opt}$ for each $i \in \mathcal{S}$.

5.2. Constructing the repositioning routes for the vehicles

Once the target values b_i^t for each station $i \in \mathcal{S}$ have been determined, the next step is to design the routes for the repositioning vehicles that will transport the bicycles.

This is a pure routing problem that is very difficult by itself. We summarize now the main characteristics of the routing problem we face and put it in the context of routing problems.

Our routing problem can be defined on the graph that was introduced in Section 3 whose set of vertices V represents the stations, the depots, and the initial vehicle locations. Data of each station include its capacity and current numbers of ready and damaged bicycles, respectively. For each vehicle we know its capacity and initial numbers of ready and damaged bicycles on the vehicle. For each depot we know the number of ready bicycles stored initially in the depot. We are also given the parking time of each vehicle, the unit time for loading/unloading a bicycle and the travel time between any pair of vertices. The goal, as said before, consists of finding a set of routes for the vehicles. Each route consists of a sequence of visits to stations/depots as well as the loading instructions for each visit. The following conditions have to be considered:

- the final number of bicycles in each station must be equal to its target value,
- all the damaged bicycles must be transported to a depot,
- the route of each vehicle starts at its initial location and ends at a station or depot,
- multiple visits to a station or depot are allowed (even by the same vehicle),
- the capacity of the vehicles and the stations is never exceeded,
- the final number of bicycles in each vehicle is zero.

The set of routes should minimize a weighted combination of the total time needed to operate all the routes and the coefficient of variation of the different duration of the routes. Thus, we seek routes that are as short as possible, but also balanced.

As far as we know there is no paper in the literature that deals with exactly this problem; nevertheless there are an increasing number of studies devoted to similar pickup and delivery problems, which appear in many different real-life situations ([22], [23], [24], [25]). Mathematically, our problem can be considered as a variant of the multi-commodity pickup and delivery problem, introduced by Hernández-Pérez and Salazar-González [26]. They propose a branch-and-cut procedure, for the case where only one vehicle is available, based on a mixed integer linear programming model. In our case the number of commodities is two (ready and damaged bicycles), but unfortunately their procedure cannot be easily adapted to our problem which considers, among other characteristics, multiple vehicles with different capacities, several depots, multiple visits at each station, etc.

We have implemented a heuristic algorithm to solve this problem, consisting mainly of two phases. In the first phase we solve a Minimum Cost Flow Problem (MCFP) whose solution is used to guide the second phase, where an insertion algorithm is used to iteratively construct the routes. Given that certain stations have a deficit of ready bicycles while others have a surplus, the MCFP is used to find an estimate of the number of ready bicycles that should be transported from stations with surplus to stations with deficit. Damaged bicycles are not considered in this problem as it is clear that all damaged bicycles must be transported from the station to the depots.

We say that a station $i \in \mathcal{S}$ is a supply node if $bs_i - b_i^t > 0$ and its supply is defined as $O(s_i) = bs_i - b_i^t$. Similarly, we say that a station $i \in \mathcal{S}$ is a demand node if $bs_i - b_i^t < 0$ and its demand is defined as $D(s_i) = b_i^t - bs_i$. The Minimum Cost Flow Problem (MCFP) is defined as follows. Let $G = (N, A)$ be a directed graph. The nodes in N and their corresponding supply/demands are:

- A node s_i for each $i \in \mathcal{S}$ that is either a supply node with supply $O(s_i)$ or a demand node with demand $D(s_i)$,

- A supply node v_l for each vehicle $l \in \mathcal{V}$ such that $bv_l > 0$, with supply $O(v_l) = bv_l$,

- A supply node d_k for each depot $k \in \mathcal{D}$ such that $bd_k > 0$, with supply $O(d_k) = bd_k$,

- An extra node d'_k for each depot $k \in \mathcal{D}$ with zero demand and also a dummy demand node U with demand $D(U) = T_b - \sum_{i \in \mathcal{S}} b_i^t$ if $T_b > \sum_{i \in \mathcal{S}} b_i^t$.

Note that, if $T_b > \sum_{i \in \mathcal{S}} b_i^t$, the optimal state for the stations involves a number of bicycles that is less than T_b, so some bicycles should be stored in the depots. The set of arcs A contains an arc from each supply node to each demand node (including those depots with zero demand) with a cost equal to the travel time between the corresponding locations. Furthermore, if U exists, arcs with zero cost are added from each node d'_k to U.

The solution of the MCFP defined on this graph provides a set of arcs with positive flows. By discarding the arcs entering the dummy node U we obtain a set of arcs $F = \{(u_j, w_j) : j = 1, \ldots n_f\}$, each arc associated with a flow f_j. If we are able to transport f_j bicycles from the supply station (or vehicle) u_j to the demand station (or depot) w_j, we will have obtained a solution that leaves each station at its target level. Thus, the solution of the MCFP is used as a guide to construct the routes for the vehicles: the nodes from where bicycles should be taken, the nodes where they have to be left and how many bicycles have to be transported between each origin and each destination.

In Figure 5 we represent an example with eight stations, one depot, and two vehicles. The numbers above each station are the current number of bicycles in the station, the number of damaged bicycles (between parentheses), and its target value. The number above each vehicle or depot represents the number of bicycles that are initially in that vehicle or depot. Figure 6 shows a solution of the corresponding MCFP. The supplies (demands) in this MCFP are depicted above each node (demands are represented as negative numbers); thus, for instance, station s_1 has demand 1 and station s_5 has supply 4. Note that the number of available bicycles (27) is equal to the sum of target values in this example, so the dummy demand nodes are not needed. The arcs depicted in Figure 6 represent a solution of the MCFP, where the number beside each arc indicates the number of bicycles to be moved; thus the MCFP solution suggests that the bicycle on vehicle 1 should be left at station s_1, four bicycles should be transported from station s_5 to station s_4, and so on.

The routes for the vehicles are incrementally constructed with an insertion algorithm. At each state of the insertion algorithm, the route for vehicle l is a sequence of stops $i_0^l, \ldots i_d^l$, where i_0^l is the initial location of the vehicle and i_d^l is the location at which the vehicle will finish its task. Recall that multiple (non-consecutive) visits to the same station or depot are allowed.

Let T_l be the total time of the route assigned to vehicle l. Our algorithm assumes that the routes to be built must leave each station at its target inventory level, all the damaged bicycles must be taken to the depots, and the total

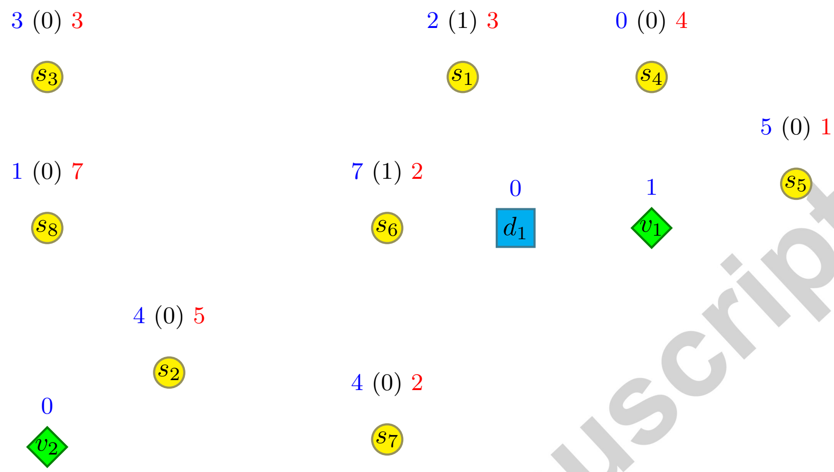


Figure 5: Example of an initial setting

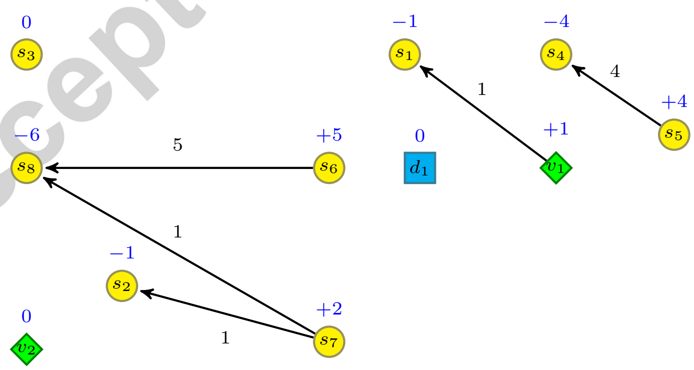


Figure 6: Solution of the Minimum Cost Flow Problem

duration of the routes has to be minimized while keeping the routes as balanced as possible. Thus the objective function to be minimized is:

$$\beta \sum_{l \in \mathcal{V}} T_l + (1 - \beta) CV \tag{6}$$

where CV is the coefficient of variation of the duration of the routes (standard deviation of the routes' duration divided by the average duration).

The insertion algorithm works as follows. At each iteration, an arc of the solution provided by the MCFP is selected and inserted in an appropriate route. Let $(u_j, w_j) \in F$, with flow f_j, be the arc to be inserted, and let $l \in \mathcal{V}$ be a candidate route (the routes are indexed the same as the vehicles) for the insertion. Insertion is in fact a *double insertion*: u_j is inserted after a given stop on the current route, say i_r^l, and w_j is inserted after a subsequent stop, say i_s^l, where $s \geq r$. The resulting route after the double insertion has been carried out is $i_0^l, \ldots, i_r^l, u_j, i_{r+1}^l, \ldots, i_s^l, w_j, i_{s+1}^l, \ldots, i_d^l$. Note that f_j more bicycles have to be on the vehicle between visits i_{r+1}^l and i_s^l, so we must check that the capacity of the vehicle is not exceeded. On the other hand, it may happen that $u_j = i_r^l$ or $u_j = i_{r+1}^l$, in which case the route would be simplified by removing the duplicated visits and accumulating the loading/unloading operations at u_j (similarly for w_j). It may happen that the flow of an arc f_j is greater than the remaining capacity of the vehicle; in this case the number of bicycles to move is set equal to the capacity of the vehicle and, once it has been inserted, the arc is maintained in the list of arcs and its corresponding flow is reduced by a quantity equal to the number of bicycles inserted. Each iteration of the insertion algorithm consists of the following steps:

- A flow arc (u_j, w_j) with flow f_j is selected.
- For every vehicle, all possible insertion positions are checked for feasibility and the ones producing the minimum increment in the weighted cost (according to (6)) are determined.
- The flow arc is inserted in the vehicle (and positions) that produce the minimum increase in the weighted cost. The flow f_j is reduced by a

quantity equal to the number of bicycles inserted and, if it becomes zero, the flow arc (u_j, w_j) is removed from the list.

Depending on the type of flow arcs the above insertion steps are applied differently. A list of types of flow arcs and the ways they have to be managed follows:

1. u_j and w_j correspond to the same depot. Therefore the corresponding bicycles do not have to be moved. The arc is discarded in the insertion algorithm.
2. u_j and w_j are both stations, or u_j is a depot and w_j is an station. The steps are applied normally.
3. u_j corresponds to a vehicle. In this case the arc must be inserted in the route corresponding to that vehicle.
4. w_j corresponds to a depot. This means that f_j bicycles have to be transported to a depot. This case is managed normally except for the fact that the bicycles can be transported to any depot, not necessarily to w_j, and this must be taken into account when considering all possible insertion positions and their costs.

Arcs where u_j corresponds to a vehicle are first managed by the algorithm and inserted in the route of the corresponding vehicle. Note that several arcs of this type may exist for the same vehicle (the number of bicycles on the vehicle may initially be large) and all of them are inserted before the other arcs. After these arcs have been inserted, the vehicle route will consist of a sequence of stations (and eventually a depot) where bicycles are unloaded, so the vehicle will be empty after the last visit. This strategy implies that any arc of the other types can always be feasibly inserted after the last visit.

There may also be some arcs that cannot be inserted in the usual way because the destination station contains a number of damaged bicycles that make it impossible to leave all the bicycles that have to be unloaded unless the damaged bicycles are picked up. We call them *conflict arcs*. Conflict arcs are inserted when all the other flow arcs have been inserted and they are managed with *triple insertions*. If (u_j, w_j) with flow f_j is such an arc, then inserting it into a route involves inserting visit u_j to take f_j bicycles, inserting visit w_j to unload f_j bicycles and pick up the necessary number of damaged bicycles from the station, and finally inserting a third visit to a depot to unload the damaged bicycles.

After all the flow arcs have been inserted in the vehicle routes, a similar insertion is performed to manage the damaged bicycles that remain in the stations. This operation can also be viewed as inserting an arc (u_j, w_j) with flow f_j where u_j is the station with damaged bicycles, w_j is any depot, and f_j is the number of damaged bicycles in u_j.

Overall, the insertion algorithm considers the insertion arcs in the following order.

1. Flow arcs (u_j, w_j) for which u_j corresponds to a vehicle,
2. The remaining flow arcs, except those that are conflict arcs, in any order,
3. Conflict arcs, for which triple insertion is used, and
4. Arcs associated with the damaged bicycles that remain in the stations.

Figure 7 illustrates the application of the insertion algorithm for the MCFP solution of Figure 6. It shows the routes obtained after each flow arc is inserted. Each route is represented as a sequence of visits: the number above each visited node indicates the number of bicycles picked up (+) or delivered (−), and the number above each arc represents the number of bicycles on the vehicle when it travels from one node to another.

This solution does not take into account the damaged bicycles that have to be picked up at the stations and transported to a depot. In this example, stations s_1 and s_2 had one damaged bicycle each. The arcs associated with damaged bicycles are inserted in route 1, producing the route depicted in Figure 8 (a). In this figure, the operations that involve damaged bicycles appear in parentheses.

We have used a simple procedure to improve the final solution given by the insertion algorithm. It consists of removing a visit from a route and inserting it

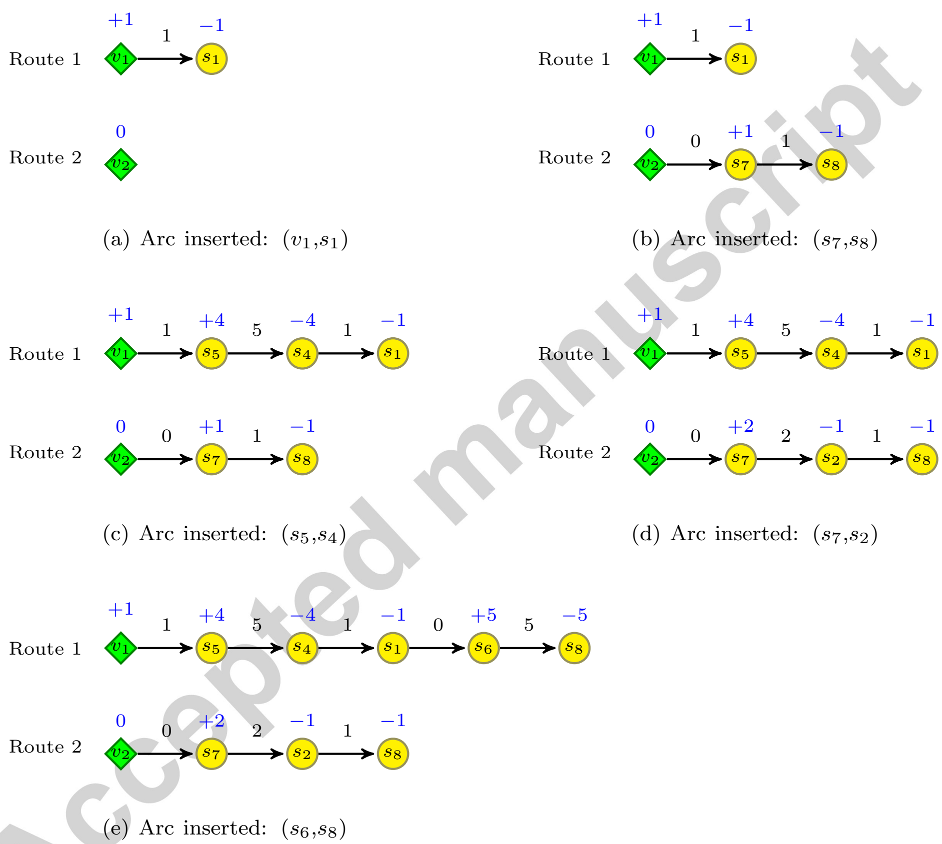
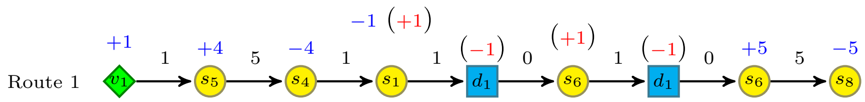
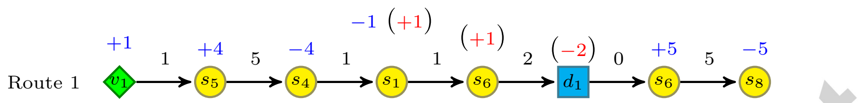


Figure 7: Insertion algorithm



(a) Insertion of damaged bicycles



(b) Route 1 after improvement

Figure 8: Improvement move

at a different position in the same route. We try to avoid, if possible, multiple visits to the same node. For instance, route 1 in the example includes two visits to the depot. If we remove the first visit to station s_6 and insert it after the visit to s_1, the result would be two consecutive visits to the depot, which can obviously be merged into only one visit to unload two damaged bicycles. The route obtained after this improvement is depicted in Figure 8 (b).

Finally, the insertion and improvement procedures have been embedded in a multistart algorithm. At each iteration of the multistart algorithm, arcs of the same type are used in random order by the insertion procedure, thus producing different solutions. The best solution in terms of total time of the routes and balance between them is then selected.

6. Computational results

6.1. Expected unsatisfied demands

We followed the process described in Section 4 using the data from the Palma bike-sharing system. The aim was to obtain tables of the expected measure of service quality of the system for the third week of November 2013, from Sunday 17th to Saturday 23rd, that will be used as cost tables in order to compute the repositioning routes. To avoid seasonal oscillations or long-term trends, we used historical records from only the three-month interval previous to that week.

The estimation of the rate parameters of withdrawals and returns, ω_m and ρ_m, was done using intervals of length $s = 60$ minutes. Once these estimates of the rate parameters were obtained for each type of day of the target week, starting at 7.00 a.m., we performed 1600 realizations of the withdrawal and return processes for each type of day and station i, for a time span of 6 hours. From each realization, we directly computed the number of thwarted withdrawals and returns for each initial state j, and therefore the corresponding realized cost. Finally, the tables of expected costs $\hat{\kappa}_{ij}$ for each type of day and for three different time spans $H = 2, 4,$ and 6 hours are computed as the average realized cost over each scenario. The simulation process took about 8 minutes for each type of day on a standard personal computer. We used these results as a reference to validate and tune the approximate (and faster) method described at the end of Section 4.

The running time of the forecast approximate approach is linear with respect to the number of prediction subintervals (Fig. 9). We obtain essentially the same expected cost matrices as with the simulation approach in 0.87 minutes, using prediction subintervals of 30 minutes. Specifically, the Mean Absolute Error of the cost, calculated across all the stations, initial states, and time spans was under 0.1.

The matrix corresponding to a weekday and $H = 2$ hours appears in Table 1. Some stations have some initial states with cost 0, meaning that if the station was in one of these states at 7.00 AM no unsatisfied demand is expected for the next two hours. For some other stations there are expected unsatisfied demands for any possible initial state, though the expected numbers vary widely and there are states clearly preferable to others. Looking at each station, the unsatisfied demands typically have asymmetric U-shapes. For initial states close to zero, the unsatisfied demands would be thwarted withdrawals, while for states close to the station capacity they would be thwarted returns.

Table 1: Expected costs of unsatisfied demand for every station and every initial state

	Station																											
State	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
0	3,57	2,01	1,70	1,08	2,75	1,53	6,50	3,89	4,86	2,19	1,42	1,68	1,28	1,59	0,83	5,58	0,43	0,53	0,89	0,75	0,87	8,77	1,12	1,47	2,62	2,79	7,02	6,38
1	2,63	1,18	1,02	0,53	1,84	0,81	5,52	2,95	3,87	1,36	0,84	1,00	0,59	0,99	0,29	4,59	0,10	0,20	0,49	0,33	0,41	7,77	0,56	0,83	1,74	1,92	6,04	5,40
2	1,83	0,63	0,58	0,26	1,11	0,38	4,56	2,11	2,94	0,77	0,48	0,55	0,22	0,59	0,08	3,64	0,02	0,07	0,27	0,15	0,18	6,78	0,27	0,43	1,05	1,21	5,10	4,45
3	1,19	0,30	0,34	0,18	0,60	0,16	3,65	1,42	2,11	0,40	0,26	0,28	0,07	0,34	0,02	2,76	0	0,03	0,15	0,07	0,07	5,80	0,12	0,20	0,57	0,71	4,21	3,56
4	0,74	0,14	0,27	0,23	0,29	0,06	2,82	0,89	1,41	0,19	0,14	0,13	0,02	0,18	0	1,98	0	0,03	0,08	0,04	0,02	4,85	0,05	0,08	0,28	0,38	3,40	2,75
5	0,44	0,07	0,32	0,39	0,13	0,02	2,09	0,52	0,88	0,08	0,06	0,06	0,01	0,09	0	1,35	0	0,06	0,04	0,03	0,01	3,95	0,02	0,03	0,12	0,19	2,68	2,05
6	0,28	0,08	0,49	0,67	0,05	0,01	1,50	0,28	0,52	0,03	0,03	0,02	0	0,04	0	0,88	0	0,15	0,02	0,04	0	3,12	0,01	0,01	0,05	0,09	2,07	1,47
7	0,24	0,15	0,79	1,08	0,03	0	1,05	0,14	0,28	0,01	0,01	0,01	0	0,02	0	0,54	0	0,35	0,02	0,06	0	2,38	0	0	0,02	0,04	1,59	1,00
8	0,30	0,33	1,23	1,65	0,04	0	0,75	0,07	0,14	0	0,01	0	0	0,01	0	0,33	0	0,74	0,02	0,10	0	1,75	0	0	0,01	0,02	1,25	0,66
9	0,50	0,67	1,85	2,38	0,12	0	0,61	0,03	0,07	0	0	0	0	0	0	0,24	0	1,35	0,04	0,17	0	1,24	0	0	0	0,01	1,10	0,41
10	0,87	1,22	2,65	3,25	0,34	0	0,71	0,01	0,03	0	0	0	0,01	0	0	0,28	0	2,21	0,07	0,29	0	0,84	0	0	0	0	1,25	0,25
11						0		0,01	0,01	0	0,01	0	0,02	0	0		0		0,14	0,46	0	0,55	0	0,01	0	0		0,14
12						0		0,02	0	0	0,03	0,01	0,04	0	0		0		0,25	0,71	0,01	0,34	0	0,03	0	0		0,08
13						0,01		0,05	0	0	0,06	0,03	0,10	0	0		0		0,43	1,05	0,02	0,21	0	0,06	0	0		0,04
14						0,03		0,18	0	0	0,13	0,07	0,20	0	0,01		0,01		0,69	1,49	0,04	0,12	0	0,12	0	0,01		0,02
15						0,08		0,56	0	0	0,26	0,14	0,37	0	0,03		0,03		1,08	2,05	0,10	0,07	0	0,23	0	0,01		0,01
16						0,18			0	0,01	0,49	0,27	0,66	0	0,07		0,08		1,59	2,72	0,22	0,03	0	0,42	0	0,04		0,01
17						0,38			0	0,04	0,87	0,50	1,09	0	0,19		0,21		2,24	3,49	0,46	0,02	0,01	0,73	0	0,09		0,02
18						0,71			0,01	0,13	1,41	0,87	1,67	0	0,44		0,50		3,02	4,35	0,87	0,01	0,01	1,18	0,02	0,21		0,05
19						1,21			0,03	0,33	2,14	1,41	2,41	0	0,89		1,02		3,91	5,28	1,49	0,02	0,03	1,80	0,10	0,45		0,15
20						1,92			0,15	0,76	3,03	2,18	3,27	0	1,59		1,79		4,86	6,25	2,33	0,10	0,06	2,62	0,43	0,92		0,45
21														0,01									0,11					
22														0,02									0,20					
23														0,04									0,35					
24														0,09									0,57					
25														0,18									0,89					
26														0,36									1,33					
27														0,66									1,90					
28														1,13									2,60					
29														1,81									3,42					
30														2,67									4,34					

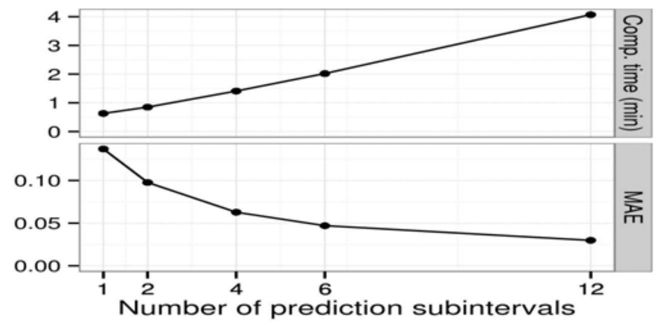


Figure 9: Computing time and Mean Absolute Error of the approximate approach as a function of the number of prediction sub-intervals

6.2. Results for the initial configuration of the repositioning system

In the initial configuration the company wanted to study, there were two vehicles with capacity 15 and one depot with 10 available bicycles. The data corresponded to the third week of November 2013, from Sunday 17th to Saturday 23rd. The costs corresponding to Tuesday 19th appear in Table 1. If we add up the number of bicycles corresponding to the minimum cost state at each station (taking the lowest value in the event of a tie), the required number of bicycles was 214. However, on this day there were only 211 available bicycles. Therefore not all the optimal states could be attained. In order to decide the optimal attainable state for each station minimizing the total cost, we solved the integer linear program described in Section 5. Its solution gave us the target number of bicycles each station should have after the repositioning moves.

Using this target number and the current state of the stations and the depot, we constructed two routes, one for each vehicle, that performed all the required moves to achieve the target state for each station. A time of 60 seconds was included for each loading or unloading move of a bicycle and for the parking maneuver. The main objective when designing the routes was to minimize the total time, while keeping the routes as balanced as possible. The two routes are shown in Figure 10. Both routes start at the depot, but they finish at the station where the last move was made. The company considered that that was the end of the service and did not want to make the vehicles return to the depot. Note that not all the stations are visited. The routes have to alternate stations with surplus and stations with shortage. Vehicle 1 (continuous lines) takes 3 bicycles from the depot and finishes empty, while Vehicle 2 (dotted line) takes the remaining 7 bicycles and finishes its route empty.

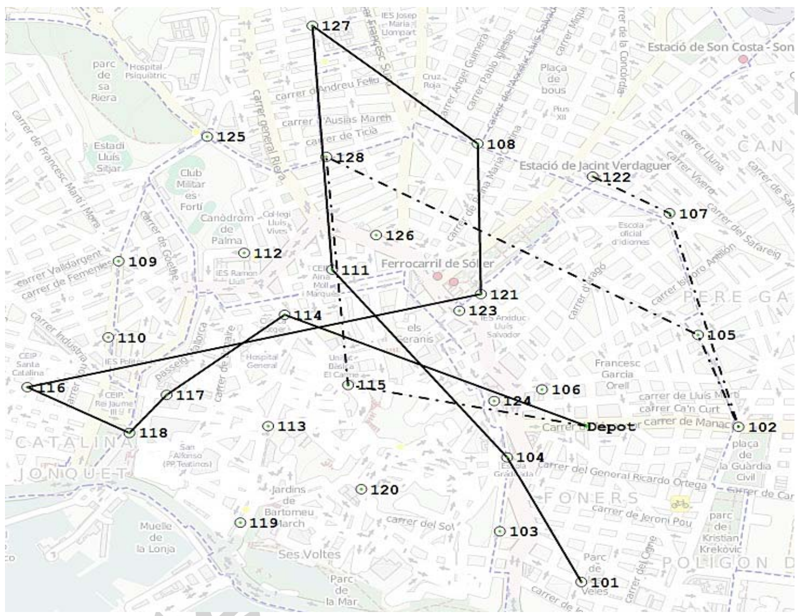


Figure 10: Repositioning routes for two vehicles

The effect of the repositioning can be observed in Figure 11, which shows the number of available bicycles at Station 127 over the course of Tuesday 19th if the initial state was 7 (solid red line), as it was at the end of the previous day, or 9, the optimal initial state to which the repositioning system would bring the station (dashed blue line). The figure shows at level 0 the intervals during which the station would be empty in the two cases. In particular, this station was empty that day for 40 minutes between 7:00 and 9:00 a.m., with an expected number of thwarted withdrawals of 4.9 during that two-hour period. If the station had been repositioned at night to its optimal initial state of 9 bicycles, the first two unsatisfied demands would have been met, but, after that, the evolution of the station would coincide with what is actually observed. A dynamic repositioning system throughout the day would be necessary to eliminate later unsatisfied demands.

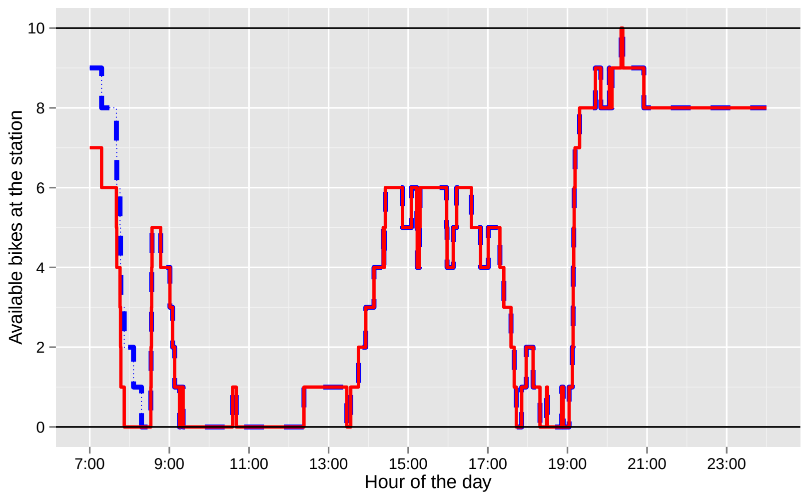


Figure 11: Evolution of Station 127 with and without repositioning

The results obtained for the seven days of the week are summarized in Table 2, which shows the number of bicycles moved, the total distance, and the total service time for each day and each vehicle. A perfect balance in service time cannot be obtained, but the routes never differ by more than 15 minutes. The routes also include the removal of damaged bicycles, as was explained in Section 5.

6.3. Exploring alternative configurations for the repositioning system

The procedures we have developed can be used to explore alternative configurations of the repositioning system. For instance, what would happen if instead of having two vehicles the system had one or three. The results for these cases appear in Tables 3 and 4 and the comparison of the maximum service time in each case is shown in Figure 12. This information could be used by the managers of the system to decide the most appropriate number of vehicles. In Table

Table 2: Repositioning routes for the seven days of the week

Day	Vehicle	Distance (meters)	Bicycles	Damaged bicycles	Time (seconds)
Sunday, 11-17-2013	1	3234	2	0	592
	2	949	6	0	908
Monday, 11-18-2013	1	10056	30	0	5284
	2	9127	28	0	4977
Tuesday, 11-19-2013	1	7357	18	0	3409
	2	9295	22	0	4149
Wednesday, 11-20-2013	1	7979	26	0	4534
	2	11827	26	4	5331
Thursday, 11-21-2013	1	9690	34	0	5497
	2	11867	26	4	5334
Friday, 11-22-2013	1	10618	24	0	4484
	2	9204	24	0	4262
Saturday, 11-23-2013	1	6461	4	4	1785
	2	6273	6	2	1771

Table 3: Repositioning routes using one vehicle

Day	Vehicle	Distance (meters)	Bicycles	Damaged bicycles	Time (seconds)
Sunday, 11-17-2013	1	3234	8	0	1432
Monday, 11-18-2013	1	14545	58	0	9927
Tuesday, 11-19-2013	1	13742	40	0	7349
Wednesday, 11-20-2013	1	15169	52	4	9732
Thursday, 11-21-2013	1	14739	60	4	10541
Friday, 11-22-2013	1	16383	48	0	8499
Saturday, 11-23-2013	1	9804	10	6	3345

4 the number of bicycles to be moved on Sunday is so small that there is no need to design a third route. Figure 12 shows the maximum time for the routes with 1, 2, and 3 vehicles. It can be observed that the maximum service time decreases dramatically when the number of vehicles is increased from one to two, but the decrease is very small when we change from two to three vehicles. If this element is important - if, for instance, the available time to provide the service is strictly limited - using two vehicles seems a much better alternative than using only one. However, using three instead of two would not add any significant advantage.

Table 4: Repositioning routes using three vehicles

Day	Vehicle	Distance (meters)	Bicycles	Damaged bicycles	Time (seconds)
Sunday, 11-17-2013	1	3234	2	0	592
	2	949	6	0	908
	3				
Monday, 11-18-2013	1	9361	16	0	3313
	2	6737	20	0	3365
	3	6841	22	0	3852
Tuesday, 11-19-2013	1	8166	14	0	2867
	2	5398	12	0	2308
	3	4495	14	0	2483
Wednesday, 11-20-2013	1	6514	20	0	3349
	2	4449	18	0	3080
	3	11412	14	4	3821
Thursday, 11-21-2013	1	4215	28	0	4263
	2	10840	16	4	3900
	3	8915	16	0	3041
Friday, 11-22-2013	1	5308	16	0	2782
	2	4723	18	0	2860
	3	10774	14	0	3175
Saturday, 11-23-2013	1	5780	2	2	1136
	2	5220	4	2	1335
	3	4219	4	2	1263

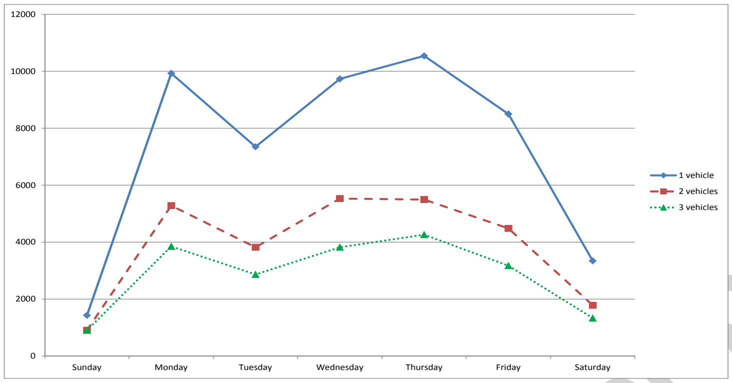


Figure 12: Comparing the maximum service time for different numbers of vehicles

Another element of the repositioning system that could be interesting to study is the capacity of the vehicles. Initially this was set at 15, but we could explore other alternatives, for instance, vehicles of capacity 10 and capacity 20. Figure 13 shows the maximum service time for these three possible capacities. It can be observed there are no significant differences between them. For this system, in which a relatively small number of bicycles are moved every day, the capacity of the vehicles is not an issue. Small vehicles can give as good a service as larger vehicles.

7. Conclusions

The repositioning system is a key factor in the quality of the service provided by the bike-sharing systems appearing and growing everywhere in the world. It is, therefore, not surprising that intense research effort has been devoted to it in recent years. Although each system has its special characteristics, all of them share two basic components, the prediction part, forecasting a stochastic demand, and the routing part, in which the forecast demands have to be met

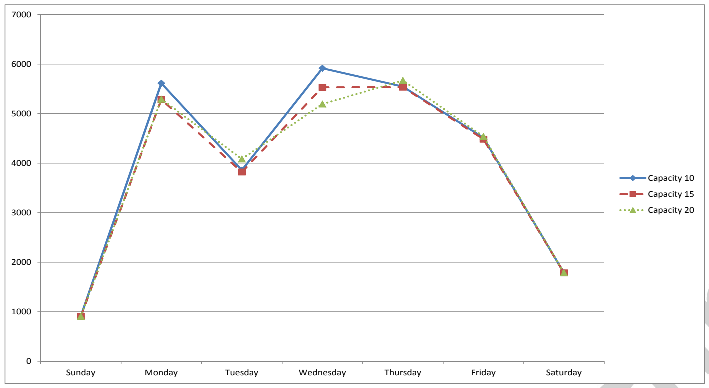


Figure 13: Comparing the maximum service time for different capacities of vehicles

to ensure the satisfaction of the users. In our proposal, developed in this study, we have considered both components jointly and designed a procedure that automatically reads the information in the system, and uses this information, past and present, to predict the demands for withdrawals and returns at each station for each time period. Combining these predictions with the current state of the system, the procedure designs the most appropriate repositioning routes for the available vehicles.

The proposed procedure has been tested on a real bike-sharing system in Spain and the results show its usefulness for solving the daily problem and also as a planning tool that allows the user to evaluate alternative configurations. Although the tests have been made on a relatively small system, all the prediction and routing procedures can be used for much larger systems. Most of the prediction part can be done off-line and the heuristic algorithms of the routing part are extremely fast and can be applied to large numbers of stations, vehicles, and bicycles.

The next step in this line of research will be to tackle the dynamic repositioning problem, in which the procedures developed here must be dynamically adapted to the actual state of the stations served by the vehicles. In the static case, when a vehicle arrives at a stations, it knows which state it is going to find, but this is no longer true in the dynamic case, in which, while the repositioning vehicle is en route to a station, the state of this station is changing with new withdrawals and returns. The dynamic problem is clearly more complex and will require new strategies and algorithms to give fast answers to continuously changing conditions.

8. Acknowledgments

This study has been partially supported by the Spanish Ministry of Science and Innovation, IPT 2011-1355-370000, the Spanish Ministry of Economy and Competitiveness, MTM2012-36163-C06-02, and the Generalitat Valenciana, PROMETEO/2013/049.

References

[1] MetroBike, The bike-sharing world map, http://www.metrobike.net/the-bike-sharing-world-map/.

[2] M. Benchimol, P. Benchimol, B. Chappert, A. de la Taille, F. Laroche, F. Meunier, L. Robinet, Balancing the stations of a self service "bike hire" system, RAIRO - Operations Research 45 (1) (2011) 37–61.

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

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

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

[6] M. Rainer-Harbach, P. Papazek, B. Hu, G. R. Raidl, Balancing bicycle sharing systems: A variable neighborhood search approach, in: M. Middendorf, C. Blum (Eds.), Evolutionary Computation in Combinatorial Optimization, Vol. 7832 of Lecture Notes in Computer Science, 2013, pp. 121–132.

[7] G. R. Raidl, B. Hu, M. Rainer-Harbach, P. Papazek, Balancing bicycle sharing systems: Improving a VNS by efficiently determining optimal loading operations, in: M. J. Blesa, C. Blum, P. Festa, A. Roli, M. Sampels (Eds.), Hybrid Metaheuristics, Vol. 7919 of Lecture Notes in Computer Science, 2013, pp. 130–143.

[8] P. Papazek, G. R. Raidl, M. Rainer-Harbach, B. Hu, A PILOT/VND/GRASP hybrid for the static balancing of public bicycle sharing systems, in: R. Moreno-Díaz, F. Pichler, A. Quesada-Arencibia (Eds.), Computer Aided Systems Theory - EUROCAST 2013, Vol. 8111 of Lecture Notes in Computer Science, 2013, pp. 372–379.

[9] J. Schuijbroek, R. Hampshire, W.-J. van Hoeve, Inventory rebalancing and vehicle routing in bike sharing systems, Tech. Rep. Paper 1491, Tepper School of Business, Carnegie Mellon University, Pittsburgh, Pensilvania, USA (2013).

[10] G. Erdoğan, G. Laporte, R. Wolfler Calvo, The static bicycle relocation problem with demand intervals, European Journal of Operational Research 238 (2) (2014) 451–457.

[11] S. C. Ho, W. Y. Szeto, Solving a static repositioning problem in bike-sharing systems using iterated tabu search, Transportation Research Part E: Logistics and Transportation Review 69 (2014) 180–198.

[12] R. Nair, E. Miller-Hooks, Fleet managment for vehicle sharing operations, Transportation Science 45 (4) (2011) 524–540.

[13] C. Contardo, C. Morency, L.-M. Rousseau, Balancing a dynamic public bike-sharing system, Tech. Rep. CIRRELT-2012-09, CIRRELT, Université de Montréal, Montréal, Canada (2012).

[14] L. Caggiani, M. Ottomanelli, A dynamic simulation based model for optimal fleet repositioning in bike sharing systems, Procedia-Social and Behavioral Sciences 87 (2013) 203–210.

[15] L. Di Gaspero, A. Rendl, T. Urli, A hybrid ACO+CP for balancing bicycle sharing systems, in: M. J. Blesa, C. Blum, P. Festa, A. Roli, M. Sampels (Eds.), Hybrid Metaheuristics, Vol. 7919 of Lecture Notes in Computer Science, 2013, pp. 198–212.

[16] L. Di Gaspero, A. Rendl, T. Urli, Constraint-based approaches for balancing bike sharing systems, in: C. Schulte (Ed.), Principles and Practice of Constraint Programming, Vol. 8124 of Lecture Notes in Computer Science, 2013, pp. 758–773.

[17] T. Raviv, O. Kolka, Optimal inventory management of a bike-sharing station, IIE Transactions 45 (10) (2013) 1077–1093.

[18] I. A. Forma, T. Raviv, M. Tzur, A 3-step math heuristic for the static repositioning problem in bike-sharing systems, Transportation Research Part B: Methodological 71 (2015) 230–247.

[19] G. Erdoğan, M. Battarra, R. Wolfler Calvo, An exact algorithm for the static rebalancing problem arising in bicycle sharing systems, European Journal of Operational Research 245 (3) (2015) 667–679.

[20] M. Defraery, I. V. Nieuwenhuyse, Staffing and scheduling under nonstationary demands for a service: a literature review, Omega 58 (2016) 4–25.

[21] J. G. Skellam, The frequency distribution of the difference between two poisson variates belonging to different populations, Journal of the Royal Statistical Society 109 (3) (1946) 296.

[22] J.-J. Salazar-González, B. Santos-Hernández, The split-demand one-commodity pickup-and-delivery travelling salesman problem, Transportation Research Part B: Methodological 75 (2015) 58–73.

[23] M. Gendreau, J. Nossack, E. Pesch, Mathematical formulations for a 1-full-truckload pickup-and-delivery problem, European Journal of Operational Research 242 (3) (2015) 1008–1016.

[24] G. Nagy, N. Wassan, M. Speranza, C. Archetti, The vehicle routing problem with divisible deliveries and pickups, Transportation Science 49 (2) (2015) 271–294.

[25] I. Gribkovskaia, O. Halskau, M. Kovalyov, Minimizing takeoff and landing risk in helicopter pickup and delivery operations, Omega 55 (2015) 73–80.

[26] H. Hernández-Pérez, J.-J. Salazar-González, The multi-commodity pickup-and-delivery traveling salesman problem, Networks 63 (1) (2014) 46–59.