A modeling framework for the dynamic management of free-floating bike-sharing systems

Leonardo Caggiani[a], Rosalia Camporeale[a], Michele Ottomanelli[a], Wai Yuen Szeto[b]

[a]Dept. of Civil, Environmental, Land, Building Engineering and Chemistry (DICATECh)
Polytechnic University of Bari, Viale Orabona, 4 - 70125 Bari, Italy
{leonardo.caggiani; michele.ottomanelli; rosalia.camporeale}@poliba.it

[b]Dept. of Civil Engineering, The University of Hong Kong
Rm 618, Haking Wong Building, Pokfulam, Hong Kong
ceszeto@hku.hk

Abstract. Given the growing importance of bike-sharing systems nowadays, in this paper we suggest an alternative approach to mitigate the most crucial problem related to them: the imbalance of bicycles between zones owing to one-way trips. In particular, we focus on the emerging free-floating systems, where bikes can be delivered or picked-up almost everywhere in the network and not just at dedicated docking stations. We propose a new comprehensive dynamic bike redistribution methodology that starts from the prediction of the number and position of bikes over a system operating area and ends with a relocation decision support system. The relocation process is activated at constant gap times in order to carry out dynamic bike redistribution, mainly aimed at achieving a high degree of user satisfaction and keeping the vehicle repositioning costs as low as possible. An application to a test case study, together with a detailed sensitivity analysis, shows the effectiveness of the suggested novel methodology for the real-time management of the free-floating bike-sharing systems.

Keywords: free-floating bike sharing systems; spatio-temporal clustering; non-linear autoregressive neural network forecasting; Decision Support System; dynamic fleet relocation.

1 Introduction

Bike sharing systems (BSSs) offering a mobility service by means of public bikes available for shared use are becoming increasingly popular in urban environments. These shared systems provide city users with an alternative and more sustainable carbon-free mode of transportation (especially suited for short-distance trips), significantly reduce traffic congestion, air pollution, noise, and the need for parking in city centers, and support a greener growth of urban environments.

BSSs allow users to take a bike from a particular position in the network, use it for a journey, give it back closer to their destination, and pay either according to the time of usage or other specific pricing policies. Thanks to BSSs, it is possible to have access to those city areas that are not allowed to other kinds of vehicles and create an alternative connection with public transit networks. In the literature, it is possible to find different studies that suggest ways to enhance the efficiency of BSSs from tactical or strategical view-points. Some authors have investigated the optimal location of stations (García-Palomares, Gutiérrez, and Latorre, 2012; Martinez et al., 2012), the network design of bike lanes (Lin and Yang, 2011; Vogel and Mattfeld, 2011; Saharidis, Fragkogios, and Zygouri, 2014), and their capacity levels (Romero et al., 2012; Garcia-Gutierrez, Romero-Torres, and Gaytan-Iniestra, 2014).

Despite their undeniable qualities, BSSs are mainly used for medium-short distances and for one-way trips. Such usage leads to an unbalanced distribution of bikes in time and space. This sometimes makes it impossible for users to find a bike when they want to start their journey, and/or the leave the bike at their preferred destinations due to full stations. Hence, aiming to increase the use of the system and user satisfaction, bike sharing service providers have to focus on the efficiency of rebalancing operations (Shui and Szeto, 2015; Li et al., 2016) and ensure that the number of bikes, as well as the number of free docking slots at each station, are periodically restored to predefined target values.

Essentially, there are two different relocation strategies: the user-based approach, where users are induced to leave their bike at a certain station in order to balance the global distribution of bikes; and the operator-based one, where the relocation process is performed by the BSS service staff.

The operator-based reallocation problem is known as a Pickup and Delivery Problem (PDP). The application of this optimization problem to BSSs has recently attracted the interest of many researchers and practitioners in this field. It can be modeled by a dynamic or a static approach (Chemla et al., 2013). In the static version, which is usually performed during the night (when the system is closed, or the bike demand is very low), a snapshot of the number of bikes in each zone is considered and utilized to plan the redistribution. On the other hand, the dynamic relocation is performed during the day when the status of the system is rapidly changing and takes into account the real-time usage of the BSS. In the dynamic case, the relocation plan is strongly supported by forecasting techniques.

In the literature, it is more common to find authors dealing with a static approach (Raviv, Tzur, and Forma, 2013; Lu, 2013; Schuijbroek, Hampshire, and van Hoeve, 2017). They solve the problem by adopting different methods, such as an iterated tabu search (Ho and Szeto, 2014a), a branch and bound algorithm (Kadri, Kacem, and Labadi, 2016), an enhanced chemical reaction optimization such as the one proposed by Szeto, Liu, and Ho (2016), and the latest hybrid large neighborhood search by Ho and Szeto (2017). Often, they have studied static bike rebalancing benefit using methodologies like forecasting and inventory optimization (Chemla, Meunier, and Calvo, 2013; Ho and Szeto, 2014b; Dell'Amico et al., 2014; Forma, Raviv, and Tzur, 2015; Erdoğan, Battarra, and Calvo, 2015).

Solutions to a dynamic public bike sharing balancing problem have been proposed at first by Contardo, Morency, and Rousseau (2012), who suggested a mathematical formulation on a space-time network, and after that by Caggiani and Ottomanelli (2012 and 2013), Benarbia et al. (2013), and Vogel, Saavedra, and Mattfeld, (2014). Regue and Recker (2014) proposed a novel approach, that results proactive instead of reactive, as the bike redistribution occurs before inefficiencies are observed, increasing system performance and, potentially, customer satisfaction; they used the outputs of a machine learning technique to decompose the inventory and the routing problem. Some other works (Pfrommer et al., 2014; Fricker and Gast, 2016) have examined incentive policies aimed at encouraging users to return bikes to the locations that need bikes.

In recent years, innovative systems for the management of bike sharing, called Free-Floating Bike Sharing Systems (FFBSSs), are gradually showing up. At the beginning they have only been applied to medium-small areas or pilot projects; however, particularly during the last two years, they have started to be implemented in many large cities in China, England, Netherlands, and other countries. In FFBSSs, users can lock shared bikes to an ordinary bike rack or to any pole, removing the concept of fixed stations, and avoiding the necessity of docking stations and kiosk machines (Pal and Zhang, 2017) with relevant physical and information communication technology infrastructures.

The position of the available bikes is identified through a satellite receiver located on each bicycle, or by means of users' smartphones (using the smartphone for locking/unlocking the bike's smart lock, the bicycle location is transmitted to the central system). Users can detect a bike through web or mobile applications. More information on how these systems work can be found on websites of FFBSSs providers such as SocialBicycle (SoBi).

The main advantages that make FFBSSs more attractive than the current BSSs generations are two-fold: (1) reduced system start-up costs and (2) the possibility to find and leave the bike almost anywhere as long as it is a location accessible to all users and within the served area. For FFBSSs, the problem of full stations does not exist, since bikes can be dropped off almost everywhere and the number of ordinary racks that could be installed is very high, thanks to their low acquisition, installation and maintenance costs. However, as in BSSs, bike distribution can be unbalanced in time and space and there is no guarantee of finding a free bike near the origin of the desired trip. Furthermore, as bikes are scattered across the territory (because of the opportunity to leave the bike anywhere) and their positions may change during the day and from day to day, the relocation process of these systems is even more challenging than that of the traditional BSSs.

To the best of our knowledge, there are only three recent papers on FFBSSs. Reiss and Bogenberger (2015 and 2016) have proposed GPS-data analyses of FFBSS with the suggestion of a relocation strategy and a validation method to show the effects of rebalancing operations through the application of their proposed methods on Munich's FFBSS. Pal and Zhang (2017) have presented a hybrid nested large neighborhood search with variable neighborhood descent algorithm to solve a static rebalancing problem.

In this paper, we propose a new, comprehensive, dynamic, and operator-based bike redistribution methodology that starts from the prediction of the number and position of bikes over an FFBSS operating area and ends with a Decision Support System (DSS) for the relocation process. The DSS is activated at constant gap times during the day in order to carry out dynamic bike redistribution, mainly aimed at achieving a high degree of user satisfaction and keeping the costs of repositioning operations as low as possible. As a matter of fact, it is important to point out that maximizing customer satisfaction is one fundamental contribution to the maximization of profitability for a company: a greater user satisfaction implies the system being used more intensively from any user. Moreover, satisfied customers may recommend the service to other possible customers, increasing the potential for additional revenue and profit, and playing an important part in shaping attitudes towards the company and the service.

With this work (the entire procedure is outlined in Fig.1), we first propose to investigate the spatio-temporal correlation pattern of different zones in a city by adopting clustering techniques on the historical usage records of bike sharing demand. After that, for each obtained spatio-temporal cluster, we apply a non-linear autoregressive neural network demand forecasting model to determine the number of bikes needed. Then, a DSS for dynamic free-floating bike redistribution is presented in order to determine optimal repositioning flows and distribution patterns.

An FFBSS simulator (which is used when real data are not available) is also proposed and explained in detail: it has been adopted in order to generate a random demand for picked-up bikes in the system and look at its operation during the day with and without the contribution of the relocation processes. Finally, a numerical application together with a deep sensitivity analysis is provided, aimed at a complete evaluation of effectiveness and efficiency of the proposed methodology. We observe that it results possible to enhance a bike-sharing system by means of a dynamic relocation process: it could be possible to find good solutions to apply to different realities, providing operators with a valuable intervention strategy on their city.
The paper closes with conclusions and suggestions for further research.

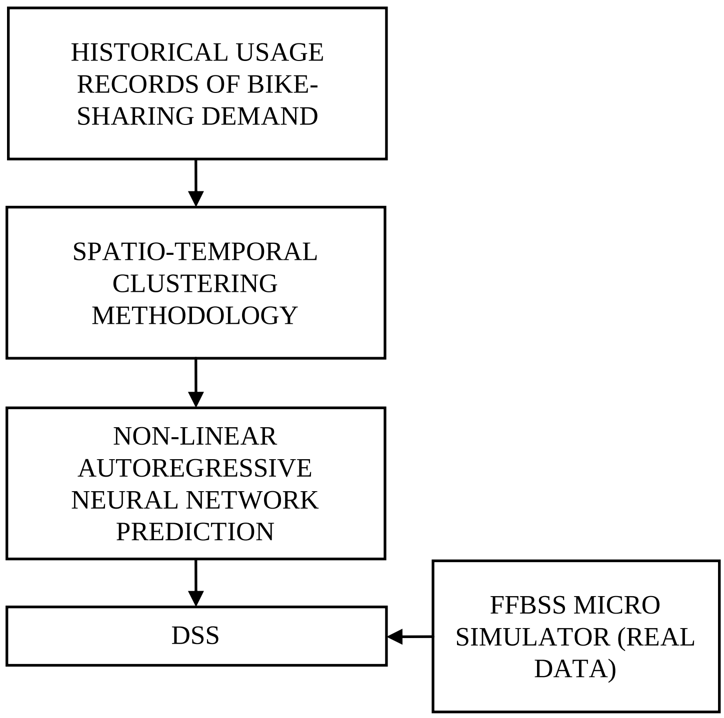

Fig. 1 Procedural scheme.

2 Notation

For clarity, this section summarizes symbols and mathematical notations adopted in the paper. Their units of measurements are reported within round brackets at the end of the row (note that where not specified, the variable is dimensionless). They are grouped into three main categories: clustering and forecasting, Decision Support System, and Bike Sharing System Simulator.

Clustering and forecasting

n	total number of temporal clusters vectors
T	temporal cluster set
$\mathbf{T}_k$	temporal cluster vector k belonging to the set T, with $k \in [1, 2, \ldots, n]$
$\bar{\varphi}$	total number of centroids
φ	generic centroid, with $\varphi \in [1, 2, \ldots, \bar{\varphi}]$
w_k	total number of $\mathbf{T}_k$ elements, $w_k \in [1, 2, \ldots, \varphi]$
t_k^y	generic element of a temporal cluster vector $\mathbf{T}_k$ with $y \in [1, 2, \ldots, w_k]$
Ξ_k	set of remaining t_k^y not included in the satisfactory clusterization
S_k	spatio-temporal cluster set associated to $\mathbf{T}_k$
m_k	total number of S_k elements, $m_k \in [1, 2, \ldots, \varphi]$
$\mathbf{S}_k^u$	spatio-temporal cluster vector belonging to the set S_k with $u \in [1, 2, \ldots, m_k]$
η_k^u	total number of $\mathbf{S}_k^u$ elements, $\eta_k^u \in [1, 2, \ldots, \varphi]$
S_k^{uh}	generic element of a spatio-temporal cluster vector $\mathbf{S}_k^u$ with $h \in [1, 2, \ldots, \eta_k^u]$
$\bar{\vartheta}$	total number of spatio-temporal cluster iterations
ϑ	generic spatio-temporal cluster iteration, $\vartheta \in [1, 2, \ldots, \bar{\vartheta}]$
α	minimum allowable value of m_k
b_k^u	total number of $S_{k'}^{uh}$, $\forall\, k' \neq k$, belonging to temporal clusters $\mathbf{T}_{k'}$ inside the spatial boundary of the cluster $\mathbf{S}_k^u$
β	maximum allowable value of b_k^u
D	number of delays (past values to consider in NARX)
Δz	width of each time interval (min)
Δt	width of the time interval in the range of an exact past value (min)

Decision Support System

N	number of available bikes (bikes)
$N(\varphi)$	number of available bikes for the centroid φ (bikes)
$\bar{f}$	total number of one-day DSS activation
f	DSS activation index
$\Delta\mu$	DSS activation time span (min)
$\Delta\lambda$	prediction time span (min)
$\Delta\upsilon_f$	prediction interval, $\Delta\upsilon_f = [1 + f \cdot \Delta\mu; 1 + f \cdot \Delta\mu + \Delta\lambda]$ (min)
Γ	total number of spatio-temporal clusters with $\Gamma = \sum_{k=1}^{n} m_k$
UNT_ξ	unavailability threshold for the spatio-temporal cluster ξ, with $\xi \in [1, 2, \ldots, \Gamma]$
χ_ξ^f	minimum number of bikes in $\Delta\upsilon_f$ for the spatio-temporal cluster ξ
ψ_ξ^f	maximum number of bikes in $\Delta\upsilon_f$ for the spatio-temporal cluster ξ
ρ^f	set of receiver spatio-temporal clusters for $\Delta\upsilon_f$
σ^f	set of spatio-temporal clusters for $\Delta\upsilon_f$

$\mathbf{R}^f$	relocation matrix for $\Delta\upsilon_f$
r_{ij}^f	generic element of the relocation matrix $\mathbf{R}^f$ (row *i*, column *j*) with $i \in [1, 2, \ldots, \Gamma]$ and $j \in [1, 2, \ldots, \Gamma]$
U_ξ^f	unfavorable time for the spatio-temporal cluster ξ for $\Delta\upsilon_f$ (time steps)
U^f	sum of unfavorable times for all the spatio-temporal clusters in $\Delta\upsilon_f$, that is $\sum_{\xi=1}^{\Gamma} U_\xi^f$ (time steps)
g_ξ	threshold number of bikes for the spatio-temporal cluster ξ
P^f	set of relocation paths for $\Delta\upsilon_f$
$\overline{tr}$	total number of trucks (trucks)
P_{tr}^f	generic element of the relocation paths set with $tr = 1, 2, \ldots, \overline{tr}$
v	truck average speed (m/s)
d^f	sum of relocation path distances for $\Delta\upsilon_f$ (m)
E^f	sum of maximum widths of all the spatio-temporal clusters visited by all trucks for $\Delta\upsilon_f$ (m)
cb	average bike load/unload lock/unlock time (min)
nr_{tr}^f	number of bikes to relocate for each truck for $\Delta\upsilon_f$ (bikes)
NR^f	total number of relocated bikes for $\Delta\upsilon_f$ (bikes)
c_{tr}	capacity of truck tr (bikes/truck)
q_{tr}^f	generic cycle of pick-up/delivery for truck tr for $\Delta\upsilon_f$
$\bar{q}_{tr}^f$	total number of cycles of pick-up/delivery for truck tr for $\Delta\upsilon_f$ (bikes)

Bike Sharing System Simulator

dw	day of the week (working day, *dw* = 1, or weekend/holiday day, *dw* = 2)
$\bar{\tau}$	total number of discrete time intervals
τ	generic discrete time interval, with $\tau \in [1, 2, \ldots, \bar{\tau}]$
Br_ξ^τ	real available bikes for the spatio-temporal cluster ξ in τ (bikes)
Bp_ξ^τ	predicted available bikes for the spatio-temporal cluster ξ in τ (bikes)
Ψ_ξ	maximum number of bikes that a cluster ξ could accept (i.e., cluster capacity) (bikes)
ζ_ξ^f	minimum number of bikes predicted for the spatio-temporal cluster ξ in the interval $\Delta\upsilon_f$ (bikes)
Φ_ξ^f	maximum number of bikes predicted for the spatio-temporal cluster ξ in the interval $\Delta\upsilon_f$ (bikes)
$\bar{p}$	total number of types of zones or centroids
p	generic type of zone, with $p \in [1, 2, \ldots, \bar{p}]$
φ_p	centroids belonging to a generic type of zone *p*
$\bar{\delta}_{p,dw}^\tau$	maximum value of pick-up bike demand in τ for *p* and for *dw* (bikes)
$\delta_{p,dw}^\tau$	possible values of pick-up bike demand in τ for p and for *dw*, with $\delta_{p,dw}^\tau \in [0, 1, 2, \ldots, \bar{\delta}_{p,dw}^\tau]$ (bikes)
$\mathbf{Db}_\xi$	pick-up bike demand vector for the spatio-temporal cluster ξ
Lu_ξ^f	lost users for the spatio-temporal cluster ξ in $\Delta\upsilon_f$ (users)
Lu^f	sum of lost users for all the spatio-temporal clusters in $\Delta\upsilon_f$, that is $\sum_{\xi=1}^{\Gamma} Lu_\xi^f$ (users)

3 Proposed spatio-temporal forecasting model

In this section, we propose a methodology to correlate the spatio-temporal analysis of FFBSS usage with the correct prediction of the trend of available bikes during the day in each cluster. This strategy can be useful in simplifying and enhancing the bike relocation process, helping the reduction of FFBSS management costs.

Since there are no fixed docking stations in FFBSS, bikes need to be aggregated as if they belong to virtual stations in order to analyze the spatio-temporal trend of picked-up and delivered bikes.

We start with the idea of studying a city with an FFBSS operating on it. As this city is divided into census districts (the smallest data units), the first operation to carry out involves the aggregation of these districts into larger zones (i.e. zoning). Each zone has its own centroid φ. To build a solid forecasting model, the next thing to do is select a set of 'uniform' days, from the point of view of the season, type of the day (working days or weekend/holiday days), and weather conditions. Starting from this data selection, for each set of 'uniform' days, it is possible to know the number of available bikes at each moment of the day for every zone (zone temporal pattern). We can alternatively state that a temporal pattern/trend representative of the number of bikes within the zone at each moment of the day is associated to each centroid φ.

This first aggregation, from census districts to larger zones, is simply spatial. In order to obtain predictions as correct as possible, it is desirable to aggregate these zones again, taking into account their temporal patterns and the similarities among their trends. This is the first step of the methodology explained in greater detail in the following subsection.

3.1 Spatio-temporal clustering method

The aim of clustering analysis of a dataset is to organize a collection of different patterns into a smaller number of homogeneous groups, without any prior knowledge. A good clustering solution is one that is able to reach a high correlation among the elements of a single cluster and a low inter-correlation between distinct clusters.

Clustering methods have been widely used in the literature to explore activity patterns connected to BSS usage, with a wide range of final purposes. This happens because datasets collected on such systems are usually sizeable. Therefore, it is difficult to try to gain knowledge from them without the help of a method capable of summarizing the underlying information.

Some authors (Vogel, Greiser, and Mattfeld, 2011; Côme, Randriamanamihaga, and Oukhellou, 2014) have found that each cluster of bike stations seems to be closely related to city functions, like employment, leisure, transportation. These results can be helpful for a variety of applications, including urban planning and the choice of a business location. This correlation between bike station activity and external factors has also been modeled (Wang et al, 2015). Looking at a free-floating BSS, although there are no stations, the origins of each trip and the destination choices are strictly related both to the generation capacity of the origins and to the attractiveness of the destinations. For example, we are expecting to find more bicycles around a local market, compared to an area with a low attractiveness. The main difference that we can point out if, in this framework, we compare a station-based system with a free-floating one, is that we are not finding anymore all the bikes together in a single station (i.e., the one in front of the main entrance of the market), but they will be spread around the market area, according to the specific choice of each user. Essentially, we can state that in an FFBSS a larger area ideally corresponds to a single BSS station. Consequently, it is true to assert that a certain correlation between bicycle locations and external factors does exist also in free-floating systems.

Different studies have also focused on the spatio-temporal data correlation, considering it a factor of paramount importance in affecting bike demand in the system. Froehlich, Neumann, and Oliver (2009) provided a spatio-temporal analysis of Barcelona's shared cycling system, identifying shared behaviors across stations, finding how these behaviors relate to location, neighborhood and time of the day. More recently, Han, Côme, and Oukhellou (2014) correlated the historical usage records of Paris' bike sharing system at both the spatial and temporal scale, integrating this analysis into forecasting goals and underlining how this represents necessary information for accurately predicting bike demand at each station.

The preliminary spatio-temporal analysis of the suggested methodology involves the definition of two main categories of clusters: *temporal clusters* (set T), and *spatial clusters* associated to each element $\mathbf{T}_k$ of the set T, called spatio-temporal clusters (set S_k).

Knowing the zone temporal patterns associated to each centroid φ and setting the total number of temporal clusters n, they can be clustered according to their temporal trends. In this way, it can be stated that each zone/centroid belonging to a given $\mathbf{T}_k$, $k \in [1, 2, \ldots, n]$, is a generic element t_k^y of it with $y \in [1, 2, \ldots, w_k]$. At first, a discrete wavelet transformation (Guan and Feng, 2004; Zhang et al., 2008) helps in the analysis of signals (in this case, they are the temporal trends representing the number of bikes for each zone) (Vlachos et al., 2003); then, the filtered data are aggregated into a given number of temporal clusters $\mathbf{T}_k$, applying hierarchical clustering methodology (Caggiani et al., 2017). Discrete wavelets are able to de-noise and compress the signals and are often used as a preprocessing step before clustering (Antoniadis and Brossat, 2013). We adopted the hierarchical clustering to perform the temporal clustering, due to its inherent data mining property of grouping the data simultaneously by creating a cluster tree (i.e. dendrogram): changing the number of clusters, the dendrogram remains the same.

The next step aims at geographically aggregating groups of zones (Xu et al., 2013; Lee, Wang and Wong, 2014) belonging to the same T_k, creating a certain number of spatio-temporal clusters $\mathbf{S}_k^u$, $u \in [1, 2, \ldots, m_k]$, associated to each temporal cluster. We propose to operate this second clustering using a k-means algorithm. The k-means method is a widely-used clustering technique, the goal of which is seeking to minimize the average squared distance between points in the same cluster. Despite not being capable of guaranteeing accuracy, it is very appealing in practical applications, thanks to its simplicity and computing speed (see MacQueen 1967, and Arthur and Sergi, 2007 for further details). In our proposed approach, the computational speed of k-means, higher compared to the one of the hierarchical clustering (Kaur and Kaur, 2013), resulted to be essential due to the complexity associated with the resolution of the bilevel problem (from Eq. (1) to (4)).

In general, it can be asserted that every $\mathbf{S}_k^u$ ideally should have a size comparable with the distance that typical users are willing to travel on foot (within their origin zones), so that at any time they will be able to easily reach the closest bike. On the other hand, looking at the same situation from the operator's point of view, having too small spatio-temporal cluster $\mathbf{S}_k^u$ could involve disproportionately high relocation costs. To clarify this concept, consider the following example: if two small $\mathbf{S}_k^u$ were to be aggregated into a larger one, the bike need of one of them could be satisfied by the bikes available in the other (adjacent) one: in conclusion, in the bigger cluster obtained by merging the two smaller $\mathbf{S}_k^u$, there would be no necessity to relocate any bikes, and there would be lower relocation costs. These savings could compensate for the eventual loss of users.

At the same time, having too large clusters $\mathbf{S}_k^u$, it could make it necessary to relocate within each of them, to avoid risking a greater number of users being unsatisfied, not being able to find a bike to pick up within a reasonable distance from the origin point of their trip. A proper size/area for each $\mathbf{S}_k^u$ should be established (and calibrated) conveniently through a model able to take into account both the total relocation costs and the number of unsatisfied/lost users (further research, currently in progress, is aimed at implementing this model).

However, as a first approximation, the following bilevel optimization problem, is able to provide a reasonable number of $\mathbf{S}_k^u$ related to each S_k:

$$\min m_k \tag{1}$$

s.t.

$$\min \mathrm{Dist}\,(t_k^y, m_k) \tag{2}$$

$$m_k \geq \alpha \tag{3}$$

$$\max b_k^u \leq \beta \quad (4)$$

The upper-level objective (1) is aimed at minimizing the total number m_k of $\mathbf{S}_k^u$ associated to each S_k. The lower level objective (2) represents the k-means optimization, that is, the minimization of the distance between the positions of the centroids of each the spatio-temporal cluster, and the elements t_k^y belonging to the temporal ones. Eq. (3) means that m_k has to be greater than (or equal to) a positive integer coefficient α. The coefficient α could be more appropriately calibrated according to the above considerations regarding the proper size/area of each $\mathbf{S}_k^u$; in any case, its purpose is to avoid $\mathbf{S}_k^u$ being too small. The last constraint (4) forces the maximum value of the total number of $S_{k'}^{uh}$, $\forall\, k' \neq k$, belonging to temporal clusters $\mathbf{T}_{k'}$ inside the spatial boundary of the cluster $\mathbf{S}_k^u$ to be less than or equal to a positive integer coefficient β; as an example, the lower the value given to β, the higher will be the number of spatial clusters associated to S_k. β should also be conveniently calibrated according to the given case study. If it is set to 0, it leads to an unfeasible problem; on the other hand, if it is too large, it will involve a lot of overlapping among areas of $\mathbf{S}_k^u$ belonging to different S_k.

At the end of this procedure, we obtain m_k number of spatio-temporal clusters $\mathbf{S}_k^u$ associated to each value of k. The spatial boundary of each $\mathbf{S}_k^u$ is (usually) a polygon, having as vertices some of the elements t_k^y belonging to T_k. We consider these spatial clusters satisfactory if the number of these perimeter vertices is greater than or equal to 3 (that is, the minimum number that defines a polygon). However, it may happen that some $\mathbf{S}_k^u$ consist of only one or two elements t_k^y; hence, for these remaining t_k^y not included in the satisfactory clusterization (and constituting the set Ξ_k), we decided to repeat the optimization for a number of times equal to $\bar{\theta}$. This procedure is carried out to take advantage of an inherent feature of k-means clusterization, that is the arbitrary choice of the initial center of each cluster (Arthur and Sergi, 2007). Each iteration selects different centers, and consequently there are more possibilities to further aggregate the remaining t_k^y. In any case, it could still happen that, at the end of all the iterations, some spatial clusters consist of only one or two elements.

Within the optimization model, there is a further general external constraint to be satisfied: in order to avoid an excess of overlapping among clusters it must be guaranteed that the intersection areas between $\mathbf{S}_k^u$ (belonging to different $\mathbf{T}_k$) is below a given threshold. Finally, we achieve the optimal set of spatio-temporal $\mathbf{S}_k^u$ clusters for each $k \in [1, 2, \ldots, n]$. It is possible to assert that elements belonging to the same $\mathbf{S}_k^u$ are close to each other in space and it has been demonstrated by Vogel, Greiser, and Mattfeld (2011), and Côme, Randriamanamihaga, and Oukhellou (2014), that temporal clusters belonging to the same category are usually adjacent spatially. Furthermore, neighboring zones are likely to have a similar capacity to generate and attract potential users. Spatio-temporal $\mathbf{S}_k^u$ clusters are the starting point to continue to the next step of the proposed methodology.

The overall clustering procedure is shown in the flowchart in Fig. 2.

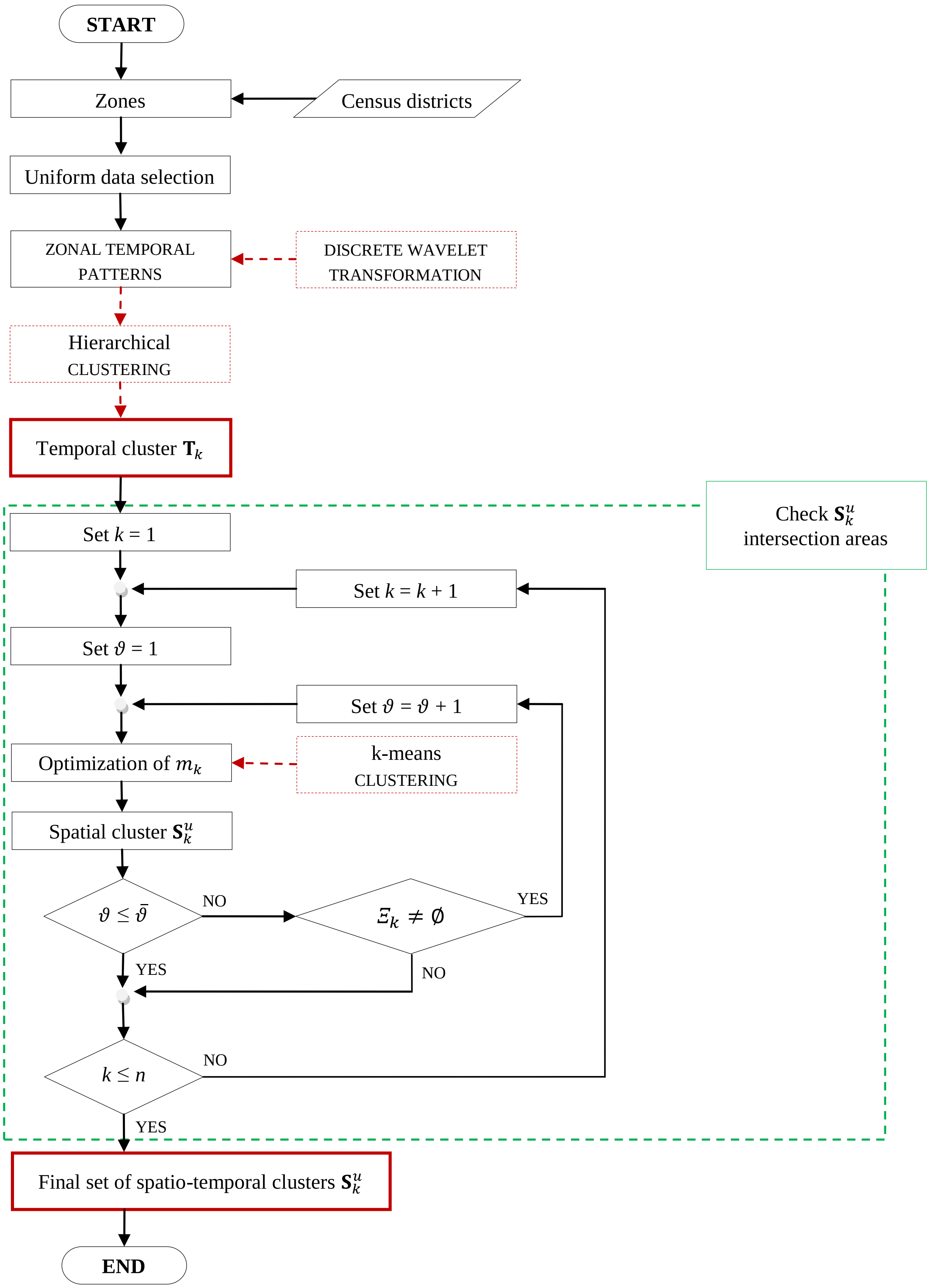

Fig. 2 Flowchart of spatio-temporal clustering methodology.

3.2 Nonlinear Autoregressive Neural Network Prediction

An adequate solution for the dynamic rebalancing problem is strictly connected to the correct prediction of the usage of the system: forecasting the usage patterns of an FFBSS is beneficial for both users, able to plan their trips better, and operators, who can better manage the system by manually relocating bikes between the zones to improve the customers' experience. Different studies on the prediction of BSSs have already been carried out, often through methods similar to the ones widely used in road traffic control (Vlahogianni, Karlaftis, and Golias, 2014; Mori et al., 2015). There are approximately 30 linear and non-linear predictive methods (Zeng et al., 2008): in the linear prediction approaches the prediction is based on the main assumption of linearity and stationarity to obtain future trends (Smith and Demetsky, 1997); whereas the non-linear prediction techniques, with a higher computing complexity, are able to capture the non-linear features of transportation systems, ensuring more accurate forecasting (Yu et al., 2005; Sapankevych and Sankar, 2009; Sun, Leng, and Guan, 2015).

Among them, the artificial neural networks represent a flexible method which is good at recognizing complex, non-linear patterns, without preliminary knowledge about the relationships between input and output variables (Wei and Chen, 2012). In particular, Kek, Cheu, and Xu (2005) and Cheu et al. (2006) made a comparison between different forecasting techniques applied to a car-sharing context (similar in its basic features to the problem under study). What emerged is that neural networks, thanks to their characteristics and adaptability, seem to be the ones with the best behavior among the compared techniques at dealing with this kind of problem.

In order to forecast the bike use in each $\mathbf{S}_k^u$ belonging to the final optimal spatio-temporal set, we decide to adopt a Nonlinear Autoregressive Neural Network with eXogenous inputs (NARX) approach (Leontaritis and Billings, 1985), suitable for forecasting data starting from a historical series. It has been demonstrated that they are well suited for modeling nonlinear systems such as time series (Connor, Atlas and Martin, 1992); as a matter of fact, these dynamic neural networks (defined as Recurrent Neural Networks (RNN), see Bianchi et al., 2017) can be trained to predict future values of a time series from past values of that time series (and from past values of a second external time series). In particular, in our framework, the first time-series is related to the number of bikes available in each cluster, while the time of the day is an exogenous (external) input. The remaining nominal and ordinal variables (season, day, and so on) have to be used upstream, dividing the entire dataset into multiple subsets, each one relative (for example) at the same season and/or same day of the week; every subset have to be used separately.

Given a time series representing the number of available bikes in a temporal interval for each spatio-temporal cluster $\mathbf{S}_k^u$, we aim to determine an optimal network structure for the NARX and train the network to minimize a performance evaluation function, such as the Mean Square Error (MSE) that is the average of the squared differences between the forecasted and observed data. We have been able to tune the parameters for this approach correctly thanks to a series of empirical analysis that we have run.

For each $\mathbf{S}_k^u$, we know the aggregate number of available bikes over time; namely, at every instant of time there is an associated exact number of bikes. Therefore, it is possible to summarize this trend by means of a graph, able to represent how the number of bikes changes over the time. The NARX starts from this series of past values to predict the number of bikes at a given moment in the future. More specifically, one of the main parameters to set in the NARX is the number of delays D, i.e. the number of past values to consider. The time step between two subsequent past values has a certain temporal width: we call it Δz. Setting D and Δz, we are able to predict the number of bikes that will be available in $\mathbf{S}_k^u$ after a time step of Δz following the moment we are doing the forecasting for (Fig. 3 – with $D = 6$).

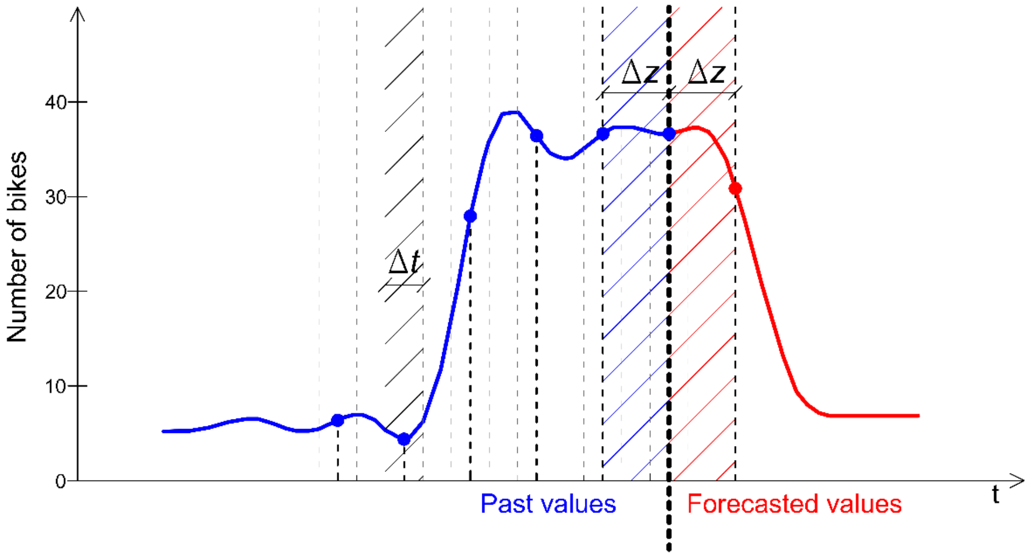

Fig. 3 NARX time steps.

Proceeding in this way, basically, we are training the neural network on the basis of *D* past exact values. However, we can also follow some alternative ways, and through a sensitivity analysis understand which one is the most suitable to forecast values for the chosen case study. At Δt (see Fig.3), an interval of time having each past exact value as its midpoint (extreme point of a Δz interval), it is possible to identify:

- the maximum value of bikes in Δt;
- the average value of bikes in Δt;
- the minimum value of bikes in Δt.

After setting the Δt value, starting from the same historical series of every spatio-temporal cluster, for each one of this cluster, we can adopt 4 different neural network starting datasets. These datasets are obtained by filtering the values within the Δt intervals, considering the past exact values, or the maximum values, or the average values, or the minimum values.

It is possible to use different methodologies to deal with the overfitting associated with the trained network. We propose the adoption of the early stopping strategy with the random data division of the input (the same used in the numerical application presented in this paper). In particular, each one of the four datasets has to be respectively divided into three subsets: training subset, validation subset, and test subset. The first one is used to train the neural network: it will stop (according to the early stopping strategy) when the performance on the validation subset increases for a set number of iterations. The test subset, on the other side, is used as an independent set, in order to verify the generalization of the network. Aiming at the highest efficacy of this checking procedure, the three subsets are extracted casually (random data division), so that the test subset does not have a similar arrangement to the training one.

Starting from the 4 datasets (exact, maximum, average and minimum values), for each cluster, we can train, validate, and test 4 different neural networks, aiming at forecasting (and comparing) the future trends of the available bikes, and leaving to the analyst the possibility to select which option is the most suitable to satisfy his/her purposes.

To be capable of predicting the trend of available bikes for all the subsequent day (and not just the exact value after a given Δz), essentially, we need to make different predictions for different (increasing) Δz intervals. In this way, we can be aware of the number of bikes, step by step, throughout the time in which we want to perform the forecasting.

While we are predicting the number of bikes for a further Δz, our database will be constantly updating values from the real-time usage of the FFBSS. Indeed, we can use these new values to perform more reliable forecasting. Consequently, after a certain period of time, we can predict the future trend on the basis not only of aforementioned past values but also using newly collected values occurred until that instant.

4 Proposed relocation model for free-floating BSS

In this section, we present a Decision Support System (DSS) for a dynamic real-time bike redistribution process. The objective of the model is to achieve a high degree of user satisfaction (assuming that it increases together with the probability of finding an available bike in any spatio-temporal cluster at any time). At the same time, we aim to keep the vehicle repositioning costs as low as possible. The proposed model considers a dynamic variation of the demand for bikes and micro-simulates the FFBSS in space and time, determining the optimal repositioning flows and bike distribution patterns by explicitly considering the choice of truck route between the centroids of the clusters.

The proposed DSS is activated $\bar{f}$ times at a constant gap time $\Delta\mu$, and it starts with the forecasting module of the bikes usage in each $\mathbf{S}_k^u$ belonging to the final optimal spatio-temporal set. Basically, we need to input into the DSS the prediction of the number of bikes in every cluster $\mathbf{S}_k^u$, for a given time interval $\Delta\upsilon_f = [1 + f \cdot \Delta\mu; 1 + f \cdot \Delta\mu + \Delta\lambda]$ where $f \in [1, 2, \dots, \bar{f}]$ and $\Delta\lambda$ is the prediction time span; in this $\Delta\upsilon_f$ we have to evaluate the objective functions (Eqs. (5) and (11)) specified in the model. The outputs are the relocation matrix ($\mathbf{R}^f$), which suggests to the operator how many bikes need to be repositioned between the clusters, and the relocation paths (P^f) (i.e. the routes that trucks have to follow to carry out their job).

The proposed DSS is based on two optimization steps: the first generates the relocation matrix ($\mathbf{R}^f$) as an output, aiming at the minimization of both the unfavorable times and the total number of lost users. Let ξ be a generic spatio-temporal cluster (regardless from the set S_k to which it belongs), U^f be the unfavorable times, which is the duration of the time (in time steps) which clusters have a number of bikes below a given threshold g_ξ. This threshold g_ξ should be set proportionally with the width/area of the cluster. This to be sure that the average probability of finding a bike available near the origins of the trips is similar for all clusters, then the ones with a larger area should have a minimum number of bikes higher than the smaller clusters.

Let Lu^f, be the number of lost users or the number of users for all clusters that decide not to pick-up a free-floating bike because they are not able to find one nearby. Essentially, $\mathbf{R}^f$ is an Origin-Destination (O-D) matrix: in each cell, we find the number of bikes to be picked-up from each origin (O) to every destination (D). This first optimization takes into account both users' and the operator's perspectives, as it is a common interest to minimize both U^f and Lu^f. The second optimization (operator's perspective) generates the relocation paths (P^f) as an output, aiming at the minimization of the travel time (costs) for trucks that provide the relocation service.

Before starting, it is necessary to check whether the forecasted data are actually reliable. Therefore, we decided upon a filtering procedure. Then, it is necessary to calculate the increase (or decrease) in the real value (i.e., already happened) of available bikes in every cluster, for each time interval τ and for all the time series (a single difference is for example $|Br_\xi^\tau - Br_\xi^{\tau-1}|$). Let BR_ξ be the maximum absolute value of the just mentioned differences for a spatio-temporal cluster ξ. We need to compare BR_ξ with the increase (or decrease) that has occurred in the forecasting trend (that is $|Bp_\xi^\tau - Bp_\xi^{\tau-1}|$) for each ξ and for each τ. If, for a given ξ at a given τ of the prediction, the gap with the number of bikes of the preceding time interval exceeds the maximum gap which occurred in the real past trend (i.e., $|Bp_\xi^\tau - Bp_\xi^{\tau-1}| > BR_\xi$), we filter the predicted value at the time interval τ, leveling it with the value forecasted at the previous instant (i.e., we set $Bp_\xi^\tau = Bp_\xi^{\tau-1}$). A special case happens if τ is the first prediction time interval: in this case, Bp_ξ^τ is set to the last known real value, that is, the final number of available bikes registered in the real trend. The aim of adopting this strategy is to avoid peaks (or drops) overstated in the forecasted trend.

Each generic spatio-temporal cluster ξ can be a donor or a receiver, i.e., it needs to give some of its bikes or to accept them during the relocation process. Before starting the first optimization, at a given τ, according to the forecasted trend, we can establish which spatio-temporal cluster is a donor and which one is a receiver, as it is not possible for them to play both these roles simultaneously. For this reason, it is helpful to establish a so-called unavailability threshold (UNT_ξ), to avoid all the bikes of a donor cluster being picked-up during the relocation, and to ensure that a fixed number of bikes always remains available. This threshold can be set according to the territorial area of each cluster, in order to be sure that the average probability of finding a bike available near the origins of unexpected trips is similar for all clusters, then the ones with a larger area should have a high minimum number of bikes, above the g threshold, greater than the one of the smaller clusters. The UNT_ξ threshold may also be seen as an expedient to compensate for any eventual error that overestimates the prediction.

For each spatio-temporal cluster ξ, we know ζ_ξ^f which is the predicted minimum number of bikes reached in the interval $\Delta\upsilon_f$. If this number is greater than zero, we treat this value by subtracting the UNT_ξ: we call this difference $\varepsilon_\xi^f = (\zeta_\xi^f - UNT_\xi)$. If $\varepsilon_\xi^f \leq g_\xi$, we assume that the cluster is a receiver and belongs to the set ρ^f, i.e., it needs a certain number of bikes at the end of the relocation process. Conversely, if $\varepsilon_\xi^f > g_\xi$ we suppose that ξ is a donor cluster and belongs to the set σ^f: trucks have to visit it to collect bikes to deliver to the receivers.

Consequently, we are able to specify the following problem with the decision variables $r_{ij}^f \in \mathbf{R}^f$:

$$\min \left[\gamma_1 \cdot U^f\ (\mathbf{R}^f) + \gamma_2 \cdot Lu^f(\mathbf{R}^f)\right] \qquad (5)$$

s.t.

$$\sum_{j=1}^{\Gamma} r_{\xi j}^f = 0, \qquad \forall\, \xi \in \rho^f \qquad (6)$$

$$\sum_{i=1}^{\Gamma} r_{i\xi}^f = 0, \qquad \forall\, \xi \in \sigma^f \qquad (7)$$

$$\sum_{j=1}^{\Gamma} r_{\xi j}^f \leq \varepsilon_\xi^f, \qquad \forall\, \xi \in \sigma^f \qquad (8)$$

$$\sum_{i=1}^{\Gamma} r_{i\xi}^f \leq \Psi_\xi - \Phi_\xi^f, \qquad \forall\, \xi \in \rho^f \qquad (9)$$

$$r_{ij}^f \geq 0, \quad \forall\, i,j \in [1, 2, \dots, \Gamma] \qquad (10)$$

The objective (5) aims at minimizing the sum of unfavorable times (U^f) for all the clusters, in which they have a number of bikes below a given threshold g_ξ (the larger is the area of each cluster, the higher the value of g_ξ has to be set), and the number of users lost to the system (Lu^f) (users who decide not to pick-up a free-floating bike because they are not able to find one nearby). These two components of the objective function are weighted by the coefficients γ_1 and γ_2, accordingly to the importance that system managers want to give them. We set $\gamma_2 = 0$ if the number of lost users associated to the operation of the FFBSS is not known. However, this information is known if the FFBSS simulator is used.

The constraints are as follows. As $\mathbf{R}^f$ is an Origin-Destination (O-D) matrix, and considering that each cluster can be either a donor or a receiver (but not both), it is not possible for a receiver cluster to coincide with an origin. Therefore, each element r_{ij}^f of the row associated with a receiver cluster must be equal to zero, which is implied by constraints (6) and (10). Similarly, a receiver cluster cannot overlap with a destination, and therefore each element r_{ij}^f of the column associated to a donor cluster must be equal to zero, which is implied by constraints (7) and (10). Constraints (8) and (9) respectively set an upper bound for each donor cluster and an upper bound related to each receiver cluster. The two bounds are respectively equal to ε_ξ^f, and to the difference between Ψ_ξ (maximum number of bikes that the cluster ξ could accept) and Φ_ξ^f (maximum number of bikes forecasted for the spatio-temporal cluster ξ in the interval $\Delta\upsilon_f$). In fact, it can be stated that constraint (9) is always satisfied, as in a free-floating system it is possible to drop a bike at a large number of points in the network (not only in a rack, but also in any location as long as it is accessible to users). Constraint (10) defines a lower bound for all the decision variables r_{ij}^f (each r_{ij}^f has to be greater than or equal to zero). It is important to highlight that if a cluster is a donor, it must not be a receiver; if every $r_{ij}^f \in \mathbf{R}^f$ is equal to zero, the system does not need relocation; moreover, if the problem is over constrained, no relocation is feasible (i.e., if all clusters are receivers, no bike can be delivered). The solution to problem (5) - (10) returns the final $\mathbf{R}^f$.

The next step (operator's perspective) involves the minimization of the travel time associated with the trucks that relocate the bikes over the network, heading towards the determination of the optimal P^f. The relocation paths for carrier vehicles are obtained by solving a Vehicle Routing Problem (VRP). In particular, we propose that each redistribution truck tr has to first pass through donor clusters starting from its depot or its actual position (Open VRP, where vehicles are not required to return to the depot/starting point). Then, it starts a new route to deliver collected bikes; therefore, the VPR is carried out twice. Each truck capacity c_{tr} is predetermined: consequently, if the sum of the picked-up bikes is greater than its capacity, it needs to carry out more than one q_{tr}^f cycle of pick-up/delivery until it manages to relocate all the bikes. The objective function to be minimized in this phase is:

$$\min\left[d^f/v\ + E^f/v + cb \cdot NR^f\right] \qquad (11)$$

Indeed, being a Free-Floating Bike-Sharing System, the bikes are spread inside each spatio-temporal cluster and carrier vehicles have to pick-up and deliver bikes traveling within this area. To reproduce this behavior as closely as possible, the objective (11) is made of three components. The first one is represented by costs (i.e., travel time) of trucks that are following a certain path during the bike relocation. They vary according to their average speed v, and to the actual distance that trucks are covering (d^f, related to P^f). We refer to the distances calculated between the centroids of clusters.

The second and the third components of (11) are the contributions given by each spatio-temporal cluster ξ. The second component consists of travel time to cover the sum of maximum widths of all the spatio-temporal cluster visited by all trucks. The maximum width of a cluster ξ is the distance between the two elements of ξ located at the most distant points of the cluster. Taking into account that, during the relocation process, bikes are picked up and dropped off by each truck uniformly inside the area of each cluster (no particular suggestion regarding the path to follow within the cluster is given) so we assume that these costs could be a good approximation of the costs related to that cluster. On the other hand, the product between cb and NR^f(third component) is proportional to the number of bikes that need to be relocated; cb is a positive constant corresponding to the average time that it takes to load/unload and to lock/unlock every repositioned bike.

The use of two VRP optimizations –first bikes are collected, then they are delivered- allows for the eventuality that the number of bikes actually picked-up does not coincide with the number of bikes which were expected (forecasted) to be picked up in the spatio-temporal clusters. This could happen if forecasted data are slightly divergent with respect to reality, and in some given spatio-temporal clusters there are fewer bikes than planned. In this situation, at the end of the pick-up process, the trucks have to start their delivery with fewer bikes than expected. Adopting two sequential VRP optimizations, it becomes possible to mitigate the negative outcomes descending from this possibility, as explained in the following paragraph.

First of all, the job of each truck driver is easier: he/she knows from the beginning of each trip what to do – to pick up or deliver bikes. Secondly, knowing how many bikes have been collected at the end of the pick-up process (and supposing there are less than expected due to some forecasting inconsistencies and to the dynamicity of the system), it is possible to plan how to redistribute/deliver them according to the real-time need in each spatio-temporal cluster, in order to alleviate the imbalance that has occurred. In particular, we assume there are a certain number of spatio-temporal clusters to which the collected bikes must be delivered. We propose considering the cluster that requires the highest number of bikes, and to subtract one bike from its forecasted share –i.e., from the number of bikes it needs. This is sufficient if the number of collected bikes is just one unit less than the number of predicted ones. Otherwise, this process has to be repeated iteratively until the forecasted number coincides with the actual one of bikes at the end of the pick-up process. At the end, some clusters may not be able to receive any more bikes. Thus, in order to speed up the delivery process, these clusters are eliminated from the calculation of the delivery route within the second VRP optimization. Let us suppose, alternatively, only one VRP optimization is carried out, collecting and delivering bikes at the same time, following the suggested path. As explained above, if the truck at some point on its trip has collected fewer bikes than expected, it might not be able to deliver the bikes to the following cluster that needs them. Consequently, the truck has to deliver to the cluster the number of bikes at its disposal, without any particular evaluations regarding its expected needs, compared to the needs of other clusters. Summing up, the two VRP optimizations allow the most important requests to be preserved, distributing bikes in a proper way.

Note that if there are more bikes than expected for the pick-up, the situation is better than if there are fewer. The problem could arise if the number of available spots in each spatio-temporal cluster is limited (then the need to have as many spots as possible could emerge). However, being a free-floating system, we are supposing that the number of available points at which it is possible to drop a bike for each cluster is always sufficient to satisfy each delivery necessity. This is true as in a free-floating system it is possible to leave the bikes not only in 'simple' racks (not connected to an electronic system as in a BSS with stations) but also at any location in the network, as long as it is accessible to every user.

An alternative and more complex approach to optimize the delivery of the collected bikes could involve an optimization model able to establish the number of bikes to allocate to each cluster, thus better responding to the dynamicity of the system (i.e. the Relocation Matrix changes within each DSS activation interval, updating its values according to real-time data and allowing the bike delivery to be enhanced); this could be the subject of future more extensive research.

The entire proposed DSS is clarified in the flowchart in Fig. 4.

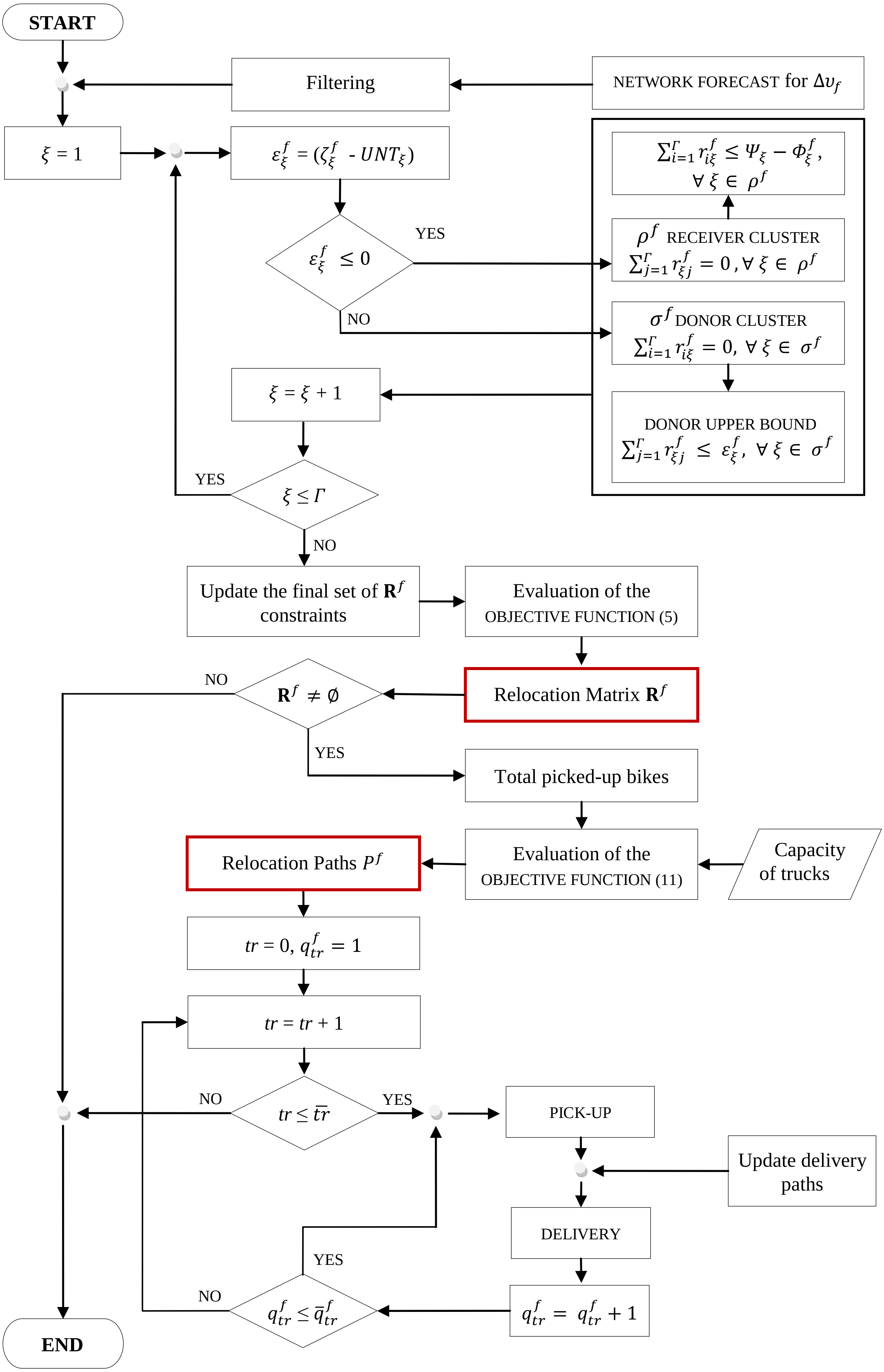

Fig. 4 Flowchart of the proposed DSS.

5 Bike sharing demand and system micro simulator

In order to apply the proposed DSS, we needed to simulate bike demand and operations of an FFBSS. In this section, we explain how to randomly generate bike demand in a network, guaranteeing an equilibrium of the number of bikes in the system at the end of the day.

We initially assume that we have to simulate a set of 'uniform' days, from the point of view of the type of the day dw (working days or weekend/holiday days).

Given a network with an FFBSS operating on it, we aggregate it in different zones as explained in the introduction of Section 3 (zoning). Each zone has its own centroid φ, and globally we can count a number of $\bar{\varphi}$ centroids on the entire network. With respect to system dynamics, we assume the operating day is divided into $\bar{\tau}$ discrete time intervals. For every time interval τ and for each centroid φ, given the pick-up bike demand, summarized in the vector $\mathbf{Db}_{\xi}$, the model simulates the destination choice in order to assess the arrival time for each user.

The starting configuration of the system (i.e., at the first-time interval τ of operation) is represented by the number of bikes *N* scattered in each zone, associated to the relevant centroid φ.

With the aim of building a system as realistic as possible, we assume there are $\bar{p}$ different types of zones, with $p \in [1, 2, ..., \bar{p}]$. Each type of zone is related to its location within the city (central or peripheral) and to the request for bikes/pick-up demand associated to it. There is a certain number of centroids φ_p belonging to every type *p*; consequently, given *p*, we have [1, 2, ..., φ_p] centroids.

Furthermore, according to the type of the day, the time of the day (i.e., τ) and the type of centroid, we can also associate a different demand level $\delta^{\tau}_{p,dw}$, namely a range of values between which to randomly extract the pick-up demand **Db** with $\delta^{\tau}_{p,dw} \in [0, 1, 2, \ldots, \bar{\delta}^{\tau}_{p,dw}]$. Knowing the pick-up demand $\mathbf{Db}_{\xi}$ in a centroid at a given time of the day (τ), we can start to simulate how the system is going to operate. First of all, we have to check if the number of bikes available in the zone is sufficient to satisfy the bike requests. If it is greater than the pick-up demand (i.e., users' bike request), then we can update the number of bikes associated to that centroid and proceed; otherwise we need to understand what happens to the 'unsatisfied' users.

Basically, we leave them two possible options. They can check (using the smartphone app of the FFBSS) if a bike is available in another zone nearby (computing distances between centroids of the two zones, and given a maximum threshold corresponding to the time the user is willing to walk): they can move toward the other zone to pick-up the bike, otherwise they are 'lost users', as they are not going to use the system. According to the user decision (i.e. if it is reasonable to change his/her trip-origin zone or not according to the threshold we have set), we then update the pick-up demand of the nearby zone at the time τ + 'walking time between the two centroids' (by adding these new users to the random demand extracted by the simulator), or we update the number of lost users. Then, we need to know *where* the user is going on 'his' free-floating bike (i.e. trip destination).
We consider two possible groups of user behavior:

1) he/she can start his/her trip from a centroid O (origin) and after a while (statistical average round-trip time) return to the starting point; or,
2) he/she can start a one-way trip from O to a destination D.

Associating a percentage to these two possibilities (depending on the statistical data collected inherent to the chosen case study), we try to estimate the probability that each of these two behaviors has of occurring in the system. If the user is going to start a one-way trip (case 2), the choice model is based on the relative O-D attractiveness and the nature of the trip. Otherwise (case 1), the attractiveness is not involved and we calculate only the average time to return to the origin point. We do not consider the possibility of the user being unsatisfied when he reaches his destination (Caggiani and Ottomanelli, 2013) since a free-floating system allows users to leave the bike (almost) anywhere near the final point of their trip (we are assuming that each zone has a very large capacity for parking bikes, as in reality).

At the beginning of each time interval τ, the number of available bikes and the pick-up demand are updated to take into account what happened in the previous steps and the eventual relocated bikes. The architecture of the FFBSS operations microsimulator, with its relevant functions, is graphically explained by the flowchart in Fig. 5.

The Decision Support System is activated at a constant gap time. At the end of the day (when the status of the system is steady), a static redistribution is carried out in order to restore the starting bike configuration of the zones.

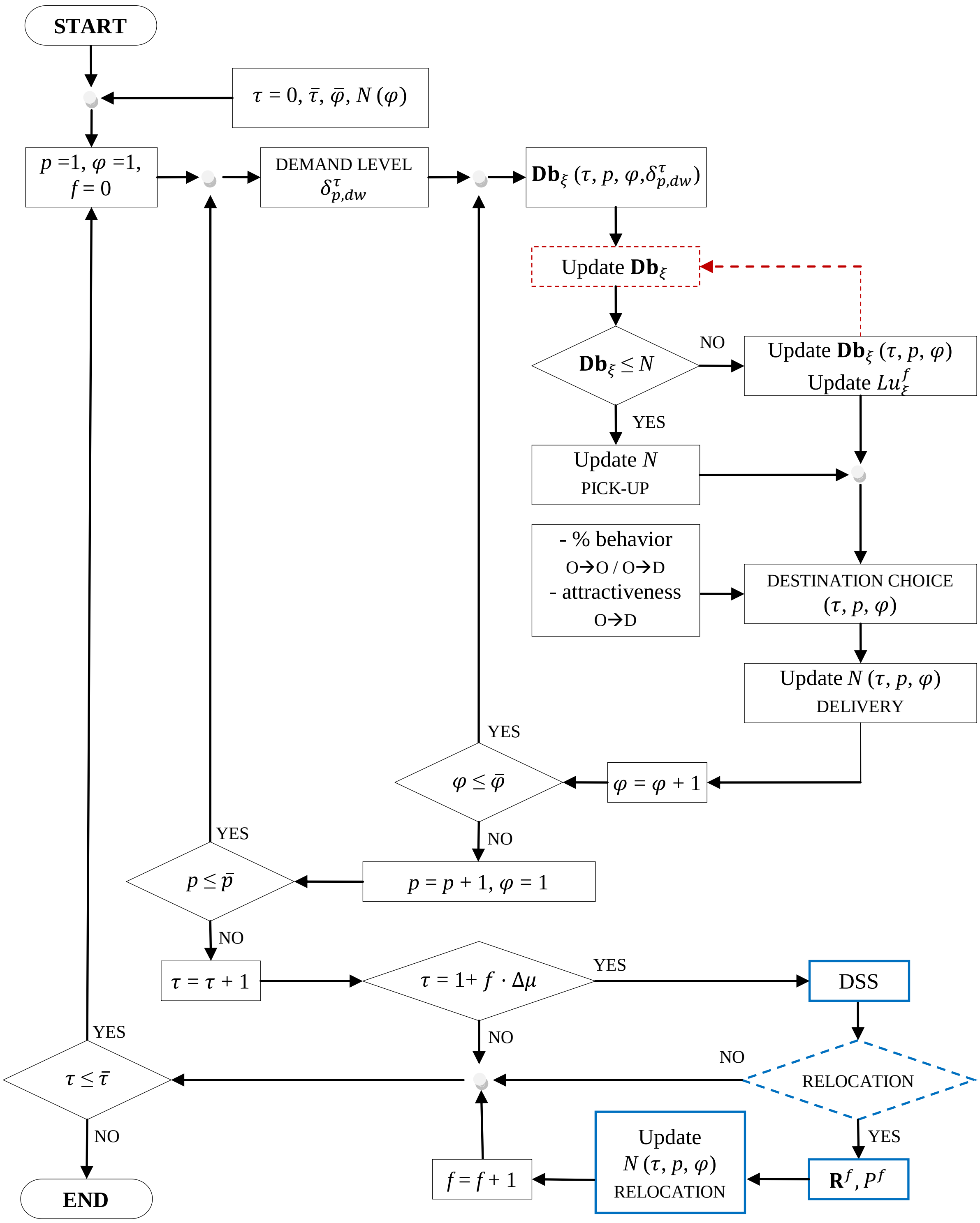

Fig. 5 FFBSS Simulation flowchart.

6 Numerical study setup

The proposed methodology is here applied to a real sized case study, in order to verify its effectiveness and robustness. In the following subsections, the test area and the set parameters are described; in Section 7, two sensitivity analyses (with different pick-up bike demand levels, and changing the initial number of clusters) are provided, and some results presented.

6.1 Description of the study area

The methodology and simulations were applied to an area of 9 km^2, divided into 36 zones of 0.5 km x 0.5 km each (we considered census districts of 100 m x 100 m in the central areas and of 250 m x 250 m for the peripheral ones. We assumed that each of these 36 zones could accept a maximum number of dropped-off bikes equal to 100 (i.e., their receiving capacity is very high). Once the total number of available bikes in the network has been fixed (set equal to 200), it is necessary to set up the starting configuration of the Free-Floating Bike Sharing System, i.e. the number of available bikes spread in each zone. The following Fig. 6 shows the configuration of the system at the beginning of each operation day: the circles are the centroids φ, while every cross corresponds to a bike. This configuration is pretty similar to the one of a real bike-sharing system. In fact, in the morning, usually there is a higher bicycles density in the peripheral zones of the city, while fewer bicycles are located in the city center.

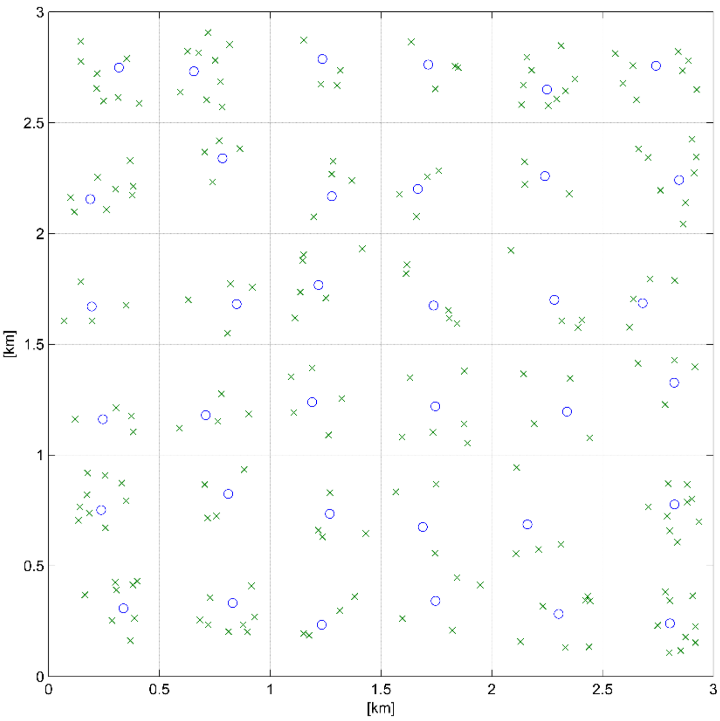

Fig. 6 Case study: aggregate zones with centroids φ and bikes.

We assumed there are $\bar{p}$ = 4 different types of zones: central, peripheral with a low demand, peripheral with an average demand, peripheral with a high demand. Finally, we set a time interval τ equal to 5 minutes, so the total number of discrete time intervals for each day ($\bar{\tau}$) is equal to 288. These parameters (τ, $\bar{\tau}$, $\bar{\varphi}$, N (φ)) are the necessary inputs to start and simulate demand and operation of the FFBSS.

The relocation is carried out by one truck with a capacity of 20 bikes. Having only one truck, we applied the Travelling Salesman Problem (with fixed start point, open endpoint) to solve the optimization problem (11), as it is a special case of the VRP. We refer to the distances between centroids.

Aiming at a sensitivity analysis to understand the importance of the pick-up demand level, we performed our simulations considering three values of the daily pick-up demand:

- **Db**1: approximately 1200 picked-up bikes/day;
- **Db**2: approximately 1600 picked-up bikes/day;
- **Db**3: approximately 2100 picked-up bikes/day.

We underline that the **Db**1 level also has enough picked-up bikes (given the size of the case study) to require relocation to enhance its operation.

We carried out the simulations for 12 days, assuming that each day was Wednesday (uniform data selection, as explained in the previous sections). However, similar conclusions would have been reached also selecting another set of days of the week (not necessarily Wednesdays). The pickup demand during the night (i.e. between midnight and 5:00 a.m.) has been set equal to zero; moreover, we have supposed that the last bicycles are delivered until 1:00 a.m. The starting configuration at the beginning of each day was always the same, as we assumed a static bike redistribution performed between 1:00 a.m. and 5:00 a.m..

6.2 Data clustering and forecasting

Knowing the 36 zones temporal patterns, we can cluster them according to their temporal trends. Thanks to a discrete wavelet transformation to analyze the signals, we aggregated the filtered data into n =3 temporal clusters $\mathbf{T}_k$. Then, we solved the upper-level objective function (1), to find the spatio-temporal clusters associated to each $\mathbf{T}_k$, by means of genetic algorithms (population size: 200; constraint tolerance: 10^{-3}; function tolerance: 10^{-6}). On the other hand, the lower level objective function (2), which is NP-hard, was solved by means of a heuristic algorithm, namely Lloyd’s algorithm (Lloyd, 1982).

In the k-means optimization (2) various kinds of distance formulation could be adopted: the most suitable one should be selected according to the road/cycling urban set-up. As a matter of fact, if distances greater than the real ones are used, there may be an increase in the number of spatio-temporal clusters that are the outputs of the optimization. The opposite effect would be obtained with distances smaller than the real ones. For those cities having a grid configuration of their road/cycling network, it could be suggested the use of the Manhattan distance (Krause, 1973), as it can be more similar to the real distances between two different locations for this typology of network set-up. However, to free our method from the layout of the traffic network and make it more general, we chose to use Euclidean distances in the k-means optimization. The k-means using Euclidean distance metric may give better results if compared to Manhattan k-means especially looking at the distortion (Singh, Yadav and Rana, 2013).

We globally found Γ = 7 spatio-temporal clusters ξ, the same number but with slightly different perimeters for each level of picked-up bikes. Fig. 7*(a)* and 7*(b)* show the obtained clusters for the three levels of daily pick-up demand (clusters for **Db**1 and **Db**2 are exactly the same): dots of the same color belong to the same temporal cluster T_k, while polygons depict the spatio-temporal aggregation, with an asterisk to indicate their centroid.

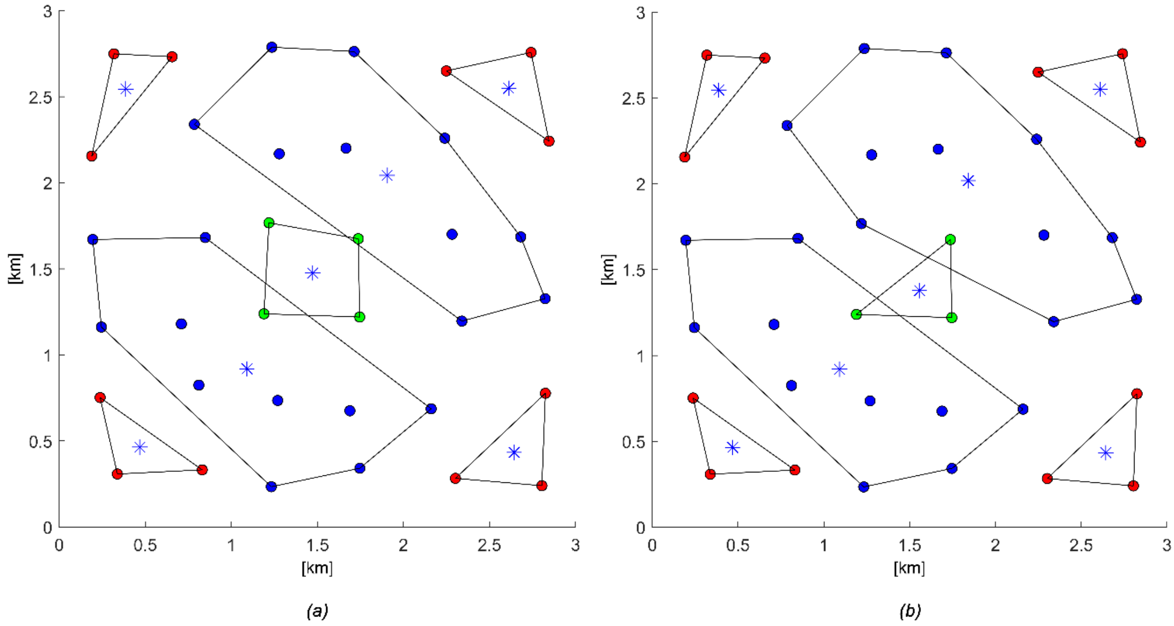

Fig. 7 *(a)* Spatio-temporal clusters for **Db**1 and **Db**2 (n = 3); *(b)* Spatio-temporal clusters for **Db**3 (n = 3).

We were able to generate the distance matrix of this new spatio-temporal clustered configuration, calculating the travel time to go from each cluster centroid to another (by walk, by bike, and by carrier vehicles): which was useful on the next steps in the procedure.

Then, we had all the necessary information to apply the NARX to our system. The structure of the adopted NARX network is a two-layer feedforward network, with a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. In particular, the hyperparameters that we have set are:

- the number of delays (number of past values to consider, D) equal to 6;
- the number of hidden neurons equal to 10.

These hyperparameters have been tuned empirically, evaluating the prediction errors obtained starting from different random configurations. The parameters of the weight matrices and bias values have been trained according to Levenberg-Marquardt optimization (Hagan and Menhaj, 1994).

We set 70% of the samples to be used for training; 15% to be used to validate what the network is generalizing and to stop training before overfitting; the last 15% to be used as a completely independent test of network generalization. By progressively increasing the value of Δz (i.e., the width of each time interval used to make the prediction), we were able, step by step, to forecast the trend of available bikes for a hypothetical following Wednesday (13th day).

The DSS time span $\Delta\mu$ was set to 48 time-steps τ (4 hours). After every $\Delta\mu$, the DSS is activated and we forecast the next 60 steps ($\Delta\lambda$). These forecasts are performed not only on the basis of the trend of the past 12 Wednesdays, but also considering the new values that occurred during the 13th Wednesday, until the moment of the prediction, to take into account the evolution of the system. Then, we applied the filtering techniques to these predictions, in order to avoid too high peaks or drops that seem to deviate too much from the real past trend of available bikes.

As explained in Section 3.2, to evaluate the performance of the proposed relocation strategy for different filters applied to the historical series, we can decide to train, validate and test the dynamic neural network relying on four NARX starting datasets. These datasets are, as explained previously, the exact past values (E_NN) corresponding to the discretization Δz, or alternatively the minimum (Min_NN), the average (Av_NN) or the maximum (Max_NN) value of bikes in Δt (interval of time having as midpoint the exact past value). Setting Δt = 13 time-intervals τ, for each level of picked-up bikes (**Db**1, **Db**2 and **Db**3), we trained the NARX according to these four possibilities, predicting the trend of the 13th Wednesday for each one of the seven ξ. In order to obtain a realistic prediction, we trained the network for every cluster (for each neural network starting dataset and for each level of pick-up demand) for 10 times: among them, we selected the best neural network (according to the lowest MSE), and we used it to perform the prediction of the 13th Wednesday for 30 times. For all the forecasting repetitions, we assumed the pick-up demand matrix (that shows how the demand is scattered among the zones) to be always the same. On the other hand, the destination choices will change, due to the random extraction of some parameters (i.e., the attractiveness of the zones).

7 Numerical study: sensitivity analyses and results

7.1 Relocation performance indicators

We simulated the behavior of the network with (Rel) and without (No_Rel) the dynamic relocation of the bikes to understand how it can help the better operation of the free-floating system.

The main outputs of the sensitivity analyses are the unfavorable times $U_t = \sum_{f=1}^{\bar{f}} U^f$, in which the spatio-temporal clusters have a number of bikes below the thresholds g_ξ; the users lost to the system $Lu = \sum_{f=1}^{\bar{f}} Lu^f$, and the satisfied users. We can find U_t and Lu by solving the aforementioned problem (5) - (10) using genetic algorithms, according to the number of bikes forecasted for each spatio-temporal cluster. In this application, we consider lost users those who are not able to find an available bike within the radius of one time step τ (travelling on foot) from their origin centroid. We count as satisfied those users able to find a bike inside their origin zone (0.5 x 0.5 km) – not including in this value the users that ride a bike found in a nearby zone.

The results are displayed as percentages (Eqs. 12, 13, 14), comparing them with the values from simulating the same system without a dynamic bike relocation process operating on it.

$$\%U_t = \frac{U_t(\text{Rel}) - U_t(\text{No\_Rel})}{U_t(\text{No\_Rel})} \cdot 100 \qquad (12)$$

$$\%Lu = \frac{Lu(\text{Rel}) - Lu(\text{No\_Rel})}{Lu(\text{No\_Rel})} \cdot 100 \qquad (13)$$

$$\%Satisfied = \frac{Satisfied(\text{Rel}) - Satisfied(\text{No\_Rel})}{Satisfied(\text{No\_Rel})} \cdot 100 \qquad (14)$$

Consequently, if the unfavorable time values and the ones related to lost users are negative numbers, it means that the relocation process has improved the free-floating BSS. In fact, we reduce the percentage of times users are not able to find an available bike to pick-up, and at the same time, we reduce the lost users, thanks to a dynamic bike relocation. On the other hand, the number of satisfied users needs to be positive to show an enhancement of the service: it means that we increase the percentage of people able to use a free-floating bike.

7.2 First analysis: different pick-up bike demand levels

The first sensitivity analysis was performed for the three levels of picked-up bikes (**Db**1, **Db**2 and **Db**3). We set the same starting configuration of the system and the same pick-up matrix, for each level of picked-up bikes, trying to have a similar trend during the days with and without relocation, aiming to compare them. As a matter of fact, there were some differences in the behavior of the system, that reacted in a different way, as during the relocation process, bikes are moved from one zone to the other, changing the configuration.

We carried out the simulation and the comparison between the system with and without relocation for four times, starting from different configurations of the pick-up matrix. In this way, although the three daily pick-up demands were assumed to be numerically the same, we wanted to understand through a sensitivity analysis whether a different distribution among the zones of the pick-up demand can affect the final results. Basically, we globally simulated four Wednesdays, hypothetically all corresponding to the 13th day of operation of the system, each of them having a different starting pick-up matrix.

Figure 8 summarizes our findings for the pick-up bike demand level **Db**1 by means of box plots with notches. In each box, the central mark indicates the median value, the edges are the 25th and 75th percentiles, the whiskers extend to the most extreme non-outlier values, and outliers are plotted individually (with red crosses). To assist in judging differences between sample medians, we used notches to show the 85% confidence interval (CI) for the median (see Table 1). Each box plot displays the results of the 30 simulations carried out for each Wednesday, for each NARX starting dataset, for the pick-up bike demand level **Db**1.

Observing Fig. 8, we can see that on the y-axis of each graph there is the percentage of reduction/improvement of the considered main outputs, namely U_t, Lu and *Satisfied*. Four groups can be observed on the x-axis, each one consisting of four box plots. Every group represents a different NARX starting dataset adopted to realize the forecasting (neural network trained relying on the exact past values E_NN corresponding to the discretization Δz, or on the basis of the minimum Min_NN, the average Av_NN or the maximum Max_NN value of bikes in Δt). The four elements that constitute each group match with the four hypothetical Wednesdays on which we carried out the simulation.

In order to have a more immediate overview of the achieved results for **Db**1, **Db**2 and **Db**3 demand levels, Table 1 summarizes the average of the medians and lower/upper limits of the 85% confidence interval for every group of four Wednesdays. It can be remarked that the zero, for all the four Wednesdays, is always outside the 85% confidence level; there are few exceptions only for the pick-up bike demand **Db**3 related to the percentage of lost users %Lu. This happens since the system, in the **Db**3 case, is oversaturated (i.e. demand is much higher than supply). Under this condition, the model finds more difficult to reduce the lost users of the system when the pick-up bike demand level is far above the capacity of the system itself. In fact, the relocation alone is not sufficient to respond to the pick-up demand, and in such a situation it would be necessary to enhance the supply systems (for example, by adding new bicycles into the system).

Looking at the unfavorable times U_t, we can assert that globally the system improves by means of the relocation process: for almost all of the 30 simulations carried out for every Wednesday, the unfavorable times U_t decreased as we expected. The number of satisfied users tended to rise thanks to the bike relocation. Considering the number of lost users, on the other hand, it can occur for some (limited) simulations that it grows in the configurations with relocation, compared with the ones without. Consequently, we decided to look at each single simulation to understand what happens when one of the three considered main outputs U_t, Lu or *Satisfied* presents worse values than the same configuration without relocation.

We found that, in almost every case, at least one of the other two outputs denotes an enhancement of the FFBSS system. Indeed, we must highlight that our user's objective function (Eq. 5) is multi-objective, as it aims to minimize both the unfavorable times and lost users, and each objective is weighted by a coefficient. In our application, we supposed that these two coefficients are the same ($\gamma_1 = \gamma_2 = 1$): this means that we are considering equivalent (for the bike-sharing system) to have one lost user or one time-step with a number of bicycles below a given threshold. This assumption depends on the fact that we have established a reasonable maximum intensity of pick-up equal to 5 bicycles for each time step (i.e. one pick-up per minute, $\bar{\delta}_{p,dw}^{\tau} = 5$). Then, on average, it is fair to consider one lost user per time step with a number of bicycles below a given threshold (for example, if the unfavorable times are equal to 6 time steps of 5 minutes, then it is equivalent to the loss of 6 users in 30 minutes).

However, it could be possible to calibrate the two coefficients γ_1, γ_2 in order to achieve further improved results. In any case, almost all the simulated configurations showed a clearly positive trend connected to the adoption of a dynamic relocation and in particular the percentage of satisfied users improves almost always (see the %*Satisfied* in Table 1 and Fig.8).

Looking at Table 1, we can state that as the pick-up demand level increases, the achieved results seem to cover a more restricted range of attainable percentages, globally narrower if compared with the previous level. Therefore, we can conclude that when there is a higher request for bikes in the system, the possible range of improvement that we can achieve is narrower and slightly less satisfying. However, it is important to underline that even a slight improvement in the system may have a great significance in the framework of the FFBSS as we are dealing with a dynamic demand for bikes.

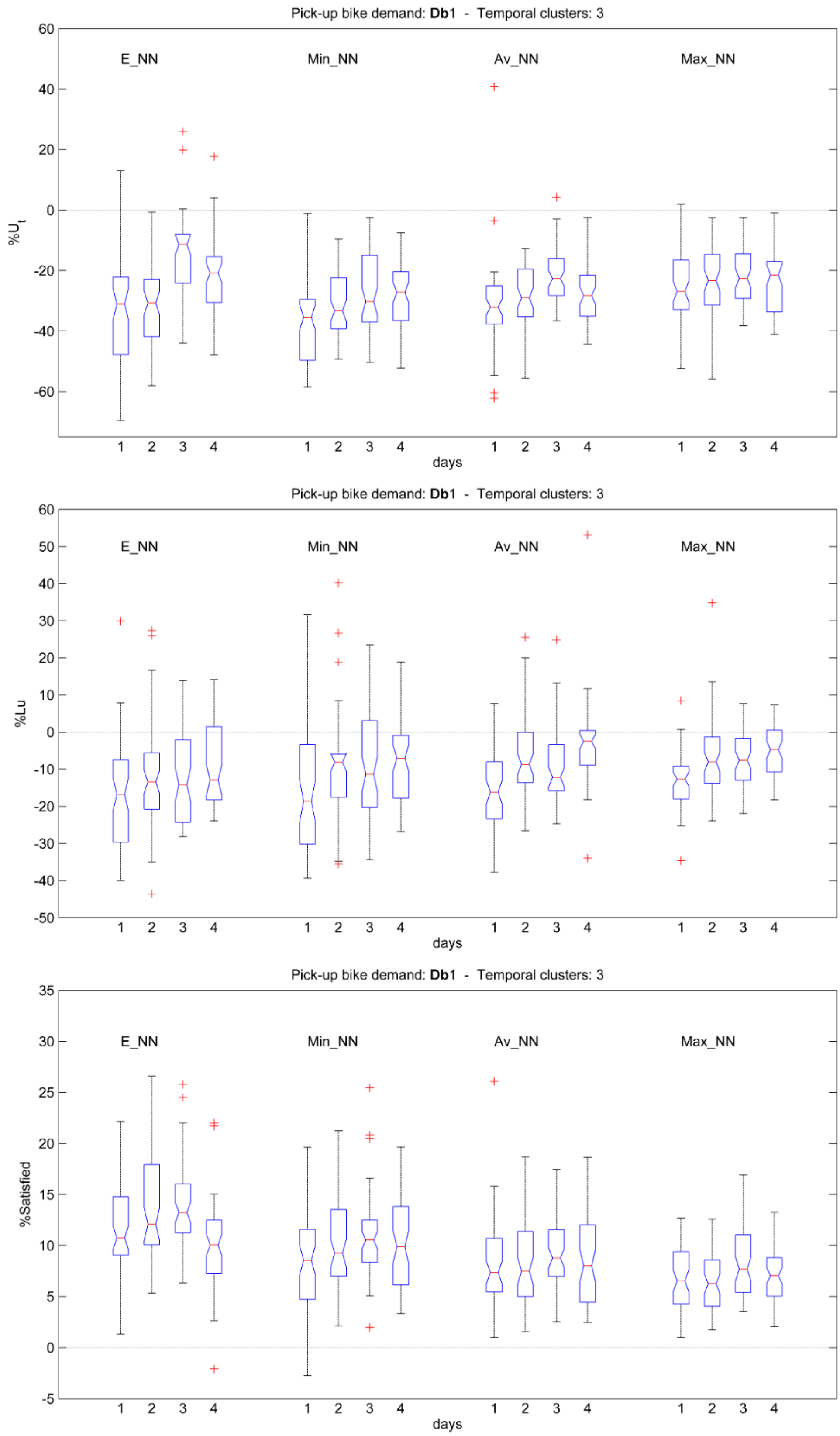

Fig. 8 Overview of the performed sensitivity analysis of the behavior or an FFBSS with and without bike relocation, represented by means of box plots. Comparison of unfavorable times (U_t), lost users (Lu)

and satisfied users (*Satisfied*) for the pick-up bike demand level **Db**1, for four neural network starting datasets (E_NN, Min_NN, Av_NN, Max_NN), in four days (1, 2, 3, 4) of operation of the system.

Table 1 Average of the medians and lower/upper limits of the 85% confidence interval for every group of four Wednesdays: comparison of unfavorable times (U_t), lost users (Lu) and satisfied users (*Satisfied*) for three bike demand pick-up levels (**Db**1, **Db**2 and **Db**3) and four neural network starting datasets (E_NN, Min_NN, Av_NN, Max_NN).

	% U_t											
	Db1; *n*=3				**Db**2; *n*=3				**Db**3; *n*=3			
	E_NN	Min_NN	Av_NN	Max_NN	E_NN	Min_NN	Av_NN	Max_NN	E_NN	Min_NN	Av_NN	Max_NN
CI upper limit	-19.5	-27.6	-25.2	-20.2	-18.8	-16.0	-21.9	-19.5	-14.6	-14.4	-17.8	-15.0
median	-23.5	-31.5	-28.0	-23.5	-22.5	-20.0	-24.7	-22.0	-16.8	-16.5	-19.8	-16.9
CI lower limit	-27.4	-35.5	-30.8	-26.9	-26.1	-23.9	-27.4	-24.6	-19.0	-18.5	-21.8	-18.8
	% Lu											
	Db1; *n*=3				**Db**2; *n*=3				**Db**3; *n*=3			
	E_NN	Min_NN	Av_NN	Max_NN	E_NN	Min_NN	Av_NN	Max_NN	E_NN	Min_NN	Av_NN	Max_NN
CI upper limit	-10.2	-7.2	-7.2	-6.0	-4.2	-5.4	-4.1	-3.8	-0.1	2.9	-2.1	0.6
median	-14.3	-11.3	-9.9	-8.2	-8.4	-8.9	-7.1	-6.4	-2.4	0.3	-4.1	-1.4
CI lower limit	-18.4	-15.4	-12.5	-10.5	-12.5	-12.4	-10.2	-9.0	-4.6	-2.2	-6.1	-3.3
	% *Satisfied*											
	Db1; *n*=3				**Db**2; *n*=3				**Db**3; *n*=3			
	E_NN	Min_NN	Av_NN	Max_NN	E_NN	Min_NN	Av_NN	Max_NN	E_NN	Min_NN	Av_NN	Max_NN
CI upper limit	12.8	10.9	9.2	7.9	12.7	10.8	9.3	9.0	14.6	11.3	13.0	11.0
median	11.5	9.6	7.9	6.9	11.4	9.5	8.5	8.1	13.1	9.8	11.9	9.9
CI lower limit	10.3	8.2	6.7	5.9	10.0	8.2	7.7	7.2	11.6	8.3	10.8	8.8

7.3 Second analysis: different spatio-temporal clustering

In this subsection, an additional sensitivity analysis is provided, to be aware of the influence that the number of temporal clusters *n* (and accordingly, the number of spatio-temporal clusters Γ) has on the final outputs of the FFBSS.

The Fig. 9*(a)* and 9*(b)* show the clusterization achieved for an average level of pick-up demand **Db**2, setting n = 5 and n = 7 as the number of temporal clusters associated to Γ = 9 and Γ = 11, respectively. Thanks to the property of the hierarchical clustering, we adopted the same cluster tree for this sensitivity analysis, varying the number of temporal clusters.

Again, Table 2 summarizes the achieved results. The main observation on the results presented in Table 2, in particular observing the values related to the unfavorable times U_t, is that the best configuration seems to be the one with the lowest aggregation level (fewer temporal and spatio-temporal clusters). Probably, this happens because for n = 3, globally, the total distance to cover is lower than that in both the n = 5 and n = 7 configurations. As a matter of fact, when there is less traveling, the bike delivery is faster, and consequently (also considering that in the n = 3 configuration the number of bikes to deliver is lower than in the remaining two) there are fewer the unfavorable times U_t. Generally, it could be said that, for each case study, there is an optimal number of temporal and spatio-temporal clusters, corresponding to the best combination of %U_t, %Lu, and %*Satisfied*.

This remark is also evident in Fig.10, which shows the box plots associated to U_t, by changing the number of the clusters. If some simulations for n = 3 are able to reach an improvement of more than 40% compared with the same configuration without relocation, for n = 7 it is evident that the best solution does not exceed a range of values around 30%.

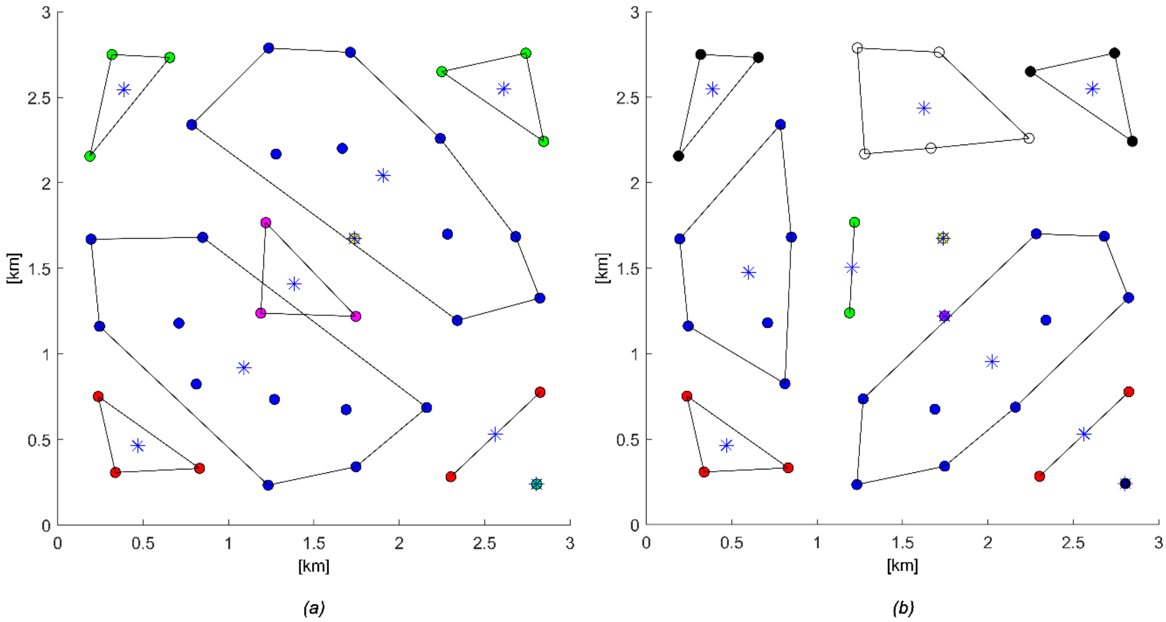

Fig. 9 *(a)* Spatio-temporal clusters for **Db**2 (n = 5); *(b)* Spatio-temporal clusters for **Db**2 (n = 7).

Table 2 Overview of the results of the sensitivity analysis performed on the second level of picked-up bikes (**Db**2) and for each neural network starting dataset (E_NN, Min_NN, Av_NN, Max_NN), comparing a different number of temporal clusters (n = 3, n = 5, n = 7).

	% U_t											
	Db2; *n*=3				**Db**2; *n*=5				**Db**2; *n*=7			
	E_NN	Min_NN	Av_NN	Max_NN	E_NN	Min_NN	Av_NN	Max_NN	E_NN	Min_NN	Av_NN	Max_NN
CI upper limit	-18.8	-16.0	-21.9	-19.5	-15.6	-16.3	-17.4	-17.1	-9.5	-6.5	-10.6	-10.5
median	-22.5	-20.0	-24.7	-22.0	-19.2	-19.9	-20.6	-19.1	-12.7	-9.0	-13.5	-13.0
CI lower limit	-26.1	-23.9	-27.4	-24.6	-22.8	-23.5	-23.8	-21.1	-15.8	-11.5	-16.3	-15.5
	% Lu											
	Db2; *n*=3				**Db**2; *n*=5				**Db**2; *n*=7			
	E_NN	Min_NN	Av_NN	Max_NN	E_NN	Min_NN	Av_NN	Max_NN	E_NN	Min_NN	Av_NN	Max_NN
CI upper limit	-4.2	-5.4	-4.1	-3.8	-7.8	-3.2	-1.4	-2.7	-5.5	-1.4	-3.1	0.9
median	-8.4	-8.9	-7.1	-6.4	-11.2	-6.3	-4.4	-5.9	-8.9	-5.6	-6.8	-2.2
CI lower limit	-12.5	-12.4	-10.2	-9.0	-14.6	-9.3	-7.4	-9.2	-12.3	-9.7	-10.5	-5.4
	% Satisfied											
	Db2; *n*=3				**Db**2; *n*=5				**Db**2; *n*=7			
	E_NN	Min_NN	Av_NN	Max_NN	E_NN	Min_NN	Av_NN	Max_NN	E_NN	Min_NN	Av_NN	Max_NN
CI upper limit	12.7	10.8	9.3	9.0	10.4	10.1	8.1	8.8	12.1	10.9	10.8	9.1
median	11.4	9.5	8.5	8.1	9.2	8.4	6.9	7.7	10.5	9.5	9.4	8.0
CI lower limit	10.0	8.2	7.7	7.2	8.0	6.7	5.6	6.7	8.9	8.0	8.0	6.9

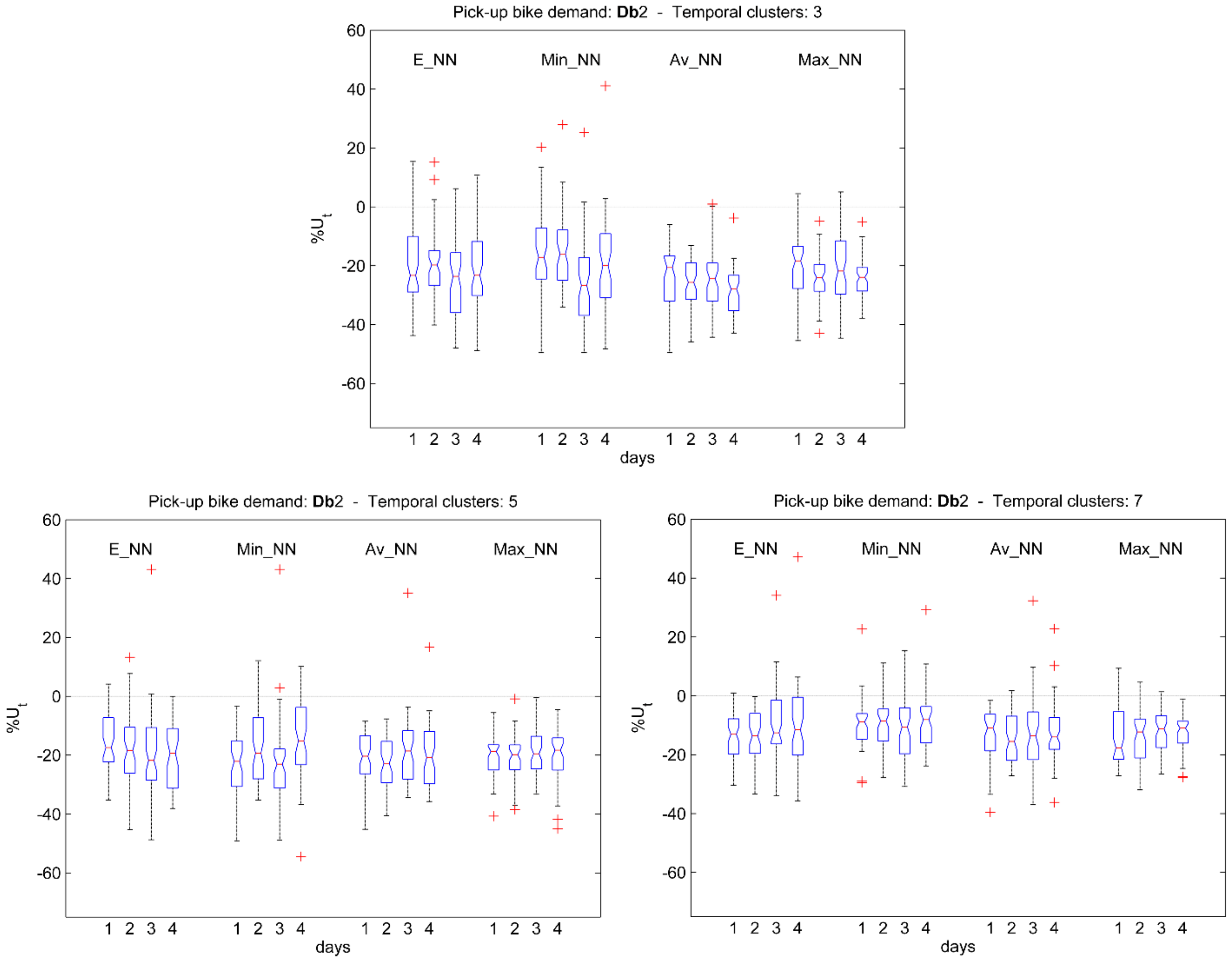

Fig. 10 Box plots of the unfavorable times (U_t) obtained for the second level of pick-up bikes (**Db**2) and for each neural network starting dataset (E_NN, Min_NN, Av_NN, Max_NN), comparing a different number of temporal clusters (n = 3, n = 5, n = 7).

7.4 Operator's costs remarks

At the end of the performed sensitivity analyses, it seems that the adoption of a dynamic process of bike relocation could lead to the better operation of the system. Therefore, we decided to present a further remark related to the kilometers covered by carrier vehicles and the number of relocated bikes, by averaging once again the results obtained from the 30 simulations of the system for each group of four Wednesdays (Table 3).

Table 3 Overview of the results of the number of relocated bikes and the kilometers covered by carrier vehicles for each level of picked-up bikes (**Db**1, **Db**2, **Db**3) and for each neural network starting dataset (E_NN, Min_NN, Av_NN, Max_NN), comparing a different number of temporal clusters (n = 3, n = 5, n = 7).

	relocated bikes					relocator paths [km]				
	Db1; *n*=3	**Db**2; *n*=3	**Db**2; *n*=5	**Db**2; *n*=7	**Db**3; *n*=3	**Db**1; *n*=3	**Db**2; *n*=3	**Db**2; *n*=5	**Db**2; *n*=7	**Db**3; *n*=3
E_NN	75.9	76.5	72.8	79.9	82.1	188.8	173.0	194.1	204.1	190.9
Min_NN	67.6	77.1	80.1	84.6	89.0	187.4	207.6	212.0	206.1	212.6
Av_NN	51.5	52.9	54.1	66.4	62.4	131.4	131.3	164.3	173.6	141.5
Max_NN	46.0	48.5	51.1	57.9	55.1	118.5	125.4	149.3	155.3	132.9

The neural network starting datasets Av_NN and Max_NN are the ones able to relocate a lower number of bikes, covering accordingly fewer kilometers on the network, and achieving final outputs values comparable with the ones obtained by the other two starting datasets. Consequently, we can assert that the operators could decide to simulate demand and operation of the FFBSS relying on one of these two methods, as they are able to reach a satisfactory improvement in unfavorable times, lost users, and satisfied users, minimizing at the same time their relocation costs.

As an example, Fig. 11 shows the behavior of the system by means of box plots, only for one level of pick-up bikes **Db**2 and for n = 3 temporal clusters, depicting the trend common also for the remaining combination of parameters of the system.

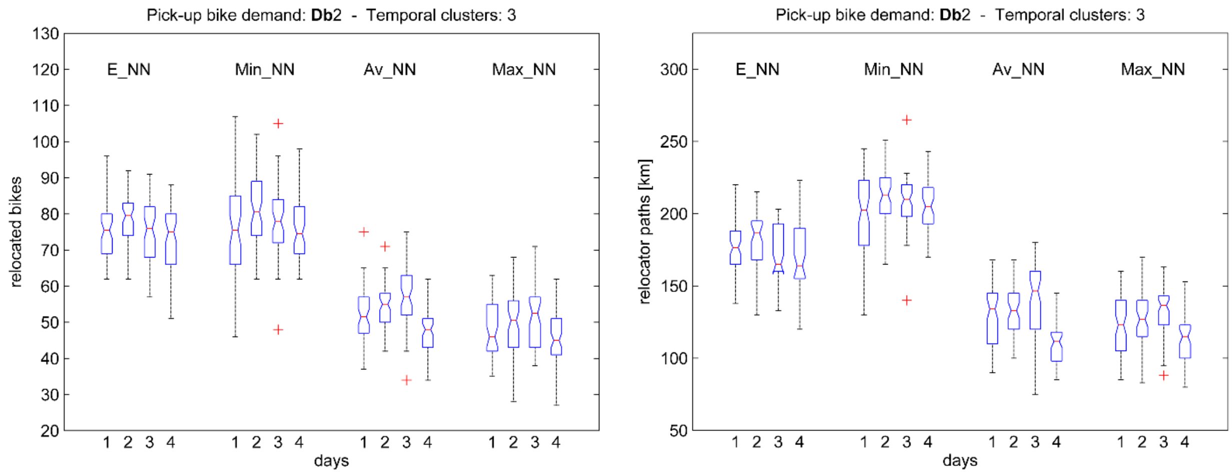

Fig. 11 Boxplots of the number of relocated bikes and the kilometers covered by carrier vehicles obtained for the second level of picked-up bikes (**Db**2) and for each neural network starting dataset (E_NN, Min_NN, Av_NN, Max_NN), with n = 3.

In general, relocation costs, besides covering the costs dependent on traveling distances (e.g. fuel costs) include also wage costs for the relocation staff, facility costs such as capital/operation costs of median or heavy-duty trucks, etc. They are a critical component of total costs. Relocation costs for FFBSSs could be larger than those related to station-based systems, assuming that, relocating the same number of bicycles, the carrier vehicles might be forced to travel along longer paths, being the bicycles spread on the territory.

Commonly, in BSSs the revenue associated with memberships and user fee is not enough to cover the total costs. Most BBSs are subsidized and the gap between revenues and costs is filled by general revenues, advertising revenues, parking revenues, government grants and/or sponsorships. Noteworthy the case of Paris (Vélib' system), where the revenue related to memberships and user fees for the first operating year went entirely to the city of Paris as revenue, since all costs have been covered by a billboard contract (Midgley, 2011).

It is, therefore, necessary to design and suitably dimension the FFBSSs with dynamic relocation so that relocation costs could be completely covered by different kind of revenues – such as those resulting from advertising. However, aside from these considerations about revenues and costs, increasing the number of satisfied users and/or minimizing the number of lost users thanks to the proposed framework lead to higher revenues. They may be direct (i.e. deriving from greater incomes associated with memberships and user fees), and/or indirect (i.e. due to a higher user demand over the time, descending from a positive message that a well operating system communicates to people; or also due to greater advertising revenues).

8 Conclusions and future research

This paper deals with bike-sharing systems, in particular, we face the new free-floating ones: these systems basically do not need racks to lock bikes at predetermined stations, as they can be dropped off almost everywhere in the network without the capacity problem of traditional docking stations occurring.

Despite the undeniable advantages of these systems, during each operation day, bikes are inevitably scattered in an unbalanced way among the different districts of a city, and consequently, they need to be properly relocated to satisfy the user needs.

The relocation process in a Free-Floating Bike-Sharing System (FFBSS) needs additional measures to be effectively carried out with respect to a traditional BSS. To the best of our knowledge, there are only two recent papers that deal with this new kind of shared mobility, and our proposed methodology is the first approach, with a real sized case study application, to the dynamic relocation problem in FFBSSs. In particular, in this paper, we suggest a comprehensive method that could help the managers and operating companies with the real-time management of FFBSSs. First of all, we propose a spatio-temporal clusterization of the system under analysis, that aggregates data for better management of available resources. We conclude that the greater aggregation of zones could bring better results.

In a second step, it is necessary to forecast the trend of available bikes in each spatio-temporal cluster in order to understand how many bikes have to be repositioned to improve user satisfaction and system attraction. We propose the use of Nonlinear Autoregressive Neural Networks, training the system according to four possible starting datasets. From the operator's point of view, it seems that training the neural network according to the maximum value of available bikes in a given interval is a good compromise to relocate only the essential number of bikes, covering fewer kilometers with the carrier vehicles, globally leading to lower costs and satisfactory results for the system users.

Finally, the methodology presents a Decision Support System aimed at the maximization of user satisfaction, in terms of reducing the number of unfavorable times which a zone has fewer bikes than necessary, and a minimum number of lost users, who decide not to pick-up a bike because it is not available (at least, not nearby). Performing sensitivity analyses (and thanks to a demand and system simulator), we manage to have a real-time overview and control of the operation of the FFBSS with and without relocation, comparing different levels of picked-up bikes, different distribution between the zones of picked-up bikes and different neural network starting datasets.

We observe that it is possible to enhance a bike-sharing system by means of a dynamic relocation process. Such conclusions are supported by the results presented that summarize the percentage of improvement in terms of unfavorable times, lost users, and satisfied users. They are the results of a simulation model that depicts as closely as possible the real behavior of a bike-sharing system, taking into account its high degree of dynamicity, the unpredictable fluctuations in the trend of the bikes, and variation in the choices made by users.

Setting all the parameters involved accordingly to the case study, it could be possible finding good solutions to apply to different realities, providing operators with a valuable intervention strategy on their city. Future research developments may address the selection of other methods capable of further reducing costs of management / relocation of the system. In particular, although the use of the two proposed VRP optimizations has led to satisfactory results, further investigations could concern a comparison between our proposal and the application -specifically for FFBSSs- of the different dynamic VRP solving approaches that can be found in literature.

Furthermore, considering the strong inter-correlation between the network design of an FFBSS and the relocation process, we would also like to propose a methodology for optimizing the number of spatio-temporal clusters, integrating a planning and design model of the system with relocation decision support techniques that could help to improve the operation and management of the system.

Finally, future developments may focus on models aiming at optimally dimensioning the FFBSS in terms of number of bicycles, number of carrier vehicles, relocation staff, etc. The goal of these models may be the maximization of net revenues, considering the dynamic relocation costs together with the other potential income sources and the positive effects on the system deriving from the relocation.

Acknowledgements

This research has been carried out within the co-funded intervention “Fondo di Sviluppo e Coesione 2007-2013 – APQ Ricerca Regione Puglia, Programma regionale a sostegno della specializzazione intelligente e della sostenibilità sociale ed ambientale – FutureInResearch”, project No. 3T23CY7.

References

Antoniadis, A., Brossat, X.: Clustering functional data using wavelets. In: International Journal of Wavelets, Multiresolution and Information Process 11(1), 1350003-1–1350003-30 (2013)

Arthur, D., Sergi, V.: K-means++: The Advantages of Careful Seeding. In: SODA ‘07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 1027-1035 (2007)

Benarbia, T., Labadi, K., Omari, A., Barbot, J.: Balancing Dynamic Bike-Sharing Systems: A Petri Nets with Variable Arc Weights Based Approach. In: 2013 International Conference on Control, Decision and Information Technologies (CoDIT), 112-117 (2013)

Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., Jenssen, R. An overview and comparative analysis of Recurrent Neural Networks for Short Term Load Forecasting. arXiv preprint arXiv:1705.04378 (2017)

Caggiani, L., Ottomanelli, M.: A modular Soft Computing Based Method for Vehicles Repositioning in Bike-Sharing Systems. In: Procedia-Social and Behavioral Sciences 54, 675-684 (2012)

Caggiani, L., Ottomanelli, M.: A Dynamic Simulation based Model for Optimal Fleet Repositioning in Bike-sharing Systems. In: Procedia - Social and Behavioral Sciences 87, 203-210 (2013)

Caggiani, L., Ottomanelli, M., Camporeale, R., Binetti, M.: Spatio-temporal Clustering and Forecasting Method for Free-Floating Bike Sharing Systems. In: Advances in Intelligent Systems and Computing: Advances in Systems Science 539, J. Swiatek, and J. M. Tomczak, eds., Springer International Publishing, Cham, 244-254 (2017)

Chemla, D., Meunier, F., Calvo, R. W.: Bike Sharing Systems: Solving the Static Rebalancing Problem. In: Discrete Optimization 10(2), 120-146 (2013)

Chemla, D., Meunier, F., Pradeau, T., Calvo, R. W., Yahiaoui, H.: Self-Service Bike Sharing Systems: Simulation, Repositioning, Pricing. Technical report (Hyper Articles en Ligne - HAL) (2013)

Cheu, R. L., Xu, J. X., Kek, A. G. H., Lim, W. P., Chen, W. L.: Forecasting Shared-Use Vehicle Trips with Neural Networks and Support Vector Machines. In: Transportation Research Record: Journal of the Transportation Research Board 1968, 40-46 (2006)

Côme, E., Randriamanamihaga, A., Oukhellou, L.: Spatio-temporal Usage Pattern Analysis of the Paris Shared Bicycle Scheme: A Data Mining Approach. In: Transport Research Arena (TRA) 5th Conference: Transport Solutions from Research to Deployment. Paris (2014)

Connor, J., Atlas, L. E., Martin, D.R.: Recurrent networks and NARMA modeling. In: Advances in Neural Information Processing Systems 4, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. San Mateo, CA: Morgan Kaufmann, 301-308 (1992)

Contardo, C., Morency, C., Rousseau, L. M.: Balancing a Dynamic Public Bike-Sharing System. Technical report 4, Cirrelt (2012)

Dell'Amico, M., Hadjicostantinou, E., Iori, M., Novellani, S.: The Bike Sharing Rebalancing Problem: Mathematical Formulations and Benchmark Instances. In: Omega 45, 7-19 (2014)

Dougherty, M.: A Review of Neural Networks Applied to Transport. In: Transportation Research Part C 3(4), 247-260 (1995)

Erdoğan, G., Battarra, M., Calvo, R. W.: An Exact Algorithm for the Static Rebalancing Problem Arising in Bicycle Sharing Systems. In: European Journal of Operational Research 245(3), 667-679 (2015)

Forma, I. A., Raviv, T., Tzur, M.: A 3-Step Math Heuristic for the Static Repositioning Problem in Bike-Sharing Systems. In: Transportation Research Part B: Methodological 71, 230-247 (2015)

Fricker, C., Gast, N.: Incentives and Redistribution in Homogeneous Bike-Sharing Systems with Stations of Finite Capacity. In: EURO Journal on Transportation and Logistics 5(3), 261-291 (2016)

Froehlich, J., Neumann J., Oliver, N.: Sensing and Predicting the Pulse of the City Through Shared Bicycling. In: The International Joint Conferences on Artificial Intelligence 9, 1420-1426 (2009)

Garcia-Gutierrez, J., Romero-Torres, J., Gaytan-Iniestra, J.: Dimensioning of a Bike Sharing System (BSS): A Study Case in Nezahualcoyotl, Mexico. In: Procedia-Social and Behavioral Sciences 162, 253-262 (2014)

García-Palomares, J. C., Gutiérrez, J., Latorre, M.: Optimizing the Location of Stations in Bike-Sharing Programs: A GIS Approach. In: Applied Geography 35(1-2), 235-246 (2012)

Guan, L., Feng, X.: Research on Factors of the Signal De-Noising Effect Basing on the Wavelet Transform and the Matlab practice. In: Automation & Instrumentation 6, 43-46 (2004)

Hagan, M. T., Menhaj, M. B.: Training feedforward networks with the Marquardt algorithm. In: IEEE transactions on Neural Networks 5(6), 989-993 (1994)

Han, Y., Côme, E., Oukhellou, L.: Towards Bicycle Demand Prediction of Large-Scale Bicycle Sharing System. In: Transportation Research Board 93rd Annual Meeting No. 14-2637 (2014)

Ho, S. C., Szeto, W. Y.: Iterated Tabu Search for the Bike Repositioning Problem. In: INFORMS Annual Meeting 2014, San Francisco, CA., 9-12 November (2014a)

Ho, S. C., Szeto, W. Y.: Solving a Static Repositioning Problem in Bike-Sharing Systems Using Iterated Tabu Search. In: Transportation Research Part E: Logistics and Transportation Review 69, 180-198 (2014b)

Ho, S. C., Szeto, W. Y.: A hybrid large neighborhood search for the static multi-vehicle bike-repositioning problem. In: Transportation Research Part B 95, 340-363 (2017)

Kadri, A. A., Kacem, I., Labadi, K.: A Branch-and-Bound Algorithm for Solving the Static Rebalancing Problem in Bicycle-Sharing Systems. In: Computers & Industrial Engineering 95, 41-52 (2016)

Kaur, M., Kaur, U.: Comparison between k-means and hierarchical algorithm using query redirection. In: International Journal of Advanced Research in Computer Science and Software Engineering 3(7), 1454-1459 (2013)

Kek, A. G. H., Cheu, R. L., Xu, J.: Trip Forecasting Models for Intelligent Community Vehicle Systems. In: Proceedings of the 84th Annual Meeting of the Transportation Research Board, Washington, D.C., CD-ROM (2005)

Krause, E. F. Taxicab geometry. The Mathematics Teacher, 66(8), 695-706 (1973)

Lee, C., Wang D., Wong A.: Forecasting Utilization in City Bike-Share Program. Technical report, CS 229 2014 Project (2014)

Leontaritis, I., Billings, S.: Input-output parametric models for nonlinear systems: Part I: deterministic non-linear systems. In: Internal Journal of Control 41(2), 303-328 (1985)

Li, Y., Szeto, W. Y., Long, J., Shui, C. S.: A Multiple Type Bike Repositioning Problem. In: Transportation Research Part B: Methodological 90, 263-278 (2016)

Lin, J. R., Yang, T. H.: Strategic Design of Public Bicycle Sharing Systems with Service Level Constraints. In: Transportation Research Part E: Logistics and Transportation Review 47(2), 284-294 (2011)

Lloyd, S. P.: Least Squares Quantization in PCM. In: IEEE Transactions on Information Theory 28, 129-137 (1982)

Lu, C. C.: Robust Multi-Period Fleet Allocation Models for Bike-Sharing Systems. In: Networks and Spatial Economics 16(1), 61-82 (2016) Lu, C. C. (2016)

MacQueen, J. B.: Some Methods for classification and Analysis of Multivariate Observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 281-297 (1967)

Martinez, L. M., Caetano, L., Eiró, T., Cruz, F.: An Optimisation Algorithm to Establish the Location of Stations of a Mixed Fleet Biking System: An Application to the City of Lisbon. In: Procedia - Social and Behavioral Sciences 54, 513-524 (2012)

Midgley, P.: Bicycle-sharing schemes: enhancing sustainable mobility in urban areas. United Nations, Department of Economic and Social Affairs, 1-12 (2011)

Mori, U., Mendiburu, A., Álvarez, M., Lozano, J. A.: A Review of Travel Time Estimation and Forecasting for Advanced Traveller Information Systems. In: Transportmetrica A: Transport Science 11 (2), 119-157 (2015)

Pal, A., Zhang, Y.: Free-floating bike sharing: Solving real-life large-scale static rebalancing problems. In: Transportation Research Part C: Emerging Technologies 80, 92-116 (2017)

Pfrommer, J., Warrington, J., Schildbach, G., Morari, M.: Dynamic Vehicle Redistribution and Online Price Incentives in Shared Mobility Systems. In: IEEE Transactions on Intelligent Transportation Systems 15(4), 1567-1578 (2014)

Raviv, T., Tzur, M., Forma, I. A.: Static Repositioning in a Bike-Sharing System: Models and Solution Approaches. In: EURO Journal on Transportation and Logistics 2(3), 187-229 (2013)

Regue, R., Recker, W.: Proactive Vehicle Routing with Inferred Demand to Solve the Bikesharing Rebalancing Problem. In: Transportation Research Part E: Logistics and Transportation Review 72, 192-209 (2014)

Reiss, S., Bogenberger, K.: GPS-Data Analysis of Munich's Free-Floating Bike Sharing System and Application of an Operator-based Relocation Strategy. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems IEEE, 584-589 (2015)

Reiss, S., Bogenberger, K.: Validation of a Relocation Strategy for Munich's Bike Sharing System. In: Transportation Research Procedia 19, 341-349 (2016)

Romero, J. P., Ibeas, A., Moura, J. L., Benavente, J., Alonso, B.: A Simulation-Optimization Approach to Design Efficient Systems of Bike-sharing. In: Procedia - Social and Behavioral Sciences 54, 646-655 (2012)

Saharidis, G., Fragkogios, A., Zygouri, E.: A Multi-Periodic Optimization Modeling Approach for the Establishment of a Bike Sharing Network: A Case Study of the City of Athens. In: Proceedings of the International MultiConference of Engineers and Computer Scientists (2014)

Sapankevych, N., Sankar, R.: Time Series Prediction Using Support Vector Machines: A Survey. In: IEEE Intelligent Systems 4(2), 24–38 (2009)

Schuijbroek, J., Hampshire, R. C.,Van Hoeve, W. J.: Inventory Rebalancing and Vehicle Routing in Bike Sharing Systems. In: European Journal of Operational Research, 257(3), 992-1004 (2017).

Shui, C. S., Szeto, W. Y.: An Artificial Bee Colony Algorithm for Public Bike Repositioning Problem. In Australasian Transport Research Forum, ATRF 2015. The National Academies of Sciences, Engineering, and Medicine (2015)

Singh A., Yadav A., Rana A.: K-means with Three Different Distance Metrics. In: International Journal of Computer Applications 67(10), 13-17 (2013).

Smith, B., Demetsky, M.: Traffic Flow Forecasting: Comparison of Modeling Approaches. In: Journal of Transportation Engineering 123(4), 261-266 (1997)

SocialBycicle (SoBi) website: http://socialbicycles.com (accessed on July 2017)

Sun, Y., Leng, B., Guan W.: A Novel Wavelet-SVM Short-Time Passenger Flow Prediction in Beijing Subway System. In: Neurocomputing 166, 109-121 (2015)

Szeto, W. Y., Liu, Y., Ho, S. C.: Chemical Reaction Optimization for Solving a Static Bike Repositioning Problem. In: Transportation Research Part D: Transport and Environment 47, 104-135 (2016)

Vlachos, M., Lin, J., Keogh, E., Gunopulos, D.: A wavelet-based anytime algorithm for k-means clustering of time series. In: Proceedings of the Workshop on Clustering High Dimensionality Data and Its Applications (2003)

Vlahogianni, E. I., Karlaftis, M. G., Golias, J. C.: Short-term Traffic Forecasting: Where We Are and Where We’re Going. In: Transportation Research Part C 43, 3-19 (2014)

Vogel, P., Greiser, T., Mattfeld D. C.: Understanding Bike-Sharing Systems using Data Mining: Exploring Activity Patterns. In: Procedia-Social and Behavioral Sciences 20, 514-523 (2011)

Vogel, P., Mattfeld, D. C.: Strategic and Operational Planning of Bike-Sharing Systems by Data Mining - A Case Study. In: Computational Logistics, 127-141. Springer (2011)

Vogel, P., Saavedra, B. A. N., Mattfeld, D. C.: A Hybrid Metaheuristic to Solve the Resource Allocation Problem in Bike Sharing systems. In: Hybrid Metaheuristics 8457, 16-29. Springer (2014)

Wang, X., Lindsey, G., Schoner, J. E, Harrison, A.: Modeling Bike Share Station Activity: The Effects of Nearby Business and Jobs on Trips to and from Stations. In: Journal of Urban Planning and Development 142(1), 04015001 (2015)

Wei, Y., Chen, M.: Forecasting the Short-Term Metro Passenger Flow with Empirical Model Decomposition and Neural Networks. In: Transportation Research Part C 21(1), 148-162 (2012)

Xu, H., Ying, J., Wu, H., Lin, F.: Public Bicycle Traffic Flow Prediction Based on a Hybrid Model. In: Applied Mathematics & Information Sciences 7, 667-674 (2013)

Yu, H., Yang, J., Han, J., Li, X.: Making SVMs Scalable to Large Datasets using Hierarchical Cluster Indexing. In: Data Mining and Knowledge Discovery 11(3), 295-321 (2005)

Zeng, D., Xu, J., Gu, J., Liu, L., Xu, G.: Short Term Traffic Flow Prediction Based on Online Learning SVR. Workshop on Power Electronics and Intelligent Transportation System, 616-620 (2008)

Zhang, Z., Zhang, P., Yin, Y., Hou, L.: Analysis on Urban Traffic Network States Evolution Based on Grid Clustering and Wavelet De-noising. In: 11th International IEEE Conference on Intelligent Transportation Systems, Beijing, 1183-1188 (2008)